KATHOLIEKE UNIVERSITEIT
LEUVEN

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science

Towards Story Understanding and Search

Web Mining Methods and Tools for Exploration, Search, and Discovery

Ilija SUBAŠIĆ

Dissertation presented in partial
fulfillment of the requirements for
the degree of Doctor
in Engineering

December 2011

Towards Story Understanding and Search

Web Mining Methods and Tools for Exploration, Search, and Discovery

Ilija SUBAŠIĆ

Jury:
Prof. dr. ir. Jean Berlamont, chair
Prof. dr. Bettina Berendt, promotor
Prof. dr. ir. Hendrik Blockeel
Prof. dr. ir. Erik Duval
Prof. dr. Dave Clarke
dr. Joris Klerkx

Prof. dr. Arjen P. de Vries
(Centrum Wiskunde & Informatica Amsterdam/TU Delft)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering

December 2011

D/2011/7515/157
ISBN 978-94-6018-457-4

Acknowledgements

I would like to thank the members of the Examination Committee (thesis jury) of this thesis: prof. dr. Bettina Berendt, who was the promotor of this thesis, prof. dr. ir. Erik Duval and prof. dr. ir. Hendrik Blockeel, who were also members of the Supervisory Committee, prof. dr. Dave Clarke, dr. Joris Klerkx, and prof. dr. Arjen P. de Vries for taking their time to evaluate this thesis and provide valuable feedback for the improvement of the final version and ideas for future work. Also thanks to prof. dr. ir. Jean Berlamont for serving as a chairman of the jury.

A special thanks goes to two of my K.U. colleagues, Sten Govaerts and Nik Corthaut. Sharing the office with Nik and Sten for over 3 years, created an atmosphere one rarely encounters in a workplace anywhere, and especially in a foreign country. Thanks to them, I leave Belgium having a much different experience than most graduate students have. My thanks also go to Mathias Verbeke (especially for the Dutch translation!) and Seda Gürses for sharing and understanding the process of doing the thesis inside this particular environment. Part of this thesis was done during my internship at Yahoo! Research, Barcelona, and I would like to thank Carlos Castillo – ChaTo, both for his help in my work, and the hospitality during my stay in the lab. The same also goes for Aris Gionis, and the rest of the lab. I would also like to thank all of the former HMDB members for their help on a number of issues. Another special thanks goes out to Tias Guns, who helped me a lot with the layout of this thesis.

One bunch of people deserves a special "thank you" section. The "old-country" folks here in Leuven really alleviated the nostalgia feelings. Thanks to (in no particular order): Mire, Dule&Milica, Bogi, Miki&Dara, Bane&Marina, Vulić, Ivan G., Tićma, Aćko, and Vukov for the basketball games, barbecues, improving my otherwise microwave-based diet, nights out, or just plain hanging around.

To my family, I do not have to say any special thanks. They already know that without them all of this would be much more difficult, and that even people with writing skills much greater than mine can not really express how much I am grateful to my family. Anyway, so that nobody gets offended, an official big thanks to my family for everything they provided me with.

This acknowledgement was written in haste, and if somebody that i should have thanked is not mentioned, it was not done intentionally. One big thank you to all of you.

Preface

At the very beginning of this thesis it is important to note that this thesis is submitted in the format of a doctoral thesis based on publications. As such, except for the first two chapters which contain the introduction and an overview of related work, and the last chapter which concludes the thesis, all chapters contain one published (or accepted for publication) scientific paper. Each of these chapters starts with a cover page that gives bibliographic data about the publication and the contributions made by the author of the thesis.

Following papers are included in the thesis:

- Chapter 3 – *Ilija Subašić and Carlos Castillo: Investigating Query Bursts in a Web Search Engine - What happens when something happens? Invited for publication in Web Intelligence and Agent Systems: An International Journal (WIAS) (conditionally accepted August 2011, revised version sent September 2011).*[1]
- Chapter 4 – *Ilija Subašić and Bettina Berendt: Discovery of interactive graphs for understanding and searching time-indexed corpora. Knowledge and Information Systems 23(3): 293-319 (2010).*
- Chapter 5 – *Ilija Subašić and Bettina Berendt: Story Graphs: Tracking document set evolution using dynamic graphs. Intelligent Data Analysis Journal, special issue on Dynamic Networks and Knowledge Discovery. Vol. 17(1), 2013. IOS Press. Accepted for publication June 2011, publication scheduled for 2013.*
- Chapter 6 – *Ilija Subašić and Bettina Berendt: From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In Proceeding of the 19th European Conference on Artificial Intelligence*

[1] A shorter version of the paper was published as: Ilija Subašić and Carlos Castillo. 2010. The Effects of Query Bursts on Web Search. Volume 01 (WI-IAT '10), Vol. 1. IEEE Computer Society, Washington, DC, USA, 374-381.

(ECAI 2010), pages 517-522, Amsterdam, The Netherlands, 2010. IOS Press.

- Chapter 7 – *Ilija Subašić and Bettina Berendt: Interactive evaluation of interfaces for story tracking. HCIR 2011: The Fifth Workshop on Human-Computer Interaction and Information Retrieval (October 2011), Electronic publication, no pages.*[2]

- Chapter 8 – *Ilija Subašić and Bettina Berendt: Peddling or Creating? Investigating the Role of Twitter in News Reporting. In Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR'11). Springer-Verlag LNCS, Berlin, Heidelberg, 207-213.*

In addition, Appendix A includes a shorter paper following a demonstration of the tool we developed:

- *Ilija Subašić and Bettina Berendt. Experience stories: A visual news search and summarization system. In Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD 2010), Springer-Verlag LNCS, Berlin, Heidelberg, 619-623. 2010.*

Note. All papers are included as they were published, with only formatting changes. This results in a layout that is different from the original publications. The papers include typographic errors and are followed by an errata section marking these errors.

[2] `https://sites.google.com/site/hcirworkshop/hcir-2011/posters`

Abstract

Over the past decade the Internet became one of the leading sources of news content, and using different news provider services available on the Internet has for many people become the main medium for staying informed about the world. Such services support Internet users in interaction with *stories*. In this thesis, we regard a story as a set of time-stamped documents describing correlated subjects, such as for example persons, event descriptions, and topics. Our particular interest is to investigate the time dimension of stories and particularly *story tracking* – following a story over time. The goal of different research areas interested in story tracking is to identify and highlight developments – novel and relevant information in a story. In this work we restrict ourselves to news collections and investigate effectiveness and usability of temporal text mining (TTM) story tracking methods.

Across the thesis we investigate four areas related to stories: (a) stories and search engines; (b) story tracking methods and tools, (c) story tracking evaluation frameworks, and (d) stories and sources. We formalize these 4 thematic areas into more concrete research questions addressed in this thesis: (Q1) How are search engines affected by story developments? (Q2) Does the semi-automatic story tracking approach we developed enable user comprehension and navigation of stories? (Q3) Can the graph-based patterns extracted by our algorithm be used for story tracking? (Q4) How can different bursty text patterns be used for discovering origins of the changes in document sets? (Q5) How do users interact with interfaces for story tracking? (Q6): How to measure differences between a story across different sources?

We start by exploring how search engine users change their behaviour when new developments emerge in a story. For this we investigate a one-year long query log from a leading commercial search engine, and describe the changes of user behaviour correlated with the emergence of new developments. Then, we continue by exploring story tracking methods and tools as means for accommodating for these changes in user behaviour. We propose a

new, graph-based, story tracking method and build a tool to support it. Additionally, we investigate the effectiveness of story tracking methods and define a new framework for automatic and user oriented evaluation. Although there are many TTM methods developed, there is a lack of common evaluation procedure. We propose an evaluation framework for measuring how different TTM methods discover novel developments. Apart from the automatic evaluation we are interested in how users interact with patterns and learn about the developments of the story they track. For this we propose a set of metrics and procedures for evaluation of user interfaces in the context of story tracking. To test our tool, we conducted a user study of four interfaces in the context of story tracking. Finally, we look at the source dimension of stories and explore the possible differences in news reporting across different families of news sources, and how to measure them.

The results of our analysis show that our method is comparable in performance to other TTM methods, and that it meets the requirements for story tracking. We also show that by leveraging the pattern structure and sentence retrieval TTM methods can help discover developments in the news domain. The user study results show that users have a preference for our tool compared to the rest of the tools used in the study. They also point out that the tool meets a number of the requirements discovered in the query log analysis.

Beknopte samenvatting

In de afgelopen jaren is het Internet één van de belangrijkste bronnen voor nieuws geworden. Het gebruik van verschillende online nieuwsdiensten is voor veel mensen dan ook het voornaamste medium geworden om op de hoogte te blijven van de actualiteit. Dergelijke diensten ondersteunen Internetgebruikers in interactie met *verhalen*. In dit doctoraat zien we een verhaal als een verzameling van documenten met een tijdsaanduiding die gerelateerde onderwerpen beschrijven, en handelen over personen of verslaggeven over gebeurtenissen en andere onderwerpen. Onze bijzondere interesse ligt in het onderzoeken van de tijdsdimensie van deze verhalen in het algemeen, en *story tracking*, i.e. het volgen van de verhaallijn doorheen de tijd, in het bijzonder. Het doel van verschillende onderzoeksdomeinen met interesse in story tracking is om de ontwikkelingen van verhalen, i.e. nieuwe en relevante informatie, te identificeren en te beklemtonen. In dit werk beperken we ons tot nieuwscollecties en onderzoeken de effectiviteit en bruikbaarheid van temporal text mining (TTM) story tracking methodes.

Doorheen het doctoraat onderzoeken we vier domeinen gerelateerd aan verhalen: (a) verhalen en zoekmachines; (b) story tracking methodes en tools; (c) raamwerken voor de evaluatie van story tracking, en (d) verhalen en bronnen. We formaliseren deze 4 thematische domeinen in de volgende, meer concrete onderzoeksvragen, die we in deze thesis beantwoorden: (Q1) Hoe worden zoekmachines beïnvloed door de ontwikkelingen in verhalen? (Q2) Stelt de semi-automatische methode voor story tracking die we ontwikkelden de gebruiker in staat om de verhalen te begrijpen en er in te navigeren? (Q3) Kunnen de graafgebaseerde patronen die geëxtraheerd worden door ons algoritme, gebruikt worden voor story tracking? (Q4) Hoe kunnen verschillende piekmomenten in tekstpatronen gebruikt worden voor het ontdekken van veranderingen in verzamelingen van documenten? (Q5) Hoe kunnen de verschillen tussen een verhaal afkomstig van verschillende bronnen gemeten worden?

We starten met het verkennen hoe verschillende gebruikers van zoekmachines hun gedrag veranderen wanneer nieuwe ontwikkelingen zich voordoen in een verhaal. Hiervoor onderzoeken we een query log van een toonaangevende zoekmachine over de periode van een jaar, en beschrijven de veranderingen in het gedrag van de gebruikers bij het optreden van nieuwe gebeurtenissen. Vervolgens verkennen we story tracking methodes en tools als een middel om te voorzien in deze veranderingen in het gedrag van gebruikers. We stellen een nieuwe, graafgebaseerde story tracking methode voor en ontwikkelen een tool om dit te ondersteunen. Bovendien onderzoeken we de effectiviteit van story tracking methodes en definiëren een nieuw raamwerk voor een automatische en gebruikersgeoriënteerde evaluatie. Hoewel er verschillende TTM methodes ontwikkeld zijn, is er een gebrek aan een gemeenschappelijke evaluatieprocedure. We stellen een evaluatieraamwerk voor om te meten hoe verschillende TTM methodes nieuwe ontwikkelingen ontdekken. Afgezien van de automatische evaluatie zijn we geïnteresseerd in hoe gebruikers interageren met patronen en leren over de ontwikkelingen in het verhaal dat ze volgen. Hiertoe voorzien we een verzameling metrieken en procedures voor de evaluatie van gebruikersinterfaces in de context van story tracking. Om onze tool te testen, voerden we een gebruikersstudie uit met vier interfaces in de context van story tracking.

Tenslotte kijken we naar de bron van verhalen, onderzoeken we wat de mogelijke verschillen zijn in de verslaggeving van nieuwsfeiten tussen verschillende nieuwsbronnen, en hoe we deze kunnen meten.

De resultaten van onze analyse tonen aan dat onze methode vergelijkbaar is met andere TTM methodes, en het tegemoet komt aan de vereisten voor story tracking. We tonen ook aan dat, door gebruik te maken van de patroonstructuur en zinsanalyse, TTM methodes gebruikt kunnen worden voor het ontdekken van ontwikkelingen in het nieuwsdomein. De resultaten van het gebruikersonderzoek tonen aan dat gebruikers een voorkeur hebben voor onze tool in vergelijking met de andere tools die gebruikt werden in het gebruikersonderzoek. Ze wijzen er ook op dat de tool voldoet aan een aantal vereisten die we ontdekten in de analyse van de query logs.

List of Abbreviations

AQE	automatic query expansion;
ASCII	American Standard Code for Information Interchange;
BP	British Petrol;
cf.	confer (compare);
ciQA	complex interactive question answering;
DCS	Document Cluster Servers;
DM	data mining;
DET	detection tradeoff curve;
DOM	Document Object Model;
DUC	Document Understanding Conference;
ect.	et cetera (and so forth);
et al.	et alii (and others);
ETP3	evolutionary theme pattern discovery, summary and exploration;
EU	European Union;
e.g.	exampli gratia (for example);
Fig.	Figure;
FSD	First Story Detection;
GE	global eventfulness;
HARD	High Accuracy Retrieval of Documents;
HCI	human computer interaction;
HTML	hyper-text markup language;
HSD	Honestly Significant Difference;
ICS	intra-cluster similarity;
idf	inverse document frequency;
IE	information extraction;
INEX	Initiative for the Evaluation of XML retrieval;
IR	information retrieval;
IIR	interactive information retrieval;
JM	Jelinek-Mercer language model smoothing;
JS	Jensen-Shannon divergence;

KL	Kullback-Leibler divergence;
LDA	latent Dirichlet allocation;
LE	local eventfulness;
LR	local relevance;
MDS	multidimensional scaling;
NBA	National Basketball Association;
NE	named-entities;
NER	named-entity recognition;
NIST	National Institute of Standards and Technology;
NLP	natural-language processing;
ONED	On-line New Event Detection;
pLSA	probabilistic latent semantic analysis;
QL	query-likelihood retrieval ;
TDT	Topic Detection and Tracking;
tf.idf	term frequency-inverse document frequency;
TF (tf)	term frequency;
TR	time relevance;
TREC	Text REtrieval Conference;
TTM	temporal text mining;
ROUGE	Recall-oriented Understudy for Gisting Evaluation;
RSS	really simple syndication (RDF site summarization);
U.S. (US)	United States of America;
URL	uniform resource locator;
Web	world wide web;
www	world wide web;
W3C	World Wide Web Consortium;
XML	eXtensible Markup Language.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Beginning with the oral history transferred from one generation to another, and continuing with cave paintings, early manuscripts, and the invention and development of printing, up to the fully digital content creation and distribution, human societies have been creating *stories* – records of the world around them. In the same way *story spaces* – media through which stories are distributed, evolved from story-telling around fires to digital sharing and personalized content delivery. In this thesis we explore how several aspects of stories transfer to the Internet era, and how can we facilitate users in accessing, following, and learning about stories in online story spaces. Although the term "story" is often used to reference descriptions of fictional events, in this thesis we investigate only those stories describing the "real world". Obviously, the development of the Internet, and especially the Web, over the last decades has created an environment which provides access to stories with ease and in numbers as never before in the history of mankind. The Web has become one of the main sources for stories providing readers (users) with rich and diverse story spaces. In this thesis we employ an array of data mining techniques to Web data with the goal to explore *changes, navigation, interaction, comprehension, and relations* in online story spaces.

1.1 Stories and Online Story Spaces

In the context of this thesis we define stories as *sets of time-stamped theme-related documents*, and story spaces as channels that provide access to stories. RSS-feeds, search engines, forums, digital repositories and similar services

provide access to a vast number of documents such as news reports, blog posts, scientific publications, or personal status messages. Many of these services serve as story spaces and allow users to dissect the available corpora based on themes documents cover and the times they were published. We call the underlying theme that documents in a story share a *story theme*. A search engine user can limit the results of his search to documents published in a selected time interval, and a query can be regarded as a story theme shared by all documents that search engine finds relevant to this query. For example, we can regard all news reports returned by a news search engine for the query *2011 earthquake in Japan* over time as a story about the devastating earthquake in Japan. Similarly, we can regard a collection of blog posts discussing a topic over time (e.g. "linux vs windows") or a set of scientific papers describing work in a specific area (e.g. "data mining") as stories. The largest[1] search service provider Google, Yahoo!, and Bing enable users to search for news, blog posts, and scientific publications based on the publishing time of the documents. Similarly, users can easily subscribe to numerous RSS feeds providing them with a constant influx of newly published documents having a common theme or originating from a single or related source.

1.2 Research Problems and Questions

The main research thread throughout most of this thesis explores changes along the temporal dimension of stories. Stories are in nature dynamic and develop over time. Users often follow a story over an extended period of time. We refer to this activity as *story tracking*. For example, a user can repeatedly query a news search engine, and each time limit his search only to documents published since his previous search. While tracking a story, users are interested in discovering novel developments regarding the subjects, topics, and events described in a story. Users expect to find both relevant and novel information represented in the documents. The key property of this kind of information is *burstiness* – surge in importance to the story theme. We refer to this information as *story developments* (developments in short). To facilitate discovery of developments we define the *story tracking task* as a task with a goal of extracting and presenting developments from stories.

In this thesis we investigate three important aspects of stories: distribution of stories to the users, user comprehension of stories over time, and diversity in story sources. More informally, we investigate how users explore stories, how different algorithms and interfaces help users comprehend the evolution of

[1] http://www.hitwise.com/us/press-center/press-releases/experian-hitwise-reports-bing-powered-share-of-s/ - retrieved April, 10th 2011

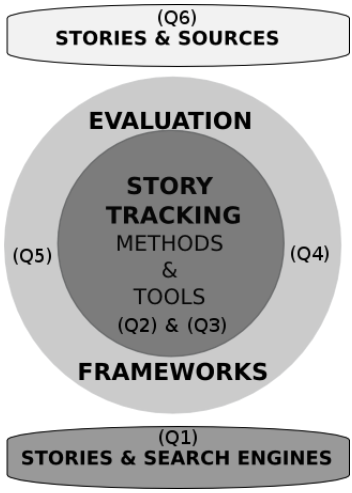


Figure 1.1: Overview of thematic areas investigated throughout this thesis and their relation to the research questions.

stories, and what the extent and type of diversity between reports on a same story originating from different sources is.

On a top level we explore the following 4 thematic areas related to stories on the Web: (1) stories and search engines, (2) story tracking methods and tools, (3) story tracking evaluation frameworks, and (4) stories and story sources. The work presented in this thesis falls into an overlap of three major research areas: *data mining*, *information retrieval*, and *human-computer interaction*. Figure 1.1 illustrates relations between general thematic areas we investigate throughout this thesis.

We formalize these 4 thematic areas into more concrete research questions addressed in this thesis:

- *Q1*:How are search engines affected by story developments?
- *Q2*: Does the semi-automatic story tracking approach we developed enable user comprehension and navigation of stories?
- *Q3*: Can the graph-based patterns extracted by our algorithm be used for story tracking?
- *Q4*: How can different bursty text patterns be used for discovering origins of the changes in document sets?

- *Q5*: How do users interact with interfaces for story tracking?
- *Q6*: How to measure differences between a story across different sources?

The first question ($Q1$) is addressed by the thematic area stories and search engines, questions $Q2$ and $Q3$ are investigated in story tracking methods and tools, while story tracking evaluation frameworks investigates questions $Q4$ and $Q5$. Finally, question $Q6$ is addressed by stories and sources theme.

Ordering remark. We start out the thesis by investigating the effects developments have on user behaviour in online story spaces, specifically in search engines. Then, we continue by exploring how developments can be tracked over time, and present a semi-automatic method and a tool based on local pattern extraction and graph visualization.

Although this ordering of questions and areas does not chronologically follow the progression of the work presented in this thesis, we consider that the proposed organization provides for an easier understanding of the thesis as a single body of research. Concretely, we started our research with the exploration of semi-automatic systems for story tracking ($Q2$). Upon the completion of this research, we were interested in the extent of a need for story tracking systems and proceeded with research related to question $Q1$. Based on the results of this analysis we continued our research as laid out in the previous paragraph.

Finally, we look at the stories not from the time aspect, but from the source aspect and investigate how to measure differences between story reports across news sources. Specifically, we are interested in a difference between the so-called social and the traditional media.

1.3 Stories and Search Engines

Search engines are one of the main online story spaces, used by around 30% of online news readers [12]. The usage analysis of search query logs can provide a valuable addition to understanding how users behave while exploring stories.

The emergence of novel developments may place a story theme into the spotlight and invoke higher attention of the users. For a search engine this is manifested as a sharp rise of frequency for a certain (story-related) query. This is often referred to as a *query burst*. The analysis of the effects query bursts have on search engines reveals how developments change the behaviour

of users while exploring stories. Our motivation for exploring these changes is as follows: if the emergence of new developments in a story causes changes in user behaviour, then the way of accessing documents should accommodate for these changes.

The assumption made in this section is that query bursts are caused by the emergence of new developments in a story, and that therefore any changes in user behaviour can be accredited to the developments. For example, after U.S. politician Sarah Palin was interviewed by her impersonator on the TV show Saturday Night Live there was a 22× increase in the frequency of the query "`snl sarah palin`".

For our analysis we used query logs from Yahoo![2] web-search and news-search logs. First, we focused on user activity during the burst and described changes in users' effort and interest while searching. We were particularly interested in what happens before and after a query burst differently than during the burst. To achieve this, we analyzed the effort and the attention of users while searching for bursty queries. Using several metrics we categorized these queries based on user activity. Then, we focused on a difference between query bursts in general web search logs and specific online-news search logs. Finally, we looked at the bursts from the general content providers' view. We investigated under which conditions content providers can "ride" a wave of increased interest on a topic, and obtain a share of the increased users' attention.

The most important results of our query burst study show that during the query burst users spend more effort into search and are more concentrated on a specific section of search results. A more detailed description of the analysis of bursty queries and results of this analysis is reported in Chapter 3.

1.4 Story Tracking

If user behaviour changes during the query burst there should be a difference in search interfaces presented to the users for bursty and non-bursty queries. Users are willing to invest more effort into search during the bursts. For example, we discovered that during the burst users tended to use more query assistance than before and after the burst. We also discovered that users tended to pick documents related to the developments that caused the burst, and access more documents. We explore story tracking methods and tools as a possible solution for accommodating these changes in user behaviour.

[2] http://www.yahoo.com

Our goal is to build systems that engage users into corpora exploration. Such systems provide an environment in which users act more as active researchers than passive readers.

1.4.1 Story Developments and Story Representation

Previously, in Section 1.2, we defined the task of story tracking as the extraction and presentation of developments from a story. The automatic extraction of the developments from stories is performed by a *story tracking method*. To capture developments, story tracking methods output a *story representation*. If we wish to capture and extract developments, first we need a way of defining how they are represented. Defining how developments are represented in natural language is a challenging task as developments are in many cases elusive, hidden in text, and hard to express. We refer to these expressions as *development representations*.

Depending on the genre of a story, developments can be expressed in a variety of ways. For example, story developments in the news domain can be expressed using natural language sentences. These sentences should describe topics, events, and subjects around which a story revolves. One development can be represented with a sentence such as *"The NBA locked out players on July 1st when the collective bargaining agreement expired"*. In a corpus of scientific publications, where topics of focus change over time, representing developments as sentences does not fully capture the semantics behind the developments. Similarly, for blog corpora, where users are interested in discovering novel opinions, stands, and sentiments of other bloggers, sentences do not fully capture the developments.

Story representations of story tracking methods can differ from development representations of a story. An obvious question that arises from this situation is whether a story representation should be in different form than a development representation? For example, why should story tracking methods output other representations if the developments are represented using sentences? We argue that story tracking is not only about discovering developments, but also about understanding and summarizing the changes in corpora, following how a story evolves over time, and providing hints when new developments occur. A story representation may provide users with a summarization of developments, freedom to explore developments, facilitate detecting the changes in corpora, and keep a way of linking to the same format as development representation. Therefore, having different development and story representations may improve the user experience during story tracking, moving away from "simple" retrieval into more engaging and intuitive ways of story tracking.

1.4.2 Story Tracking and Similar Tasks

The story tracking task has been tackled using various approaches.[3] Several *topic detection and tracking* (TDT) tasks [3, 2], most notably the First Story Detection task (FSD) [4], solve the story tracking task by outputting document(s) as story representation. The goal of FSD is to identify the first document (from a stream) discussing a novel development. *Update summarization* [1] uses temporal-oriented multi-document summaries as story representation, and *novel sentence retrieval* methods [14] retrieve the most salient and novel sentences which are used as story representation. These two approaches are closely related, but differ mostly in the way the task and evaluation frameworks were standardized. More recent methods have focused on mining for lower-level text elements. We refer to these approaches collectively as *temporal text mining* (TTM) [11, 8], and to the elements they extract as *bursty patterns* [9]. TTM methods hold much promise in terms of the additional flexibility afforded by the discovery of sub-sentential patterns.

One of the first distinctions between story tracking and similar tasks we wish to describe, is the difference between information retrieval and story tracking tasks. Information retrieval tasks (as defined in the TREC framework [22]) in all of their different formats (document retrieval, sentence retrieval, passage retrieval,...) are mostly defined as ad-hoc tasks whose goal is to retrieve the most relevant information (document, sentence, passage...) on a topic. Story tracking takes time into account, and searches for relevant information that is new. It differs from retrieval of the most relevant documents, and refines search results describing novel developments of a story.

Another highly similar task to story tracking is *timeline generation* [21]. Timeline generation aims to create a temporal ordering of important developments, independent of the time when developments become known (e.g. published in the media). Story tracking differs from this and summarizes developments based on the time when they become known, and not the actual time when the development occurred.

The focus of story tracking is on evolving corpora in which all documents have been related to the high-level theme a story follows. Detecting the emergence of new high-level story themes is not explored in this thesis.

[3] A more detailed description of related work can be found in Chapter 2 of this thesis.

1.4.3 Story Tracking Use Cases

We identify the two main use cases for story tracking: *online* and *retrospective* story tracking.

The goal of users in the online story tracking use case is to stay aware of recent changes in a story. This is a use case in which users are exposed to a stream of documents, and their information need is to understand the novel developments. For example, a user is subscribed to a RSS feed which aggregates documents belonging to the same story. Each time she logs on to a system, it provides an overview of the developments from the time of her last log-in (or request).

In the retrospective use case, users are oriented towards the past and the system provides users with an overview of the story over time. In this scenario documents are stored in some archive, and users explore this archive with the goal of understanding how the story developed over time. For example, a news web site publishes a document describing some event and users wish to explore previously published documents of the same story. In difference to the on-line use case, in this use case typical users are either professionals (e.g. journalists witting a story, researchers moving into a new research area) or pro-am users (e.g. a blogger following reports on a topic related to his blog) wishing to gain a deeper perspective on the evolution of a story.

Although both use cases are closely related, in this thesis we mostly concentrate on the retrospective use case. This was mostly due to the available data, but the method and the tool we developed can easily be applied to the on-line use case as well.

1.4.4 Method, Tool, and Example

We explore two goals users have while story tracking: (1) *story understanding* and (2) *story search*. The goal of story understanding is to comprehend the story's developments and track their evolution. In order to achieve this, users will want to inspect the documents of a story (story search). Here, finding the most relevant documents is only a means to the (generally more important) end of discovering the developments and their evolution. This situation calls for systems that: (a) identify developments in a story, (b) show how these developments emerge, change, and disappear (and maybe re-appear) over time, and (c) give users intuitive interfaces for interactively exploring the story landscape and at the same time the underlying documents. Such system should not expose users to well-formatted, predefined and global patterns from

a machine intelligence system, but should treat users as its equally important part. We approach story tracking as an interactive task. Specifically, in this thesis we concentrate on temporal text mining story tracking methods and combine them with visual document-search interfaces.

Following this idea, we developed an interactive semi-automatic system for story tracking described in Chapter 4. Our approach is based on the interaction between users and the patterns produced by the story tracking method via a story tracking tool. We consider story tracking as a task demanding synergy of both the method and the tool supporting it. The method side of story tracking has to tackle three sub-tasks, referred to as the components, and produce such temporal patterns that: (a) are *understandable* to human readers, (b) can be used to *detect* the emergence of new developments, and (c) can be used to *discover* details behind these developments. For story tracking tools we identified 3 sub-tasks which should allow users to: (i) *track* the developments of the story, (ii) *discover* the original representation of the developments, and (iii) *learn* the context and details around the story user track.

To solve the method and tool sub-tasks we employ a *graph-based* approach. Using graphs on one side allows us to utilize the graph structure in order to extract development representations, detect the emergence of novel developments and link them with the appropriate representation. On the other side, it has been shown [20, 5] that graph-based visualizations create eye-pleasing and understandable pictures for document search and corpora exploration.

We developed a method for bursty-patterns extraction based on the frequency lift of normalized bi-gram co-occurrences inside a window of words. The extracted patterns are used to build a graph-based visual summarization. We refer to these graphs as *story graphs*. Interacting with story graphs enables users to build up the topics they are interested in, and discover the "facts" behind the developments by retrieving sentences or documents. To support these interactions we created a web based tool named *STORIES* (see Appendix A).

An example story graph is shown in Figure 1.2. The graph summarizes the main questions in a news report: *who* (missing child as a central node in the largest component, and the parents in the top right corner), *what* ("disappear" and "miss" nodes are linked to the child), and *where* ("holiday" and "apartment" are linked to the child, and the country is shown in the top left corner). Another important news question, *when*, is implied by the period users explore.

In-depth descriptions of the story tracking method is presented in the following chapters:

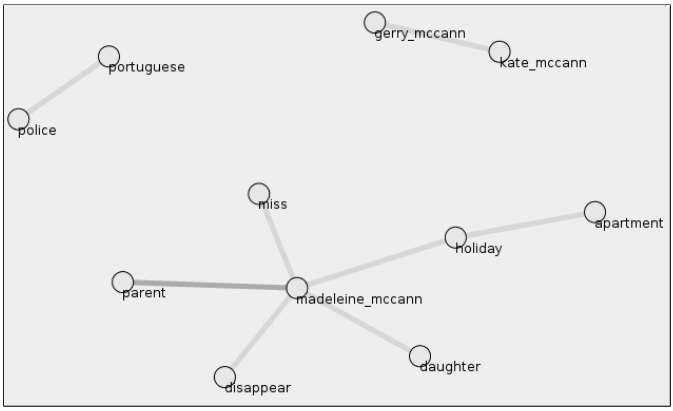


Figure 1.2: An example story graph. The concrete story revolves around a missing child case.

- In Chapter 4 we explore extraction, visualization, and comprehension of story graphs, a problem we call *Evolutionary Theme Pattern Discovery, Summary and Exploration (ETP3).*
- In Chapter 5 we explore how changes in the properties of story graphs can be used as indicators for detecting the emergence of new developments. We investigated whether it is possible to use topological properties of the story graphs for pointing users to more promising parts of the story space. The results indicated that global properties of the graphs are useful for this purpose.
- In Chapter 6 we explore how to link graph-based story representation to sentence development representation. To this aim we developed a framework that transforms the TTM produced story representations in queries used for sentence retrieval.

The STORIES tool is described in Apendix A of the thesis, and some descriptions of earlier versions of the tool can be found in Chapter 4.

1.5 Story Tracking Evaluation Frameworks

1.5.1 Automatic Evaluation

Evaluation of text-oriented versions of the story tracking task have been standardized in the Document Understanding Conference (DUC) Update Summarization task [1] and in the TREC Novelty Detection task [14]. However, although much effort has been put into TTM by the scientific community, it has been mostly directed towards applying different families of algorithms in order to extract various patterns from evolving corpora. Less effort was put towards defining a framework for evaluating extracted patterns. This resulted in a lack of common procedures to compare TTM methods in a principled way. To close this gap, we (a) investigate how different TTM methods discover novel development representations and (b) present an evaluation framework for assessment of TTM methods. The main challenge in cross-evaluating different TTM methods is overcoming the different formats of bursty patterns extracted by various methods (e.g. a set of probability distributions over terms, a list of key-words...). The problem is how to cross-evaluate quality of different methods if their output is described in different formats?

Our basic assumptions are that users construct their own description of developments out of the patterns they are presented with. Specifically for news this description is sentential, such that its quality can be assessed by the degree to which these constructed sentences (the presumed novel "facts") resemble "true" sentences – selected by human editors as ones describing the developments. The challenge is to measure an aggregate reconstruction quality over the possible/plausible fact constructions.

We therefore present procedures and metrics for (i) focusing on patterns or pattern combinations that a TTM method highlights; (ii) turning these into "fact" sentences; (iii) inspecting and comparing the degree of resemblance between the "fact" and the ground-truth sentences. The procedure is composed of the following steps: (a) bursty pattern generation, (b) pattern to query transformation, (c) sentence retrieval using generated queries, and (d) comparison of retrieved sentences and editor-selected "fact" sentences.

The in-depth description of the framework for evaluating TTM methods is presented in Chapter 6.

1.5.2 Interactive Evaluation

Users track stories via story tracking tools, and over time many different interfaces have been used for story tracking. Most of these interfaces were designed for different information retrieval tasks. However, unlike some information retrieval tasks which have a standardized interactive (user-oriented) evaluation frameworks [7], an interactive evaluation procedure of story tracking has not been defined so far (to the best of our knowledge). We propose an evaluation framework for interactive story tracking. In developing this framework we have two goals: a more general goal of enabling cross-tool evaluation through unified procedures, and a specific goal of evaluating our tool based on story graphs. We formulate story tracking as an interactive IR task of fact finding and summary creation along a time dimension. We compare interfaces on 4 sets of measures: (a) summary quality, (b) user activity, (c) interface suitability, and (d) task suitability. Additionally, we define a set of comparison metrics for a direct comparison of multiple interfaces. In Chapter 7 we describe the results of our study of 4 interfaces used in the context of story tracking.

1.6 Stories and Sources

Along with the time dimension, we also investigate the source dimension of stories. Our goal is to understand not only how stories change over time, but also across the sources. Newspaper archives, digital libraries, digital encyclopedias, online book repositories and similar services allow easy access to large collections of documents originating from various sources. A family of these sources, the so-called social media has recently gained much popularity, positioning itself as a premium and unique fully digital-born online story source. Riding on the wave of "democratization" in content creation and distribution, a number of platforms for web-logging (blogging), social networking, and micro-blogging became one of the main sources of information for staying in touch with current events. We frame our exploration of differences in story sources as a study of differences between emerging social and more traditional media.

We focus on Twitter as a prime "real-time" social media outlet, and analyze the differences between its content and the content of other media (professional news outlets, news-wire agencies, and blogs). Story themes highlighted on Twitter are different than ones highlighted in the mainstream media, putting citizen journalists in the role of *editors* filtering and spotlighting certain story themes. However, sometimes stories overlap in both traditional and social media and our goal is to analyze the role of social media in such situations. We

ask whether citizen journalists tend to *create* news or *peddle* (re-report) existing content. To explore these roles, we devised a set of comparison measures. These measures compare similarity of multiple components that are important for news reporting. The main idea behind using corpora similarity is that higher similarity of content would suggest "peddling", while low similarity would suggest more originality and "creation".

In Chapter 8 we describe the framework in detail and present a case study reporting on differences between Twitter and other news sources.

1.7 Exploring News Story Spaces

As mentioned in Section 1.1, numerous online story spaces provide access to corpora of various genres. In this thesis we focus on online news story spaces, and investigate story tracking in the news domain.

In recent years online news spaces have been gaining momentum, and in the U.S., online sources surpassed newspapers for the first time in 2008, and became after television the most important source of national and international news among the general population, and the most important news source for the people under the age of 30 [12]. With the recent advances and adoption of new technologies such as smartphones and tablet computers, it is expected that this trend will continue.

Compared with other story genres such as blogs or scientific publications, research into online news has been less tackled. Research into scientific publication has developed into a scientific discipline – bibliometrics, while different communities have devoted full venues to social media and blog research, e.g. *The International Conference on Weblogs and Social Media* conference series. Although news corpora have been used extensively [10, 13] in the past, it has mostly been used only as a data source, while the task for which the data was used was not modeled itself based on news reading activities of users. In the past there have been several initiatives for creating news oriented venues and forums, e.g. *Intelligent Analysis and Processing of Web News Content Workshop* held in 2009. However, none of the venues like this one have been established as long-term venues for computational news exploration. Having in mind the impact and the wide use of news data in various domains, combined with relatively less research into news mining compared to blog or bibliometrics research, we consider that our research will have a stronger impact if we focus on online news story spaces.

To define an evaluation framework for TTM, we needed clear and easily

extractable development representations. In Section 1.4.1 we discussed the differences between development representations across story genres. For news stories, developments can be represented by an assertion, in sentential form (e.g., "The ski slalom ended with ... winning the gold medal"). Using news data and sentences as development representation allows for easier evaluation and results interpretation of TTM methods compared to using data from other domains and different development representation.

For the above mentioned reasons, we focused on the news domain, and used corpora constructed from news sources. However, our method does not include any news-specific components in the extraction of the bursty patterns, and can easily be applied to data originating from any other genre as long as this data is comprised of textual and time-stamped documents.

1.8 Contribution and Outline

In this section we summarize the main thesis contributions and describe the outline of the thesis in the following chapters. This thesis is submitted as a compilation of research papers over the period of the last 4 years[4], and each chapter presented in the rest of the thesis, except the next and the last chapter, is a published paper.

1.8.1 Thesis Contributions

Each of the papers included in this thesis lists several of its specific contributions. In this section, we list the more general contributions of the thesis as a single body of work. The major contributions of the work presented in this thesis are:

I) development of a graph-based system for story tracking;

II) introduction of a new framework for evaluating temporal text mining methods; and

III) unifying of different research areas (topic detection and tracking, update summarization, sentence retrieval, temporal text mining) in the context of story tracking.

In addition we also made a number of smaller contributions by: (a) providing a description of the effects query bursts have on web search, (b) developing of

[4]From 2008.

novel tools for story tracking and news browsing, (c) defining a framework for interactive story tracking evaluation, and (d) defining a framework for exploring divergence between news sources.

1.8.2 Thesis Organization

The next chapter includes a more detailed overview of story tracking and related research. After this, the thesis is split into 4 parts based on the 4 thematic areas defined in Section 1.2. Parts contain one or more chapters which correspond to 6 research questions (research question $Q3$ is addressed in two chapters - 4 and 5 both discuss use of graph-based patterns for story tracking). However, this is not a strict organization and it can be the case that several chapters contain overlapping work. The thesis then continues by presenting:

Part II Stories and Search Engines

Chapter 3 – The Effects of Query Bursts on Web Search:
Ilija Subašić and Carlos Castillo: Investigating Query Bursts in a Web Search Engine – What happens when something happens?
Invited for publication in Web Intelligence and Agent Systems: An International Journal (WIAS) (conditionally accepted August 2011, revised version sent September 2011).[5]

Part III Story Tracking Methods and Tools

Chapter 4 – *Story Graph Extraction and Visualization*:
Ilija Subašić and Bettina Berendt: Discovery of interactive graphs for understanding and searching time-indexed corpora. Knowledge and Information Systems 23(3): 293-319 (2010) [15].[6]

Chapter 5 – Tracking Document-set Evolution
Ilija Subašić and Bettina Berendt: Story Graphs: Tracking document set evolution using dynamic graphs. Intelligent Data Analysis Journal, special issue on Dynamic Networks and Knowledge Discovery. Vol. 17(1), 2013. IOS Press. Accepted for publication June 2011, publication

[5] A shorter version of the paper was published as: Ilija Subašić and Carlos Castillo. 2010. The Effects of Query Bursts on Web Search. Volume 01 (WI-IAT '10), Vol. 1. IEEE Computer Society, Washington, DC, USA, 374-381, [17]. This paper won the Best Student Paper award at the same conference (WI-IAT'10))

[6] This paper is partly based on the paper: Ilija Subašić and Bettina Berendt. 2008. Web Mining for Understanding Stories through Graph Visualisation. In: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08). IEEE Computer Society, Washington, DC, USA, 570-579. This paper won the Best Application Runner-up Award at the same conference (ICDM '08).

scheduled for 2013.

Part IV Story Tracking Evaluation Frameworks

Chapter 6 – *Ilija Subašić and Bettina Berendt: From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In Proceeding of the 19th European Conference on Artificial Intelligence (ECAI 2010), pages 517-522, Amsterdam, The Netherlands, 2010. IOS Press.* [18].

Chapter 7 – Interactive Evaluation of Interfaces for Story Tracking:
Ilija Subašić and Bettina Berendt: Interactive evaluation of interfaces for story tracking. HCIR 2011: The Fifth Workshop on Human-Computer Interaction and Information Retrieval (October 2011), Electronic publication, no pages.[7]

Part V Stories and Sources

Chapter 8 – Investigating Differences in Story Sources
Ilija Subašić and Bettina Berendt: Peddling or Creating: Investigating the Role of Twitter in News Reporting. In Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR'11). Springer-Verlag LCNS, Berlin, Heidelberg,207-213 [19].

Note. Naturally, over the 4 years it took to complete this thesis terminology we used in different papers has changed, and readers should take this into account when reading the full text. For the same reason there are overlapping sections in certain papers, as it was necessary to introduce previous work to make the papers self-contained. The largest overlap is between Chapters 5 and 6. To some extent Chapter 5 can even be regarded as a super-set of Chapter 6. However, the reasons for including Chapter 6 in this thesis are three-fold. First, although the procedures, metrics, and data used in the automatic story tracking evaluation framework are the same in both chapters, Chapter 6 contains much deeper motivation and rationale for adopting this framework. The second reason concerns an additional evaluation of query generation procedures that was reported only in the paper now included in Chapter 6. Finally, we considered that grouping the work described in Chapter 6 together with the interactive evaluation framework described in Chapter 7 into Part IV, allows potential future readers of this thesis to easily discover our work on story tracking evaluation. For these reasons, rather than including it as a lengthy addendum to Chapter 5 we decided to make Chapter 6 an independent chapter in this thesis.

[7] `https://sites.google.com/site/hcirworkshop/hcir-2011/posters`

References

[1] Duc 2007: Task, documents, and measures, 2007. http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot, National Institute of Standarts and Technology, US Departmant of Commerce, 2007, retrieved July 2011.

[2] James Allan, editor. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[3] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, Yiming Yang, Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[4] James Allan, Victor Lavrenko, and Hubert Jin. First story detection in TDT is hard. In *Proceedings of the ninth international conference on Information and knowledge management*, CIKM '00, pages 374–381, New York, NY, USA, 2000. ACM.

[5] James Allan, Anton Leuski, Russell Swan, and Donald Byrd. Evaluating combinations of ranked lists and visualizations of inter-document similarity. *Information Processing and Management*, 37(3):435 – 458, 2001.

[6] Clough, P.; Foley, C.; Gurrin, C.; Jones, G.J.F.; Kraaij, W.; Lee, H.; Murdoch, V., Eds. *Advances in Information Retrieval - 33rd European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 18-21, 2011. Proceedings*, Vol. 6611, *Lecture Notes in Computer Science*. Springer, 2011.

[7] S. T. Dumais and N. J. Belkin. The TREC interactive tracks: Putting the user into search. In *Text REtrieval Conference*, Digital Libraries and Electronic Publishing. MIT Press, September 2005.

[8] Matthew Hurst. Temporal text mining. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[9] Jon Kleinberg. Bursty and hierarchical structure in streams. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 91–101, New York, NY, USA, 2002. ACM.

[10] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Res.*, 5:361–397, December 2004.

[11] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21–24, 2005*, pages 198–207. ACM, 2005.

[12] Kristen Purcel, Lee Rainie, Amy Mitchell, Tom Rosenstiel, and Kenny Olmstead. Understanding the participatory news consumer, http://www.pewinternet.org/Reports/2010/Online-News.aspx, retrieved May 2011. PEW Research, 2010.

[13] Evan Sandhaus. The New York Times annotated corpus, 2007. http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T19. Retrieved July 2011.

[14] Ian Soboroff and Donna Harman. Novelty detection: the TREC experience. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 105–112, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[15] Ilija Subašić and Bettina Berendt. and Bettina Berendt. Discovery of interactive graphs for understanding and searching time-indexed corpora. *Knowledge and Information Systems*, 23(3):293–319, 2010.

[16] Ilija Subašić and Bettina Berendt. Experience stories: A visual news search and summarization system. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *ECML/PKDD (3)*, volume 6323 of *Lecture Notes in Computer Science*, pages 619–623. Springer, 2010.

[17] Ilija Subašić and and Carlos Castillo. The effects of query bursts on web search. In Jimmy Xiangji Huang, Irwin King, Vijay V. Raghavan, and Stefan Rueger, editors, *Web Intelligence*, pages 374–381. IEEE, 2010.

[18] Ilija Subašić and Bettina Berendt. From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 517–522, Amsterdam, The Netherlands, 2010. IOS Press.

[19] Ilija Subašić and Bettina Berendt. Peddling or Creating? Investigating the Role of Twitter in News Reporting. In [6], pp. 207–213.

[20] Russell C. Swan and James Allan. Aspect windows, 3-d visualizations, and indirect comparisons of information retrieval systems. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 173–181, New York, NY, USA, 1998. ACM.

[21] Russell C. Swan and James Allan. Automatic generation of overview timelines. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR 2 000, pages 49–56, New York, NY, USA, 2000. ACM.

[22] Ellen M. Voorhees and Donna Harman. Overview of the eighth text retrieval conference (TREC-8). In *Proceedings of the 8h Text Retrieval Conference (TREC-8)*, pages 1-24, 1999. NIST.

Addendum A – Definitions of key terms

This addendum contains the definitions of the most important terms used in this thesis.

- **CORE**:
  - *story* – a set of time-stamped theme-related document (e.g. all news reports about a specific natural disaster such as earthquake in Japan 2002);
  - *story theme* – an underlying theme of all documents of one story share;
  - *story space* – channels for the distribution of stories (e.g. a news report aggregator allowing access to documents for a theme and over time.);
  - *(story) developments* – novel and relevant information contained in a story;
  - *development representation* – natural language expressions of developments in the story documents.
- **STORY TRACKING**:
  - *story tracking* – activity of following the story over time;
  - *story tracking task* – a task of extracting and presenting developments from stories;
  - *story tracking method* – an automatic method for extracting developments from stories;
  - *story representation* – outputs of story tracking methods;
  - *story understanding* – a goal in story tracking; understanding the developments in a story;
  - *story search* – a goal in story tracking; discovering the most relevant documents for the developments;
  - *story graphs* – a graph-based story representation.
- **BURST RELATED**:
  - *burst* – sudden surge of importance of some element, where the importance of an element can be measured with frequency, probability...;
  - *bursty patterns* – a text pattern from a story going through a burst;
  - *temporal text mining* – a family of temporal text mining methods that outputs bursty text patterns (bursty patterns);

- **NEWS DEVELOPMENT REPRESENTATION**:
  - *novel facts* – sentential development representation for the news domains;
  - *ground-truth facts/sentences* – a set of editor selected novel facts that describe developments in a story;
  - *retrieved facts/sentences* – a set of novel facts retrieved by an automatic method;

Chapter 2

Related work

Story tracking methods and tools in online story spaces pertain to several data mining (DM), information retrieval (IR), and human-computer interface (HCI) related research topics developed over the past decades. We start this chapter with an overview of the past work in different families of story tracking methods. We divided story tracking methods into document-oriented and text-oriented methods based on the story representation they use. Document-oriented methods use groups of documents as story representation, while text-oriented methods extract content from documents and use text elements (paragraphs, sentences, or words) as story representation. After an survey of story tracking methods we continue with an overview of web mining research related to story tracking. Finally, we review some of the works in visualizing evolving corpora, and evaluating user interaction with such systems.

2.1 Document-oriented Story Tracking

The idea of document-oriented story tracking is that developments can be represented using groups of documents. Each group should contain documents describing a single (or highly related) developments. Most of the work in this area has been standardized through NIST[1] sponsored Topic Detection and Tracking (TDT) framework.

[1]National Institute of Standards and Technology, U.S. Department of Commerce - http://www.nist.gov/index.html

2.1.1 TDT Tasks

The TDT framework was launched in 1997 with a pilot study [6] and ended in 2004. The main idea behind the entire TDT framework is the *event-based* information organization. Event-based document organization structures documents based on the events (topics) discussed in the stream of incoming documents. The TDT framework uses a slightly different terminology than the one used in this theses and uses the terms *topic* and *event* in the same way we use the term story theme, while the term story is used in the same way we use the term document. Although it is not formally defined in TDT, we can regard the set of on-topic documents as a story. Details about the history and evolution of the TDT framework can be found in [4]. In the same book, in Chapter I, Allen describes the following five TDT tasks [4]:

- story segmentation,
- story link detection,
- cluster detection,
- tracking, and
- first story detection (FSD).

In the first task (story segmentation) the goal is to segment an incoming stream into documents (TDT stories) based on the themes (TDT topics) they discuss. This task is mostly concerned with segmenting transcripts of audio news broadcasts. The story link detection task detects whether two documents (TDT stories) discuss the same story theme (TDT topic). Cluster detection groups documents into clusters, where each cluster contains documents on a story theme (TDT topic). The tracking task monitors a stream of documents to find documents belonging to a set of predefined stories (TDT on-topic documents). FSD monitors an incoming stream of documents with a goal of finding the documents describing a novel (previously unseen) story theme (TDT topic). In contrast to the cluster detection task, FSD task is concerned with discovering only the first documents on a topic. In cluster detection task the goal is to discover all documents that are on specific topics.

2.1.2 TDT Evaluation Framework

The TDT evaluation is based on a 2×2 contingency matrix resembling one used for two-way classification (positive/negative). Documents are pre-labelled

by a number of human assessors as target (positive) and non-target (negative). TDT uses a slightly different than usual machine learning terminology. Instead of True Positive (positive examples assigned to the positive class) and True Negative (negative examples assigned to the negative class) TDT uses the term *correct*. False Positive examples (negative examples assigned to the positive class) are referred to as the *false alarms*, and the False Negative examples (positive examples assigned to the negative class) as *missed detections*. Based on this contingency matrix the methods are evaluated using two metrics: detection cost (C_{Det}) and decision error tradeoff curve (DET).

The detection cost is defined as a linear combination of the probabilities of producing a missed detection and false alarm. In TDT evaluation framework [25] it is defined as:

$$C_{Det} = C_{Miss} * P_{Miss} * P_{Target} + C_{Fa} * P_{Fa} * (1 - P_{Target}). \tag{2.1}$$

where C_{Miss} and C_{Fa} are preset costs assigned to missed detections and false alarms (in TDT they are usually set to 10 and 0.001 respectively). The idea is that while tracking a story users have a cost of reading documents and this cost is increased if they are presented with a false alarm document, or a document is not detected as belonging to the story. P_{Miss} and P_{Fa} are the probabilities of missed detections and false alarms. They are estimated as $|missed\ detection|/|target|$ and $|false\ alarm|/|non - targets|$. Finally, P_{Target} is a preset probability of observing a target document. It is a priori estimated based on the corpus statistics of training corpora. Due to the low values of C_{Fa} it is hard to interpret the results of C_{Det}, as the preset costs are in different orders of magnitude. To overcome this, TDT uses normalized detection cost ($(C_{Det})_{Norm}$). This cost is calculated by dividing the detection cost with the minimum cost if all documents are labeled as target or non-target. The normalized detection cost is derived as:

$$(C_{Det})_{Norm} = C_{Det}/MIN((C_{Miss} * 1.0 * P_{Target} + C_{Fa} * 0.0 * (1 - P_{Target})), \tag{2.2}$$

$$C_{Miss} * 0.0 * P_{Target} + C_{Fa} * 1.0 * (1 - P_{Target})), \tag{2.3}$$

$$(C_{Det})_{Norm} = C_{Det}/MIN(C_{Miss} * P_{Target}, C_{Fa} * (1 - P_{Target}). \tag{2.4}$$

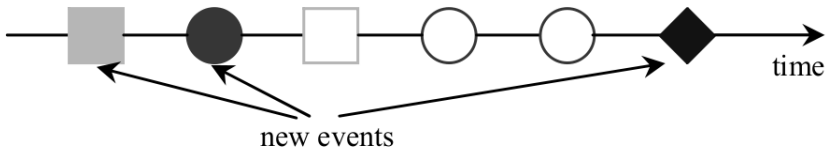


Figure 2.1: Events in a document stream (from [48]). Different shapes correspond to different events. Filled shapes represent the documents that need to be captured.

Detection error tradeoff (DET) curve graphically depicts the performance tradeoff between P_{Miss} and P_{Fa}. It makes use of scores attached to target/non-target decisions. On the X-axis of DET is the probability of false alarms and on the Y-axis is the probability of missed detection. Improvement in the method is seen if the curve moves to the left hand side. The minimum DET point is the best score a system could achieve with proper thresholds.

2.1.3 First Story Detection

Out of all tasks in the TDT framework, First Story Detection (FSD) [8] is the most similar to story tracking as defined in this thesis. Sometimes this task is also referred to as the Online New Event Detection (ONED) task. In this task a system has to decide if a newly arrived document is discussing "a new event" or not. Figure 2.1 (from [48]) illustrates the flow of the FSD task. In TDT events are defined as *"something that happens at a particular time and place"* [8], and in TDT evaluation, events correspond to story themes in this thesis. In that sense FSD differs from story tracking as it aims to detect documents discussing a new story theme. However, it is easy to imagine that all documents of an incoming stream belong to the same story. It this case the "new events" would correspond to developments making FSD more similar to our notion of story tracking task.

One of the early, pre-TDT, works in FSD is described in [49]. The authors use a k-nearest neighbour approach to assign documents from a stream to groups of previously observed similar documents. During the 7 years the TDT framework was active, a number of approaches to solving the FSD task have been developed. In [8] the authors generate queries and use cosine similarity to compare them with already seen documents. A threshold value determines if a document is discussing an already seen or a new event. Most of the methods developed in the TDT framework concentrate on building models

that maximize effectiveness for given evaluation metrics, and disregard issues like efficiency, data quality, and user interaction. Luo et al. tackle these problems and in [48] explore document source quality, efficient indexing, and user interfaces for FSD. An interesting extension of the FSD task is described in [53] where authors go beyond detecting events and aim to discover relations between events. A more recent application of FSD to social network status update data is described in [54].

2.2 Text-oriented Story Tracking

In contrast to the document-oriented story tracking methods, text-oriented story tracking methods do not use the whole documents as development representation, but aim to represent developments using content extracted from the documents. We identify 3 main bodies of work in this area: update summarization, novel sentence retrieval, and temporal text mining.

2.2.1 DUC Update Summarization

In the DUC (Document Understanding Conference) Update Summarization task[2] participants are given a data set of documents divided into topics. A topic is divided into three time periods (A,B,C) each containing around 10 documents. The time periods are consecutive, but differ in length ranging from several days to several months. The task is to produce a 100 word summary for each of the three periods so that the summaries in consecutive periods do not contain information from previous periods. From a story tracking perspective, we can regard each period summary as story representation for the same period.

Evaluation framework. Introduced in 2007 as a part of the DUC workshop series [1], the evaluation framework for update summarization combines human and automatic evaluation. It has been shown that automatic evaluation using the ROUGE framework is highly correlated with human evaluation [45]. The ROUGE framework measures the recall of n-grams between the human- and machine-produced summaries. Most commonly used ROUGE scores are measured on 2-grams ($ROUGE.2$) and skip-4 2-grams ($ROUGE.SU4$). The former are continuous sequences of two terms, the latter are two terms that may be separated by up to 4 other terms in-between. Given a machine-produced

[2] In 2008, DUC became a Summarization track in the Text Analysis Conference (TAC).

(m) summary, and a set (H) of human-produced summaries (h) these measures are defined [45] as:

$$ROUGE.n = \frac{\sum_{h \in H}(\sum_{gram_n \in h}(Count_{match}(gram_n)))}{\sum_{h \in H}(\sum(Count(gram_n)))}. \quad (2.5)$$

where n stands for the length of the n-gram (e.g. $gram_2$ is used to refer to bi-grams), and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in machine-produced summary (m) and all $h \in H$.

Methods. Many multi-document summarization methods have been applied to update summarization task. Inspired by PageRank, Günes and Radev [30] present the LexRank summarization algorithm that selects the most salient sentences from a graph-based text representation. A similar approach was followed in [17]. Other methods specifically designed for update summarization use information distance based summaries [47], integer linear programing [27], or latent semantic analysis [65]. An in-depth review of different approaches to document summarization is presented in [19] where a number of methods is surveyed.

2.2.2 TREC Novelty Detection

In the TREC Novelty detection task participants are given a set of documents on a topic and their task is to extract sentences that are relevant to the topic and to select sentences that are "new". In this context "new" is defined as containing information that has not appeared previously in a topic's set of documents. Therefore, this task corresponds to story tracking, and a set of retrieved "new" sentences for a time period corresponds to the story representation of the same period.

Evaluation framework. The evaluation framework for sentence retrieval is standardized though the TREC Novelty detection track [63] which ran from 2002 until 2004. As mentioned previously, novel sentence retrieval is the Novelty track task which mostly resembles story tracking. In this task, the participants are provided with a set of 25 topics each having 25 relevant documents and zero or more irrelevant documents (the number of irrelevant documents varied over the years). Relevancy of the document to the topics is decided by human annotators. Every sentence in the relevant documents is judged as "relevant", "new" or "irrelevant". "New" sentences are a subset

of relevant sentences. In the novel sentence retrieval task the methods should retrieve new sentences. The participants are presented with a training set with annotated sentences and their final solutions are evaluated on a held-out data set. Novelty track adopts precision/recall style measures. For retrieved sentences precision@k, MAP (mean average precision) and F-measure are calculated against editor judgements. The baseline used in the Novelty track is human performance for the same task.

Methods. A comparison of well-known information retrieval models for sentence retrieval tasks is presented in [10]. Another in-depth analysis of the sentence retrieval is presented by Murdock [51]. Experiences on running the TREC Novelty detection task are summarized by Soroboff and Harman [64].

2.2.3 Temporal Text Mining

Both update summarization and novel sentence detection rely on the extraction of larger units of natural language as story representation (e.g. sentences or paragraphs). However, systems that rely on atomic units, such as words, allow for the discovery and understanding of the "anatomy" of a story in a more fine-grained manner. These methods have been usually refered to as temporal text mining (TTM) methods [50]. Certain words may be particularly interesting for story representation when they are *bursty* [38], i.e. when publication activity on them is very strong in a certain time period, picking up volume fast at this period's beginning and (usually) disappearing again as fast. Burstiness is the key notion of TTM, and all TTM methods model burstiness and output bursty patterns in some format.

Based on the differences in the format of story representations they use, we divide TTM methods in three groups: (a) keyword representation, (b) group representation, and (c) combo representation methods. As story representation the first group, presented in [38, 26, 29, 61], uses a list of bursty N-grams ranked by their burst scores. The second one [50, 26, 72, 34, 52] joins bursty N-grams into groups which point to developments. Combo representation group methods use a combination of the previous two approaches [66, 7]. Figure 2.2 shows an example of story representation for the three groups of TTM methods.

Burstiness has been explored with respect to various domains and phenomena including "buzz" in text and news streams [26, 29, 31]. Fung et al. [26] group "bursty features" into "bursty events" based on co-occurrence. Mei and Zhai [50] use a mixture model to model brursty topics in a corpora following the same story theme. In [72], LDA with an added time variable is used for the same

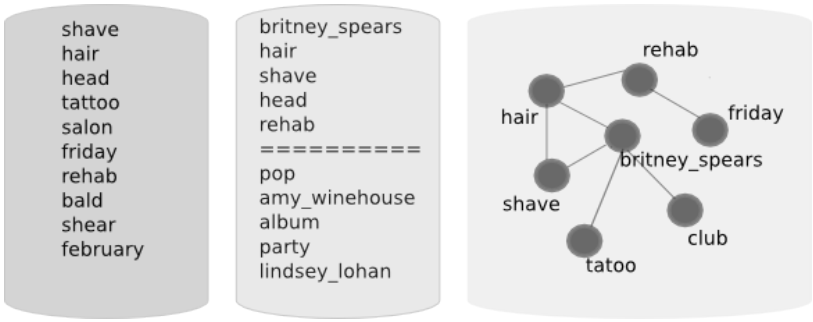


Figure 2.2: Example story represenation generated by 3 TTM method groups. The left-most rectangle shows bursty keyword list generated by [38] - keyword representation; the middle rectangle shows 2 bursty topics (separated by "=") generated by [50] - group representation, and the right-most rectangle shows an graph representation generated by [66] - combo representation.

purpose. TTM methods have been applied for the analysis of news [66, 62], quotations [44], blog posts [41], and Twitter update messages [42].

Temporal text mining evaluation procedures. Most of the evaluation procedures for TTM are tailored to evaluate only one method, and do not tackle cross-evaluation of competing methods. Roy et al. [57] presented a method for semi-automatic detection and labeling of topics and compared these topics with an editor-created list. Wang and McCallum [72] modified the LDA topic model to include a time variable, compared it to a standard LDA model, and compared the differences in the distribution of words over time between the models. The idea is that more bursty words should have different distributions over topics in different time periods, while the less or non-bursty patterns should have more similar distributions over topics in different periods. To test the accuracy of their measures of burstiness defined on a word-topic distribution, Knights et al. [39] created an artificial set by drawing words from a set of word-topic distributions. In selected periods, the words were drawn from a subset of topics, making these topics bursty. The authors measured whether their method captures this artificial burst. Wang et al. [71] tested their method by comparing bursts discovered in multilingual corpora on the same topic.

2.3 Web Mining for Story Tracking

Applying various data mining techniques to the data available on the Web, so-called Web mining, has been widely used in order to understand and evaluate users and content on the Web. As we investigate story tracking as an online activity, several Web mining problems are closely related to the work in this thesis. Rather than adopting an often used division of Web mining [14] into Web content mining, Web usage mining, and Web structure mining, we look at a number of tasks related to story tracking which can fall under multiple Web mining types.

2.3.1 Web Search Result Clustering

Web search result clustering aims at learning and presenting the structure within a query result set by applying clustering algorithms (see [32] for a classical approach). Presenting structured results of their search to the users provides them with insights as to the better understanding of the retrieved documents. It also enables users to focus their exploration the specific area of their interest. For example, users follow news about a sporting event ("Football World Cup"), and instead of a unified list of results, the users are presented with grouped documents discussing different sub-themes of the corpora (e.g. games and results, specific spotlighted player, fan incidents...). In that sense web search results clustering is similar to story tracking by supporting both story understanding and story search goals.

Web search search result clustering can be seen as an instantiation of TDT detection tasks to web search. However, the major difference is that web search result clustering is usually regarded as an ad-hoc activity not taking into account the time dimension of corpora. An overview of a number of techniques for web search results clustering and cluster visualizations that have been developed over time is presented in [13].

2.3.2 Automatic Query Expansion (AQE)

AQE is a set of techniques for the expansion of the initial search query with semantically similar terms. Guiding users towards semantically similar terms can be also viewed as guiding users towards discovering developments in the story. If we assume that users start their search with a high level story-related query – a story theme (e.g. "earthquake in Japan"), then semantically related queries may lead users towards queries that relate to the developments (e.g.

"Fukushima nuclear plant"). Thus, AQE relates to the story search task, but most work in this area disregard the temporal dimension of the corpora.

Generally, query expansion methods can be global or local methods [74]. Global methods try to expand queries by analyzing the complete corpus. Local methods are based on pseudo-relevance feedback and try to expand the query by analyzing the relevance of a subset of top-ranked documents. For specific domains, the general approaches can be improved upon by domain-specific methods, as shown for blogs or news in [23].

2.3.3 Web Information Extraction (IE)

IE aims at automatically finding facts and relations in text. If the specific natural language structure of developments is known in advance, IE techniques can be used to point out the emerging developments. For example, a user follows a stream of business news with the goal of learning about the major acquisitions by companies. In such scenario IE system could extract only those sentences which fit its model of natural language expression of "company acquisitions" (e.g. "COMPANY NAME" bought "COMPANY NAME").

A well-known IE system is KnowItAll [24]. A more recent system that improves on efficiency limitations encountered by KnowItAll is TextRunner [75]. Although these systems have good results in fact retrieval, they discard the time dimension and the internal structure of corpora.

2.3.4 Query Log Analysis

As search engines are among the most commonly used story spaces, analyzing their usage is important for understanding user behaviour during web corpora exploration. Since the early studies of query logs presented in [33, 43, 60], the field has branched out into several areas, and our coverage of them in this section is by no means complete. Out of these areas the most related to this thesis is the research into the temporal dimension of query logs. We are specifically interested in high increases of query frequencies (query bursts) inside a query log.

Temporal query analysis. The analysis of an hourly time series in [12] and a long-term time series in [11] show distinct properties in the frequency profile for queries belonging to different topical categories. Conversely, the authors of [9] study whether the different frequency profiles can be used to improve

query classification. In [20, 21] instead of topical categories the authors look for differences between high-frequency and low-frequency queries. Adar et al. [3] compare time series of queries from different sources. Their results are a description of different classes of temporal correlation and a visual tool for summarizing them. Previously, using the correlation between query frequency time series analysis, Chien at al. [18] uncovered semantic similarity between time-aligned series. Time-based query similarity discovery is also the topic of [78], where clustering of a bipartite graph of queries and pages is used.

Query bursts. Current applications of query burst detection include such applications as the detection of real-world events [79] and tracking the spread of diseases such as flu [28]. In [70], query bursts are detected as outliers in the query frequency series, specifically as moments at which the query shows 1.5-2 standard deviations higher frequency than its average in previous periods. In [55], increases in normalized query frequency are used to discover query bursts and investigate their possible use in different fields, such as sociology.

2.4 Interactive Story Tracking

While story tracking, users interact with tools that provide them with an interface to stories. In this section we review works on visualizing and exploring evolving corpora. Additionally, we investigate the work in interactive evaluation of such systems.

2.4.1 Visual Corpus Exploration

Visualization. Visualizations are probably best suited to display the complex relationships in a corpus. For the research in this thesis the specific interest is the visualization of co-occurrences of different elements in a corpus. Smith [61] provided users with an interactive map browser for exploring the location-time co-occurrences. Wong et al. [73] show a domain-independent way of visualising pairwise associations of words that also takes the strength of these associations into account. They plot words against time and show co-occurrences by connecting lines in a format that is related to parallel coordinates. Their graphs provide an excellent overview of the occurrence or recurrence of pairwise associations over a whole timeline. However, because time takes up one visual dimension, higher-order patterns of associations cannot easily be detected. In contrast to this, some work show associations per time point/period. This "snapshot" idea is the same as that used in the graph sequences used for

visualising scientific publications and topics in, e.g., [15, 16, 34]. Interesting work on using graphs to visualize corpora is presented in [52, 69, 58]. Zoetrope [2] presents an interactive interface that allows users to track single DOM elements of an HTML page over time.

Temporal Search Interfaces. Usually, the results of a Web search are presented as a ranked list of documents based on their relevance to a specified query. The use of graphical representation of search results has been well studied; see [40] for an overview. Apart from the "classical" news search engines like Google News and Yahoo! News, recently many alternative ways of tracking and browsing news collections have been developed. Summarization like that provided by Google Trends show surges in publication and query activity in certain time periods. *Google News Timeline* (`http://newstimeline.googlelabs.com/`) provides a pre-set time period (day, week, month, year) overview of news using a timeline interface. It allows for the tracking of news sources, arbitrary queries or entities such as movies, books, music... Another Google system, named *Fast Flip* (`http://fastflip.googlelabs.com/`), provides an interface for browsing news articles resembling hard-copy newspaper reading. *The Yahoo! Correlator* (`http://correlator.sandbox.yahoo.net/`) associates a search term with all its related "events". A similar system named Time Explorer is designed for exploration of news stories (`http://fbmya01.barcelonamedia.org:8080/future/`). *EMM NewsExplorer* (`http://emm.newsexplorer.eu`) and *EMM NewsBrief* (`http://emm.newsbrief.eu`) are news search and summarization services tracking news from a large number of multi-lingual news sources. *MemeTracker* tool (`http://www.memetracker.org/`), based on research described in [44], tracks quotes from news and visualises their "burstiness" using interactive charts.

2.4.2 Interactive IR (IIR) Evaluation

Most of the research in information retrieval and story tracking has been oriented towards developing more effective algorithms and more efficient indexing schemes. Less effort has been put in investigating and evaluating the interactive side of these tasks. However, there has been substantial work in Interactive IR – "the study of human interaction with information retrieval systems", as defined by Robins [56]. One of the most elaborate projects in IIR is the TREC Interactive Track [22] series. Started with TREC-3 in 1994, this framework lasted for 9 years, and was succeeded by High Accuracy Retrieval of Documents (HARD) [5] and Complex Interactive Question Answering (ciQA) [37]. The TREC Interactive Track provided participates with the same

data sets, tasks, experimental settings, and evaluation measures. This made possible to cross-evaluate user interaction in different systems developed for the same retrieval task. A similar evaluation framework started in 2004 with the INEX (Initiative for the Evaluation of XML retrieval) Interactive Track [68].

A valuable overview of interactive information retrieval presented in a survey by Ruthven [59] provides an in-depth analysis of interaction-based solutions to various information retrieval related tasks. A similar overview of interactive evaluation of information retrieval system is the subject of a book by Kelly [36].

A meta-analysis of different experimental settings for controlled interactive information retrieval is presented by Julien et al. in [35]. The authors survey 31 interactive information retrieval studies, and conduct an additional meta-analysis on 8 studies. The results of meta-analysis show no significant effects of visual search interfaces when compared to non-visual ones.

2.5 Extensions and Integration in the Thesis

In the previous sections we looked at the number of research areas related to story tracking. We aimed to extend all of these areas to better understand and facilitate users while story tracking.

To understand what the specific needs of users for story tracking are, we analyzed usage data from a standard (search-box) interface (Chapter 3). In regards to similar research focused on discovering query bursts or finding temporal patterns and correlations in the query log, our analysis investigates how the user behaviour changes when developments in a story occur. We incorporated the results of this study as requirements for the method and interface we created.

Among different approaches to temporal analysis of stories, we set on temporal text mining. There were two reasons for this. First, we noticed that the majority of works in this area output keyword lists or word-topic distribution patterns. We were interested in fairly less researched graph-based bursty patterns such as ones providing both analytical power using graph analysis methods, and human understanding through visualization and layout algorithms. We investigated how to generate graph patterns for story tracking (Chapters 4 and 5), and how people understand and interact with these patterns (Chapters 8 and 7). The second reason for investigating TTM is the lack of a standardized evaluation framework in this area. Unlike update summarization and novelty detection there is no standardized community-wide

effort in establishing an evaluation framework for TTM. Therefore, in this thesis (Chapter 6) we defined such a framework.

Despite a large number of different visualizations and visual search interfaces, most of them are still not widely accepted by users in general (web) document search. We focused on a specific application (story tracking), and developed an interface for it. In this way we tailored an application-specific interface aiming to overcome some of the drawbacks of general document visual search interfaces (Chapters 4 and 7).

2.6 Conclusion

Due to the wide scope and the interdisciplinary nature of this thesis, in this chapter we provided a somewhat high level description of numerous related areas and frameworks, rather than a detailed description of a limited number of methods and algorithms. This is done with the intention to introduce various fields and relate them to story tracking in the context of this thesis. Each of the chapters in the following parts of the thesis also contains a related work section which introduces and explains specific methods, as well as differentiates them in regards to the work described in this thesis.

References

[1] Duc 2007:task, documents, and measures, 2007. http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot.

[2] Eytan Adar, Mira Dontcheva, James Fogarty, and Daniel S. Weld. Zoetrope: interacting with the ephemeral web. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 239–248, New York, NY, USA, 2008. ACM.

[3] Eytan Adar, Daniel S. Weld, Brian N. Bershad, and Steven S. Gribble. Why we search: visualizing and predicting user behavior. In *Proc. WWW*, pages 161–170, Banff, Canada, 2007. ACM Press.

[4] James Allan, editor. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[5] James Allan. Hard track overview in trec 2004 - high accuracy retrieval from documents. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-261. National Institute of Standards and Technology (NIST), 2004.

[6] James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron and Yiming Yang. Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218, 1998.

[7] James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of news topics. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–18. ACM, 2001.

[8] James Allan, Victor Lavrenko, and Hubert Jin. First story detection in tdt is hard. In *Proceedings of the ninth international conference on Information and knowledge management*, CIKM '00, pages 374–381, New York, NY, USA, 2000. ACM.

[9] Sitram Asur and Gregory Buehrer. Temporal analysis of web search query-click data. In *Proc. SNA-KDD*, Paris, France, 2009. ACM Press.

[10] Niranjan Balasubramanian, James Allan, and W. Bruce Croft. A comparison of sentence retrieval techniques. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 813–814, New York, NY, USA, 2007. ACM.

[11] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, Ophir Frieder, and David Grossman. Temporal analysis of a very large topically categorized web query log. *J. Am. Soc. Inf. Sci. Technol.*, 58(2):166–178, 2007.

[12] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman, and Ophir Frieder. Hourly analysis of a very large topically categorized web query log. In *Proc. SIGIR*, pages 321–328, Sheffield, UK, 2004. ACM Press.

[13] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41:17:1–17:38, July 2009.

[14] Soumen Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kauffman, 2002.

[15] C. Chen. *Mapping Scientific Frontiers*. Springer, London, 2003.

[16] C. Chen. Citespace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377, 2006.

[17] Chien Chin Chen and Meng Chang Chen. TSCAN: a novel method for topic summarization and content anatomy. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 579–586, New York, NY, USA, 2008. ACM.

[18] Steve Chien and Nicole Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Proc. WWW*, pages 2–11, Chiba, Japan, 2005. ACM Press.

[19] Dipanjan Das and Andre' F. T. Martins. A Survey on Automatic Text Summarization, November 2007.

[20] Doug Downey, Susan Dumais, and Eric Horvitz. Heads and tails: studies of web search with common and rare queries. In *Proc. SIGIR*, pages 847–848, Amsterdam, The Netherlands, 2007. ACM.

[21] Doug Downey, Susan Dumais, Dan Liebling, and Eric Horvitz. Understanding the relationship between searchers' queries and information goals. In *Proc. CIKM*, pages 449–458, Napa Valley, CA, USA, 2008. ACM.

[22] S. T. Dumais and N. J. Belkin. The trec interactive tracks: Putting the user into search. In *Text REtrieval Conference*, Digital Libraries and Electronic Publishing. MIT Press, September 2005.

[23] Jonathan L. Elsas, Jaime Arguello, Jamie Callan, and Jaime G. Carbonell. Retrieval and feedback models for blog feed search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 347–354, New York, NY, USA, 2008. ACM.

[24] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in KnowItAll (preliminary results). In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA, 2004. ACM.

[25] Jonathan G. Fiscus and George R. Doddington. *Topic detection and tracking evaluation overview*, pages 17–31. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[26] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S. Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[27] Dan Gillick and Benoit Favre. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, ILP '09, pages 10–18, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[28] Google Inc. Google Flu Trends. http://www.google.org/flutrends/, 2009.

[29] Daniel Gruhl, Ramanathan V. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The predictive power of online chatter. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 78–87. ACM, 2005.

[30] Erkan Günes and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22, 2004.

[31] Qi He, Kuiyu Chang, Ee-Peng Lim, and Jun Zhang. Bursty feature representation for clustering text streams. In *Proceedings of the Seventh SIAM International Conference on Data Mining*. SIAM, 2007.

[32] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84, New York, NY, USA, 1996. ACM.

[33] Bernard J. Jansen, Amanda Spink, and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Proc. & Mgmt.*, 36(2):207–227, March 2000.

[34] Frizo A. L. Janssens, Wolfgang Glänzel, and Bart De Moor. Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. In Pavel Berkhin, Rich Caruana, and Xindong Wu, editors, *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12–15, 2007*, pages 360–369. ACM, 2007.

[35] Charles-Antoine Julien, John E. Leide, and France Bouthillier. Controlled user evaluations of information visualization interfaces for text retrieval: Literature review and meta-analysis. *J. Am. Soc. Inf. Sci. Technol.*, 59:1012–1024, April 2008.

[36] Diane Kelly. Methods for evaluating interactive information retrieval systems with users. *Found. Trends Inf. Retr.*, 3:1–224, January 2009.

[37] Diane Kelly and Jimmy J. Lin. Overview of the trec 2006 ciqa task. *SIGIR Forum*, 41(1):107–116, 2007.

[38] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7:373–397, October 2003.

[39] Dan Knights, Michael Mozer, and Nicolas Nicolov. Detecting topic drift with compound topic models. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*, ICWSM'09. AAAI, 2009.

[40] William Kules, Max L. Wilson, m.c. schraefel, and Ben Shneiderman. From keyword search to exploration: How result visualization aids discovery on the web. Technical report, University of Southampton, February 2008. http://eprints.ecs.soton.ac.uk/15169/.

[41] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 568–576, New York, NY, USA, 2003. ACM.

[42] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 591–600, New York, NY, USA, 2010. ACM.

[43] Tessa Lau and Eric Horvitz. Patterns of search: analyzing and modeling web query refinement. In *Proc. UM*, pages 119–128, Banff, Canada, 1999. Springer.

[44] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506, New York, NY, USA, 2009. ACM.

[45] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[46] Xu Ling, Qiaozhu Mei, ChengXiang Zhai, and Bruce Schatz. Mining multi-faceted overviews of arbitrary topics in a text collection. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–505, New York, NY, USA, 2008. ACM.

[47] Chong Long, Minlie Huang, Xiaoyan Zhu, and Ming Li. Multi-document summarization by information distance. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 866–871, Washington, DC, USA, 2009. IEEE Computer Society.

[48] Gang Luo, Chunqiang Tang, and Philip S. Yu. Resource-adaptive real-time new event detection. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 497–508, New York, NY, USA, 2007. ACM.

[49] Brij Masand, Gordon Linoff, and David Waltz. Classifying news stories using memory based reasoning. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '92, pages 59–65, New York, NY, USA, 1992. ACM.

[50] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21–24, 2005*, pages 198–207. ACM, 2005.

[51] Vanessa Graham Murdock. *Aspects of sentence retrieval*. PhD thesis, 2006. AAI3242373.

[52] G. Heyer, F. Holz, and S. Teresniak. Change of topics over time and tracking topics by their change of meaning. In *KDIR '09: Proc. of Int. Conf. on Knowledge Discovery and Information Retrieval*, pages 223–228, October 2009.

[53] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. Event threading within news topics. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 446–453, New York, NY, USA, 2004. ACM.

[54] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *HLT '10: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 181–189, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

[55] Matthew Richardson. Learning about the world through long-term query logs. *ACM Trans. Web*, 2(4):1–27, 2008.

[56] David Robins. Interactive information retrieval: Context and basic notions. *Informing Science Journal*, 3:57–62, 2000.

[57] Soma Roy, David Gevry, and William M. Pottenger. Methodologies for trend detection in textual data mining, 2002.

[58] D. Rusu, B. Fortuna, D. Mladenić, M. Grobelnik, and R. Sipoš. Visual analysis of documents with semantic graphs. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*, VAKD '09, pages 66–73, New York, NY, USA, 2009. ACM.

[59] Ian Ruthven. Interactive information retrieval. *Annual Review of Information Science and Technology*, 42(1):43–91, 2008.

[60] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[61] David A. Smith. Detecting and browsing events in unstructured text. In *Proceedings of the 25th Annual ACM SIGIR Conference*, pages 73–80. VLDB Endowment, 2002.

[62] Tristan Snowsill, Ilias N. Flaounas, Tijl De Bie, and Nello Cristianini. Detecting events in a million new york times articles. In José L. Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *ECML/PKDD (3)*, volume 6323 of *Lecture Notes in Computer Science*, pages 615–618. Springer, 2010.

[63] Ian Soboroff and Donna Harman. Novelty detection: the trec experience. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 105–112, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[64] Ian Soboroff and Donna Harman. Novelty detection: The trec experience. In *HLT/EMNLP*. The Association for Computational Linguistics, 2005.

[65] Josef Steinberger and Karel Jezek. Update summarization based on novel topic distribution. In Uwe M. Borghoff and Boris Chidlovskii, editors, *ACM Symposium on Document Engineering*, pages 205–213. ACM, 2009.

[66] Ilija Subasic and Bettina Berendt. Discovery of interactive graphs for understanding and searching time-indexed corpora. *Knowl. Inf. Syst.*, 23(3):293–319, 2010.

[67] Yizhou Sun, Kunqing Xie, Ning Liu, Shuicheng Yan, Benyu Zhang, and Zheng Chen. Causal relation of queries from temporal logs. In *Proc. WWW*, pages 1141–1142, Banff, Canada, 2007. ACM Press.

[68] Anastasios Tombros, Saadia Malik, and Birger Larsen. Report on the inex 2004 interactive track. *SIGIR Forum*, 39:43–49, June 2005.

[69] M. Trampus and D. Mladenic. Constructing event templates from written news. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '09, pages 507–510, Washington, DC, USA, 2009. IEEE Computer Society.

[70] Michail Vlachos, Christopher Meek, Zografoula Vagena, and Dimitrios Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *Proc. SIGMOD*, pages 131–142, Paris, France, 2004. ACM Press.

[71] Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 784–793, New York, NY, USA, 2007. ACM.

[72] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA, 2006. ACM.

[73] Pak Chung Wong, Wendy Cowley, Harlan Foote, Elizabeth Jurrus, and Jim Thomas. Visualizing sequential patterns for text mining. In *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis'00)*, pages 105–111, 2000.

[74] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, New York, NY, USA, 1996. ACM.

[75] Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, NAACL-Demonstrations '07, pages 25–26, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[76] Oren Zamir and Oren Etzioni. Web document clustering: a feasibility demonstration. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54, New York, NY, USA, 1998. ACM.

[77] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, New York, NY, USA, 2004. ACM.

[78] Q. Zhao, S. C. H. Hoi, T.-Y. Liu, S. S. Bhowmick, M. R. Lyu, and W.-Y. Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 543–552, New York, NY, USA, 2006. ACM.

[79] Q. Zhao, T.-Y. Liu, S. S. Bhowmick, and W.-Y. Ma. Event detection from evolution of click-through data. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 484–493, New York, NY, USA, 2006. ACM.

Chapter 3

The Effects of Query Bursts on Web Search

Ilija Subašić and Carlos Castillo: Investigating Query Bursts in a Web Search Engine – What happens when something happens? Invited for publication in Web Intelligence and Agent Systems: An International Journal (WIAS) (conditionally accepted August 2011, revised version sent September 2011).[1]

Contributions as first author:

- (a) Co-defining the analysis problem;
- (b) Literature overview;
- (c) Data analysis and result testing;
- (d) Co-interpretation of the results.

[1] A shorter version of the paper was published as: Ilija Subašić and Carlos Castillo. 2010. The Effects of Query Bursts on Web Search. In Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01 (WI-IAT '10), Vol. 1. IEEE Computer Society, Washington, DC, USA, 374-381.

3.1 Abstract

Internet has become for many the most important medium for staying informed about the current news events. Some events cause heightened interest in a topic, which in turn yields a higher frequency of the search queries related to it. These queries are going through what is referred as a "query burst". In this paper we examine the behavior of search engine users during a query burst, compared to before and after it. We are interested how this behavior changes, and how it affects other stake-holders in web search.

We analyze one year of web-search and news-search logs, looking at query bursts from multiple perspectives. First, we adopt the perspective of search engine users, describing changes in their effort and interest while searching. Second, we adopt the perspective of news providers by comparing web search and news search query bursts. Third, we look at the burst from the perspective of content providers.

We study the conditions under which content providers can "ride" a wave of increased interest on a topic, and obtain a share of the user's increased attention. We do so by identifying the class of queries that can be considered as an opportunity for content providers that are "late-comers" for a query, in the sense of not being among the first to write about its topic. We also present a simple model for predicting the click share they could obtain if they decide to provide content about these queries.

3.2 Introduction

People use a variety of sources to stay in touch with current events, including television, Internet, radio, newspapers, etc. On a given day a person typically uses more than one source [25]. Among these sources, television continues to be the most important one. But since 2008, in for the general public the U.S., Internet is more important as a source of news than newspapers, and the most important news source among people under the age of 30 [24]; by 2010 Internet was the source of news for 61% of users [25] and kept growing. Search engines are one primary tool for online news discovery and access, and analyzing their query logs can answer many questions about how people inform themselves.

Users express a heightened interest in queries related to current events, which leads to sharp increases in their frequency in web search query logs. For instance, on October 18, 2008, after being parodied several times in the TV show Saturday Night Live, U.S. politician Sarah Palin appeared in the show

and met her impersonator, comedian Tina Fey. This led to a 22× increase in the frequency of the query "`snl sarah palin`" compared to two days before the event. This is referred to as a query burst [19].

From an economics point of view, higher attention on a topic, quantified as query frequency, can be regarded as an increase in the "demand" for an informational good. The "supply" that can satisfy this demand are the documents that are relevant for the topic. An increase in the demand generates an increase in the "price" users pay for accessing the information (quantified as the effort they spend searching), which is matched later by an increase in the supply of the informational good, as content providers notice the information need, and write about the topic. Following this marketplace metaphor, for the content providers we can measure their market share by the number of clicks their contents receive.

During the query bursts we know that demand increases. In this paper we investigate how are other components of this "marketplace" affected by the query bursts, motivated by the following set of questions for:

- *goods:* What are the types of bursty information?
- *price:* Does the effort users spend change during the burst?
- *supply:* How do bursts affect the production of documents?
- *market share:* How are clicks distributed over the created documents?

We set these high level research questions to encompass our motivation in investigating query bursts, and further develop them in a number of specific research questions. In addition, we investigate the origin of the bursts and their relations to actual news events.

In our research, we first detect query bursts, and then go beyond detection into characterizing their effects on the users of search engines. We also realize that not all query bursts are related to what would be considered a newsworthy event by traditional news outlets, and we compare searches in a news portal with general web searches.

Next we look at query bursts from the users' perspective, trying to uncover how does higher interest in a query change their behavior. We are particularly interested in what happens before and after a query burst. To investigate this, we analyze the effort and attention of users while searching for bursty queries. Using several metrics we categorize these queries based on user's behavior.

Contributions. This study contributes to the understanding of the effect of query bursts on web search results by observing that:

- Query bursts are not equal among them, but can be grouped in classes having distinctive properties.
- During a query burst, not only query frequency, but per-query user effort is higher according to several metrics. At the same time, clicks on query results tend to be more concentrated at the top documents for each search.
- Same queries have higher burst intensity and shorter duration in the news log than in the web search log.
- After a query burst, the distribution of clicks into search results is substantially different from that before the burst.
- Publishing early represents a clear advantage for content providers, and for some queries this advantage is unsurmountable: For other queries, a late-comer indeed has an opportunity of obtaining a non-trivial part of the users' attention.

The analysis of a log of users' activity is a type of field study, and methodologically there are advantages and disadvantages of this approach. In particular, there are many variables that we can neither observe nor infer accurately. We recognize this limitation, and support our findings through careful comparison of multiple independent metrics.

Roadmap. The next section describes previous work on temporal aspects of web usage mining. Section 3.4 defines formally the concepts we use. Section 3.5 describes in detail our experimental setting, sampling methods and metrics. Section 3.6 presents a characterization of query bursts based on evidence from search logs. Section 3.7 models changes in click share before, during, and after the query bursts. Finally, Section 3.8 presents our conclusions.

3.3 Previous work

Query-log analysis is a research topic that has received substantial attention in the last few years, with even entire venues devoted to the topic, such as the *Web Search Click Data* and the *Query Log Analysis* workshops. Since the early studies of query logs presented in [17, 21, 28], the field has branched out into several areas, and our coverage of them in this brief section is by no means complete.

Query categories. User behavior while searching for different content categories has been studied using different notions of categories and different methods for assigning queries to categories. The analysis of an hourly time series in [4] and a long-term time series in [3] showed the distinct properties in the frequency profile for queries in different editor-assigned topical categories. Conversely, the authors of [2] study if the different frequency profiles of queries can be used to improve query classification. In [10, 11] instead of topical categories authors look for differences between common (high frequency) and rare (low frequency) queries. In the present study, we do not categorize general queries but only bursty ones, and our categories are based on multiple factors which are neither topics nor overall frequencies.

Temporal query analysis. A related study by Adar et al. ([1]) compared time series from different sources. The study resulted in a description of different classes of temporal correlation and a visual tool for summarizing them. Previously, using correlation between query frequency time series [7] uncovered semantically similarity between time-aligned series. Time-based query similarity discovery using clustering of a bipartite graph of queries and pages is described in [35]. In [31] Sun at al. present a method for uncovering possible causal relationships between queries. In contrast to previous work, our paper focuses on differences on user behavior before and after a certain disruptive event, and compares it to user behavior on randomly-chosen queries and on queries that are stable over time.

Our research over a 1-year period can be considered long-term with respect to a majority of works on query-log mining. Query logs of this length have been shown useful for learning about changes and trends in user interest [27].

Query bursts. Burst analysis includes methods for detecting queries currently in a period of increased user interest. In [32], query bursts are detected as outliers in the query frequency series, specifically as moments at which a query shows 1.5-2 standard deviations times higher frequency than its average in previous periods. In [27], increases in normalized query frequency are used to discover query bursts; this is the method we use in this paper and impose other constraints to the detection of query bursts (such as having a single burst during a 1-year period) increasing precision at the expense of recall.

One of the main applications of query-burst detection has been the detection of real-world events, as in [6, 36]. One particularly interesting usage of this data is to epidemiology for instance to track the spread of flu [14]. In recent years, several tools that allow for the tracking and comparison of query frequencies have been developed [13, 15, 34].

Studying evolution of documents. There has been a substantial amount of research on the detection and evolution of bursts of activity in text corpora. Many of these works are based on [19]. Burstiness has been explored with respect to various domains and phenomena including so-called "buzz" in text and news streams in [12, 29, 33]. In particular, blogs are analyzed in [20], while the method presented in [22] is applied to both blogs and traditional news outlets.

Some of the results presented here appeared in summary form in [30]. We extend this work in several ways and: (a) present deeper background and motivation for this research, (b) widen the scope and interpretation of the initial results, and (c) introduce a new analysis of differences between bursts in news and general web search engine, exploring how users search for bursty information using a specialized news search engine, as opposed to a more general web search engine.

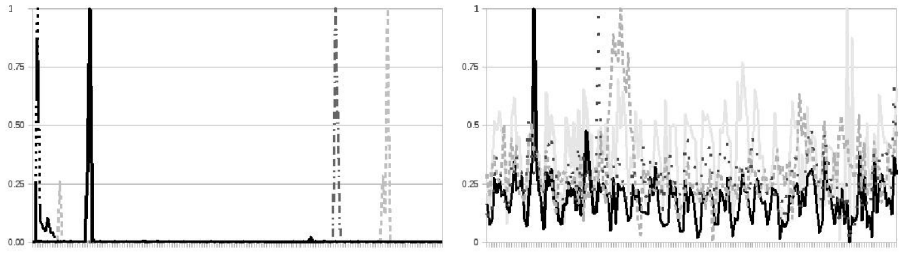
3.4 Preliminaries and notation

This section introduces some concepts and the notation that is used in the rest of the paper.

3.4.1 Query bursts

There is no standard or widely-accepted test for query bursts detection. This largely depends on the application for which the test is developed. In the case of this paper, we are interested in precisely identifying query bursts. Therefore, we define our burst measure to be precision-oriented, and include the queries which are clear outliers from a stable frequency, possibly at the expense of missing some query bursts that are not so pronounced.

Specifically, we apply a bursty measure based on normalized lift in query probability. This measure has been used for discovering bursty queries in query logs [27] as well as bursty keywords in news documents [29]. We impose a large increase in frequency, and the property of having a single distinctive burst during the one-year observation period. In practice and with the parameter setting we use, this turns out to be more restrictive than the test shown in [32]. As a consequence, the query bursts we sample are very clear (some examples are in Figure 3.1) and would be detected as bursts by any reasonable test.



((a)) Bursty queries ((b)) Stable queries

Figure 3.1: Examples of bursty and stable queries time series; x-axis is time in days, y-axis is normalized frequency (thus, the large variation for stable queries.)

Query burstiness. Let $\mathcal{Q}$ be the set of all queries. Let $\mathcal{T}$ be the set of observation periods $\mathcal{T} = \{t_0, t_1, \ldots, t_{|\mathcal{T}|-1}\}$, in which each period represents an interval of time. In this study, each $t \in \mathcal{T}$ corresponds to one day. Let $f : (\mathcal{Q} \times \mathcal{T}) \to N$ be such that $f(q,t)$ is the number of occurrences of query q in the period t.

For each query q and period t we derive a BURST INTENSITY index $b(q,t)$ which tells us how "bursty" this query is in that period, by measuring its relative increase in frequency compared to the past. This is obtained by computing:

$$b(q,t) = \frac{\frac{f(q,t)}{\sum_{q' \in \mathcal{Q}} f(q',t)}}{\frac{\sum_{u \leq t} f(q,u)}{\sum_{q' \in \mathcal{Q}} \sum_{u \leq t} f(q',u)}} \ . \tag{3.1}$$

Whenever $b(q,t) \geq \beta \sum_{u \in \mathcal{T}} b(q,u)/|\mathcal{T}|$, we say that the query q is going through a query burst at time t. If a query has no bursty period, we say the query is *non-bursty*.

If the query has bursty periods that are not contiguous, we say the query is *bursty during multiple episodes*. If all the periods in which the query is bursty are contiguous, we say the query is *bursty during a single episode*.

In the following, we refer to a sample of bursty queries during a single episode as the BURSTY queries. We also built a sample of queries having a very small variation of $b(q,t)$ in the observed series. In the following we refer to this sample as the STABLE queries. Figure 3.1 shows some of the queries from both sampled subsets. The parameters for this specific sample are presented in Section 3.5.

These samples represent extremes; most of the queries are neither STABLE nor BURSTY, therefore for some experiments we introduce a third sample of RANDOM queries chosen uniformly at random, having at least K appearances during the year.

3.4.2 Pre-episode, episode, and post-episode

For each query that is *bursty during a single episode*, i.e. in the BURSTY sample, we let $E_q = \{s_q, s_q+1, s_q+2, \ldots, s_q+d_q-1\}$ be the set of consecutive periods in $\mathcal{T}$ were the query is undergoing a query burst. We name s_q the *start* of the episode, and d_q the *duration* of the episode. In our experiments we select only queries having a minimum duration $d_q \geq \delta$.

We also obtain time intervals before and after the episode for comparison, and refer to them as *pre-episode* and *post-episode*. These time intervals are obtained in such a way that they (i) are not too close to the episode, and (ii) comprise a number of occurrences of a query that is at most the occurrences of the query in the episode.

Formally, the pre-episode of a query ends at the time period $s_q - d_q$, and starts at a time $\text{pre}(q)$ such that

$$\sum_{\text{pre}(q) \leq t \leq s_q - d_q} f(q,t) \approx \sum_{t \in E_q} f(q,t) \tag{3.2}$$

in which the approximation is due to the time granularity of one day, so we approximate $\text{pre}(q)$ to the nearest possible whole day. If there are not enough query occurrences before the episode, we set $\text{pre}(q) = t_0$. We do the same for the post-episode period, starting at $s_q + 2d_q$ and ending at $\text{post}(q)$ so that the total frequency during the post-episode period is at most the total frequency during the episode. If there are not enough query occurrences, we set $\text{post}(q) = t_{|\mathcal{T}|-1}$.

Figure 3.2 depicts graphically the relationship between pre-episode, episode, and post-episode.

3.4.3 Pseudo-episodes

For some experiments we want to study if a phenomenon is related to the bursty nature of the query or not. In the case of STABLE and RANDOM queries, we

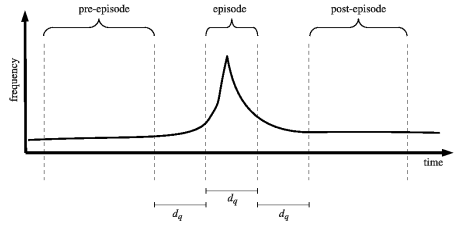


Figure 3.2: Depiction of pre-episode, episode, and post-episode.

create *pseudo-episodes* that have the same query volume as the episodes of BURSTY queries, but usually have a longer duration.

Specifically, for each of the queries in these samples, we select a starting date uniformly at random (leaving the first-3 and the last-3 months out), then pick the volume of queries in the *pseudo-episode* according to the distribution of query volume in the episodes of the BURSTY sample. The pre- and post-episode periods are created in the same manner as for the bursty queries.

We decided to sample based on volume, rather than time. This certainly introduces a time bias on our results, and we can not say how much the different lengths of time periods affects our results. However, for most of our analysis we needed to have approximately the same query volume during, before, and after li burst, making volume-based sampling a reasonable choice. Due to the short length of the bursts, we observed that time-based sampling would have produced samples of largely disproportional query volumes, and for our analysis we regard that time-bias has less effect than volume-bias would have if we employed time-based sampling.

3.5 Experimental framework

3.5.1 Dataset and sampling

We processed an in-house query log[2] to obtain one year of web searches originated in the US.

The activity of each user in the query log is first divided into logically-coherent groups of queries, using the method in [5]. In the following, when we refer to *sessions* we always mean groups of related queries, known in the literature as query chains [26] or search missions [18].

[2]http://search.yahoo.com/

From this log we sampled three subsets, the BURSTY queries subset, the STABLE queries subset and the RANDOM query subset.

Selecting bursty queries

Given the huge amount of data to process, we did an iterative process in which we started by sampling uniformly at random sessions that contained at least one of 1,400 "torso" queries (having frequencies that were neither too low nor too high), and continued by rounds deepening (sampling more sessions) and narrowing (sampling less queries) our sample. The process was completed with a full sample of all the user sessions during 13 months containing 190 queries that are *bursty during a single period*. In our experiments we set $\beta \geq 3.5$, meaning that the $b(q,t)$ index must be 3.5 times larger than the average. We also set $\delta \geq 3$, meaning that the duration of the single episode must be of 3 days or more. Figure 3.1(a) shows the normalized frequency of a few queries in this sample.

Selecting stable and random queries

For the STABLE set, we set the maximum standard deviation of b to 0.5 during the entire year, obtaining a set of 768 stable queries candidates, and sub-sampled 200 queries from this set using the empirical frequency distribution from our BURSTY sample. Figure 3.1(b) shows the normalized frequency of queries in this sample.

To select the RANDOM queries we first binned the bursty queries based on their frequency during the episode. Then from each bin we randomly selected queries having a 1-year frequency at most three times larger to ensure that pseudo-episodes have complete pre-episodes and post-episodes periods. Using this process we created a sample of 340 queries.

3.5.2 Metrics

To characterize the queries we chose to use a broad set of different metrics that covers different aspects of the search queries. The first three groups are computed for each particular period (pre-episode, episode, and post-episode), while the last group is computed for the entire time series:

- *Activity/effort metrics* capture in general how much effort users invest in locating information.

- *Attention metrics* show the concentration of user clicks.
- *Comparative metrics* compare the behavior of users between two periods.
- *Global metrics* include general properties of the query being analyzed.

Activity/effort metrics

The first group of metrics captures the users' effort in finding the information they searched for. Most of these metrics are session-level, in which a session is a set of related queries obtained using the method in [5].

For a given query q, these metrics include:

- Session duration: average duration in seconds of sessions containing q, this is the time from the first query in the session to the last query (or click on a search result).
- Dwell-time: average time in seconds from an occurrence of q to the next query done by the user, limited to 30 minutes.
- Queries/sess.: average number of queries in sessions containing q.
- Clicks/sess.: average number of clicks on search results in sessions containing q.
- Events/sess.: average number of events per session, including queries, clicks on search results, and clicks on the pagination links "previous-page/next-page".
- Clicks/query: number of clicks on search results, on average, after a query q and before the next query in each the session.
- Non-clicks %: fraction of issued queries that are not followed by a click on a search result (either because none of the results was relevant, or because the user found the information directly on the document snippets shown in the search results).
- Assistance %: fraction of query reformulations that were the result of a search suggestion. Most search engines display for some queries a few suggested queries, usually with a label such as "also try" or "related searches". This variable measures how often, when doing a reformulation, users click on one of these suggestions instead of typing by themselves a new query.

- USERS/QUERY: number of distinct users issuing q, divided by number of occurrences of q. A small number indicates that a small group of users is repeatedly issuing the same query. A large number indicates that the query is of interest to a larger audience.

Attention metrics

The second group corresponds to a variety of metrics that describe how concentrated or disperse are users' clicks on search results. For a particular period (episode or pre/post-episode) and a specific query, we sort the URLs clicked for that query during the period in decreasing order according to the observed click probability. In the following, the "top URL(s)" for a period are the most clicked search results. This usually, but not always, matches the ordering in which URLs are shown to users, because of positional bias [9]. These metrics include:

- DISTINCT URLS: number of distinct search results clicked.
- TOP-1 SHARE: fraction of clicks on the search result with the highest number of clicks for a query. For example, if a query q has a frequency of 10, and the highest clicked-on returned page 6 clicks, then the TOP-1 SHARE is 60%.
- URLS 90%: minimum number of search results required to cover 90% of users' clicks.
- RANK-CLICK DROP: steepness of rank-click frequency curve, measured by the exponent resulting of fitting a power-law to the curve of click probability.
- CLICK ENTROPY: entropy of the distribution of clicks on search results, as used in [23], for every query q and a set of clicked results U. This is defined as:

$$H(q) = \sum_{url \in U_q} p(url|q) \times \log p(url|q) \tag{3.3}$$

The first three attention metrics are straight forward and obtained directly from the query log, while the last two are slightly more complex and encompass the full click share distribution. The motivation for using RANK-CLICK DROP as a measure of attention is in the long-tailed nature of clicked distribution. If we fit a power law function of a form $y = ax^{-\alpha} + \epsilon$ to the click distribution, the value of the (positive) parameter α suggests the steepness of a power law

curve. The steeper a curve is the head of the distribution has most of the clicks, and therefore we can say that users attention is focused on a section of the results. Similarly to this, CLICK ENTROPY tells us how much information bits of a query a url "carries". It has previously been used as a proxy for how difficult it is to satisfy the information need behind a query [23]. A higher CLICK ENTROPY indicates more disperse clicking (users click on more different documents) suggesting a more complex search, since users need to read more documents in order to satisfy their information need. The converse is also assumed: lower entropy indicates that users click on a smaller subset of the search results, suggesting that their information need is somehow easier to satisfy.

Comparative metrics

The third group of metrics compares different periods of time (e.g.: pre-episode and post-episode), focusing on changes in their click probability distributions. The goal of these metrics is to discover what is the impact of the query burst on the share of users' attention received by different search results.

- CLICK DIVERGENCE: KL-divergence of click distributions. For a query q, a set of URLs U, and two periods, t_1, t_2, the KL-divergence is defined as:
$$D_q\langle t_1||t_2\rangle = \sum_{url\in U} P(url|q,t_1) \times \log \frac{P(url|q,t_1)}{P(url|q,t_2)} \tag{3.4}$$
- TOP-1 CHANGE: difference in the probability of the URL with the highest probability in the first period with respect to the second period.
- TOP-N OVERLAP: overlap of URLs sorted by click probability, at position $n = 1$ and $n = 5$, between the two periods.

We also considered variations in the activity/effort and attention metrics, e.g.: differences in DISTINCT URLS.

Global metrics

The fourth group of metrics considers the entire time-series:

- PEAK BUILD-UP RATIO: for a URL u, this is the difference between the episode peak, and the first date in which u is seen. This is normalized using the difference between the episode peak and the start of the dataset.

For instance, a value of 1 indicates the URL has existed since the beginning of the observation period, and a 0 indicates it was created the day of the peak of the query burst. Other cases are simply linearly interpolated, as described in Section 3.7.2.

- Burst intensity: the b index described in Section 3.4.1.

3.6 Characterizing query bursts

The broad variety of topics that are covered by bursty queries (as can be seen in the Appendix A, suggest that the nature of the underlying events which caused the bursts, and the way they develop, are also different. We wish to discover the different patterns of query bursts based on user behaviour during them. Apart from topical categories of queries, we would expect differences between query bursts related to new entities, e.g.: a criminal case involving a previously not-well-known person; and query bursts related to existing entities, e.g.: a new movie by a known director. We would also expect differences between query bursts occurring periodically e.g. every year, and query bursts occurring for the first time.

Our main goal is a descriptive analysis of bursty queries, with the goal of discovering features that point to different classes of bursty queries. Therefore, the first application of the metrics we described in Section 3.5.2 is to the characterization of different types of query bursts. Since there is no ground truth for this type of classification, we choose to discover different types of bursts using an unsupervised approach. For this we apply k-means clustering algorithm using all the metrics extracted as input features.

We experimented varying the number of clusters from two to 30 and found no clear evidence of an inherent number of clusters in the data (e.g.: looking at the sum of distance square from clusters centroids, there is no steep drop when increasing the number of clusters).

We use a partition into three clusters because it uncovers clusters with distinct features and an easy-to-grasp interpretation, and because it is also useful in practice for the predictive task of Section 3.7.3. A high-level depiction of the clusters and the relative influence of the features to each cluster is shown in Figure 3.3. The distribution of queries over three clusters was: 76 in cluster A, 66 in cluster B, and 48 in cluster C. The list of queries on each cluster is included in Appendix B.

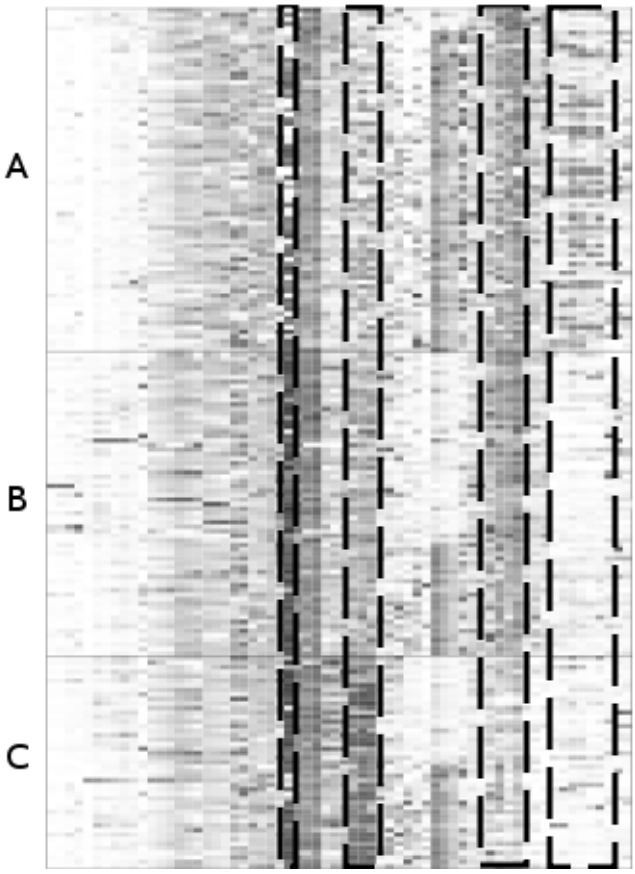


Figure 3.3: Depiction of the relative influence of features in the obtained clusters. Each row represents a bursty query (rows are sorted by similarity), and each column a feature. The most important features are marked by the rectangles in the following order (from the left): Peak build-up ratio, Top-1 share(for 3 periods), Click entropyand Rank-click drop(for 3 periods each), Top-5 and all Click divergence(all comparisons).

3.6.1 Types of bursty queries

Next, we inspected the queries in each cluster, and their feature values, to try to understand which were their key characteristics. Our interpretation of the clusters is the following:

Type A: bursts that fade out completely afterwards. These queries are not very frequent during the pre-episode, and fade away quickly in the post-episode. They have a high divergence (high CLICK DIVERGENCE) between the pre- and post-episode, meaning that the episode changes completely the search results for the query. There is also no strong authoritative URL (low TOP-1 SHARE, high CLICK ENTROPY), which partially explains why click share is so strongly affected by the episode.

This cluster contains many queries related to entertainment, some examples are: *katt wiliams, super bowl 2009 commercials, snl sarah palin, jett travolta, air car, kawasaki disease*. Typical behaviour of this type can be represented by the query *snl sarah palin*. The mentioned TV show caused a huge increase of the queries' frequency, and created a new, previously non-existing, topic without an authoritative source. These are "buzz" topics that after an initial hype quickly lose the interest of the users.

Type B: bursts that create new topics. These queries are also not very frequent during the pre-episode, but contrary to Type A, they maintain some presence in the post-episode. They have a less dominant top URL (medium TOP-1 SHARE) and less click concentration (medium CLICK ENTROPY).

This cluster contains many queries related to new scientific/technical developments and events that have long-term effects, for instance: *2008 olympics, joe biden, obama mccain polls*. For example, the information on *2008 olympics* is present long before the games commence, but it is the start of the games that triggers the increased user interest in the topic, and changes the click distribution to, in this case, sporting events result pages.

Type C: bursts on existing topics. These queries appear both in the pre-episode and in the post-episode with non-negligible frequency. They have an authoritative top result with a high click share (high TOP-1 SHARE) and a low CLICK ENTROPY, so the users' attention is concentrated. For these queries, the episode does not change the distribution of clicks, reflected by the fact that the CLICK DIVERGENCE is low.

This cluster contains many queries related to topics that are searched during the entire year, but for which a real-world event triggers heightened user interest. Examples: *teen choice awards, national hurricane center, saturday night live.* For example, the burst of *saturday night live* is caused by the same previously discussed TV appearance of U.S. politician Sarah Palin, but the query itself is present before that particular show and its burst does not have long-lasting effects on the search results for the query.

Remark. This classification of query bursts matches the classes of bursts predicted by the model of Crane and Sornette [8] using completely different methods. Type A corresponds to exogenous sub-critical, expected in cases of external events that do not propagate well virally. Type B corresponds to exogenous critical, expected in cases of external events that are highly viral. Type C corresponds to endogenous critical, expected in cases of internally-motivated messages that are highly viral.

3.6.2 Characteristics of query bursts

Next, we look at specific sets of metrics, studying them during the *pre-episode*, *episode*, and *post-episode* periods defined as in Section 3.4.2. With respect to query bursts, our main findings can be summarized as follows:

1. Per-user effort/activity is higher during query bursts.
2. Users' clicks are more concentrated during query bursts.

These findings are supported by changes in multiple query attributes during the query burst, as detailed in the rest of this section.

User effort/activity is higher during query bursts

Table 3.1 shows an increase in several metrics of activity/effort for bursty queries during the *episode* compared to pre-episode and post-episode. During the *episode*, sessions are not significantly longer in duration, but contain more queries, more clicks, and more events in general; also more individual sessions have clicks.

Bursts of query activity are driven mostly by an increase in the number of users issuing the query, given that the ratio USERS/QUERY does not change

Metric	Pre-	Episode	Post-	Stable
Session duration	1768.6	1886.00	1624.10	2238.1**
Dwell-time	175.13	178.00	157.80	216.7*
Events/sess.	5.06***	7.64	4.57***	4.69***
Queries/sess.	2.67***	3.19	2.28***	2.14***
Clicks/sess.	2.29***	3.73	1.96***	1.87***
Clicks/query	0.79	1.81	1.39	0.86***
Assistance %	11.90***	13.18	12.29***	4.69**
Non-clicks %	35.97***	28.22	41.84***	22.25***
Users/query	1.47*	1.65	1.47	2.87**

Table 3.1: Averages of activity/effort metrics from Section 3.6.2. Statistically significant differences with episode: $p < .01$ (***), $p < .05$ (**), $p < 0.10$(*)

significantly. The fact that on average users click on search assistance more often during the episode, may indicate less familiarity with the topic being queried; the comparison with the stable queries also points in that direction.

Query-sessions during the *episode* are in general more "intense" than regular search sessions. This increase may be due to a number of causes, including increased interest and increased difficulty in locating information. Given that most episodes tend to be short (Table 3.3), the effect of the episode in effort and activity could be attributed more to increased user interest.

We find that feature Assistance % which measures the fraction of query reformulations that are the result of clicking on a search suggestion, exhibits an interesting behavior from the point of view of query bursts. Figure 3.4 shows distributions of Assistance % for burst episode, pre-episode, post-episode, and stable queries. Higher values for the burst episodes suggest that users click on the search suggestions more during the burst. On the other hand, for the Stable queries users do not do this so frequently.

A possible hypothesis, whose empirical analysis goes beyond the scope of this research, is that users who participate in a query burst become "activated" after the signals they receive go beyond an activation threshold (see e.g. [16]). In other words, users who query about a topic for the first time, must be sufficiently interested in the topic to query about it.

Comparison with stable queries: in terms of activity/effort, stable queries are part of longer sessions with less events, hence longer dwell times. Stable queries also have much less use of search assistance.

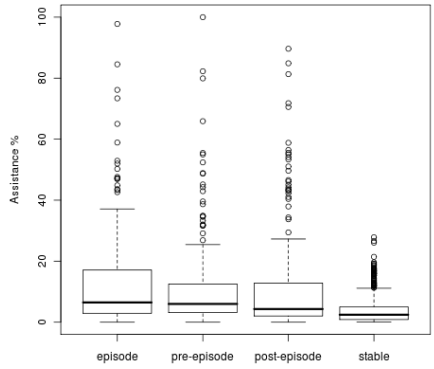


Figure 3.4: Distribution of the fraction of query reformulations that are the result of clicking on a search suggestion (feature Assistance %) for burst episodes, pre-episode, post-episode, and stable queries.

Metric	Pre-	Episode	Post-	Stable
Top-1 share	0.52	0.56	0.52	0.71***
Rank-click drop	1.15***	1.01	1.10***	0.55***
Click entropy	1.54**	1.44	1.61***	0.93***
URLs 90%	5.12	4.40	5.46**	4.69
Distinct URLs	32.95	35.57	41.03*	59.17***

Table 3.2: Averages of concentration metrics from Section 3.6.2.

Clicks are more concentrated during episodes

Table 3.2 shows that clicks tend to be more concentrated during the query burst, than in the pre-episode and post-episode periods. The share of clicks of the single top URL does not change significantly, but click probabilities on the top clicked URLs are higher, as evidenced by a more steep rank-click drop and a lower entropy.

In the post-episode, there is an increase in the number of distinct URLs, and the number of search results required to cover 90% of the clicks, indicating that new relevant search results are present after the query burst.

Table 3.2 shows that there are no statistically significant differences between Top-1 share before, during the burst episode, and after it. We investigated

the concentration of users on all results. For this we used RANK-CLICK DROP and CLICK ENTROPY measuring concentration of users on a portion of search result. Figure 3.5 shows in more detail the distribution of the two measures. For both, the results are aligned and show that during the burst episode users attention is more concentrated than before and after it. This suggests that during the bursts users are interested in some specific information relevant to the query. As expected, for the stable queries users clicks are less dispersed than for the bursty ones. There is a larger number of documents that are clicked (DISTINCT URLS) but the share of clicks most documents receive is small.

Comparison with stable queries: it is clear that stable queries have clicks that are even more concentrated at the top that in the case of bursty queries, according to all metrics we examined. Information relevant to stable queries changes rarely, and thus the top documents satisfy user information needs by themselves.

3.6.3 Relationship with news searches

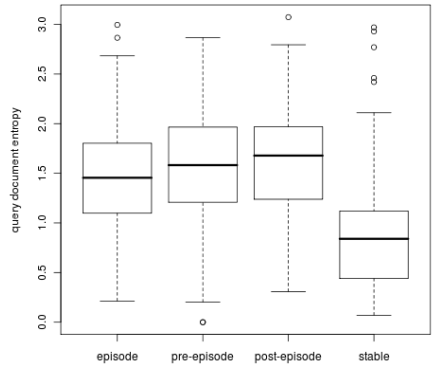
In this section we introduce data obtained from a news search engine[3] whose sources are from an editorially-selected list of thousands of news providers such as CNN, BBC, etc. In the following, we refer to general web search logs as "web searches" and to news search logs as "news searches". We use one year of news searches (from the same year as web searches).

Specifically, we seek to uncover (1) if there is a correlation of the query frequencies in web search and news search; (2) if there is a dependency between bursts; and (3) if there are differences in query burst intensity and duration.

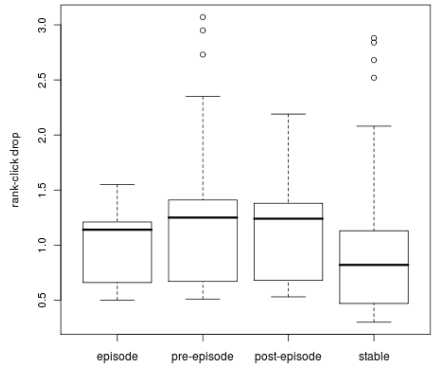
Intuitively, in the case of news searches, one may expect that query bursts would tend to appear after an event is reported by traditional media. However, in our BURSTY sample from web search, we observe many queries about subjects that would not be considered as newsworthy by traditional media (e.g.: "`fallout 3 walkthrough`" , "`big brother spoilers`", etc.). Hence, we believe that in the case of web searches, query frequencies are often not related to the presence of a topic in news reports.

Looking at their entire one-year time series, we checked if the frequencies in web searches and new searches are correlated. Measuring the Pearson correlation

[3] http://news.search.yahoo.com/



((a)) Click entropy

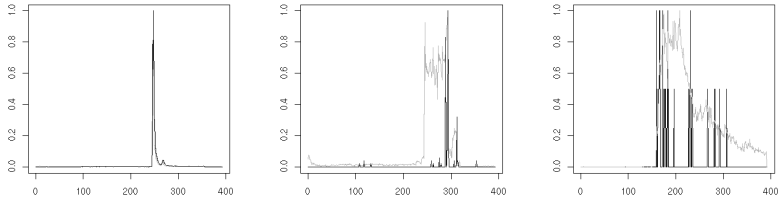


((b)) Rank-click drop

Figure 3.5: Distribution of concentration measures Click entropy (a) and Rank-click drop (b).

coefficient between these series for each query, we find values that vary widely from very strong correlation to very weak correlation (median $r = 0.7$).

Burst alignment. Alignment between time series of frequency in different search systems is not perfect, as observed in [1]. The measure we used for capturing the intensity and the length of a query burst does not guarantee that



((a)) aligned ((b)) non-aligned ((c)) non-captured

Figure 3.6: Normalized frequencies for three queries in web searches (light gray) and news searches (dark gray). Three distinct cases are shown: (a) aligned bursts, (b) non-aligned bursts, (c) non-captured burst.

the captured bursts in the two logs are in the same time period. We identified three possible cases of alignment between the web and the news queries: "non-captured", "aligned", and "non-aligned". Figure 3.6 shows different cases of burst alignment.

We took the queries from the BURSTY sample and observed all their occurrences in the news search log. The first thing we checked if bursty queries from the web searches appear in the news searches at all. For this we set a threshold of two occurrences per day during the observed year. All queries whose frequency was bellow this threshold were labeled as non-captured. In total we found 59 (out of 190) non-captured queries in news searches.

For the queries that were present both in web and news searches (131 out of 190), we analyzed their burstiness. To discover if they are bursty, we applied the same method from Section 3.5.1. We consider bursts to be aligned when the burst peak in web searches and news searches occur within ten days of each other. Out of 131 queries that appear in both logs, we found 94 to be aligned according to this definition. The rest of the queries were labeled as non-aligned (37 out of 131).

Burst intensity and duration. For the bursts that were captured in news searches, we compared the burst intensity and duration in both types of searches. Burst intensities are measured using the peak of the BURST INTENSITY $b(q,t)$ (defined in Section 3.4.1), and duration is measured in days.

Table 3.3 compares these indicators, incorporating per-cluster values for the clusters from Section 3.6.1. We observe that differences in intensity between

	Frequency		Intensity			Duration (days)		
Cluster	Web	News	Web	∩	News	Web	∩	News
ALL	190	131	4.8	4.9	5.5	7.7	7.9	5.2
A	76	54	5.0	4.9	5.5	7.2	7.4	5.3
B	66	41	4.5	4.6	5.4	7.4	7.7	4.9
C	48	36	5.1	5.3	5.5	7.6	8.8	5.6

Table 3.3: Burst intensity and burst duration in Web and News search logs. The inter-section marks the restriction of queries in Web search to queries discovered in News search logs.

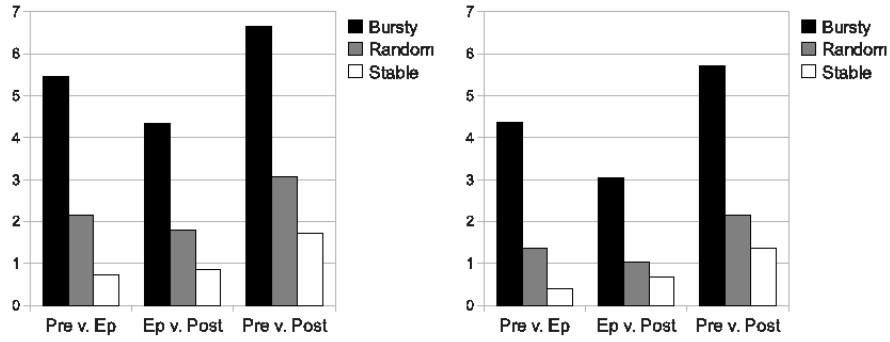
web searches and news searches are minor, but statistically significant at $p \leq 0.01$; they show that bursts in news searches are slightly more intense. Differences in duration are substantial, and indicate that in news searches the average duration of the burst is shorter by at least 2 days. The news searches peaked 0.78 days ($\approx$ 18 hours) before web search on average. Users expect to see results about many emerging topics first on traditional news, consistently with findings in [22] showing that traditional news sites mention new "memes" on average 2.5-hours before other sites. Few days after the initial news event, users will stop using the news search engine to get information about it, apparently, after this period the query is no longer perceived as "news" by users.

3.7 Search results and click share

Next we investigate the effect of the query burst on the distribution of clicks on search results, referred in the following as simply the "click distribution". This distribution is a function of both search engine ranking and page quality.

Basically, we aim to discover if the query burst presents an opportunity for publishing a web page about the topic of the query burst. We expect that documents that exist before the query burst will have the largest share of clicks, but that perhaps new documents can also capture some clicks. Concretely, we investigate the following questions:

- How much is the click distribution changed by the query burst?
- Is it necessary to have a page that existed before the burst to have a large share in the click distribution?
- Is it possible to predict the share of new documents during the burst?



((a)) All results ((b)) Top-5 results

Figure 3.7: Change in click distributions for BURSTY, RANDOM, and STABLE queries, measured using KL-divergence.

3.7.1 Changes in click share

We measure the effect of the *episode* in the click distribution using the CLICK DIVERGENCE measure defined previously. We compared the click distributions of pre-episode, episode, and post-episode for the BURSTY sample, and *pseudo-episodes* (as defined in Section 3.4.3) for the RANDOM and STABLE samples.

The results shown in Figure 3.7(a) confirm the intuition with respect to the effects of query bursts. According to KL-divergence, the click distribution of BURSTY queries changes on average about 3× and 6× more than for RANDOM and STABLE queries respectively.

If we focus on the top-5 results only, as in Figure 3.7(b), we see that the changes are smaller but the separation between BURSTY queries and the rest is even larger.

3.7.2 Click share of late-comers

When the frequency of a query increases, most content providers that already have pages on the topic will receive an increased number of visits and will thus benefit from the heightened user's interest. Our observations confirm that publishing early represents an advantage.

To quantify how early a URL is published with respect to a query burst, we use the metric PEAK BUILD-UP RATIO of a URL u in query q measures how soon

the URL appears in the query log in comparison with the peak of the query burst. Let $t_{u,q}^{first}$ be the first time the URL u is clicked for query q, and let t_q^{peak} be the time of the peak of the query burst of q. Let t_0 be the beginning of the observation period, then this metric is equal to:

$$\max\left\{\frac{t_q^{peak} - t_{u,q}^{first}}{t_q^{peak} - t_0}, 0\right\} \tag{3.5}$$

A value close to 1 means the URL's first click was close to the beginning of the observation period, while a 0 indicates the URL's first click occurred on the day of the peak. The first click in a specific URL could be observed *after* the episode peak, but this is a rare event and for simplicity of the presentation we truncate those values to zero. In the following, we will refer to documents whose PEAK BUILD-UP RATIO is non-zero as *old pages* (as they existed before the burst) and to documents whose PEAK BUILD-UP RATIO is close to zero as *new pages*.

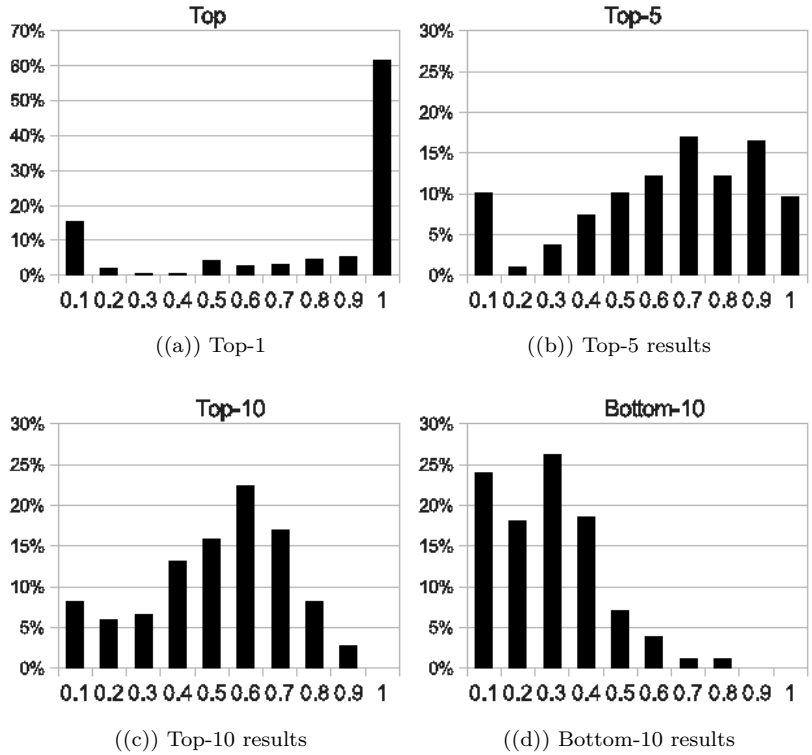
Figure 3.8(a) indicates that 61% of the top-URLs have existed since the beginning of the observation period, while only 16% of the top-URLs are *new pages* created on or after the query burst.

When examining the top-5, top-10, and bottom-10 results (Figures 3.8(b), 3.8(c), and 3.8(d)), we see that publishing late, i.e.: having PEAK BUILD-UP RATIO close to zero, means a lower share of clicks during the episode. For instance in the case of top-10 results, on average about 3 results are new pages, while in the bottom-10 results, on average about 5 results are new pages.

Next, we consider the *share* of clicks the new pages will obtain. This information is presented in Table 3.4, where we take the Top-1, Top-5, Top-10, and All of the *new pages* and look at their click share. In general, they obtain a minority of clicks during the episode (27.5%), and this is distributed among many queries: even the Top-10 most clicked new pages considered together obtain only 8.9% of the clicks.

Our findings from Section 3.6.1 suggest that the click share of at least the top-URL is different across clusters. Therefore, Table 3.4 also includes per-cluster results.

The per-cluster analysis shows that there wide variability among the clusters. The best opportunity for publishing new pages are queries of type A (bursts that fade out completely afterwards) for which they obtain 52.1% of the clicks. Next, for queries of type B (bursts that create new topics) the new pages obtain 25.2% of the clicks. Finally, for queries of type C (bursts on existing topics) the new pages obtain only 9.8% of the clicks; in this last cluster, it is in practice



((a)) Top-1 ((b)) Top-5 results

((c)) Top-10 results ((d)) Bottom-10 results

Figure 3.8: PEAK BUILD-UP RATIO for the (a) the top result, (b) the top-5 results, (c) the top-10 results, (d) the bottom-10 results.

hopeless for a publisher that wants to profit from a query burst, to publish an article about the topic of the burst.

3.7.3 Finding opportunities for late-comers

From the content-providers' perspective, the question of finding *which* are the "waves" that should be ridden is a central one. The resources of the content-providers are limited so they can not write a new page for any bursty query related to their expertise, and moreover the time they have to react is very short given that query bursts do not last for long.

Assuming that not all query bursts can be predicted (some can be predicted, e.g. when they are related to newsworthy events that are planned well in

Query cluster	New URLs	Click share
All queries		%
	Top	3.1
	Top-5	5.5
	Top-10	8.9
	All	27.5
A: bursts that fade out completely afterwards		%
	Top	37.8
	Top-5	41.1
	Top-10	20.2
	All	52.1
B: bursts that create new topics		%
	Top	5.9
	Top-5	5.6
	Top-10	5.2
	All	25.2
C: bursts on existing topics		%
	Top	2.5
	Top-5	3.5
	Top-10	4.2
	All	9.8

Table 3.4: Click share of the new URLs as a percentage of total clicks. Top-k indicates the k most clicked new URLs. "All" indicates all the new URLs

advance), a system that were to help content providers in deciding what to write about, should be able of (a) identifying query bursts and (b) predicting the expected benefit. Question (a) was the subject of Section 3.4.1 while (b) turns out to be more difficult.

As mentioned in the previous sections, the target of this prediction task is the click share of new pages. We first use a logistic regression model (M_p) with the features from the pre-episode and episode described in the Section 3.5.2. Its performance, measured using the correlation coefficient between the predicted click share and the actual click share for a hold-out test set of queries is reported in Table 3.5.

The insights from Table 3.4 can be used to improve this prediction, given that the average share of newly published pages depends clearly on the cluster to which the query belongs. Thus, we build a model (M_c) that first computes the probability of a query belonging to each cluster using a Naive Bayes classifier, and then includes these predictions in the logistic regression model. Table 3.5

Model	Top	Top-5	Top-10	All
Simple model M_p	0.59	0.71	0.69	0.42
Cluster-based model M_c	0.64	0.77	0.77	0.46

Table 3.5: Correlation coefficient between predicted and actual click share of new documents.

shows the correlation coefficients between the original and predicted values and the improvement that the cluster prediction brings. The results show that it is hard to predict the values for all the pages and for the very first page, while a fair performance can be obtained with Top-5 and Top-10 results.

3.8 Conclusions

Query bursts are observed in a search engine query log whenever there is increased interest in a certain topic. Looking back at our fictional search "marketplace" for the main market components we discovered that:

- Not all queries are equal and that there are distinct types of query bursts (*goods*). Our research over a 1-year-long query log uncovered different types of query bursts, including (A) bursts that fade out completely afterwards, (B) bursts that create new topics, and (C) bursts on existing topics.
- The analysis of several metrics indicates that during query bursts users invest more effort in search and that their clicks are concentrated on a smaller group of search results (*price*).
- Publishing documetns (*supply*) early, before the bursts, is the only way towards obtaining the proportion of increased user attention. For some queries, publishing during the queries can lead to the non-trivial click share.
- After the query burst, the distribution of clicks (*market share*) into search results for a query is substantially different from that before the query burst.

Based on these findings the main stakeholders in a search market may take different strategies during the query bursts.

Content providers that intend to capture users' attention on emerging topics should attempt to publish early. If not, they should target query bursts on

topics that did not exist before (types A and B). Writing during a query burst about a previously-existing topic is unlikely to yield a substantial share of clicks.

Search engines should, according to our findings, treat queries undergoing query bursts differently. For instance, search suggestions are much more important for these queries. A search engine may introduce user-interface changes to support the needs of users entering bursty queries.

We consider this work as a part of a broader effort, which is to provide the right signals about users' needs to web authors. Search engines should help to detect scarcity of information on certain topics so that content providers can supply this information. A system that were able of telling a content provider e.g. "if you write about environmental issues, you should be writing about solar energy", would be a big step forward for the Web ecosystem.

This involves creating models that also take into account content providers' features such as topic, influence and authority, and that are able to detect users' unsatisfied needs for information in certain areas. A promising approach to this problem would be to perform a topic-sensitive analysis in which queries (and pages) are classified into topical categories, and then studied independently for each topical category.

Acknowledgements: the authors thank Aris Gionis for his help, and Bettina Berendt, Yoelle Maarek and Ingmar Weber for helpful comments on an earlier version of this manuscript.

References

[1] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble. Why we search: visualizing and predicting user behavior. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 161–170, New York, NY, USA, 2007. ACM.

[2] S. Asur and G. Buehrer. Temporal analysis of web search query-click data. In *WebKDD/SNAKDD 2009: Web Mining and Social Network Analysis Workshop*, Paris, France, 2009. ACM Press.

[3] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. Temporal analysis of a very large topically categorized web query log. *J. Am. Soc. Inf. Sci. Technol.*, 58(2):166–178, 2007.

[4] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 321–328, New York, NY, USA, 2004. ACM.

[5] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 609–618, New York, NY, USA, 2008. ACM.

[6] L. Chen, Y. Hu, and W. Nejdl. Using subspace analysis for event detection from web click-through data. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 1067–1068, New York, NY, USA, 2008. ACM.

[7] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 2–11, New York, NY, USA, 2005. ACM.

[8] R. Crane and D. Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008.

[9] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 87–94, New York, NY, USA, 2008. ACM.

[10] D. Downey, S. Dumais, and E. Horvitz. Heads and tails: studies of web search with common and rare queries. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 847–848, New York, NY, USA, 2007. ACM.

[11] D. Downey, S. Dumais, D. Liebling, and E. Horvitz. Understanding the relationship between searchers' queries and information goals. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 449–458, New York, NY, USA, 2008. ACM.

[12] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 181–192. VLDB Endowment, 2005.

[13] Google Inc. Google Corelate. http://correlate.-googlelabs.com, 2009.

[14] Google Inc. Google Flu Trends. http://www.google.org/flutrends/, 2009.

[15] Google Inc. Google Trends. http://www.google.com/trends/, 2009.

[16] M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83(6):1420–1443, 1978.

[17] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Proc. & Mgmt.*, 36(2):207–227, March 2000.

[18] R. Jones and K. L. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proceeding of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 699–708, New York, NY, USA, 2008. ACM.

[19] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7:373–397, October 2003.

[20] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. volume 8, pages 159–178, Hingham, MA, USA, June 2005. Kluwer Academic Publishers.

[21] T. Lau and E. Horvitz. Patterns of search: analyzing and modeling web query refinement. In *Proceedings of the seventh international conference on User modeling*, pages 119–128, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[22] 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*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM.

[23] Q. Mei and K. Church. Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 45–54, New York, NY, USA, 2008. ACM.

[24] Pew Research Center. Internet Overtakes Newspapers As News Outlet. http://pewresearch.org/pubs/1066/internet-overtakes-newspapers-as-news-source 2008.

[25] Pew Research Center. The New News Landscape: Rise of the Internet. http://pewresearch.org/pubs/1508/internet-cell-phone-users-news-social-experience.

[26] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 239–248, New York, NY, USA, 2005. ACM.

[27] M. Richardson. Learning about the world through long-term query logs. *ACM Transaction on the Web*, 2(4):1–27, 2008.

[28] C. Silverstein, H. Marais, M. Henzinger, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33:6–12, September 1999.

[29] I. Subasic and B. Berendt. Discovery of interactive graphs for understanding and searching time-indexed corpora. *Know.and Inf. Sys.* volume 23, pages 293–319, 2010.

[30] I. Subasic and C. Castillo. The effects of query bursts on web search. In *2010 IEEE/ACM International Conference on Web Intelligence-Intelligent Agent Technology (WI-IAT)*, pages 374–381. IEEE, Aug. 2010.

[31] Y. Sun, K. Xie, N. Liu, S. Yan, B. Zhang, and Z. Chen. Causal relation of queries from temporal logs. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 1141–1142, New York, NY, USA, 2007. ACM.

[32] 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*, SIGMOD '04, pages 131–142, New York, NY, USA, 2004. ACM.

[33] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 424–433, New York, NY, USA, 2006. ACM.

[34] Yahoo! Inc. Yahoo! Clues. http://clues.yahoo.com, 2011.

[35] Q. Zhao, S. C. H. Hoi, T.-Y. Liu, S. S. Bhowmick, M. R. Lyu, and W.-Y. Ma. Time-dependent semantic similarity measure of queries using historical click-through data. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 543–552, New York, NY, USA, 2006. ACM.

[36] Q. Zhao, T.-Y. Liu, S. S. Bhowmick, and W.-Y. Ma. Event detection from evolution of click-through data. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 484–493, New York, NY, USA, 2006. ACM.

Appendix A:

BURSTY Queries per Cluster

Cluster A ("bursts that fade out completely afterwards"): criselda volks scandal, kawasaki disease, groundhog day, oj simpson, gi joe, jessica simpson weight gain, hgtv dream home, fiesta bowl 2009, groundhog day 2009, saturday night live sarah palin, cyber monday deals, christian bale, super bowl commercials, polling place, gustav, snl sarah palin, hgtv dream home giveaway, jett travolta autism, superbowl commercials, blackberry storm release date, kimbo slice vs ken shamrock, michael phelps bong, last day to register to vote, jett travolta, ground hog day, kawasaki syndrome, gi joe trailer, cyber monday sales, is katt williams dead, plaxico burress, go daddy commercial, california propositions 2008, hurricane gustav, brooke satchwell, wwe svr 2009, kelly preston, hurricane hanna, neel kashkari, halle berry baby photos, deborah lin, energy saving tips, cyber monday 2008, super bowl 2009 commercials, caylee anthony update, bristol palin, compressed air car, samantha mumba, mary-kate olsen, superbowl ads, cyber monday, octuplets, misty may, peanut butter recall, michael phelps smoking, fallout 3 walkthrough, anne pressly, successful resume examples, sarah palin vogue magazine, palin, the strangers true story, josiah leming, super bowl ads, latest presidential polls, michael phelps girlfriend, election map, if i were a boy lyrics, niki taylor, free christmas wallpaper, bernie mac illness, montauk monster, katt williams dead, air car, virginia themadsen, soyouthinkyoucandance, volam.com.vn, brangelina twins.

Cluster B ("bursts that create new topics"): obama mccain polls, black friday 2008, pineapple express, ducati 1098, register to vote online free, where to vote, morgan freeman, groundhog, register to vote online, scientology, kimbo slice, lita ford, houston weather, cybermonday, tropic thunder, big brother 10 spoilers, sarah palin, where do i vote, taylor swift, turbo tax online, electoral votes, sophie okonedo, madden 09, presidential polls, brett favre, zuleyka rivera, chinese new year 2009, tina fey scar, voting locations, voting, bill ayers, register to vote, breaking dawn, (redacted: adult query), election polls, free turbotax, kimbo, us open tennis, prop 8, burning man 2008, the curious case of benjamin button, mary mccormack, black friday, gina carano drunk, kathy griffin, hotjobs yahoo com, transformer 2, john travolta, labor day, hurricane katrina, poea open jobs in canada, voter registration, marley and me, olympics, bernie mac, the mummy, labor day 2008, irs refund status, john mccain, www.azmoon.com, 2008 olympics, twilight book, sarcoidosis, anthrax, joe biden, michael phelps, cindy mccain.

Cluster C ("bursts on existing topics"): elite xc, tampa bay rays, saw 5, puppy bowl, teen choice awards, cell for cash, taxact, turbotax online, fiesta bowl, hurricane center, special k, christian songwriting, lollapalooza, pixie hollow, rasmussen poll, www.mysoju.com, turbotax.com, www.watchmovies.net, bradley effect, turbotax, can i vote, obama stimulus package, gallup poll, mda telethon, khou, the mole, us open, white sox, mccain, snl, shawn johnson, gallup, hurricane tracker, taxact.com, khou.com, kprc, republican national convention, chicago white sox, gi joe movie, fdic, playatmcd.com, taxact online, click2houston, saturday night live, butterfinger, www.pch.com, national hurricane center.

Appendix B:

STABLE Queries

Sample of queries that seldom fluctuate in frequency: holland america, national geographic channel, midas, rheumatoid arthritis, dudetube, baby depot, dereon, jimmy johns, essence, ac moore, tribal tattoos, court tv, zoloft friends reunited, viewpoint bank, redtub, boston market, car payment calculator, heidi klum, chicos, af portal, low income apartments, postsecret, philadelphia, mspace, tiger airways, liberty university, ftvgirls, charmeddisney movie club, photography, hydrocodone, mike in brazil, tribune review, yahooligansl, (redacted: adult query), spiegel, netflex, pal, bitcomet, toutube, mr skin, greek mythology, extenze, ebay motors parts, paint colors, stupid videos, english to french translation, yout, vans shoes, bigtitsatschool, pump it up, spa.gov.my, veterans administration, radisson hotel, myspace music, education, candylist, us navy, the gas company, arizona, mcdonald's, nylottery.org, coke rewards, slacker, googlemap, american airline, valley national bank, sports authority store, new jersey lottery, gimp, commerceonline, west elm, university of chicago, mta nyc, knotts berry farm, dragon fable, flicker photo site, alienware, american signature furniture, intervention, akhbar harian metro, city of houston, south bend tribune, sims, pink eye, tabnak, compaq, shyla stylez, cms, faa, suze orman, crigslistlist, malibu strings, asda, long and foster, democrat and chronicle, acs student loan, la fitness locations, basspro, kiss fm,ethan allen, texas child support red, happy birthday, quixtar, hotmai, dailyniner, adolf hitler, hepatitis, baskin robbins wirefly, usps tracking number, simslots, honolulu star bulletin, department of homeland security adobe acrobat reader, pancreatitis, american standard, alloy, at&t universal card, web, red roof inn, jc penney catalog, lexmark drivers, gsc, genealogy, pc world, qoutes, arby's, press democrat, bentley, penndot, kbr, sony digital camera,

whole foods market, belize, sheboygan press wynn las vegas, randy blue, inquirer, baby boy names, el salvador, tampa tribune, ohio university myspace', sexyclips, kementerian sumber manusia, kentucky fried chicken, marriott rewards, ace, sugarland, brazil, cold stone creamery, celebrity hairstyles, coast to coast am, starbucks locations, bargain news, yahoo malaysia, general electric, collections etc, terra, proactiv, cheap ticket, crohn's disease, spanx, entergy, wthr, bipolar disorder, currency calculator, tillys, 1800contacts, galottery, odd news, virginia, albert einstein, (redacted: adult query), (redacted: adult query), trilulilu, adobe photoshop,spybot search and destroy,sean cody cover letter,hartford courant, citicard, goodyear tires, advanced auto parts, metric conversion mary kay, kaiser permanente california, hotmail email, rapidshare, baby names meaning, sherwin williams wescom credit union, cialis, cathay pacific, livejournal, subaru, netflix.

Errata

- Section 3.1, page 46, paragraph 1: *Internet has become* should be The Internet has become;
- Section 3.1, page 46, paragraph 1: *staying informed about the current news events* should be staying informed about current news events;
- Section 3.1, page 46, paragraph 3: *the click share they could obtain* should be the click share content providers could obtain;
- Section 3.2, page 46, paragraph 4: *including television, Internet, radio* should be including television, the Internet, radio;
- Section 3.2, page 46, paragraph 4: *since 2008, in for the general public the U.S., Internet is* should be since, 2008, for the general public in the U.S., the Internet is;
- Section 3.2, page 46, paragraph 4: *by 2010, Internet was the source* should be by 2010, the Internet was the source;
- Section 3.2, page 47, paragraph 2: *for the content providers we can measure their market share by the number* should be we can measure the market share of the content providers by the number;
- Section 3.2, page 47, paragraph 3: *During the query bursts* should be During query bursts;
- Section 3.2, page 47, paragraph 6: *how does higher interest in a query change their behaviour* should be how higher interest in a query changes their behaviour;
- Section 3.2, page 48, paragraph 1: *Same queries have higher burst* should be The same query has a higher burst;
- Section 3.2, page 48, paragraph 1: *the distribution of clicks into search results* should be the distribution of clicks among search results;
- Section 3.2, page 48, paragraph 1: *a late-comer indeed has an opportunity of obtaining* should be a late-comer indeed has the opportunity of obtaining;
- Section 3.3, page 49, paragraph 1: *the authors of [2] study if* should be the authors of [2] study whether;
- Section 3.3, page 49, paragraph 1: *instead of topical categories authors look for differences* should be instead of topical categories, the authors look for differences;

- Section 3.3, page 49, paragraph 2: *A related study by Adar et al. ([1])* should be A related study by Adar et al [1];
- Section 3.3, page 49, paragraph 2: *uncovered semantically similarity* should be uncovered semantic similarity;
- Section 3.3, page 49, paragraph 2: *Sun at al.* should be Sun et al.;
- Section 3.3, page, 49, paragraph 2: *our paper focuses on differences on user behaviour* should be our paper focuses on differences in user behaviour;
- Section 3.4.1, page 50, paragraph 4: *test for query bursts detection* should be test for query burst detection;
- Section 3.4.3, page 52, paragraph 6: *we want to study if a phenomenon is related* should be we want to study whether a phenomenon is related;
- Section 3.4.3, page 53, paragraph 3: *during, before, and after li burst* should be during, before, and after a burst;
- Section 3.5.2, page 55, paragraph 3: *directly on the document snippets* should be directly in the document snippets;
- Section 3.5.2, page 56, paragraph 2: *concentrated or disperse* should be concentrated or dispersed
- Section 3.5.2, page 56, paragraph 3: *metrics are straight forward and obtain directly* should be metrics are straightforward and obtained directly;
- Section 3.6, page 58, paragraph 2: *(as can be see in the Appendix A suggest* should be (as can be seen in Appendix A) suggests;
- Section 3.6, page 58, paragraph 2: *based on user behaviour during them* should be based on user behaviour during query bursts;
- Section 3.6, page 58, paragraph 2: *periodically e.g. every year* should be periodically, e.g. every year
- Section 3.6, page 58, paragraph 3: *metrics we described* should be metrics described;
- Section 3.6, page 58, paragraph 4: *an inherent number of clusters in the data* should be an inherent number of clusters in the cluster number evaluation data;
- Section 3.6.1, page 60, paragraph 4: *episode changes completely* should be episode completely changes;

- Section 3.6.2, page 62, paragraph 5: *as part of longer sessions with less events* should be as part of longer sessions with fewer events;
- Section 3.6.2, page 63, paragraph 1: *during the query burst, than in the pre-episode* should be during the query burst than in the pre-episode;
- Section 3.6.2, page 63, paragraph 1: *evidenced by a more steep rank-click drop and a lower entropy* should be evidenced by a steeper rank-click drop and lower entropy;
- Section 3.6.2, page 64, paragraph 1: *documents that are clicked (DISTINCT URLs) but the share* should be documents that are clicked (DISTINCT URLs), but the share;
- Section 3.6.2, page 64, paragraph 2: *more concentrated at the top that in the case* should be more concentrated at the top than in the case;
- Section 3.6.3, page 64, paragraph 4: *we seek to uncover (1) if there is a correlation of the query frequencies in web search and news search; (2) if there is a dependency between bursts; and (3) if there are differences in query burst intensity and duration.* should be we seek to uncover (1) whether there is a correlation of the query frequencies in web search and news search; (2) whether there is a dependency between bursts; and (3) whether there are differences in query burst intensity and duration.;
- Section 3.6.3, page 64, paragraph 6: *we checked if the frequencies* should be we checked whether frequencies;
- Section 3.6.3, page 66, paragraph 4: *To discover if they are bursty* should be To discovery whether they are bursty;
- Section 3.6.3, page 66, paragraph 4: *we applied the same method from* should be we applied the method from;
- Section 3.6.3, page 67, paragraph 1: *Few days after* should be A few days after;
- Section 3.7.2, page 68, paragraph 4: *the heightened user's interest* should be the heightened users' interest;
- Section 3.7.2, page 69, paragraph 7: *there wide variability* should be there is wide variability;
- Section 3.7.2, page 70, paragraph 1: *are limited so thay can not write a new page* should be are limited so they cannot write a new page;
- Section 3.7.2, page 70, paragraph 1: *profit from a query burst, to publish an article* should be profit from a query burst to publish an article;

- Section 3.7.3, page 71, paragraph 1: *should be able of* should be should be capable of;
- Section 3.8, page 72, paragraph 2: *Publishing documetns* should be Publishing documents;
- Section 3.8, page 73, paragraph 3: *A system that were able* should be A system that is able;
- Acknowledgements: *the authors thank* should be The authors thank;

Chapter 4

Story Graphs Extraction and Visualization

Ilija Subašić and Bettina Berendt: Discovery of interactive graphs for understanding and searching time-indexed corpora. Knowl. Inf. Syst. 23(3): 293-319 (2010)

Contributions as first author:

- (a) Co-defining the research problems;
- (b) Parts of the related work overview;
- (c) Implementation of the STORIES method and the tool;
- (d) Conducting the case study and user studies;
- (e) Co-interpreting the results.

4.1 Abstract

Rich information spaces (like the Web or scientific publications) are full of "stories": sets of statements that evolve over time, manifested as, for example, collections of news articles reporting events that relate to an evolving crime investigation, sets of news articles and blog posts accompanying the development of a political election campaign, or sequences of scientific papers on a topic. In this paper, we formulate the problem of discovering such stories as Evolutionary Theme Pattern Discovery, Summary and Exploration (ETP3). We propose a method and a visualisation tool for solving ETP3 by understanding, searching and interacting with such stories and their underlying documents. In contrast to existing approaches, our method concentrates on *relational* information and on *local* patterns rather than on the occurrence of individual concepts and global models. In addition, it relies on interactive graphs rather than natural language as the abstracted story representations. Furthermore, we present an evaluation framework. Two real-life case studies are used to illustrate and evaluate the method and tool.

4.2 Introduction

The Web has led to a proliferation of news (and other broadcast media like blogs) that continuously report on current events and other topics. Several search-engine innovations of the past few years like the grouping of news articles by topic in Google News have made it easier to keep abreast when one reads the news every day. However, a Web user who misses several days or who wants to gain an overview of major events and developments in a "story" that lies in the past, is today faced with a situation that is reminiscent of the early days of the Web. Search in most archives is based on keyword search and therefore returns an unmanageable number of results. Summarisation like that provided by Google Trends[1] or BlogPulse's Trend Search[2] shows surges in publication and query activity in certain time periods, but these tools require one to know which sub-topic to look for (and how to describe it in keywords).

The same problem arises in other areas with high publication intensity and readers who aim to gain, refresh, and/or extend overviews of topical developments – scholarly publications are a prime example. Regardless of the domain, users have two main goals when dealing with such corpora: story understanding and story search. The goal of *story understanding* is to

[1] `http://www.google.com/trends`
[2] `http://www.blogpulse.com/trend`

comprehend the story's events, facts or other temporal subtopics and to track their evolution. In order to achieve this, users will want to inspect the story as well as the underlying documents (*story search*). Here, finding the most relevant documents is only a means to the (generally more important) end of discovering the events and their evolution and comprehending the general the story development.

This situation calls for systems that (a) identify topical sub-structure in a set (generally, a time-indexed stream) of documents constrained by being about a common topic, (b) show how these substructures emerge, change, and disappear (and maybe re-appear) over time, and (c) give users intuitive interfaces for interactively exploring the topic landscape and at the same time the underlying documents. In an extension of [35], we call the resulting problem *evolutionary theme pattern discovery, summary and exploration* (ETP3). In the past years, a number of powerful methods for solving subsets of these three requirements have been proposed. However, systems that address all three challenges are still lacking. In particular, we argue that semi-automaticity is the main element of such a system. The user (human intelligence) should not be exposed to well-formatted, predefined and global patterns from a machine intelligence system, but should be an integral part of information processing. Following this idea, we have built an interactive semi-automatic visual tool that provides users with local patterns and enables them to deepen their understanding of the story by searching for the content from which these local patterns are extracted.

Consequently, the first contribution of the paper is a (re-)appraisal of the ETP3 problem as one that requires a semi-automatic solution, and a proposal for a system that offers such a semi-automatic solution. Specifically, we believe that such a system should not be overly prescriptive. In particular, the user's interpretation of subdivisions within a topic will depend on her current tasks and other situational variables. We therefore aim, in contrast to the existing approaches, not at a global model of the topic (such as a clustering into exhaustive sub-topics); instead, we are interested in high-resolution local patterns and interaction options that support users in finding and exploring their own interpretations. Specific attention will be paid to supporting understanding via a constant interplay of abstracted representations of the story and the original representations: the underlying documents. The second contribution is an evaluation framework for ETP3 and a demonstration using two case studies.

The paper is structured as follows: In Section 4.3, we give an overview of related research. Sections 4.4 and 4.5 present our solution approach "STORIES": Section 4.4 describes the computational method and Section 4.5 the tool. Two case studies demonstrate method and tool in Section 4.6. Section 4.7 describes the evaluation method and results. Section 4.8 concludes with an outlook.

Parts of this substantially revised and extended paper were presented at the International Conference on Data Mining [44].

4.3 Related work

Our work builds on several areas of research, in particular the identification and tracking of topics in text streams, the identification of "bursty" events, the use of co-occurrence information for content extraction, query expansion, Web search result clustering, Web information extraction and information visualisation.

Temporal text mining. Mei and Zhai [35] described evolutionary theme pattern discovery as one key subproblem of temporal text mining. They presented a fully automatic method that extracts subtopics and creates a graph that shows their lifecycles and dependencies on each other. A mixture model was used to model documents as expressing (potentially several) themes (corresponding to sub-topics). These word clusters are tracked over time with Kullback-Leibler divergence measuring similarity, and the lifecycle of themes as well as cross-theme transformations are modelled as a Hidden Markov Model. The use of clustering models for finding emergent sub-topics and tracking them over time is also the subject of [42, 27]. In [49], LDA with an added time variable is used for the same purpose. Systems that do not rely on NLP (natural-language processing) summarisation can produce comparable results to those using NLP, as shown by [12] in their work on tracking and summarising events based on matrix decomposition for discovering (inter)event dependencies based on similarity. This approach allows for the discovery and understanding of the "anatomy" of a story in a more fine-grained manner. Kim and Lee [28] track and evolve a topic hierarchy over time, an approach that could also be modified to deal with story-related topic hierarchies.

Evolutionary theme pattern discovery is related to topic detection and tracking, specifically first story detection [3]. However, it is more fine-grained than TDT since it delves into a topic's substructure, and its aim is not only to classify something as a new (or old) topic, but to describe it. Within the TDT framework but by extending its standard task structure, a finer level of granularity can be achieved by the technique presented in [36]: "Event threads" are dependencies of sub-events of a topic. Event threads are created based on similarity of the terms belonging to the event (cluster) as well as named entities (persons and locations). Although the framework of event threading is similar to temporal text mining, it does not take time into account. Evolutionary

theme pattern discovery is also related to the document update problem in text summarisation, which is discussed in more detail in Section 4.7.1.

All these methods rely on the notion of sub-topics that cover the space of reported content, such that it is difficult to identify local details and their changes over time.

Burstiness. (Sub)topics may be particularly interesting when they are *bursty* [29], i.e. when publication activity on them is very strong in a certain time period, picking up volume fast at this period's beginning and (usually) disappearing again as fast. Burstiness has been explored with respect to various domains and phenomena including "buzz" in text and news streams [21, 22, 23]. Fung, Yu and Lu [21] group "bursty features" into "bursty events" based on co-occurrence, thereby creating an analogue of sub-topics.

So far, burstiness has only been investigated as a characteristic of single text features (words or topics). We extend this to an analysis of burstiness of associations.

Co-occurrence analysis. The analysis of bursty events points to the merits of focussing on specific parts of contents and their relations with each other, rather than on finding a global model. In general, the analysis of co-occurrences allows for a more fine-grained analysis of texts and has been investigated for example in text summarisation. Biryukov, Angheluta and Moens [6] show that *topic signatures* [33] provide a simple and effective way to summarise multiple documents. Smith [43] used co-occurrences to find historical associations between places and times in a digital library. He analysed how various interestingness measures rank these associations and showed that they behave differently, for example in the ranking of rare events. This indicates that different interestingness measures may be more or less adequate for the analysis of different corpora, domains and/or different tasks, an interpretation also supported by the findings of Feldman et al. [19]. These authors found co-occurrence lift to be an adequate interestingness measure to analyse perceptions of (car) brands and markets in user forums.

Choudhary et al. [13] propose application-domain interpretations of temporal changes in the frequencies of co-occurrences. They argue that agents (person names in the texts) can exist independently of each other, join, split again, etc. These developments create specific "story lines".

All these approaches are restricted to analysing co-occurrences between typed elements (names, places, ...). We take a more general approach and identify "story lines" between arbitrary words or concepts.

Allan, Gupta and Khandelwal [2] applied text summarisation to news streams, their focus was however more on finding the best sentences to be (re-)used in the summaries than on distilling concepts from these sentences. In contrast to this work, we focus not only on content that is new (i.e., different from what was reported before), but on content that is characteristic for a time period (i.e., also different from what was reported later).

Automatic query expansion (AQE). AQE is a set of techniques that expand the initial search query with semantically similar terms. Generally, query expansion methods can be global or local methods [56]. Global methods try to expand queries by analysing the complete corpus. Local methods are based on pseudo-relevance feedback and try to expand the query by analysing the relevance of a subset of top-ranked documents. In [57], local context analysis is proposed that combines both general approaches; in [15], query logs analysis is used to expand queries. Possible expansion terms may be ranked by their information entropy, and the most promising ones selected [48]. For specific domains, the general approaches can be improved upon by domain-specific methods, as shown for blogs or news in [17].

AQE tries to find more relevant results for an initial query with the goal of higher effectiveness, rather than structuring the results or involving the user. In contrast, we look at the situation where the relevance of a result set does not change and users are involved in query expansion in order to discover and/or explore a certain area of a result set in which they have an interest. The AQE approach most similar to the task investigated in the current paper is presented in [20]. By applying association rules to concepts (previous queries), the authors create a concept relation graph in which each subgraph is a more general concept relating to one of the user's interests. In [7], the problem of topical query decomposition was introduced. The task is to start from a query and then find a set of queries whose result sets cover the same documents as the initial query. The authors proposed two solutions: a top-down approach based on set coverage and a bottom up approach that utilises hierarchical clustering. Our objective is related to this but differs in that we focus more on user-defined topical substructures and less on a complete decomposition.

Web search result clustering aims at learning and presenting structure within a query result set by applying clustering algorithms (see [24] for a classical approach). In contrast to traditional document clustering, overlapping clusters may be more adequate, for example in the form of Suffix Tree Clustering [58]. By transforming the clustering problem to a ranking problem, [59] try to find the most meaningful clusters. They rank by the TF.IDF weight of a term, its length, the intra-cluster similarity of documents and cluster entropy.

In the search and exploration of stories, one must take into account a time dimension as well as more fine-grained local relationships between the concepts, which allow users to both search for the content and understand the facts and events surrounding a story. A semi-automatic solution to this problem using facet search was proposed in [34]. A facet is defined as a semantically coherent model over words pointing to one part of a corpus. The multifaceted overview mining task is then to provide the user with an overview of word-topic distributions to which given query terms correspond best, starting from a small number of initial query terms. The authors propose a probabilistic mixture language model to solve the problem. However, the resulting facets are fully determined given the initial query terms, while we aim at a more flexible story exploration, in which users can explore arbitrary subparts of a story.

Web information extraction. IE aims at automatically finding facts and relations in texts. Many IE approaches rely on expert-made rules or full-fledged linguistic parsers, which does not scale to a Web environment. A well-known working system that overcomes these constraints is KnowItAll [18]. A more recent system that improves on efficiency limitations encountered by KnowItAll is TextRunner [4]. It uses a self-supervised learner on a subset of its base corpus to create possible facts, and it then runs a single pass run through the entire corpus to retrieve relations. SRES [41] focuses on extracting relations for which the Web corpus has limited redundancy. It utilises a more expressive extraction pattern language than KnowItAll, which enables it to extract information from a broader set of sentences. Although these systems have good results in fact retrieval, they discard the time dimension and the internal structure of corpora. They allow the user to search for a specific relation, but not to gradually discover the events, relations and elements of a story.

Visualisation. The main focus of most of the above studies were challenges (a) and (b) mentioned in the Introduction. Visualisations are probably best suited to displaying the complex relationships found. Smith [43] provided users with an interactive map browser for exploring the location-time co-occurrences. This is a good example of how to meet challenge (c) in a way that is adapted to the application domain. Wong et al. [55] show a domain-independent way of visualising pairwise associations of words that also takes the strength of these associations into account. They plot words against time and show co-occurrences by connecting lines in a format that is related to parallel coordinates. Their graphs provide an excellent overview of the occurrence or recurrence of pairwise associations over a whole timeline. However, because time takes up one visual dimension, higher-order patterns of associations cannot easily be detected. In contrast to this, we will show

associations per time point/period. This "snapshot" idea is the same as that used in the graph sequences used for visualising scientific publications and topics in, e.g., [10, 11, 27]. In contrast to that, we use a layout strategy that is more amenable to highlighting emerging and disappearing topics, and offer the alternative of a dynamic layout between successive time periods (morphing). The morphing visualisation is similar to that in [31].

Usually, the results of a Web search are presented as a ranked list of documents based on relevance to a specified query. The use of graphical representation of search results has been well studied, see [30] for a recent overview. Visualisations are usually displayed either as relations between single documents or as relations between structures such as predefined categories, learned clusters or freely annotated tags. Systems like KartOO[3] create visualisations that represent documents inside a cluster space. The common idea is to present some predefined structure such as global clusters or local relations among documents. Zoetrope [1] presents an interactive interface that allows users to track single DOM elements of an HTML page over time. By interacting with the Web at this atomic level, users can track and discover parts of a story without any linguistic processing. However, in ETP3, users are less interested in a single document or its relations to the others than in learning about the underlying story. Our visualisation presents users both with an interactive search and an interface which allows for fact discovery.

4.4 The STORIES method

4.4.1 The method for story understanding

The basic assumptions of our method are that (a) there is a set of time-stamped documents that are all relevant to a top-level story and, when read by a human reader, reveal the story and its evolution and (b) the words in these documents also reveal the story and its evolution when processed by simple text mining methods. We conceptualise

- *story basics* as the high-ranking terms (words, compounds, named entities, concepts, ...) from all documents of a corpus of relevant documents, where the ranking reflects the importance of these terms in the corpus,

[3] www.kartoo.com

- *story elements* as the high-ranking relationships between story basics, where the ranking reflects the importance of these relationships in the corpus,
- *story stages* as networks of salient story elements in a certain time period, where salience is measured based on co-occurrence frequency and its relevance in a current time window and in the whole corpus,
- *story evolution* as the temporal sequence of story stages.

This basic scheme can be operationalised in several ways. To create a baseline, we have started with very simple versions of each of these constructs' operationalisations. Specifically, the method involves the following stages. First, a corpus of text-only documents is transformed into a sequence-of-terms representation. Subsequently, basic term statistics are calculated to identify candidates for story basics. We chose *term frequency TF* for the whole corpus, which is defined as *(# occurrences of the term in the whole corpus) / (# all terms in the whole corpus)*. We define the *content-bearing terms* as the 150 top-TF terms.

Next, the whole corpus C is partitioned into sets of documents that were published in time periods following one another, e.g. within one calendar week. Thus, C is the union of all document sets c_i, with $i = 1, \ldots, I$ the time periods.

For each c_i, the *frequency* of the co-occurrence of all pairs of content-bearing terms b_j within a window of w terms in documents is calculated as follows:[4]

$$freq_i(b_1, b_2) = \frac{\#\ occu.\ of\ both\ b_1, b_2\ within\ w\ terms\ in\ doc.s\ from\ c_i}{\#\ all\ doc.s\ in\ c_i}. \quad (4.1)$$

This measure of frequency and therefore relevance is normalised by its counterpart in the whole corpus to yield the measure *time relevance*:

$$TR_i(b_1, b_2) = \frac{freq_i(b_1, b_2)}{freq_C(b_1, b_2)}. \quad (4.2)$$

This measure is based on the *domain relevance* metric [37], which measures the relevance of a term in a (subject-domain) subcorpus relative to the whole corpus. When used, as here, for time-specific subcorpora, it also measures "burstiness". Thresholds are applied to avoid singular associations in small subcorpora and to concentrate on those associations that are most characteristic

[4]This measure takes into account multiple co-occurrences within one document, in contrast to the *support* measure which uses the number of documents containing the co-occurrence as numerator. Prior tests showed that support did not find out salient co-occurrences well enough.

of the period and most distinctive relative to others . We define two sets *Non-singular* and *Characteristic*:

$$\textit{Non-singular}_i = \{(b_1, b_2) \mid (\#\ \textit{co-occurrences of}\ b_1, b_2 \\ \textit{within}\ w\ \textit{terms in articles from}\ c_i) \geq \theta_1\} \tag{4.3}$$

$$\textit{Characteristic}_i = \{(b_1, b_2) \mid TR_i(b_1, b_2) \geq \theta_2\} \tag{4.4}$$

for some thresholds $\theta_1 \in N$ and $\theta_2 \in R$, with $\theta_2 > 1$ to select co-occurrences that are bursty in i. This gives rise to

- the *story stage* i: $\textit{Non-singular}_i \cap \textit{Characteristic}_i$. This creates a *story graph* with terms as nodes and associations as edges.
- the *story elements*: all edges of the story stage.
- the *story basics*: all nodes of the story stage.
- the *story evolution*: the sequence of story stages.

To obtain a smoother story evolution, we use the moving average of co-occurrence frequency values. This was done by replacing for each period c_i, the document base set in both numerator and denominator of the right-hand side of the *freq* definition by the union over periods $i, \ldots, (i + l - 1)$, for a time window size $l \in N$.

Investigations of different parameter settings showed that in most cases, only associations with $TR > \theta_2 = 3$ are interesting and allow for a tractable graph. However, the advantage of an interactive approach is that we can let the user explore different values of θ_2 and thereby create their individual story stages. Visualisation options (see Section 4.5) help to accentuate the differences in time relevance. Users are also able to control θ_1.

4.4.2 The method for story search

The STORIES approach to search relies on a semi-automatic interaction between users, the story graphs, and the underlying documents. Based on the automatically created story graphs (see previous section), users choose edges and compile subgraphs for searching (see Section 4.5.3). The search component then automatically generates the document subset relevant to this user selection.

Let c_i be the full set of documents for a time period i in which a user wishes to search for and learn more about a specific event or fact. The story elements, which are contained in the story graph for i, are then the key information elements used for describing this event or fact. Story elements in turn represent associations between story basics. We assume that the user's interest is directed towards a sub-corpus of documents characterised by nodes (story basics) which are connected by the chosen edge(s). Users can be interested in more than a single edge, trying to semantically relate more edges. So by interacting with a story graph G_i, users create a search restriction R of story basics. This restriction can be created using a single edge or any possible connected sub-graph (tree or a path) R of a story graph for a time period. STORIES then uses all the nodes n as a query (restriction) for the documents inside c_i to obtain the pertinent document subset:

$$c_i^R = \{d \in c_i \mid \forall n \in R : n \text{ is a term in } d\} \tag{4.5}$$

This set c_i^R consists of only those documents that contain all the nodes of R. This binary IR model was used in order to simplify the approach, and it could be replaced by any IR model that employs R as query and c_i as corpus.

4.5 The STORIES tool

We applied the method to news articles downloaded from different sources on the Web, as indexed by Google News. In this section, we describe the data cleaning and further pre-processing applied to this kind of data.

4.5.1 Data cleaning

Data cleaning represented a challenging first step in data preparation. Virtually all news sources present their content in Web pages with a multitude of other content: navigation menus, advertising, ... We therefore included an automated wrapper-induction component in the tool, following state-of-the-art approaches such as [16]. For the first case study described below, we extracted the content into ASCII by manual copy-and-paste. After informal tests showed that the automatic content extraction produced highly similar results, we decided to rely on it for the second case study.

4.5.2 Text pre-processing

The documents were first tokenised; subsequently, several further pre-processing options were investigated. Named entity recognition (NER) was done as a two-phase process. In the first phase, the Open Calais[5] semantic toolkit was used to extract NEs. Pilot tests showed that pronoun resolution did not work well on our materials; therefore pronoun resolution was filtered out using a stopword list. Since Open Calais operates on a per-document basis, it cannot map a term to named entities if the named entity does not appear in the document that is currently inspected, despite the fact that in the entire corpus the same term is mapped to the same named entity. To overcome this problem, in the second phase, each term x that was mapped to some named entity in at least one document in the first phase, was treated as follows: Let $x_1, ..., x_n$ be the NEs to which x was mapped in the first phase. Let x_{max} be the NE from $x_1, ..., x_n$ to which x was mapped most often. Then, in each document containing x but not x_{max}, we map x to x_{max}. (A similar NER solution was proposed by [14].) This was followed by lemmatization using the TreeTagger.[6] Stopwords were removed using the stopword list from the Terrier project,[7] manually enhanced by HTML-specific and application-specific words.

All parameters for text pre-processing can easily be configured, and the architecture provides the needed modularity for, e.g., using different interestingness measures and thereby re-using and/or evaluating other proposals for temporal text mining.

4.5.3 The graphical usage interface

We implemented the method and generated visualisations using GUESS.[8] The visualisations comprise static visualisations of the story stages of individual periods, and a morphing sequence that traces story evolution through the sequence of all periods. In addition to this "scanning", users can "(un)zoom" by adapting the period-window size l.

The visualisations are enhanced by salience slide rulers that allow the user to filter out story elements below individually set θ_1 (absolute number of occurrences of an association) or θ_2 (time relevance) thresholds.

A configurable colour scheme accentuates time relevance differences. For on-screen viewing, different users expressed preferences for sequential schemes

[5] http://www.opencalais.com
[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger
[7] http://ir.dcs.gla.ac.uk/terrier
[8] http://graphexploration.cond.org

using other colours or for divergent schemes, in particular the harmonious colours from blue (high *TR*) via red (medium) to yellow (low), cf. [50]. A screenshot is shown in Fig. 4.4. In the remaining story-graph figures 4.3–4.6, the graphs have been extracted from the tool environment for better legibility. Different figures illustrate different colour-scheme options.

In addition, the GUI supports document search: By clicking on a single edge, the user can select documents that contain the associated term pair. For easier and more flexible search, users may also select an edge and then highlight a subgraph which contains the selection's adjoining edges and neighbouring nodes. Each selected edge expands the query by adding its nodes to the query 'shopping cart', as long as the query has fewer than six distinct terms. This restriction was chosen because users rarely specify queries longer than six words [40]. At any time, the user can choose to see the complete graph with all subgraphs. In this way the user incrementally builds the query and at the same time can discover and learn about the story. Figure 4.7 shows a screenshot of the search functionality.

All programs can be executed on a local computer, after an initial indexing of documents.

4.6 Case studies

For demonstration, we used two real-life stories with comparatively clear and well-known courses of events. The first story (S_1) was the disappearance of Madeleine McCann on May 3rd, 2007 and the development of the criminal investigation. For the second case study (S_2), we followed the events surrounding Britney Spears in January and February 2007, chosen because media-gossip intensity surrounding the pop singer was particularly high during this time.[9]

Two main events of the first story were the early suspicion of a man with the

[9]We wish to emphasise that in no way do we want to capitalise on the sad story of a missing child. However, in the present case, media attention was specifically asked for, at least in the beginning: The child's parents established an unprecedented media campaign to ensure that any hints that anyone might have would be reported. On the first anniversary of the disappearance, the family used the Web site to ask for an end of media attention. The intensity of reporting on celebrity "meltdowns" may be considered similarly disquieting. It is unfortunate that personal and public catastrophes seem to lend themselves most easily to automated story analysis; witness for example the 2005 London bombings (e.g., [45, 39]) or the 2004 Tsunami [35].

initials R.M.[10] as kidnapper, the discovery of the child's blood in a car rented by the parents (established as hers by a DNA test) and the associated police questioning and suspicion of her parents. These were interspersed by long periods of less media attention with little to report (or misleading incidents like the arrest of two people unrelated to the case).[11]

Main events of the second story include Britney Spear's surprising new hair style, a fallout with her ex-husband, and her substance addiction problems and clinical treatment.

The corpora. We used articles from the Google News archive.[12] For S_1, data was collected for the period between May and December 2007 (week 17 in which the girl disappeared until week 52). The data were then filtered as follows: only English-language articles; for each month, the first 100 hits, and of those, only those that were still freely available in April 2008. After a first round of analysis, these were restricted to documents from weeks 17–37, the "eventful" weeks of that story. This resulted in a corpus of 215 documents.

This set was extended by the set of all retrievable, English-language news articles referenced in the Wikipedia article [54], from the investigated time period. This provided another 91 articles. This selection constitutes a kind of opposite extreme of the first document selection, because the occurrence of an article in the reference list indicates that its content passed a manual quality control and was integrated into the Wikipedia article. Due to the collaborative authoring of the Wikipedia article, this selection can also be said to represent a wide variety of viewpoints and (potentially) consensus on the quality of the individual articles.

The combined corpus contained 306 articles with 174,886 words. The corpus contained 8,075 (6,089) unique words (lemmas).

The data for S_2 consists of 3000 articles from January and February 2007. The restrictions were the same as for S_1 except for the number of hits, which was set to 400 for each week. Some weeks had fewer than 400 hits, which explains the total of 3000. Since there was no specific Wikipedia entry that describes the events for S_2, the corpus was not extended by such references.

The S_2 corpus contained 1,679,894 words, resulting in an average of 560 words per article. The corpus contained 58,509 (41,558) unique words (lemmas).

[10] In the text and figures of this article, we have anonymised all person names except that of the missing girl, which we need to report to identify our data, and consequently also her family name.

[11] All three suspects were cleared later; and the case was closed in July 2008, see [53].

[12] `http://news.google.com/archivesearch`

The two different types of stories with different numbers of articles and a different total time-span were chosen in order to test the robustness of our approach.

This was regarded as a good approximation of the real-life situation confronted by a deployed STORIES algorithm: Articles are found to be candidates based solely on keyword matching (in this case: using the first and last name of the key person as the query in the Google News archive), they come from sources of varying quality, and there is no ranking on the news sources in the Google News archive after some months.

Results Figures 4.3–4.6 show selected individual story stages of S_1 and S_2.[13] In particular, Fig. 4.3 shows the *description of an event* (missing British child MM) in S_1. Figure 4.4 illustrates how the key first suspect becomes an (also visually) *"central" element* of the story in S_1, and how the celebrity is always at the "centre" of her own story in S_2. Figures 4.6 (a) and (b) show how the interface is used by changing the threshold in order to *"uncover" a story stage*. Specifically, Fig. 4.6 (b) explains some of the reasons for the connections in Fig. 4.6 (a). An *eventless period* in a story is characterised by a small number of disconnected subgraphs like the ones in Fig. 4.5.

4.7 Evaluation

Temporal text mining is still a young area, so unlike for example in TDT, no standards exist yet for evaluating approaches, and the existing literature often restricts itself to plausibility checks. Therefore, the quest for an evaluation of the STORIES approach involves finding answers to the following questions:

(1) Can existing evaluation frameworks and/or datasets be used as benchmark?
(2) How should the ground truth be defined?
(3) Can evaluation be (partially) automated to reduce human evaluators' workload?
(4) What instructions should human (or machine) evaluators get?
(5) How can the results be interpreted?

We address questions (1)–(5) in turn, once for each component of STORIES. The ETP3 problem can be decomposed into two subproblems: Evolutionary theme pattern discovery and summary on the one hand, and evolutionary theme pattern exploration on the other hand. Evolutionary theme pattern discovery and summary is the core part of the tool's story understanding

[13] Preprocessing with $w = 5, l = 2$. Visualisations of the corpus with θ_2 adjusted for maximum visibility and $\theta_1 = 5$ throughout.

component and will be addressed in Section 4.7.1. Evolutionary theme pattern exploration consists of interactions with the story graphs and with the underlying documents. This will be addressed in Section 4.7.2.

4.7.1 Evaluation of the story understanding component

(1) Existing evaluation frameworks

Evolutionary theme pattern discovery and summary is related to the *update task* first formulated in the Document Understanding Conference (DUC) 2007: "The update summary pilot task will be to create short (100-word) multi-document summaries under the assumption that the reader has already read a number of previous documents."[14] This contest supplied a test corpus of news stories (documents assigned to 10 topics, each divided into 3 time periods, were supplied), summaries of the updates manually generated by 4 independent human raters, and detailed evaluation reports (precision, recall and F1) of baselines and all the contenders. The evaluation reports were generated with the ROUGE software ("Recall-oriented understudy of gisting evaluation"). ROUGE has also been used to evaluate multi-faceted overview mining [34] and sentence-based multi-document summarisation [52].

The DUC/ROUGE concept is not directly applicable to STORIES because it assumes that the summaries are natural-language texts, whereas we generate graphs. Yet, we created a way of applying the ROUGE evaluation framework and software to our representation (see (3) below).

However, the DUC/ROUGE dataset cannot be used to benchmark our approach. The reason is that the dataset is not a stream (a large set of documents following in quick succession and with usually relatively small differences to the previous one). Rather, it is (for each topic) a set of 3 small-cardinality (usually below 10) sets of documents that were published in 3 disjoint and subsequent time periods, but have very little connection to the other 2 periods' content. This resulted in *all* interesting co-occurrences being "bursty" in each of the 3 periods, making it impossible to select the really important ones.

Another candidate dataset is the Tsunami dataset used and provided by [35].[15] However, since it is not associated with a ground truth and since it is not straightforward to compare the output of STORIES with the output of the method of [35], this dataset could not be used either.

[14] `http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot`
[15] `http://sifaka.cs.uiuc.edu/~qmei2/data.html`

We therefore decided to use our own case-study datasets and to concentrate on defining a method for evaluating evolutionary theme pattern discovery and summary.

(2) Finding a ground truth

One of the biggest problems of finding a ground truth for story understanding is that in many text tasks, the agreement between human raters is not very high. Thus, it is necessary to have a ground truth that reflects a wide range of human raters. In some evaluation frameworks, this goal is achieved by employing several ground truths by different people (e.g., 4 in the DUC/ROUGE evaluation, see (1) above).

Fortunately, the Web itself provides us not only with streams of news, but also with documents that come close to the goal of a multi-rater truth. These documents contain a description of the "timeline" concerning some story, i.e. a chronological account of what happened. We chose two different types of timeline (ground truth) sources. The first is Wikipedia as a source written and revised by hundreds of authors, cf. for example [8]. The specific source for S_1 was [54]. The second is an editor-created timeline, in this case from the HollyScoop site which tracks the news on celebrities. The specific source for S_2 was [25].

Timeline articles are often very long, full of detail, and only occasionally written with a story progression in mind. Therefore, such a document must be transformed in order to serve as a ground truth to be used in a (machine or human) evaluation. We proceeded as follows: First, all sentences that contained a date were extracted from the article. To minimise bias and errors, we had two independent raters extract these sentences (and if necessary perform minor reformulations to make them understandable out of context). Only those assertions that both found in the text, plus a maximum of five others from each rater, were included in the final set of ground truth assertions. In the case studies, this resulted in a total of 31 ground-truth *events* for S_1 and 12 events for S_2.

All ground-truth events were indexed by the calendar week (S_1) or day (S_2) in which they had occurred, such that they could later be assembled easily into the ground-truth of the time window (e.g., 3 weeks) that was covered by the method.

(3) Partially automating the evaluation

The goal was to present both the STORIES output and the ground truth to human evaluators in order to determine precision and recall. However, in a

pilot study with human raters, we had found that this was a very laborious task and could not easily be repeated for different settings because after seeing the first setting, a human rater knows the story.

Therefore, prior to presenting people with the ground truth and the algorithm output, the best parameter setting had to be found. Towards this end, we employed the ROUGE software, which was kindly provided to us by its creator Chin-Yew Lin. Recall from Section 4.4 that the method has as parameters l (the number of time periods that make up a story stage, where story stages are overlapping when $l > 1$), w (the window size within the texts that is inspected for co-occurrences), θ_1 (the minimum total number of co-occurrences), and θ_2 (the minimum time relevance of a co-occurrence). The pilot study had also suggested that for humans to be able to read the graph, the *cardinality of the story stage* (the number of edges of each graph, $|StoryStage|$) should be limited.

For S_1, we choose 1 week as the interval between successive time periods and varied $l = 1, 2, 3$ week(s), and for S_2 we chose 3 days as the interval and varied $l = 3, 5, 10$ days. For both case studies we kept $w = 5$ and $\theta_1 = 5$ based on common values found in the literature, and, starting from a value of $\theta_2 = 2$, varied $|StoryStage|$ from 10 to 30, in increments of 5. $\theta_2 = 2$ was an intuitive value based on the pilot study ("at least twice as frequent in this period than on average"), and 10 to 30 was considered to be a realistic range for human graph reading usability [51].

To evaluate this large number of combinations, we used the findings of [32], who showed that the automated word-pair matching rules of ROUGE correspond to human ratings in the following sense: The *ranking of quality assessments* by humans corresponds to the ranking of quality assessments by ROUGE. This does *not* necessarily mean that the *absolute values* of precision or recall correspond to each other. However, it means that the setting with the best ROUGE results should be chosen for presentation to humans.

ROUGE evaluates natural-language texts against other texts (the ground truth). STORIES outputs graphs instead of natural-language texts. These two forms of representation are not directly comparable (see for example [26]); however, pilot tests showed us that people interpret paths in the STORIES graphs in a similar way as sentences. We therefore used the following heuristic: We extracted all paths from each STORIES graph[16] and ordered them by descending average TR path weight[17]. We then truncated these "pseudo-sentences" at 100 characters to generate ROUGE-style summaries.

ROUGE has different evaluation functions. In our case, the applicable ones

[16] using an adapted version of the Gaston software [38]

[17] the average, over all edges in the path, of these edges' TR values

were ROUGE-1 (overlap of unigrams), which however always favoured larger graphs – a result that is in conflict with the usability requirement of smaller graphs. The only other applicable function is ROUGE-SU4, which measures the overlap of skip-bigrams of at most length 4. A skip-bigram is any pair of words in their sentence order, allowing for arbitrary gaps. Skip-bigram co-occurrence statistics measure the overlap of skip-bigrams between an automatically generated text and a ground-truth text. We used the ROUGE parameter values that were employed in the DUC 2007 update task evaluation.

The resulting ranking for S_1 was (pairs denote $|StoryStage|$ and l): $20-2, 30-3, 25-2, 15-3, 20-3, 25-3, 10-3, 15-2, 30-2, 10-2, 10-1, 25-1, 30-1, 15-1, 20-1$. The ranking for S_2 was $25-5, 20-3, 25-3, 15-3, 20-5, 10-3, 20-10, 30-3, 15-5, 30-5, 10-10, 25-10, 30-10, 15-10$.

Thus, 20 edges and a window of 2 weeks produced the best story stage descriptions for S_1; 25 edges and a window of 5 days were the optimal values for S_2.

(4) Procedure Two raters from different backgrounds, both with a good command of English and only a superficial knowledge of the story, volunteered to rate the STORIES summaries for S_1; three other raters with analogous characteristics volunteered for S_2. They were given the 20-2 (S_1) and 25-5 (S_2) graphs for the case-study corpora described in Section 4.6.

The raters received the GUESS software (without search functionality) and the graphs together with a driver script, an Excel sheet with one tab for each time period and one ground-truth event per line, and a set of instructions. They then worked individually at their own pace. The raters were asked to inspect the graphs in temporal order using the slide ruler for θ_2, starting from a high value so as to "uncover" the graph, and to rate the first 20 (S_1) respectively 25 (S_2) edges as follows: If it describes an aspect of an event, then annotate the event with the edge number (visualised by a change in the script). If multiple matches seem appropriate, multiple annotations should be made. The raters were asked to stop when they reached 20 edges or before if the graph "stopped making sense".

The filled-in Excel sheets were the basis for the following quantitative evaluation.

(5) Results and interpretation The measured outcomes for story understanding were precision at $n = 5$, 10, 15, 20 (and 25 for S_2) (since edges were numbered, the top-TR edges could easily be identified) and recall at the same

n, the latter defined as the number of correctly retrieved events for the top-n edges. This notion of recall differed slightly from the standard one because the number of events differed between periods (such that 1 mapping edge would give rise to a recall of 0.2 in a period with 5 events, but 0.5 in a period with 2 events). Some graphs contained fewer than n edges, for these, precision and recall at n are not defined.

Per.	$p20_U$	$p20_M$	$p15_U$	$p15_M$	$p10_U$	$p10_M$	$p5_U$	$p5_M$	$r20_U$	$r20_M$	$r15_U$	$r15_M$	$r10_U$	$r10_M$	$r5_U$	$r5_M$
17	.55	.40	.47	.47	.50	.40	.80	.40	.80	1	.80	1	.60	1	.60	.40
18	.40	.25	.33	.27	.40	.30	.40	.60	.80	.40	.60	.40	.60	.40	.40	.40
19	.55	.20	.47	.20	.50	.30	.20	.40	1	1	.67	.67	.67	.67	.67	.67
20	.20	.30	.20	.33	.10	.30	.20	.20	.33	1	.33	1	.33	.67	.33	.67
21	.15	.25	.20	.27	.30	.40	.20	.60	.40	.60	.40	.60	.40	.60	.20	.60
22					.10	.30	.20	.40					.67	.67	.67	.33
23					0	0	0	0					0	0	0	0
24			.27	0	.30	0	.60	0			1	0	1	0	1	0
26					.30	.10	.40	.20					1	1	1	1
27					.25	.10	.20	.20					1	1	1	1
29	.05	.10	.07	.13	0	.20	0	0	1	1	1	1	0	1	0	0
30	.20	.25	.13	.27	.20	.30	0	.20	1	1	1	1	1	1	0	.50
31	.29	0	.20	0	0	0	0	0	1	0	1	0	0	0	0	0
32					.29	0	.2	0					1	0	1	0
34	.35	.35	.40	.47	.40	.60	.60	.80	1	1	1	1	.60	1	.60	.60
35	.30	.60	.33	.53	.30	.70	.20	.60	.75	1	.75	1	.75	1	.13	.75
36	.67	.67	.47	.47	.40	.40	.20	.20	.83	.83	.67	.67	.50	.50	.17	.17
37	.15	.15	.07	.07	.10	.10	.20	.20	1	1	1	1	1	1	1	1
Avg.	.32	.29	.28	.27	.25	.25	.26	.28	.83	.82	.79	.72	.62	.64	.49	.45
S.D.	.19	.19	.15	.18	.16	.21	.23	.25	.24	.32	.24	.38	.36	.40	.40	.36

Table 4.1: Precision and recall for $n = 5, 10, 15, 20$ edges for raters U, M over the time periods in case study S_1 (eventless periods are excluded). Empty cells in a column "...n" indicate a period with $< n$ edges. The last two lines show averages and standard deviations over periods.

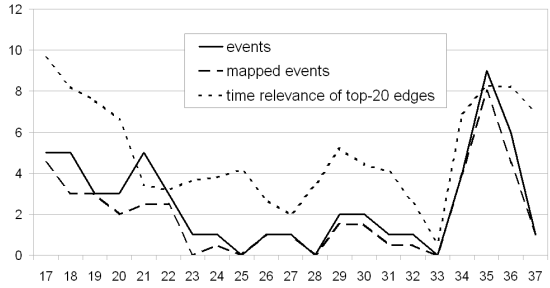


Figure 4.1: Case study S_1: events (ground-truth), edges and their burstiness profile (average TR), and represented events.

The results for S_1 are shown in Table 4.1. In the table, $p20$ denotes precision at 20, $r20$ denotes recall at precision 20, etc., as defined above. They illustrate that (a) the judgements by both raters were highly similar in terms of overall quality; (b) recall was quite high – on average, nearly half the events were found in a graph as small as 5 edges, and over 80% in a graph of 20 edges; (c) precision was uniformly acceptable but strictly lower than recall (about one third of all edges were content-bearing).

The relatively large values of the standard deviation indicate that the quality of representation varies by period. To investigate whether certain "ground-truth event patterns" cause these variations, we illustrate in Figure 4.1 more detail by plotting the number of events against the number of events that were represented in the graphs (for 20 edges, averaged over the two raters). It indicates that the number of events does *not* influence the quality of representation as measured by recall. The figure also plots the average TR values of the top 20 edges; they too follow the same pattern as the events. Thus, the burstiness measured by TR is a good measure of ground-truth "eventfulness".

Only in period 25 is there is a marked difference between a high average TR and a low number of events. The reason is that in week 26 (which affects 25 due to $l = 2$), a couple had been implicated and arrested. These soon turned out to be con artists who had nothing to do with the case. The incident is not reported in Wikipedia (which we used as "ground truth"), but made headlines at the time.

Per.	$p25_G$	$p25_B$	p25A	$p20_G$	$p20_B$	$p20_A$	$p15_G$	$p15_B$	$p15_A$	$p10_G$	$p10_B$	$p10_A$	$p5_G$	$p5_B$	$p5_A$
1										.33	.20	.13	.40	.17	.20
2	.17	.16	.08	.25	.15	.10	.13	.07	.13	.20	.10	.10	.20	.00	.00
3	.29	.24	.16	.31	.20	.15	.27	.13	.13	.40	.20	.20	.40	.00	.00
4	.80	.24	.24	.50	.30	.33	.33	.40	.25	.31	.60	.30	.21	.80	.40
5				.56	.35	.35	.60	.47	.40	.60	.50	.50	.60	.60	.40
6										.60	.20	.30	.60	.20	.40
7	.42	.17	.24	.45	.20	.30	.40	.20	.40	.40	.20	.40	.40	.20	.40
8	.32	.20	.12	.30	.20	.10	.50	.27	.13	.30	.10	.20	.40	.00	.20
9	.28	.16	.32	.35	.20	.35	.47	.27	.33	.40	.30	.30	.40	.20	.20
10	.28	.24	.19	.35	.30	.20	.47	.44	.27	.50	.40	.20	.60	.60	.20
Avg.	.37	.20	.19	.38	.24	.24	.40	.28	.26	.40	.28	.26	.42	.28	.24
S.D.	.21	.04	.08	.11	.07	.11	.15	.15	.12	.13	.17	.12	.15	.29	.16

Per.	$r25_G$	$r25_B$	$r25_A$	$r20_G$	$r20_B$	$r20_A$	$r15_G$	$r15_B$	$r15_A$	$r10_G$	$r10_B$	$r10_A$	$r5_G$	$r5_B$	$r5_A$
1										1	1	1	1	1	1
2	1	.50	1	1	.50	1	1	.50	.50	1	.50	.50	1	.00	1
3	1	.67	1	1	.33	.67	1	.33	.67	1	.33	.33	.33	.00	.00
4	1	1	.67	1	1	.67	1	1	.67	1	1	.67	1	1	.67
5	1	1	1	1	1	1	1	1	1	1	1	1	1	.50	1
6	1	.50	1	1	.50	1	1	.50	1	1	.50	1	1	.50	1
7	1	.50	1	1	.50	1	1	.50	1	1	.50	1	1	.50	1
8	1	1	.67	1	1	.67	1	1	.67	1	.67	.67	1	.00	.67
9	1	1	1	1	1	1	1	1	1	1	.67	.67	1	.67	.67
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Avg.	1	.80	.93	1	.76	.89	1	.76	.83	1	.72	.78	.93	.52	.80
S.D.	.00	.25	.15	.00	.29	.17	.00	.29	.20	.00	.26	.25	.21	.41	.32

Table 4.2: Precision and recall for $n = 5, 10, 15, 20, 25$ edges for raters A, B, G over the time periods in case study S_2. Empty cells in a column "...n" indicate a period with $< n$ edges. The last two lines show averages and standard deviations over periods.

The results for S_2 are shown in Table 4.2. Numerically, they are quite similar to the results for S_1. In addition, they show individual differences more strongly: It seems that rater G could make more sense of the graphs and found the story better represented in it than raters A and B.

4.7.2 Evaluation of the story search component

The previous study concerned the quality of the STORIES graphs as summaries of a corpus of news documents. However, human story understanding does not stop with inspections of these re-representations of the documents. The use of the graphs for describing interesting events or event structures in terms of subgraph selection, the retrieval of document sets associated with these events/event structures, and the inspection of these documents are further integral parts of tool use for in-depth human story comprehension. The objectives of an evaluation of the search functionality are therefore twofold: to find out whether subgraphs indeed index "focussed" sets of documents that describe events or event structures, and to find out whether this is helpful for human search.

To answer these two questions, we performed an automated analysis of the document sets created by subgraph selection and a user study that investigated whether the documents were helpful for understanding facts about the underlying real-world story.

The evaluation of story search faces some of the same problems as the evaluation of story understanding:

(1) Existing evaluation frameworks Applicable existing frameworks are Web search clustering and query expansion. However, neither of them really answers our questions. Existing evaluation approaches to Web search clustering focus on the quality of the groups (in a precision/recall framework, e.g. [58], or in terms of cluster quality, e.g. [59]), but in general not on the usefulness of the groups for human understanding. Nonetheless, internal coherence of document groups appears to be a necessary condition for good search; we therefore adopt this approach as part of our evaluation framework (see Section 4.7.2).

Query expansion evaluation relies on predefined document relevance to known topics. This kind of external validity criterion does not exist in our problem setting (all documents are already relevant to the main story, the question is how to separate this into subtopics in a flexible way).

(2) Finding a ground truth With regard to the underlying real-world story, the same problems and opportunities arise as with respect to story understanding in general. Therefore, we also adopt the same solution approach and re-use the ground truth as defined in Section 4.7.1. This ground truth also assumes some notion of atomic events.

However, the problem of substructure is more difficult to address: what are legitimate structures of these atomic events in a real-world story? This may be regarded as a problem of combinatorics, since (too) many combinations of atomic events are possible. Inter-rater disagreement is also likely to be higher for these questions; although we are not aware of any research on this.

For these reasons, we decided to not carry out a direct external evaluation of the validity of the substructures created in search. However, we performed an indirect evaluation by seeking answers to the question whether the substructuring allows users to discover ground-truth events (see Section 4.7.2).

(3) Partially automating the evaluation We continued to use the optimised parameters obtained by the automated method described in Section 4.7.1.

In the following, we describe two studies: The one on the internal validity of document groups produced by the search tool is fully automated; it is complemented by a user study that profits from the automated first steps in an analogous way as the user study described in Section 4.7.1 above.

Search: automated evaluation

Search helps to focus on event structures identified by subgraphs of a story graph. Each subgraph indexes a document set which is a subset of the whole document set from the graph's time period. Each of these subsets should ideally focus on some event structure in the real-world story; however as we have argued above, the nature of the events and the combinatorics of the subgraphs that may be created and selected by users preclude an objective "true structure"; thus, evaluation cannot use external validity criteria like precision and recall. Therefore, we evaluated the query result sets by internal validity criteria. We measured whether these document sets deal with common content by applying measures of intra-cluster similarity to these sets of documents. A high intra-cluster similarity signals a high degree of topical focus in a document set.

For a group of documents D, the intra-cluster similarity (ICS) is computed as follows:

$$ICS_D = \frac{1}{\mid D \mid} \sum_{d \in D} cos(d, c), \tag{4.6}$$

where c is the centroid of D and $cos(d, c)$ is the cosine similarity between document d and c.

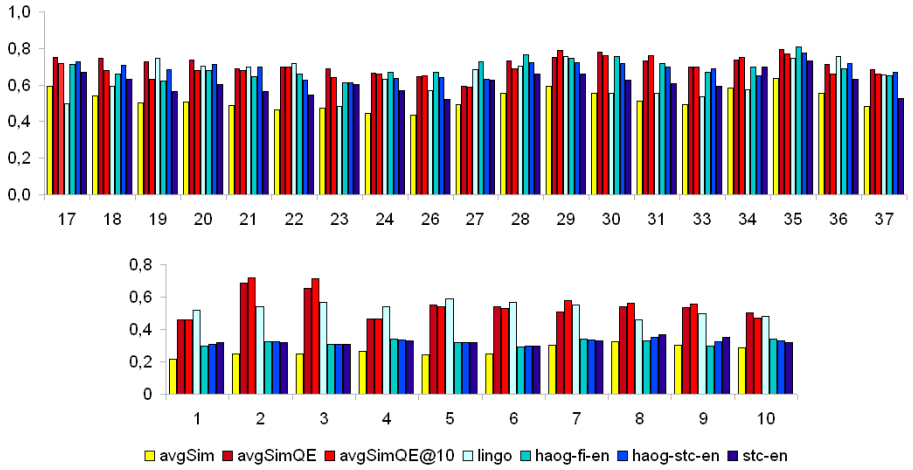


Figure 4.2: Search: average intra-group / intra-cluster similarities for case studies S_1 (top) and S_2 (bottom).

(4.1) Procedure We compared average values of ICS_D for the following ways of grouping all the documents belonging to one time period i into groups D.

avgSimQE From all of the period graphs, we extracted all possible connected subgraphs with size smaller than six. Then for each path, we used node names as R to produce c_i^R if the restriction R returns more than one document, as defined in equation (4.5). This produces one document group per restriction R.

avgSimQE10 The c_i^R for R restricted to the 10 paths with the highest average TR path weight.

avgSim All c_i for the time period. This represents a very conservative baseline of documents only indexed by time and not sub-structured at all.

lingo, haog-fi-en, haog-stc-en, stc-en These grouping methods are a second baseline. They are alternative ways of structuring a corpus while searching it. We used Carrot2's open source DCS (document cluster servers) and chose the four main clustering methods provided by the DCS (lingo, stc-en, haog-fi-en, haog-stc-en), representing state-of-the-art flat and hierarchical methods.[18]

(5.1) Results and interpretation Figure 4.2 shows the results for all periods with non-empty graphs.

The results show an improvement in similarity between the documents selected with the query expansion relative to the overall corpus (avgSim). Also, in most cases our approach provides users with more similar documents than the investigated clustering approaches. Out of the state-of-the-art methods, lingo performs well for most cases; the hierarchical haog-fi-en has an advantage for the relatively small corpus of S_1, where it creates many small clusters with high intra-cluster similarity. The figure also shows that much larger corpus of S_2 leads to, on average, lower intra-cluster/intra-group similarities.

One of the problems we faced was that some of the paths do not return any documents (empty paths). As would be expected, the number of empty paths increases with the number of possible paths. In S_1, the proportion of empty paths in all extracted paths was 19.4%, and in the 10 paths with the highest average weight, it was 11.4%. The corresponding percentages for S_2 were 25.4% and 10.6%. In further work, we aim to develop interface solutions for best indicating empty paths and helping to avoid ineffective searches and user disappointment.

Search: manual evaluation

Even if search with the tool can create internally highly coherent document sets as query results, this will only be really valuable if it also helps users. One important goal of story search is to help users decide whether certain things happened (or not) in the real-world story or, more generally, whether certain statements are true. To investigate this, we performed a study of how users, while interacting with the tool, answered questions about the story, and we also automatically observed selected aspects of their search behaviour.

(4.2) Procedure For each of the case studies S_1 and S_2, two raters from different backgrounds, all with a good command of English and no prior

[18] see `http://project.carrot2.org/algorithms.html` and the references given there

knowledge of the respective story, volunteered to answer questions helped by STORIES. (The rater teams for each of the four human-subjects studies reported in Section 4.7.1 and here were disjoint.) They were given the 20-2 (S_1) and 25-5 (S_2) graphs for all periods of the case-study corpora with non-empty graphs.

The raters received the GUESS software (with search functionality) and the graphs together with a driver script, as well as an Excel sheet with one tab for each time period. Each tab contained two events from the ground truth events that were randomly generated for one of three possible true/false situations (both true, one true, none true).[19]

In addition, the raters received a set of instructions. They then worked individually at their own pace. The raters were asked to inspect the graphs in temporal order and, for each time period, to assign truth values to both statements. They were advised to collect information about the statements by performing one or more searches via highlighting a subgraph and reading as many of the returned documents as necessary. They were asked to attempt to answer the questions quickly.

The filled-in Excel sheets as well as the logfiles of users' interactions were the basis for the following quantitative evaluation.

(5.2) Results and interpretation For each rater, the proportion of correctly assigned truth values was measured (accuracy), as well as the proportion of correctly assigned truth values for in fact true statements and in fact false statements. In addition, the number of searches, the average query size, the average number of returned documents, and the total number of empty paths obtained during the task were measured. (The query size is the number of nodes in the restriction R.) Finally, the average over the raters was formed.

The results are shown in Table 4.3. They illustrate that (a) the judgements by both raters were highly similar in terms of overall accuracy; (b) searching was quite effective, at least in S_1: on average, one or at most two searches with a small size of the restriction graph sufficed to judge two statements; and (c) empty result sets were chosen only very seldomly. The proportion of empty paths was 3.8% for S_1(4.7% for S_2); compare this with the proportion of empty

[19]The questions for a time period i were compiled as follows: First, the value of $numTrue$ was determined by a random choice between 0, 1 and 2. Then, *numTrue* statements were selected randomly from the ground-truth events of i, and *(2-numTrue)* statements were selected randomly from the ground-truth events of $i+1, i+2, ..., I$. The former were treated as the true statements for i, the latter as the false statements for i. A manual check ensured that the latter statements were indeed false in i and not simply repetitions or rephrasings of a true statement in i. The question set was identical for both raters.

Measure	Z	N	Avg.	T	B	Avg.
Accuracy	.75	.75	.75	.67	.72	.69
True positive rate	.81	.71	.76	.50	.50	.50
True negative rate	.67	.80	.73	.88	1	.94
Total number of searches	20	33	26.5	50	45	47.5
Average query size	3.60	3.18	3.39	5.30	5.06	5.18
Average no. of returned docs.	5.10	5.79	5.45	6.30	16.70	11.50
Total no. of empty paths queried	1	1	1	7	0	3.5

Table 4.3: Results of the search task for raters Z, N (S_1, left) and T, B (S_2, right): averages and totals over the time periods with non-empty graphs.

paths that could have been chosen, which was 19.4% (25.4%), see Section 4.7.2 above.

As the true positive and true negative values show, while the overall accuracy was the same for both raters, this did not derive from identical answers. Rather, they were wrong on different events to be rated. The lower true positive rate in S_2 may be the result of the differences between the corpora and the ground truths: The reported ground-truth events of S_2 were more complex than those of S_1, shown by longer sentences with more details. Also, the style of reporting differed; there was a lot of temporally overlapping reporting. Together, this may have made it more difficult to identify the correct facts in the documents. It may also have contributed to the substantially larger number of searches in S_2.

The query size is the number of nodes in the restriction R, thus at the same time the number of words in the extended query. Interestingly, in S_1 query size is very similar to the typical query size in search engines that require users to type their queries: two or three, cf. for example [40]. A new study indicates that (for Google), a long-term average of three words has, in 2008, increased to four words [46]. This result may also indicate that STORIES provides a simplification of search for users: with just one click (one edge = two nodes), they can specify a preferred query size of two; this specification as well as extensions to larger and more informative queries are done by clicking and thus easier than searching. The larger query size in S_2 may partly be explained by a combination of (a) the tendency for participants to include the central node in their queries (which in this case was “Britney Spears” in all periods, cf. Fig. 4.4), and (b) the fact that in indexing, such proper names were counted as two words. (In future versions of the tool, such named entities will be counted as one term.)

Finally, the total number of documents in S_2 returned by each search was

expected to be higher due to the larger size of the corpus and the smaller number of periods. This is reflected in rater B's results, but not in rater T's results. In future work, we will investigate human search behaviour more closely in order to be able to interpret such results better.

Taken together, the evaluation of the search functionality showed that using our approach for structuring story search produces sensible document groups (in the sense that they are more coherent than the non-structured results), and that this is useful for helping users to quickly find out facts about the underlying real-world story.

4.8 Conclusions and outlook

This paper has presented a new problem in the area of temporal text mining: the tracking of story evolution. More specifically, the *ETP3* (evolutionary theme pattern discovery, summary and exploration) problem consists of (a) identifying topical sub-structure in a set (generally, a time-indexed stream) of documents constrained by being about a common topic, (b) showing how these substructures emerge, change, and disappear (and maybe re-appear) over time, and (c) giving users intuitive interfaces for interactively exploring the topic landscape and at the same time the underlying documents. The problem is related to, but extends known problems, in particular evolutionary theme pattern discovery, cf. [35, 42, 27] and the document update problem [32], as well as the detection of bursty events, cf. [21].

By using simple co-occurrence measures on elements that make up a story through the STORIES method, we created a tool that allows users to look at and actively explore story evolution from their individual perspectives through a simple relevance-feedback interface. The tool enables story search and understanding through the same interface, and we have shown that it can be used to structure a corpus better than simple clustering methods. An easily-usable, interactive GUI for tracking story evolution and for story search is a specific focus of this work. Graphs that consist of elements of a co-occurrence network are an easy and understandable way of presenting the development of a story.

We also presented an evaluation framework for approaches to the ETP3 problem and demonstrated the quality of the approach, using two real-life case studies with different domains, timescales, and corpus sizes. This represents an advancement over the state of the art because so far, evaluation with respect to a "ground truth" is mostly lacking from temporal text mining (with TDT and the DUC update tasks, which however address different computational

problems, notable exceptions). The results of our evaluation indicate that non-natural-language interfaces can be used for understanding the events of a story and searching the documents for concrete facts. In the future, we want to extend this framework to also allow for a comprehensive cross-evaluation of different methods (such as "global" clustering or (P)LSA-based methods vs. "local" co-occurrence analysis), different result presentations (natural language, graphs, or other forms) and interestingness measures for patterns (such as time relevance or other measures of burstiness). In addition, the framework will complement our IR/data-mining oriented evaluation by usability assessments. The developed evaluation framework will allow us to carry out larger-scale and more comprehensive evaluations that extend the largely exploratory and formative user studies presented in the current paper.

We are currently experimenting with changing the focus from graphs describing one time period to the *changes* between them. Towards this end, we are investigating graph properties and their correspondences with story ground truth. A first evaluation is reported in [5]; in the future, we plan to include more advanced techniques for describing graph properties [9].

Automatic language processing of the type presented here has a number of limitations. These concern both natural-language understanding and media reception. For example, methods that focus on words/terms, whether local or global, cannot detect negation well. Our method cannot detect possible multiple meanings of one term (homonyms), and a dictionary would be needed to conflate different terms with the same meaning (synonyms). Frequency-based interestingness measures like our time relevance generally single out dominant themes (or ways of reporting) and, by design, neglect outliers that may still be important. Also, the method at present has no notion of or differentiation between news sources of different quality.

Further method and tool developments and evaluations will address these issues. These developments will draw on work in temporal text mining, a field which has generated substantial research but is still lacking a general framework covering the different approaches. We will also draw on other language processing techniques including lexical methods, cf. for example the use of a thesaurus built from Wikipedia in [47]. We will investigate more complex terms and concepts (e.g., n-grams) and syntactical analysis including POS tagging. These variations will be investigated with respect to their usefulness for different kinds of corpora (news, blogs, scientific publications, ...). Another area of improvement is the detection of events represented by bursty features as suggested by Fung et al. [21]. Also, the end-user tool will be developed further to provide more interactions with the corpora.

Due to the modularity of our approach, we expect that it can be generalised to

include different methods for choosing story basics, generating story elements, measuring story salience and tracking story evolution. Modularity will also support different methods for searching and exploring the underlying documents in parallel with the interaction with the story representation.

References

[1] E. Adar, M. Dontcheva, J. Fogarty, and D. S. Weld. Zoetrope: interacting with the ephemeral web. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 239–248, New York, NY, USA, 2008. ACM.

[2] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of news topics. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 10–18. ACM, 2001.

[3] J. F. Allan. *Topic Detection and Tracking.* Springer, Berlin etc., 2002.

[4] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670–2676, 2007.

[5] B. Berendt and I. Subašić. Measuring graph topology for interactive temporal event detection. *Künstliche Intelligenz*, 02/09:11–17, 2009.

[6] M. Biryukov, R. Angheluta, and M.-F. Moens. Multidocument question answering text summarization using topic signatures. *Journal on Digital Information Management*, 3(1):27–33, 2005.

[7] F. Bonchi, C. Castillo, D. Donato, and A. Gionis. Topical query decomposition. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 52–60. ACM, 2008.

[8] U. Brandes and J. Lerner. Visual analysis of controversy in user-generated encyclopedias. *Information Visualization*, 7(1):34–48, 2008.

[9] J. Chan, J. Bailey, and C. Leckie. Discovering correlated spatio-temporal changes in evolving graphs. *Knowledge and Information Systems*, 16(1):53–96, 2008.

[10] C. Chen. *Mapping Scientific Frontiers*. Springer, London, 2003.

[11] C. Chen. Citespace II: Detecting and visualizing emerging trends and transient patterns in scientific literature. *Journal of the American Society for Information Science and Technology*, 57(3):359–377, 2006.

[12] C. C. Chen and M. C. Chen. TSCAN: a novel method for topic summarization and content anatomy. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 579–586, New York, NY, USA, 2008. ACM.

[13] R. Choudhary, S. Mehta, A. Bagchi, and R. Balakrishnan. Towards characterization of actor evolution and interactions in news corpora. In *Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008*, volume 4956 of *Lecture Notes in Computer Science*, pages 422–429. Springer, 2008.

[14] C. Clifton, R. Cooley, and J. Rennie. Topcat: Data mining for topic identification in a text corpus. *IEEE Trans. Knowl. Data Eng.*, 16(8):949–964, 2004.

[15] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 325–332, New York, NY, USA, 2002. ACM.

[16] S. Debnath, P. Mitra, N. Pal, and C. Giles. Automatic identification of informative sections of web pages. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1233–1246, 2005.

[17] J. L. Elsas, J. Arguello, J. Callan, and J. G. Carbonell. Retrieval and feedback models for blog feed search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 347–354, New York, NY, USA, 2008. ACM.

[18] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in KnowItAll (preliminary results). In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 100–110, New York, NY, USA, 2004. ACM.

[19] R. Feldman, M. Fresko, J. Goldenberg, O. Netzer, and L. H. Ungar. Extracting product comparisons from discussion boards. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007)*, pages 469–474. IEEE Computer Society, 2007.

[20] B. M. Fonseca, P. Golgher, B. Pôssas, B. Ribeiro-Neto, and N. Ziviani. Concept-based interactive query expansion. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 696–703, New York, NY, USA, 2005. ACM.

[21] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[22] D. Gruhl, R. V. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 78–87. ACM, 2005.

[23] Q. He, K. Chang, E.-P. Lim, and J. Zhang. Bursty feature representation for clustering text streams. In *Proceedings of the Seventh SIAM International Conference on Data Mining*. SIAM, 2007.

[24] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 76–84, New York, NY, USA, 1996. ACM.

[25] Hollyscoop. Britney Spears News & Pictures, 2007. `http://www.hollyscoop.com/britney-spears/16.aspx`, retrieved 1 March 2009.

[26] W. Huang and P. Eades. How people read graphs. In *APVis '05: proceedings of the 2005 Asia-Pacific symposium on Information visualisation*, pages 51–58, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[27] F. A. L. Janssens, W. Glänzel, and B. D. Moor. Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 360–369. ACM, 2007.

[28] H.-J. Kim and S.-G. Lee. An intelligent information system for organizing online text documents. *Knowledge and Information Systems*, 6(2):125–149, 2004.

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

[30] W. Kules, M. L. Wilson, m.c. schraefel, and B. Shneiderman. From keyword search to exploration: How result visualization aids discovery on the web. Technical report, University of Southampton, February 2008. http://eprints.ecs.soton.ac.uk/15169/.

[31] L. Leydesdorff and T. Schank. Dynamic animations of journal maps: Indicators of structural change and interdisciplinary developments. *Journal of the American Society for Information Science and Technology*, 59(11):1810–1818, 2008.

[32] C.-Y. Lin. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*, 2004.

[33] C.-Y. Lin and E. Hovy. Automated multi-document summarization in neats. In *Proceedings of the second international conference on Human Language Technology Research*, pages 59–62, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[34] X. Ling, Q. Mei, C. Zhai, and B. Schatz. Mining multi-faceted overviews of arbitrary topics in a text collection. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–505, New York, NY, USA, 2008. ACM.

[35] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207. ACM, 2005.

[36] R. Nallapati, A. Feng, F. Peng, and J. Allan. Event threading within news topics. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 446–453, New York, NY, USA, 2004. ACM.

[37] R. Navigli and P. Velardi. Learning domain ontologies from document warehouses and dedicated web sites. *Computational Linguistics*, 30(2):151–179, 2004.

[38] S. Nijssen and J. N. Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 647–652. ACM, 2004.

[39] M. Oka, H. Abe, and K. Kato. Extracting topics from weblogs through frequency segments. In *Proc. of WWW2006 3rd Annual Workshop on the Weblogging Ecosystem*, 2006. http://www.blogpulse.com/www2006-workshop/papers/wwe2006-oka.pdf.

[40] OneStat.com. Most people use 2 word phrases in search engines according to onestat.com, 2004. http://www.onestat.com/html/aboutus_pressbox27.html.

[41] B. Rozenfeld and R. Feldman. Self-supervised relation extraction from the web. *Knowledge and Information Systems*, 17(1):17–33, 2008.

[42] R. Schult and M. Spiliopoulou. Discovering emerging topics in unlabelled text collections. In *Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006*, volume 4152 of *Lecture Notes in Computer Science*, pages 353–366. Springer, 2006.

[43] D. A. Smith. Detecting and browsing events in unstructured text. In *Proceedings of the 25th Annual ACM SIGIR Conference*, pages 73–80. VLDB Endowment, 2002.

[44] I. Subašić and B. Berendt. Web mining for understanding stories through graph visualisation. In *Proceedings of the 2008 IEEE International Conference on Data Mining (ICDM 2008)*, pages 570–579, Los Alamitos, CA, 2008. IEEE Computer Society Press.

[45] M. Thelwall. Blogs during the london attacks: Top information sources and topics. In *Proc. of WWW2006 WS Weblogging Ecosystem*, 2006. http://www.blogpulse.com/www2006-workshop/papers/blogs-during-london-attacks.pdf.

[46] B. Ussery. Google – average number of words per query have increased!, 2008. http://www.beussery.com/blog/index.php/2008/02/google-average-number-of-words-per-query-have-increased/.

[47] P. Wang, J. Hu, H.-J. Zeng, and Z. Chen. Using Wikipedia knowledge to improve text classification. *Knowledge and Information Systems*, 19(3):265–394, 2009.

[48] S.-C. Wang and Y. Tanaka. Topic-oriented query expansion for web search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 1029–1030, New York, NY, USA, 2006. ACM.

[49] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA, 2006. ACM.

[50] C. Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann, 2004.

[51] C. Ware and R. Bobrow. Motion to support rapid interactive queries on node–link diagrams. *ACM Trans. Appl. Percept.*, 1(1):3–18, 2004.

[52] F. Wei, W. Li, Q. Lu, and Y. He. A document-sensitive graph model for multi-document summarization. *Knowledge and Information Systems*, 2009. DOI 10.1007/s10115-009-0194-2.

[53] Wikipedia. Disappearance of Madeleine McCann, 2008. `http://en.wikipedia.org/w/index.php?title=Disappearance_of_Madeleine_McCann&oldid=224183687`.

[54] Wikipedia. Disappearance of Madeleine McCann, 2008. `http://en.wikipedia.org/w/index.php?title=Disappearance_of_Madeleine_McCann&oldid=215814790`.

[55] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas. Visualizing sequential patterns for text mining. In *Proceedings of the IEEE Symposium on Information Visualization 2000 (InfoVis'00)*, pages 105–111, 2000.

[56] J. Xu and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, New York, NY, USA, 1996. ACM.

[57] J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inf. Syst.*, 18(1):79–112, 2000.

[58] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 46–54, New York, NY, USA, 1998. ACM.

[59] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, New York, NY, USA, 2004. ACM.

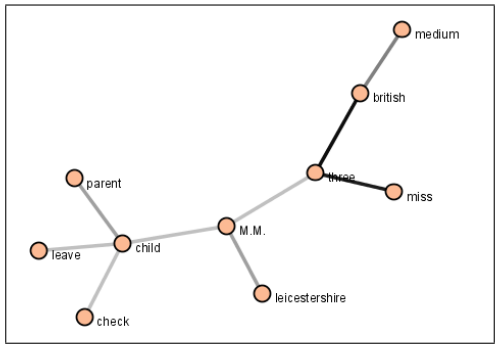


Figure 4.3: S_1, starting period 17 ($TR \geq 3$): Description of an event: a missing child.

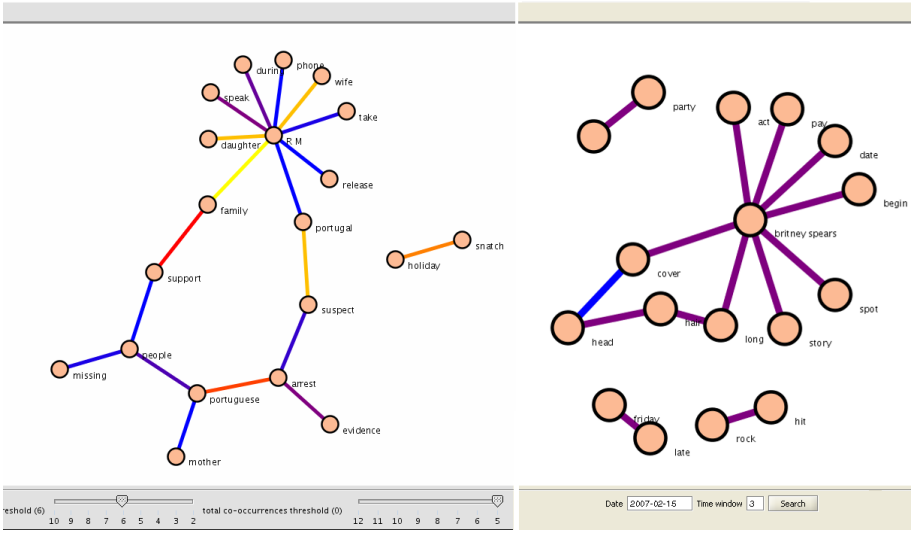


Figure 4.4: Central story figures emerge as central story-graph nodes. Left: R.M. as the prime subject in story S_1 (period 18, $TR \geq 6$), shown with sliders for θ_2, θ_1. Right: Britney Spears is the centre of her own story S_2 ($TR \geq 2$), shown with date-based search and date-based zoom/unzoom function.

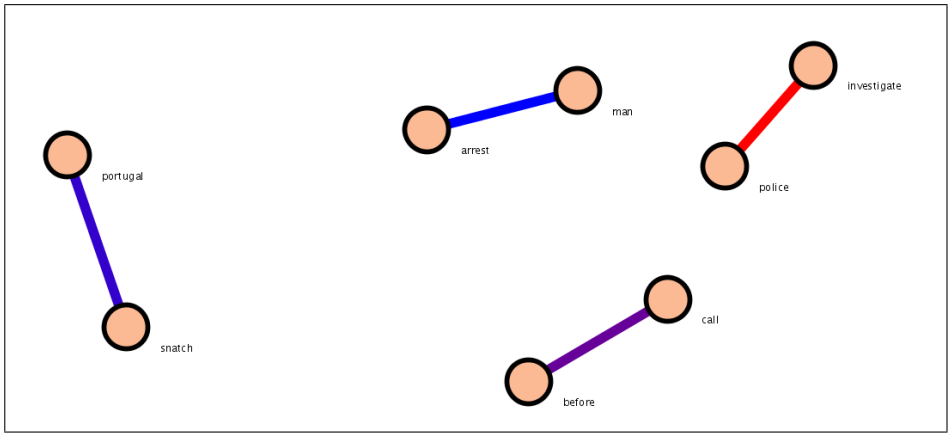


Figure 4.5: S_1, period 26. ($TR \geq 3$): An eventless time.

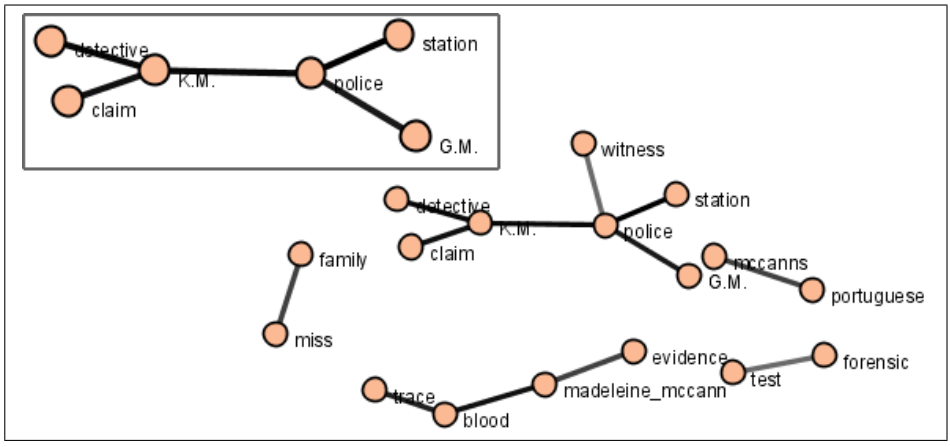


Figure 4.6: Period 34. Event uncovering in S_1. Top ($TR \geq 10$): The police are questioning K.M. ... Bottom ($TR \geq 5$): ... in relation with the blood found in the car.

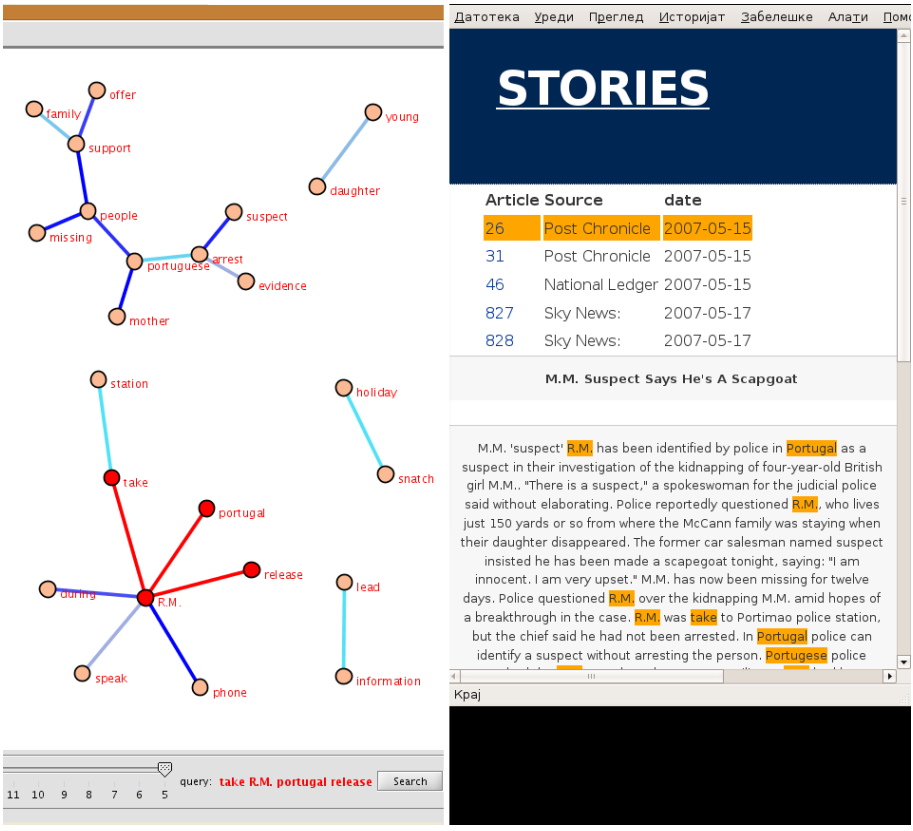


Figure 4.7: Subgraph search: Red (dark) edges and nodes specify the query (bottom) and select documents (right).

Errata

- Section 3.2, page 86, paragraph 2: *media like blogs* should be media such as blogs;
- Section 4.2, page 86, paragraph 2: *innovations of the past few years like the grouping of news articles by topic in Google News have made it easier to keep abreast when* should be innovations of the past few years such as the grouping of news articles by topic in Google News, have made it easier to keep abreast of when;
- Section 4.3, page 90, paragraph 4: *overlapping clusters may be more adequate, for example in the form of Suffix Tree* should be overlapping clusters may be more adequate, for example, in the form of Suffix Tree;
- Section 4.3, page 93, paragraph 2: *fully-fledged linguistic parsers, which does not scale* should be fully-fledged linguistic parsers. This does not scale;
- Section 4.4.1, footnote 4: did not find out salient co-occurrences should be did not determine salient co-occurrences;
- Section 4.5.2, page 96, paragraph 1: *enhanced by HTML-specific and application specific words* should be enhanced with HTML-specific and application specific words;
- Section 4.5.3, page 96, paragraph 1: *enhanced by salience slide rulers that allow* should be enhanced with salience slide rulers (sliders) that allow;
- Section 4.6, page 98, paragraph 4: *This set was extended by the set* should be This set was extended with the set;
- Section 4.6, page 98, paragraph 6: *not extended by such references* should be not extended with such references;
- Section 4.6, footnote 11: *All three suspects were cleared later* should be All three suspects were subsequently cleared;
- Section 4.6, page 99, paragraph 2: *confronted by a deployed STORIES algorithm* should be confronted by the deployed STORIES algorithm;
- Section 4.7.1, page 100, paragraph 2: *ROUGE ("Recall-oriented understudy of gisting evaluation")* should be ROUGE (Recall-oriented understudy of gisting evaluation);

- Section 4.7.1, page 101, paragraph 2: *employing several ground truths by different people* should be employing several ground truths from different people;
- Section 4.7.1, page 101, paragraph 5: *in which they had occurred, such that they could be later assembled* should be in which they had occurred, so that they could be later assembled;
- Section 4.7.1, page 104, paragraph 1: *fewer than n edges, for these, precision and recall* should be fewer than n edges; for these, precision and recall;
- Section 4.7.2, page 113, paragraph 2: *different events to be rated* should be different events;
- Section 4.8, page 114, paragraph 5: *over the state of the art because so far, evaluation with respect to* should be over the state of the art, because so far evaluation with respect to;
- Section 4.8, page 115, paragraph 1: *data-mining oriented* should be data-mining-oriented;
- Section 4.8, page 115, paragraph 1: *more comprehensive evaluations that extend* should be more comprehensive evaluations that will extend;
- Section 4.9, page 115, paragraph 2: *their correspondences with story ground truth* should be their correspondence with story ground truth;
- Section 4.9, page 115, paragraph 3: *measures like our time relevance* should be measures such as our time relevance;
- Section 4.9, page 115, paragraph 4: *cf. for example the use* should be for example, the use.

Chapter 5

Tracking Document-set Evolution

Ilija Subašić and Bettina Berendt: Story Graphs: Tracking document set evolution using dynamic graphs. Intelligent Data Analysis, special issue on Dynamic Networks and Knowledge Discovery. Vol. 17(1), 2013. IOS Press. Accepted for publication June 2011, publication scheduled for 2013.

Contributions as first author:

- (a) Related work overview;
- (b) Conducting the case study;
- (c) Co-defining story graphs;
- (d) TTM evaluation framework definition and implementation;
- (e) Co-interpreting the result of the study.

5.1 Abstract

With the growing number of document sets accessible online, tracking their evolution over time (story tracking) became an increasingly interesting problem. In this paper we propose a story tracking method based on the dynamics of keyword-association graphs. We create a graph representation of the story evolution that we call story graphs, and investigate how graph structure can be used for detecting and discovering new developments in the story. First we investigate the possibly interesting graph properties for development detection. We continue by investigating how graph structure can be linked to the sentences representing developments. For this we create an evaluation framework which bridges the gap between temporal text mining patterns and sentences. We apply this framework to evaluate our method against other temporal text mining methods. Our experiments show that story graphs perform at similar levels overall, but provide distinctive advantages in some settings.

5.2 Introduction

In many genres, text collections are not static and change over time, creating a *story* about the content they discuss. We define stories as corpora of time-indexed textual documents, all relevant to a top-level theme (the whole story, e.g., "Asia Tsunami 2004" or "Enron"). For users (readers), keeping abreast with the changes in large text collections such as news, blogs, or scientific publications is a challenging task. Many services such as RSS feeds aggregators, real-time social media updates, professional news outlets and search engines largely facilitate staying up to date with the current events. Users often follow documents belonging the same story over an extended period of time. We refer to this activity as *story tracking*. For example, a user repeatedly searches for news articles reporting on a sport competition (e.g. "Copa America") from the start until the end of the competition. While tracking a story, users are interested in discovering novel developments regarding the subjects, topics, and events described in the story. This contrasts with regular web search: in story tracking users expect to find both relevant and novel information represented in the documents. We refer to this type of information as *story developments* (developments in short). Such a service will implement a *story tracking method*. Story tracking methods should capture developments with their output (*story representation*). A number of research fields employ different types of story representation ranging from natural language summaries (in document summarization [2]) via sentences (in sentence retrieval [21]), to *temporal text*

patterns – sets of different abstract representations (e.g. a list of words, a distribution of words over topics) used in temporal text mining (TTM) research [12].

The question arises whether graphs, and in particular dynamic graphs, can make a difference in this domain: Do graphs as a special form of temporal summarization have properties and/or allow for operations that can help users in the task of story tracking? To answer this question, one first has to better understand the task and how a computational method could support it. The ultimate motivation for this is to build tools that help users by providing understandable overviews of developments, alerting them to potentially interesting new developments, and of course ensuring that the graphs accurately reflect the content. We propose that a computational story tracking method has to solve three sub-tasks, which we will refer to as components. It has to (a) produce human-comprehensible output (*understandability*), (b) detect the emergence of new developments (*detection*), and (c) discover the details behind these developments (*discovery*). In this paper, we focus on formal questions to provide a basis for such user-centered functionalities. We argue that graph-based structures can be used for story tracking, and present a graph-based story tracking method that solves all the important story tracking sub-tasks. First, we create graph-based patterns, and continue by investigating graph evolution and topology for detecting and discovering developments of a story. In [35] we showed this approach being understandable for humans.

In this paper, we show that graphs are useful for representing the evolution of a document set. To demonstrate our approach we collected five real-life news stories and conducted a case study tracking their evolution.

The major contributions of this paper are: based on a graph-based approach to tracking document-set evolution, described in [35], we (a) investigate topological properties of the dynamic graph (and the story graphs it generates) as indicators of story evolution (detection), (b) study the discovery of novel sentences using graph structure (discovery), and (c) present an evaluation framework for temporal text mining. The paper is a substantial revision and extension of the ideas presented in [4, 36].

The paper is structured as follows: After an overview of related work in Section 5.3, we describe relevant terminology and the method for creating story graphs in Section 5.5. The paper then continues with the discussion on the detection and discovery components in 5.6 and 5.7. Two examples will be used to illustrate key ideas. In Section 5.8 we present the results of our case study. Section 5.9 closes with a summary and an outlook.

5.3 Related work

Major related areas to the work presented in this paper include story tracking methods, story tracking evaluation frameworks, and graph-based text analysis.

Story tracking. In the past decade, a number of methods for story tracking have been proposed. Text-oriented versions of the story-tracking task have been described in the Document Understanding Conference (DUC) Update Summarization task [2] and in the TREC Novelty Detection Track [33]. The basic task of the Update Summarization is to produce a short textual summary of new developments, which is then evaluated against human-written summaries using text-similarity metrics. In the Novelty Detection Track, the most similar task to story tracking is the Novel Sentence Retrieval Task in which the goal is to retrieve sentences previously judged as "new" by human judges. Various text-summarization methods such as [5, 6] and sentence-retrieval methods such as [21, 37] have been applied to these tasks.

Both the DUC Update Summarization and the TREC Novelty Detection tasks are discovery-oriented, and output story representation in a priori defined formats, namely textual summaries and sentences. In contrast to this, more recent methods have focused on mining for lower-level elements or patterns such as keywords, N-grams/term sequences, or LDA components. We refer to these approaches collectively as *temporal text mining* (TTM), e.g. [14, 19, 12]. The key notion of TTM is burstiness – sudden increases in frequency of text fragments, and all TTM methods aim to model burstiness.

Apart from the three groups of methods mentioned above, topic detection and tracking (TDT) [3] tasks such as New Event Detection or First Story Detection tackle the same problem. These TDT tasks decide whether a new document is reporting on an already existing story/topic or a novel (emerging) one.

TDT-oriented methods use documents or groups of documents as story representation, while in this paper we explore methods that extract the story representation from the documents. Although reading full documents may provide a deeper understanding of the story, interacting with story representations such as for example graphs provides users with a different user experience for story tracking.

Relations between developments have been explored in [22, 30, 31]. These methods aim to create a chain of developments in a story. In contrast to this, we are more focused on detecting the intensity of developments in a story and summarizing the story over time periods. To focus on this, we regard developments as independent rather than as (causally) chained.

Like other TTM approaches, we propose a new method for extracting temporal patterns from text. In addition, we also investigate what makes a method suitable for story tracking, and how to evaluate this across methods.

Evaluation frameworks. The DUC evaluation framework [2] for update summarization is a combination of human and automatic evaluation. In this paper we only address the automatic evaluation framework. It has been shown that the automatic evaluation using the ROUGE framework [18] is highly correlated with human evaluation. The ROUGE framework measures the recall of N-grams between the human and machine produced summaries. The most commonly used ROUGE scores are measured on bi-grams ($ROUGE.2$) and skip-4 bi-grams ($ROUGE.SU4$). The TREC Novelty Track [33] which ran from 2002 until 2004, provides an evaluation standard for sentence retrieval. This is a precision/recall oriented framework based the match of retrieved and pre-judged "new" sentences. The main problem of adopting one of these two frameworks for the evaluation of TTM methods is the limited number of documents they use (10 for DUC and 25 for TREC per one topic). TTM methods rely on data mining techniques which require a larger document set for pattern extraction. Another problem is the difference in patterns that TTM methods extract. Summaries for DUC, and sentences for TREC methods are standardized outputs which are directly comparable. The diversity of patterns and the absence of standardised tasks and evaluation procedures, such as in DUC or TREC, render it basically impossible to compare their quality for the story tracking task.

In [25] Roy et al. present a semi-automatic detection of emerging topics in a corpus, and compare these topics with an editor-created list of topics. The application of modified LDA in [39] uses time as one of the latent variables for bursty topic detection, and compares them to topics extracted with the standard LDA. The results show that there are differences in the distribution of bursty-patterns over time between these methods. The idea behind this comparison is that more bursty words will have different distributions over topics in different time periods, while the less or non-bursty patterns will have more similar distribution over topics in different periods. To test the accuracy of their measures of burstiness defined on word-topic distribution, Knights et al. [15] create an artificial document set by drawing words from a set of word-topic distributions. In selected periods the words are drawn from a subset of topics making these topics bursty. The authors measure whether their method captures this artificial burst. Wang et al. in [40] test their method by comparing bursts discovered in multilingual corpora on the same topic.

Most of the TTM evaluation procedures are tailored for evaluating only one

method, assessing how well it discovers bursts in evolving corpora. We wish to measure not only whether a burst is discovered, but also how story representation helps in detection and discovery of the developments that created the burst. In sum, therefore, none of the existing methods satisfy our evaluation goal complexity.

Text as graphs. A number of papers looked into representing text using graphs and use its structure for specific tasks. Schenker et al. in [27] describe possible uses of graph analytics methods for analyzing content of Web pages. Inspired by Page Rank, Günes and Radev [6] present the LexRank summarization algorithm. This algorithm, creates a summary by selecting the most salient sentences in a sentence similarity matrix. A similar approach was followed in [5, 20]. Wong et al. in [42] present a domain-independent way of visualising pairwise associations of words using graph-based representation. Templates of natural language expression of events in news corpora using graph-based representation of text is presented in [38]. Similarly to this, Rusu et al. [26] present a graph-based visual analysis technique based on semantic (grammatical) relationships between words in a document. Lexical graphs [41] are used for many text mining tasks, such as clustering [29] and recommendation [23]. Similar to our work, Heyer at al. in [11] employ graphs based on co-occurrences to track topics over time. However, there graphs are focused on one "topic" word and they track how co-occurrences with this world alone changes.

In contrast of most above mentioned graph-based text mining methods, in this paper we include the time dimension and explore how to use graph-representation of text content along a timeline.

5.4 Preliminaries

The developments of a story are conceptual and therefore in many cases elusive, hidden in the text, and hard to express using natural language. To investigate the effectiveness of story graphs in understanding, detecting, and discovering the developments, we need to define a way of representing them (*development representation*). Depending on the genre of the corpus that a user explores, developments can be expressed in a variety of ways, most common of which are sentences. News reports story developments can be expressed using natural language sentences describing topics, events, and subjects around which the story revolves. An important reason for this is the mainly narrative structure of news reports: developments are described by propositional statements. For

example, one development in the story about "famine in Africa" could be represented by a sentence: "The United Nations declared a famine in two regions of southern Somalia on July 20;". In the following, we equate sentences with developments and say that each sentence is a (potential) development.

For other document genres, developments may be harder to express. For example, in a corpus of scientific publications where topics of focus change over time, representing developments as sentences does not fully capture the semantics behind them. Similarly for blog corpora, where users are interested in discovering novel opinions, stands, and sentiment of other bloggers, sentences do not fully capture the developments.

For evaluating story graphs for all three story tracking components, we wish to control for the effects of developments representation. Our goal is to minimize the possible effects artificial (not natural language) development representation might have on our results. Therefore, in this paper we focus on news articles.

Story representations produced by different story tracking methods often differ from development representation of the story. Should this be the case? Concretely for this work, why should we output graphs when the developments are represented using sentences? There are two reasons to do this; the first is formal, the second user-related. Formally, we expect that the structure of graphs (their topology and evolution as dynamic graphs) may suggest effective mechanisms for detection and discovery – in particular, subgraphs with specific properties may emerge that correspond to story developments. The second, user-related, reason is that story tracking is not only about discovering developments, but also understanding and summarizing the changes in a document set, following how a story evolves over time, and providing hints when new developments occur. Different story representations may provide users with summaries of developments, give them freedom to explore them, facilitate detecting the changes in a document set, and keep a way of linking to the development representations. Therefore, having story representations differ from development representations may improve user experience during story tracking, moving away from "simple" retrieval into a more engaging and intuitive way of story tracking. We approach story tracking as an interactive task in which users interact with the system which enables understanding, detection, and discovery of story developments.

5.5 Creating story graphs: method and understandability

5.5.1 The STORIES method: story graphs and the evolution graph

The graph-based method we use in this papers takes a corpus of time-stamped documents as an input. First, this corpus D is transformed into a sequence-of-terms representation. Subsequently, the *content-bearing terms* are extracted. We defined content-bearing terms as 100 most frequent terms plus 50 most frequent named-entities appearing in the corpus. Next, the corpus is partitioned by publication periods, e.g. calendar weeks. Thus, D is the union of all document sets D_t, with $t = 1, \ldots, n$ the time periods.

For each D_t, the *frequency* of the co-occurrence of all pairs of content-bearing terms b_j in documents is calculated as the number of occurrences of both terms in a window of w terms, divided by the number of all documents in D_t. This measure of frequency and therefore relevance is normalised by its counterpart in the whole corpus to yield *time relevance* as the measure of burstiness: $TR_t(b_1, b_2) = freq_t(b_1, b_2)/freq_D(b_1, b_2)$. Thresholds are applied to avoid singular associations in small sub-corpora and to concentrate on those associations that are most characteristic of the period and most distinctive relative to others: $\theta_1 \in N$ is a lower bound on the total number of co-occurrences, and $\theta_2 \in R$, usually with $\theta_2 > 1$, is a lower bound on the time relevance of a co-occurrence. This gives rise to the *story graphs* $N_t = \langle V_t, E_t \rangle$ for time period t. The edges E_t of N_t are $\{(b_1, b_2) \mid \#\ co\text{-}occ.s\ of\ b_1, b_2\ within\ w\ terms\ in\ doc.s\ from\ freq_t(b_1, b_2) \geq \theta_1\ and\ TR_t(b_1, b_2) \geq \theta_2\}$. The nodes V_t of N_t are the terms involved in at least one association in this symmetric graph: $\{b_j \mid \exists b_k : (b_j, b_k) \in E_t\}$. To obtain a view into document evolution over multiple time periods we define an *evolution graph* EG. At time period t', it is defined as a union of all story graphs N_t where $t \leq t'$. An evolution graph can be viewed as a dynamic graph, and each story graph as its current graph obtained by restricting it to a selected time period. Algorithm 1 illustrates the story graphs and the evolution graph generation process.

Figure 5.1 illustrates the relations between story graphs and the evolution graph over three time periods ($t_0 - t_2$). In the first period both story graph (N_0) and the evolution graph (EG) are the same. In the next periods story graphs summarize developments of the periods they belong to, while the evolution graph summarizes the developments starting with the first time period. For example, in t_1 evolution graph contains all nodes and edges from t_0 plus the

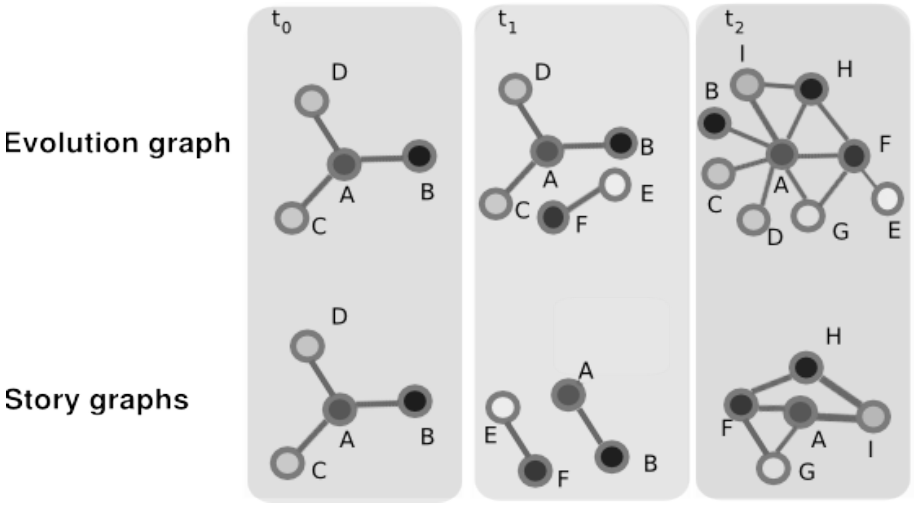


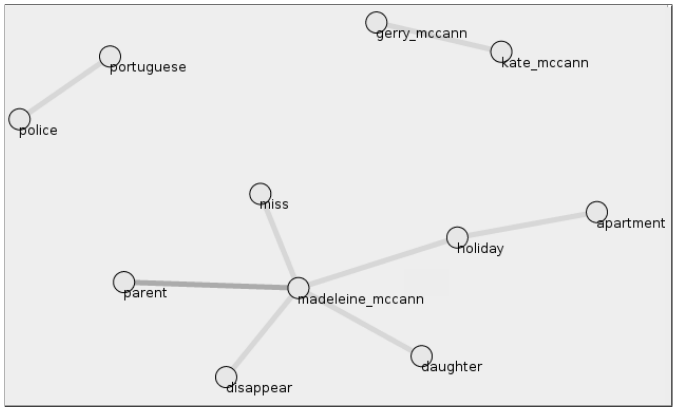
Figure 5.1: Relations between document set, story graphs and their corresponding evolution graph.

newly appeared nodes and edges in t_1 (E – F), while the story graph N_1 contains only edges that are bursty for the period t_1 (edges $A - D$ and $A - C$ are not in the story graph for t_1).

Example 1: Figure 5.2(a) shows a story graph. The concrete story in this case regards a missing child and the criminal investigation around it. The graph shown in the figure describes the first four days after the initial event. The graph summarizes the main questions in a news report: *who* (missing child as a central node in the largest component, and the parents in the top right corner), *what* ("disappear" and "miss" nodes are linked to the child), and *where* ("holiday" and "apartment" are linked to the child, and the country is shown in the top left corner). Another important news question, *when*, is implied by the period users explore. In total the graph shown in the image summarizes 263 documents.

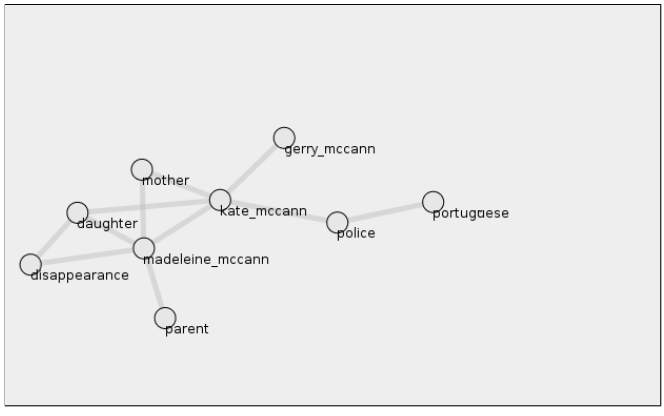
5.5.2 Understandability

A first question to ask of such a method is whether it generates human-understandable story representations. To test this (as well as to make the method as a whole accessible to human users for summarizing news), we created the STORIES tool [34] that supports story tracking using the method



((a))

((b))



((c))

Figure 5.2: Example story graphs: (a) shows a summarization in the initial period of a story; (b) and (c) show story graphs in two periods with different eventfulness (low (b), high (c)).

described in this section. The STORIES tool demo is available at `http://ariadne.cs.kuleuven.be/WebStories/viz/fds.html`.

Previously, in [35], we evaluated the understandability component in a series of user studies and automatic procedures. In the first study we presented the participants with a set of story graph visualizations and a set of descriptions of important developments for the same period. Participants' task was to match the edges of the graph to the descriptions. In total about 80% of the descriptions were matched to the edges of the graph. We also report the results on story search evaluation in which participants were presented with a set of story graphs (with search functionality) and a set of YES/NO questions. The task in this study was to explore the documents in a time period using the story graph and to answer the questions. In total participants answered correctly on about 75% of the questions. Additionally, we looked at the coherence of the data sets created by path restriction from story graphs. We compared them to state-of-the art document clustering algorithms. The results show that using story graphs to create structure in a document sets yields a more coherent document grouping.

5.6 Detection

When users track a story over time, they are particularly interested to learn when "something has happened". In a first step, they want to be alerted to periods in which a lot has happened (detection), and in a second step, they want to get "the best overview of what it was" (discovery). In this section, we deal with the first step (in Section 5.7 we explore the second step). We call periods in which a lot has happened "eventful" and measure this concept via a ground truth that we obtain from external sources. Formally, the ground truth is a set of sentences, and we measure eventfulness by the number of sentences in this set. A story tracking method is good at detection, if the patterns it generates have a measurable and human-detectable property that correlates with the eventfulness.

Example 1 (cont.): an example of changes in story over time is shown in Figure 5.2. The figure shows (in 5.2(b)) the story graph around the 4th of September (with an interval of seven days), a rather eventless time. The change effected by moving the search forward by one day to the 5th is shown in Figure 5.2(c). It marks a major change in the real-life story (a crime case: after a long period of no new findings in the criminal investigation, DNA evidence was found, which led to the identification of new suspects); the event becomes visible by the large increase in graph size and connectedness.

Hypotheses and measures of developments and graphs We investigated the relationships between topological properties and indicators of eventfulness. We measure the *local eventfulness* (LE) as the number of developments in a time period. Local eventfulness corresponds to discovering developments by observing properties of a single story graph. We measure the *global eventfulness* (GE) at time t_n as a number of all developments occurring in time periods up to and including t_n. The properties are at two levels – local and global. A single story graph has *local properties*. Local properties can be graph properties and node properties. An evolution graph has *global properties*. The idea of this is that changes of an evolution graph can point to new developments.

We investigated the following properties and hypotheses:

Local graph properties related to size: number of nodes and number of edges of a story graph. These properties stand out in the visualization as a "more densely populated" story graph. The hypothesis is that these measures are positively related to LE. For the example in Figure 5.1: in period t_1 the size of graph is 4 nodes and 2 edges, while in t_2 the size is 5 nodes and 5 edges. We investigate whether this difference in size of the graph correlates with the number of developments summarized by these graphs.

Local graph properties related to connectedness: size of the largest connected component (LCC) of the story graph and the average size of connected components (CC) (standing out visually in an obvious way), expected to be positively related to LE.

We also considered the number of connected components. This was because informal inspection of the story graphs indicated that low-LE times were often marked by many small subgraphs, while high-LE times may have more cross-connected "story-lines". In Figure 5.1 the size of LCC for t_0 is 4 and there is only 1 component, while in t_1 the size of LCC is 2, and there are 2 components. On the other hand, if connected components mark different events, their number should be positively related to LE.

Local properties of individual story-graph nodes: degree (in the graph layout chosen, leading to "central-looking" nodes), degree centrality (normalized by the maximum possible degree given by the number of nodes - 1, which visually requires the user to relate the node to the whole graph), and sum of TR weights (see Section 5.5.1) of the adjacent edges. The sum of weights was used in a non-normalized fashion because average weights led to artifacts for small connected components. For each of the measures we take the maximum value from the nodes of a story graphs. The hypothesis is that the existence of a central, highly and strongly connected node points to developments. Thus we expect that these measures are positively related to

LE. Observe the difference between node A in story graphs for periods t_1 and t_2 in Figure 5.1. In period t_2 node A has higher degree centrality and appears to be more central than in t_1. This arises from many bursty co-occurrences A is involved in this period. Thus, it is reasonable to assume that A is put into a spotlight due to its importance in representing the developments of a period.

Global properties: the number of nodes, the number of edges, the size of the largest connected component, the number of connected components and the average size of connected components of the evolution graph.

We expect that the inclusion of new (previously not present) concepts (nodes) and relations between them (edges) is a result of new developments. In Figure 5.1 observe the difference between the evolution graph in t_1 and t_2. In t_1 there was an addition of 2 nodes and 1 edge, while in t_2 there were 4 new nodes and 6 new edges. We suspect that the newly included nodes in the evolution graph are related to the new developments. Thus, we investigate the correlation of the in size and connectedness of the evolution graph and the total number of developments.

By measuring graph topology of story graphs and the evolution graph we aim to discover which properties can detect the existence of new developments in a story. These properties can be used to alert users of the existence of possible new developments in an observed time period. This can be done, for example, by highlighting certain parts of a story graph.

5.7 Discovery

The question whether certain graph properties reflect eventfulness can obviously be asked of a method that outputs dynamic graphs as story representation, but not of methods that generate natural language or keyword representations. In addition, even if we find that a certain graph property signals eventfulness, we do not know whether the corresponding graph highlights "the right" events. To address these two questions we developed a framework for cross-method evaluation of discovery.

As explained in Section 5.6, we focus on the news domain where developments can be represented using sentences to which we refer as ground-truth sentences. Although many TTM methods generate story representations using different bursty patterns, there is no well-defined approach to linking them with the ground-truth sentences. Our basic assumption is that the format of the patterns suggests – to human users and also to a formal approach – a way of combining

them into sentences and that these sentences can then be compared against the ground-truth sentences.

In this section we define a procedure for linking story representation with sentential description of developments in a story. The procedure is based on sentence retrieval using queries generated from story representation. The query generation procedure is defined for three major groups of TTM methods listed in Section 5.7.2. To evaluate the results from this procedure, we define an evaluation framework.

5.7.1 Linking to Sentences

To enable the direct comparison of different bursty patterns we developed a procedure for obtaining sentences which these patterns resemble in the best way. We use sentence retrieval methods to obtain the same representation (sentence representation) for different methods. The large number of developed sentences retrieval algorithms and the lack of benchmark method for sentence retrieval (in the TREC framework [33]) make the decision on using one method challenging. However, a detailed analysis of sentence retrieval presented in [21] used Query-likelihood retrieval method (QL) with Jelinek-Mercer smoothing on a pseudo-document index of sentences as a baseline. Therefore, we consider the QL method to be a sensible choice for our framework. Input for this model is an index of pseudo-documents and a set of queries used for retrieving. Pseudo-documents are created from the sentences of a corpus D, and queries are generated from story representations.

5.7.2 Query generation

For generating the queries used for sentence retrieval, we combine elements of the story representation (story elements) – e.g. an element of keyword list story representation is a single keyword. Each element in story representation has a burst score attached to it. The combination of elements greatly depends on the internal structure of the story representation. We consider two approaches to query generation. The first one (general query generation) is the same for all methods, and uses top $maxR$ story elements from story representation. The second approach (model-specific query generation) takes into account the structure of different story representations. The idea of model-specific query generation to combine the basic bursty pattern elements into more complex queries. In our previous work [36] we tested the difference between general and model-specific query generation procedures. The results of that analysis showed that in almost all cases model-specific query generation improves upon

the results of sentence retrieval. Therefore, in this paper we concentrate on model-specific query generation.

Story graphs query generation. The basic story element of our method is keyword association, while graphs are more complex pattern generated by joining binary associations. Therefore, for model specific query generation we use the graph structure of story graphs. We first extract paths from each story graph up to size $maxQ$ and order them by descending average TR path weight. Then we sort them based on the average edge weight. The final list of queries is obtained by cutting the the sorted path list at position $maxR$. The query is formulated by joining the node names (with white space between them) starting from the first node in a path.

Query generation for other models. The lack of query generation procedures and a large number of developed TTM methods makes defining query generation a challenging task. It would be virtually impossible to retrospectively define these procedures for each model. Therefore, we propose a grouping the methods based on story representation. We divide TTM tracking methods in three major groups: (a) keyword representation, (b) group representation, and (c) combo representation methods. For story representation, the first group [14, 7, 9, 10, 32] uses a list of bursty N-grams ranked by their burst scores. The second one [7, 39, 19, 28, 13] joins bursty N-grams into groups which point to developments. The last group combines simple patterns into more complex ones; story graphs are an example of this group.

The model-specific query generation of keyword representation methods is based on the combination of bursty keywords. First, we extract the $maxR$ highest ranked story elements from model's story representation. Then we combine them by creating all possible combinations not larger than $maxQ$. We rank these newly formed combinations based on the average burst scores of their elements and use the top $maxR$ as queries for retrieving sentences.

Group representation methods output groups of text content as story representation (e.g. a distribution of words over topics, where each topic is one group) describing subjects. The procedure for query generation based on the story representation with k *groups* is as follows. For each group we extract $maxR/k$ story elements with the highest in-group burst score and combine them into all possible combinations not larger than $maxQ$. Then we rank the new in-group element combinations based on the average burst scores of their elements. For each group we use the top $maxR/k$ combinations as queries for retrieving sentences. We assume that all groups are equally important and use $maxR/k$ story elements from each group.

Example 2: Figure 5.3 in rectangles II and III shows an example of model-specific query generation procedure for all three groups of methods.

5.7.3 Evaluation framework

We first obtain a corpus of news-article documents and development representation ground-truths, all of them time-stamped. We divide the corpus into time periods $t = \{t_1, t_2, \ldots, t_n\}$ of equal length. For every time period, we build an index I_t out of sentences belonging to the documents from the time period. For each t, we obtain the development representation as a external (not in a corups) set of ground-truth sentences $G_t = \{g_{1t}, g_{2t}, \ldots, g_{nt}\}$, where n is the index number of ground-truth sentence in period t. Given a set of model-specific generated queries Q we retrieve the top ranked sentence using the QL retrieval method, and create a set of retrieved sentences R.

Metrics. We define the measures of similarity between retrieved and ground-truth sentences using the ROUGE [18] framework.

Ultimately, we want a set of measures which shows to what extent retrieved sentences capture the ground truth, as well as how many sentences are needed to obtain a "good" ground truth match.

Atomic measures. We use the ROUGE framework, which measures the recall of n-grams between the method-generated and the ground-truth sentences. The most commonly used ROUGE scores are measured on bigrams ($ROUGE.2$) and skip-4 bigrams ($ROUGE.SU4$). For every retrieved sentence r_{tk} $(1 \leq k \leq n)$ we calculate the $ROUGE.2$ ($s1_{tkj}$) and $ROUGE.SU4$ ($s2_{tkj}$) scores against every ground truth sentence in the same time period – g_{tj} $(1 \leq j \leq n)$. This will give us a Cartesian product of R_t and G_t where each element has attached scores. We refer to this set as the scores set $C_t = \{(r_{tk}, g_{tj}, s1_{tkj}, s2_{tkj})\}$.

Aggregated measures. We define a set of aggregated measures to quantify how well the set of retrieved sentenced matches the ground-truth set, what percentage of the best possible ground truth match is obtained from the retrieved set, and how the number of retrieved sentences influence these scores.

To find the best possible match in the set of retrieved sentences we define $maxM$ measure as the maximum score any retrieved sentence in a period t has for the ground truth j. For $ROUGE.2$ scores $maxM$ is defined as:

$$maxM.2_{tj} = max(s1_{tkj}) \in C_t \tag{5.1}$$

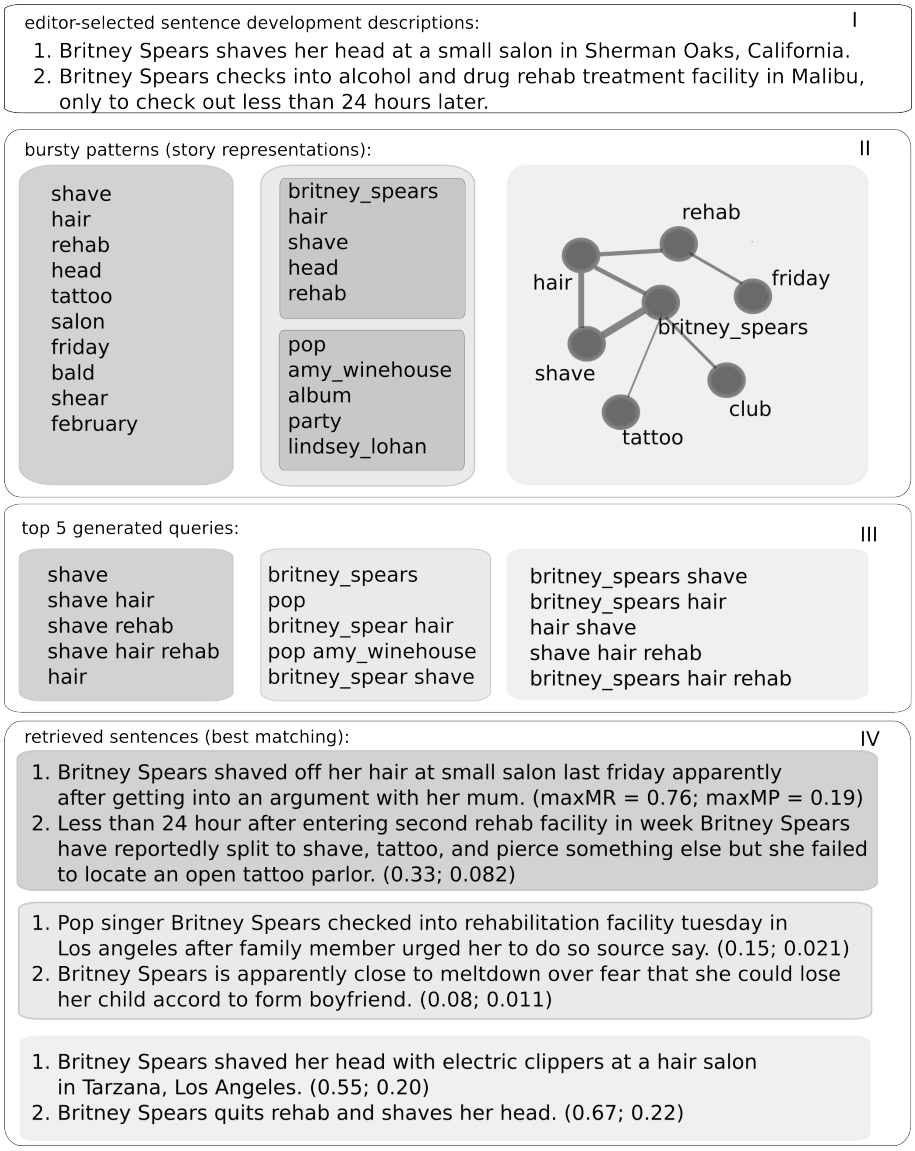


Figure 5.3: Framework for discovering developments with the example story Britney Spears. Rectangle I shows the ground-truth editor-selected sentences; II shows example story representation for three groups of methods (sub-rectangles mark different groups - topics); III shows top 5 generated queries from the above story representation; and IV shows the best retrieved sentences, with $maxMR$ and $maxMP$ scores in brackets.

With $maxM$ we measure how retrieved sentences match the ground truth sentences. However, since the ground truth sentences do not originate from the same corpus as the retrieved sentences it is hard, if not impossible, to obtain the maximum match of 1. The maximum $maxM$ score a ground truth can obtain varies from one ground truth to the other. So, in order to normalize the $maxM$ score, we introduce $maxMR$ measure. It shows what percentage of the optimal match with the ground truth is obtained by the retrieved sentences. . We create a set of maximum match scores $O_t = \{(s_{tk}, g_{tj}, s1_{tkj}, s2_{tkj})\}$ having maximum $ROUGE.2$ and $ROUGE.SU4$ scores between G_t and S_t, where $S_t = \{s_{1t}, s_{2t}, \ldots, s_{ht}\}$ is a set of *all* sentences from a period t. For $ROUGE.2$ scores it is defined as:

$$maxMR.2_{tj} = max(s1_{tkj}) \in C_t / max(s1_{tkj}) \in O_t \tag{5.2}$$

$maxMR$ is a "recall oriented" measure. However, different methods may retrieve different number of sentences, and the ones with a larger retrieved sentence set increase the chance of having a better match. We take this into account and define a new measures – $maxMP$. For $ROUGE.2$ scores $maxMP$ is defined as:

$$maxMP.2_{tj} = maxMR.2_{tj} * min(|G_t|, |R_t|)/|R_t| \tag{5.3}$$

The $maxMP$ rewards the methods that produce a good fit with a small number of retrieved sentences (matching the usually small number of ground-truth sentences). In this sense the measures correspond to "top heavy precision-oriented" measures like precision@k.

Analogously with the Equations 6.1, 6.2, and 6.3 we define $maxM$, $maxMR$, and $maxMP$ measures for $ROUGE.SU4$.

The motivation behind these metrics is to punish those methods which retrieve too many sentences and reward those which retrieve approximately the same number of sentences as the number of ground truth sentences.

Testing procedure. Lacking a baseline method, we turn to the cross evaluation testing procedure, assessing the performance of multiple methods against each other. Given a set of time periods T, a set of methods M, a set of ground truths G, a set of metrics X, and a set of indices I: for every $t \in T$ and for every $M_\bullet \in M$, we calculate all measures from the previous paragraphs. This gives rise to, for every metric $x \in X$, $|M|$ sets $F_{mx} = \{f_{m11}, \ldots, f_{mtj}\}$, where j is a number of ground truths in a period t. For every x, we then test results of different methods using Friedman's and Tukey's multiple comparison test to asses the differences between different methods.

5.7.4 Framework illustration

Example 2 (cont.): Figure 5.3 illustrates an example of the development discovery framework described in this section. Rectangle I shows the editor-selected, ground truth sentences – descriptions of developments in a time period. Rectangle II shows examples of story representation (bursty patterns) extracted from the corpus for the three method groups. The left-most rectangle shows bursty keyword list generated by [14] – keyword representation; the middle rectangle shows 2 bursty topics (in embedded rectangles) generated by [19] – group representation, and the right-most rectangle contains a story graph representation generated by STORIES [35] – combo representation. Based on these story representations we generate the queries shown in rectangle III. Top 5 queries are generated from the bursty patterns in rectangle II following the generation procedure described in 5.7.2. Using the queries and the QL retrieval method, the "best" sentences for all 3 methods are retrieved from the corpus (shown in rectangle IV). Best sentences are here defined as the ones most similar to the editor-selected sentences in rectangle I. The $maxMR$ and $maxMP$ scores for sentences are shown in the brackets.

5.8 Case study

In this section we present a case study that explores story graphs. We focus on the detection and discovery components and report on the results for 5 news stories from different domains.

5.8.1 Corpora and ground truth

Corpora. To demonstrate our approach we needed a large-enough (for pattern extraction) and timestamped document set describing the same (news) story. Many available evaluation corpora only partly satisfy these conditions. Therefore, we decided to collect corpora using Google News[1] and Google News Archive search.[2] In total, we collected 5 stories using queries issued to the search engines as the identifiers of a story. We collected stories following pop singer Britney Spears (D_1), the accident in a Chilean mine in 2010 (D_2), the Greek financial crisis (D_3), a missing child case (D_4), and the British Oil Horizon platform oil spill (D_5).[3] Table 5.1 gives an overview of sizes and

[1] `http://news.google.com`
[2] `http://news.google.com/archivesearch`
[3] Example 1 is from the story D_4, and Example 2 from story D_1

query	days	start	|docs|	|words|	|lemmas|
britney spears (D_1)	59	01-01-2007	2,871	51,049	48,087
chile miners (D_2)	73	20-08-2010	2,552	50,823	43,316
greek crisis (D_3)	90	01-12-2009	1,865	38,962	36,982
madeleine mccann (D_4)	173	04-05-2007	1,541	31,050	29,332
oil spill (D_5)	124	20-04-2010	18,589	184,687	143,457

Table 5.1: Corpora basic statistics.

time spans of these corpora.[4] After harvesting the documents we applied the content extraction algorithm described in [24] to remove all auxiliary content such as navigation, related pages, and ads. The rest of the pre-processing was done in the same way as in [35].

Ground truth. Apart from creating corpora, we also needed to define a ground truth describing the developments in the observed stories. In similar evaluation frameworks ([2, 33]) the ground truth was established using several human editors. The Web itself provides us with the documents that contain a similar result to a multi-rater ground truth definition. We first turned to Wikipedia, and searched for timeline[5] pages about the stories we collected. We consider that the crowd-sourcing strength of Wikipedia does a good job in emulating a large number of human editors. For D_2[6], D_3[7] and D_5[8] we were able to find pages containing the annotated timeline of the developments. The Wikipedia page about D_4[9] contained only free text, and not a timeline format description of the developments. To extract the timeline from Wikipedia text we employed two human judges. Our previous work [35] contains details about the annotation procedure. For D_1 we did not find enough content in Wikipedia, and turned to other sources.[10] The procedure for extracting the ground truth for D_1 was the same as for D_4.

[4] We made the corpora freely available at `https://sites.google.com/site/subasicilija/corpora-for-story-tracking`

[5] Timeline creation aims to create a temporal ordering of important developments independent on the time when developments become known (e.g. published in media). Story tracking differs from this and summarizes developments based on the time when they become known, and not the actual time when the development occurred. These two tasks slightly differ, but in our case the developments are transparent making timeline creation almost identical task to story tracking.

[6] `http://en.wikipedia.org/wiki/2010_Copiapo_mining_accident#Timeline_of_events`

[7] `http://en.wikipedia.org/wiki/2000s_European_sovereign_debt_crisis_timeline`

[8] `http://en.wikipedia.org/wiki/Timeline_of_the_Deepwater_Horizon_oil_spill`

[9] `http://en.wikipedia.org/wiki/Disappearance_of_Madeleine_McCann`

[10] `http://www.hollyscoop.com/britney-spears/16.aspx`

	|V|	|E|	|LCC|	#CC	avg.size CC
D_1^{30}	.37	.04	.42	.36**	.23
D_2^{30}	.57**	.29*	.26	.62**	.51
D_3^{30}	.57**	.52**	.47*	.39***	.38
D_4^{30}	.72***	.75***	.79***	.62***	.60***
D_5^{30}	.28	.46*	.19	.25	.24
D_1^{full}	.05	.22	.29	.24	.34*
D_2^{full}	.44	.41	.40	.44	.23
D_3^{full}	.47	.41	.43**	.42	.45*
D_4^{full}	.75***	.77***	.77****	.63***	.77***
D_5^{full}	.40**	.24*	.21**	.09	.08

Table 5.2: Local graph properties and their Pearson correlation coefficients with local eventfulness (LE). *(**/***) indicates significance at the 10% (5%/1%) level.

5.8.2 Detection results

Settings. Before the analysis of the detection component, we generated story and evolution graphs. This included setting a number of parameters to the method described in Section 5.5.1. Our initial experiments showed that 30 edges is a reasonable number of edges to both represent developments and be "human understandable". Since detection can be automated, and does not necessarily include users, the understandability constraint could be relaxed. Therefore, we used two settings for story graphs, one with a maximum of 30 edges referred to as $D_\bullet^{30}$, and the other with no constraint on the number of edges, referred to as $D_\bullet^{full}$. For D_1 we set the story graph time span to 5 days, and for the rest of the stories to 1 week. The thresholds θ_1 and θ_2 were set to 4 and 2 respectively. In total we created 148 story graphs for each edge number setting. For each story, we also created an evolution graph.

Results and discussion. We measured the Pearson correlation between number of developments in a time period, and properties of story and evolution graphs in the same time period.

Local graph properties and LE. Table 5.2 shows the correlation coefficients between local graph properties and local eventfulness. Detection is somewhat more successful using a restriction on a number of edges (rows 1-5). This is illustrated by a higher number of significant correlations, when compared to the non-restricted story graphs (rows 6-10).

	degree	degree centrality	edge-weight sum
D_1^{30}	.25	.70***	.20
D_2^{30}	.08	.30	.74***
D_3^{30}	.25	.42	.13
D_4^{30}	.56***	.01	.20
D_5^{30}	.17	.08	.01
D_1^{full}	.72***	.50	.53
D_2^{full}	.01	.16	.01
D_3^{full}	.11	.46*	.10
D_4^{full}	.19	.15	.27
D_5^{full}	.23**	.25	.30

Table 5.3: Local node properties (maximum values for a story graph) and their Pearson correlation coefficients with local eventfulness (LE). *(**/***) indicates significance at the 10% (5%/1%) level.

For the story graphs with maximum 30 edges the highest number of significant correlations (4 out 5) is for the number of connected components. Other important local graph properties for the same graphs are the number of nodes and the number of edges. For the non-restricted story graphs there is no property that stands out.

Looking at the individual stories, the number of significant correlations differs in many cases. We found the reason for this in the distribution of developments over time periods. For example, in the most extreme case for D_4, where all properties are highly significant, the average number of developments per time period is 2.13, with a standard deviation of 2.98. For this story there are time periods with no developments followed by time periods with a high number of developments. In such cases it was easier to "capture" the important properties. On the other side, for stories with more uniform development distribution like S_5 (average number of developments: 1.12, and st.dev:0.18), it was hard to find differences in the story graphs.

Local node properties and LE. The next analysis we performed investigated the local node properties correlation to local eventfulness. The results are summarized in Table 5.3. Based on our initial user tests, we suspected that centrality of nodes in a story graph points to developments. However, we were not able to find consistent significant correlations. This leads us to the conclusion that centrality of nodes does not suggest new developments.

Global properties and GE. Finally we analyzed the correlation between evolution graph properties and global eventfulness. The results are shown in

	|V|	|E|	|LCC|	#CC	avg.size CC
D_1^{30}	.84**	.37	.09	.17	.71
D_2^{30}	.08	.37	.41	.61**	.80
D_3^{30}	.66*	.66*	.62*	.01	.07
D_4^{30}	.61***	.80***	.76***	.39***	.39*
D_5^{30}	.58**	.85*	.20	-.24	.45*
D_1^{full}	.59**	.19	.11	.33	.45*
D_2^{full}	.03	.50	.48	-.29	.85***
D_3^{full}	.85***	.48***	.48***	-.01	- .38
D_4^{full}	.66***	.84***	.80***	.45*	.39***
D_5^{full}	.59***	.46**	.47**	.54**	.52**

Table 5.4: Global graph properties and their Pearson correlation coefficients with global eventfulness (GE). *(**/***) indicates significance at the 10% (5%/1%) level.

Table 5.4. We found that the most important property is the addition of new nodes. Both with and without restriction on the number edges the number nodes was significant in 4 out of 5 cases. Important properties are also the number edges ($|E|$) and size of the largest connected component ($|LCC|$).

Discussion. We analysed properties of the whole graphs and specific nodes. The most important properties for development detection we discovered are: number of connected components and number of nodes for static graphs and the number of newly added nodes for dynamic graphs. In contrast with the local properties, for global properties, connected components were not systematically significant. The reason for this is that the increase in graph size does not always lead to an increase in the number of connected components. In this case, the number of connected components does not change over time, while the global eventfulness changes, yielding no correlation. The analysis of global properties showed that tracking the newly used concepts (nodes) evolution graphs can be used to detect new developments.

5.8.3 Discovery results

Settings. The framework for evaluating discovery component (Section 5.7) described a way of obtaining sentences from story representation. We used the same settings for story graph generation as for the detection evaluation. Apart from the STORIES method, referred to in this section as M_1 (with the same

settings as for the detection evaluation), we needed to choose methods for cross-evaluation. The question is how to choose the methods which both represent the groups described in Section 5.7.2 and are of high quality. We decided to use the methods that received the most attention by the community over the last few years.[11] Namely, for keyword representation we chose Kleinberg's burst detection algorithm (M_2) [14] and for group representation we used a probability mixture method developed by Mei and Zhai (M_3) [19].

Method M_2 [14] discovers bursty words by minimizing their state-transition cost between bursty and non-bursty states. The cost is defined as the increase in proportion of documents relevant to an observed word between two time periods. The method takes a scaling factor as a parameter, which determines the lift in the proportion of relevant documents between two time periods needed for a word to reach the bursty state. We use the parameter value 2, as set in [14]. In the original paper, the algorithm was applied to scientific publications. The same method was later used to find bursts in blogs [16] and to track quotes in news [17]. We coded the algorithm following the description in the original paper.

An extension of the probabilistic mixture method for topic discovery presented described in [43] was used in [19]. This method, M_3, outputs set of bursty topics represented by word distributions. Each of the different distributions points to a subject. The original paper reports on a number of parameters for document-topic, topic-time, and word-topic distributions. We used the same values as reported by the authors. However, the authors report on a threshold value set by empirical testing which is not described in detail in the papers. We were therefore not able to replicate this, and manually set the threshold to 5 developments ("topics"). We coded the algorithm employing a modification of the implementation of [43] in the DRAGON NLP Toolkit.[12]

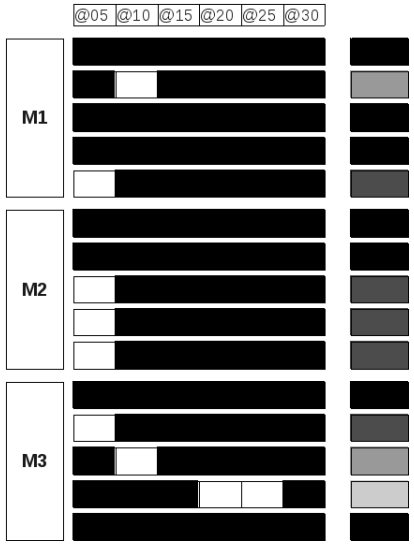
As discussed in Section 5.7.2, for each method we generated a set of retrieved sentences using model-specific query generation. We set the query-generation parameter $maxQ$ to 5, and we varied $maxR$ from 5 to 30 with a step of 5. The value of the first parameter was chosen based on query length in major search engines [1]. Variations of the second parameter simulate the situation in which different number of top story elements are used.

We calculated $maxM$, $maxMR$, and $maxMP$ for all settings and for both $ROUGE.2$ and $ROUGE.SU4$, and tested them as described in Section 6.6, resulting in a total of 24 tests.

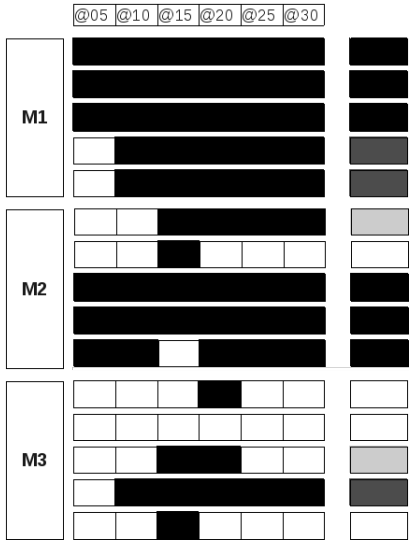
[11] By no means we are arguing that the chosen methods are the best among all possible methods, nor trying to diminish the quality of other work.

[12] `http://dragon.ischool.drexel.edu/features.asp`

Results. The large number of test settings and methods made it difficult to aggregate the results of the tests, so we created multiple aggregations. With these aggregations we aimed at capturing the following information: (a) which methods are robust to different $maxR$ settings, and (b) whether there is a difference between "precision-like" and "recall-like" performance.



((a)) $maxMR$



((b)) $maxMP$

Figure 5.4: Top group method comparison matrix of the recall-oriented $maxMR$ (a), and the precision-oriented $maxMP$ (b) measures based on ROUGE.2.

To aggregate the tests of $maxMR$ and $maxMP$ results and summarize the performance of models, we created the matrix shown in Figure 5.4. Each row shows a combination of one method and one corpus; grouped by methods to highlight similarities across corpora. Each row show results for one of the stories starting with D_1.

Columns show test settings: the use of only the top $maxR$ patterns or pattern elements for sentence retrieval. "Top" was measured by the respective methods' quality measure (M_1: time relevance, M_2: burstiness score, M_3: probability of being in a bursty topic); $maxR$ is equal to the number of bursty story elements (edges) in M_1, the number of bursty keywords in method group M_2, and in M_3 to keywords with highest probability from each topic.

A cell is black if, according to the Friedman test, this method-setting combination was in the group with the best results of the measure, i.e. there was no statistically significant difference in results between all the methods with a black cell in one column, but all of these significantly outperformed all of the others (the ones with a white cell in this column). Further significant differences between these lower-quality methods existed, but are not shown.

The heatmap to the right of the table shows the robustness of the method quality over settings, ranging from low (= only in good group for few settings, light grey) to high (= always in good group, black). Differentiation is measured as: $number\ of\ blocks - (number\ of\ white\ cells\ +\ number\ of\ block\ holes)$. The larger number of holes shows that the method is less robustne. Values were then binned into four categories of grey shades.

Figure 5.4(a) shows that for the "recall-oriented" $maxMR$ measure, the differentiations between methods are low. Story graphs extracted using M_1 perform slightly better than the other two methods. However, there is small differentiation (maximum 2 blocks out of 30) leading to the conclusion that for $maxMR$ methods have similar scores. This means that differences between the ground-truth sentences and all sentences retrieved by all methods are low. For example, M_1 – M_3 were indistinguishable in quality for D_1, regardless of number of patterns used. This is shown by each cell (settings @5-30 patterns) being black in the first row. In contrast, M_1 and M_2 were indistinguishable for @5 and Story D_2 (the first cells in row 2 are both black), but significantly better than M_3 (whose first cell in row 2 is white). For @10 and D_2, M_2 and M_3 outperform M_1. For settings $\geq$@15, all three methods were indistinguishable for D_2.

On the other hand, when we look at the results of the "precision-oriented" measure $maxMP$ (Figure 5.4(b)), the differentiation between method is higher. As for $maxMR$, method M_1 performs the best. Results for M_3 vary most

between the two measures. We suspect that the reason for this is the structure of M_3 group based story representation. Each group should point to one development. However, the number of ground-truth sentences and the number of groups (topics) was fixed to 5 following the approach in [19], while on average for all stories we had 2.54 events. This means that there were fewer developments than groups that describe them, and this reduces precision.

5.9 Future work and conclusions

Summary and conclusions. Stories change over time, and we explored how to use graphs for tracking these changes. In this paper we focused on how graphs help solve two story tracking sub-tasks: detecting the emergence of new developments and discovery of original description of the developments. Indicating the existence of new developments during the story is an essential task of story tracking. In Section 5.6 we explored static and dynamic properties of graphs to identify the ones correlated with the existence of new developments. The results presented in Section 5.8.2 show that often the size of connected component for static graphs, and the number of edges and nodes for dynamic graphs indicate new developments. On the application side, this finding can be used for highlighting a story graph if it has many new edges and/or nodes compared to the previous one. This can draw users attention to new developments, and to automatically detect interesting time periods.

As shown in Section 5.7, obtaining a representation of developments in the news domain is a challenging task. To evaluate our method, we first developed a framework for evaluating temporal text mining algorithms, and then cross-evaluated our method in Section 5.8.3. The results show that our method performs higher or at the same level as other methods in most tested settings.

To sum up, in this paper we investigated graphs as a possible solution for story tracking. We showed that our graph-based method can be used for detecting new developments and discovering their representation in a corpus.

Limitations. Story tracking is a wide research area, and we focused this paper on a number of issues surrounding it. Most of our goals were oriented towards exploring the news domain, and it could be the case that in other domains some of our assumptions are weaker. The case study presented in this paper is of limited size, and results should be understood as such. Testing more methods with more stories would require a large effort from the community in setting standardized data sets and benchmarks. We consider that by defining

an evaluation framework and conducting a case study, we have made a first step necessary towards this goal.

Future work. There are many ways in which we can expand on the graph-based story tracking. One of the possible directions is to define an automatic procedure for development detection based on the graph properties, and build an automatic detection component. More generally, we believe that many text mining tasks can benefit from graph mining algorithms, and that graph-based representation of text present a interesting challenge in graph mining. For example, possible tasks that would benefit from graph based approach are corpus visualization, exploration, and comparison. Although there has been substantial research in graph-based text analytics, we hope that this paper will increase the interest of the graph mining community in further exploration of textual data and tasks related to them.

Algorithm 1 Story graphs and evolution graph extraction algorithm.

Input: $D = \{D_0, D_1, ..., D_n\}$ // document set divided into n time periods t; θ_1, θ_2 // frequency and TR thresholds; $B = \{b_0, \ldots, b_n\}$ // set of content-bearing terms; w - co-occurrence window size; c // maximum edges in story graphs
Output: N // a set of story graphs for all periods;
EG // the evolution graph
Procedure:
$N \leftarrow \emptyset$
$EG \leftarrow \{V_{eg} = \emptyset, E_{eg} = \emptyset\}$
$freq_T \leftarrow \mathbf{0}$ // co-ccurrence frequency vector for all periods
$t \leftarrow 0$ // initial time period
while $t \leq n$ **do**
  $N_t \leftarrow \{V_t = \emptyset, E_t = \emptyset\}$ // a story graph for period t
  $freq_t \leftarrow \mathbf{0}$ // co-ccurrence frequency vector for period t
  $TR_t \leftarrow \mathbf{0}$ // *time relevance* vector for period t
  for all $d \in D_t$ **do**
    for all pairs of content-bearing words (b, b') *within* w *terms in* d **do**
      $freq_t(b, b') + +$ // increase the co-occurency frequency in t
      $freq_T(b, b') + +$ // increase the co-occurency frequency in T
    end for
  end for
  for all $(b, b') \in freq_t$ **do**
    $TR_t(b, b') \leftarrow freq_t(b, b')/freq_T(b, b')$
    if $(freq_t(b, b') > \theta_1 \wedge TR_t(b, b') > \theta_2)$ **then**
      $V_t \leftarrow V_t \cup V' = \{b, b'\}$ // add b and b' to node set of the story graph for period t
      $E_t \leftarrow E_t \cup E' = \{b', b, TR_t(b, b')\}$ // add the edge between b and b' with the TR as weight in the story graph for period t
    end if
  end for
  $Sort(N_t)$ // sort the story graph edges based on TR
  $Top(N_t, c)$ // cut the story graph at top-c edges
  $N \leftarrow N \bigcup N_t$ // add the story graph for t into the set of all story graphs
  for all edges $e(b', b', TR(b', b')) \in E_t$ **do**
    $V_{eg} \leftarrow V_{eg} \bigcup V' = \{b, b'\}$ // add b and b' to node set of the evolution graph
    $E_{eg} \leftarrow E_{eg} \bigcup E' = \{b, b', TR_t(b, b')\}$ add the edge between b and b' with the TR as weight in the evolution graph
  end for
  t++
end while
return N, EG

References

[1] Hitwise: Google's search share continues to grow. "http://www.bizreport.com/2009/05/hitwise_googles_search_share_continues_to_grow.html". HitWise Inc., 2009, retrieved July 2011.

[2] Duc 2007: Task, documents, and measures, 2007. http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot, National Institute of Standards and Technology, US Department of Commerce, 2007, retrieved July 2011.

[3] J. F. Allan. *Topic Detection and Tracking*. Springer, Berlin, 2002.

[4] B. Berendt and I. Subašić. Measuring graph topology for interactive temporal event detection. *Künstliche Intelligenz*, 02:11–17, 2009.

[5] C. C. Chen and M. C. Chen. Tscan: a novel method for topic summarization and content anatomy. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 579–586, New York, NY, USA, 2008. ACM.

[6] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. (JAIR)*, 22:457–479, 2004.

[7] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[8] R. Grossman, R. J. Bayardo, and K. P. Bennett, editors. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21–24, 2005*. ACM, 2005.

[9] D. Gruhl, R. V. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In Grossman et al. [8], pages 78–87.

[10] Q. He, K. Chang, E.-P. Lim, and J. Zhang. Bursty feature representation for clustering text streams. In *Proceedings of the Seventh SIAM International*

Conference on Data Mining, April 26–28, 2007, Minneapolis, Minnesota, USA, pages 491–496. SIAM, 2007.

[11] G. Heyer, F. Holz, and S. Teresniak. Change of topics over time and tracking topics by their change of meaning. In *KDIR '09: Proc. of Int. Conf. on Knowledge Discovery and Information Retrieval*, pages 223–228, October 2009.

[12] M. Hurst. Temporal text mining. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[13] F. A. L. Janssens, W. Glänzel, and B. D. Moor. Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12–15, 2007*, pages 360–369. ACM, 2007.

[14] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7:373–397, October 2003.

[15] D. Knights, M. Mozer, and N. Nicolov. Detecting topic drift with compound topic models. In *Proceedings of the 3rd International AAAI Conference on Weblogs and Social Media*, ICWSM'09. AAAI, 2009.

[16] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 568–576, New York, NY, USA, 2003. ACM.

[17] 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*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM.

[18] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[19] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In Grossman et al. [8], pages 198–207.

[20] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[21] V. Murdock and W. B. Croft. A translation model for sentence retrieval. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 684–691, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[22] R. Nallapati, A. Feng, F. Peng, and J. Allan. Event threading within news topics. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 446–453, New York, NY, USA, 2004. ACM.

[23] S. Papadopoulos, F. Menemenis, Y. Kompatsiaris, and B. Bratu. Lexical graphs for improved contextual ad recommendation. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 216–227, Berlin, Heidelberg, 2009. Springer-Verlag.

[24] J. Prasad and A. Paepcke. Coreex: content extraction from online news articles. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1391–1392, New York, NY, USA, 2008. ACM.

[25] S. Roy, D. Gevry, and W. M. Pottenger. Methodologies for trend detection in textual data mining. In *Proceedings of the Textmine '02 Workshop*, Washington, DC, USA, 2002. SIAM.

[26] D. Rusu, B. Fortuna, D. Mladenić, M. Grobelnik, and R. Sipoš. Visual analysis of documents with semantic graphs. In *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration*, VAKD '09, pages 66–73, New York, NY, USA, 2009. ACM.

[27] A. Schenker, H. Bunke, A. Kandel, and M. Last. *Graph-theoretic Techniques For Web Content Mining; electronic version.* World Scientific, Singapore, 2005.

[28] R. Schult and M. Spiliopoulou. Discovering emerging topics in unlabelled text collections. In *Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006, Thessaloniki, Greece, September 3–7, 2006, Proceedings*, volume 4152 of *Lecture Notes in Computer Science*, pages 353–366. Springer, 2006.

[29] Y. Sha, G. Zhang, and H. Jiang. Text clustering algorithm based on lexical graph. *Fuzzy Systems and Knowledge Discovery, Fourth International Conference on*, 2:277–281, 2007.

[30] D. Shahaf and C. Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 623–632, New York, NY, USA, 2010. ACM.

[31] B. Shaparenko and T. Joachims. Information genealogy: uncovering the flow of ideas in non-hyperlinked document databases. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 619–628, New York, NY, USA, 2007. ACM.

[32] D. A. Smith. Detecting and browsing events in unstructured text. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and*

development in information retrieval, SIGIR '02, pages 73–80, New York, NY, USA, 2002. ACM.

[33] I. Soboroff and D. Harman. Novelty detection: the trec experience. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 105–112, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[34] I. Subašić and B. Berendt. Experience stories: a visual news search and summarization system. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III*, ECML PKDD'10, pages 619–623, Berlin, Heidelberg, 2010. Springer-Verlag.

[35] I. Subašić and B. Berendt. Discovery of interactive graphs for understanding and searching time-indexed corpora. *Knowl. Inf. Syst.*, 23(3):293–319, 2010.

[36] I. Subašić and B. Berendt. From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 517–522, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[37] S. O. Sweeney, F. Crestani, and D. E. Losada. 'show me more': Incremental length summarisation using novelty detection. *Inf. Process. Manage.*, 44(2):663–686, 2008.

[38] M. Trampus and D. Mladenic. Constructing event templates from written news. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '09, pages 507–510, Washington, DC, USA, 2009. IEEE Computer Society.

[39] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA, 2006. ACM.

[40] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 784–793, New York, NY, USA, 2007. ACM.

[41] D. Widdows and B. Dorow. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[42] P. C. Wong, W. Cowley, H. Foote, E. Jurrus, and J. Thomas. Visualizing sequential patterns for text mining. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 105–, Washington, DC, USA, 2000. IEEE Computer Society.

[43] C. Zhai, A. Velivelli, and B. Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 743–748, New York, NY, USA, 2004. ACM.

Errata

- Section 5.1, page 130, paragraph 1: *(story tracking) became an increasingly interesting problem* should be (story tracking) has become an increasingly interesting problem;
- Section 5.2, page 130, paragraph 2: *keeping abreast with the changes in large text collections* should be keeping abreast of the changes in large text collections;
- Section 5.3, page 132, paragraph 1: *Major related areas to the work* should be Major areas related to the work;
- Section 5.3, page 132, paragraph 2: *The basic task of the Update Summarization* should be The basic task of Update Summarization;
- Section 5.3, page 133, paragraph 2: *The TREC Novelty Track [33] which ran from* should be The TREC Novelty Track [33], which ran from;
- Section 5.3, page 133, paragraph 3: *a semi-automatic detection of emerging* should be a semi-automatic detection method of emerging;
- Section 5.3, page 133, paragraph 2: *less or non-bursty patterns will have more similar distribution over topics* should be less bursty or non-bursty patterns will have more similar distributions over topics;
- Section 5.3, page 134, paragraph 2: *words using graph-based representation* should be words using a graph-based representation;
- Section 5.3, page 134, paragraph 2: *co-occurrences with this world* should be co-occurrences with this word;
- Section 5.4, page 134, paragraph 4: *in a variety of ways, most common of which are* should be in a variety of ways, the most common of which are ;
- Section 5.4, page 134, paragraph 4: *News reports story developments* should be News reports' story developments;
- Section 5.4, page 135, paragraph 1: *of southern Somalia on July 20;* should be of southern Somalia on July 20th.;
- Section 5.5.2, page 139, paragraph 2: *Participants' task was* should be The participants' task was;
- Section 5.6, page 139, paragraph 4: *a crime case* should be a criminal case;

- Section 5.6, page 140, paragraph 4: *size of the largest connected component* should be the size of the largest connected component;
- Section 5.6, page 140, paragraph 4: *expected to be positively related to LE* should be ; both are expected to be positively related to LE;
- Section 5.6, page 141, paragraph 3: *we investigate the correlation of the in size and connectedness* should be we investigate the correlation in the size and connectedness;
- Section 5.7.1, page 142, paragraph 3: *The large number of developed sentences retrieval algorithms and the lack of benchmark method* should be The large number of developed sentence retrieval algorithms and the lack of a benchmark method;
- Section 5.7.1, page 142, paragraph 3: *presented in [21] used Query-likelihood* should be presented in [21] used the Query-likelihood;
- Section 5.7.2, page 143, paragraph 3: *while graphs are more complex pattern generated by* should be while graphs are more complex patterns generated by;
- Section 5.7.3, page 146, paragraph 5: *multiple comparison test to asses the differences* should be multiple comparison test to assess the differences;
- Section 5.7.4, page 147, paragraph 3: *The left-most rectangle shows the bursty keyword list generated by [14]* should be The left-most rectangle shows the bursty keyword list generated by the method described in [14]
- Section 5.8.2, page 150, paragraph 2: *st.dev.* should be standard deviation;
- Section 5.8.3, page 152, paragraph 2: *In the original paper* should be In the original paper [14];
- Section 5.8.3, page 155, paragraph 6: *the differentiation between method is higher* should be the differentiation between methods is higher;

Chapter 6

Temporal Text Mining Evaluation Framework

Ilija Subašić and Bettina Berendt: From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In Proceeding of the 19th European Conference on Artificial Intelligence (ECAI 2010), pages 517-522, Amsterdam, The Netherlands, 2010. IOS Press.

Contributions as first author:

- (a) Related work overview;
- (b) TTM evaluation framework definition;
- (c) Conducting the case study.

6.1 Abstract

Many document collections are by nature dynamic, evolving as the topics or events they describe change. The goal of temporal text mining is to discover bursty patterns and to identify and highlight these changes to better enable readers to track stories. Here, we focus on the news domain, where the changes revolve around novel, previously unpublished, "facts" that have an effect on the story developments. However, despite intense research activities on bursty patterns, a lack of common procedures today makes it impossible to compare methods in a principled way. To close this gap, we (a) investigate how different temporal text mining methods discover novel facts and (b) present an evaluation framework for methods assessment, consisting of a set of procedures and metrics for cross-evaluating models. Bursty patterns are transformed into queries for sentence retrieval, either with or without taking into account internal pattern structure, and these sentences are compared with a set of editor-selected ground-truth reference sentences. Our experiments on different classes of temporal text mining show that different methods including our own story-graph method, perform at similar levels overall, but provide distinctive advantages in some settings. The experiments also demonstrate the benefits of using patterns' internal structure for query generation.

6.2 Introduction

One of the frequent use cases of today's Internet is *story tracking*: the use of search engines, archives or other sources for following the developments of a topic over time. Usually, this is done by executing a series of searches, often with the same search query such as the name of a person ("Amy Winehouse"), an event ("Tsunami", "Winter Olympics"), or a scientific area ("text mining"). The information need behind story-tracking searches differs from that behind one-off searches: search results should not only be relevant, but also novel. Novelty is generally conveyed by documents with *bursty* content elements: words, n-grams, terms highlighted by LDA distributions, or similar elements appearing significantly more frequently in a time window of search than in other times. This calls for new analysis, search and interaction methods.

In the past decade, a number of methods for story tracking have been proposed. Text-oriented versions of the story-tracking task have been described in the Document Understanding Conference (DUC) Update Summarization task [19] and in the TREC Novelty Detection task [23], and several text-summarisation or sentence retrieval methods have been applied to these tasks. These methods re-use sentences from the original documents to produce a description of the

novel developments, either as a summary or as a set of retrieved sentences. The tasks are associated with a well-defined evaluation procedure suited to natural-language texts. In contrast to this, more recent methods have focused on mining for lower-level elements or patterns such as keywords or n-grams/term sequences. We refer to these approaches collectively as *temporal text mining* (TTM) [12, 17, 9]. These methods bear much promise in terms of the additional flexibility afforded by the discovery of sub-sentential patterns. However, the diversity of patterns and the absence of standardised tasks and evaluation procedures such as in DUC or TREC have rendered it basically impossible to compare their quality for the task.

To close this gap, we (a) investigate how different TTM methods discover novel or bursty "facts" and (b) present an evaluation framework for methods assessment. We concentrate on the news domain, in which most novel developments can be expressed in sentential form (e.g., "The ski slalom ended with ... winning the gold medal"). Our basic assumptions are that users construct their own description out of the patterns they are presented with and that this description is sentential, such that its quality can be assessed by the degree to which these constructed sentences (the presumed novel "facts") resemble "true" sentences. The challenge is to measure an aggregate re-construction quality over the possible/plausible fact constructions. We therefore present procedures and metrics for (i) focusing on patterns or pattern combinations that a TTM method highlights; (ii) turning these into "fact" sentences; (iii) inspecting and comparing the degree of resemblance between the "fact" and the ground-truth sentences. We do this by (i) formulating new pattern-specific ordering and combination operations, drawing on method-specific burstiness scores for the former; (ii) employing sentence retrieval methods in a new way; (iii) proposing new metrics that capture "recall" and "precision" characteristics and build on the ROUGE evaluation framework for summarisation evaluation [16]. We illustrate this cross-method evaluation method by (iv) comparing three typical TTM methods [12, 17, 24] on two corpora consisting of online news documents and an editor-selected set of ground-truth sentences.

To the best of our knowledge, this is the first systematic cross-methods evaluation framework for TTM methods. Our contributions are thus the formulation of such a framework and the demonstration of its use and of the interpretability of its results. After a brief overview of related work given by Section 6.3, Sections 6.4–6.7 describe the above steps (i)–(iv), and Section 6.8 concludes.

6.3 Related work

In this section, we give a short overview of related evaluation approaches and frameworks, and we explain why they are not directly applicable to our questions for comparing TTM methods.

Evaluation frameworks for the DUC update and TREC novelty tasks. Introduced in 2007 as a part of the DUC workshop series [19], the evaluation framework for update summarization combines human and automatic evaluation. In this paper, we only address the automatic evaluation part. It has been shown that the automatic evaluation using ROUGE framework is highly correlated with the human evaluation [16]. The ROUGE framework measures the recall of n-grams between the human and machine summaries. Most commonly used ROUGE scores are measured on bigrams ($ROUGE.2$) and skip-4 bigrams ($ROUGE.SU4$). An evaluation framework for sentence retrieval methods has been standardized through the TREC Novelty track, which ran from 2002 until 2004 [23]. Among the different tasks of the Novelty track, the one most similar to story tracking is novel-sentence retrieval, in which the goal is to retrieve sentences previously judged as "new" by human editors.

The main problem one faces when trying to adopt one of these two frameworks for the evaluation of TTM methods, is the limited number of documents they use (10 for DUC and 25 per topic for TREC). TTM methods rely on data mining techniques which require a larger document set for pattern extraction. Another problem are the differences between the patterns that TTM methods extract. Summaries for DUC and sentences for TREC methods are standardized outputs which are directly comparable. In contrast, the output of TTM methods differs, and it is hard to compare it without an intermediate step that creates comparable representations out of the different patterns.

Temporal text mining evaluation procedures. Most of the evaluation procedures in this research field were developed to evaluate a single TTM method. Roy et al. [20] presented a method for semi-automatically detecting and naming emerging topics and compared these topics with an editor-created list. Wang and McCallum [26] modified LDA using time as one of the latent variables for bursty-pattern detection. They compared this modified LDA to standard LDA and showed differences in the distribution of bursty patterns over time. The idea is that more bursty words should have different distributions over topics in different time periods, while the less or non-bursty patterns should have more similar distributions over topics in different periods. To test

the accuracy of their measures of burstiness defined on word-topic distribution, Knights et al. [13] created an artificial set by drawing words from a set of word-topic distributions. In selected periods, the words were drawn from a subset of topics, making these topics bursty. The authors measured whether their method captures this artificial burst. The STORIES method [24], which produces "story graphs", was evaluated with a user test in which participants were asked to connect bursty graph elements with sentential ground-truth events. This gave rise to precision and recall scores. In a second study, participants were given the story graphs and asked to make a true/false decisions on an event. Wang et al. [25] tested their method by comparing bursts discovered in multilingual corpora on the same topic.

Most of these evaluation procedures are tailored to evaluating only one method, assessing how well it discovers bursts in text streams. We wish to measure not only if a burst is discovered, but also whether and how this burst's representation helps users discover the ground truth sentences that created the burst.

Topic detection and tracking (TDT). In addition to the foregoing, TDT tasks such as new event detection, on-line new event detection, and story-link detection also address the same problem [2]. The TDT tasks decide whether a newly arrived document reports on an already existing story/topic or on a novel (emerging) one. However, all of the TDT tasks operate at a document level, and they only decide between "old" and "new". In contrast, we want to evaluate novelty detection at a more fine-grained level, both in terms of the syntactical units and in terms of a content characterisation.

6.4 Patterns, representations, and TTM groups

TTM methods output bursty patterns that point to the changes in the story they track, and the subjects arising from these changes. *Subjects* constitute the high-level story; they can be for example events (e.g., a specific ski slalom in the Winter Olympics) or topics (e.g., doping). The patterns consist of *story elements*, syntactical units extracted from the underlying documents. For example, an element could be a term, and the pattern this term plus some score assigned to it. We also define a *story representation* as a set of bursty story elements used to represent a subject. Story elements have different levels of expressiveness, i.e. different amounts of information about subjects conveyed. TTM methods operate on sub-sentence story elements, and we distinguish the following elements: tokens, n-grams, and n-gram groups. Further filters on

these types are possible, giving rise to elements such as terms with above-threshold frequencies or other weights, or n-grams identified as named entities or similarly semantic entities.

Token is used in a computational-linguistics sense: a series of characters not containing any of a set of pre-defined delimiters.

N-grams are content-bearing tokens. Basic n-grams are unigrams (1-grams), where every token is a unigram. More advanced n-grams are sequences of n contiguous (or not) tokens extracted from the text. Non-consecutive, or skip-m n-grams, contain n tokens appearing in a window of m tokens. For example, in a text *[big bad wolf sleeps]*, skip2 bigrams are *[big-bad], [bad-wolf],[big-wolf], [big-sleeps], [bad-sleeps]*, and *[wolf-sleeps]*.

N-gram groups are collections of n-grams pointing to the same subject. These groups can be n-gram cluster center values, latent variables' probability distributions over n-grams, or some other way of grouping by similarity.

Every element of a story representation has some weight assigned to it. This weight can be a specific "burstiness score", a probability of an element appearing in a bursty subject, the relative importance in a bursty subject cluster center, or a weight in a latent component. While the detailed mathematical properties of these weights vary, they all induce some order of importance on the elements; we therefore regard all these weights as *burst scores* of the respective story element and use these scores in the same way in the query-generation processes that are explained in Section 6.5.

Based on the differences in their story representations, we divide TTM tracking methods into three groups: (a) keyword representation, (b) group representation, and (c) combo representation methods. Group (a), presented in [12, 4, 6, 7, 22], uses a list of bursty n-grams ranked by their burst scores. Group (b) [4, 26, 17, 21, 10] joins bursty n-grams into groups which point to subjects. Group (c) methods use a combination of the previous two approaches [24, 1].[1]

6.5 From patterns to sentential facts

For news, the "real-life" changes in a text corpus can be pinpointed using sentences. Due to differences in the expressiveness levels of story representations, comparing the patterns directly with sentences would be biased

[1] Examples of representations, methods, model-specific query generation, and the corpora can be found at `https://sites.google.com/site/subasicilija/ttm-evaluation`.

towards the patterns with sentence-like structure. Therefore, to make direct comparisons possible, we developed a process for obtaining the sentences that story elements best resemble to. This task is very akin to that of sentence retrieval, a task defined as follows: Given a query, rank sentences based on some measure of their similarity to that query. Our approach is therefore to transform the patterns into queries and then use sentence retrieval. Direct comparisons are then possible on the retrieved sentences.

The large number of developed sentences retrieval alorithms and the lack of a benchmark method for sentence retrieval in the TREC sentence-retrieval task make it difficult to decide which specific retrieval method to use. However, a detailed analysis of sentence retrieval [18] used Query-likelihood retrieval method (QL) with Jelinek-Mercer topic smoothing on a pseudo-document index of sentences as a baseline. Therefore, we consider the QL method a sensible choice for our framework. The inputs for this model are an index of pseudo-documents and a set of queries used for retrieving. An index is created from sentences of a document set, and queries are obtained from story representations of evaluated tracking methods.

Generating the pseudo-document index We first parse the complete document set, creating a set of sentences S. Then we store every sentence $s \in S$ as pseudo-document, creating an index I.[2]

Query generation for sentence retrieval For generating the queries used for sentence retrieval, we combine story elements. The combination greatly depends on the internal structure of the story representation and the relations between its story elements.

To obtain the same number of queries and to limit their length, we impose two parameters: maximum query length ($maxQ$) and maximum representation size ($maxR$). For every group, we consider two approaches. The first, *generic query generation*, is the same for all groups, and uses the top $maxR$ story elements from story representation, where the order used to determine the top elements is determined by the burst score. The second approach, *specific query generation*, takes into account the semantics of different story representations. The idea is to combine the basic story elements into more complex queries.

Specific query generation based on the output of keyword representation methods uses the following procedure. First, we extract the $maxR$ highest ranked story elements from the method's story representation. Then we combine them by creating all possible combinations not larger than $maxQ$.

[2]We used the Lemur Toolkit `www.lemurproject.org`.

We rank these newly formed combination based on the average burst scores of their elements and use the top $maxR$ as queries for retrieving sentences.

Group representation methods output groups of elements describing subjects. The procedure for query generation based on the story representation with i groups is as follows. For each group, we extract $maxR/i$ story elements with the highest in-group burst score and combine them into all possible combinations not larger than $maxQ$. Then we rank the new in-group element combinations based on the average burst scores of their elements. Then, for each group, we use the top $maxR/i$ combinations as queries for retrieving sentences. We assume that all groups are equally important and use $maxR/i$ story elements from each group. If an evaluated method gives preference to some groups, this could, in future work, be incorporated by choosing more elements from the more important groups.

For combo representations, we utilize the pattern structure of the specific method to determine $maxR$ elements of size $maxQ$. For example, the STORIES method that relies on graphs is a combo method [24]. This pattern structure arises from a skip-bigram network model; combined patterns are subgraphs with at most $maxQ$ edges, scored by the edges' average burst score.

Sentence retrieval procedure For every given method $M_\bullet$ and its story representation of documents of sentences S indexed in I, we generate a query set $Q_\bullet$ based on the generic query generation procedure, and a query set $Q^*_\bullet$ for the specific query generation procedure. Then, for every $q \in Q_\bullet$ and $q^* \in Q^*_\bullet$, we retrieve the top ranked sentence using the QL retrieval method. This creates two sets of retrieved sentences R and R^*.

6.6 Evaluation framework

Evaluation consists of comparing the sentences obtained by the procedure described in the previous section, with a set of ground-truth sentences. This requires a corpus (news-article documents and a ground truth), measures, and a procedure including a statistical test.

Corpus We divide the document set into time periods $T = \{t_1, t_2, \ldots, t_N\}$ of equal length. For each $t \in T$, we (a) build an index I_t as described in the previous section; and (b) obtain a set of ground-truth sentences $G_t = \{g_{t1}, g_{t2}, \ldots, g_{tN}\}$, where N is the index number of ground truth in t. G is the union of all these sets.

6.6.1 Evaluation measures, procedure and tests

We wish to capture novelty at the sentence (fact) level, and therefore we define the performance measure on the same level. We compare a set of retrieved sentences with set of ground truth sentences in a same time frame. As atomic-level measure, we adopt *ROUGE* metrics [16], and create a set of aggregate measures (X) to show (a) to what extent the retrieved sentences capture the ground truth ("recall-oriented" measures) and (b) how many sentences are needed to obtain a satisfactory ground-truth match ("precision-oriented" measures).

Atomic measures. For each $t \in T$, let $R_t = \{r_{t1}, r_{t2}, \ldots, r_{tK}\}$ be the set of retrieved sentences, and $G_t = \{g_{t1}, g_{t2}, \ldots, g_{tN}\}$ the set of ground truth sentences. For every retrieved sentence r_{tk} $(1 \leq k \leq K)$, we calculate the *atomic measures* $ROUGE.2$ and $ROUGE.SU4$ scores against every ground truth sentence in the same time period: g_{tj} $(1 \leq j \leq N)$. This will give us a Cartesian product of R_t and G_t, where each element has attached scores.

Aggregate measures. Based on this, we define *aggregate measures* to quantify how well a retrieved sentence set matches ground truth set, what percentage of the best possible ground truth match is obtained from the retrieved set, and how the number of retrieved sentences influences these scores. In the following, these three measures will be explained in their form based on $ROUGE.2$; their forms based on $ROUGE.SU4$ are directly analogous.

To find the best possible match in the set of retrieved sentences, we define the $maxM$ measure as the maximum score that any retrieved sentence in a period t has for the ground truth j:

$$maxM.2_{tj} = max_{r_{tk} \in R_t} ROUGE.2(r_{tk}, g_{tj}). \tag{6.1}$$

Since the ground truth sentences do not come from the same corpus as the retrieved sentences, it is hard, if not impossible, to obtain the maximum match of 1. The maximum $maxM$ score that a ground truth can obtain, varies from one ground truth to the other. So, in order to normalize the score, we introduce the $maxMR$ measures to capture how much of the possible maximum match between the ground truth and all sentences is captured in the retrieved sentences set.

We derive the maximum $ROUGE.2$ scores between G_t and S_t, where $S_t = \{s_{t1}, s_{t2}, \ldots, s_{tH}\}$ is the set of *all* sentences from t.

$$\begin{aligned} maxMR.2_{tj} &= (max_{r_{tk} \in R_t} ROUGE.2(r_{tk}, g_{tj}))/ \\ &\quad (max_{s_{th} \in S_t} ROUGE.2(s_{th}, g_{tj})).\text{„} \end{aligned} \tag{6.2}$$

$maxM$ and $maxMR$ measures give us "recall oriented" measures, telling us what is the proportion of the best possible match between the ground truth and the entire corpus obtained by the retrieved sentences. However, different methods may retrieve different numbers of sentences, and the ones with larger retrieved sentence set increase the chance of having a better match. We take this into account and define a new measures:

$$maxMP.2_{tj} = maxMR.2_{tj} * min(|G_t|, |R_t|)/|R_t| \tag{6.3}$$

$maxMP$ rewards the methods that produce a good fit with a small number of retrieved sentences (matching the usually small number of ground-truth sentences). In this sense, the measures correspond to "top heavy precision-oriented" measures like precision@k or discounted cumulative gain.

The motivation behind these metrics is to punish those methods which retrieve to many sentences and reward those that retrieve approximately the same number of sentences to the number of ground truth sentences.

Procedure and test In a cross-evaluation testing procedure, we compare the performance of different methods.[3] Given time periods T, methods M, ground truths G, metrics X, and a set of indices I: For every $t \in T$ and for every $M_\bullet \in M$, we calculate all three measures from the previous paragraph. This gives rise to, for every metric $x \in X$, $|M|$ sets $F_{mx} = \{f_{m11}, \ldots, f_{mtj}\}$ where j is a number of ground truths in a period t. For every x, we then test the results of different methods using Friedman's and Tukey's multiple comparison test [11].

6.7 Case study

To investigate our measures and how they behave in the different scenarios, we undertook a case study. We compared 3 TTM methods, one from each group discussed in Section 6.4, using 2 well-known news stories.

[3] An alternative would be to compare all of them to a baseline, but such a baseline would yet need to be defined.

Data. We refer to the corpora consisting of news article documents and ground-truth sentences as A and B, and to the methods as M_1, M_2, and M_3. We re-used the corpora from [24], see the reference for a detailed description of sources, features and construction. The corpora are available at `https://sites.google.com/site/subasicilija/ttm-evaluation`.

Methods. The question is how to choose the methods which both represent the groups described in Section 6.5 and are of high quality. We decided to use the methods that received the most attention by the community over the last few years.[4] Namely, for keyword representation we chose Kleinberg's burst detection algorithm (M_1) [12], for group representation, we used a probability mixture method developed by Mei and Zhai (M_2) [17], and for the combo group we used the STORIES method (M_3) [24].

Method M_1 [12] discovers bursty words by minimizing their state-transition cost between bursty and non-bursty states. The cost is defined as a lift in proportion of relevant document to an observed word in a data set. The method takes a scaling factor value as a parameter, which determines the lift intensity needed for word to reach the bursty state. We use the parameter value 2, as set in [12].

In the original paper, the algorithm was applied to scientific publications. The same method was later used to find bursts in blogs [14] and to track news [15]. We coded the algorithm following the description in the original paper.

An extension of probabilistic mixture method for topic discovery presented described in [27] was used in [17]. This method, M_2, outputs set of bursty topics represented by word distributions. Each of the different distributions points to a subject. The original papers reports on a number of parameters set on document-topic, topic-time, and word-topic distributions. We used the same values as reported by the authors. However, the authors report on a threshold value set by empirical testing which is not described in detail. We were therefore not able to replicate this, and manually set the threshold to 5 subjects ("topics"). We coded the algorithm employing a modification of the implementation of [27] in DRAGON NLP toolkit. [5]

M_3 [24] defines burstiness via a skip bigram's co-occurrence frequency normalized by the count of documents. A skip bigram is bursty if this frequency in a period exceeds a lift threshold relative to the overall frequency. Skip bigrams are created by combining unigrams as "story basics". A story

[4] By no means we are arguing that the chosen methods are the best among all possible methods, nor trying to diminish the quality of other work.

[5] `http://dragon.ischool.drexel.edu/features.asp`

representation is built by joining skip bigrams into a graph ("story graph"). In the original paper, the 150 most frequent terms (not in the stopword list) were used to extract skip bigrams. We keep this settings and add another 3 choices of story basics: (1) the 150 top-tf.idf terms; (2) 100 most frequent terms plus 50 most frequent named entities (with no overlap); and (3) all terms from the document collection. We refer to the different versions of the method as M_3^{tf}, $M_3^{tf.idf}$, M_3^{ne}, and M_3^{all}, depending on the story basics used in graph creation. We modified the original code accordingly.

Procedure We re-used the period partitioning first used with these corpora. Corpus A was split into 18 periods, and corpus B into 10 periods. Corpus A contained 24 and corpus B 12 ground-truth sentences.

As discussed in Section 6.5, for each method we generated two sets of retrieved sentences, with generic and with specific query generation. This yields 12 sets (2 for both M_1 and M_2, and 2 for every version of M_3). We set the query-generation parameter $maxQ$ to 5, and we varied $maxR$ from 5 to 30 in increments of 5. The value of $maxQ$ was chosen based on query length in major search engines [8]. Variations of the second parameter simulate the situation in which different number of top story elements are used. The query generation process for M_1 and M_2 follows the procedure described in Section 6.5, while for M_3 we first extract all the paths up to size $maxQ$ from the story graph, and then sort them based on the average edge weight, following the evaluation procedure described in [24].

We calculated $maxM$, $maxMR$, and $maxMP$ for all 12 settings and for both $ROUGE.2$ and $ROUGE.SU4$, and tested them as described in Section 6.6, resulting in a total of 72 tests.

Results: Overview The large number of test settings and methods made it difficult to aggregate the results of the tests, so we created multiple aggregations. With these aggregations we aimed at capturing the following information: (a) which methods are robust to different $maxR$ settings, (b) whether there is a difference between "precision-like" and "recall-like" performance, and (c) whether specific query generation process improves retrieval.

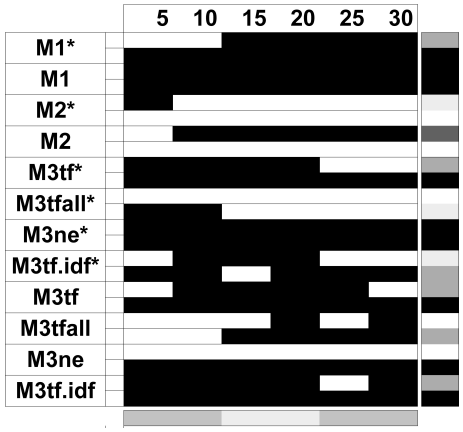
Results: Test settings First, we investigated how different test settings influence the results of the methods. Figure 6.1 shows the results for $ROUGE.2$. The results for $ROUGE.SU4$, not shown for reasons of space, were very similar; in particular, the order of method quality was the same. We

restrict ourselves to $maxMR$ (Fig. 6.1 (a)) and $maxMP$ (b). The reason is that the $maxM$ results were similar to $maxMR$, with the difference that $maxM$ was, as expected, due to the different numbers of ground-truth sentences less robust over different number of $maxR$. Each row shows a combination of one method and one corpus; grouped by methods to highlight similarities over corpora. The first row is for corpus A and the second for corpus B. Rows without an asterisk show the results of the method with generic query generation, rows with an asterisk show the results for specific query generation. Columns show test settings: the use of only the top $maxR$ story elements or story-elements combinations for sentence retrieval.

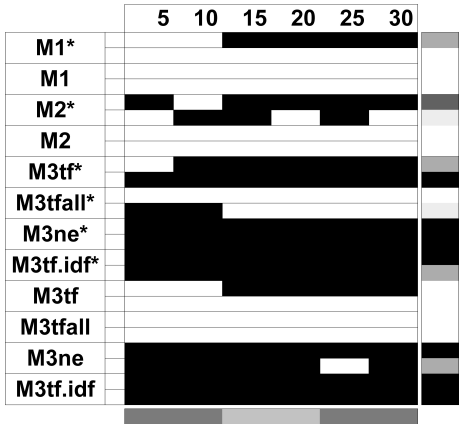
A cell is black if, according to the Friedman/Tukey test, this method-setting combination was in the group with the best results of the measure, i.e. there was no statistically significant difference in results between all the methods with a black cell in one column, but all of these significantly outperformed all of the others (the ones with a white cell in this column). Further significant differences between these lower-quality methods existed, but are not shown.

The "heatmap" under the table shows how much this setting differentiates between methods, ranging from low (= all methods are the same, light grey) to high (= only a small fraction of methods were in the best group, black). Differentiation is measured by the total number of white cells. The heatmap to the right of the table shows the robustness of the method quality over settings, ranging from low (= only in good group for few settings, light grey) to high (= always in good group, black). Differentiation is measured by *number of blocks - (number of white cells + number of block holes)*. Values for both heatmaps were binned into four categories of grey shades.

Figure 6.1 (a) and (b) show that the setting M_3^{ne} performs the best for both A and B, having only 3 settings in which they are not in the top group. Slightly less compact blocks are created by M_3^{tf} and $M_3^{tf.idf}$. This can be explained by the nature of A and B, which are both stories revolving around persons, such that tracking named entities makes more sense. Story basics in the $tf.idf$ (tf) based set overlap 72% (69%) with the named-entity set. M_3^{all} performs worse than the top group in most cases; for corpus A even in all cases. Thus, choosing story basics for M_3 has an influence on the results. For M_1, performance improves as $maxR$ rises, suggesting that capturing bursts with keyword lists needs lists of certain size. The M_2 results vary the most of all 3 methods, and they are much better for A than for B. We believe that the cause for this is the "group" story representation that M_2 uses. Each group should point to one of the events, but the number of ground-truth sentences for B is small and averages at 1.56 (A has on average 2.72 events) for a time period. This means that there more groups than subjects.



((a)) $maxMR$



((b)) $maxMP$

Figure 6.1: Top group method comparison matrix of $maxMR$ (a), and $maxMP$ (b) measures based on $ROUGE.2$.

Results: "Precision" and "recall" The "precision-oriented" measure $maxMP$ shows that the settings M_3^{tf} and $M_3^{tf.idf}$ have the same results as M_3^{ne}. This suggests that the difference in overall results comes from the "recall" oriented measures. The results of M_1 show that the method performs well for the recall oriented measure $maxMR$, being in the best category in over 80% of the settings, and poorly for precision oriented measures, being in the best category in only 22% of the settings. This clearly shows that M_1 outputs patterns that are related to the ground truth, but also some that are the results of the changes in language use, vocabulary, etc. over time. For M_2, specific query generation clearly improves the results.

Results: Internal pattern structure for query generation We also investigated whether specific query generation improves the results. Observing patterns from Fig. 6.1 reveals that settings that use specific query generation have, on average, a larger number of black blocks for M_1 and M_3, while for M_2 specific query generation diminishes "recall" results by 20% (in block counts), while it slightly improves precision by 6.1%. However, that figure only differentiates between a method being in the top group or not. We wish to directly compare settings with and without the use of specific query generation procedure.

Figure 6.2 investigates the effect of query generation procedures. Each row describes the same basic method (e.g., M_3^{tf} with 150 top keywords in row 3) with or without combination, for a given corpus. Columns are the same as in Fig. 6.1. A cell is black if the method with combination significantly outperformed its counterpart without combination (regardless of whether any of them was in the top group or not), white if it significantly underperformed the latter, and grey if there was no significant difference.

Figure 6.2 shows that for the "recall" oriented measure, specific query generation does not much improve the results. It improves them in 11.1% of the settings for $maxMR$, while it diminishes them in 8.3% settings. This suggests that methods output sentences with the best match to the ground-truth sentences without specific query generation. This is not the case for the "precision-oriented" measure: for M_1 and M_3, specific query generation always improves the results. This is to be expected due to the nature of the story representation these methods use. Both the keyword representation and the group representation have little structure built into them, and the specific query generation combines the story elements so that the queries become more similar to the top weighted story elements in the story representation. More similar queries limit the number of retrieved sentences and preserve the "good" ones. As for M_3, the specific query generation never diminishes the results,

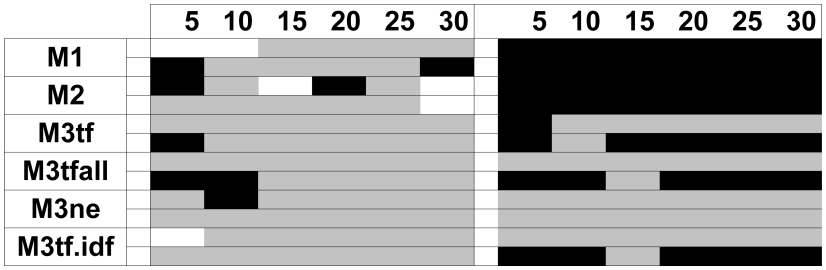


Figure 6.2: Query generation comparison for different test settings.

and for B it improves the results in almost all settings, except for $M3_{ne}$. The difference in the performance over different corpora most likely comes from the different number of ground truth sentences. If the story graph produced by M_3 is highly connected, the generated queries are more alike resulting in retrieving same sentences. The correlation of graph topology and number of events for M_3 was presented in [3].

These results indicate that specific query generation in most cases does not affect the results. In some cases, notably for M_1 and M_2 precision metrics, the specific query generation always performs better. This indicates that the retrieval process benefits from the use of specific query generation. One possible way of improving this process for M_3 would be to take into account the similarity of already extracted queries, and in that way include more diverse queries rather than the ones with the highest weight.

Results summary. The main results of our case-study experiments are: (1) method M_3^{ne} is the most robust method to test settings changes; (2) keyword representation methods need larger $maxR$; (3) M_3 variations outperform M_1 and M_2 in "precision-oriented" measures; and (4) specific query generation improves "precision-oriented" results, especially, for methods M_1 and M_2.

6.8 Conclusions and outlook

This research started with the aim of creating an evaluation procedure for evaluating different temporal text mining approaches. We concentrated on news stories, where changes over time can be discovered through "sentential" facts. An inspection of existing evaluation frameworks showed that most slightly differ from our aim of investigating how different patterns match to the ground-truth sentences. Therefore, the paper first proposed a process

which connects patterns and sentences, and then evaluates the results against an editor-created ground truth. Following the experience in the similar fields we defined measures that capture both the "recall" and "precision" oriented performance of the methods. We presented a case study evaluating 3 methods over 2 corpora. The results suggest that the method presented in [24] is the most robust of all tested methods. The research also shows that using bursty patterns' internal structure for connecting them with ground truth sentences improves the results.

The evaluation framework and the query generation procedure presented here have some limitations, and overcoming them will be a challenging task for future research. First, a creation of larger, cleaner, and editor-annotated data sets comprising different stories would help avoid generalization errors. Second, we devised our performance measures based on text summarization frameworks, which are geared towards comparing longer text sequences. We miss a clear notion of how the scores generated by our measures transfer into "real-life" similarity between retrieved sentences and ground truth sentences. Therefore, we did a cross-method evaluation that, while suggesting which methods perform better than which others, cannot tell how "good" these matches really are. Thus, creating a baseline for evaluation would be an useful addition to the framework. Comparing against a baseline would simplify the testing procedure, and it would give clues as to whether extracting temporal patterns outperforms non-temporal patterns for creating story representation.

Different settings in which different TTM methods build and evaluate their results are not always compatible. Even with taking this into account, parts of our framework favor some methods in certain settings. However, we believe that any bias is distributed over the methods, not systematically favoring any of the evaluated methods. We consider this paper to be just a first step into defining a widely accepted test settings for testing TTM methods, and by no means consider our framework as the only solution for TTM evaluation. One of the messages we hope to have conveyed is that a larger effort by the community is needed to create a set of unified settings for evaluating different methods.

References

[1] James Allan, Rahul Gupta, and Vikas Khandelwal. Temporal summaries of news topics. In *SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9–13, 2001, New Orleans, Louisiana, USA*, pages 10–18. ACM, 2001.

[2] James F. Allan. *Topic Detection and Tracking*. Springer, Berlin etc., 2002.

[3] Bettina Berendt and Ilija Subasic. Measuring graph topology for interactive temporal event detection. *KI*, 23(2):11–17, 2009.

[4] Gabriel Pui Cheong Fung, Jeffrey Xu Yu, Philip S. Yu, and Hongjun Lu. Parameter free bursty events detection in text streams. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment, 2005.

[5] Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors. *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21–24, 2005*. ACM, 2005.

[6] Daniel Gruhl, Ramanathan V. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The predictive power of online chatter. In Grossman et al. [5], pages 78–87.

[7] Qi He, Kuiyu Chang, Ee-Peng Lim, and Jun Zhang. Bursty feature representation for clustering text streams. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26–28, 2007, Minneapolis, Minnesota, USA*. SIAM, 2007.

[8] HitWise, Inc. Hitwise: Google's search share continues to grow. `"http://www.bizreport.com/2009/05/hitwise_googles_search_share_continues_to_grow.html"`.

[9] Matthew Hurst. Temporal text mining. In *AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*, 2006.

[10] Frizo A. L. Janssens, Wolfgang Glänzel, and Bart De Moor. Dynamic hybrid clustering of bioinformatics by incorporating text mining and citation analysis. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12–15, 2007*, pages 360–369. ACM, 2007.

[11] M.G. Kandell. *Rank correlation methods.* Oxford press, 2076.

[12] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7:373–397, October 2003.

[13] Dan Knights, Michael Mozer, and Nicolas Nicolov. Detecting topic drift with compound topic models. 2009.

[14] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 568–576, New York, NY, USA, 2003. ACM.

[15] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506, New York, NY, USA, 2009. ACM.

[16] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 71–78, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[17] Qiaozhu Mei and ChengXiang Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In Grossman et al. [5], pages 198–207.

[18] Vanessa Murdock and W. Bruce Croft. A translation model for sentence retrieval. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 684–691, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[19] National Institute of Standarts and US Departmant of Commerce Technology. Duc 2007:task, documents, and measures, 2007. `http://www-nlpir.nist.gov/projects/duc/duc2007/tasks.html#pilot`.

[20] Soma Roy, David Gevry, and William M. Pottenger. Methodologies for trend detection in textual data mining. In *Proceedings of the Textmine '02 Workshop*, Washington, DC, USA, 2002. SIAM.

[21] Rene Schult and Myra Spiliopoulou. Discovering emerging topics in unlabelled text collections. In *Advances in Databases and Information Systems, 10th East European Conference, ADBIS 2006, Thessaloniki, Greece, September 3–7, 2006, Proceedings*, volume 4152 of *Lecture Notes in Computer Science*, pages 353–366. Springer, 2006.

[22] David A. Smith. Detecting and browsing events in unstructured text. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '02, pages 73–80, New York, NY, USA, 2002. ACM.

[23] Ian Soboroff and Donna Harman. Novelty detection: the TREC experience. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 105–112, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[24] Ilija Subasic and Bettina Berendt. Discovery of interactive graphs for understanding and searching time-indexed corpora. *Knowl. Inf. Syst.*, 23(3):293–319, 2010.

[25] Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 784–793, New York, NY, USA, 2007. ACM.

[26] Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, New York, NY, USA, 2006. ACM.

[27] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 743–748, New York, NY, USA, 2004. ACM.

Errata

- Section 6.1, page 168, paragraph 1: *a lack of common procedures* should be the lack of common procedures;
- Section 6.1, page 168, paragraph 1: *an evaluation framework for methods assessment* should be an evaluation framework for method assessment;
- Section 6.1, page 168, paragraph 1: *different classes of temporal text mining show* should be different classes of temporal text mining algorithms show;
- Section 6.3, page 170, paragraph 2: *the automatic evaluation using ROUGE framework* should be the automatic evaluation using the ROUGE framework;
- Section 6.3, page 170, paragraph 2: *Most commonly used ROUGE scores* should be The most commonly used ROUGE scores;
- Section 6.3, page 171, paragraph 1: *and asked to make a true/false decisions on an event* should be and asked to make true/false decisions on various events;
- Section 6.4, page 171, paragraph 3: *an element could be a term, and the pattern this term plus some score* should be an element could be a term, and the pattern could be a term plus some score;
- Section 6.5, page 173, paragraph 1: *story elements best resemble to. This task is very akin to* should be story elements best resemble. This task is akin to;
- Section 6.5, page 173, paragraph 2: *analysis of sentence retrieval [18] used Query-likelihood retrieval method* should be analysis of sentence retrieval [18] used the Query-likelihood retrieval method;
- Section 6.5, page 174, paragraph 2: *if an evaluated method gives* should be if an evaluated method were to give;
- Section 6.6.1, page 175, paragraph 3: *Based on this, we define* should be Based on the atomic measures, we define;
- Section 6.6.1, page 175, paragraph 3: *three measures will be explained in their form based on ROUGE.2* should be three measures will be explained, specifically their form based on ROUGE.2;
- Section 6.6.1, page 176, paragraph 1: *telling us what is the proportion of the best possible match between the ground truth and the entire corpus obtained by the retrieved sentences* should be telling us what the proportion

of the best possible match between the ground truth and the entire corpus obtained by the retrieved sentences is;

- Section 6.6.1, page 176, paragraph 3: *punish those methods which retrieve to many sentences and reward those that retrieve approximately the same number of sentences to the number* should be punish those methods which retrieve too many sentences and reward those that retrieve approximately the same number of sentences as the number;
- Section 6.7, page 176, paragraph 5: *how they behave in the different scenarios* should be how they behave in different scenarios;
- Section 6.7, page 177, paragraph 3: *The cost is defined as a lift in* should be The cost is defined as the lift in;
- Section 6.7, page 177, paragraph 6: *The original papers reports* should be The original paper reports;
- Section 6.7, page 181, paragraph 1: *some that are the results of the changes* should be some that are the result of the changes;
- Section 6.8, page 183, paragraph 1: *the experiences in the similar fields we defined* should be the experiences in similar fields, we defined:
- Section 6.8, page 183, paragraph 2: *First, a creation of larger* should be First, the creation of larger.

Chapter 7

Interactive Evaluation of Interfaces for Story Tracking

Ilija Subašić and Bettina Berendt: Interactive evaluation of interfaces for story tracking. HCIR 2011: The Fifth Workshop on Human-Computer Interaction and Information Retrieval (October 2011). Electronic publication, no pages.
(https://sites.google.com/site/hcirworkshop/hcir-2011/posters)

Contributions as first author:

- (a) Co-defining the evaluation framework;
- (b) Implementing the user interfaces and result processing;
- (c) Conducting and administering the study;
- (c) Co-interpreting the study results.

7.1 Abstract

Users often follow the the same news story over an extended period of time and repeatedly search for the documents on the same topic (story tracking). In this paper we investigate how to evaluate document-search interfaces in this scenario. We first define a novel evaluation task and measures for interactive story tracking evaluation. Then, we present a study of four document-search interfaces based on graph and textual search functionality. The results of the study show that users slightly prefer the graph-based interfaces.

7.2 Introduction

Most systems providing access to online news regard news search as an ad-hoc activity in which a user at one point in time searches for documents relevant to the news story of his interest. However, news stories span over an extended period of time continuously attracting user's attention. We refer to the activity in which user searches for the news belonging to the same story over time as *story tracking*. For example, a user follows all news reports on elections every week form the campaign start to government creation. The task of users in this activity is to discover not only relevant but also novel information about the story they explore.

Various approaches to (news-)story tracking led to the development of different research areas like update summarization and temporal text mining. In this paper, we focus on the users and investigate document-search interfaces for story tracking. With a number of interfaces used for story tracking at users disposal, a natural questions arises: How to evaluate the performance of these interfaces? In this paper we tackle this problem, and propose a set of tasks and measures (framework) to evaluate document-search interfaces for story tracking. To the best of our knowledge there is a lack of work in cross-evaluation of document-search interfaces in the context of story tracking. Many tasks related to information retrieval have been approached with interaction in mind [4], and several of these tasks have been assessed using standardized evaluation frameworks [2]. However, all of these disregard the temporal dimension of corpora – and thus are not fully appropriate for interactive story tracking evaluation. We transform the story tracking task into a fact finding and summary creation task. On one side this is similar to the fact finding tasks of the HARD track [1], but we provided less focused and broader description of the topics and include search over several time periods.

Apart from the general problem of evaluating story tracking interfaces (Section 7.3), in this paper we investigate a specific question regarding our previously developed story tracking interface [5] (see Section 7.4 for the investigated interfaces, and Sections 7.6 and 7.5 for the experiment). The core of this system is a graph-based visual document-search interface that we call *story graphs* and a temporal pattern extraction algorithm. We test these components by creating the interfaces based on only one component. In addition, we include a standard search-box search interface to serve as a baseline interface. In total, in our study we evaluate four interfaces – two graph-based and two text-based. The results of the study show that participants found graph-based interfaces most suitable to the task and express a personal preference towards them. In addition the analysis of user activity suggests that users are more engaged while exploring stories with graphs-based than with text-based interfaces.

7.3 Task and Measures Framework

7.3.1 Task description

We framed the story tracking task as a fact finding and summary creation task. While reading news, users search for the most important bits of information about the events, topics, and subjects news story revolves around. In most cases this information is represented with sentences (often referred to as facts), e.g. *Uruguay won the Copa America for a record 15th time after beating Paraguay 3-0.* By reading news users discover facts and collection of these facts is in a way a summary of the story. Therefore, we frame the task as the discovery of the most informative sentences in a period; the collection of informative sentences is considered a summary. We defined informative sentences as ones best describing the time period they belong to. To simulate story tracking the task was solved in three consecutive time periods. Task repetition over time allows us to explore how the interfaces facilitate users in discovery of both novel and relevant information.

7.3.2 Metrics

We defined five sets of measures to capture the various aspects of interfaces for story tracking. The first two are observation-based and quantify the quality of human produced summaries (summary quality measures) and the extent of user activity while creating these summaries (user activity measures). The next

two sets, interface and task measures, are self-report-based and are based the on user's perception of the interfaces and the task. Finally, the comparative measures capture the direct comparison of the interfaces based on user rankings.

Summary quality measures. We quantify the summary quality using measures we previously defined in [6] for automatic story tracking evaluation. In short, these metrics are based on ROUGE scores [3] used in automatic summarization evaluation. The scores are calculated between a set of user selected and ground-truth sentences. Ground-truth sentences are those selected by human judges as being informative in a period. We use two ROUGE scores $S2$ (bigram overlap) and $SU.4$ (skip-4 bigram overlap) and define the summary quality in the following way: per user for an interface, for each ground truth sentence calculate the maximum $S2$ or $SU.4$ scores among all selected sentences; then average over these maxima; and calculate the mean over all users. This yields $(S2/SU4)_x$ measures, where $x = (all, last)$ is a period range for which the scores are averaged. The *all* measures capture the performance in all periods, and the *last* measures only in the last time period. The measures take values from $[0 - 1]$ interval. The value of one would mean the participants selected the sentences that are most similar to the ground truth sentences.

User activity measures. Activity measures show the level of user engagement in story tracking. We wish to investigate whether different interfaces provide an incentive for corpus exploration. In total we defined four activity measures. First two explore the queries participants used; query length – number of words in a query, and query number – number of issued queries. We also tracked the number of accessed documents – document count. The last measure – exploration time, measures minutes from the start of the search until the summary completion.

Interface measures. We asked participants to rate on a five-point Likert scale the following interface aspects: quick scan of the documents, exploration of the different aspects of the document set, and discovery of relevant documents.

Task measures. We also measured several task-oriented aspects to learn how users perceive the given task. The participants rated on five-pointed Likert scale the interfaces on how tedious and overwhelming the task was, and how participants perceive their success in solving the task.

Comparative measures. All four previously described sets of measures are defined for a single interface. Additionally, we asked participants to rank the interfaces based on four criteria: easy to learn, easy to use, suitable to solving the task and personal preference (likeness).

7.4 Interfaces

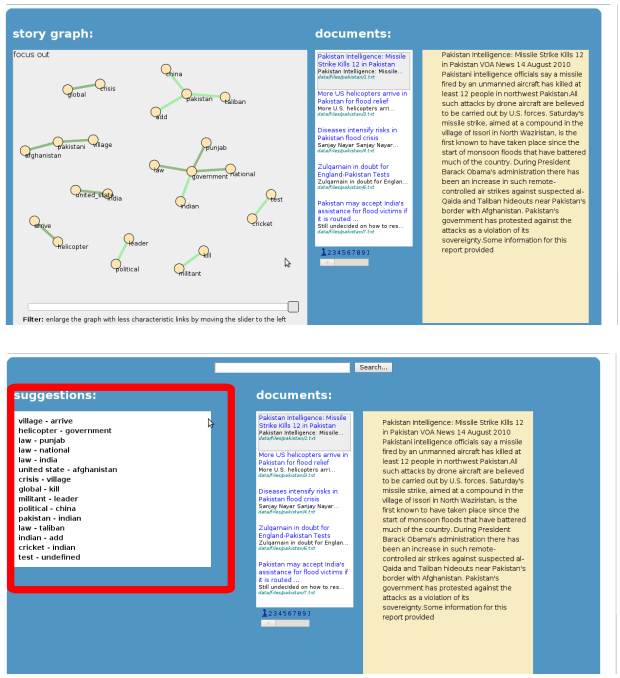


Figure 7.1: Screen-shoots the interfaces – column (a) *graph-based interfaces* *STORIES* and *GRAPH* and column (b) *text-based interfaces* *SUGGEST* and *S.BOX*. *SUGGEST* and *S.BOX* are identical except from the highlighted area (marked with a red rectangle) which is visible only in *SUGGEST*.

We were particularly interested to evaluate our previously developed story tracking interface [5] (*STORIES*) based on graph visualization we call story graphs. Users interact with the story graph by selecting edges they find relevant. These edges are then transformed into a query by concatenating the names of nodes they connect. The system returns a set of relevant documents for this query. The main components of our interface are temporal patterns which we use as *suggestions* and visual *graph-based* search. To test these components we create additional into two interfaces: *GRAPH*, a graph-based visual search interface based on non-temporal patterns and *SUGGEST*, a text-based search using temporal patterns presented as a list. We add a standard key-word based interface (*S.BOX*) as a baseline interface. Figure 7.1 shows the interfaces we tested. Please note that *GRAPH* and *STORIES* visually

look the same, and *S.BOX* and *SUGGEST* differ only in the presence of a bi-gram list (marked by red rectangle in the Figure).

Generating suggestions and graphs. We generate two types of suggestions, temporal and non-temporal. First, we extract the *content-bearing terms* – defined as the 150 top-TF.IDF terms. The suggestions are generated for a time period t, and for each document in the period the *frequency* of the co-occurrence of all pairs of content-bearing terms in a window of w words is calculated. We normalize this with the number of documents in a period to obtain *local relevance* (LR). This is the basis for non-temporal suggestion in *GRAPHS*. For temporal suggestions used in *STORIES* and *SUGGEST*, local relevance is normalised by its counterpart in the whole corpus to yield *time relevance* (TR).

Once we obtain the TR or LR for all bi-grams, we create graphs. Content-bearing words become nodes connected with edges with TR/LR weight. To avoid singular associations in small sub-corpora we apply thresholds on a co-occurrence number and set a minimum TR. Finally, we sort the edges by weight and keep the first 30.

7.5 Study Method

Participants. In total we recruited 24 participants (9 female) using a student forum. As an incentive for participation we offered a 20 euro voucher for a retail store. Participants had a wide range of study directions, and most (16) were PhD or master students.

Materials. The corpora we collected consists of two parts – the document set and the ground-truth sets. The document set is divided into stories which are the sub-corpora discussing the same news themes. The theme is regarded as a higher level news story (e.g. earthquake in Japan). In total, we compiled four stories covering: Britney Spears, Greek debt crisis, BP oil spill, and Chile miners accident. A ground-truth set is created for each story, and contains editor-selected sentences describing important developments in the story. It was created using Wikipedia entries on the stories. The same method of corpora creation was applied in our previous work [6, 1]. Participants explored the document sets using the four interfaces described in Section 7.4.

Design. We used within-subject design and each user was assigned a unique ordering of the interfaces. We counterbalanced the order in which the interfaces were presented to the users by generating all possible orderings of the four interfaces. There was no reason to expect an effect of the order of the stories;

therefore, this was kept constant. This produced a total of 24 orderings in which participants tested the interfaces. Each participant was given an unique ordering of interfaces. Using a single interface a participant always explored the same news story.

Procedure. The participants were asked to read the task instruction and solve the task described in Section 7.3. Before using each interface, participants were provided with a tutorial followed by a test run of the interface. For each period participants were presented with the interface and the initial document list sorted on date and relevance. The time limit of seven minutes per period was set following a pilot study in which participants completed the task without the time limit. After seven minutes participants could finish reading the document they were reading and select sentences from it, but not search for nor read other documents. Before the study started participants filled in a demographics pre-study questionnaire, and after each interface they filled in a post-system questionnaire. At the end of the study (after the fourth interface) participants were given an exit questionnaire in which they were asked to compare all the interfaces they used. Together with the exit questionnaire participants were presented with the screenshots of the interfaces in the same order as they were used during the study.

Participants were tested in individual sessions lasting about two hours each. After the second interface test was completed there was a 15 minute break. The participants worked with a Mac mini computer using a 17 inch monitor with the web based software and documents residing on a server.

7.6 Results and Discussion

The value for all summary quality measures (Figure 7.2(a)) was always highest for the graph-based interfaces (*GRAPH* and *STORIES*). We tested the differences using the Kruskal-Wallace test (due to the non-normal distribution used instead of ANOVA) with Tukey's HSD correction for multiple comparisons. However, no statistically significant differences in summary quality were found among interfaces. The analysis of user activity measures (Figure 7.2(b)) shows that on average participants used more and longer queries with *GRAPH* and *STORIES*. The longest queries (avg. 2.33) were issued with *STORIES* and the shortest (1.7) with *S.BOX*. We found the differences to be statistically significant (at $p<.05$) between all interfaces except between *GRAPH* and *STORIES*. With the average of 10.4 queries per period *STORIES* engaged participants in issuing most queries. The fewest queries were issued using *S.BOX* (4.84). The differences between *S.BOX* and

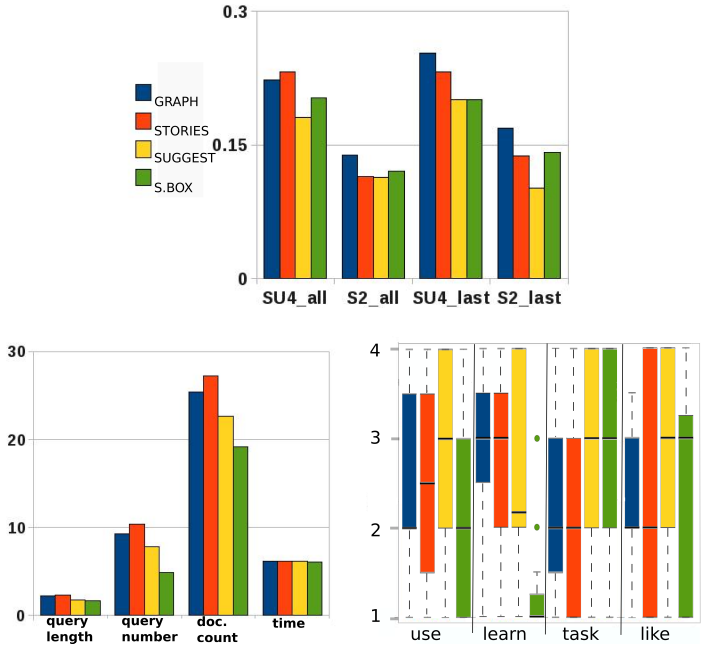


Figure 7.2: Study results for the observed measures, columns: summary quality (a) and user activity (b); and comparative measures (column (c)).

all other interfaces were statistically significant. As expected, issuing more queries results in accessing more documents and with *STORIES* participants on average read around 27 documents. The only statistically significance difference was between *STORIES* and *S.BOX*. We found no differences among interfaces in exploration time.

The results for self-reported measures in Table 7.1 show (the results are on a Likert scale, but we assumed equal distances between responses and calculated the mean) that users found the graph-based interfaces to provide for an easier access to the relevant documents and exploration of aspects in a story. Participants felt that the task is slightly more tedious to complete with graph-based interfaces. However, the differences between interfaces were not significant for any measure.

The results of interface comparison (Figure 7.2(c)) show that participants ranked *S.BOX* as the easiest for using and learning. On the other two criteria users ranked *GRAPH* and *STORIES* as the most fitting to the task of the study and expressed personal preference towards them. All four criteria were

Table 7.1: Average values of self-reported measures.

	GRAPH	*STORIES*	*SUGGEST*	*S.BOX*
quick scan	3.76	3.32	3.11	3.39
aspect explor.	3.62	3.32	3.06	3.13
relevant docs.	3.81	3.55	3.22	3.35
overwhelming	2.95	2.91	2.56	2.17
tedious	2.67	2.86	2.83	2.65
successful	3.81	3.5	3.5	3.7

tested using Friedman test with Tukey's HSD. The differences in criteria easy to learn between *S.BOX* and all others are significant ($p<.05$), while there were no significant differences between ranks of other interfaces. For the criteria easy to use, the only significant difference of ranking is between *S.BOX* and *GRAPH*. Both for criteria fitting to the task and personal preference, there is a significant difference in ranks of *GRAPH* and *STORIES* and the ranks of *SUGGEST* and *S.BOX*.

Discussion. For all interfaces, we found no significant differences in summary quality measures. The observed low performance of participants is most likely the cause of the lack of differentiation between the interfaces. We speculate that the reason for a low performance is the complexity of the task. Creating a summary is a job more suited to professional journalists than regular web users. Nevertheless, we discovered several valuable insights into interfaces for story tracking. First, we observed that with time summary quality improves. In the last periods, summary quality is higher than in the first two by around 20% for *GRAPH* and *STORIES*. In addition, we found that *GRAPH* and *STORIES* engaged participants in deeper exploration of the document set as they issued more queries, longer queries and accessed more documents compared to other interfaces. All interfaces except *S.BOX* include query suggestions, and queries can be issued without typing. This provides an easier way of querying, and the study shows that users will use facilitated querying (clicking rather than typing) if provided.

For self-reported measures, we found no statistically significant differences, but the analysis of the direct comparison of the interfaces shows that users expressed higher preference for using graph-based interfaces and ranked these two interfaces as the most fitting to the task of the study.

Looking from the perspectives of our interfaces, the biggest differences between *SUGGEST* and *STORIES*/*GRAPH*. This suggests that out of graph representation and suggestion generation components, the first one is a more important one. This points that temporal patterns are not more useful than

non-temporal ones for graphs, but it could also be the case that low performance on the tasks accredits to this. As this study does not follow the full factorial design this finding should be taken having this in mind. In most criteria the standard search-box interface is outperformed by graph-interfaces.

7.7 Conclusions and Outlook

In this paper we described a framework for evaluating user interfaces for story tracking – following the developments in news stories over time. We framed story tracking as a fact finding and summary creation task and defined a set of measures for evaluating interfaces for this task. Based on this we undertook a study of four interfaces, three based on our previous interface and one baseline interface. We found no differences in the quality of the task solutions as measured with the summary quality measures. However, we found that graph-based interfaces allow for more active and engaging exploration of corpora. In addition, participants expressed higher preference towards using graph-based than towards using text-based search interfaces. Although text-based interfaces are still the most widely used interfaces for search today, we were able to show that graph-based interfaces can provide for a different, and for some users better, search experience while not diminishing the task performance.

We recognize a number of limitations of this work. It is not clear whether the task we defined is a "good" simulation of story tracking. We asked participants to rate on a five point scale how similar the defined task is to the way they follow news, and the average rating was just over three (neutral). The study should be replicated with more users. Although users had not seen the interfaces prior to the study, text-based interfaces resemble most commonly used search interfaces. Thus, some bias towards these interfaces was expected, as shown by user perception on learning and operating difficulty.

Overcoming these limitations offers a number of possible extension of this work. An interesting extension is to investigate further tasks to better simulate story tracking. A meta-study of different tasks such as question answering, focused search, and fact checking could provide some answers to this question. Another valuable extension is the definition of a baseline interface scores for story tracking.

In this paper we aimed to define a reusable framework for the evaluation of story tracking interfaces. To improve on this we rely on a community feedback and agreement on the baseline interfaces, measures, and tasks.

References

[1] J. Allan. Hard track overview in TREC 2004 - high accuracy retrieval from documents. In *Proc. TREC '04.*, pages 1–17. NIST, 2004.

[2] S. T. Dumais and N. J. Belkin. The trec interactive tracks: Putting the user into search. In *Proc. TREC '05.*, pages 22–31. NIST, 2005.

[3] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *In Proc. NAACL '03*, pages 71–78. ACL, 2003.

[4] I. Ruthven. Interactive information retrieval. *Annual Review of Information Science and Technology*, 42(1):43–91, 2008.

[5] I. Subašić and B. Berendt. Discovery of interactive graphs for understanding and searching time-indexed corpora. *Knowl. Inf. Syst.*, 23(3):293–319, 2010.

[6] I. Subašić and B. Berendt. From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In *Proc. ECAI'10*, pages 517–522, 2010.

Addendum A

In this addendum we include the instructions users were given after prior to the test, and the post-system and exit questionnaires users filed out after each system and at the end of the study.

Interfaces for News Tracking User Study

User instructions

In this experiment we are investigating different interfaces for news exploration on the Web.

Your goal is to investigate news stories and - with the help of different document-search interfaces - select the most informative bits of the reported news.

You will participate in 4 studies, each consisting of 3 parts. All studies have an identical task, but use different search interfaces and different news stories. The 3 parts of each study are identical except for the publishing time of the documents (the parts are ordered chronologically). The total time of the study is estimated to be 1,5 - 2 hours.

Please take time to read these instructions, and feel free to ask the study administrator if something is not clear or if you have any questions.

1. The study administrator will provide you with a tutorial of all interfaces included in the tests. You are free to use the tutorials for as long as you see necessary to get acquainted with the interfaces.

2. Once you are ready to start, the study administrator will set up the first study. You now have 7 minutes to explore the documents and summarize the news. Do not hurry, but try to work fast and efficiently and to produce a good summary. You may finish the task before the 7-minute timeout if you consider it done.

3. Use the search interface to explore documents describing the story (the *Info* button on the top of the page provides information about the news story documents relate to). Similarly as in web search engines (e.g. Google or Yahoo), you can query the documents to find the ones relevant to a specific keywords. Depending on the interface, the way in which you can express queries will differ.

4. Read as many documents as you wish. Once you find an informative sentence (or a part of a sentence) in a document, copy it to the text-box at the top of the screen. Press "+" button to add a sentence or "-" button to remove it. Regard a sentence as informative if it describes a new event or development in the main story.

5. For each part of the test, you are allowed to save at most 5 sentences. You do not have to add sentences for each document or each of your queries. At any point you can remove previous sentences and add new ones.

6. When selecting sentences keep in mind not to select multiple sentences with the same information (describing the same event). Try to find unique sentences for each part of a study.

7. At any moment you can decide that there are no more sentences you wish to add. After 7 minutes, the administrator will ask you to finish reading and progress to the next part of the study.

8. Once you are done with one part of the study, press the *Next* button on the top of the page to proceed to the next part (please remember to press *+* to add the last sentence in the text box).

9. When you have completed all 3 parts of a single study (there is text on the screen confirming this), ask the study administrator to set up the next study. In the meantime you will be asked to fill in a questionnaire. In total, you need to do 4 studies. After the first 2 studies, take a 10-15 minute break.

10. After completion of all 4 studies, please fill in the questionnaire about the interfaces in the study.

All the collected data will be treated confidentially and used only for this study. The identity data will not be used for the study, and it is collected for administrative reasons. By starting the first part of the test, you are agreeing on these terms.

Thank you for participating.

Interfaces for News Tracking User Study
Participant ID: _____
Interface ID: _____

POST-SYSTEM QUESTIONNAIRE

The following questions relate to your experiences using the interface.

INSTRUCTIONS: Please rate how strongly you agree or disagree with each of the following statements by circling the appropriate number.

strongly disagree	**disagree**	**neutral**	**agree**	**strongly agree**	
1	2	3	4	5	I was **familiar** with the topic of the story.
1	2	3	4	5	I found that the interface enabled me to **quickly scan** the corpora.
1	2	3	4	5	I found that I was able to **explore every aspect** related to the story.

strongly disagree	**disagree**	**neutral**	**agree**	**strongly agree**	
1	2	3	4	5	I found it **easy to find relevant** documents using the interface.
1	2	3	4	5	I found that using the interface was **overwhelming**.
1	2	3	4	5	I found completing the task using the interface to be **tedious**.
1	2	3	4	5	I think I was able to **complete the task successfully**.

If you have any other comments about the interface or the study procedures, please write them below.

Interfaces for News Tracking User Study
Participant ID: _____

EXIT QUESTIONNAIRE

The following questions relate to your experiences using the interfaces for news search during this study.
INSTRUCTIONS: Please rank the interfaces you used based on the following criteria. To rank write the letter of the interface in order you saw them (A-D) (you can ask the administrator to show you the interfaces again). Rank 1 is the best rank. If you want to assign the same rank to two interfaces please write the same letters in both cells.

1	2	3	4	
				Please rank the 4 interfaces in order of how **easy to use** they were.
				Please rank the 4 interfaces in order of how **easy to learn** they were.
				Please rank the 4 interfaces in order of **how fitting to solving the tasks** they were**.**
				Please rank the 4 interfaces in order of **which system you liked best**.

INSTRUCTIONS: Please rate how strongly you agree or disagree with each of the following statements by circling the appropriate number.

strongly disagree	disagree	neutral	agree	strongly agree	
1	2	3	4	5	I found the **task easy to understand**.
1	2	3	4	5	I found the task to be **similar** to my regular news search activities.

Please write the answers to following questions:

1. What were the main differences between the task and how you normally search for and read news?

2. My usual online service for news reading is (circle the most appropriate):
   1. news portals and search engines (like Google News);
   2. RSS aggregators;
   3. Twitter, Facebook and other social network feeds;
   4. news websites;
   5. other (please specify):_____

If you have any other comments about interfaces, your news search activities, or the study procedures, please write them below.

Addendum B

Due to the page number we did not include screenshots of the interfaces we tested in the original paper. In this addendum we include the the screenshots of interfaces $STORIES1$ and $GRAPH$ in Figure 7.3, interface $SUGGEST$ in Figure 7.4, and interface $S.BOX$ in Figure 7.5. All three schreenshot provided access to a story about *Pakistan floods*, we used in the tutorials shown to the participants in the test.

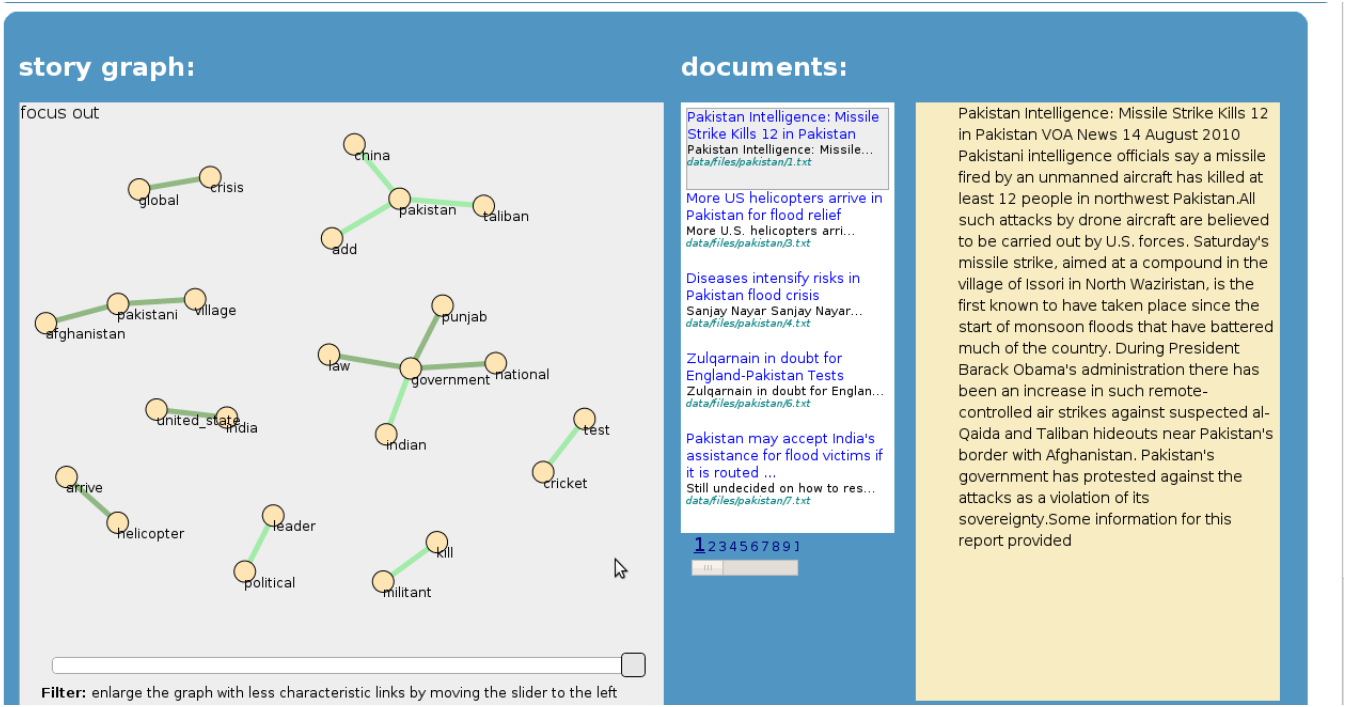


Figure 7.3: Screenshoot of the graph-based interfaces *STORIES* and *GRAPH*. On the top of the screen is the "fact pane" used for sentence selection. Note that both interfaces visually look the same, but differ in the way graph on the right is created.

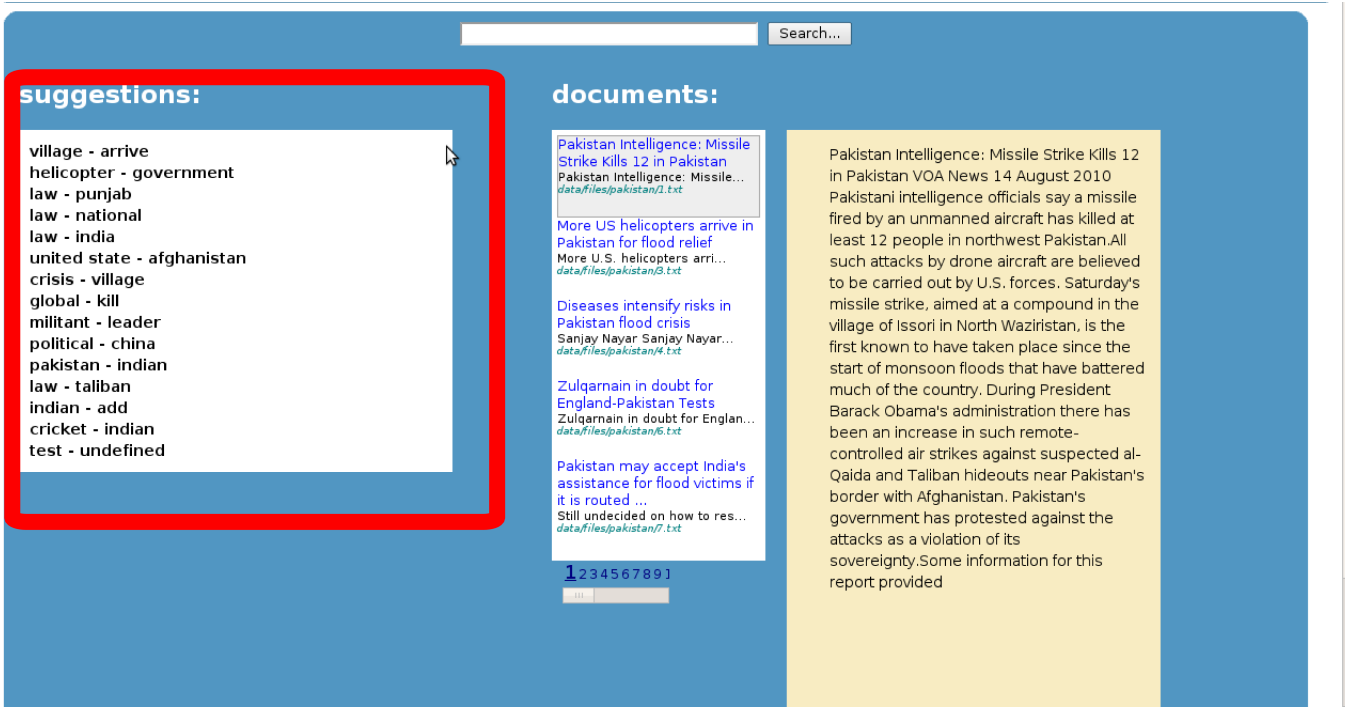


Figure 7.4: Screenshoot of the text-based interfaces with suggestions – *SUGGEST*. The suggestions are shown on the left pane of the interface.

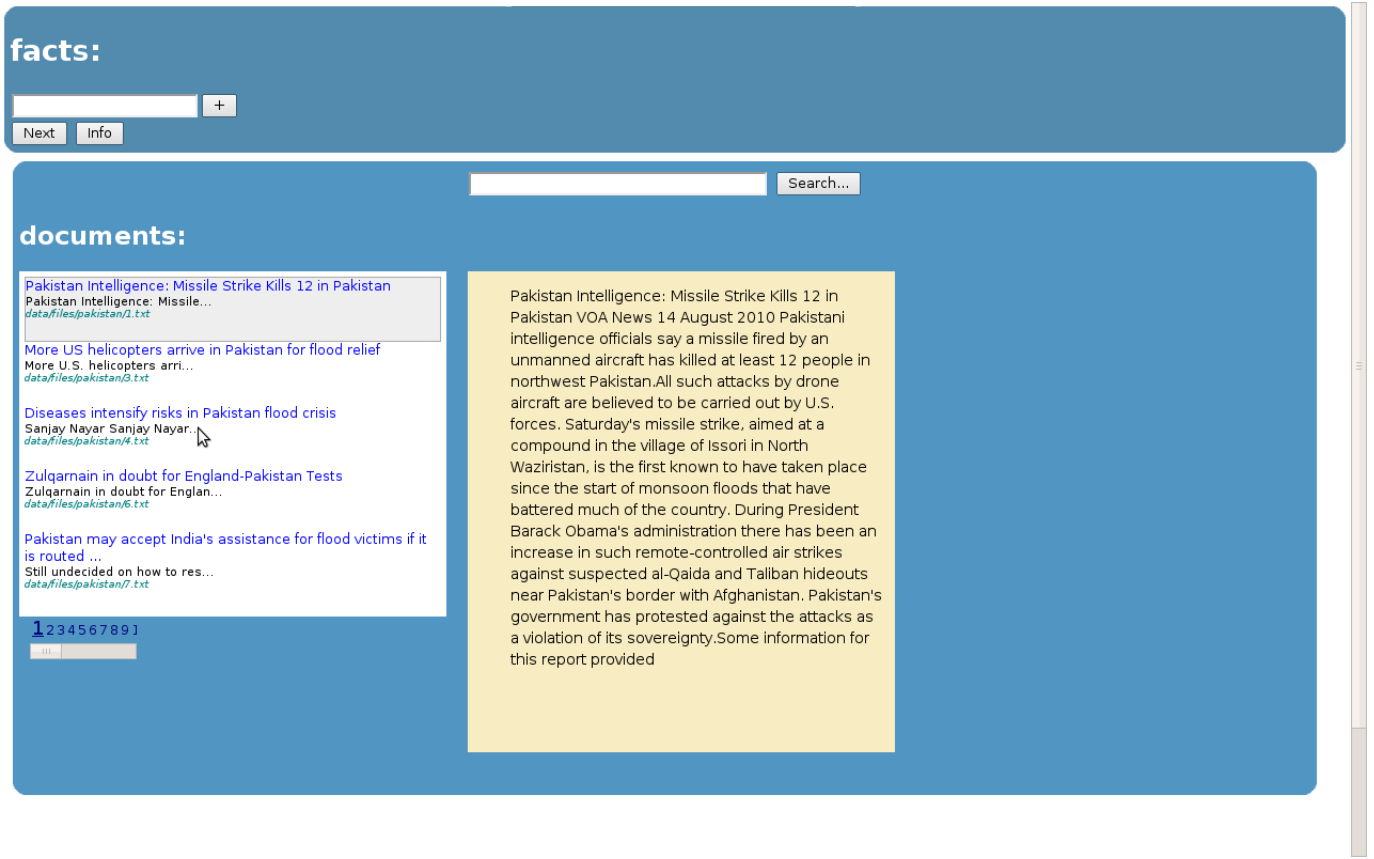


Figure 7.5: Screenshoot of the text-based interfaces with suggestions *S.BOX*.

Errata

- Section 7.1, page 190, paragraph 1: *developments in a news story* should be developments in the news story;
- Section 7.2, page 190, paragraph 2: *every week form the campaign start to government creation* should be every week from the campaign start to government formation;
- Section 7.2, page 190, paragraph 2: *information about the story they explore* should be information about the story they are exploring;
- Section 7.2, page 190, paragraph 3: *areas like update summarization* should be areas such as update summarization;
- Section 7.2, page 190, paragraph 3: *a natural questions arises* should be a natural question arises;
- Section 7.2, page 191, paragraph 1: *interface that we call story graphs* should be interface called story graphs;
- Section 7.4, page 193: *a standard key-word interface* should be a standard keyword interface;
- Section 7.5, page 195, paragraph 2: *read the task instruction* should be read the task instructions;
- Section 7.5, page 196, paragraph 2: *provide for an easier access* should be provide for easier access;
- Section 7.5, page 197, paragraph 2: *the reason for a low performance* should be the reason for the low performance;
- Section 7.5, page 197, paragraph 2: *the biggest differences between SUGGEST and STORIES/GRAPH* should be the biggest differences were between *SUGGEST* and *STORIES/GRAPH*;
- Section 7.6, page 198, paragraph 1: *accredits to this* should be accredits to no differences between temporal and non-temporal patterns.

Chapter 8

Investigating Diversity in News Sources

Ilija Subašić and Bettina Berendt. Peddling or creating? Investigating the role of twitter in news reporting. In Proceedings of the 33rd European Conference on Advances in Information Retrieval (ECIR'11). Springer-Verlag LCNS, Berlin, Heidelberg, 207-213. 2011.

Contributions as first author:

- (a) Related work section;
- (b) Similarity framework definition;
- (c) Conducting the case study;
- (d) Co-interpretation of the case study results.

8.1 Abstract

The widespread use of social media is regarded by many as the emergence of a new highway for information and news sharing promising a new information-driven "social revolution". In this paper, we analyze how this idea transfers to the news reporting domain. To analyze the role of social media in news reporting, we ask whether citizen journalists tend to *create* news or *peddle* (re-report) existing content. We introduce a framework for exploring divergence between news sources by providing multiple views on corpora in comparison. The results of our case study comparing Twitter and other news sources suggest that a major role of Twitter authors consists of neither creating nor peddling, but extending them by *commenting* on news.

8.2 Introduction

On January 16, 2009, a US Airways airplane made an emergency landing on the Hudson river. First reports on this events were spread via social media web sites. Although the idea of "citizen journalism" was present much before this incident took place, it has provided a major boost to citizen journalism platforms, placing them shoulder to shoulder with "traditional" news outlets. By some [14], this new way of discovering news is hailed as a beginning of a "social revolution" driven by information sharing, promising stronger social action and making *vox populi* a more important factor in all spheres of life. However, some researchers [8, 6] have expressed doubts about such a social-media-led revolution. Transferring the same principles and contrasting standpoints to the news reporting domain, one could expect that social media either have a great potential for introducing and spreading new information, or alternatively serve solely as a channel for spreading the content produced by traditional media. Thus, is the (main) role of citizen journalists to *create* news or rather to *peddle* (re-report) existing content? In this paper, we aim to provide some insight into this question by defining a set of corpora-similarity measures on corpora created from Twitter and other news sources. The main idea of using corpora similarity is that higher similarity would suggest "peddling", while lower would suggest originality and "creation".

There has been substantial research into discovering the point of origin of a news story [7] and into the dynamics of content between news and blogs. We take a different approach: we start with news story already "discovered" and investigate whether social media provides a *different* reporting to traditional media. We start by collecting corpora containing documents on the same news story originating from different sources. Our goal is to analyze differences

between corpora by providing a multi-aspect view on similarity. Some news stories describe breaking events or spotlighted topics. These stories are often referred to as "breaking news". We broaden our analysis and investigate whether during a breaking event reporting converges across sources.

The main contribution of this paper is a framework for comparing social media with traditional media that provides: (a) multiple-aspect corpora difference measures, (b) analysis of social vs. traditional media content, and (c) aggregation and visualization of news sources relations. We complement the framework (Section 8.4) by a case study (8.5).

8.3 Related work

Twitter research. Out of many areas of Twitter research, we focus on the ones related to news mining. [9, 16, 8] investigate user motivation behind twittering. All of these studies report on news sharing as one of the main motivations for Twitter use. Studies of the role of news medium in influence spread [16, 3] found that traditional news sources and celebrity-owned Twitter accounts were among the most influential posts. In contrast to these works, our objective is to detect the differences between news reports covering the same story on Twitter and other media.

Corpora and text similarity. Similarity between texts has been a long-standing topic in different fields producing a wide range of text similarity measures. [2] provides a valuable overview of different text similarity measures. Work in corpus linguistics [10] compares text similarity metrics on a corpus scale. This work introduces a χ^2-test based model of corpus similarity and compares it with the probabilistic similarity measures perplexity [5] and mutual information [4]. Another family of probabilistic similarity measures, based on Kullback-Leibler (KL) divergence [11], has been widely used in different domains as a measure of text similarity. It is used for measuring similarity between queries and documents in information reterieval [12], for detecting plagiarism in Wikipedia articles [2], and for comparing traditional and Open Access medical journals content [15]. We adopt a KL-based approach to corpora similarity, but provide multiple perspectives on similarity by combining several aspects of the corpora.

8.4 Measures of corpora divergence

Notation. A corpus C^{story}_{source} is a set of documents covering the same news *story* collected from a single *source* (e.g. Twitter, AP) or a family of sources (e.g. blogs). A representation of a corpus C^{story}_{source} is a language model Θ^{story}_{source}, where the probability of a token t is denoted as $\Theta^{story}_{source}(t)$. We define token categories as: (1) plain words (pw) - words in a document; (2) headline words (hw) - words in headlines; (3) entity words (ew): words referring to a semantic entities (names, locations, companies, ...); and (4) sentiment words (sw): words expressing sentiment. $\Theta^{story}_{source|category}$ denotes the category language model (e.g. for a unigram model, $\Theta^{story}_{source|hw}$ are the probabilities of headline words).

Divergence measures. Among many different language models we choose the unigram model as it fits the writing style of tweets. Given two corpora from sources a and b covering a story x, we use a symmetrical variant of KL divergence, the Jensen-Shannon divergence (JS), between their language models Θ^x_a and Θ^x_b to measure their distance as:

$$JS(\Theta^x_a, \Theta^x_b) = \frac{1}{2} KL(\Theta^x_a, \Theta^x_m) + \frac{1}{2} KL(\Theta^x_b, \Theta^x_m); \tag{8.1}$$

where the probability of every t in Θ^x_m is the average probability of t in Θ^x_a and Θ^x_b. We define a set of measures differing by token categories.

Language divergence (LD). The first measure we define covers the entire content of the corpora. In other words, we build a language model using pw. The reason for this is to capture stylistic, terminological, and content differences of sources.We define language divergence (LD) of two sources a and b reports on a story x as:

$$LD^{a,b}_x = JS(\Theta^x_{a|pw}, \Theta^x_{b|pw}). \tag{8.2}$$

Due to many differences in the format and length between documents between corpora, using this measure mostly captures differences in writing styles and vocabulary between sources.

Headline divergence (HD). Headlines in traditional news summarize their articles; they are a standard unit of analysis in media studies. Tweets have no substructure and are at most 140 characters long, making them their own headlines ($hw = pw$). We define the headline divergence as:

$$HD^{a,b}_x = JS(\Theta^x_{a|hw}, \Theta^x_{b|hw}). \tag{8.3}$$

Using headlines to measure difference between reports in social and traditional media tackles problems of style and length used among sources. However, it still does not take into account the semantic difference between reports.

Named-entity divergence (ND). News stories revolve around different subjects, places, and organizations they describe or "feature". We introduce a semantics divergence measure as:

$$ND_x^{a,b} = JS(\Theta^x_{a|ew}, \Theta^x_{b|ew}). \tag{8.4}$$

Named entities carry semantically rich information conveyed by the reports, but fail to capture the position of the reporters towards the story. News texts are often more than reporting, and express opinions and sentiments towards the story.

Sentiment divergence (SD). We therefore define a last measure based on the differences in used sentiment words. Since many words used to express sentiment are rarely used and the probability of observing them in a corpus is low, we follow the approach described in [1] and bin sw tokens into 7 categories of strength and type of the sentiment they express (ranging form strong negative to strong positive). Therefore, SD measures differences in probability distributions over the categories of sentiment, and not sentiment-bearing words:

$$SD_x^{a,b} = JS(\Theta^x_{a|bin(sw)}, \Theta^x_{b|bin(sw)}). \tag{8.5}$$

To be able to relate more than two sources to one another and to abstract from the (non-interpretable) absolute values of JS, we apply multidimensional scaling, projecting the obtained distance matrices into two dimensions.

8.5 Case study

We present a case study comparing news reports from Twitter (tw), blogosphere (bl), professional news outlets (nw), Reuters (rt) and Associated Press (ap).

Corpora and procedure. We obtained the story reports using the same query across different sources' search engines. For news and blogs, we used Google News and Blog search to harvest web pages, and extracted content as described in [13]. We collected 3 breaking stories covering the BP oil spill, the Pakistan floods in 2010, and the Chilean miners' accident, and 3 non-breaking stories about Belgian politics, Iraq, and the European Union. As an indicator of breaking stories we used Twitter's trending topic list. The upper bound of corpus size was the number of tweets for the respective story, and the lower bound was set to 50. For collected documents we extracted named entities with Open Calais (www.opencalais.com), removed stop words, and lemmatized the rest.

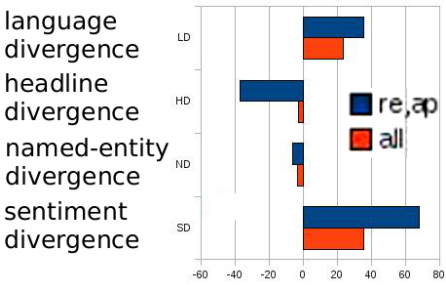


Figure 8.1: Average RD for all stories comparing divergences between **Twitter** and other sources with *all* and re, ap baselines.

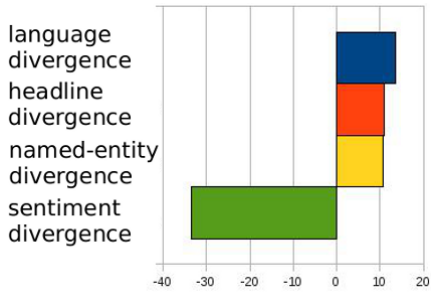


Figure 8.2: Average RD for **breaking stories** with the *non-breaking stories* baseline.

Relative divergence aggregation. The absolute values of KL based measures have no clear interpretation; we therefore concentrated on values relative to a baseline. We defined 3 baseline divergences. The first one (all) averages over all pair-wise divergences across all sources. The second base divergence value (re, ap) is the divergence between Reuters and Associated Press corpora. Due to the same type of media, format, and reporting style, we consider this as a reasonable baseline. To compare breaking and non-breaking stories, we used the average divergence for non-breaking stories over all sources as a baseline. We denote these 3 value as $baseline_{(all;re,ap;breaking)}$. The relative divergence (RD) of a source a for a story x is then:

$$RD_a^x = (\ (avg_{b \in sources} D_{a,b}^x - baseline_x)\ /\ baseline_x\ )\ \times 100. \tag{8.6}$$

Results. Figure 8.1. shows the results of applying RD to Twitter. We start with the interpretation of the re, ap baseline. The largest relative divergence is for sentiment divergence. The RD value of 67.87 shows higher sentiment divergence between Twitter and other sources. This result can lead to two conclusions: (a) Twitter contains more contrasting sentiment than news-wire reports, and (b) Twitter expresses more sentiment than news-wire reports. To decide betweeen these two, we calculated the share of sentiment words across sources. In ap corpora, there are 1.7% sentiment words, in re corpora 2.8%, and in tw corpora 4.2%. We find that both the share and the type of sentiment words influence the differences between corpora. For example, strongly negative words make up 0.17% of the ap and 0.9% of the tw corpora. For other categories, it is expected that language divergence (LD) has a positive value, because more authors in Twitter use different writing styles, wording,

and non-standard grammar. This is partly shown by the average number of unique language-words tokens (10221 in *ap*, 13122 in *re*, and 89619 in *tw*). The high positive value of HD can be explained by the differences in the sizes of the different corpora, where a small number of documents in news-wire agencies do not converge to the same headline words. On average for a story, we collected 69 documents from news-wire agencies and 3314 from Twitter. The lowest RD value (-6.13) is found for named-entity divergences. This points to the conclusion that all sources are reporting on the same entities, but using different language and sentiment.

The difference between average RD values for the *all* baseline is always lower than for the re, ap baseline, on average $\approx 19\%$ lower. The extreme case is the difference of headline divergence values, which is 12.9 times lower when using the *all* baseline. We see this as an effect of having more documents to compare tweets to, because the average number of documents across all sources is 451, and due to many similar titles the divergence measure converges across sources. The lower RD values for *all* compared to re, ap suggest that Twitter is more similar to other sources than to news agencies.

Figure 8.2 shows RD values that describe the difference between breaking and non-breaking news. As a baseline divergence, we used the *non-breaking* value, comparing the average of the *breaking* stories to it. Positive values of RD reflect a higher divergence of reporting for breaking news. The figure 9.2 shows that breaking news is consistently more different across sources, except for sentiment divergence (SD). This suggests that for breaking news, informing the readers about the story is the main objective of the authors, while for non-breaking stories authors express their standpoints and analysis of the story.

To further investigate these differences and see which divergence contributes most, consider the MDS plots in Figure 8.3. Figure 8.3(a)shows that the headline distance between reports in News and Twitter is the lowest. Many news-related tweets come from Twitter accounts operated by professional news outlets [3]. In our dataset, we found an average of 2.9% identical entries in the *tw* and *nw* corpora. Figure 8.3(b) shows that Twitter uses language closer to news and blogs than news-wire agencies, while news and blogs use similar language when compared to other sources. In terms of named entities (Figure 8.3(c)), *re* corpora are far from other sources. This probably arises from the much number of named entities used by Reuters: an average per-document of 21.9 (compared to 0.22 in *tw*, 8.6 in *bl*, 12.91 in *nw*, and 13.2 in *ap*). Figure 8.3(d) visualizes sentiment divergence, showing that *bl*, *re*, and *nw* corpora contain similar amounts of sentiment, which are more different when compared to *tw* and *ap* corpora.

8.6 Conclusions and outlook

This work is our starting effort in defining an easily interpretable, multi-aspect similarity measures for comparing news sources. Of course, this work cannot cover all the possible or interesting aspects of divergence in news reports, and absolute values of divergence measures are hard to interpret. Nonetheless, as the paper has shown, the inspection of relative differences can give interesting insights, opening many interesting research directions.

We started this investigation by focusing on two roles of social media platforms: to create new and different news, or to peddle or spread existing news. In contrast to both, our results suggest that the biggest role of citizen journalists in news is the role of a *commentator*, not only reporting but expressing opinions and taking positions on the news. Investigating this role will yield further measures of relations between corpora and a deeper understanding of the dynamics of social media in today's news environment.

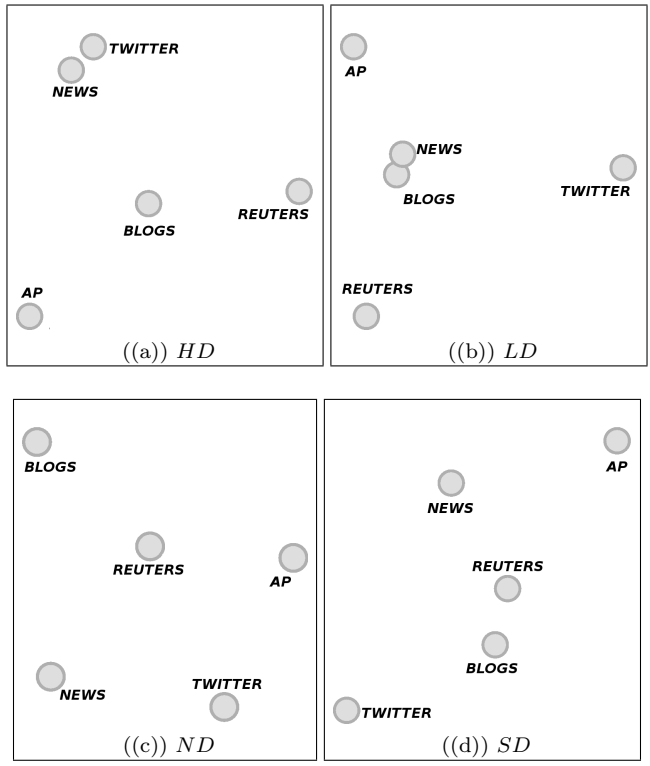


Figure 8.3: MDS maps of divergence measures: (a) HD, (b) LD, (c) ND, (d) SD .

References

[1] Andrea Esuli Stefano Baccianella and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

[2] Alberto Barrón-Cedeño, Andreas Eiselt, and Paolo Rosso. Monolingual Text Similarity Measures: A Comparison of Models over Wikipedia Articles Revisions. In Dipti Misra Sharma, Vasudeva Verma, and Rajeev Sangal, editors, *In proceedings of the workshop on Comparing Corpora, held in conjunction ACL 2000. October 2000, Hong Kong*, pages 29–38, Hyderabad, India, 2009. Macmillan Publishers.

[3] Meeyoung Cha, Hamed Haddadi, Fabrício Benevenuto, and P. Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM*, 2010.

[4] Kenneth W. Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

[5] Peter E Brown, Vincent J. Della Pietra, Robert L. Mercer, Stephen A. Della Pietra, and Jennifer C. Lai. Entropy Of English – An estimate of an upper bound for the. *Computational Linguistics*, 18:31–40, 1992.

[6] Malcolm Gladwell. Small change: Why the revolution will not be tweeted. *The New Yorker*, 86(31), October 2010. web only, http://www.newyorker.com/reporting/2010/10/04/101004fa_fact_gladwell.

[7] Lev Grossman. Iran protests: Twitter, the medium of the movement. *Time Magazine*, jun 2009.

[8] Bernardo A. Huberman, Daniel M. Romero, and Fang Wu. Social networks that matter: Twitter under the microscope. *Social Science Research Network Working Paper Series*, Dec 2008.

[9] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, New York, NY, USA, 2007. ACM.

[10] Adam Kilgarriff. Comparing corpora. *International Journal of Corpus Linguistics*, 6:1–37, 2001.

[11] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[12] John Lafferty and Chengxiang Zhai. Probabilistic relevance models based on document and query generation. In *Language Modeling and Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2002.

[13] Jyotika Prasad and Andreas Paepcke. Coreex: content extraction from online news articles. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1391–1392, New York, NY, USA, 2008. ACM.

[14] Clay Shirky. How social media can make history, Jun 2009. http://www.ted.com/talks/lang/eng-/clay_shirky_how_cellphones_twitter_facebook_can_make_history.html, TED talk.

[15] Karin Verspoor, K. Bretonnel Cohen, and Lawrence Hunter. The textual characteristics of traditional and open access scientific journals are similar. *BMC bioinformatics*, 10(1):183+, 2009.

[16] Dejin Zhao and Mary B. Rosson. How and why people twitter: the role that micro-blogging plays in informal communication at work. In *GROUP '09: Proceedings of the ACM 2009 international conference on Supporting group work*, pages 243–252, New York, NY, USA, 2009. ACM.

Errata

- Section 8.2, page 212, paragraph 2: *By some [14], this new way* should be Shirky [14] hails this new way;
- Section 8.2, page 212, paragraph 3: *social media provides a different reporting to* should be social media provides different reporting to;
- Section 8.2, page 213, paragraph 2: *complement the framework (Section 8.4) by a case study (8.5)* should be complement the framework (Section 8.4) with a case study (8.5);
- Section 8.3, page 213, paragraph 4: *[2] provides* should be Barron-Cedeno et al. [2] provide;
- Section 8.5, page 215, paragraph 5: *Twitter(tw), blogosphere (bl)* should be Twitter (tw), the blogosphere (bl);
- Section 8.5, page 216, paragraph 1: *We denote these 3 value as* should be We denote these 3 values as;

Chapter 9

Conclusions and outlook

9.1 Thesis Summary

In this thesis we investigated stories in online news spaces. We defined stories as sets of theme-related and time-stamped documents, and story spaces as channels that allow access to stories. Our main focus was on investigating stories along their temporal dimension. Specifically, we explored story tracking – an activity in which a user follows the same story over an extended period of time. While engaged in story tracking, users are interested in discovering the novel information about the developments in the story. Among many approaches to story tracking, this thesis investigated temporal text mining methods. These methods aim to discover the developments in a story by extracting bursty patterns from corpora. We developed a method for extracting graph-based bursty patterns we named story graphs, and built a tool based on user interaction with the story graphs visualization. In addition we developed frameworks for automatic evaluation of bursty patterns and interactive evaluation of user interfaces in the context of story tracking. Apart from investigating the temporal dimension of stories, we also investigated several other dimensions. First, we investigated whether and how users react to the developments in a story. We analysed a one year query log from a large scale industry search engine, and explored the differences in user behaviour before, during, and after an increase of interest in a story. Finally, we investigated stories from the aspects of story sources. With the growing body of news published via social media, we framed the investigation of story sources as an investigation of differences between content of social and traditional media. We developed a framework for comparison of news corpora, and a tool for

divergence based corpora exploration.

In Chapter 1 we defined the following 6 research questions we investigated in this thesis:

- *Q1*: How are search engines affected by story developments?
- *Q2*: Does the semi-automatic story tracking approach we developed enable user comprehension and navigation of stories?
- *Q3*: Can the graph-based patterns extracted by our algorithm be used for story tracking?
- *Q4*: How can different bursty text patterns be used for discovering origins of the changes in document sets?
- *Q5*: How do users interact with interfaces for story tracking?
- *Q6*: How to measure differences between a story across different sources?

During the course of work on this thesis we developed algorithms, defined evaluation frameworks, built systems, analyzed usage data, and conducted user studies in order to provide some answers to the above questions. We summarize the answers in the following list:

- *A1*: We discovered that an increased frequency of queries (query bursts) is correlated with changes in user behaviour during the search. The results of our analysis show that users spend more effort on and are more concentrated during the search when developments in a story occur.
- *A2*: User studies of the interactive story graph visualization showed that: (a) users were able to map edges to development descriptions; and (b) users were able to discover story developments. We discovered that structuring the story based on story graphs presents users with a more coherent structure of the documents, when compared to some of the state-of-the-art algorithms for web search results clustering.
- *A3*: We discovered that certain graph properties of story graphs point to the emergence of new developments in a story. Additionally, we showed that story graphs can be transformed and used to pin-point the original development representation.
- *A4*: The evaluation of TTM methods in the news domain, framed as sentence retrieval task showed that our method performs the same as or better than other methods we evaluated.

- *A5*: The user study of document-search interfaces in the context of story tracking showed that, in our experiment, users are more engaged in search with the interfaces that provided some guidance for search direction. In addition, when compared with the alternative interfaces evaluated in the study, the participants expressed higher preference towards interfaces based on story graphs.
- *A6*: The study of divergence between content of social and traditional media revealed that social media differ from traditional media mostly in the strength of expressed opinions and sentiment.

In Chapter 3 we investigated query bursts – periods of intensified user interest in a search engine query. We analysed query logs from a large scale search engine. Our motivation for this analysis was to discover whether users change their behaviour when a development in a story occurs. Our assumption is that query bursts are caused by developments in a story. We defined measures for user activity, effort, and concentration during web search. Then we compared these measures for periods before, during, and after a query burst. The results of this comparison show that during the query burst users are willing to spend more effort on web search, and that their interest is more concentrated on a specific section of search results. Using these measures we identified 3 distinct sets of query bursts. Then, we looked at query bursts from the content-provider aspect with the goal of exploring how content-providers can benefit from the increased query frequency. We presented a logistic regression model which estimates the click share a content-provider may expect to gain from this increased attention of the users.

In Chapter 4 we described our method for extraction of bursty patterns from evolving corpora. We devised the story graphs method for graph creation based on the co-occurrence of words in documents. Graphs outputted by our method present a visual summary of the developments in the story. To evaluate the usefulness of this summary to humans we conducted two user studies. In the first study we presented the participants with a set of story graph visualizations summarizing the developments in several periods and a set of development descriptions for the same periods. The task given to participants was to match the edges of the graphs to the descriptions of the developments. In the second study participants were presented with a set of interactive story graphs (with search functionality) and a set of YES/NO questions – descriptions of the developments in the period summarized by a story graphs. The task was to explore the documents using the story graph and answer the questions. Additionally, we looked at the coherency of document clusters created by using story graph paths as restrictions on a corpus. We compared these clusters with the ones produced by state-of-the art web search result clustering algorithms.

The results show that using story graphs to create structure in document sets leads to a more coherent document grouping.

In Chapter 5 we explored how story graphs support automatic tracking of story evolution. Our focus was on automatic detection of the emergence of new developments, and discovery of the development representation. If story graphs summarize the developments in a period of a story, then the intensity of changes in the properties of story graphs should point to the emergence of novel developments. We analysed local (node-level) and global (graph-level) properties of story graphs. We looked at properties related both to the size and the connectives of story graphs, as well as the intensity of changes between story graphs that summarize consecutive periods. The results of our analysis revealed that some of the properties we explored correlate with the number of new developments. Apart from detecting the emergence of new developments, we also investigated how to link story graphs to the original development representation. In this thesis we explored online news, and in this domain developments are usually represented in form of sentences. Therefore, we devised a procedure for development discovery based on the transformation of bursty patterns into queries used for sentence retrieval. We tested three different families of methods and the results of the evaluation point out that our method is comparable with the state-of-the art TTM methods.

In Appendix A we described *STORIES*, the visual document-search tool we developed based on the methods described in Chapters 4 and 5. The tool allows users to track the changes in story graphs, discover the important facts, and learn about a story by navigating through its document-space.

Chapters 6 and 7 described the evaluation frameworks for automatic and interactive evaluation of story tracking. Work presented in Chapter 6 overlaps with the work in Chapter 5 on the discovery of developments, but we provide a deeper motivation and an additional analysis of the methods. We defined a set of measures for evaluating sentences retrieved by TTM methods against an editor-selected set of sentences describing developments in the story. Apart from the results shown in Chapter 5, in Chapter 6 we evaluated query generation procedures. The results of this evaluation show that specific query generation procedures improve the results of sentence retrieval in almost every experimental setting.

In Chapter 7 we compared different interfaces in the story tracking context. We conducted a user study in which participants were asked to compile a 5 sentence summary of the documents they explored using different document-search interfaces (including ones based on story graphs). This summary was then compared with an editor-selected one. The results of this study show that the system based on story-graph visualization performs better or the same as

others, while users express higher preference towards story-graph interfaces.

Finally, in Chapter 8 we described the framework for news corpora comparison. We then applied this framework and compared news content of Twitter with other, more traditional, news media. The framework we developed compares news corpora along 4 dimensions, and we found that Twitter differs from traditional media mostly in the strength of sentiment and opinions its reports contain.

9.2 Limitations

Although we carefully planned and conducted the work on this thesis, we had to make a number of assumptions about the problems we explored. Due to the data at our disposal, lack of standardized evaluation frameworks, and user availability we recognize a number of limitations of the work presented in this thesis. In this section we list some of the limitations which one should take into account when inspecting the results and the contributions of this thesis. We report on limitations concerned the data we used, the methods we developed, the tools we built, and the experiments we run to evaluate them.

Data. For testing our methods we needed several data sets that fall under our definition of stories. Although the scientific community has made an effort in compiling different news data sets, we found them difficult to transform to our notion of stories (see Chapter 4, Section 4.7). Therefore, we compiled the data sets used in this thesis relying on online sources - namely Google and Wikipedia. Using Google as a data source for scientific research has several availability and replication drawbacks as described by Kilgarriff in [2]. We relied on Google relevance scores for the documents. However, we used Google News Archive search[1] which partly relaxes the replication problem[2] and we made our data-sets available to other researchers. To obtain the news content from the news media web pages we used a web page content extractor described in [5]. The authors report that this method performs at around 90% recall from the original news content text. This introduces some noise in our data, and we did not test how this may affect different methods used throughout the thesis.

Methods. The extraction of story graphs largely depends on the selection of "content-bearing" terms (See Section 4.3). Due to the time and resource

[1] http://news.google.com/archivesearch

[2] Relevance judgements inside Google archive change at a slower pace than in the regular web search.

limitations we implemented the most straightforward ways for selecting these terms. Another assumption we made concerns the query generation process described in Chapter 5. We assumed that if presented with bursty patterns users would generate queries following the procedure we described. However, this is mostly based on our intuition. It is also important to note that in investigating differences between news corpora we investigated a limited number of divergence dimensions.

In Chapter 8 we made a number of assumptions on how to interpret the results of the divergence measures and the resulting relation between the news sources. However, we did not validate our assumptions, and cannot tell for sure if our findings fully substantiate our hypothesis.

Tools. The tools we built as a part of this thesis are built mostly for demonstration purposes. They can handle only limited size data sets. We used HTML5 as the front-end technology, and given that this is still a pending W3C standard, our tools are not fully cross-browser enabled.

Experiments. In all experiments we used a limited number of data sets, and all results of our experiments should be understood taking this into account. For some experiments the number of participants was low and the results could be affected by this. However, we regard all results useful for providing insights into the contribution of the methods and tools described in this thesis.

9.3 Future Work

Research in story tracking is still young compared to some other data mining and information retrieval related research. This opens many interesting directions along which the research presented in this thesis can be extended.

One of the biggest drawbacks of the story graphs method is the low recall of the graph with respect to the development representations. We suspect that this is the result of the *story basics* selection (see Chapter 4) based on rather simple and straightforward methods. Improving this corresponds to finding those terms that distinguish the documents in a story. One of the possible ways of improvement is to introduce a background model of all stories and use it similarly to the background models in parsimonious language models [1] used in information retrieval. In this way a story graph would reflect both the specificity of a time period in a story and the difference between various stories.

In this thesis we were concerned with representing, detecting, and discovering developments in a story. We assumed that these developments are independent of each other. However, this is usually not the case and developments are usually connected by causal relationships. Relations between developments have already been explored in [4, 6, 7], and an interesting extension would be to go beyond detecting relations between developments and look into the ways of providing a "guided" navigation through the story document-space. This guidance would involve presenting the users with the documents based on the causality of the content these documents contain.

The frameworks for automatic and interactive story tracking evaluation we defined were tested on a limited number of systems. Many developed methods and tools for story tracking made it virtually impossible to cross-evaluate all of them. In our attempts to include more methods and tools into the evaluation, we discovered that their replication based on research reported in the scientific papers is an extremely challenging task. Therefore, we consider that for a larger scale evaluation, a community wide effort in organizing workshops, unified tasks or data challenges would lead towards somewhat more standardized story tracking research.

Putting the effort into building Web-scalable systems based on the prototypes described in this thesis would allow for studies in more naturalistic settings. Building such systems would include overcoming challenges of data quality, efficient indexing schema, document source reliability, and tool usability as pointed out in [3] for real-time First Story Detection systems.

Another interesting direction of research would be to investigate story tracking not from a user side, but from a content-provider side. In Chapter 3 we presented a model for estimating the click share a content-provider may expect from publishing a document relevant to a recent development in a story. However, it would be interesting to go beyond this and try to discover scarcity in aspects discussed in a story. By analysing both document content and user queries the aim would be to discover those aspects of a story for which the quality and quantity of content does not match the users interest.

9.4 Final Reflections and Conclusion

At the very end of this thesis, we look back at what we have learned during this research, and what can be learnt by reading this thesis. One of the main things we learnt, is that there is a need for story tracking systems. If not on a Web scale, we strongly believe that systems like the ones built in this thesis have their place in smaller in-house systems, especially for applications like archive

search. We also consider that a large portion of our research can be reused by other researchers, and that we made important, and often difficult, first steps.

During the time it took to complete this thesis, story tracking has become an increasingly popular topic both in academia and industry. Story tracking research had been publicised in the leading news media[3], and some systems started operating on a Web scale. For example, Google News has only recently[4] introduced the "follow story" functionality. This functionality enables users to constantly monitor developments on a story. Similarly as in our research, stories are identified by key-words used as queries.

To sum up, the growing body of online available news combined with the rising trend of smartphones and tabled devices will make online news reading omnipresent. This will provide users with an easy access to a large number of news sources. The question that arises is how (will) users cope with this and how (will) they benefit from a constant influx of news articles? We consider that this largely depends on the systems that present the news to the users. Today most of the systems for following news are oriented towards an ad-hoc and fast browsing of news, and rarely provide for a deeper analysis of the history of a news story or its evolution. These tools consider users as an external part of the system, and their only goal to transmit information produced by a news provider. In contrast, during the work on this thesis we aimed towards developing such systems that engage users into news reading. Rather than considering users only as passive readers of the content, we regard them as an important, if not central, part of our systems. We believe that such systems will provide users both with a deeper understanding of news and encourage a more critical position towards news reports.

[3]New York Times report on the MemeTracker system: `http://www.nytimes.com/2009/07/13/technology/internet/13influence.html`

[4]June, 2011

References

[1] Djoerd Hiemstra, Stephen Robertson, and Hugo Zaragoza. Parsimonious language models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '04, pages 178–185, New York, USA, 2004. ACM.

[2] Adam Kilgarriff. Googleology is bad science. *Comput. Linguist.*, 33:147–151, March 2007.

[3] Gang Luo, Chunqiang Tang, and Philip S. Yu. Resource-adaptive real-time new event detection. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 497–508, New York, NY, USA, 2007. ACM.

[4] Ramesh Nallapati, Ao Feng, Fuchun Peng, and James Allan. Event threading within news topics. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 446–453, New York, USA, 2004. ACM.

[5] Jyotika Prasad and Andreas Paepcke. Coreex: content extraction from online news articles. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1391–1392, New York, USA, 2008. ACM.

[6] Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 623–632, New York, USA, 2010. ACM.

[7] Benyah Shaparenko and Thorsten Joachims. Information genealogy: uncovering the flow of ideas in non-hyperlinked document databases. In *Proceedings of the 13th ACM SIGKDD international conference on*

Knowledge discovery and data mining, KDD '07, pages 619–628, New York, USA, 2007. ACM.

Appendix A

STORIES – a story tracking tool

Illija Subašić and Bettina Berendt: Experience stories: A visual news search and summarization system. In Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD), Springer-Verlag LNCS, Berlin, Heidelberg, 619-623. 2010.

Contributions as first author:

- (a) Design and implementation of the STORIES system.

A.1 Abstract

Using data collections available on the Internet has for many people become the main medium for staying informed about the world. Many of these collections are dynamic by nature, evolving as the subjects they describe change. We present the STORIES system for (a) learning an abstracted story representation from a collection of time-indexed documents; (b) visualizing it in a way that encourages users to interact and explore in order to discover temporal "story stages" depending on their interests; and (c) supporting the search for documents and facts that pertain to the user-constructed story stages.

A.2 Introduction

Search engines, RSS feeds, micro-blogging tools, and many other services support Internet users in *story tracking*: following the developments of topics over time. This is usually done by manually or automatic issuing a series of same queries about an event ("Haiti Earthquake"), a person ("Britney Spears"), or a scientific area ("text mining"). This creates a challenge for information seekers because large numbers of new documents from different sources keep arriving at a fast rate. There is clearly a need for different search interfaces and search experience, which provide a concise abstracted representation of information from all the pertinent parts of a source or source set. Users will be able to profit most from summarising services that provide convenient interfaces to both the abstracted summary and the underlying documents, and that allow for and encourage a flexible, (inter)active exploration of the space of the abstracted "stories" and at the same time searches of the space of documents.

To enhance the story tracking search experience, we pursue two goals: (a) give users a more active role in the search process, and (b) break away from traditional "top-10"ranked document search interface. As a part of this we present the STORIES system for news-stories tracking, which instantiates these ideas. It consists of story learning (done by the system) and graphical support for story understanding and story search (provided to the user). The paper builds on the detailed explanation in [1]; it describes a re-implementation with new interface features and examples from a new corpus.

A.3 Related work

Apart from "classical" news search engines like Google News or Yahoo! News, recently many alternative ways of tracking and browsing news collections have been developed. *Google News Timeline* (`http://newstimeline.googlelabs.com/`) provides a pre-set time period (day, week, month, year) overview of news using a timeline interface. It allows for the tracking of news sources, arbitrary queries or entities such as movies, books, music... Another Google system, named *Fast Flip* (`http://fastflip.googlelabs.com/`), provides an interface for browsing news articles resembling hard-copy newspaper reading. *The Yahoo! Correlator* (`http://correlator.sandbox.yahoo.net/`) associates a search term with all its related "events". *EMM NewsExplorer* (`http://emm.newsexplorer.eu`) and *EMM NewsBrief* (`http://emm.newsbrief.eu`) are news search and summarization services tracking news from a number of multilingual sources. *SearchPoint* (`http://searchpoint.ijs.si/`) system allows users to focus on a specific sub-topic of search results. *MemeTracker* (`http://www.memetracker.org/`) tracks quotes from news and visualises their "burstiness" using interactive charts. These deployed systems rest on a growing body of work on topic detection and tracking, temporal text mining, and visual web search surveyed in [1].

In contrasts to other tools, our system combines visual search, summarization, and burst-pattern detection into a single interface which provides an interactive inspection of temporal changes in a corpus.

A.4 Method

First, a corpus of text-only documents is transformed into a sequence-of-terms representation. Subsequently, basic term statistics are calculated to identify candidates for story basics. We applied different measures to obtain the story basics including the top ranked words based on term frequency, TF.IDF weight, combining "regular"terms with terms referencing some named entities, and all terms form a corpus.

For *story understanding*, we analyse a text corpus and its (user-definable) time-indexed subsets. For each time-indexed subset of the whole corpus c_i, the *frequency* of the co-occurrence of all pairs of content-bearing terms b_j in documents is calculated as the number of occurrences of both terms in a window of w terms, divided by the number of all documents in c_i. This measure of frequency and therefore relevance is normalised by its counterpart in the whole corpus C to yield *time relevance* as the measure

of burstiness: $TR_i(b_1, b_2) = (freq_i(b_1, b_2))/(freq_C(b_1, b_2))$. Thresholds are applied to avoid singular associations in small sub-corpora and to concentrate on those associations that are most characteristic of the period and most distinctive relative to others: This gives rise to the *story graphs* $G_i = \langle V_i, E_i \rangle$ for time periods i. The edges E_i of G_i are the *story elements*: all pairs (b_1, b_2) with absolute frequencies and TR above the respective thresholds. The nodes V_i of G_i are the *story basics*, the terms involved in at least one association in this symmetric graph: $\{b_j \mid \exists b_k : (b_j, b_k) \in E_i\}$. From each document, we extract sentences containing "facts", short statements with semantic role labeling, as returned by Open Calais (`http://www.opencalais.com/`). The full set of these sentences for each time period is indexed using Lucene (`http://lucene.apache.org`). We then use story graphs to filter the most important facts: for each of the graph's edges, we query the index, using node names of the edge as query terms, and select the top sentences as defined by Lucene. We treat the resulting set of short textual statements as a summary of the story.

Story search can be constrained by the nodes of a subgraph of the story graph. Retrieval is then restricted to documents relevant to these subgraphs. The selection of documents of the starting corpus C corresponds to a top-level query; this query is expanded by the information from the subgraph and the time restriction. STORIES then uses all the nodes n as a query (restriction) for the documents inside c_i to obtain the pertinent document subset, as identified by a search over a Lucene index.

A.5 Tool

Figure A.1 shows a screenshot of our STORIES front-end. We apply the method to news articles obtained from different sources on the Web. Corpora can be compiled either on a continuing basis (e.g., subscribed-to feeds) or in response to a top-level query to a search engine. Our indexing service crawls a number of news aggregators and retrieves documents for a given top-level query. The top-level query describes the whole story (e.g., "Haiti Earthquake" or "*person name*"for crime cases or celebrity reporting). Data cleaning and other data preparation steps are then applied, in particular HTML wrapper induction and removal, tokenisation, cross-document named-entity recognition, lemmatisation, and stopword removal. Finally, document and term measures as described in the previous section are computed. The system backend is developed in Java and the front-end using GUESS (`http://graphexploration.cond.org/`) library for the stand-alone version, and a

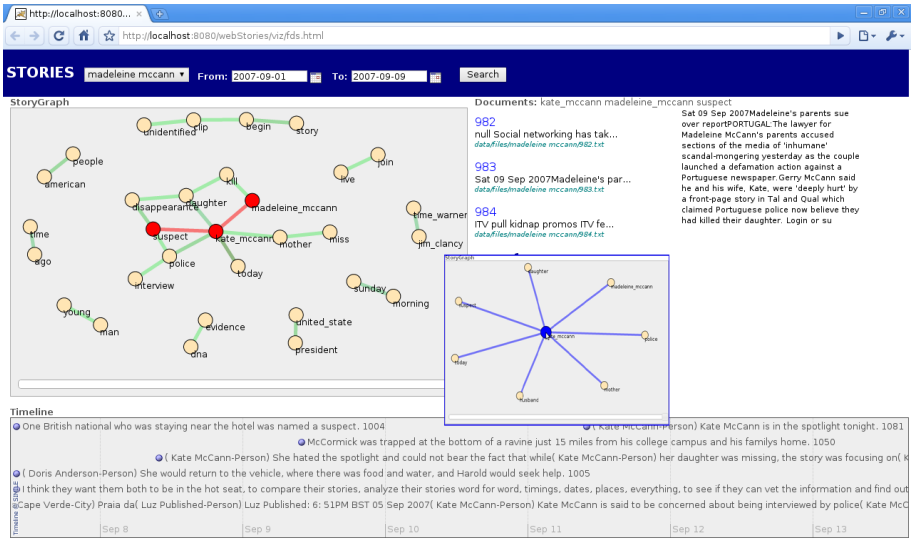


Figure A.1: Web interface: A *story graph* (left) is built based on articles about the disappearance of a person. By marking the edges connecting the person's name (top node of the subgraph marked in red) with another name (middle node) as a "suspect" (left node), the user obtains a list of pertinent documents (centre), whose text can be inspected (right). The *timeline* (bottom) shows the important "facts" from a selected time period. The overlaid *tracking story graph* shows how the searchcan be focused on the chosen node over different time periods.

JavaScript implementation of the spring-layout algorithm for the web version of the tool.

The primary representations are visualisations of story graphs. They provide functionalities to: (1) scan over time to track the global story evolution, (2) zoom in and out by time, by adapting the period-window size, (3) zoom in and out by detail, uncovering more details about the story by setting the TR values (a configurable colour coding schema of edges points to the different values of time relevance), (3) tracking certain terms or entities in time, by selecting the corresponding node. This outputs a graph of bursty co-occurrences including this "tracking node"as its central node.

By clicking on a single edge, the user can select documents associated with the term pair. For easier and more flexible search, users may also select an edge and then highlight a subgraph which contains the selection's adjoining edges

and neighbouring nodes. Each selected edge expands the query and restricts the document set based on it. In this way, the user incrementally builds the query and at the same time can discover and learn about the story. Search provides a list of documents, and a set of sentences visualized along the timeline using one day as an atomic period. This allows users to interact with the story using different views: patterns, sentences (“facts"), and documents.

A.6 Evaluation

Evaluations of *search quality* demonstrated that STORIES finds coherent subsets of documents, that its quality is comparable to or better than state-of-the-art clustering, and that the tool enables people to answer questions on ground-truth events accurately and quickly [1]. We also *compared our method with other “bursty-pattern discovery” methods*, with a framework that leverages sentence retrieval and internal pattern structure and evaluates the sentences against a ground truth. Our experiments showed that different methods perform at similar levels overall, but provide distinctive advantages in some settings [2]. In a third, ongoing round of evaluations we perform a *user study* evaluating how users can discover information using our and other search interfaces.

A.7 Outlook

In future work, we will investigate more advanced language processing (linguistic parsing, semantic role labeling for story graphs, etc.), the use of lexical resources and other background knowledge, as well as different sources of media bias/viewpoints. We also plan to explore aggregation and analysis dimensions other than time, such as multilinguality. Further quantitative and qualitative evaluations will be carried out.

References

[1] Ilija Subašić and Bettina Berendt (2010). Discovery of interactive graphs for understanding and searching time-indexed corpora. *Knowledge and Information Systems, 23(3),* 293–319.

[2] Ilija Subašić and Bettina Berendt. From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. In *Proceeding of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 517–522, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

Errata

- Section A.1, page 234, paragraph 1: *Using data collections available* should be Using news collections available;
- Section A.2, page 234, paragraph 3: *present the STORIES system for news-stories tracking* should be present the STORIES system for news-story tracking;
- Section A.2, page 234, paragraph 3: *detailed explanation in [1]; it describes* should be detailed explanation in [1]; the present paper describes;
- Section A.3, page 235, paragraph 2: *In contrasts to other tools* should be In contrast to other tools;
- Section A.4, page 236, paragraph 2: *this query is expanded by the information* should this query is expanded with the information;

Arenberg Doctoral School of Science, Engineering & Technology
Faculty of Engineering
Department of Computer Science
Declaratieve Talen en Artificiële Intelligentie
Celestijnenlaan 200A box 2402
B-3001 Heverlee

KATHOLIEKE UNIVERSITEIT LEUVEN

ASSOCIATIE K.U.LEUVEN