From Detection to Discourse: Tracking Events and Communities in Breaking News

by

Igor Brigadir

This thesis is submitted to University College Dublin in fulfilment of the requirements for the degree of Doctor of Philosophy

in

School of Computer Science
Head of School: Pádraig Cunningham

Principal Supervisor: Dr. Derek Greene
Secondary Supervision: Professor Pádraig Cunningham

External Examiner: Prof. Udo Kruschwitz
Internal Examiner: Dr. Michael O'Mahony
Chair of the Examination Committee: Prof. John Murphy

December 2016

Statement of Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree stated on the Title Page, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Signature _____

ACKNOWLEDGEMENTS

Foremost, I offer my sincerest gratitude to my supervisors Derek Greene and Pádraig Cunningham for invaluable guidance, continued support, patience, and for giving me the opportunity to pursue a PhD. in the first place.

Throughout the years I've had the opportunity to work closely with many people. I am deeply indebted to all the members of the *Clique* Strategic Research Cluster where I began my work, everyone I've had the pleasure of working with at Storyful, and all my co-workers at the Insight Centre for Data Analytics.

I also owe a great debt to the many, many people that I've either communicated with or met in person, from different academic and non-academic backgrounds, all of whom influenced my work in big or small ways.

I have been privileged to have had the opportunities, resources, and support without which this work would not have been possible.

ABSTRACT

Online social networks are now an established part of our reality. People no longer rely solely on traditional media outlets to stay informed. Collectively, acts of citizen journalism have transformed news consumers into producers. Keeping up with the overwhelming volume of user-generated content from social media sources is challenging for even well-resourced news organisations. Filtering the most relevant content, however, is not trivial. Significant demand exists for editorial support systems that enable journalists to work more effectively. Social newsgathering introduces many new challenges to the tasks of detecting and tracking breaking news stories. In detection, substantial volumes of data introduce scalability challenges. When tracking developing stories, approaches developed on static collections of documents often fail to capture important changes in the content or structure of data over time. Furthermore, systems tuned on static collections can perform poorly on new, unseen data. To understand significant events, we must also consider the people and organisations who are generating content related to these events. Newsworthy sources are rarely objective and neutral, and in some cases, purposefully created for disinformation, giving rise to the *fake news* phenomenon. An individual's political ideology will inform and influence their choice of language, especially during significant political events such as elections, protests, and other polarising incidents. This thesis presents techniques developed with the intention of supporting journalists who monitor social media for breaking news. Starting with the curation of newsworthy sources, through to implementing an alert system for breaking news events, tracking the evolution of these stories over time, and finally exploring the language used by different communities to gain insights into the discourse around an event. As well as detecting and tracking significant events, it is of interest to identify the differences in language patterns between groups of people around those events. Distributional semantic language models offer a way to quantify certain aspects of discourse, allowing us to track how different communities use language, thereby revealing their stances on key issues.

CONTENTS

Acknowledgements ii

Abstract iii

1 Introduction 1

1.1 Aims and Overview . . . 2
1.2 Twitter as a Data Platform for News . . . 4
1.3 From Detection to Discourse, One Component at a Time . . . 11
1.4 Filtering and Curating Sources . . . 13
1.4.1 Newsworthy Sources . . . 14
1.4.2 Pre-processing and Features . . . 15
1.5 Detecting Breaking News Events . . . 16
1.5.1 Detection as Tracking . . . 16
1.5.2 Anomaly-based Detection . . . 16
1.5.3 Event Detection with Humans-in-the-loop . . . 17
1.5.4 Parameter Tuning and Evaluation Issues . . . 18
1.6 Tracking Events Over Time . . . 18
1.6.1 Tracking the Now: Following Breaking News . . . 18
1.6.2 Tracking the Past: Retrospective Analysis of News . . . 19
1.7 Discourse Communities in News . . . 20
1.7.1 Critical Discourse Analysis . . . 20
1.7.2 Corpus-driven CDA . . . 21
1.8 Summary of Challenges . . . 22
1.9 Contributions . . . 23
1.9.1 Publications . . . 23
1.9.2 Supplementary Materials . . . 25

2 Filtering and Curating Sources **26**
2.1 A System for Twitter User List Curation 28
2.1.2 Human-in-the-loop Curation System 28
2.1.3 From Curation to Monitoring 30
2.1.4 Conclusions 32

3 Detecting Breaking News Events **34**
3.1 Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering 36
3.1.2 Related work 37
3.1.3 SNOW Data Challenge 39
3.1.4 Hierarchical Clustering Approach 41
3.1.5 Development Evaluation and Parameter Settings 47
3.1.6 Challenge Evaluation 48
3.1.7 Discussion 50
3.2 Effectiveness of Breaking News Event Detection 54
3.2.2 Task and Scope 56
3.2.3 Source Data: Reportedly Dataset 59
3.2.4 Evaluation Strategies 62
3.2.5 Baseline Systems 65
3.2.6 Results 69
3.2.7 Related Work 75
3.2.8 Conclusion 78

4 Tracking Events Over Time **81**
4.1 Real-Time Event Monitoring with Trident 83
4.1.2 System Description 84
4.1.3 Evaluation 86
4.1.4 Future Work 88
4.2 Adaptive Representations for Tracking Breaking News on Twitter . . . 90
4.2.2 Problem Formulation 91
4.2.3 Related Work 92
4.2.4 Tweet and Timeline Data 93
4.2.5 Sliding Window Timeline Generation 95
4.2.6 Creating Text Representations 97
4.2.7 Parameter Selection 101
4.2.8 Evaluation 103
4.2.9 Future and Ongoing Work 109
4.2.10 Conclusion 111

4.3 Detecting Attention Dominating Moments Across Media Types 112
4.3.2 Related Work 113
4.3.3 Document Diversity Across Media Types 114
4.3.4 Datasets 116
4.3.5 Attention Dominating Events 117
4.3.6 Discussion 121

5 Discourse Communities in News **123**
5.1 Dimensionality Reduction and Visualisation for Voting Records 124
5.1.2 Related Work 125
5.1.3 Dimensionality Reduction for Roll Call Votes 126
5.1.4 Visualising 6th and 7th European Parliament 129
5.1.5 Discussion 134
5.1.6 Conclusion 134
5.2 Discourse Communities with Distributional Semantic Models 135
5.2.2 Related Work 137
5.2.3 Exploring Discourse with Distributional Semantic Models . . . 139
5.2.4 Datasets 146
5.2.5 Case Study: 2014 Scottish Referendum 147
5.2.6 Case Study: 2014 US Midterm Elections 152
5.2.7 Conclusion 156

6 Conclusions **158**
6.1 Recent Advancements 159
6.2 Future Work 162

CHAPTER

ONE

INTRODUCTION

Historically, the mainstream journalism industry had significant control over production, distribution and consumption of news. Journalists acted as *gatekeepers*, exercising control over newsgathering, publishing, and even news commentary in terms of editorial selection of public letters or comments to be included alongside articles or broadcasts [1].

However, with the growth of the World Wide Web, and a variety of social network services providing efficient platforms for communication, news production and consumption practices have changed radically [2]. In particular, Twitter has established itself as a prominent news platform in this environment.

In comparison with other platforms, the emphasis on "peripheral social awareness" [3], and a real-time, reverse-chronological feed made Twitter an attractive tool for journalists. In turn, the presence of an increasing number of journalists active on the platform [4] makes it an attractive source of data for studying breaking news.

For academics, the focus on Twitter was initially driven by the ability to access a large volume of public conversations, and the fact that most Institutional Review Boards (IRBs) have not considered data gathered from Twitter as "human subject data", reducing the amount of work required to carry out a study.

Other social networks such as Facebook have larger user bases, but the quantity of publicly available data is smaller, and it is more difficult to obtain.

Twitter first launched in March 2006[1]. The microblogging service allowed people to share short, 140 character messages mimicking SMS, and quickly became popular with journalists. More recently, it is estimated that about a quarter of Twitter's *Verified*[2] users are journalists [5].

[1] `https://about.twitter.com/company/press/milestones`

[2] The *verified* status of users on Twitter is reserved for accounts that are "of public interest and authentic" eg: corporations, celebrities, key figures in government and religious leaders.

Two events, in particular, are worth highlighting in Twitter's history, as they represented key points when the potential for newsgathering and influencing social movements became widely accepted in mainstream media. On January 15, 2009, US Airways flight 1549 was forced to make an emergency landing on the Hudson River. Within minutes, Janis Krums who was a passenger on a nearby ferry, took a photo of the scene, and posted an update to Twitter[3]. All passengers and crew were evacuated safely. News about the plane crash spread quickly on social networks, with eyewitness accounts appearing long before they were picked up by mainstream media. In a second key event in 2011, widespread civil unrest in Tunisia and Egypt was widely reported on the platform [6]. In this case, Twitter served as an important communication tool for increasing global awareness of these events [7], and in gaining international support for the protest movements.

These and other significant events have reaffirmed that in the distribution and production of news, there is no longer a distinction between "online" and "offline" [8], or indeed between news "producers" and "consumers".

Building upon concepts and techniques from the fields of Recommender Systems, Information Retrieval, and Natural Language Processing, this thesis focuses on the use of social media for news production and consumption.

The chapters and articles are organised into four high-level *component* parts, covering the stages involved in news production and consumption. How these combine and which research areas these components draw on are discussed in Sections 1.4 to 1.7. Taking this *component-based* approach to think about the overall challenge serves as a guide for navigating the research areas this thesis contributes to.

1.1 Aims

The first research aim of this work was to support and enable journalists to effectively curate "newsworthy" sources on Twitter. With 185 million monthly active users at the end of 2012[4], the key challenge with social newsgathering was identifying relevant sources and content from the millions of personal accounts and updates.

The focus on Twitter was largely driven by its popularity among journalists as a data platform for news.

[3] `https://twitter.com/jkrums/status/1121915133` At the time, Twitter did not implement sharing of photos embedded in tweets, so the photo was posted to Twitpic, a third party service which Twitter later acquired.

[4] `https://investor.twitterinc.com/financial-information/quarterly-results/`

At that time, large media organisations began using more user-generated content, emphasising the role of "curator" in the list of skills expected of journalists working with social media.

Filtering techniques for other source sites like Instagram and YouTube also started becoming lucrative around the same time.

As part of this shift, finding valuable social media content became a critical task. Journalists would typically monitor several "feeds" consisting of tweets from collections of users via TweetDeck[5], where these collections of users might relate to a specific news story, topic area, or geographic region. The manual process of filtering and verifying content from these feeds was time-consuming and labour intensive, but necessary, as existing automated systems for news event detection were not sufficiently robust or reliable.

In addition, other related tasks, such as contacting eyewitnesses and verifying reports, could not be automated.

Journalists required tools to curate the sources they monitored, discovering new or important accounts in specific topics or geographic areas of interest. Collaborating with journalists from Storyful[6], this led to the development of a recommender system and uncovered the need to provide more effective monitoring and alerting tools.

Previous research in Topic Detection and Tracking (TDT) [9] was the starting point for investigating ways to provide journalists with timely alerts about potentially interesting breaking news events.

The demand for timely processing of large volumes of data necessitated investigating techniques capable of working with streams of data, and not just static collections of documents. While the volumes of data were manageable without the need for distributed systems, the rapid growth of the number of tweets and users that were of interest led to developing systems that could scale over time.

Once a system for alerting journalists was implemented and deployed, tracking the development of new stories was a natural extension. Existing techniques from TDT designed for longer news articles performed poorly on short, informal and platform specific language on Twitter.

The real-time search task in TREC 2011 microblog track [10] showed that finding the most recent but relevant information to a query was far from solved. To deal with this highly dynamic environment, we explored adaptive ways of updating query and document representations to facilitate more effective tracking.

[5] https://tweetdeck.twitter.com
[6] https://storyful.com/about/

Many of the locations and topics journalists were interested in involved political events, such as elections, uprisings, protests. These events frequently generate large volumes of highly-partisan discourse. Getting a balanced view of an event became more challenging, with certain groups being over-represented on social media.

In order to get a deeper understanding of events and the people involved, and examine how language was used during events with political significance, components for analysing partisan communities were developed.

Overview

This thesis is comprised of content chapters adapted from reviewed publications. Additionally, the introduction chapter outlines the use of Twitter data for newsgathering in 1.2, 1.3 outlines specific techniques and components involved in building systems for detecting, tracking, and analysing the language use of different communities. Sections 1.4 to 1.7 describe the background and additional details for each component.

Section 1.9 lists the publications and other contributions. Chapters 2 to 5 are comprised of the published papers.

Chapter 6 provides a summary of the latest developments introduced since the publication of the papers in each chapter, with 6.2 outlining promising directions for future work.

1.2 Twitter as a Data Platform for News

What makes Twitter an interesting subject of study and use as a data platform for news, as opposed to other social networks? An often cited reason is the growing, global user base and the availability of large volumes of data through an application programming interface (API). A more important reason is the presence of politicians, celebrities, and journalists on the platform [11].

However, other platforms and social networks such as Facebook, and more recently Snapchat and Instagram also provide some access to data and also have prominent organisations and figures as members. What makes Twitter different and more useful for breaking news?

In contrast to other social networks, shared posts are public by default, and *following* on twitter does not have the same social pressure as adding a *friend*, creating a directed network with low reciprocity [12].

Another interesting aspect of content on Twitter is that it is not purely informative or newsworthy, containing a mix of what's often called "pointless babble". This kind of content is better characterised as peripheral social awareness [3] serving a social grooming role that fosters trust and a sense of community through shared experiences of events.

These features combined with a real-time feed comprised of mostly reverse chronological posts from accounts a user chooses to follow[7], make Twitter particularly suitable for dissemination information.

While Twitter was initially used for breaking news by early adopters, the change in the default new tweet prompt from *"What are you doing?"* to *"What's Happening?"* in 2009 strongly emphasised that the service had moved beyond sharing personal updates, towards becoming an "information network"[8].

According to survey estimates, a large proportion of Twitter users say they use the platform for news: 59% in [13], and 86% in [14]. Widespread use by many politicians, government agencies, and celebrities attract even more users, and lacking a viable alternative, Twitter is the dominant microblogging service in English speaking areas. In China, *Sina Weibo* offers a similar service and is more popular. In a large-scale study of Twitter users and their social connections, 85% of trending topics were related to news [12]. A large proportion of content on Twitter is clearly news focused.

Twitter allows anyone to follow any other account without obligation to reciprocate. Based on the declared connections (a user's followers and the accounts they follow), the Twitter network was characterised as having a non-power-law follower distribution, a short effective diameter, and low reciprocity. These are all features that differentiate it from human social networks [15]. However, connections between Twitter users do not imply an interaction between them [16]. Other ways of interacting such as following people via Twitter lists must also be considered. This is an important consideration, as it suggests that actual interactions between users matter more than declared connections. In a news context, this implies that Twitter users can potentially reach a much wider audience than the one defined by the declared network of followers.

The characteristics of Twitter networks, the public nature of the platform, and emphasis on timeliness are all factors that facilitate and encourage news sharing. At the same time, this openness and the ability to easily create pseudonymous accounts has also encouraged large quantities of spam [17], abusive behaviour [18], and disinformation [19] on the platform.

[7] More recently, older tweets and recommendations have been added to the timeline, resurfacing older popular or recommended tweets.

[8] `https://blog.twitter.com/2009/whats-happening`

Technical solutions to some of these problems have been effective, as in the case of spam[9], but dealing with other types of misleading and malicious content remains a challenge. Therefore, the effective use of Twitter for newsgathering or consumption requires careful curation of newsworthy sources.

As a data source, platform-specific features and conventions complicate the analysis, with some introducing ambiguity in sampling and feature extraction steps. In addition, the platform imposes a number of restrictions, including rate limits for gathering data, introduction or removal of platform features, modifications of existing features that change the nature of interactions, and technical changes that break compatibility.

All of these factors can potentially introduce error and variability in the analysis of Twitter-based data. Technical changes in the platform mostly affect experimental reproducibility, with older implementations becoming incompatible with new platform features. Other platform changes, however, such as *likes* or *quotes* significantly alter usage patterns, often making previous work impossible to reproduce or apply in a new context.

Separately, there is also an issue with generalising results obtained from Twitter-based studies to larger populations. For example, while Twitter has a large number of active users, it often makes a poor sample for election forecasting [20]. A 2015 study [21] failed to reproduce 6 out of 10 such propositions made in previous studies. Generalising to other domains, as opposed to the population at large also presents challenges: Tweets, with their short, terse style and length restrictions are often compared to SMS messages, and while Twitter was inspired by SMS, few findings about communication patterns, or techniques developed for Twitter are directly applicable to SMS text. Similarly, the practice of using hashtags on Twitter compared to hashtags on another platform like Instagram or Facebook are not comparable directly.

Twitter has experienced significant growth, as well as changes in behaviour over time [22]. Table 1.1 presents a non-exhaustive list of significant changes to the platform highlighting specific impacts.

The 2008 Chino Hills Earthquake, and Hudson River Plane Crash in 2009 were two major breaking news events early in Twitter's history that had a significant impact on the use of the platform as a news source. At the time, Twitter was still advertised as a personal microblog, lacking adequate real-time search tools, which were only introduced in April 2009.

The introduction of *retweets* in November 2009 significantly altered the way in which people used the platform to share information.

[9] `https://blog.twitter.com/2014/fighting-spam-with-botmaker`

Date:	Change:	Impact:
2009-01-15	Hudson River Crash	Widely cited incident supporting Twitter's place in breaking news.
2009-04-30	Search	Users gain the ability to search and discover content in real-time.
2009-10-30	Lists	Lists added the ability to curate sources. Fort Hood shooting highlighted the importance of well-curated lists [23].
2009-11-06	Retweets	Now a native feature, previously retweets were created manually by copy-pasting text. The use of retweets had an immediate effect on the dissemination of rumours [24].
2009-11-12	What's Happening?	Default prompt for new tweets changed from "What are you doing?" to "What's Happening?", marking a change in how Twitter viewed itself in the context of news, focusing less on personal updates.
2011-05-25	TweetDeck	Twitter buys TweetDeck. First version was released in 2008, and was a popular client with journalists.
2012-08-16	APIv1.1	Significant changes to the API included removing the ability to request all replies to a tweet and introduced different rate limits, which affected data collection efforts.
2013-11-12	Collections	Twitter introduced the ability to curate collections of tweets using TweetDeck. Initially called Custom Timelines.
2015-04-06	Quote Tweets	Other tweets can be embedded, introducing a new way to disseminate information [25].
2015-08-06	Moments	Like Collections, but emphasising media. News organisations were the first to access this feature.
2015-11-03	Likes	Twitter "favourites" [26, 27] changed to "hearts" or "likes".
2016-05-24	Extended Tweets	Photos, quoted tweets, videos and mentions no longer count towards 140 character limit, making longer tweets possible.
2017-09-26	Longer Tweets	Languages except Japanese, Chinese, and Korean can tweet 280 characters.

Table 1.1: Significant changes to the Twitter platform, each having an impact on existing work and the kinds of research which can be conducted on the platform.

Retweets have been used to study information propagation [28], political polarisation [29] and other online social interactions. After their introduction, the spread of rumours increased [24], and the number of "manual retweets" where users would copy another tweet verbatim, sometimes with added comments decreased.

This "sharing" behaviour would later emerge again, in the form of Quoted Tweets in 2015—where one tweet could be embedded in another.

For journalists, the release of the TweetDeck client provided an invaluable means of monitoring and searching Twitter. Rather than having a single curated feed, journalists were able to monitor several feeds simultaneously, usually defined by lists of accounts or search terms. Twitter bought the company in 2011, and the service is still being maintained.

Twitter is a popular source of data for researchers, owing to the vast quantities of tweets and network connections obtainable with relative ease through APIs. A new version of the API in 2012 changed the way data could be accessed. As a result, certain types of information became difficult to obtain due to more restrictive rate limiting, affecting the kinds of research questions that could be adequately backed up by data. For instance, a study of tweet replies [30] is now far more challenging to carry out, due to lack of API access to tweet reply threads.

Other significant changes included the removal of count statistics for shared URLs in 2015, affecting any research involving links in tweets, such as [31], and changing *Favourites*, which were used for multiple functions [26, 27], to *Likes*. While the change is cosmetic, the impact on user behaviour is unknown, and presents an opportunity for further studies.

The most recent changes introduced *expanded* tweets and 280 character tweets. Most recent tweets can now have 280 characters of text, as well as a link, a quoted tweet or media attachment, effectively making tweets longer, altering any prior assumptions about tweet document lengths.

The terms of service associated with the Twitter platform can also prove problematic for researchers who wish to reproduce results or work with a shared collection of tweets. Restrictions on data sharing (only identifiers may be distributed) and rate limits of the API make some collections prohibitively expensive to acquire, in terms of time[10] or cost if purchasing data from Premium APIs introduced in November 2017, Enterprise APIs offered by GNIP or from third parties[11].

Twitter data also tends to decay over time [32]. For example, as of June 2016, around 20% of tweets in the TREC 2015 Microblog [33] collection are now permanently unavailable – these include tweets from suspended users and accounts that became private.

Losing access to documents in a collection is problematic for reproducing retrieval experiments, and expensive, considering that some deleted tweets were part of the evaluation, and significant human effort was involved in producing relevancy judgements for those tweets.

[10] As of November 2016, rate limits for gathering datasets where tweet IDs are known have significantly increased, downloading large collections now takes less time.

[11] Texifter charges 20 USD per day, and 30 USD per 100,000 tweets `http://texifter.com/`

Deletions in these valuable, relevancy judged tweets from other TREC Microblog collections are: 19% for 2011, 11% for 2012, 19% for 2013, and 20% for 2014[12].

All of these changes highlight the difficulties of working with Twitter data, and as a platform. Unlike traditional Information Retrieval corpora that are stable over time, Twitter data is continually changing[13].

Despite these issues, Twitter still plays a large part in the media landscape and is therefore worth studying closely.

[12]Deletions were checked by crawling provided judgment Tweet IDs `https://trec.nist.gov/data/microblog.html`

[13]The latest platform changes are now announced on a public roadmap `https://trello.com/b/myf7rKwV/twitter-developer-platform-roadmap`

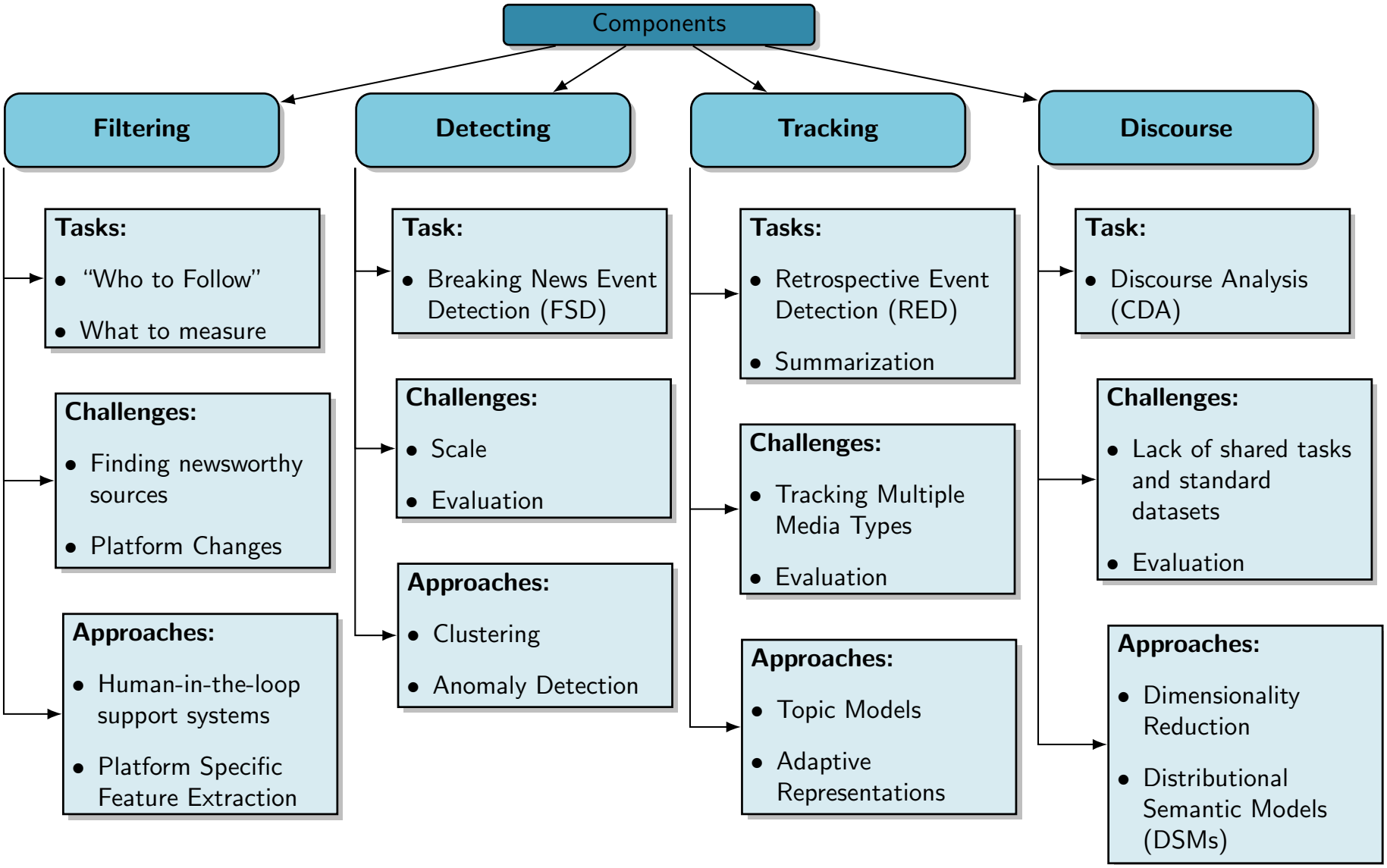


Figure 1.1: Overview of key components in the news analysis workflow. In each component, different tasks present different challenges, but some, like *evaluation*, are not unique to a particular task. Approaches are also broadly applicable across components and tasks—*Clustering* or *Dimensionality Reduction* for instance.

1.3 From Detection to Discourse, One Component at a Time

Reporting news is a complex task, and journalists naturally require different solutions for many different aspects of this task. As a guide to the research presented in this thesis and as a useful classification of other related work, we now consider four high-level *components* for tracking events and communities in breaking news, as listed below and as illustrated in Figure 1.1.

1. **Filtering:** Who should we listen to? Who is important?
2. **Detecting:** When do breaking news events occur?
3. **Tracking:** What has changed over time?
4. **Discourse:** What distinguishes different groups of people? What are they saying, and how is language used to construct their message?

Having defined these high-level components, we now briefly introduce specific analytical *approaches*, with a more detailed discussion of these choices and how they contribute in Sections 1.4 to 1.7. This component view shares some task definitions with TDT tasks [9], namely *First Story Detection*, and *Tracking*. A particular approach, or family of approaches can contribute to multiple high-level components and should be viewed as interchangeable. For example, we used Non-negative Matrix Factorization (NMF) for topic modeling in Section 4.3, but this should not suggest that it is the only applicable approach for the task.

Anomaly Detection:
Several anomaly detection techniques are used later in Section 3.2, where they contribute to the *Detection* (1.5) component. Anomaly-based approaches for event detection usually involve time series analysis, where the time series is built from features extracted from a continuous stream of tweets or by aggregating several signals originating from different sources.

Representing Text with Distributional Semantic Models:
When dealing with text, a suitable representation is critical for the success of any subsequent analysis. Traditional bag-of-words models rely solely on the frequencies of terms in documents. Text is represented as an unordered set of terms, usually with some weighing scheme, such as Inverse Document Frequencies (IDF).

These simple tf-idf models do not consider word order (a critical aspect when studying discourse) but remain useful for many applications, such as BM25 retrieval described later in Section 4.2.

Based on the Distributional Hypothesis [34], that "linguistic items with similar distributions have similar meanings", distributional semantic models (DSMs) encode words as high-dimensional vectors, with the similarity between words measured in terms of vector similarity. These "word space" models have gained in popularity with the introduction of efficient techniques for building word vectors from large text corpora, such as *word2vec* [35]. Despite their recent prominence [36], DSMs have a long history in Information Retrieval literature. The *MatchPlus* [37] system was one early application of a distributional semantic representation of text in a retrieval setting, representing documents and queries in the same high-dimensional space.

Different variants of DSMs are used later in Sections 4.2 and 5.2 for creating suitable representations of text, thereby contributing to both *Tracking* (1.6) and *Discourse* (1.7). This kind of representation is extremely flexible, and different approaches to creating models can be used depending on the task requirements. DSMs are closely related to more general dimensionality reduction techniques, such as Random Indexing and singular value decomposition (SVD), discussed below.

Dimensionality Reduction:
Dimensionality Reduction is a fundamental requirement in a variety of machine learning tasks. A vast array of techniques have been proposed, some task-specific and some general. Several of these are used throughout the different components in our work: SVD for aggregating multiple network views in Section 2.1, Non-negative Matrix Factorization (NMF) for topic modeling in Section 4.3, and Random Indexing in Section 4.2. In addition, t-SNE [38] is a widely-adopted, nonlinear technique that aims to preserve the local structure of data – it is used for visualisation in Section 5.1.

Clustering:
In general terms, cluster analysis involves organising objects into groups, such that objects in one group are more similar to each other than to objects in other groups. There are many applications for clustering, but in the context of this thesis, these approaches contribute mainly to the *Detection* (1.5) and *Tracking* (1.6) components, and the *Discourse* (1.7) component to a lesser extent.

Agglomerative hierarchical clustering with the average linkage measure is used in Section 3.1 for detecting news events. Variants of Sequential Leader clustering are used in Section 4.1 to track stories over time. The *Moving Leader* variant serves as a form of online query expansion.

A third type of clustering is used for multi-document summarization in Section 4.2. The SumBasic SUM_{Σ} variant in [39] is used to de-duplicate content in generated timelines of events.

The choice of specific approaches to serve in the various components of the news analysis workflow depends on the goals of the journalist and the nature of the data that is available to them. Approaches used in one component can also be engineered to serve in a different one, but this is not always the best strategy, as in the case of using tracking systems for detection tasks [40].

It is important to recognise that each choice of approach will introduce different parameters and assumptions. Since a high-level component, such as a *Breaking News Detection* system, will be composed of many different sub-systems and data pipelines, evaluating the effectiveness of each component becomes extremely challenging.

1.4 Filtering and Curating Sources

A critical component of social newsgathering is the sourcing strategy used by a journalist. The sources they draw on will dictate the information that they obtain, which will, in turn, influence their reporting. The selection of newsworthy sources is just one part of the *Filtering* component.

Pre-processing and feature extraction steps applied to data should also be considered part of the filtering component, as these steps exclude content, in the same way as selecting a subset of sources.

From a research perspective, filtering source data also presents a number of issues [41]. Some of these, like pre-processing steps, can be controlled and accounted for. However, researchers carrying out studies on data originating from a platform such as Twitter cannot control the limits and nuances imposed by that platform.

One filtering challenge researchers have no control over is the sampling of public tweets. The 1% sample of public tweets that is freely available via the Twitter API is sometimes biased, in the sense that some trends are significantly under-represented or over-represented [42].

Top hashtags can be misrepresented when comparing data provided by the API with tweets sampled uniformly at random from the set of all public tweets [43]. The larger 10% *Decahose*[14] sample, like the set of all public tweets or the *Firehose*, is prohibitively expensive in most cases.

[14] `http://support.gnip.com/apis/firehose/overview.html`

In comparison to the sample of tweets provided by the default *Sample* API, an *expert sample* in [44] was shown to contain almost no spam, more information on specific topics, and more timely, newsworthy tweets.

Consequently, investing time in building newsworthy lists has definite advantages, in terms of providing cleaner data with less requirement to filter irrelevant or spam tweets at a later point in the analysis process.

1.4.1 Newsworthy Sources

At first glance, the problem of curating newsworthy sources suggests a classification task, where "newsworthy" accounts from corporations, politicians, and institutions are classified and monitored and "personal" accounts are filtered. Treating Twitter as a source of press releases is not very useful.

Capturing *news curators* [45] as well as official sources, allows an organisation monitoring Twitter to take advantage of earlier reporting for some types of events such as sports and natural disasters [46], as well as relevant user-generated content.

Useful user-generated content [47] can potentially come from anyone, and while journalists and official sources are often primary sources for breaking news, most user-generated content is provided by people who have little or no connection to news media.

As a motivating example, in a case study on Andy Carvin's sources on Twitter during the Arab Spring [6], nearly half of the tweet mentions came from *nonelite* sources. These sources included activists, bloggers, and non-affiliated activists. Mainstream media and official government sources are labelled *Institutional Elites* in the study. Such nonelite sources are often eyewitnesses and first responders. An example of this type of eyewitness reporting is the case of Sohaib Athar[15], an IT consultant in Pakistan, who unwittingly reported on a military raid conducted by American Special Forces during which Osama bin Laden was killed [48].

However, no list of sources is perfect, and even traditionally trustworthy sources sometimes make mistakes or become targets of attacks intended to spread disinformation. An example of such an attack in April 2013 involved the Associated Press Twitter account reporting on apparent explosions in the White House.

These reports led many automated trading systems to react, causing massive losses, while journalists were able to immediately dismiss the announcement as erroneous [49].

[15] `https://twitter.com/reallyvirtual/status/64780730286358528`

These cases demonstrate two important points relevant to the *Filtering* component in tracking events and communities in breaking news: collectively, journalists are still the most effective information filters; and support for human intervention is necessary for automated decision-making systems.

1.4.2 Pre-processing and Features

It has been shown that Natural Language Processing systems often perform poorly when applied to social media data [50]. Part-of-speech tagging and named entity recognition approaches achieve high levels of accuracy on well-structured news article text, but fail to perform on short, informal text on Twitter.

Lexical normalisation for Tweets has been somewhat successful, serving to improve the performance of existing tagging and parsing systems on the cleaned text [51]. Purpose built systems for Tweets are more effective [52] but require expensive efforts to construct platform-specific training and evaluation sets.

This mismatch between the language used on social media platforms and that which appears in traditional newswire text is important to consider when training systems on one genre of text, and evaluating the system on another. This problem is exacerbated by the changing nature of the Twitter platform, and different versions of software libraries that may be out of date and fail to take advantage of new features in tweets (such as quoted tweets).

Furthermore, different supervised and unsupervised approaches respond to standard preprocessing decisions (such as stopword removal, stemming, extracting n-grams) in unexpected ways [53]. These preprocessing decisions have important consequences for the quality of features used by both supervised and unsupervised learning approaches. Latent feature models are more useful for the task of *Discourse Analysis*, as we show in Section 1.7.

These are just some of the choices that must be made during the design of a system. These choices, in turn, introduce small differences between systems, and create problems for evaluating approaches, as the effects of preprocessing decisions can sometimes outweigh the effects of interest, such as a new type of model for retrieval [54].

1.5 Detecting Breaking News Events

1.5.1 Detection as Tracking

In the research area of TDT, systems built for *tracking* have been adapted for *detection*, where detection was cast as a tracking task [40]. In tracking tasks, systems are provided with a query or initial set of documents. For each new document, the system decides whether the new document should be included with a known set.

Turning a tracking system into a detection system involves a slight modification: systems are not given initial queries. Instead, given a stream of documents, each new document is compared to each known set. If a new document is dissimilar to any known set, it is marked as a new "event" and a new set is created for tracking.

This formulation has a number of drawbacks. Tracking systems are not perfect, and detection tasks are more challenging. Improving the results of a system in detection, requires unrealistic improvements in tracking performance, as the two are closely related [40], and the performance of detection is always bound by tracking performance.

Many clustering-based approaches adopt this detection-as-tracking problem formulation. A variant using Locality Sensitive Hashing [55] did not perform better than the UMass FSD system [56] but was vastly more efficient than the baseline. Adopting a 3-nearest-neighbour clustering approach [57] improves results, and larger gains are made by incorporating an *anomaly-based* component, using signals from Wikipedia views [58] or edit spikes [59].

1.5.2 Anomaly-based Detection

Twitter streams are naturally bursty in nature, especially during breaking news events [60]. Accounting for bursts also improves other models of human activity [61].

Rather than examining whole documents, as is the case with most TDT systems, signals can be constructed from individual words [62], and then combined with clustering. Further improvements can be made by exploiting Twitter-specific features, in Mention-anomaly-based event detection [63] for example.

Under certain evaluation settings, anomaly-based approaches that aggregate multiple signals outperform approaches that consider whole documents (such as LDA) [64, 65]. However, this does not suggest that anomaly-based approaches are generally better in FSD, but highlight that measuring "burstiness" is beneficial in FSD. Given these improvements, anomaly-based breaking news detection approaches are promising.

Furthermore, there are many existing time series anomaly detection approaches that can be incorporated into a *detection* component, and as these operate on time series data, they do not require any alterations to work in the news domain.

In anomaly-based detection, tuning the approach, and constructing the signal are both important considerations. We explore a range of choices in Section 3.2. One of the most useful features is the number of different people or sources discussing or reacting to a news event.

In earlier TDT tasks, the number of different sources of documents was limited, so this bursty activity was not as evident. The pilot TDT corpus contained just two sources [40] Reuters and CNN. TDT-2 corpus contained six different sources, which is still not enough to detect an anomalous signal.

Breaking news events affect large groups of people, creating temporary communities around shared experiences and reactions [66]. Measuring this wider human impact can provide a better feature set for breaking news detection than text content alone.

1.5.3 Event Detection with Humans-in-the-loop

Human-in-the-loop refers to systems and components that require human intervention or judgement to function. Systems that involve *human computation* [67] or Interactive Machine Learning [68] are sometimes referred to as "human-in-the-loop" systems.

Detection components benefit from human intervention, in similar ways to *Filtering* discussed in Section 1.4. Human interventions can involve query expansion, relevance feedback, or verifying alerts to reduce false alarm rates. In the commercial *event detection* system *Banjo*, human interventions filter false alarms[16] for example.

The best performing system in *Scenario A*[17] task in the 2015 TREC Microblog track [69] involved human intervention. While *manual* runs (involving human intervention) are not the top results overall, in teams that submit both manual and automatic runs, the manual runs perform better.

As well as human interventions in systems, usability is another concern. The high-volume, informal, and length restricted nature of tweets often require non-trivial efforts to de-duplicate and summarize in readable form. As a result, explaining why a system issued an alert at a particular point in time becomes difficult.

[16] `http://www.inc.com/magazine/201504/will-bourne/banjo-the-gods-eye-view.html`

[17] The "push notification" task is similar to breaking news detection but restricted to a number of *interest profiles*.

Presenting a journalist with large quantities of tweets to read is impractical. Event alerts consisting of top-n terms are typically issued by systems. In Section 3.1, to address the challenges of presenting interpretable and coherent alerts, we propose clustering approaches to select representative tweets, and extracting headlines to present users, as opposed to synthesising a human-readable alert.

1.5.4 Parameter Tuning and Evaluation Issues

Experimental design choices for breaking news event detection systems are often not justified or explored in detail. Social media monitoring is an application area of interest to both journalists and researchers and this task presents many challenges. Parameter choices, both within system components and during the evaluation of the system, can strongly influence the eventual output of the system.

As systems become increasingly complex, it becomes more difficult to evaluate which component parts of a system contribute to good performance, and which components require the investment of additional effort. Certain experimental design choices and baselines can lead to significantly different conclusions regarding the performance of a detection system. Examples of this are detailed further in Section 3.2.

1.6 Tracking Events Over Time

1.6.1 Tracking the Now: Following Breaking News

In online breaking news event detection, the emphasis is primarily on timely alerts. Once an event of interest has been identified, tracking an event then requires producing timely updates with new information on that event.

Online Learning, or learning in non-stationary environments, is still an active area of research [70]. There are numerous computational and engineering challenges associated with systems that must adapt over time or work with data incrementally.

Both breaking news and discourse analysis are perfect examples of non-stationary environments. In breaking news, it is rare to have all the information available immediately after an event occurs. Instead, facts are usually revealed slowly over time, often with incomplete information. In discourse, even the very meanings of words cannot be considered static [71]. Furthermore, in both breaking news and discourse, actors may actively disseminate false or misleading information in order to advance some agenda.

Dealing with large volumes of text in a scalable and fault-tolerant manner requires well-engineered systems and an appropriate choice of a learning algorithm. In an online learning setting, performing multiple passes over data is usually intractable, and it is assumed that examples are processed exactly once. Streams can be processed a tuple-at-a-time or using a sliding window approach, processing tuples in small batches. We explore practical challenges with building and deploying such systems in Section 4.1.

An estimated 500 million tweets are created every day[18]. The *firehose* containing every public tweet is not openly available, and so most studies using Twitter data do not operate at these scales. However, rather than building hypothetical systems that *could* process the firehose of tweets, in cases where datasets are smaller, there are still advantages to using these systems—experiments frequently involve tuning parameters, and running different variations of techniques on the same dataset. These experimental settings benefit from massively parallel processing.

The ability of systems to adapt to changing input data is a key requirement when analysing social media data. In work presented in Section 4.1, we adopt an online clustering strategy with a query expansion approach to follow a story as it develops. Section 4.2 explores a different solution, where the underlying representation of queries and documents is adapted and changed over time, as opposed to changing the query.

1.6.2 Tracking the Past: Retrospective Analysis of News

Tracking and summarizing rapidly developing stories as they happen is an important and useful application in online journalism. To analyse stories over longer time spans, we require a retrospective analysis of events.

In a retrospective setting, the emphasis is on novelty—finding previously missed events, or patterns of activity around a known set of events, or looking for all instances of a particular type of event.

Many approaches to Retrospective Event Detection (RED) treat the task as a typical TDT detection task, but parameters are optimised for recall. For journalists, and researchers in other fields, RED is extremely useful for producing visualisations and timelines of news stories or instances of events in a location. Analysis of casualties from suicide bombings in Iraq for example [72], is impossible without retrospective analysis, as many individual events would not be reported immediately, and only documented long after the event.

[18] Tweets per day estimated using the 1% Sample Stream `https://dev.twitter.com/streaming/reference/get/statuses/sample`

In Section 4.3 we explore attention dominating moments across media types. By measuring the diversity of textual content coming from Twitter, newswire services and blogs, we explore how certain significant events cause a collapse in diversity across different sources, corresponding to moments when everyone online appears to be talking about the same thing.

Retrospective Event Detection is an important aspect of the overall goal of tracking news stories from initial reports, to a deeper analysis of the communities and discourse around events.

The variant of discourse analysis used in Chapter 5 requires placing texts in wider social, political and cultural contexts. In Section 4.3 the contexts we examine are different publishing mediums. Without having a way to re-examine events in different settings, this kind of analysis is impossible.

1.7 Discourse Communities in News

The term *discourse* can mean different things in different contexts. A useful definition is, "written or spoken communication". Depending on the context, the study of discourse implies different things and involves different theoretical, and computational approaches.

Discourse in natural language processing literature usually refers to *discourse parsing* [73, 74]. This is a task where the representation of structure plays a central role—for example, extracting discourse relations to describe the high-level organisation of text or speech, facilitating Rhetorical structure theory (RST) discourse parsing [75].

Here we are interested in discourse in a more general sense. In breaking news, the vocabulary used to describe events can evolve as new facts emerge, and different groups of people may describe events from different perspectives.

A *discourse community* is a group of people who share a common goal, or interest. In the news context, a community might correspond to the people participating in the online discussion of a particular news story.

1.7.1 Critical Discourse Analysis

Critical Discourse Analysis (CDA) [76] is a type of discourse analysis that considers text and talk as social practices. The primary concern is the study of naturally occurring language use and patterns, particularly in social and political contexts.

CDA is concerned with how power abuse, dominance, and inequality are expressed in text, and emphasises a close, qualitative examination of the text. CDA is not simply a set of prescribed steps one can follow. Rather, CDA consists of an array of theories, methods, and practices that stem from linguistics, pragmatics, sociolinguistics and other related fields. The diversity of these discourse theories is explored in [77].

CDA takes many forms and can apply at different scales, from the manual analysis of an individual document such as a speech given by a politician, to a sample of documents—how a particular event is reported in news for example [78]. Large-scale analysis usually involves a mixed approach, where features such as word frequencies, co-occurrences, and n-grams are extracted from a large corpus.

1.7.2 Corpus-driven CDA

Critical discourse studies often involve in-depth analysis of a small selection of texts, but in corpus-driven CDA, large corpora are examined for patterns of language, combining corpus linguistics with CDA. Features such as lexical richness, relative frequencies of word types are examined. Large-scale studies help reduce selection bias to some extent [79], but like all social research involving human judgement, it is not entirely neutral.

In Section 5.2, we propose a corpus-driven technique for discourse analysis of ideologically opposing communities during the Scottish independence referendum and U.S. Midterm elections in 2014.

The differences in language use between communities are informed by the differences in the language models trained on text produced by the communities. Machine Learning algorithms should not be seen as more neutral or objective than human judgement, however. The effectiveness of a technique rests on the ability of an algorithm to model the statistical properties of training data.

Word embeddings for example, can effectively capture and also inadvertently amplify gender-based profession stereotypes [80]. For some applications, this is clearly a disadvantage, and techniques to reduce this bias have been proposed [81]. However, for other applications, where the objective is to reveal such biases, this is potentially useful.

1.8 Summary of Challenges

To summarize, for each of the high-level components in the news analysis workflow (Filtering, Detection, Tracking, Discourse), a number of approaches can potentially be used interchangeably.

Each specific approach introduces its own set of assumptions, parameters and limitations. Different high-level components and approaches that address specific challenges can also be organised by algorithm type or event type, rather than task type [9, 82].

Regardless of how a system is organised conceptually, there are common, concrete challenges that must be addressed by researchers and practitioners. These include:

Source Data:
This relates to the availability and format of data. In particular, platform-specific restrictions and changes will limit and impact on what can be studied, and how it can be studied. In some cases, data collection is impossible without ignoring platform rules and terms of service.

Scale:
The problem of dealing with an evolving, high-volume stream of documents can be addressed via efficient algorithms and parallelisation. For example, *TF-SIDF* [83] uses efficient approximate counts for the idf component in $tf - idf$. Where processing on one document does not depend on the processed results of another, handling data in parallel or parallelising computation can be effective.

Parameters:
Initialisation and optimisation parameters must be tuned. For instance, in the case of topic models generated using NMF in Section 4.3, the choice of initialisation and optimisation strategy influences which documents will be assigned to which topics. Visualisation techniques also suffer from parameter tuning challenges. Section 5.1 stresses the importance of tuning parameters in t-SNE for visualising voting records.

Domain Adaptation:
The data domain is also an important consideration. The domain problem extends to *Evaluation*. For example, ROUGE BEwTE [84] toolkit for summarization evaluation used in Section 4.2 performs Part-of-speech tagging and Named Entity Recognition with non-Twitter-specific models, but still uses text originating from the news domain[19].

[19] `http://opennlp.sourceforge.net/models-1.5/`

Evaluation:
The more complex a system becomes, the more difficult it is to reliably evaluate. When it is composed of multiple sub-systems, the effect of each on the overall result will be difficult to estimate. Ideally, having vast quantities of training data for each component and sub-system would enable a detailed analysis of the complete pipeline [85]. In this "ideal" evaluation, each component is given a perfectly accurate input, gradually adding noise in order to quantify system effects.

1.9 Contributions

The chapters in this thesis are based on the following articles:

1.9.1 Publications

- 2.1 **Igor Brigadir**, Derek Greene, Pádraig Cunningham — *A System for Twitter User List Curation* [86] Demo in 2012 proceedings of the sixth ACM conference on Recommender Systems (RecSys).

  D.G. developed the original recommendation system. I.B. developed alerting extension for monitoring activity of lists. I.B., D.G, and P.C. contributed to the analysis of the results. I.B. wrote the manuscript in consultation with D.G. and P.C.

- 4.1 **Igor Brigadir**, Derek Greene, Pádraig Cunningham, Gavin Sheridan — *Real Time Event Monitoring with Trident* [87] in 2013 RealStream: Real-World Challenges for Data Stream Mining workshop at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD).

  I.B. developed the system. D.G. and P.C. contributed to the design and implementation of the research, G.S. contributed to the evaluation. I.B. wrote the manuscript in consultation with D.G. and P.C.

- 3.1 Georgiana Ifrim, Bichen Shi, **Igor Brigadir** — *Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering* [88] in 2014 Proceedings of the SNOW Data Challenge at International World Wide Web Conference (WWW).

  G.I. wrote the published report. G.I, I.B, B.S. contributed to the design and implementation of the research. I.B., B.S. contributed additional data gathering and preprocessing. I.B. wrote the extended chapter section based on the report.

- 4.2 **Igor Brigadir**, Derek Greene, Pádraig Cunningham — *Adaptive Representations for Tracking Breaking News on Twitter* [89] in 2014, expanded version of a paper in NewsKDD Data Science for News Publishing Workshop at ACM Conference on Knowledge Discovery and Data Mining (KDD).

  I.B. developed the system and gathered the data. I.B., D.G, and P.C. contributed to the design and implementation of the research. I.B. wrote the manuscript in consultation with D.G. and P.C.

- 5.2 **Igor Brigadir**, Derek Greene, Pádraig Cunningham — *Analyzing Discourse Communities with Distributional Semantic Models* [90] in 2015 Proceedings of the ACM Web Science Conference (WebSci).

  I.B. gathered the data and performed experiments. I.B., D.G, and P.C. contributed to the design and implementation of the research. I.B. wrote the manuscript in consultation with D.G. and P.C.

- 4.3 **Igor Brigadir**, Derek Greene, Pádraig Cunningham — *Detecting Attention Dominating Moments Across Media Types* [91] in 2016 Proceedings of the First International Workshop on Recent Trends in News Information Retrieval at European Conference on Information Retrieval (ECIR).

  I.B. gathered additional data and performed experiments. I.B., D.G, and P.C. contributed to the design and implementation of the research. I.B. wrote the manuscript in consultation with D.G. and P.C.

- 5.1 **Igor Brigadir**, Derek Greene, James P Cross, Pádraig Cunningham — *Dimensionality Reduction and Visualisation Tools for Voting Records* [92] in 2016 at Irish Conference on Artificial Intelligence and Cognitive Science (AICS).

  I.B. gathered additional data and performed experiments. I.B., D.G, and P.C. contributed to the design and implementation of the research. I.B. wrote the manuscript in consultation with D.G. and P.C.

- 3.2 **Igor Brigadir**, Pádraig Cunningham, Derek Greene — *An Investigation into the Effectiveness of Breaking News Event Detection* in 2016 *in review* [93] Contributions: Paper, dataset and experimental design and evaluation using existing implementations of anomaly detection approaches.

  I.B. performed experiments. I.B., D.G, and P.C. contributed to the design and implementation of the research. I.B. wrote the manuscript in consultation with D.G. and P.C.

1.9.2 Supplementary Materials

Supplementary materials related to the publications in this thesis include:

- System generated event timelines and ground truth evaluation live-blogs[20] used in Section 4.2.
- Users and tweets from Section 5.2 including communities of republicans and democrats from midterm elections, and supporters of Yes and No Scottish Independence Referendum[21] [94].
- Parallel tweet corpus[22] [95] for the Signal Media One-Million News Articles Dataset used in NewsIR'16 ECIR Workshop [96].
- European Parliament roll call voting records and tools[23] used in Section 5.1.
- Ground truth evaluation data from the *Reportedly*[24] newsroom project used in Section 3.2, with additional sources of event data.

These publications have emanated from research conducted with the support of Science Foundation Ireland (SFI) under Grant Numbers 08/SRC/I1407, SFI/12/RC/2289, 07/CE/I1147

[20] http://mlg.ucd.ie/timelines/
[21] http://dx.doi.org/10.6084/m9.figshare.1430449.v1
[22] http://dx.doi.org/10.6084/m9.figshare.2074105.v5
[23] https://github.com/igorbrigadir/vote2vec
[24] https://github.com/igorbrigadir/newsir16-data

CHAPTER

TWO

FILTERING AND CURATING SOURCES

This chapter introduces a system developed to help journalists curate newsworthy sources on Twitter. Given a seed list of Twitter users, the system discovers and recommends new users to journalists. Lists are comprised of Twitter user communities which can potentially cover a geographic region, a topic, or a planned event. The core recommender system was developed in previous work [97] and this chapter also describes a *velocity* extension [86] to prioritise data gathering for active lists, and serve as a prototype event detection system[1].

The system was used extensively at *Storyful*[2] and the filtered stream of tweets from newsworthy sources is used as input data in later chapters. Since publication when 124 lists were reported, the number of lists monitored has grown considerably to around 370, with roughly 2.9 million activity alerts sent from May 2012 to 2017.

Other aspects of the filtering component introduced in Section 1.4, such as dealing with Tweet-specific data, are included in data and pre-processing sections of later chapters.

The sources curated by journalists using this "human-in-the-loop" approach formed an *expert sample* [44] of tweets used in Sections 3.2, 4.1, 4.2, and 4.3, growing from about 20,000 to 30,000 tweets between 2013 and 2016.

The focus on Twitter lists is largely driven by the Tweetdeck-based workflow recommended by many journalists[3]. Tweetdeck originally started as a management tool for multiple Twitter accounts and was later acquired by Twitter in 2011. Multiple "columns" that update in real-time based on keyword searches, groups of users collected in lists and other filters like geolocation appeal to journalists monitoring latest updates or spotting breaking news.

[1] https://www.journalism.co.uk/news/four-new-ways-to-use-social-media-for-newsgathering/s2/a556164/

[2] https://storyful.com

[3] https://medium.com/reportedly/a-rundown-of-some-of-reported-lys-favorite-tools-89b8ba59606e

Typically, a journalist would compile a list of newsworthy sources for a location, topic of interest or some other grouping. This Tweetdeck column would be one of many defined by a journalist. When a breaking news event occurs, journalists would also create a specific column just for that event, usually based on a hashtag or other keywords to gather user-generated content, and eyewitness accounts [98].

What constitutes a newsworthy source worth following on Twitter is sometimes different from what traditional newsgathering requires. In traditional news reporting, most sources are drawn from *elite sources* such as government officials, police forces, etc. whereas on Twitter, it is often useful to monitor updates from highly active, *non-elite sources* [6].

Finding and verifying these *non-elite* sources is an important and time-consuming activity for journalists. Related approaches for finding newsworthy users include [99] that also leverage twitter lists, [100] that focus on users that tend to share news, and [101] that attempt to classify eyewitness accounts.

Treating the filtering of newsworthy sources as a separate component creates more reusable document collections and reduces the need to perform additional processing steps in later analysis.

2.1 A System for Twitter User List Curation

With the increased adoption of social networking tools, it is becoming more difficult to extract useful information from the mass of data generated daily by users. Curation of content and sources is an important filter in separating the signal from noise. A good set of credible sources often requires painstaking manual curation, which often yields incomplete coverage of a topic. In this chapter, based on work in [86] we present a recommender system to aid this process of improving the quality and quantity of sources. The system is highly adaptable to the goals of the curator, enabling some novel uses for curating and monitoring lists of users.

2.1.1 Introduction

Storyful is a social media news agency established in 2010 with the aim of filtering newsworthy content from the vast quantities of noisy data on social networks such as Twitter and YouTube. To this end, Storyful invests considerable time into the manual curation of content on these networks. Twitter users can organise the users they follow into *lists*. Storyful maintains user lists as a means of monitoring breaking news. These lists can be constructed manually, but this process is time-consuming and risks incomplete coverage of all aspects of a news story. Therefore, to support these curation tasks, we have developed and deployed a web-based system for exploring the Twitter network and recommending the important users that form the "community" around a news story (see Figure 2.1). Currently, the system is being used to monitor over 100 news stories, mining microblogging data for a diverse range of topics, from the United States 2012 presidential election to the political situation in Afghanistan. A video of the system in use is available online[4].

2.1.2 Human-in-the-loop Curation System

The input to the system is an initial *seed list* of Twitter users that were manually labelled as being relevant to a particular news story, a topic of interest or location. The bootstrap phase retrieves network structure around the egos in the seed list. Information retrieved consists of user profile information, friend and follower links, user list membership information, and tweets. The extent of the exploration process can be easily controlled by pre-set configuration settings – effectively controlling the trade-off between running time and accuracy.

[4] `http://www.youtube.com/watch?v=rMfN59bmEyc`

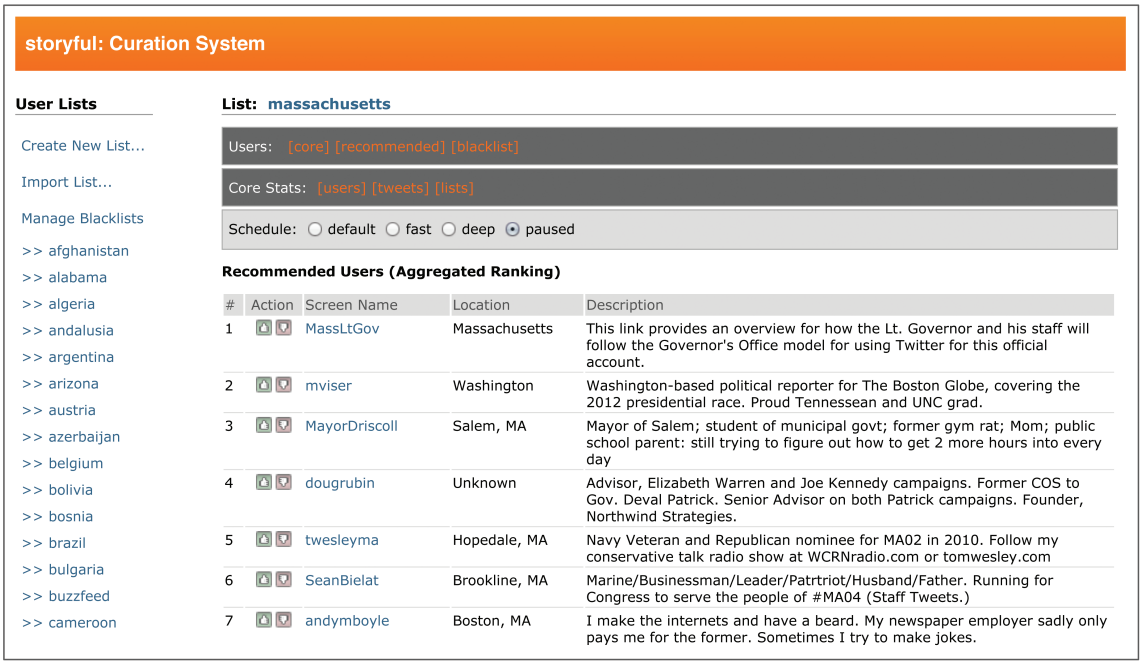


Figure 2.1: Screenshot of the curation system, showing candidate users recommended for addition to a user list covering the 2012 Republican Party nomination for the state of Massachusetts.

After the initial bootstrap phase, the system maintains two distinct lists of users:

Core list: List of the actual Twitter user accounts used by journalists during content curation for the chosen news story. Initially, this will contain the members of the seed list.

Candidate list: User accounts that are not in the core list, but may potentially be relevant for curation. Initially, this will consist of the set of non-seed list users identified during the bootstrap phase.

Based on the candidate list, the recommendation engine will then produce a ranked list of potentially relevant users for promotion to the core list. Based on these recommendations, a human curator can select users to add to the core list, or filter incorrect recommendations.

Once the core list has been modified, the system updates the network structure around the core list, to reflect (a) changes in the membership of the core list, and (b) general changes in the larger Twitter network since the last update. Again the extent of the exploration during the update process can be controlled by user-defined settings. The system then iterates between recommendation and update phases (see Figure 2.2).

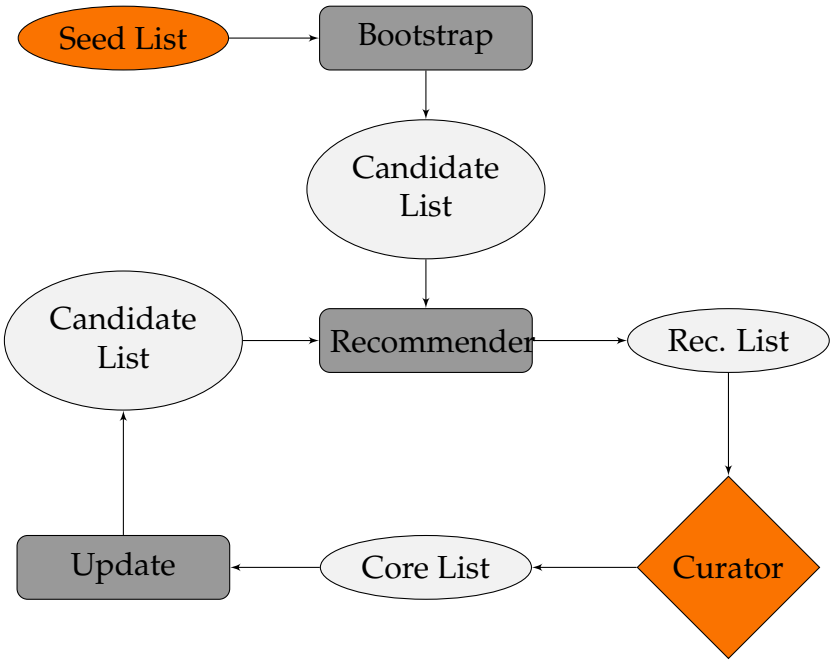


Figure 2.2: Overview of the curation support system, illustrating the workflow between phases.

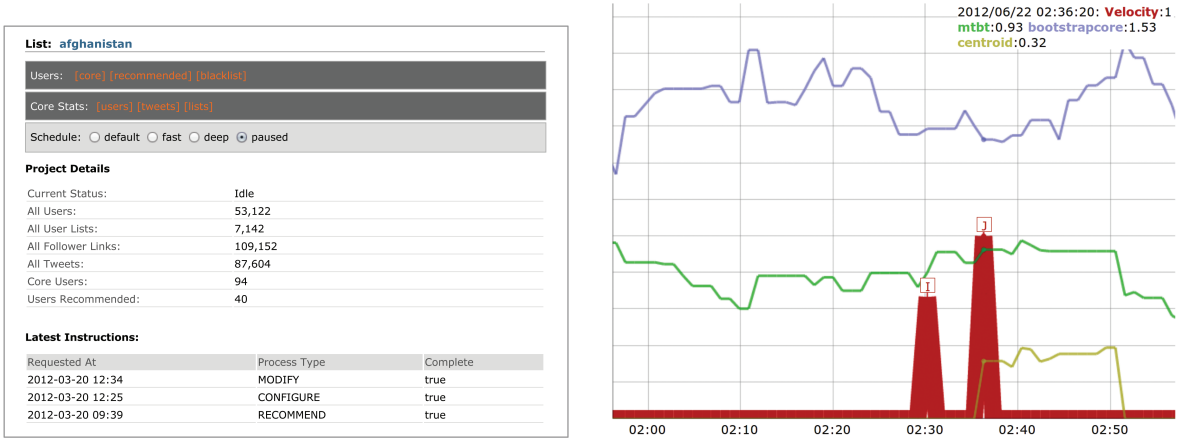
Rather than using a single view of the network to produce recommendations, we employ a multi-view approach that produces recommendations based on different graph representations of the Twitter network surrounding a given user list, and combines them using an SVD-based aggregation approach [102].

Information from multiple views is also used to control the exploration of the Twitter network. This is an important consideration due to the limitations surrounding Twitter data access. These network views include friend and follower graphs, mention and retweet graphs and views based on how users are sorted into lists by other users – effectively crowd-sourcing the list curation in part.

The system runs on open source software and can be deployed on a single server or an Amazon EC2 instance for example. The specific hardware requirements will depend upon the total number of lists being monitored. In April 2012, Storyful were monitoring 124 stories, topics, and geopolitical regions using the deployed system.

2.1.3 From Curation to Monitoring

The curation system is not limited to generating recommendations. When the system is used to monitor lists covering dozens or hundreds of news stories, it will often be important for journalists to focus their limited time on a subset of these lists, where breaking developments are occurring.



(a) Screenshot presenting data statistics regarding a single user list.

(b) A *velocity chart* showing spikes of tweeting activity corresponding to breaking news events being discussed.

Figure 2.3: User List view and Velocity chart for a user list relating to Afghanistan.

To facilitate this kind of prioritisation, we monitor the *velocity* of the lists being monitored by the system. The velocity measure is a combination of several indicators including tweet similarity, the level of activity of the core users, and tweet frequency. The velocity measure can detect significant or unusual tweet activity, often indicating a breaking news story (see Figure 2.3).

The velocity of a list is based on a moving z-score anomaly detector. Specifically, changes over time in three measures are considered: the proportion of users that tweet in a specific list, pairwise tweet similarity, and a keyword counter.

Alerts are issued when there is a significant deviation from an average in two of these measures. *Average* list activity, in this case, means the same 15-minute window, on the same day, measured across a number of previous weeks (Tuesday 13:00 to 13:15, in the last 4 weeks, for example).

Manually set thresholds on these measures allow journalists to get alerts for specific events or when specific words are mentioned.

The effectiveness of this rolling, detrended timeseries approach in removing short-term seasonality is based on two assumptions as previously discussed in [103]: the existence of a daily "news cycle" where most stories are published during business hours when journalists are at work, and weekly cycles where weekends generally see less activity from journalists on lists.

The approach is interpretable by journalists and simple to configure per list by setting the number of days to consider as part of the rolling z-score calculation.

However, a drawback is that during periods of high list activity alerts may be missed as they may not cross the threshold for deviation, or alternatively, during periods of lower tweeting volume (during the night, on weekends) spurious alerts may be generated as the usual activity for that time window may be relatively low.

Different sizes of lists must be configured individually, as settings that work well on one list may not be appropriate for a different one.

2.1.4 Conclusions

While the curation system presented here is primarily used as a support tool within Storyful for curating lists of sources for online journalism, we are currently investigating its use in other applications that involve social media exploration and insight.

For instance, one current experiment uses the system to identify the presence of extremist groups on Twitter. Another application is the identification of spam accounts or bots that share common links in one or more Twitter network views.

System Utility

First deployed in May 2012, alerts were issued as tweets from a set of Twitter accounts, allowing journalists to monitor system output as part of their standard workflow, using Tweetdeck columns to monitor and issue breaking news.

Tweeting a batch of alerts once every 15 minutes, there was an average of 17.5 alert tweets across all monitored lists, showing journalists significant activity for a list that may warrant manual inspection, summarizing an alert with the most frequently used terms for a given list in that 15 minute time window.

The list recommendation system reported over 4,700 recommendation iterations for various lists, mostly active from 2012 to 2014. A single recommendation iteration involves updating network and content information for each list member, generating new recommendations and tweeting a link to a list of new recommended users journalists can manually inspect.

In terms of accuracy, in a previous case study [97] where the top 5 recommended users were automatically accepted without a human curator, precision ranged from 0.97 in the first iteration to 0.88 in the 6th.

With human curators, between 2012 and 2016 the number of curated users increased from around 17,000 to 30,000, and the number of lists from about 150 to 350.

These lists cover every country, as well as a number of significant topics of interest (entertainment, sports) and events (e.g. United Nations Climate Change Conference, Olympics, Formula 1, etc.). The number of journalists using the system also grew from about a dozen to about 50–100[5] in the same timespan.

Datasets

Datasets derived from these lists are used in later chapters of this work, both as a single, combined, high-volume stream of tweets, and segmented by extracted topics:

- 4.1 Real-Time Event Monitoring with Trident
- 4.2 Adaptive Representations for Tracking Breaking News on Twitter
- 4.3 Detecting Attention Dominating Moments Across Media Types
- 3.2 Effectiveness of Breaking News Event Detection[6]

[5] `https://www.crunchbase.com/organization/storyful`
[6] The Reportedly lists of newsworthy sources were based on Storyful lists.

CHAPTER

THREE

DETECTING BREAKING NEWS EVENTS

This chapter consists of two research projects dealing with the task of breaking news detection. In the previous chapter, a prototype event detection system based on the activity of curated lists was introduced in Section 2.1, which is revisited in more detail in Section 3.2 in this chapter.

Breaking news detection on Twitter is made more challenging by difficulties with finding reliable sources, data volumes, platform changes, and misinformation. Another recurring problem with breaking news studies is the lack of realistic evaluations. There are two main ways to bridge this gap: 1) a thorough user-study with multiple annotations and evaluations measures, such as those used in Section 3.1; 2) comparing to output from journalists manually detecting significant events from the same sources, as used in Section 3.2.

The work presented in Section 3.1 onwards emphasises multiple filtering steps to reduce duplicate tweets and present detected events as interpretable headlines. The 2014 SNOW Data Challenge [104] differs from traditional First Story Detection in several important ways. As opposed to judging each document as relevant or not, and each first story as a hit or false alarm, the SNOW evaluation considers a ranked list of topics in 15 minute time windows. Evaluation is based on precision and recall, readability, coherence, and diversity, annotated by a group of judges.

The subsequent work described in Section 3.2 onwards explores evaluation issues and parameter settings for components of these kinds of decision support systems. We show that parameter settings for each individual system component can have a significant effect on detection rates and evaluation outcomes. The paper also addresses the FSD task, but treating the stream of incoming documents in batches, as opposed to one by one. Similarly to Section 3.1, evaluation is based on time windows as opposed to individual documents.

While the evaluation is not as detailed as in Section 3.1, it is based on actual newsroom output from Reportedly[1], and uses a dataset of sources that was curated especially for breaking news monitoring and used by Reportedly journalists, making system output and human-generated output directly comparable.

To journalists, it is not enough to detect the first sign of a breaking news event, as in traditional TDT settings. Breaking news on Twitter often evolves rapidly and unexpectedly. The language around an event can change as the story develops, and systems must be adaptive enough to deal with concept drift. The problem of tracking in Chapter 4 expands on these issues.

[1]Reportedly was an online newsroom that ran from May 2015 to September 2016

3.1 Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering

This section is based on the paper describing the winning submission to SNOW Data Challenge [104]. It has been adapted from [88] and updated to include the results of the challenge evaluation. The challenge descriptions and our technical report detailing the system described *topics* whereas, in this chapter, we refer to *events*. A *topic* in the challenge consists of a news event represented by a headline, a set of tags, a set of tweets, and a representative image.

The SNOW Data Challenge addressed the problem of breaking news event detection in a streaming environment. A unique aspect of the challenge was a thorough manual evaluation of the quality of detection outputs in terms of newsworthiness, readability, coherence, and relevancy. This human annotation of results is labour intensive and expensive to perform, but offers a better comparison of systems than automatic evaluations based on precision and recall only.

Aggressive filtering of tweets, feature selection, and appropriate platform-specific tuning all serve to reduce both the computational load and the redundancy in Twitter streams. A number of heuristics are used, which are based on the content quality, tweet length and news sharing behaviour.

We show that filtering tweets based on length and structure, followed by hierarchical clustering and ranking of clusters, can form a strong baseline system for event detection on Twitter. We present results and discussion for two different Twitter streams focusing on the US presidential elections in 2012 and events related to Ukraine, Syria and Bitcoin in February 2014.

In applying a certain approach to a domain, the need to adjust parameters to domain-specific data yields the best results. We observe that simple methods, combined with appropriate platform-specific pre-processing, feature extraction, and hyperparameter settings produce most timely and coherent results. When applying suggested techniques, particular attention should be paid to adjusting hyperparameters and feature engineering specific to Twitter.

3.1.1 Introduction

The task of detecting news events in text streams has a long history and a rich variety of applications. Issues of scale, concept drift, and evaluating and comparing system performance are key recurring challenges.

Breaking news detection on Twitter is still an active area of research. As the platform develops, however, new features present additional challenges for systems.

Activity on Twitter is often driven by real-world events and dynamics. Important external stimuli, like elections, natural disasters, concerts, and sports events, have an immediate and direct impact on the quality and the volume of tweets posted. Because of its real-time and global nature, many journalists use Twitter as a primary source of breaking news updates, in addition to sharing personal updates and engaging with audiences.

When evaluating event detection systems, the recurring problem of evaluation quality arises. This is especially pronounced in breaking news detection on social media platforms. Twitter is a major communication channel for journalists, and is often the first platform where breaking news is shared and discussed [46]. Given the significant number of journalists active on the platform, the processes of news selection and presentation themselves can be affected by events on Twitter [105, 106].

Topic Detection and Tracking (TDT) [9] has been well-studied for static document corpora, but in the social media context, there are additional factors that make the problem more challenging. These include different language styles between Twitter and traditional news media, fragmented and possibly ambiguous text due to tweet character length constraints, and the high amount of noise in the user-generated content. In addition, researchers face issues with scale, and real-time data processing requirements. High-quality test collections and evaluations of system outputs such as those produced in TDT evaluations are too costly to apply to Twitter event detection. While tweets are short, it is often impractical to have annotators evaluate significant numbers of tweets and other system outputs.

To address the difficulties involved in detecting breaking news in Twitter streams, we propose a number of useful filtering approaches to reduce the redundancy and improve output quality by detecting and filtering low-quality tweets early on in the process. Our techniques produced event detections that were rated best overall in the SNOW challenge evaluation.

3.1.2 Related work

In [107] six techniques are compared for detecting breaking news events on Twitter, and the authors propose a term-clustering approach for detecting trends. The compared approaches can be classified into two main types: document-clustering or term-clustering.

This view of topic detection is similar to other previous work on casting detection as tracking, discussed in Section 1.5. These techniques can be further divided into: probabilistic models (*e.g.* Latent Dirichlet Allocation), classical Topic Detection and Tracking (*e.g.* Document-Pivot Topic Detection) and feature-pivot methods (*e.g.* n-gram clustering). As an alternative to *tracking*, the problem can be solved as an anomaly detection task, as discussed later in Section 1.5.

Document-clustering

Latent Dirichlet Allocation (LDA) [108] is a popular topic modeling approach that assigns each document a probability distribution over topics, which are in turn represented as distributions over words. Topic distributions per document, as well as term distributions per topic, are commonly estimated using Bayesian inference. According to results in [107], LDA models can capture stories happening during events with narrow topical scope, although their performance can be dramatically lower when considering more *noisy* events.

First Story Detection (FSD) is a TDT task [9] where the first news story discussing a news event must be retrieved. On Twitter, this translates to detecting the first Tweet. Typically this detection task is solved with a topic tracking system, where a first story is detected if it does not get assigned to an existing, already tracked story. The FSD approach in [55] uses Locality Sensitive Hashing to rapidly retrieve the nearest neighbour of a document. Documents are clustered based on the cosine similarity of their corresponding *tf-idf*-weighted bag-of-words representations. Using paraphrases has also been shown to improve FSD performance [109] .

Although the initial event recall of plain FSD is not very high, it can significantly improve when employing document aggregation [107] via two-stage clustering, to avoid fragmentation, where the same event is present in several clusters.

Term-clustering

As proposed in [107], *BNgram* is an n-gram feature-pivot method that clusters terms rather than documents. To cluster terms, a document co-occurrence-based distance measure between terms is used. *BNgram* assigns a term burstiness score ($df - ifd_t$) in order to re-rank and penalise terms that have frequently occurred in the past. In a performance comparison considering different types of n-gram representations, the 3 and 4-gram settings were similar, and significantly better than using unigrams alone [110].

BNgram has high recall for topic keywords, and while these are interpretable, output requires further processing to create or retrieve human-readable headlines.

Anomaly-based Detection

As an alternative to detection as tracking, anomaly-based methods can also be effective. A keyword-lifecycle event detection framework was introduced in [111], in which a keyword's standard behaviour is modelled by its frequency and its average daily behaviour. An event is detected when a keyword's frequency is abnormal. In an empirical evaluation, 80% of large earthquakes were detected by this framework, and false positive rates are low. The Window Variation Keyword Burst Detection [112] is another recent anomaly-based event detection method.

Building on FSD and TDT

Following on from recent work, we propose an approach based on tweet clustering, combined with several filtering, aggregation and ranking steps, in order to create an efficient event detection method.

Our choice of tweet clustering, as opposed to term clustering, is based on the following rationale: Tweet clustering methods have shown high recall, particularly in cases where there are many events being discussed. Tweets are often self-contained, and do not require related documents to be retrieved. Generally, tweet text is similar to news headlines, while term-clustering approaches require recreating a sentence from an unordered list of terms, where swapping the order of terms in a cluster can change the meaning of a headline. Another advantage of document-clustering approaches is that we can introduce various tweet-importance metrics for re-ranking content produced by trustworthy sources, such as domain experts and officials.

3.1.3 SNOW Data Challenge

The SNOW Data Challenge [104] was a shared task created to produce benchmarks and evaluations of breaking news event detection systems on Twitter. Given a time interval, the objective is to identify the most significant news events that may be of interest to journalists. The challenge refers to news events as *topics* that comprise of the following: a headline, hashtags, representative tweets, and optionally, an image. A Stream of tweets is presented to systems with a tumbling 15-minute window (non-overlapping sliding window). For each time window, a list of 10 events of interest must be produced by the system.

Dataset	Focus / Keywords	Tweets	Ground Truth Labels?
Development	2012 US Elections	1,084,200	Provided
Rehearsal	UK News, floods, flooding	1,088,593	None
Test	UK News, Syria, Ukraine, Bitcoin	1,041,062	Manual

Table 3.1: Dataset details for the SNOW Data challenge.

The challenge is described in detail in [104]. The *topics* or breaking news events, are produced per time window, and must be ranked by newsworthiness, where the definition of "newsworthy" was a story that would end up being covered in mainstream news sites. For the challenge, 11 teams submitted runs for evaluation, with 9 teams contributing reports describing their approaches in the workshop proceedings.

Three datasets were used in the challenge, as summarised in Table 3.1. The Development set was an older dataset gathered around 2012, covering the Presidential Election, while the Rehearsal and Test sets were gathered in real time. The original 2012 US Elections dataset contained 1,106,712 tweets, the missing IDs are due to deleted tweets, or unavailable tweets due to user suspensions.

The Rehearsal set was used to validate and check the system prior to the competition. We used a Google search-based evaluation (Section 3.1.5) to validate our method. Note that the size of the Test dataset on which the challenge rankings were calculated differs slightly from team to team, due to network connectivity and rate limits. However, this difference was small.

The Rehearsal and Test sets were collected by participants with the provided tools and seed user IDs and track keywords. Provided user IDs were a curated set of 5,000 mostly UK-based newsworthy accounts, derived from public Twitter lists. This approach of selecting newsworthy sources was very similar to the approach described in Section 1.4. The focus on UK-based journalists had several consequences. It reduced the need to filter non-English tweets in evaluations but restricted the types of stories that could potentially be covered. For example, content covering events in Venezuela was lacking, and mostly in Spanish.

The streaming Twitter API operating with these parameters provided: Tweets created or retweeted by the users, or, any tweet mentioning the specific keywords. By using user IDs, the stream of potentially newsworthy tweets is reproducible across teams. However, processing and platform API issues create some discrepancies between the teams, in the reports, teams reported slightly different tweet counts for the test set. For the test set, different keywords were selected by the organisers: "syria", "terror", "ukraine", "bitcoin".

The evaluation period was 24 hours from 18:00 GMT on February 25th 2014. This required the participating systems to generate 96 15-minute time windows with ranked news events of interest.

As the source accounts for the test set were UK based journalists, UK mainstream media was used to generate 59 ground truth events for the 24 hours in the test set, sourced from BBC and Newswhip UK news aggregator. To produce the ground truth, organisers manually merged duplicate headlines, removed opinion editorials, and any stories that had received no mentions on Twitter.

Rather than focusing on recall and attempting to construct an exhaustive list of events, the ground truth was more precision focused, selecting 59 stories that were covered in both Twitter and mainstream media. In Section 4.3 we explore detecting these types of important events across platforms, but retrospectively rather than in a breaking news detection setting.

3.1.4 Hierarchical Clustering Approach

The main idea behind our approach was to filter as much as possible, and re-rank results to present users with the most relevant events. Specifically, aggressive tweet and term filtering to remove noisy tweets and vocabulary. followed by hierarchical clustering of tweets, dynamic dendrogram cutting and ranking of the resulting clusters.

For collecting the Twitter stream we used code provided by the SNOW challenge organisers, based on the Twitter4J library[2]. All other development (e.g., data preprocessing, clustering, ranking, producing final results), was implemented in Python, due to the ease of development and its available range of powerful libraries (e.g., *scipy, numpy, scikit-learn*). In particular, we made use of the *tweet-NLP* library for named entity recognition, (*CMUTweetTagger* library[3]), and *fastcluster* library [113] for an efficient implementation of hierarchical clustering. Our code is available online [4].

Data Collection

We ran tests and selected parameters using two different Twitter streams, a development set collected during the US presidential elections in 2012, between 6 Nov 2012, and 7 Nov 2012 and a rehearsal set, collected between 25 Feb 2014, and 26 Feb 2014. The final results for the competition were extracted from a new, unseen stream.

[2] `http://twitter4j.org/en/index.html`
[3] `https://github.com/ianozsvald/ark-tweet-nlp-python`
[4] `https://github.com/heerme`

The first stream was reconstructed from Tweet IDs, using tweet timestamps, IDs, usernames, the text of the tweet, and whether the tweet is a retweet or not as features. There were 1,084,200, English and non-English tweets in this stream. In order to extract the user mentions, hashtags and URLs from the text of the tweet, we used the twitter-text-python[5] library.

The second stream was collected in real-time from Twitter's Streaming API. This consisted of 1,088,593 raw tweets, of which we only used 943,175 English tweets, by filtering using the *lang='en'* field of the tweet object.

To reduce the data required to process, for each tweet we extracted only the date, id, text, user mentions, hashtags, URLs, and media URLs. For re-tweets, we replace the text of the re-tweet with the original text of the tweet that was re-tweeted (although we only do this for the tweets where this metadata is available, older Tweets in the first stream can miss this feature, as tweets were manually written beginning with "RT"). We use this reduced feature set, with one tweet per line, for all our experiments.

Platform-specific Preprocessing

Preprocessing choices for tweets had a significant effect on output. Generally, preprocessing approaches such as tokenizers, are commonly expected to be used on grammatically correct news corpora. These tokenisers and other pre-processing tools are not appropriate for use on Tweet text, which is structured, given the additional metadata, but creates noisy text data when processed inappropriately.

Tweet text features such as user mentions, replies, retweets, URLs can be extracted either from the tweet metadata or with tweet text processing tools for older collections where metadata is not present[6]. An expanded stopword list was used to remove additional tokens related to common replies and reactions on Twitter (yay, lol, wtf), which helped filter non-informative tweets.

We processed the tweet stream as follows: URLs, user mentions and hashtags, as well as digits and other punctuation, are removed. Next, we tokenise the remaining text by white space and remove tweet-specific stopwords.

In order to prepare the tweet corpus, in each time window, for each tweet, we first append the removed user mentions and hashtags and clean text tokens together to reconstruct a tweet. We check the structure of the resulting tweet and filter out tweets that have more than 2 user mentions or more than 2 hashtags, or less than 4 text tokens.

[5] `https://github.com/ianozsvald/twitter-text-python`
[6] `tweet-text-processor`

The idea behind this structure-based filtering is that tweets that have many user mentions or hashtags, but lack enough clean text features, do not carry enough news-like content.

This step filters many noisy tweets. For example, for the 15-minute time window, starting on 25 Feb 2014, at 18:00, and ending at 18:15, there are 12,589 raw tweets, out of which the first filtering step (that checks the length and structure of tweets) keeps only 9,487. Further processing is applied to this filtered set of tweets.

The next step applies vocabulary filtering. For each time window in the corpus, we create a binary tweet-term matrix, where we remove user mentions, but keep hashtags.

The vocabulary terms are *bi-grams* and *tri-grams* that occur above a certain threshold (in at least 10 tweets). This threshold is set based on the window corpus length, to $max(int(len(window\_corpus) * 0.0025), 10)$.

This threshold does not grow very quickly, for example, for 10,000 tweets, the term should occur in at least 25 tweets to be selected in the vocabulary. The idea behind this filtering step is that clusters should gather enough tweets to be considered an event at all (*i.e.* at least 25 in 10,000 tweets should discuss an event).

For the above example, the term filtering step reduces the vocabulary to 670 terms, producing a matrix with 9,487 by 670 tweet-term matrix. In the next filtering step, we reduce this matrix to only the subset of rows containing at least 5 terms. This step removes out-of-vocabulary tweets, as well as tweets that are too short to be meaningfully clustered.

We varied the parameters for filtering tweets and terms, and noticed that the above values were stable with regards to the events that were produced.

This third filtering step further reduces the original tweet-term matrix to 2,677 by 670, effectively using only 20% of the raw corpus. We have found that for Twitter streams where the language information is not available, e.g., for the 2012 US presidential elections stream, it is much faster to filter tweets and terms as above, rather than using a language identification library to remove non-English tweets.

Hierarchical Clustering of Tweets

After pre-processing we perform hierarchical clustering, headline extraction, and re-ranking of clusters to construct the final output.

As an overview, the following are applied to each *time window* represented by a tweet-term matrix:

1. Calculate pairwise distances for clustering.
2. Apply hierarchical clustering.
3. Cut dendrogram at a distance threshold.
4. Rank clusters by *df-idf* and presence of named entities.
5. Extract headline tweets (earliest tweet in each cluster).
6. Re-cluster the headlines tweets.
7. Extract final headline, keywords, and an image for system output.

The steps are described in more detail below:

1. Calculate Pairwise distances for clustering: For pairwise distances, we scale and normalise the tweet-term matrix, and use *cosine similarity* as a metric. Using Euclidean distance produced similar results.

2. Hierarchical Clustering: The $fastcluster$ library [113] is used for computing a hierarchical clustering of tweets. The idea behind tweet clustering is that tweets belonging to the same news event will cluster together, and thus we can consider each cluster as a detected event, without specifying the number of clusters ahead of time.

3. Cut dendrogram: Hierarchical clustering does not require specifying the number of clusters a priori, as in k-means or other popular clustering algorithms, but we can control how tight or loose we require our final clusters to be with a distance threshold. We cut the resulting dendrogram at a 0.5 distance threshold. A higher threshold would result in looser clusters, that potentially collate different events in the same cluster. A lower threshold would result in tighter and cleaner clusters, but potentially lead to too much topic fragmentation, i.e., the same event would be reflected by lots of different clusters. We found that a value of 0.5 works well for our method.

4. Rank clusters: Given a set of clusters from the previous step, we rank them with a "newsworthiness" score. The newsworthiness score we used is based on term frequency with named entity boosting within a time window.

A first attempt was to use cluster size, allowing clusters with a lot of tweets to rank first as breaking news events. In this setting, many popular, but less newsworthy clusters are promoted. Another issue with using the size of clusters alone is that the same events tend to get frequently repeated for several time windows, as cluster size does not consider burstiness in each time window with respect to the previous time windows.

Using term weighting with named entity boosting was a more effective measure. We use the *df-idf* measure from [107], that discounts the term-frequency in the current time window using the average frequency in the previous t time windows (as shown in Equation 3.1).

$$df - idf_t = \frac{df_i + 1}{\log\left(\frac{\sum_{j=i}^{t} df_{i-j}}{t} + 1\right) + 1} \tag{3.1}$$

Setting the parameter t controls how much history should affect the current weight of a term. We set $t = 4$ in our runs, in order to allow for hourly updates (where a time window is set to 15-minutes). Note the *log* in the denominator, allowing the current document frequency to have more weight than the previous/historical average frequency.

A critical element of this cluster ranking component is named entity extraction. We experimented with the Stanford NLP[7] and the NLTK pos-tagger [114, 115], but found that they fail to recognise entities due to the specific language of tweets, *e.g.* arbitrary capitalisation of words (e.g. "AWESOME vs obama", many NER taggers rely on capitalisation for clues on potential entities [116]), short names (*e.g.* fb for Facebook).

For this reason, we used the Tweet text specific CMU Tagger[8] for recognising entities [52]. Specifically, the Python wrapper around the Java implementation[9]. This tool is trained on tweets and had better accuracy for named entity recognition in our tests. The main advantage of this tagger is the ability to identify named entities from very short stings, and insensitivity to capitalisation and common abbreviations.

We use the tagger as a flag to check if the bi-gram or tri-gram is an entity, rather than applying the tagger to the whole tweet and then aligning the named entities with the vocabulary terms.

We assign a weight to each term in a time window (bi or tri-gram) using $df - idf_t * entity\_boost$, where the entity boost was set to 2.5 as opposed to 1.5 used in [107]. We found that higher entity weights lead to retrieving more news-like breaking news events.

Given the term weights, the cluster score is the score of the term with the highest weight (as in [107]), but we normalise this by the cluster size. This last normalisation step seems to lead to less topic fragmentation, allowing smaller clusters with prominent terms, to rank higher.

We have also experimented with cluster scores that average the score of the terms of a cluster. Interestingly enough, when using unigrams rather than bi-grams and tri-grams for the vocabulary, ranking clusters by averaging term scores worked better than using the maximum term score.

[7] `http://nlp.stanford.edu/software/`
[8] `http://www.ark.cs.cmu.edu/TweetNLP/`
[9] `https://github.com/ianozsvald/ark-tweet-nlp-python`

We chose to keep using bi-grams and tri-grams for scoring clusters as this pre-processing step is already performed, and using a different tokenizer for clustering and scoring introduces additional complexity.

We have also attempted to assign a boost to terms based on their occurrence in news articles that are streamed in a similar time window as the tweets. This approach may work for some types of events, such as politics-related events, where the news travels from the news outlets to Twitter. This may not work for events that first break on Twitter, such as sports events, that are later reported and summarised by the news outlets.

How events are reported across different platforms such as Twitter, news articles and blogs is explored in Section 4.3.

Heavier NLP feature extraction including pos-tagging and extracting nouns and verbs was attempted, however, minimal stopword removal and tweet filtering proved to be much more efficient and equally accurate. As in [107], we found that stemming hurt the quality of results.

The scoring used in result runs was the maximum entity boosted term in bi-grams and tri-grams normalised by the cluster size.

We rank the clusters using this score, and retain only top-20 clusters, subject to a size constraint: for a cluster to be considered valid, it should have at least 10 tweets.

5. Extracting headline tweets: To present a human-readable *headline* for system output, we select the earliest tweet in each of the top-20 clusters.

This clustering and ranking strategy works well for many different events but suffers from topic fragmentation, where we see several headlines about the same news story across clusters. This issue has also been found previously in [107].

6. Re-cluster the headline tweets: Our final step involves clustering only the headline tweets selected in the previous step. These are pre-processed tweets used for clustering in the first stage (excluding user mentions and URLs, with filtered vocabulary).

We build a headline-by-term matrix, this time using unigrams for our vocabulary, without any minimum count thresholds or other restrictions on terms. We re-cluster the headlines using hierarchical clustering, and cut the dendrogram at the maximum distance (*i.e.* 1.0 for cosine similarity). Setting this threshold decides how many headlines we want to collate into a single event.

We rank the resulting headline-clusters using the headline tweet with the highest score inside a cluster, in this way, if the headline tweets do not cluster at all, the ranking of headlines will stay the same as in the previous step.

7. System output: Finally, after clustering and re-ranking headlines, we select the headline with the earliest publication time, and present the raw tweet text (without URLs) as a final system output *headline*.

We pool the keywords of the headline tweets in the same headline-cluster to extract tags (a list of keywords as a description of the event).

For selecting tweet IDs relevant to the extracted event, we use the IDs of the clustered headline tweets.

For extracting URLs of photos relevant to the event, we first check if the headline tweets have any *media_url* tags, and if not, we loop through the cluster (from stage 1) and pick out the media URLs. Restricting the number of media URLs to 1 or 2 directly affects the speed of the overall extraction process since we don't have to dive too deep into the previous (potentially large) clusters.

3.1.5 Development Evaluation and Parameter Settings

For development and parameter settings for pre-processing, clustering, and filtering thresholds, we used the ground truth provided for the 2012 US elections stream. Tables 3.3 and 3.4 show ground truth and our system output for one of the time windows in the evaluation.

For the rehearsal set, no ground truth was available. To evaluate performance on the rehearsal set, we use a Google search-based evaluation, similar to [117]. The extracted *headline* tweet is used as a query, and the presence of mainstream news articles in the results is used as a relevancy measure. We selected 100 headline tweets, the top-10 from the first 10 windows, searching for mainstream news articles. Using this semi-manual evaluation, 80% of our headline tweets had mainstream news article equivalents.

Parameter Sensitivity

Our system has a number of tunable parameters, that can be set depending on the type of tweet stream available.

Filtering tweets: The filtering of tweets by structure, where we reject tweets with more than 2 user mentions, more than 2 hashtags, or less than 4 text tokens had the most impact on the tweet-term matrix. Most noisy and less newsworthy tweets are filtered in this step. Due to this aggressive initial filtering step, other parameter settings downstream appear less dramatic.

Tweet length and structure: Relaxing the tweet length restriction from a minimum of 5 tokens to 3 tokens, includes an extra 500 tweets in the final tweet term matrix. The effect on the final output is low, but the number of tweets to process is reduced from 3,777 to 3,258.

Unigrams Vs bi and tri-grams: Keeping all other parameters fixed, and only altering the tokenisation to use unigrams as opposed to bi and tri-grams, results in far more tweets included in the tweet-term matrix (9,028 as opposed to 3,258). This significantly impacts processing time, and cluster scoring. In cluster scoring, using unigrams and averaging term scores produced better results, but maximum entity boosted term in bi-grams and tri-grams normalised by the cluster size was a more performant compromise between processing time and precision.

Processing time: On commodity hardware[10], the total processing time was approximately 1 hour for 24 hours worth of tweets. This processing is split into 96 time windows, updating every 15-minutes. The most computationally costly steps are the pairwise distance calculations for hierarchical clustering. Filtering tweets before this step has the most benefit for run times.

3.1.6 Challenge Evaluation

Systems were expected to produce a maximum of 10 events per 15-minute time window, for a total of 960. Three independent annotators used a web-based interface to rate outputs on a 5 point scale for Readability, coherence and relevance, and an optional image representing a story. Evaluators manually annotated five time windows. For each event in the 5 time windows, annotators first marked newsworthiness on a binary scale - discarding any instances with less than 2 votes. 70 breaking news events were included in this pooled set. Assessment where non-newsworthy stories included: Opinion article, Analysis Article, Speculative Article, Jokes, gossip, parody. Valid, newsworthy events are: major news stories, local news stories, photo-driven stories, announcements of future events, sub-events such as goals scored in a match. Full details of the rating scales and instructions provided to annotators, inter-annotator agreement and raw result tables are available in [104]. Here, Table 3.2 summarises the measures, what they were based on, and how they were calculated.

Each score was normalised by the maximum attained for each measure, and an aggregate score for a system (v) was derived by:

$$\text{Score}(v) = 0.25 \cdot R_{ref}(v) \cdot F_{ext}(v) + 0.25 \cdot Q(v) + 0.25 \cdot C(v) + 0.25 \cdot D(v) \tag{3.2}$$

[10]PC with 8GB Memory, non-SSD HDD, 2.7GHz CPU.

Measure:	Name:	Calculated by:
Newsworthy	Newsworthyness	Qualitative judgment, given positive and negative examples
R_{ref}	Reference Corpus Recall	Manual Match: 59 Ground truth events
P_{ext}	Pooled run Precision	Manual Match: 70 Participant-pooled events
R_{ext}	Pooled run Recall	Manual Match: 70 Participant-pooled events
F_{ext}	F-score of P_{ext} and R_{ext}	F1(P_{ext}, R_{ext})
Q	Readability	Qualitative, 1–5 Scale
C	Coherence	Qualitative, 1–5 Scale
D	Diversity	Qualitative, 1–5 Scale
I	Image Quality	Binary Relevant / Not Relevant, Average of 3 annotators

Table 3.2: Measures used in the challenge evaluation.

This score was used as the final system ranking. Table 3.6 lists the raw scores for the top performing systems.

Our system was ranked first using the combined measure. Overall, the top 3 systems were ranked in the same order using alternative score aggregations. The high recall for both the reference and pooled results, as well as high coherence, contributed most to the final ranking. The high scores for coherence and readability are due to selecting textually-rich tweet content as the headline after filtering uninformative and short tweets, as opposed to trying to generate a headline.

One drawback of this evaluation approach is that time is not accounted for. Systems that produce events significantly earlier or later than the mainstream news story are treated equally.

For setting parameters, we use the subset of ground truth events provided by the challenge organisers for the 2012 stream, a sample of which is shown in Table 3.3. For comparison, in Table 3.4, we show the top-10 events detected by our method (with parameters set as described in the previous section) for the same stream, for the time slot starting at 07-11-2012 00:00.

In Table 3.5, we show the top-10 events produced by our method for the 2014 stream (parameters are the same as for Table 3.4 run), for the time window starting at 25-02-2014 18:00.

	Headline	Keywords
1	Obama wins Vermont	Obama,Vermont,wins,projects,VT
2	Romney wins Kentucky	Romney,wins,Kentucky,projects,KY
3	Bernie Sanders wins Senate seat in Vermont	Sanders,wins,Senate,Vermont, independent,VT
4	Romney wins Indiana	Romney,wins,Indiana,IN

Table 3.3: Ground truth for the 2012 US elections Twitter stream for the 07-11-12 00:00 time slot.

Comparison with Other Systems

Table 3.6 lists the raw scores for the top-3 systems. In comparison with the other competing systems, there are some notable similarities and differences. The next best performing system, *RGU* [118], also used a similar n-gram clustering and ranking approach, but described fewer filtering steps, and did not perform additional merging of headlines in the same way. In RGU, events were merged based on URLs whereas our system performed hierarchical clustering.

The *math-dyn* [119] system used *Joint Complexity*, an information-theoretic approach, where each time window is a weighted graph of tweets and their similarities defined by *Joint Complexity*. The *math-dyn* system did not report heavy use of filtering in contrast to our system.

Data pre-processing steps that split tweets into parts, recombining them and filtering on platform-specific features have the most impact on results.

After the structure-based filtering, vocabulary filtering and other parameter choices such as bi-grams vs tri-grams have less impact, as the least newsworthy tweets have already been discarded in the earlier pre-processing step. In comparison with other systems, our approach contained more filtering and pre-processing steps.

3.1.7 Discussion

It is interesting to observe that the most effective approaches in the challenge all involved simple, n-gram models, and focused on document pivot techniques. Using original tweet text as the headline has the advantage of presenting users with interpretable headlines, but tweets may appear out of context and require some background knowledge for some events (*e.g.* cryptic abbreviations and acronyms can appear as hashtags).

Topic fragmentation (the same events being discussed across clusters) is still an issue.

	Headline	Keywords
1	WASHINGTON (AP) - Obama wins Vermont; Romney wins Kentucky. #Election2012	#election2012, @ap, ap, begins, breaking, calls, carolina, close, cnn, fox, georgia, indiana, kentucky, news, obama, presidential, projects, race, romney, south, vermont, washington, wins
2	Not a shocker NBC reporting #Romney wins Indiana & Kentucky #Obama wins Vermont	#obama, #romney, indiana, kentucky, nbc, reporting, vermont, wins
3	RT @SkyNewsBreak: Sky News projection: Romney wins Kentucky. #election2012	#election2012, @skynewsbreak, indiana, kentucky, news, obama,
4	AP RACE CALL: Democrat Peter Shumlin wins governor race in Vermont. #Election2012	#election2012, ap, bernie, call, democrat, governor, peter, race, sanders, seat, senate, shumlin, vermont, wins
5	CNN Virginia exit poll: Obama 49%, Romney 49% #election2012	#election2012, cnn, exit, obama, poll, romney, virginia
6	Mitt Romney Losing in Massachusetts a state that he governed. Why vote for him when his own people don't want him?	#Obama2012 #obama2012, governed, losing, massachusetts, mitt, people, romney, state, vote, want
7	Twitter is gonna be live and popping when Obama wins! #Obama2012	#obama2012, gonna, live, obama, popping, twitter, wins
8	INDIANA RESULTS: Romney projected winner (via @NBC) #election2012	#dumbasses, #election2012, @huffingtonpost, @nbc, indiana, projected, results, romney, winner
9	If Obama wins I'm going to celebrate... If Romney wins I'm going to watch Sesame Street one last time #Obama2012	#obama2012, celebrate, going, last, obama, one, romney, sesame, street, time, watch, wins
10	#election2012 important that Romney won INdependents in Virginia by 11 pts. With parties about even, winning Inds is key	#election2012, even, important, independents, inds, key, parties, pts, romney, virginia, winning, won

Table 3.4: Detected top-10 events using our method for the 2012 US elections Twitter stream for the 07-11-12 00:00 time slot.

This is most pronounced in news events that are discussed from multiple points of view. Different groups of people tend to discuss the same events in different ways. As well as fragmentation, some events are mistakenly merged into a single cluster (*e.g.* stories about Bernie Sanders and Peter Shumlin in Vermont elections.).

Another promising avenue could be to segment the stream by location or keywords, applying the system for extracting sub-events, then recombining the detected events.

Twitter is a highly-dynamic platform, with new features and conventions introduced on a regular basis. For example, in 2014, the practice of quote tweeting did not yet exist, and these "retweets with comments" were instead broadcast as public replies, and not as commonly used. This is one example of a new feature that wasn't present that will need to be accounted for if applying these techniques on a new dataset.

	Tweet Id and Headline	Keywords
1	438373491440500737 The new, full Godzilla trailer has roared online:	awesome, brand, full, godzilla, landed, new, online, roared, trailer
2	438372831081279488 At half-time Borussia Dortmund lead Zenit St Petersburg 2-0. #bbcfootball #ChampionsLeague	#bbcfootball, #championsleague, @bbcsport, borussia, dortmund, half, lead, petersburg, st, time, zenit
3	438373672412143616 Ukraine Currency Hits Record Low Amid Uncertainty: Ukrainian currency, the hryvnia, hits all-time low against ...	amid, currency, hits, hryvnia, low, record, time, ukraine, ukrainian, uncertainty
4	438372908814303232 Ooh, my back! Why workers' aches pains are hurting the UK economy	aches, back, economy, hurting, pains, uk, workers
5	438373369491505152 Uganda: how campaigners are preparing to counter the anti-gay bill	anti, bill, campaigners, counter, gay, preparing, uganda
6	438372882088226816 JPost photographer snaps what must be the most inadvertantly hilarious political picture of the decade	@jerometaylor, decade, hilarious, inadvertantly, jpost, photographer, picture, political, snaps
7	438375154008461313 Fans gather outside Ghostbusters firehouse in N.Y.C. to pay tribute to Harold Ramis	fans, firehouse, gather, ghostbusters, harold, nyc, outside, pay, ramis, tribute
8	438373191762059265 Man survives a shooting because the Bible in his top pocket stopped two bullets	@metrouk, bible, bullets, man, pocket, shooting, stopped, survives, top, two
9	438374254002700288 #Ukraine's toppling craze reaches even legendary Russian commander, who fought Napoleon	#ukraine, commander, craze, even, fought, legendary, napoleon, reaches, russian, toppling
10	438372863377408000 Newcastle City Hall. Impressive booking first from bottom on the left...	@robbrydon, booking, bottom, city, first, hall, impressive, left, newcastle

Table 3.5: top-10 detected events using our method for the 2014 Syria, Ukraine, Bitcoin Twitter stream in the 25-02-2014 18:00 time-slot, including the Tweet Id, extracted headline, tags. Event 10 was the only one that was not published in mainstream media.

Our approach did not address the problem of assigning relevant lead images to events, as this was a less significant aspect of the challenge, the image selected was the headline media URL if present, in the tweet metadata, with the rationale being that a textually relevant tweet would also have a relevant image.

Since 2014 a number of important tweets in the corpus (those included in evaluations with human judgments and tags) have been removed for various reasons. This tweet decay is of a similar rate to other Twitter-based collections, such as TREC.

Team	N_{ref}	R_{ref}	N_{ext}	P_{ext}	R_{ext}	F_{ext}	Q	C	D	$Image$
Insight	39	0.660	28	0.560	0.357	0.436	4.74	4.97	2.11	0.274
RGU	33	0.600	19	0.388	0.243	0.299	4.71	4.22	3.27	0.588
math-dyn	37	0.630	18	0.462	0.200	0.279	4.59	4.91	2.11	0.520

Table 3.6: Results for the top-3 systems. N_{ref} is the number of events produced for the *Reference* Recall, N_{ext} is the number of events from pooled submissions

This decay can affect the reproducibility of system performance scores in future. Removing these tweets from the evaluation makes the result sets fairer to new methods, but this also renders comparison to older implementations difficult.

The techniques we used and parameter settings suggest that a large proportion of the accuracy can be gained with appropriate platform-specific pre-processing. Performance may not generalise, however, when choosing similar settings for an event detection system on a different platform, even if the features appear similar.

The SNOW Data challenge produced a useful benchmark dataset for breaking news event detection on Twitter. Our proposed approach shows that simple, n-gram-based models with some platform-specific heuristics can create a strong baseline for comparing future systems

3.2 Effectiveness of Breaking News Event Detection

Breaking news event detection on Twitter has become an increasingly popular application area in the context of online journalism. A host of techniques have been proposed in the literature, but there are few shared test collections and little agreement on evaluation measures. This leads to significant difficulties when attempting to compare different systems.

Event detection systems are composed of multiple components performing functions such as filtering, feature extraction, and issuing alerts. These components are rarely evaluated separately, making it difficult to identify which techniques or parameter settings offer the most benefit and which are detrimental.

In the remainder of this chapter, we explore different experimental design choices for the task of breaking news event detection, using a new Twitter-specific ground truth collection curated by journalists. For a set of established baseline systems, we show that parameter choices involving thresholds, which are normally set at design time, can have a dramatic effect on system performance—evaluation parameters especially. Specifically, these are window parameters in methods that rely on sliding window techniques, and thresholds in automatic evaluations that measure text overlap.

We demonstrate that, with adequately-tuned settings, even relatively simple approaches can perform well. However, performance can suffer if parameters chosen at design time are not suitably adjusted in a deployed system, without updates to account for new data.

We propose a generally applicable approach to aggregating performance evaluations of different systems and parameters, that can be used for prioritising future development efforts or justifying the selection of one set of parameters over another.

3.2.1 Introduction

The detection of breaking news events has a long history in the Information Retrieval literature, where many standard collections draw on newswire sources. Strong user involvement in the evaluation process is desirable, but often impractical or prohibitively expensive [120].

Social media sources contain a wealth of timely information and user-generated content. Filtering relevant content, however, is not trivial. Significant demand exists for editorial support systems enabling journalists to work more effectively. Social newsgathering introduces a number of new challenges to breaking news detection tasks.

These include verification of content and sources, and timely filtering of large volumes of content. Furthermore, a breaking news alert issued hours after the story breaks will not be especially useful to a journalist, while false alarms will distract a journalist from more important reporting, fact-checking, and editorial work.

A recent review of popular tools used by journalists in [121] motivates research in developing better event detection and alerting systems. Only one event detection system [122] is mentioned in passing in the survey. This suggests that either state-of-art systems described in the literature have not been widely adopted, or that monitoring tasks are adequately served by tools internal to newsrooms or some other commercial services. Either way, it is clear that there is much work left to be done before journalists widely accept news event detection tools.

The problem of detecting breaking news events is related to previous work on Topic Detection and Tracking (TDT) [9]. Topic detection and tracking is still an active area of research, heavily focused on specific applications. The field continues to generate considerable interest, with many works claiming state-of-the-art performance.

Breaking news event detection focuses on delivering relevant alerts to journalists about developing stories, when information about "who" or "what" is often lacking. The task is more closely related to the real-time filtering task in the TREC Microblog track [123]. However, the interest profiles may not be defined a priori, since a breaking news event is relevant to a journalist regardless of their preference for topic coverage.

In particular, a common theme across the research area of event detection is the issue of defining an appropriate evaluation methodology. This is exacerbated by the fact that event detection systems are often complex, consisting of multiple distinct components, where each component typically has its own set of user-specified parameters.

The deployment of such systems presents non-trivial engineering challenges, regardless of the algorithms used. Differences in implementations between systems can hide the effects that researchers are most interested in studying [124], while the choice of unsuitable parameter values at design time may limit the subsequent usefulness of a system to journalists.

When deploying an event detection system, a number of fundamental questions arise. How do we know if the system is performing well? Which components of the system are contributing to good performance, and which components require the investment of additional effort and resources?

Answering these questions requires a detailed evaluation. However, conducting extensive live user studies can be expensive and time-consuming, while providing a system with frequent relevance judgements places an unnecessary burden on journalists.

Therefore, evaluations that rely on test collections remain important.

In order to evaluate systems in a realistic setting, we introduce the *Reportedly* test collection, which provides a new source of event annotations as curated by journalists actively monitoring Twitter for breaking news.

Unlike previous evaluations which have relied on newswire services and mainstream news articles to evaluate breaking news event detection on Twitter, we suggest that evaluation data derived directly from the Twitter activity of an online newsroom is more appropriate in this context, given the emphasis that is increasingly placed by journalists on using Twitter to both find sources for, and disseminate breaking news [125]. We make this collection publicly available for future research in this area[11].

Our contributions are twofold. Firstly, as described above, we provide a domain-specific collection of tweets and ground truth events, curated by journalists, also potentially useful for other tasks beyond breaking news detection. Secondly, we use this test collection to explore parameter choices often omitted in previous literature, highlighting significant differences between systems and components.

Our evaluations show that even simple detection systems can perform well with suitable parameter settings. However, the choice of parameters is not always straightforward, with certain experimental design choices during the evaluation phase leading to significantly different conclusions being drawn regarding the performance of a detection system.

3.2.2 Task and Scope

The task of breaking news event detection on Twitter depends on the precise definition of an *event* and the specific needs of the journalist. For example: for a sports journalist, a breaking news event might be a "goal scored" event [126]. Other types of journalists covering specific areas or topics may have different requirements for a system.

We will not explore different requirements for specific news niches here, but rather we will focus on *breaking news* reporting in general. In our case, we define *breaking news* as any activity or information identified by journalists, that warrants their immediate attention.

In contrast to traditional pool-based evaluations and annotators producing relevance judgements, we rely on the output of a real news team who are actively producing breaking news alerts. We describe this data in more detail later.

[11] `https://github.com/igorbrigadir/newsir16-data`

In the existing literature, a wide range of information needs are classified under "breaking news". This creates a diverse range of tasks and definitions. The only common feature across the literature is that the task is somehow *temporal* in nature and relates to some human activity [127]. Therefore, we broadly define the task as follows:

Systems are presented with a continuous stream of time-ordered documents (*e.g.* tweets). Based on certain features of these documents, the system issues "alerts" to a journalist. An alert at time t containing a text description is compared to a ground truth text close to time t.

The above formulation can be related to other tasks, but differs in a number of important ways:

Topic detection and tracking (TDT): Here an *event* is "a particular thing that happens at a specific time and place, along with all necessary preconditions and unavoidable consequences."[128]. This work only relates to the online New Event Detection subtask (NED), where "stories", or in this case tweets, are presented in sequence to the system, and the system makes a decision to issue an alert or not. The difference between NED and our experiments is that tweets are presented to the system in batches, using a sliding window as opposed to one by one. In tracking tasks, systems attempt to retrieve *all* relevant, non-duplicate updates to a developing topic, whereas in breaking news detection the focus is on the initial alert, and minimizing subsequent false alarms.

First story detection (FSD): FSD is largely synonymous with NED but more commonly associated with clustering approaches [55]. In contrast to clustering, we focus on anomaly-based approaches. Outputs of clustering-based and anomaly-based systems are equivalent—in that both issue "alerts" to users, and the same evaluation can apply to both. FSD is sometimes used to describe Retrospective Event Detection (RED) where the entire corpus is available to a system, and the task is to pinpoint exactly when an event emerged, knowing what to look for ahead of time, or attempting to discover previously missed events [129]. First Story Detection, strictly speaking, describes a system that must classify each story in a stream as *first* or *not first* [40]. The end result is a classification of documents. Rather than classifying documents, breaking news event detection seeks to generate an alert, that may or may not be based on a new document. The distinction is important, as activity measures that do not involve document features can be used, without requiring a *first story* to be published, *e.g.* a significant increase in some activity measure, as opposed to a new article.

Event Detection Components

Breaking news detection is a complex task that necessitates separating functionality into several components. While some systems may perform the functions of multiple components in a single step, these can be treated as separate sub-systems. The advantage of organising a system in this way is a greater degree of interoperability and more interpretable evaluation. The effects of changing parameters in one sub-system can be measured on the overall task, and efforts can be concentrated on components that have the greatest impact on results. We categorise components into four broad areas:

1. *Source data:* What data is considered as input? What is filtered on arrival?
2. *Evaluation:* How is performance measured? What is considered a success?
3. *Feature extraction:* What features and what representation is used? What counts?
4. *Detection:* How are events generated by the system and alerts issued?

Decisions about the source data and initial data filtering are often overlooked as significant "parameters" of a system. In our experimental setting, this translates to which tweets and accounts the system ingests.

Extracting features is another labour intensive part of system design and development. Often features are selected at design time and will differ significantly between systems. Decisions on how to treat Twitter specific mentions or hashtags, or how to represent textual content are also parts of the *feature extraction* component.

The detection step is commonly cast as an *anomaly detection* problem, but other formulations view the problem as a classification task, or a clustering task. Regardless of the approach, all breaking news detection systems must somehow alert a journalist. Alerts can be actively pushed to the user, or can be presented via passive dashboards that display metrics or items which require further attention.

Even the most complicated systems can be broken down into several component parts. For breaking news event detection—each component has a number of parameters that could be tuned. We focus on the feature extraction and evaluation parameters, as opposed to the detection techniques themselves.

Goals

Given the above problem description, and an appropriate source dataset, we outline an approach for configuring and evaluating a breaking news detection system as follows:

1. Enumerate and select all the tuneable system parameters (Window size, thresholds, etc) in systems.
2. Select a single dependent variable, such as the number of detected events.
3. Generate and gather results for all the systems.
4. Estimate System Component Effects with a General Linear Model.

The component and parameter effects can be used to explain the selection of a particular system over another, or to guide future development efforts. Our goal is not to create a state-of-the-art breaking news detection system, rather, it is to perform a controlled experiment on a range of commonly fixed parameters, such as detection windows, text overlap, and alternate ways of detecting anomalies in time series.

3.2.3 Source Data

The input data is an important system parameter. The importance of filtering newsworthy sources is discussed in more detail in Chapter 2. Strategies for the acquisition of Twitter data can be broadly categorised into three types:

Sample streams: These include subsets of the *firehose* which represents random samples of all public tweets. Typically these stream sources use Twitter's 1% sample stream. The 10% *Gardenhose* stream is not generally available to researchers, and the full firehose of tweets is rarely used as source data.

Keyword-based streams: Constructed from sets of keywords, either retrospectively via the Twitter Search API, or in real-time through filtering the Streaming API. The selection of keywords can have a significant effect on results, as well as the changing nature of the Search API—which optimises for relevance as opposed to retrieving all matching tweets. The upper bound for the volume of tweets retrievable through the Streaming API, filtering on keywords is still 1% of the Firehose. Geotagged tweet searches and filters can also be considered as "keyword" based, where tweets are matched by location instead of content.

Curated streams: Attempt to overcome some of the disadvantages of search and sample streams by monitoring all tweets from a limited set of accounts, as opposed to sampling a subset of tweets from all accounts. This approach is sometimes referred to as an *expert sample* [130]. One drawback of using curated streams sourced from journalists is the lack of clear criteria for including or excluding sources. Journalists may have different reasons for including or excluding a source when monitoring Twitter activity.

To collect the test data for our evaluations, we made use of a curated stream of users identified as "newsworthy" by journalists at Reportedly. Our shared collection consists of 3,274,088 tweet identifiers.

The corresponding tweets should be retrievable in approximately 35 hours when optimally using the Twitter API rate limits. Twitter imposes restrictions on the ability to share full tweet payloads. This is a major issue when sharing tweet collections for the purpose of reproducibility.

Following their terms of service, collections must consist of tweet identifiers only, omitting actual tweet content and other metadata. An advantage of this approach is that users included in collections maintain some control and consent over the use of their data. If an account becomes private, or tweets are deleted, they will no longer be retrievable. Users are rarely notified about the use of their data in research, so this approach strikes some balance between making data freely available, while also respecting user privacy.

Reportedly Dataset

The source stream of tweets for the Reportedly collection contains $\approx 30,000$ user accounts organised into user lists by journalists. These lists were assembled in order to cover a specific topic such as "US Politics" or geographical regions such as "UK". Reportedly journalists monitor these lists for developing stories, summarizing, verifying information and tweeting updates to a story.

The Reportedly online newsroom project[12] was active from December 2014 to August 2016. A unique aspect of Reportedly was its strong emphasis on social media platforms, especially Twitter. The reliance on user-generated content, curated lists of trustworthy accounts, and effective use of Twitter-specific features makes this an attractive source of ground truth for Twitter event detection tasks. Specifically, the use of hashtags, quoted tweets, user mentions, and verified sources offer a much better match for automatically detected events than traditional newswire services, which have been previously used as a ground truth source.

If the choice of external evaluation data matches the system input data, simpler detection and evaluation techniques can be used.

While these sources are also present on Twitter, newswire services generally do not focus on Twitter-specific activities, and their reports lack the use of important mentions or hashtags relating to events.

[12] `https://reported.ly`

Row	Time	Event description and number tweets in thread	
0	09:20	Flooding and Landslides in Japan.	6
1	10:00	Updates on refugees in Greece to Hungary.	8
2	10:20	#YouStink Protests in Lebanon in response to waste crisis.	4
3	10:30	Turkey's offensive on the PKK and deportation of journalists.	7
4	11:20	European Parliament backs Commission's refugee plan	2
5	13:30	Protest in Baltimore at hearing for #FreddieGray case.	6
6	15:30	Deteriorating humanitarian situation in Yemen	2

Table 3.7: Text descriptions of example events as shown in Figure 3.1.

In contrast, Reportedly journalists directly used tweets to highlight and disseminate information related to breaking news events, which often included relevant trending hashtags or mentions of important user accounts. An advantage of using these sources is that event detection methods have access to the same feeds as the journalists, and Twitter-based ground truth maintains Twitter-specific features and conventions.

In the Reportedly Twitter data, "major" news events are organised as reply threads, where an initial "breaking" tweet posted by a journalist is followed by several updates (*i.e.* replies to the original tweet) as the story develops. We treat the time window of the initial announcement tweet as the detection window for the purpose of evaluation.

In this work, we selected September 2015 as the evaluation period to overlap with the NewsIR'16 shared collection [131], which also includes news articles and blog sources. While we do not incorporate these sources in our experiments, we envisage that the two collections could be combined for further evaluations in future work. This time period contains 227 major news events. As an illustrative example, Figure 3.1 shows a small sample of 6 such events on September 10th between 09:00 and 16:00 UTC. See Table 3.7 for event descriptions. Note that, for event detection components which require an initial training phase, we use Reportedly tweets from August 2015 as a "warm-up" period.

While the Reportedly dataset provides a suitable Twitter-specific set of manually curated events, there are a number of issues for future work and reproducibility such as the limited coverage of events during off-peak days, and limited time span of activity. The source set of tweets we use is English language only, and is derived from manually curated lists of users journalists built over time, with unknown criteria for inclusion or exclusion *i.e.* when including a Twitter account in a list for monitoring, the reason for inclusion is not available. Therefore, we exclude time windows where the Reportedly team was not active, so that systems are not penalized for false alarms when there is no ground truth available.

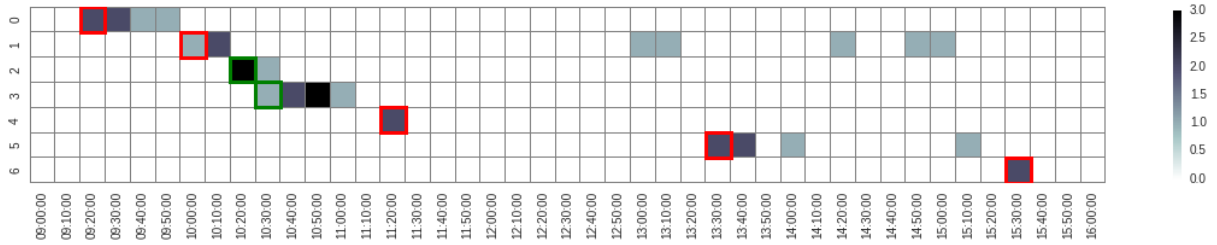


Figure 3.1: Each row in the grid is an event, and each column is a 10-minute time window. Darker squares represent time periods in an event with more ground truth tweets. Green and Red highlights show Hits and Misses by one of the systems. Some events, like Row 4 consist of a single breaking news update, others like 1 or 5 have periods with no updates after the initial announcement.

In total, there are 514 tweet threads, varying in length. Additionally, there are a number of summaries written by journalists about 95 such threads, and a further 46 longer articles providing a wider context about events.

The descriptions and articles were not used as part of our evaluation but are included in the dataset as they are useful for interpreting results and may be used for different tasks, such as abstractive summarization, or automated fact-checking.

3.2.4 Evaluation Strategies

The vast majority of Twitter-based event detection evaluations are based on criteria such as Precision, since complete annotations on all relevant events are not present in the data [132]. The large number of documents in tweet datasets make it extremely difficult to create exhaustive relevance judgements. Manual or qualitative evaluations focusing on a handful of events are commonly used for smaller studies, while larger ground truth event sets are derived from full articles coming from newswire services, such as Bloomberg, New York Times, Reuters or AP *e.g.* TDT corpora [40].

A major limitation when relying on newswire content to evaluate Twitter event detection relates to the difference in language, vocabulary, and style used in full-length articles, when compared to 140 character tweets. Newswire text lacks Twitter-specific vocabulary terms and conventions, leading to term matching based evaluations that do not account for hashtags, mentions, and abbreviations – these can be central to a breaking news event on Twitter.

Evaluations involving human annotation, such as the TREC microblog track [33], are expensive to perform, but are domain-appropriate, since human annotators label tweets, and the language in the ground truth matches the input data.

In order to avoid the problems of evaluating breaking news events across domains (*i.e.* training on Twitter data, but evaluating on newswire article data) and the cost of manually constructing an annotated corpus of labelled events.

Evaluation Criteria

We process the stream of input Tweets, and ground truth Tweets with a sliding window. Systems are presented with a batch of Tweets, in sequence, and are expected to produce a number of alerts, which are matched to unseen ground truth Tweets. Systems are also provided with a budget of alerts (maximum number of alerts) and other hyperparameters.

This evaluation setting simulates the requirements of a live system, where future data is never available. Cross-validation is not appropriate for our evaluation setting, as train and test splits are not temporally ordered.

An *event* is defined as a set of tweets created by journalists that has a beginning, a number of updates, and ends when coverage stops. Events are extracted from Reportedly reply threads - where updates to a developing story are grouped in conversation threads, highlighting important developments. For evaluation purposes, the start of an event is the timestamp of the first tweet in a thread, and the end of an event is the timestamp of the last tweet in a thread. The detection period for the "first story" is the time window when the first post was made. For example, if the stream is being evaluated with a 30 minute time window, and Reportedly started a new thread at 13:37, a system must issue an alert between 13:30 and 14:00 to be valid.

Our first evaluation criterion measures whether or not a system can successfully detect significant events which overlap temporally with those events identified by Reportedly. Formally, we say that a *detection* occurs when a system issues an alert in the same time window as a newly created thread by a Reportedly journalist, otherwise the alert for the event is treated as a *miss*. A *false alarm* is a system alert issued in a time window when there is no ground truth alert.

In our evaluation model, we concentrate on *detections* as opposed to misses. The rationale being that the value of a successful alert (*i.e.* not missing an important breaking news event) is much higher to a journalist than the cost of a false alarm, which may be read and dismissed quickly.

All events in our dataset are weighted equally. Quantifying exactly how valuable a detection is, or how much a miss or false alarm costs is subjective and beyond the scope of the evaluation.

While our baseline systems produce an output in the form of an anomaly score, they differ in how they treat the input signal and how they need to be tuned. Therefore, we set alert *budget* thresholds so that each combination of a signal and detection approach produces the same number of alerts. The number of alerts is based on the number of events present in an initial "warm-up" period. In this way, all systems will produce approximately the same amount of alerts. For example, 289 unique events are present in the warm-up data, but rather than targeting only this number of alerts, we consider three different budget levels:

1. *Low budget*: 10% fewer alerts than the number of events in the ground truth.
2. *Medium budget*: the same as the number of alerts in ground truth.
3. *High budget*: 10% more alerts than the number in the ground truth.

As we see later, this parameter choice around budget is one that significantly affects system performance.

To check that the alert event matches with ground truth event, we introduce a second evaluation criterion: *event text matching overlap*. This criterion is used to ensure that an event detected by a system genuinely matches the event present in the ground truth, and is not the result of one or more unrelated phenomena (*e.g.* viral memes on Twitter). The event text matching parameter considers the text overlap between the ground truth tweets and the alert text.

Jaccard Similarity is used to determine how much of the text overlaps between the ground truth tweets, and tweets within the alert. Tweet text is pre-processed to remove numbers, URLs and stopwords. Tweet specific features such as mentions and hashtags are preserved. If there is insufficient overlap, the alert is treated as a false alarm. Since our goal is to generate an alert, we do not require a system to generate a summary or explanation, a system is only required to produce the text of the tweets within the alert time window.

We consider three increasingly conservative matching levels: Minimal (0.20), Medium (0.50), and High (0.80). As news tweets are short, but keyword-dense, we found that even a relatively low text overlap (0.2) can still be meaningful, based on a small manually examined sample of alerts. However, these values were selected to investigate a range of performance settings. Quantifying how this measure correlates with a journalist's sense of relevancy is out of scope, and left for future work.

Estimating Component Effects with General Linear Models

Typically, system-based evaluations would use a *t-test* to determine significant differences in the effectiveness of a new system and an existing baseline. A system is treated as a "black box", where individual components and parameters are not taken into account.

The large parameter space we explored presents a problem for interpretation. Given the same data and pre-processing choices, which component parts of a system contribute most to overall accuracy? How sensitive are the evaluation results to individual parameter choices?

To answer these questions and isolate component effects, we adopt the methodology suggested in [133]. We fit an Analysis of Variance (ANOVA) - a linear model, using the results of all the runs with family-wise Tukey multiple comparisons of means adjustment. The *family-wise* error rate is defined as the probability of observing a false positive (a seemingly significant improvement when there is none) in k experiments [133]. The choice of defining a *family* of tests where p values should be adjusted is somewhat subjective, and there is little agreement on what a family of retrieval experiments should consist of. Here, we define *family* to be a set of *systems* or *evaluation* parameters, as detailed in Table 3.9. We then formally model system performance as:

$$\text{detections} = \text{feature} + \text{window} + \text{detector} + \text{budget} + \text{text match} \tag{3.3}$$

where a *detection* occurs when a system issues an alert in the same time window as a newly created thread by Reportedly, and the alert text also overlaps with the ground truth text.

3.2.5 Baseline Systems

In order to demonstrate our evaluation approach and highlight the importance of hyperparameter settings, we build a number of baseline systems composed of different types of features, as detailed below.

In a Twitter event detection setting, the most widely-used source of features is tweet text. Metadata regarding a user and multimedia content is also available but may be missing. More advanced approaches may also use pre-trained models to assign labels to tweets. In our evaluations, we compare the performance of systems using Twitter-specific text extraction, which attempts to extract all "visible" text in a tweet (*e.g.* appending quoted tweet text).

A number of new features have been introduced to the Twitter platform in recent years. Along with changes to the API and general usage conventions, these features offer an opportunity to extract additional signals useful for event detection. For example: Introduced in April 2015, quoted tweets can be treated as a retweet with a comment or as a reply. For systems that rely on retweet, mention or reply based features, these tweets can be treated as either. In [25] the effect of quoted tweets on political discourse is explored, characterising quoted tweets as "opinion", "public reply" or as a means of "forwarding" and sharing tweets. In our setting, we append the quoted tweet text to the original tweet, treating this as a longer document. These and other, future changes to the platform should also inform adjustments to both systems and evaluation settings.

Specifically, the following *types* of features are explored:

1. tf: Term frequency features are any features derived from counts of unigrams, n-grams, hashtags, mentions, named entities or other units of text.

2. df: Document frequencies are based on the whole document, influencing inverse-document frequencies, and tweet counts over time. Counts of "unique" tweets, retweets, replies or quotes are df features.

3. lf: "Label"-based features are any features that rely on some metadata about a document, either manually or automatically assigned: such as topic labels, location or metadata associated with the tweet author's account

4. $similarity$: Similarity or Distance-based features require a defined measure of affinity between documents.

Other features used in related work can be derived from these basic counts. For instance, tf-idf based signals rely on tf and df, topic modeling and clustering approaches rely on tf-idf and lf (where topic probabilities are treated as topic label features). The informal nature and restricted length of tweets impact tf-based features: abbreviations, misspellings, hashtags, mentions, and Twitter-specific slang can result in a very large vocabulary. Therefore, tf features are typically filtered and pre-processed using stopwords or frequency thresholds (*e.g.* discarding rate words). In our experiments, we filter stopwords, numbers, special characters, and URLs, but preserve tweet specific hashtags and mentions.

Approaches that cluster individual documents using terms or other features are referred to as *document-pivot*, while *feature-pivot* approaches model the bursty activity of the extracted features [134].

Given these 4 feature types, we generate 6 features: 3 count-based document-pivot, 2 lexicon-based feature-pivot, and 1 similarity-based feature.

Count-based signals

Count-based features are surprisingly effective, but often overlooked as baselines. We tested three types of count-based features:

1. *df-count*: The number of tweets within a time window.
2. *uf-count*: The number of unique users who tweeted at least once within a time window.
3. *lf-count*: The number of "lists" that had new tweets within a time window. Since our set of users is derived from lists curated by journalists, and many accounts contribute to several lists, this signal works to amplify the activity of important accounts. For example as a member of six lists of interest, a tweet from Iranian President (@hassanrouhani) would contribute to all six.

Both lexicon-based signals and count-based signals can be viewed as *feature-pivot* techniques.

Lexicon-based signals

CrisisLex [135] is a lexicon of terms relating to humanitarian crises and natural disasters. As a simple baseline, we construct a time series of the counts of CrisisLex terms over time. There are 288 words in this lexicon after pre-processing, where we split multi-word phrases like "flood victims" into individual tokens such as "flood" and "victims".

As an alternative to CrisisLex, the *FirstDraft* news project[13] has proposed a set of 88 unique terms to use when finding breaking news on Twitter. Journalists frequently use Tweetdeck[14] columns to filter tweets based on individual search terms. We include this list as a second lexicon-based baseline in our evaluations. We refer to the two lexicon methods as **crisislex-sum** and **fd-sum** respectively.

[13]https://firstdraftnews.com/how-to-find-breaking-news-on-twitter-social-media-journalism/
[14]https://tweetdeck.twitter.com

Content diversity-based signals

The **tweet-sim** measure attempts to capture the extent to which multiple users are talking about the same subject in their tweets during a given time window. To measure diversity, we calculate the mean cosine similarity between all unique pairs of tweets for a fixed time window. Given a set of documents D in a time window, the diversity is defined as:

$$diversity(D) = -\frac{\sum_{i,j \in D, i \neq j} cosSim(D_i, D_j)}{\sum_{i=1}^{|D|-1} i} \tag{3.4}$$

where $cosSim(D_i, D_j)$ is the cosine similarity of $tf\text{-}idf$ vectors of documents i and j in a time window. As this signal is based on document similarities, it is a *document-pivot* approach. This feature was originally developed in [91] and is discussed in Section 4.3.

More advanced similarity-based measures using word embeddings [136] can be used, however, for our experiments, we chose a simpler approach, as training word embeddings is challenging in a streaming setting, and different embedding approaches introduce yet more hyperparameters to evaluate. Chapter 4.2 explores the use of word embedding approaches for breaking news *tracking*.

Time Series Construction

Sliding window based-approaches represent a popular choice for dealing with evolving streams of features: the key parameters requiring tuning for these are the size of the window and the update rate. The size of a window can be fixed, or adaptive, and consider time or number of elements. In our evaluations, we adopt a *tumbling* window, where the window size and update rates are the same: windows of 5, 10, 30, and 60 minutes. For each feature, we create a time series using a sliding window, producing a time series.

Anomaly Detection

Given a time series constructed from one or more features acquired from Twitter data, a host of online anomaly detection approaches can be used to detect a breaking news event and subsequently issue an alert. In our experiment we test four state-of-the-art anomaly detection techniques: CAD OSE, Skyline [137], Numenta HTM [138], and EXPoSE [139]. The anomaly detection approach itself is not important in our evaluation, the ability to process a signal incrementally is the only requirement.

3.2.6 Results

Based on the event detection components, the Reportedly dataset, and the evaluation criteria as described in the last two sections, we conducted a comprehensive set of experiments to evaluate the performance of breaking news event detection techniques.

Table 3.8 provides a summary of the best parameter choices for each group. The best-performing parameters were those with a high alert budget, and minimal text overlap. However, given the high budget of alerts increases the total number of generated alerts, increasing the false alarm rate.

Overall, the best performing features were *uf-count* — a count of unique users in a time window, and *fd-sum* — a limited set of keywords derived from suggested search terms journalists use.

Window	Feature	Detector	Detected	Missed	False Alarm
60 Min	uf-count	EXPoSE	209	16	119
30 Min	fd-sum	skyline	153	72	225
10 Min	fd-sum	contextOSE	46	179	21
5 Min	lf-count	contextOSE	28	197	121
60 Min	**uf-count**	EXPoSE	209	16	119
60 Min	**fd-sum**	EXPoSE	201	24	119
60 Min	**df-count**	EXPoSE	197	28	94
60 Min	**tweet-sim**	contextOSE	188	37	207
60 Min	**crisislex-sum**	EXPoSE	180	45	42
30 Min	**lf-count**	skyline	126	99	153
60 Min	uf-count	**EXPoSE**	209	16	119
60 Min	tweet-sim	**contextOSE**	188	37	207
30 Min	fd-sum	**skyline**	153	72	225
60 Min	uf-count	**numenta**	143	82	94

Table 3.8: A sample list of single best performing systems for each group of parameters, in terms of detections, misses, and alerts generated, given a high budget and a low text overlap threshold. There were 225 event threads in the evaluation window.

The worst performing feature *lf-count* created problems for setting thresholds. Training a threshold on the warm-up with this feature failed to generate any alerts in some runs. Overall, the "activity of lists" was much more sensitive to concept drift, where activity varied significantly over time.

As the feature considers labels derived from topics and locations, this suggests that approaches that rely on some form of classification of users or tweets would benefit from adaptive approaches, *e.g.* retraining classifiers, topic models and thresholds periodically on new data, as opposed to relying on a static pre-trained model.

In terms of timeliness of alerts, the best performance was achieved with long 60-minute time windows. Hourly alerts for breaking news may be useful to some, but given how quickly stories can develop, they may not be especially useful for journalists.

For our experiments, the alert text (and processed ground truth text) consist of a bag of words. This makes controlling the experiment straight-forward, but in reality, this may not be the best way to present alerts. A separate component to generate a summary description of a detected event is outside the scope, but existing abstractive summarization approaches are good starting points.

The overall best performing combination for each *system* parameter is listed in Table 3.8. Groups highlighted in bold, show a ranking of each system parameter. For example, if a 10-minute window is required, the *fd-sum* feature with *contextOSE* detector is the optimal choice, or if the only available feature is tweet counts (*df-count*) the best choice is a 60-minute window, with the EXPoSE detector.

Detections are true positives, misses or false negatives are events when a system failed to issue an alert, false alarms or false positives are times when systems issued alerts that did not match up with the ground truth. As our ground truth is not exhaustive, some of these instances could have been valid alerts.

Since we are interested in quantifying the effects of different parameter choices, ranking best-performing systems is not enough. We would like to know if these differences between parameter choices are significant, across different combinations of system and evaluation parameter choices. This exhaustive testing can lead to the multiple comparisons problem. This issue arises when looking for significant results after tweaking multiple parameters and testing each one. The more comparisons between systems are made, the more likely a statistically significant improvement will be observed. This can be eliminated by adjusting p values for multiple comparisons, requiring stronger evidence of improvements. To this end, we modelled system performance as a linear combination of parameters, and performed a post hoc analysis to determine which choices are significant. We chose the number of detections as the target variable of interest. Details are given in the next section.

Each parameter setting listed in Table 3.9 is categorical, and examining the linear model coefficients can help identify the most important parameter choices. The estimate column in Table 3.10 is the difference in detections over the base group (intercept) – a system using a **5 Min** window, **lf-count** feature, **skyline** detector, with a low budget, and a high text overlap threshold.

Linear Model Evaluation: System Component Effects

Type	Parameter	Choice
System	Time window	5 Minute Window, 10, 30, 60
System	Feature	crisislex-sum, fd-sum, tweet-sim, df-count, uf-count, lf-count
System	Anomaly detector	CAD OSE (contextOSE), Numenta, Skyline, EXPoSE
Evaluation	Alert budget	Medium (match event volume in previous month), Low (-10%), High ($+10\%$)
Evaluation	Event text overlap	Minimal (0.2), Medium (0.5), Max (0.8)

Table 3.9: Summary of parameter choices considered in experiments. Time window and text overlap parameter ranges were chosen to match with values reported in other end-to-end systems.

Coefficient	Estimate	Std. Error
window: 60 Min	59.046	3.184
detector: contextOSE	35.435	3.184
detector: expose	33.625	3.184
window: 30 Min	33.472	3.184
textMatch: min	23.142	2.757
textMatch: med	20.979	2.757
feature: uf-count	20.236	3.9
feature: df-count	17.813	3.9
detector: numenta	15.019	3.184
feature: fd-sum	13.785	3.9
budget: high	13.608	2.757
feature: crisislex-sum	12.542	3.9
budget: medium	5.983	2.757
window: 10 Min	4.153	3.184
feature: tweet-sim	2.201	3.9

Table 3.10: Linear model coefficients, with a 5 minute window, lf-count feature with skyline detector, maximum text match and low budget as intercept. The Estimate column can be interpreted as the difference in number of detections, given a change in the parameter. A window size of 60 minutes is better than 30 minutes, which is in turn better than 10 minutes.

The heatmaps shown in Figures 3.2 to 3.4 outline the differences between systems using different parameter settings. Best performing to worst performing parameters are listed top to bottom, and largest difference to smallest difference from left to right.

Each individual heatmap cell shows the difference in detections from Tukey HSD multiple comparisons of means test (See 3.2.6 for details). Significant improvements at $p < 0.001$ are highlighted in red, while the cell colour and corresponding values indicate the differences in detections.

For example, in Figure 3.2, choosing a **30 Min** window (2nd row) instead of a **10 Min** window (2nd row, 2nd column), shows a significant difference of 29.32 detections, whereas the difference between a 10 Min window and a 5 Min window (4.15) is not significant.

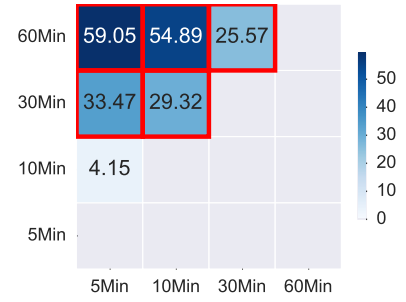


Figure 3.2: The differences between systems altering sliding window length are all significant, apart from the difference between a 5 Min window and a 10 Min window. There is no significant difference in this case.

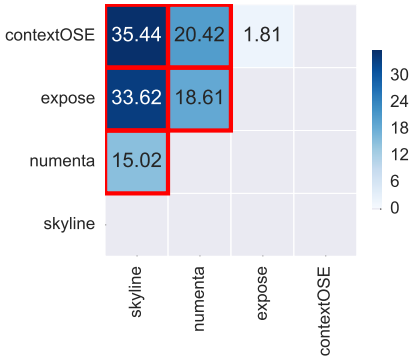


Figure 3.3: The choice of anomaly detector is a significant factor in a system, but the difference between *contextOSE* and *EXPoSE* is not significant.

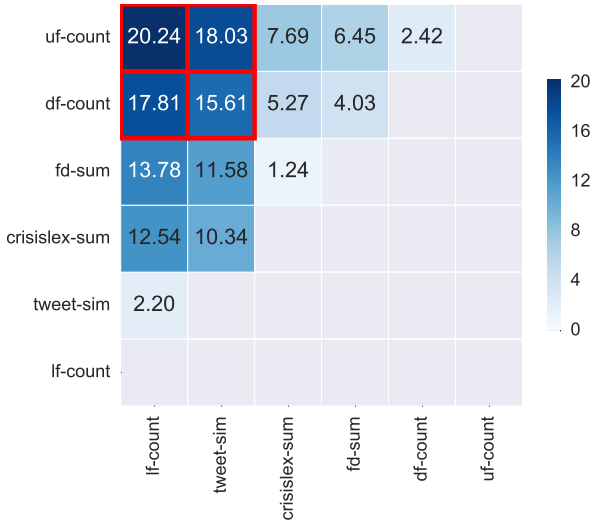


Figure 3.4: Effects of using different features. Using **fd-sum** instead of **crisislex-sum** shows a slight improvement of 1.24 detections, but this difference is not significant. Systems using **uf-count** feature instead of **lf-count** detect 20.24 more events, a significant difference.

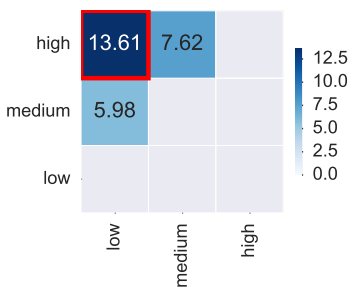


Figure 3.5: The alert budget limits the number of alerts a system can produce. A significant difference of 13.61 detections exists between the low budget (10% fewer alerts than the ground truth in the previous month) and high budget (10% more). The 7.62 difference between high and medium is significant but at $p < 0.05$ and difference between medium and low settings is only marginally significant ($p = 0.077$)

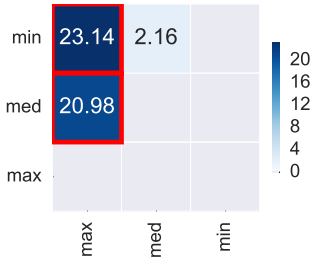


Figure 3.6: A high threshold value (0.8) for text match in evaluation produces significantly worse results than medium or minimum text overlap, but there is no significant difference between the medium (0.5) and minimum (0.2) settings.

It is worth noting that there are even more sources of variance between systems that our approach does not consider. For example, all *events* have their own noise and variance. This problem is explored in detail in [140]. Just as some *topics* in TREC collections are generally hard to perform well on, some events in our ground truth are also challenging to detect.

To find "difficult" and "easy" to detect events, we count how many different system variants detected each event, producing a ranked list. Highest ranked events are, therefore "easy" in the sense that many combinations of system parameters (even sub-optimal ones) still result in successful output. Lowest ranked events are events that very few variants were able to detect.

Three *events* (tweet threads) in particular, were not detected by any combination of feature or detector. One corresponded to a background summary of events relating to the rescue of refugees in the Mediterranean[15], another event reported on violence in Central African Republic[16], and the third was an update regarding flooding in Japan with new satellite imagery[17].

In contrast, events such as the release of Kim Davis from jail[18] and #YouStink protests in Lebanon[19] were detected by many system variants.

[15]Difficult event 1: `https://twitter.com/reportedly/status/648834211705659392`
[16]Difficult event 2: `https://twitter.com/reportedly/status/648785246691983360`
[17]Difficult event 3: `https://twitter.com/reportedly/status/642495636718288896`
[18]Easy event 1: `https://twitter.com/reportedly/status/641308170124689408`
[19]Easy event 2: `https://twitter.com/reportedly/status/638772034798026752`

Significant differences in performance are observed when adjusting alert budgets and text matching to ground truth. This suggests that these parameter choices can have a noticeable effect on results. While increasing the window length can help detect more events overall, whether or not this is useful is debatable. Alerts issued up to an hour late are unlikely to be useful in breaking news scenarios.

Model and adjusted p-values for all pairwise comparisons between all systems in our evaluation are listed in Figure 3.7.

```
lm(formula = detected ~ window + feature + detector + budget +
    textMatch, data = results)

Residuals:
    Min      1Q  Median      3Q     Max
-81.191 -19.565  -3.138  15.615 119.425

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -50.432      4.503 -11.200  < 2e-16 ***
window10Min             4.153      3.184   1.304  0.19249
window30Min            33.472      3.184  10.513  < 2e-16 ***
window60Min            59.046      3.184  18.545  < 2e-16 ***
featuretweet-sim        2.201      3.900   0.565  0.57254
featuredf-count        17.813      3.900   4.568 5.66e-06 ***
featureuf-count        20.236      3.900   5.189 2.64e-07 ***
featurecrisislex-sum   12.542      3.900   3.216  0.00135 **
featurefd-sum          13.785      3.900   3.535  0.00043 ***
detectornumenta        15.019      3.184   4.717 2.80e-06 ***
detectorexpose         33.625      3.184  10.561  < 2e-16 ***
detectorcontextOSE     35.435      3.184  11.129  < 2e-16 ***
budgetmedium            5.983      2.757   2.170  0.03031 *
budgethigh             13.608      2.757   4.935 9.65e-07 ***
textMatchmed           20.979      2.757   7.608 7.38e-14 ***
textMatchmin           23.142      2.757   8.393  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 33.09 on 848 degrees of freedom
Multiple R-squared:  0.4769,Adjusted R-squared:  0.4677
F-statistic: 51.54 on 15 and 848 DF,  p-value: < 2.2e-16
```

Figure 3.7: Linear model fit.

3.2.7 Related Work

Test Collection-Based Evaluation

Offline evaluation with reusable test collections is the dominant approach for evaluating effectiveness in Information Retrieval [141]. TREC-style test collections for event detection such as [9] define a fixed collection of documents, queries, and evaluation metrics. Participating teams perform experiments and submit results, and relevance judgements are derived from pooling results from participating systems.

Other shared collections that are partially suitable for event detection tasks include the SNOW 2014 data challenge [104] where the ground truth consists of human annotations, trending topics from Twitter where the ground truth originated from a set of mainstream media accounts [107], and combinations of both Wikipedia current events and human annotations [127].

In contrast to existing test collections, our evaluation is not based on soliciting relevance judgements, but rather on the output of an active newsroom, offering a potentially more relevant source of ground truth events.

In [142] a review of Interactive Information Retrieval publications from 1967–2006 suggests that system-centred performance measures often do not accurately reflect the user's experience. Breaking news event detection is fundamentally an interactive retrieval problem, but lacking a large group of journalists providing feedback in an interactive setting, our non-interactive test setting aims to partially overcome these drawbacks by relying on real newsroom output, essentially, pre-recorded interactions on Twitter.

The authors in [143] review ad-hoc retrieval results, with an emphasis on statistical significance in effectiveness improvements. [144] diagnose the strengths and weaknesses of retrieval functions. The proposed diagnostic evaluation methodology is used to make improvements to retrieval functions, highlighting the importance and influence of parameter settings.

The Multiple Comparisons Problem inherent in simultaneous testing of many hypotheses using a fixed set of data is explored in [133] and [145]. [146] study variance in system performance using General Linear Mixed Models, and a similar approach is used to isolate query/topic effects in [140]. In our work, we adopt a similar linear model-based approach, which allows us to quantify the effects of altering both the detection procedure and also the evaluation parameters.

Other Sources of Event Data for Evaluation

Wikipedia Current Events Portal (CEP) [147] curated by Wikipedia editors offers a useful source of ground truth for events. One drawback is the focus on US-based sources and a mismatch between Twitter-specific language, and newswire. The events extracted from [147] are aggregated by day, making exact timestamps difficult to extract, but in some cases, these are available from metadata in the cited news article sources, or could be inferred from edit history. ICEWS [148] coded events offer another potential ground truth set, but like CEP are reported on a daily basis. GDELT [149] offers another potential source of ground truth. GDELT attempts to automatically code events with actors, locations, activities and sentiment. An issue with using GDELT as a source of ground truth, is that GDELT events can contain duplicates, occasionally miss events. Version 2 of the corpus provides 15-minute updates, whereas earlier, timestamps for events were reported per day.

Breaking News Events

Breaking news events or *media explosions* as described in [150] and [66] are characterised by a sudden spike in activity generated by multiple reports from social media sources. How offline phenomena link to bursty behaviour online is discussed in [151] and [152]. Combining signals from multiple sources for detecting or tracking the evolution of events proved effective in the past, [58] used signals from Wikipedia page views, together with Twitter to improve FSD. Concurrent Wikipedia edits were used as a signal for breaking news detection in [59].

A number of surveys discuss various other approaches: [153], [132], and [154]. In [132] the survey emphasised different problem definitions and types of event detection in detail. Techniques for detecting and tracking mass emergency situations can be found in [155]. A common theme across all surveys are the difficulties associated with evaluating approaches effectively. In [156] techniques were evaluated on their runtime performance, memory usage as well as similarity to news sources. One of the evaluation parameters we examine includes altering thresholds for this similarity.

Other End-to-End Systems

We consider end-to-end systems as those which are purpose-built for breaking news event detection on the Twitter platform. A number of 3rd-party commercial services also fall under this category.

EDCoW [62] uses document frequency–inverse document frequency signals constructed for each word in the vocabulary. Using discrete wavelet analysis and clustering of signals, significant events are detected with a threshold parameter. Mention-anomaly-based Event Detection (MABED) [63] relies on signals extracted from the frequency of interactions (mentions) between users on Twitter. WikiLiveMon [59] combines signals from both Twitter and edits to Wikipedia pages.

Changes in the platform over time and the treatment of Twitter-specific features makes a direct comparison of these systems difficult, as newer features available to one system may not be available to another. For example: since the introduction of v1.1 of the Twitter API, access to tweet replies is now limited, but new features like quoted tweets and Twitter "moments" have been introduced. Even minor differences in experimental setup can alter results dramatically [157, 158]. There is little reuse of collections, and some systems only focus on particular types of news events (*e.g.* sports, natural disasters).

Two notable commercial services that provide breaking news alerts are *Banjo* and *Dataminr*. While these companies do not publish evaluation results or details of their approaches, since they are popular in large newsrooms, their advertised features offer a useful insight into what some journalists consider useful. *Banjo*[20] works by aggregating geo-tagged posts from many platforms (Twitter included), segmenting this stream into "squares"[21] covering a small area, and monitors these for anomalous activity, issuing alerts with some human intervention. *Dataminr*[22] performs clustering, novelty detection, and measuring content density, and can also issue alerts for a specific geographic region or topic. These, and other 3rd-party Twitter tools are reviewed in [159].

Related Tasks

These tasks are not part of the *detection* component, but are often reliant of good detection performance, and sometimes share annotated news corpora, where the same set of articles can be used for different tasks.

Temporal summarization (TS): The primary concern here is updating time-critical information, given a known event of interest. An event in TS is a "high impact" event, such as a protest, accident or natural disaster. Each event has a topic description, a query, and a fixed duration. In contrast, we do not attempt to issue updates to already detected events, focusing on the initial "alert".

[20] `http://ban.jo`

[21] `http://www.inc.com/magazine/201504/will-bourne/banjo-the-gods-eye-view.html`

[22] `https://www.dataminr.com/technology/`

Event Extraction: The task of *breaking news event detection* differs from the problem of *event extraction* from news sources. The broader field of Information Extraction features news corpora prominently, but is out of scope for this work. A detailed overview of *event extraction* systems is available in [160]. While event extraction and information extraction in general deal with specific information [161], breaking news often deals with unknown or unspecified information. Once a breaking news event occurs however, information extraction becomes important for tracking and updating a story, given that a journalist specifies their information needs.

Political Event Extraction: A more specialised application of event extraction from news is the task of generating political event data, identifying "who-did-what-to-whom" [162]. Representative systems include ICEWS [148], GDELT [149], and PETRARCH [163]. The key distinguishing feature between these kinds of coded events and breaking news events is that the former have known actor codes and defined activities, whereas the latter may lack this type of information. For example, reports of hearing an "explosion" on Twitter may only provide a rough estimate of a location, with unknown consequences, involving unknown actors, and unknown causes—but should be considered a "breaking news event" as it is of interest to journalists.

Predicting or Forecasting Future Events: Relates to information extraction, in that the event is usually planned in advance, as described in [164]. The focus is on classifying events by type and extracting temporal expressions from tweets, producing an event type, date, named entity, event phrase. Examples include scheduled product launch announcements, live concerts, and sports events [165].

3.2.8 Conclusion

With Twitter establishing itself as a critical tool in online journalism for both news gathering and dissemination, systems that generate automated or semi-automated alerts from large data streams have considerable potential in allowing journalists to discover breaking news stories in a timely manner. While a range of systems have recently been proposed in the literature, these systems are often complex and difficult to evaluate in a transparent manner.

In this work, we have proposed and documented a detailed experimental process which robustly measures the effectiveness of the individual components of breaking news detection techniques, using a new ground truth corpus based on the online activity of journalists. As opposed to making judgements based on a fixed set of evaluation parameters, measuring the individual effects of different system design choices can serve to illustrate the key differences between systems more clearly.

The parameter choices that showed the most impact and produced the most detections in our experiments are specific to our input data and choice of evaluation. These should not be directly applied in a different setting, however, the process of decomposing a system into component parts, and performing a posthoc test to measure effects can be applied in a variety of evaluation settings. Decomposing systems into component parts still requires significant engineering effort when applying our approach to other forms of streaming news data.

While the test collection of tweets is more generally applicable in other Twitter-based evaluation settings, it may not be appropriate to use for evaluating data from other platforms or blogs, unless events are stripped of Twitter-specific content. The Reportedly dataset contains a wide variety of different types of events - both rapidly developing and slowly evolving. Slowly developing stories present a challenge to anomaly-based approaches, and other methods may be more appropriate.

Our evaluations demonstrate that, while a range of different detection techniques can often be reasonably effective, practitioners must pay particular attention to both the source of the input data and the impact of parameter selection decisions. In relation to the latter, systems with more advanced components will introduce even more parameter settings that will alter performance in potentially unpredictable ways. The selection of parameters should also be taken into account in the context of the requirements of a journalist or newsroom. For instance, how quickly do alerts need to be delivered relative to the start of an event (*i.e.* the time window)? How many alerts can a journalist realistically process within a given time period (*i.e.* the alert budget)? In our experiments, longer detection windows spanning 30–60 minutes improve system performance, but may not be very useful in practice. A high budget may increase recall, but how tolerant will a journalist be to false alarms? Validating alerts by matching ground truth text with a threshold is straightforward to test, but how useful and interpretable are such bag-of-words alerts to journalists? These remain open questions which need to be addressed in collaboration with domain experts.

Given the same input sources, most time series anomaly detection techniques are capable of detecting spikes in activity with similar accuracy. With optimal parameter settings, we observe that even simple approaches such as counting the occurrence of "newsworthy" words and phrases over time can produce strong baselines for breaking news event detection. Certain parameter settings, however, when not updated over time, can lead to a significant deterioration in system performance due to the inherent concept drift present in news content. This requirement for regular tuning has clear implications for the practical application of many popular detection techniques which have been proposed in the literature.

Rather than solely focusing on anomaly detection techniques themselves, we recommend that more effort should be devoted towards the selection of relevant ground truth sources, and ways of approximating human perception in performance with domain-appropriate evaluation metrics.

We observe that there are no shortcuts in developing complex decision support systems for event detection, and evaluating them will always present significant challenges. There is a significant amount of work required to bring user involvement into evaluations, but this will invariably lead to better systems in the long term.

CHAPTER

FOUR

TRACKING EVENTS OVER TIME

In the previous chapter, the focus was on detecting when a breaking news event occurs, issuing alerts to journalists and applying suitable content filtering to select relevant headlines. This chapter tackles the problem of tracking developing stories over time, both in an online setting and retrospectively.

Section 4.1, *Real-Time Event Monitoring with Trident*, describes implementation challenges with building and deploying scalable, fault-tolerant implementations of event monitoring systems. An online clustering approach aggregates event updates using a query tweet that changes over time as the story develops. Two clustering algorithms suitable for clustering streams of data were implemented on the *Trident* framework. A sequential leader clustering algorithm, and a variant—moving leader clustering algorithm were implemented, clustering tweets in a scalable, online approach.

Breaking news tracking often calls for online learning and incremental training, as documents (especially tweets) arrive one by one in a streaming setting. However, this is not always the best way to approach the problem. Treating one document at a time as they arrive can become computationally intensive as Tweet activity is bursty in nature. Sliding window approaches or mini-batches, as used in the *Trident* framework, are adequate in addressing this problem.

In Section 4.2, *Adaptive Representations for Tracking Breaking News on Twitter*, the system presents an alternative to query expansion for tracking breaking news. Rather than modifying or expanding the query, the underlying representation of documents and queries is modified, relying on distributional semantic models to represent words and documents. The version included in this chapter is expanded, and differs from the paper presented in NewsKDD workshop [89].

The work in Section 4.3, *Detecting Attention Dominating Moments Across Media Types*, explores the retrospective detection of significant events on three different types of news media: tweets, news articles, and blogs.

Significant events often span across different platforms, and given a measure of text similarity, tracking how similar or diverse discussions are between many users can be used as an indicator of a significant event.

4.1 Real-Time Event Monitoring with Trident

Building a scalable, fault-tolerant stream mining system that deals with realistic data volumes presents unique challenges. Considerable work is being done to make the development of such systems simpler, creating high-level abstractions on top of existing systems. Many of the technical barriers can be eliminated by adopting a state-of-the-art interface, such as the Trident API for Storm. In this section, based on [87] we describe a stream mining tool, based on Trident, for monitoring breaking news events on Twitter, which can be extended quickly and scaled easily.

4.1.1 Introduction

Recently there has been a significant shift online towards the task of content curation for online journalism. Media agencies such as Storyful[1] can now break or cover stories as they evolve by leveraging the content produced on social media platforms such as Twitter. However, given the massive volume of content produced by users of these platforms on a daily basis, the task of extracting content that is relevant to individual real-world news events presents a number of challenges. In particular, mining streams of this scale in real-time requires the adoption of new data processing frameworks and data mining algorithms.

In this work, we present a new system for real-time monitoring streams of tweets to cluster relevant tweets around news events. First, we describe an initial application of this system in the context of breaking news on Twitter, and evaluate the usefulness of gathered tweets by allowing a journalist from Storyful to rate the relevancy of the resulting clusters with respect to six major news events from 2013. Based on this evaluation, we highlight specific issues that make the Twitter event monitoring task particularly difficult. We conclude with suggestions for future work to overcome these issues.

Given a target tweet of interest, our system performs a variant of the *leader clustering algorithm* [166], assigning subsequent relevant tweets to one of the groups selected by a journalist. Extracting relevant content from a stream of tweets can be a time-consuming task, automating this process lets journalists spend more time on verifying content and sources.

[1] `http://www.storyful.com`

4.1.2 System Description

Our proposed stream monitoring system is built upon *Storm* [167], an open source framework for real-time distributed computation and data processing[2]. From an architectural perspective, the topology of a Storm system is formed from directed acyclic graphs containing two fundamental node types: "spouts" and "bolts". Spouts produce tuples of data as their output, while bolts perform operations on tuples they receive as inputs.

Trident is a high-level abstraction framework for computation on Storm. As with the core Storm API, Trident uses spouts as the source of data streams. These streams are processed as batches of tuples partitioned among workers in a cluster. Trident provides means for transforming and persistently storing streams.

The framework handles batching, transactions, acknowledgements, and failures internally. Tuple processing follows an "exactly once" semantic, making it easy to reason about processes, apply functions, filters and create aggregations from streams of tuples.

Developing a fault-tolerant, scalable stream mining system can be time-consuming, but most implementation and deployment challenges can be avoided by using the Trident API and supporting software.

To cluster tweets, we have implemented two variations of a single pass algorithm suitable for streaming data. The first is the standard *sequential leader clustering algorithm* [166], a simple incremental approach that divides a dataset into k non-overlapping groups such that, for each group, there is a "leader" data point and all other data points have similarity $\leq \tau$ to the leader.

The second algorithm is a variation of this approach, which is described in Algorithm 1 and which we refer to as *moving leader clustering*.

The leader is *moving* in the sense that the query point can change as new tweets arrive. The moving leader approach attempts to capture new developments in a news story over time. Both algorithms use cosine similarity between tweets in the context of our proposed system.

In this system, the journalist selects a query of interest. This query is either manually entered or is based on the text of a user-selected tweet. This query becomes the *leader* point. As the stream is processed, if a new tweet is within a specific similarity threshold, it then becomes the new leader, altering the original query. Tweets that are close to this threshold are included in the cluster, but do not change the query text.

[2] `http://storm-project.net`

Data: A stream of tweets; User assigned initial tweet leaders
Result: Sets of tweets attached to leaders
while *A tweet Leader exists* **do**
    Compare each new tweet similarity to each leader;
    if $NewTweetDensity < LeaderTweet$ **then** Discard Tweet;
    if $Sim > NewLeaderThreshold$ **then** Assign tweet as new Leader;
    else if $Sim > InclusionThreshold$ **then** Include tweet in Set;
    else Discard Tweet;
end

Algorithm 1: Moving Leader Clustering algorithm.

Event	*Date*	*First Leader Tweet*
Pope Benedict to resign	Mon Feb 11 10:54:43	VATICAN: Italian news agency ANSA reports Pope Benedict to resign.
North Korean nuclear test	Tue Feb 12 03:22:36	SOUTH KOREA: @W7VOA South Korean gov't official say they are checking to verify whether "North Korea conducted a nuclear test."
Chelyabinsk meteor	Fri Feb 15 05:50:16	RUSSIA: We have cleared video of a blast in the sky (believed to be a meteor shower) over Chelyabinsk
Death of Hugo Chavez	Tue Mar 05 21:58:02	VENEZUELA: via official @PresidencialVan Hugo Chavez died at 4:25p today
Cyprus EU bailout	Sat Mar 16 02:52:02	CYPRUS: Follow for reaction to bailout package agreement by Eurozone Finance Ministers of up to 10 billion Euros
Canada exits UN convention	Wed Mar 27 21:30:32	Canada pulling out of UN convention on desertification

Table 4.1: Queries and start times for each news event used in evaluation.

The intuition behind this is that highly relevant tweets (that are not exact duplicates) carry new and interesting information for journalists, while less similar tweets may also be relevant to show, but should not alter the query. The query tweets used in the evaluation are listed in Table 4.1.

Using only the tweet text of the leader Tweet as opposed to a representative term vector from the entire cluster, ensures that the method is sensitive to changes in the story, even after clustering a significant number of tweets. Given large clusters, the most frequent terms would dominate, which is not useful in this setting.

The system performs a form of query replacement, rather than query expansion.

Tweets have several characteristics that create challenges for traditional text analysis. Messages are very short (impacting term frequency), often contain misspellings or abbreviations, words are sometimes concatenated - especially in hashtags, punctuation is sparse (making tokenization a challenge). Twitter specific terms (@mentions, "RT", "via") and emoticons using Unicode characters can also cause problems with automatic language detection. Prior to clustering, each tweet is stripped of these entities and English stop-words, and a corresponding term frequency vector is created.

Tweets are also assigned a density score, which attempts to quantify the quality of a tweet, and promote longer, more informative tweets. Tweet density is the sum of term frequencies of non-stopwords, divided by the number of stopwords they are apart, squared [168].

$$density(tweet) = \frac{K}{K-1} \sum_{k=1}^{K-1} \frac{\text{tf-idf}\left(w_k\right) + \text{tf-idf}\left(w_{k+1}\right)}{\text{distance}\left(w_k, w_{k+1}\right)^2} \tag{4.1}$$

Here, K is the total number of non-stopwords in the tweet, w_k and w_{k+1} are adjacent words, with $distance(w_k, w_{k+1})$ being the number of stopwords between two words. The idf component is calculated based on the current cluster of tweets, not the entire stream.

Tweets with a density score less than the query tweet are discarded. The density filter effectively removes short tweets that would otherwise be assigned very high similarity scores.

Exact duplicates and retweets of already processed tweets are also filtered. While these retweets can be used as a good signal for story popularity, they do not add any new details to a developing story. Given a retweet, only the first instance of a tweet is included in a cluster, subsequent retweets of a known tweet are filtered out.

4.1.3 Evaluation

The public stream of tweets is generally useful for global trend detection and tracking, but is not so useful for tracking developing news stories, as it generally contains too many irrelevant messages for this task [44]. As a solution, a set of "newsworthy" accounts that have been curated and maintained by Storyful, provides a useful filter.

In our evaluation, we monitor a stream of tweets produced by this set of about 15,000 accounts (20,000 if retweeted users are also counted).

Event	*Tweets Streamed*	*Clustered*	*Leader (Query) Changes*
Pope Benedict to resign	17,260,343	19,036	2,654
North Korean nuclear test	21,436,958	56,845	8,833
Chelyabinsk meteor	21,782,200	66,287	3,506
Death of Hugo Chavez	11,276,989	45,197	3,356
Cyprus EU bailout	6,879,389	13,329	85
Canada exits UN convention	4,942,253	1,832	17

Table 4.2: Data for a set of six news events from February and March 2013.

Filtering of this type vastly reduces the volume of spam tweets, and can help balance underrepresented groups of users. One notable difference between our tweet stream and the public sample stream from Twitter is our stream is almost entirely English tweets, as accounts were selected by English speaking journalists.

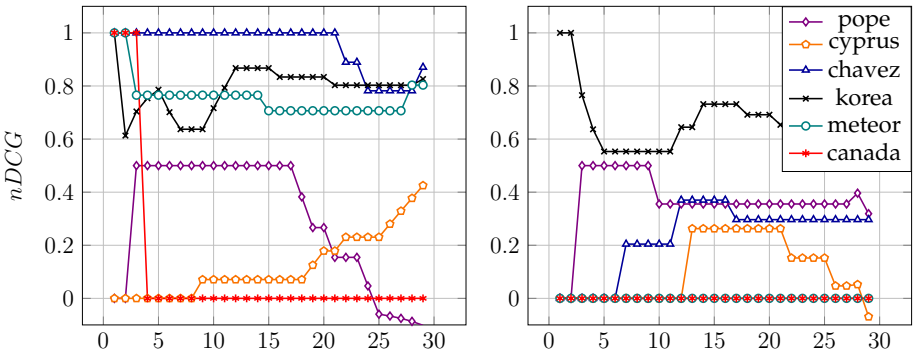
To evaluate the two clustering algorithms described previously, we examined their ability to find future relevant tweets for the six news events listed in Table 4.2, based on the provision of a single query "seed tweet". For every new tweet in the stream, 6 comparisons to the *leader* tweets are performed in parallel. The maximum number of tweet leaders the system can handle is dependent entirely on the hardware the Storm cluster is running on, and it is possible to scale to hundreds of nodes.

For each story, the first 30 tweets posted after the seed tweet and ranked by the system, were presented to a journalist. These tweets would typically represent the first few minutes of a breaking story, when the need for information is greatest. For each event, 24 hours of tweets posted after the query tweet were streamed.

A Storyful journalist manually provided relevance judgements on a scale of 1 for a timely and useful tweet, 0 for a relevant tweet, and -1 for an irrelevant tweet. This scale mirrors the system decisions: an incoming tweet can be timely and useful, and assigned as a new leader tweet (1), assigned to the cluster but have no impact on the query (0), or should not be included in the cluster (-1).

During the evaluation, 30 tweets for each event were presented one-by-one, in chronological order, without additional information (density score, leader status).

Quality of the results is based on the popular normalised discounted cumulative gain measure [169], as this measure reflects information usefulness for a journalist, discounting irrelevant and low ranking results. A summary of the scores achieved by both clustering techniques is shown in Figure 4.1.



(a) Moving Leader (b) Sequential Leader

Figure 4.1: nDCG at k tweets for both clustering methods, for six different news events.

Results

The moving leader approach was more suited to evolving stories, such as the bank bailout in Cyprus. However, constantly changing the leader in the cluster can impact results negatively, especially as users post more personal reactions after unexpected events, such as the resignation of the pope. The "Canada exits UN convention" story was interesting as it received very little attention overall. The scores for this story were low, however - the moving leader variant was able to retrieve the small number of relevant tweets.

The sequential leader was found to be more conservative, and better suited for gathering similar content for stories that do not develop beyond an initial announcement. There is also less chance of the cluster drifting off-topic, due to leader changes. Overall, the sequential leader clustering approach tended to capture many relevant, but near-duplicate tweets, as was the case with the Meteor story - there were no tweets in this cluster that were marked useful by a journalist.

4.1.4 Future Work

The pilot evaluation with a single journalist annotating 6 events revealed a number of problems: this type of manual annotation is too expensive to perform, and not enough judgments can be made in a timely manner without a significant investment of time. Using several annotators could provide a more robust measure of relevancy, however, a much cheaper alternative is to use an automated approach, using existing output from journalists. We explore evaluating system detection output in Section 3.2.

Specifics of tweets mean that term frequency-based similarity measures will often capture redundant information, and fail to explore related topics associated with a query.

While the moving leader approach captured slightly more relevant tweets, performance on different types of events varied.

We plan to investigate alternative ways of expanding tweet clusters around stories, and how topics associated with a news story change over much longer periods of time [170]. Other aspects of the data that are not fully explored involve the social connections between users.

News stories tend to form unique collections of users that actively follow a story [45]. Following other interactions from these users could be a means to expand original queries, or to discover novel insights into an event.

The next key component of our event monitoring system is an approach to summarize news events, based on the selection of a subset of key tweets from a given cluster, which serves to describe the evolution of an event as it unfolded on Twitter. Summarization and timeline generation is explored in more detail in the next section.

4.2 Adaptive Representations for Tracking Breaking News on Twitter

Twitter is often the most up-to-date source for finding and tracking breaking news stories. Therefore, there is considerable interest in developing filters for tweet streams in order to track and summarize stories. This is a non-trivial text analytics task as tweets are short, and standard retrieval methods often fail as stories evolve over time. In this section, based on [89] we examine the effectiveness of adaptive mechanisms for tracking and summarizing breaking news stories. We evaluate the effectiveness of these mechanisms on a number of recent news events for which manually curated timelines are available. Assessments based on ROUGE metrics indicate that adaptive approaches are best suited for tracking evolving stories on Twitter.

4.2.1 Introduction

Manually constructing timelines of events is a time-consuming task that requires considerable human effort. Twitter has been shown to be a reliable platform for breaking news coverage, and is widely used by established newswire services. While it can provide an invaluable source of user-generated content and eyewitness accounts, the terse and unstructured language style of tweets often means that traditional information retrieval approaches have difficulty with this type of content.

Recently, Twitter has introduced the ability to construct *custom timelines*[3] or *collections* from arbitrary tweets. The intended use case for this feature is the ability to curate relevant and noteworthy tweets about an event or topic.

We propose an approach for constructing *custom timelines* incorporating distributional semantic language models (DSMs) trained on tweet text. DSMs create useful representations of terms used in tweets, capturing the syntactic and semantic relationships between words.

We evaluate several retrieval approaches including a neural network language model introduced by Mikolov et al. [35], Random Indexing [171], and a BM25 based method.

Usually, DSMs are trained on large static datasets. In contrast, our approach trains models on relatively smaller sets, updated at frequent intervals. Regularly retraining using recent tweets allows our proposed approach to adapt to temporal drifts in content.

[3] blog.twitter.com/2013/introducing-custom-timelines

This retraining strategy allows us to track a news event as it evolves, since the vocabulary used to describe it will naturally change as it develops over time. Given a seed query (typically a single tweet, or a short phrase, *e.g.* "lax shooting"), our approach can automatically generate chronological timelines of events from a stream of tweets, while continuously learning new representations of relevant words and entities as the story changes. Evaluations performed in relation to a set of real-world news events indicate that adaptive approaches allow us to track events more accurately, when compared to nonadaptive models (models that rely on large, static datasets).

4.2.2 Problem Formulation

Many news outlets have embraced *Live Blogging* and content syndication from social media platforms for breaking news coverage. *Custom timelines*, curated tweet collections on *Storify*[4], and liveblog platforms such as *Scribblelive*[5] are conceptually similar and are popular with many major news outlets.

For the most part, liveblogs and timelines of events are manually constructed by journalists. Rather than automating the construction of timelines entirely, our proposed approach offers editorial support for this task, allowing smaller news teams with limited budgets to use resources more effectively. Our contribution focuses on retrieval and tracking rather than new event detection or verification.

We define a timeline of an event as a timestamped set of tweets relevant to a query, presented in chronological order. The problem of adaptively generating timelines for breaking news events is cast as a topic tracking problem, comprising of two tasks:

Real-time ad-hoc retrieval: For each target query (some keywords of interest), retrieve all relevant tweets from a stream posted after the query. Retrieval should maximize recall for all topics (retrieving as many possibly relevant tweets as available).

Timeline Summarization: Given all retrieved tweets relating to a topic, construct a timeline of an event that includes all detected aspects of a story. Summarization involves removal of redundant or duplicate information while maintaining good coverage.

[4]www.storify.com
[5]www.scribblelive.com

4.2.3 Related Work

The problem of generating news event timelines is related to topic detection and tracking, and multi-document summarization, where probabilistic topic modeling approaches are popular. Our contribution attempts to utilise a state-of-the-art neural network language model (NNLM) and other distributional semantic approaches in order to capitalise on the vast amount of microblog data, where semantic concepts between words and phrases can be captured by learning new representations in an unsupervised manner.

Timeline Generation. An approach by Wang [172] that deals with longer news articles, employed a Time-Dependent Hierarchical Dirichlet Model (HDM) for generating timelines using topics mined from HDM for sentence selection, optimising coverage, relevance, and coherence. Yan et al. [173] proposed a similar approach, framing the problem of timeline generation as an optimisation problem solved with an iterative substitution approach, optimising for diversity as well as coherence, coverage, and relevance. Generating timelines using tweets was explored by Li & Cardie [174]. However, the authors solely focused on generating timelines of events that are of a personal interest. *Sumblr* [175] uses an online tweet stream clustering algorithm, which can produce summaries over arbitrary time durations, by maintaining snapshots of tweet clusters at differing levels of granularity.

Tracking News Stories. To examine the propagation of variations of phrases in news articles, Leskovec et al. [103] developed a framework to identify and adaptively track the evolution of unique phrases using a graph-based approach. In [170], a search and summarization framework was proposed to construct summaries of events of interest. A Decay Topic Model (DTM) that exploits temporal correlations between tweets was used to generate summaries covering different aspects of events. Osborne & Lavrenko [109] showed that incorporating paraphrases can lead to a marked improvement in retrieval accuracy in the task of First Story Detection.

Semantic Representations. There are several popular ways of representing individual words or documents in a semantic space. Most do not address the temporal nature of documents but a notable method that does is described by Jurgens and Stevens [176], adding a temporal dimension to Random Indexing for the purpose of event detection. Our approach focuses on summarization rather than event detection, however, the concept of using word co-occurrence to learn word representations is similar.

4.2.4 Tweet and Timeline Data

The corpus of tweets used in our experiments consists of a stream originating from a set of manually curated "newsworthy" accounts created by journalists[6] as Twitter lists. Such lists are commonly used for monitoring activity and extracting eyewitness accounts around specific news stories or regions.

Our stream collects tweets from a total of 16,971 unique users, segmented into 347 geographical and topical lists. This sample of users offers a reasonable coverage of potentially newsworthy tweets, while reducing the need to filter spam and personal updates from accounts that are not focused on disseminating breaking news events. While these lists of users have natural groupings (by country, or topic), we do not segment the stream or attempt to classify events by type or topic.

Event Data

As ground truth for our experiments, we use a set of publicly available *custom timelines* from Twitter, relevant content from *Scribblelive* liveblogs, and collections of tweets from *Storify*. Multiple reference sources are included when available.

It is not known what kind of approach was used to construct these timelines, but as our stream includes many major news outlets, we expect some overlap with our sources, although other accounts may be missing. Our task involves identifying similar content to event timelines posted during the same time periods.

Since evaluation is based on content, reference sources may contain information not present in our dataset and vice versa. Where there were no quoted tweets in ground truth, the text was extracted, and treated as an update instead. Photo captions and other descriptions were also included in ground truth. Advertisements and other promotional updates were removed.

For initial model selection and tuning, timelines for six events were sourced from Twitter and other liveblog sources:

- "BatKid": Make-A-Wish foundation event.
- "Iran": Follows Iranian Nuclear proliferation talks.
- "LAX": A shooting at LAX.
- "RobFord": Senator Rob Ford Council meeting.
- "Tornado": Reports of multiple tornadoes in US Midwest.
- "Yale": An Alert regarding a possible gunman at Yale University.

[6]Tweet data provided by *Storyful* (www.storyful.com)

These events were chosen to represent an array of different event types and information needs. Timelines range in length and verbosity as well as content type. See Table 4.3 below.

id: Event Name:	Query:	Reference Sources:	Duration: (Hrs:min)	Total Updates	Tweets	Update Freq.
Train 1: BatKid	batkid #sfbatkid	2	5:30	294	123	13.36
Train 2: Iran	iran nuclear deal	3	4:15	197	190	11.59
Train 3: LAX	lax shooting	5	7:15	1186	944	40.90
Train 4: RobFord	rob ford drugs council hearing	4	6:45	1219	904	45.15
Train 5: Tornado	storm tornado alert	5	9:0	2224	1617	61.78
Train 6: Yale	police lockdown #yale	1	7:15	124	124	4.28

Table 4.3: Details of training set events used for parameter fitting. Update Frequency is average number of updates every 15 minutes.

"Batkid" can be characterised as a rapidly developing event, but without contradictory reports. "Yale" is also a rapidly developing event, but one where confirmed facts were slow to emerge. "Lax" is a media-heavy event spanning just over 7 hours while "Tornado" spans 9 hours and is an extremely rapidly developing story, comprised mostly of photos and video of damaged property. "Iran" and "Robford" differ in update frequency but are similar in that related stories are widely discussed before the evaluation period. These training events were used in setting parameter choices such as similarity thresholds and window lengths.

For evaluation, once parameters are fixed, several new events are considered: Table 4.4 gives an overview of the reference sources, durations, content types, and update frequency for each event.

- "MH17": Follows shooting down of Malaysian Air Flight.
- "Metro": Timeline describes a Metronorth train derailment.
- "Westgate": Follows the Westgate Mall Siege.
- "MH370": Details the initial reports of the missing flight.
- "Crimea" follows an eventful day during the annexation of Crimea.
- "Bitcoin": Reporters chase the alleged creator of Bitcoin.
- "Mandela": Reactions to illness & death.
- "P. Walker": Reactions to car accident & death.
- "WHCD": White House Correspondents Dinner.
- "WWDC": Follows the latest product launches from Apple - characterised by a very high number of updates and rapidly changing context.

id: Event Name:	Query:	Reference Sources:	Duration: (Hrs:min)	Total Updates	Tweets	Update Freq.
Test 01: MH17	#mh17 crash	5	7:30	554	487	18.47
Test 02: Metro	#metronorth train derailment	2	10:0	472	469	11.80
Test 03: Westgate	westgate mall shooting	3	18:15	73	62	1.00
Test 04: MH370	mh370 missing	1	7:0	42	7	1.50
Test 05: Crimea	ukraine crimea	1	7:0	34	34	1.21
Test 06: Bitcoin	nakamoto chase	2	4:15	157	149	9.24
Test 07: Mandela	mandela dies	2	4:45	89	51	4.68
Test 08: WHCD	white house correspondence dinner	2	8:0	617	440	19.28
Test 09: P.Walker	paul walker dies	2	5:45	152	106	6.61
Test 10: WWDC	apple #wwdc14	2	3:30	1069	81	76.36

Table 4.4: Details of test set events used for evaluation. Update Frequency is average number of updates every 15 minutes.

The evaluation events were not selected for any particular property or content, but were sourced using the same approach, extracting updates from *custom timelines* from Twitter, *Scribblelive* liveblogs, and tweets from *Storify*.

4.2.5 Sliding Window Timeline Generation

The task of real-time ad-hoc retrieval for constructing timelines is made challenging by the continuously updating collection of documents. Traditional approaches perform poorly lacking global term statistics (IDF counts for example) or become intractable as the collection of documents grows over time. The impact of "cheating" by using future term statistics in a related, but notably different real-time tweet search task is discussed in [177]. The key difference between the TREC real-time tweet search task, and the real-time retrieval task posed here is that the TREC task involves retrieving relevant tweets *before* the query time, whereas for timeline generation, the task is to retrieve tweets posted *after* the query time.

Building Timelines

Timelines are generated using three main components: 1) The curated tweet stream pre-processed and segmented using a sliding window; 2) for each window, an approach is applied to represent the text as a vector; 3) tweets are included or excluded in a timeline based on similarity to the event query.

We compare three adaptive models: BM25 (with updating IDF component), Word2Vec and two Random Indexing approaches (with updating training data), and static variants.

In each case, we initialise the process with a query. For a given event, the tweet stream is then replayed from the event's beginning to end, with the exact dates defined by tweets in the corresponding human-generated timelines. The inclusion of a tweet in the timeline is controlled by a cosine similarity with a fixed similarity threshold (0.75) set using the training events 1–6. Lower similarity thresholds (0.5) can increase recall at the cost of precision, including many marginally relevant tweets, and higher thresholds (0.9) tend to miss new tweets that are not similar enough. The stream is processed using a fixed length sliding window (24 hours) updated at regular intervals (15 minutes) in order to accommodate model re-building time.

The fixed length sliding window approach used to build models of tweet representations (TF-IDF in BM25 or DSM in Word2Vec and Random Indexing models) means that new tweets arriving from the stream are analysed with trained models that are at most *refresh rate* (15) minutes old. Parameter settings for the *window length* and refresh rates are set using the training events and discussed in *Parameter Selection* below.

Pre-processing:

A modified stopword list was used to remove Twitter specific terms (*e.g.* "MT", "via"), together with common English stopwords. URLs and media items are removed, but mentions and hashtags are preserved. For distributional semantic models, filtering stopwords entirely had a negative impact on overall accuracy. Alternatively, we filter stopwords by replacing them with a placeholder token, in order to preserve relative word positions. This approach showed an improvement when compared with no stopword removal, or complete removal of stopwords.

For a term to be included in the training set, it must occur at least twice in the set. These words are removed before training a model.

Extracting potential phrases before training the model, as described in [136] did not improve overall accuracy. In this pre-processing step, frequently occurring bigrams are concatenated into single terms, so that phrases like "trade agreement" become a single term when training a model.

While models can be trained on any language effectively, to simplify evaluation only English tweets were considered. Language filtering was performed using Twitter metadata. On average, there are 150k-200k terms in each sliding window. Updating the sliding window every 15 minutes and retraining on tweets posted in the previous 24 hours was found to provide a good balance between adaptivity and quality of resulting representations. All adaptive and non-adaptive approaches share the same pre-processing steps.

Nonadaptive Approaches: The nonadaptive representation models are variants where word vectors or term frequencies are initially trained on a large number of tweets, and no further updates to the model are made as time passes.

Adaptive Approaches: The adaptive versions use a sliding window approach to continuously build new models at a fixed interval. The trade-off between recency and accuracy is controlled by altering two parameters: *window length* (*i.e.* limiting the number of tweets to learn from) and *refresh rate* (*i.e.* controlling how frequently a model is retrained). No updates are made to the seed query, only the representation of the words changes after retraining the model.

Post-processing For all retrieval models, to optimise for diversity and reduce timeline length the same summarization step was applied to remove duplicate and near-duplicate tweets. The SumBasic [178] algorithm was chosen for producing tweet summaries with high recall [179]. The target length for a summary is determined by the average length of the reference summaries for an event.

4.2.6 Creating Text Representations

For both adaptive and non-adaptive variants, we use three approaches to represent documents:

- **TF-IDF Model:** a BM25 Retrieval Model.
- **Skip-Gram Language Model:** a Neural Network Language Model (NNLM)
- **Random Indexing:** a Dimensionality Reduction Model

These are descibed in detail below.

TF-IDF Model

BM25 [180] with microblog specific settings [181] are a family of scoring functions used to rank documents according to relevance to a query. For a query Q comprising of terms $q_1, \ldots, q_n$ the document D is scored with:

$$\text{Score}_{bm25}(D, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{tf(q_i, D) \cdot (k_1 + 1)}{tf(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \tag{4.2}$$

Where $tf(q_i, D)$ is the term frequency of q_i in D, $|D|$ is document length, and $avgdl$ is average document length in the collection.

Parameter choices of k_1 and b are set to $k_1 = 1.2, b = 0.75$. The IDF calculation can also be substituted for alternatives, but most implementations, including *Lucene*, calculate IDF of term t using IDF$(t) = \log \frac{N-n(t)+0.5}{n(t)+0.5}$, N being the total number of documents in the collection and $n(t)$ the number of documents containing t.

The document and term frequency counts are periodically updated as new information becomes available, using the same sliding window approach for generating training data for other models.

Skip-Gram Language Model

Recent work by [136] introduced an efficient way of training a Neural Network Language Model (NNLM) on large volumes of text using stochastic gradient descent. This language model represents words as dense vectors of real values. Unique properties of these representations of words make this approach a good fit for our problem.

The model produces continuous distributed representations of words, in the form of dense, real-valued vectors. These vectors can be efficiently added, subtracted, or compared with a cosine similarity metric.

The high number of duplicate and near-duplicate tweets in the stream benefits training by providing additional training examples. For example: the vector for the term "LAX" is most similar to vectors representing "#LAX", "airport", and "tsa agent" - either syntactically or semantically related terms.

Moreover, retraining the model on new tweets create entirely new representations that reflect the most recent view of the world. In our case, it is beneficial to have representations of terms where "#irantalks" and "nuclear talks" are highly similar at a time when there are many reports of nuclear proliferation agreements with Iran. At a later time, if "#irantalks" is used in the context of some other negotiations in the future, the term will become more closely associated with the new event vocabulary.

The vector representations do not represent any intuitive quantity like word co-occurrence counts or topics. Their magnitude though, is related to word frequency. The vectors can be thought of as representing the distribution of the contexts in which a word appears.

Typically, these models are trained on large, static datasets. In this case, smaller sets are used, with lowered thresholds for rate terms (minimum count of 2), more training epochs and a lower learning rate.

These parameters produced better performance on smaller datasets in this retrieval task, but may not be optimal for other tasks.

Vector size is also a tunable parameter. While larger vector sizes can help build more accurate models in some cases, in our retrieval task, vectors larger than 200 did not show a significant improvement in scores. (See Figure 4.3).

Additive compositionality is another useful property of these vectors. It is possible to combine several words via an element-wise sum of several vectors. There are limits to this, in that summation of multiple words will produce an increasingly noisy result. Combined with standard stopword removal, and URL filtering, and removal of rare terms, each tweet can be reduced to a few representative words. The NNLM vocabulary also treats mentions and hashtags as words, requiring no further processing or query expansion. Combining these words allows us to compare similarities between whole tweets. Another side effect of comparing similarity of terms, hashtags and user account mentions in the same latent space, is that key user accounts get associated with event hashtags and terms, which can be useful for identifying accounts central to a story.

Training Objective: An alternative to the skip-gram model, the continuous bag of words (CBOW), approach was considered. The skip-gram model learns to predict words within a certain range (the context window) before and after a given word. In contrast, CBOW predicts a given word given a range of words before and after. While CBOW can train faster, skip-gram performs better on semantic tasks. Given that our training sets are relatively small, CBOW did not offer any advantage in terms of improving training time. Negative sampling from [136] was not used. The context window size was set to 5. During training however, this window size is dynamic. For each word, a context window size is sampled uniformly from $1, \ldots, k$. As tweets are relatively short, larger context sizes did not improve retrieval accuracy.

The computational complexity of the skip-gram model is dependent on the number of training epochs E, total number of words in the training set T, maximum number of nearby words C, dimensionality of vectors D and the vocabulary size V, and is proportional to:

$$O = E \times T \times C \times (D + D \times \log_2(V))$$

The training objective of the skip-gram model, revisited in [35], is to learn word representations that are optimised for predicting nearby words. Formally, given a sequence of words $w_1, w_2, \ldots w_T$ the objective is to maximize the average log probability:

$$\frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

In effect, word context plays an important part in training the model.

Random Indexing

Random Indexing (RI) [182] is based on seminal work on sparse distributed memory [183]. While not as popular as NNLM models, it is very well suited to distributed computation, can be highly efficient and has comparable performance to more advanced techniques such as Latent Semantic Analysis/Indexing (LSA/LSI) [184].

The general approach to creating a word space model involves creating a matrix F where each row F_w represents a word w and each column F_c represents a context c. The context can be another co-occurring word, or a document. F is then either a word-by-word or word-by-document matrix, as in the case of LSA.

These types of word spaces suffer from efficiency and scalability problems. The number of words (vocabulary) and documents can make the matrix extremely large and difficult to use. The matrix F is also extremely sparse.

LSA solves this dimensionality and sparsity problem with Singular Value Decomposition (SVD) though this creates other problems, as the SVD operation still requires the full co-occurrence matrix and is difficult to update with new information.

The Random Indexing approach to this problem is to use a random projection of the full co-occurrence matrix in a much lower dimensional space. Random indexing can then be thought of as a dimensionality reduction technique.

The standard Random Indexing technique is a two-step process: An *index vector* is created for each document (or word). This index vector is still high-dimensional, though in the region of several thousand elements, which is still much lower than an entry in a full co-occurrence matrix.

The vectors are sparse (most entries are 0) and *ternary* where randomly distributed $+1$ and -1 values ensure near-orthogonality. The near-orthogonality property is an important attribute of the index vectors in the word space created by RI [185].

The second step involves an element-wise sum of index vectors for each co-occurrence of a word in the text. Words are then represented as d-dimensional vectors consisting of the sum of the contexts in which a word is found. Simple co-occurrence only considers immediate surrounding words, though in practice a context window is extended to include several words.

The accumulation stage results in a d-dimensional space $F_{w\times d}$ which is an approximation of the full $F_{w\times c}$ matrix. This insight is based on the Johnson-Lindenstrauss lemma [186] which states that distances between points are approximately preserved when projecting points into a randomly selected subspace of high dimensionality.

The matrix F can then be approximated by projecting (multiplying) it with a random matrix R:

$$F_{w \times d} R_{d \times k} = F'_{w \times k}$$

The Random Indexing approach is incremental, easily applicable to parallel computation, and efficient, involving simple integer operations.

Variants of Random Indexing approaches involve different strategies for adding *index* or *elemental* vectors. Index vectors can also be initialised for terms rather than documents, in both cases, this "training" step produces vectors that encode meaningful relationships between words that do not co-occur. A variant of the standard RI approach is *Reflective Random Indexing* (RRI) [187] where vectors are built in a slightly different way.

Two variants of RI approaches are implemented as alternatives to the NNLM:

Term-term RI (TTRI)

1. Assign index vector for each term.
2. For each term, sum the index vectors for each co-occurring term in a context window.

Term-based Reflective RI (TRRI)

1. Assign index vector for each term.
2. Generate document vectors by summing index vectors of all terms contained in the document.
3. For each term, sum document vectors for each document the term occurs in.

The trained model represents a word space similar to the model created by the skip-gram (word2vec) model. The same additive composition approach is used to create a vector representing a whole tweet, with an element-wise sum of the individual word vectors.

4.2.7 Parameter Selection

Our system has a number of tunable parameters that suit different types of events. When generating timelines of events retrospectively, these parameters can be adapted to improve accuracy. For generating timelines in real-time, parameters are not adapted to individual event types.

For all models, the *seed query* (either manually entered, or derived from a tweet) plays the most significant part. Overall, for the NNLM and RI models, short event specific queries with few terms perform better than longer, expanded queries which benefit term frequency (BM25) model. In our evaluation, the same queries were used while modifying other parameters. Queries were adapted from the first tweet included in an event timeline to simulate a lack of information at the beginning of an event.

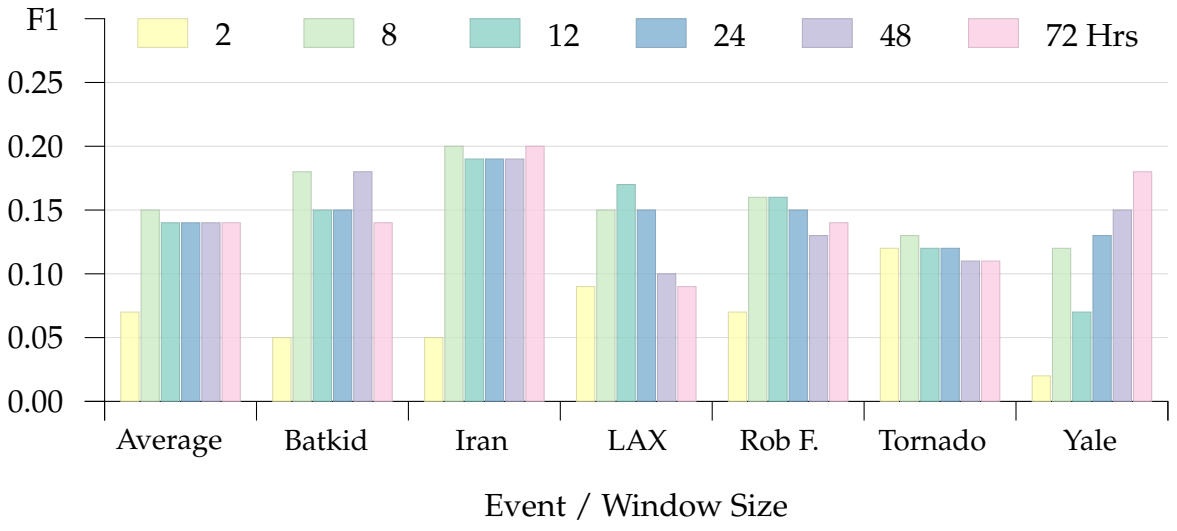


Figure 4.2: F1 scores for Adaptive model accuracy in response to changing window size

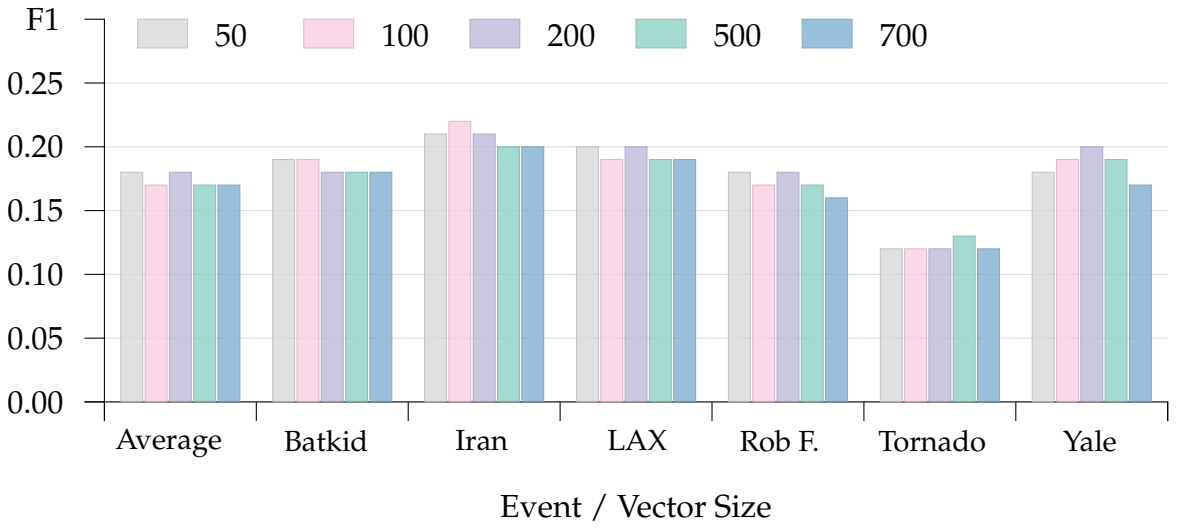


Figure 4.3: F1 scores for Adaptive model accuracy in response to changing vector size

The *refresh rate* parameter controls how old the training set of tweets can be for a given model. In the case of BM25 model, this affects the IDF calculations, and for NNLM and RI models, the window contains the preprocessed text used for training.

As such, when the system is replaying the stream of tweets for a given event, the model used for similarity calculations is *refresh rate* minutes old.

Window length effectively controls how many terms are considered in each model for training or IDF calculations. While simpler to implement, this fixed window approach does not account for the number of tweets in a window, only the time range is considered. The volume of tweets is not constant over time—leading to training sets of varying sizes. However, since the refresh rate is much shorter than the window length, the natural increase and decrease in tweet volume is smoothed out.

On average, there are 150k-200k unique terms in each 24-hour window. Figure 4.2 shows how varying window size can improve or degrade retrieval performance of different events.

Updating the sliding window every 15 minutes and retraining on tweets posted in the previous 24 hours was found to provide a good balance between adaptivity and quality of resulting representations.

Larger window sizes encompassing more tweets were less sensitive to rapidly developing stories, while smaller window sizes produced noisier timelines for most events.

Figures 4.2 and 4.3 are showing the word2vec model performance. Random Indexing approaches showed a similar pattern when changing window size and vector length, though in the random indexing case, the vector size is set to 2500, larger vectors showed no increase in retrieval performance.

4.2.8 Evaluation

In order to evaluate the quality of generated timelines, we use ROUGE [188] and a diversity score to compare human curated timelines and system-generated timelines.

The human-curated timelines are used as ground truth for both the text content of the timeline, update frequency and length of the final timeline. For all retrieval models, to reduce timeline length the same summarization step was applied to remove duplicate and near-duplicate tweets. The SumBasic [178] algorithm was chosen for producing tweet summaries with high recall [179]. The target length for a summary is determined by the average length of the reference summaries for an event. This step ensures that timelines with significantly more tweets do not have an automatic advantage in recall. ROUGE scores are calculated using the entire timeline at the end of the event. As an illustrative example, Table 4.5 shows a 15-minute segment for updates for the *MH17* story before and after this summarization step.

Time:	Before Summarization:	Included in Summary:
15:42	#BREAKING Malaysian Airlines passenger jet reportedly shot down near Russia/Ukraine border. 280 people 15 crew on board.	#BREAKING Malaysian Airlines passenger jet reportedly shot down near Russia/Ukraine border. 280 people 15 crew on board.
15:42	*Report: Malaysian airliner with 295 aboard crashes in Ukraine*	-
16:01	Terrorists were probably using russian -300/Buk systems. #MH17 flight altitude was 10 000 m. #malaysia #ukraine	Terrorists were probably using russian -300/Buk systems. #MH17 flight altitude was 10 000 m. #malaysia #ukraine
16:01	*Ukrainian official: Malaysian passenger plane carrying 295 people shot down over Ukrainian airspace*	-

Table 4.5: A short section of a longer timeline describing the MH17 event, showing the timeline content for the same time window before and after applying summarizarion. The tweets excluded from the summary contained information that was already present in the timeline. While this reduces duplicate information, some important aspects may be missed, *e.g.* it may be relevant to see that a *Ukrainian official* made an official statement, this tweet was excluded from the summary.

The popular ROUGE set of metrics, which measure the overlap of n-grams, word pairs and sequences between the ground truth timelines, and the automatically generated timelines. ROUGE parameters are selected based on [189]. ROUGE-1 and ROUGE-2 are widely reported and were found to have good agreement with manual evaluations in related summarization tasks. In all settings, stemming is performed, and no stopwords are removed. The text is not pre-processed to remove tweet entities such as hashtags or mentions but URLs, photos and other media items are removed. Several ROUGE variants for automatic evaluation are considered, as there is currently no manual evaluation of the generated summaries.

Only *MH370* and *Crimea* events have one reference timeline to compare against. The other events have a minimum of two, sourced from different news outlets. Details about each event are listed in Table 4.4.

ROUGE-N is defined as the n-gram recall between a ground truth (reference) and system-generated summary:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{\text{Reference Summaries}\}} \sum_{gram_n \in S} Count(gram_n)} \tag{4.3}$$

where N is the n-gram value, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in the reference summaries and system-generated summary.

ROUGE-L variant is based on the longest common subsequence (LCS) match between the reference and generated summaries. ROUGE-1 and ROUGE-L scores were highly correlated (Pearson's r > 0.99), producing the same ranking of system performance.

ROUGE-SU or Skip-bigram variant, measures the overlap of skip-bigrams between summaries. In contrast to LCS, this variant counts all matching word pairs. In the sentence *"Satoshi got free sushi"* has 6 skip-bigrams: [*"Satoshi got"*, *"Satoshi free"*, *"Satoshi sushi"*, *"got free"*, *"got sushi"*, *"free sushi"*]. Typical settings for the maximum skip distance between two words is set to 4.

A more robust variant of ROUGE, BEwT-E: Basic Elements with Transformations [84] is also reported. Basic Elements are variable sized, syntactically coherent units extracted from text. Transformations are applied to the generated and reference summaries with named entity recognition, abbreviation expansion and others. While this evaluation approach more closely correlates with human judgements in other tasks, the lack of Twitter-specific transformations could negatively impact performance - mapping @barackobama to "Barack Obama" for example. All default BEwT-E settings, part of speech models and named entity recognition models are used.

As an illustrative example, Table 4.6 shows a system-generated timeline and a reference timeline curated by a journalist. In some cases the same tweets present in a human-generated timeline appeared in our automatically generated timelines (time period 17:34–18:15), indicating that our data source provides good coverage of newsworthy sources for a variety of events.

An example where neither approach seemed to capture relevant tweets was the "Tornado" event. In this case, the human-generated timeline was comprised mostly of photos and videos from the affected areas. In contrast, the text content in many tweets for this timeline was not event-specific. Terms in content such as "house in the middle of road" or "just shared a photo" can occur in many contexts on Twitter, so the vectors associated with those terms typically yield noisier results.

Performance on Unseen Events

In most cases, shown in Figure 4.4, adaptive approaches perform well on a variety of events, capturing relevant tweets as the event context changes. This is most notable in the "WWDC14" story (Event 10 in Table 4.8), where there were several significant changes in the timeline as new products were announced for the first time.

Event Period	Ground Truth	Retrieved Tweets (NNLM)
15:30–16:11	Confirmed report of a person w/ gun on/near Old Campus. SHELTER IN PLACE.	NOW: Police responding to reports of a person with a gun at Yale University. Shelter in Place issued on Central Campus (via @Yale)
17:34–18:15	New Haven police spokesman says there is no description of a suspect @Yale and "This investigation is in its infancy" #NHV #Yale	New Haven police spokesman says there is no description of a suspect @Yale and "This investigation is in its infancy" #NHV #Yale
18:57–19:38	hartman: possibility that witnesses of long guns saw instead law enforcement officers responding to the scene #Yale	RT @NBCConnecticut: Police say witnesses who saw person with long gun at @Yale could have seen law enforcement personnel. #Yalelockdown

Table 4.6: A manual selection of retrieved tweets for "Yale" event highlighting key developments, and how the adaptive NNLM model can handle concept drift with high recall.

id	ROUGE-BEwT Scores									
	Recall					Precision				
	BM 25	w2v dyn.	RI dyn.	w2v stat.	RI stat.	BM 25	w2v dyn.	RI dyn.	w2v stat.	RI stat.
1	0.32	**0.33**	0.33	0.18	0.25	0.32	**0.33**	0.33	0.18	0.25
2	**0.19**	0.19	0.19	0.15	0.14	**0.19**	0.19	0.19	0.15	0.14
3	0.17	0.18	0.19	0.19	**0.20**	0.17	0.18	0.19	0.19	**0.20**
4	0.20	0.23	0.21	0.21	**0.24**	0.20	0.23	0.21	0.21	**0.24**
5	0.14	0.17	0.16	**0.18**	0.16	0.14	0.17	0.16	**0.18**	0.16
6	**0.19**	0.17	0.16	0.15	0.14	**0.19**	0.17	0.16	0.15	0.14
7	0.22	0.18	0.23	0.18	0.13	0.22	0.18	0.23	0.18	0.13
8	**0.08**	0.07	0.07	0.05	0.07	**0.08**	0.07	0.07	0.05	0.07
9	0.20	0.19	0.21	0.14	0.16	0.20	0.19	0.21	0.14	0.16
10	**0.16**	0.15	0.14	0.13	0.09	**0.16**	0.15	0.14	0.13	0.09

Table 4.7: Detailed Precision & Recall scores for ROUGE-BEwT for unseen events. Best score in bold.

id	ROUGE-SU4 Scores									
	Recall					Precision				
	BM 25	w2v dyn.	RI dyn.	w2v stat.	RI stat.	BM 25	w2v dyn.	RI dyn.	w2v stat.	RI stat.
1	0.15	0.14	**0.16**	0.10	0.13	0.23	**0.25**	0.22	0.12	0.18
2	0.17	0.17	**0.19**	0.17	0.16	0.40	**0.41**	0.35	0.25	0.33
3	0.12	**0.12**	0.11	0.11	0.10	0.09	**0.10**	0.09	0.09	0.09
4	0.14	0.17	0.15	0.16	0.17	0.20	0.25	0.23	0.25	0.27
5	0.12	**0.15**	0.13	0.15	0.13	0.12	**0.15**	0.13	0.15	0.13
6	**0.22**	0.21	0.19	0.20	0.16	0.19	0.20	0.22	0.20	0.23
7	**0.17**	0.15	0.16	0.15	0.13	0.20	0.19	0.19	**0.21**	0.19
8	**0.08**	0.07	0.08	0.05	0.07	0.20	0.31	0.22	**0.32**	0.21
9	**0.21**	0.20	0.21	0.14	0.19	0.16	0.15	0.15	0.14	0.17
10	**0.12**	0.10	0.11	0.10	0.07	0.30	**0.35**	0.30	0.33	0.30

Table 4.8: Detailed Precision & Recall scores for ROUGE-SU4 for unseen events. Best score in bold.

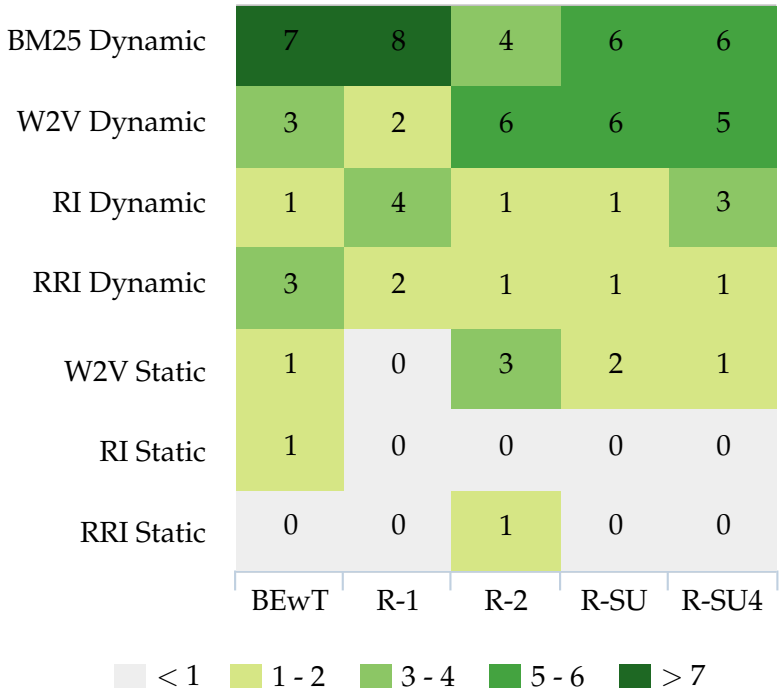


Figure 4.4: Methods Ranked 1st in F_1 Score, under various ROUGE settings, in all 16 events. There is variation between rankings of approaches across different ROUGE settings, however, dynamic approaches are always ranked higher than static.

While adaptive approaches can follow concept drift in a news story, a notable drawback of DSMs was the lack of disambiguation between multiple meanings of some terms. Even though relevant tweets are retrieved as the news story evolves, irrelevant but semantically related tweets were also present in some timelines—mentions of other car accidents from earlier in the case of the "Paul Walker" event for example.

Overall the adaptive NNLM approach performs much more effectively in terms of recall rather than precision. A more effective summarization step could potentially improve accuracy further. This property makes this model suitable for use as a supporting tool in helping journalists find the most relevant tweets for a timeline or liveblog, as the tweets retrieved tend to be much more diverse than those retrieved by the BM25 approach, which favours longer tweets with more repetitive use of terms.

Diversity of generated timelines: The average pairwise cosine similarity of tweets in the timelines was used as a measure of diversity. Using the diversity score of the reference timelines, diversity and redundancy can be compared relative to the available references. Scores above 1.0 indicate that timelines have less repetition and redundant information than human-generated timelines. Scores below 1.0 indicate that tweets in the timeline are very similar and repetitive. Figure 4.5 shows events where at least 1 method produces a more diverse timeline than the reference.

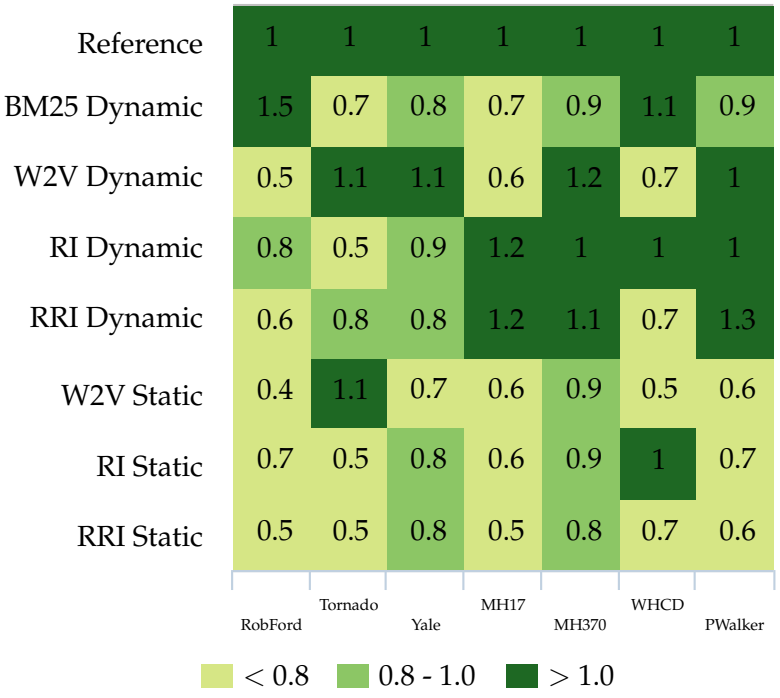


Figure 4.5: Diversity relative to reference timelines for events and approaches. Scores are normalised relative to human curated timelines, that have a diversity of 1. Scores above 1 indicate that timelines had more novelty and diverse tweets, while lower scores have more repetitive content.

The Nonadaptive approach performs well in cases where the story context does not change much, tracking reactions of celebrity deaths for example. Timelines generated with this variant tend to be more general.

An example where neither approach seemed to capture relevant tweets was the "Tornado" event. In this case, the human-generated timeline was comprised mostly of photos and videos from the affected areas. In contrast, the text content in many tweets for this timeline was not event-specific. Terms in content such as "house in the middle of road" or "just shared a photo" can occur in many contexts on Twitter, so the vectors associated with those terms typically yield noisier results.

While the additive compositionality of learnt word representations works well in most cases, there are limits to this usefulness. Short, focused seed queries tend to yield better results. Longer queries benefit baseline term frequency models but hurt performance of the NNLM approach.

4.2.9 Future and Ongoing Work

Currently, there is a lack of high quality annotated Twitter timelines available for newsworthy events, as current methods provided by Twitter for creating custom timelines are limited to either manual construction, or accessed through a private API. Other forms of liveblogs and curated collections of tweets are more readily available, but vary in quality.

As new timelines are curated, we expect that the available set of events to evaluate will grow. We make our dataset of our reference timelines and generated timelines available[7].

We adopted an automatic evaluation method for assessing timeline quality. A more qualitative evaluation involving potential users of this set of tools is currently in progress. The *Pyramid Method* [190] could be an appropriate user-based evaluation of our system-generated timelines. Reference summaries can still be sourced from liveblogs from newsrooms, but annotators are required for identifying and matching content. The approach requires an assessor to extract Summary Content Units (SCUs) from reference timelines and assign these SCUs weights. An example SCU is shown in Figure 4.6 below.

Given SCUs, weights, reference and system-generated summaries, the SCUs can then be arranged by weight, and partitioned in a *pyramid* where each tier will have SCUs of the same weight.

[7] `http://mlg.ucd.ie/timelines`

SCU Label: Batkid saves Lou Seal, and captures the Penguin.

Weight: 2

Reference 1: #SFBatkid has apprehended @PenguinSF and saved Lou Seal.

Reference 2: BatKid saves Giants mascot @LouSeal01, foils Penguin's evil plan.

Figure 4.6: An example Summary Content Unit (SCU). An annotator is tasked with manually highlighting, labeling and matching up SCUs from reference summaries. An SCU is a continuous or discontinuous sequence of words that expresses the same information.

Given a pyramid of tier n, an optimal summary should, therefore, contain all SCUs from the top tier, all SCUs from the next tier, and so on to the n-th tier. Further details of the Pyramid method are described in [190].

There is also room for improving the model retraining approach. Rather than updating the model training data with a fixed length moving window over a tweet stream, the model could be retrained in response to tweet volume or another indicator, such as the number of "out of bag" words, *i.e.* words for which the model does not have vector representations for.

Retrieval accuracy is also bound by the quality of our curated tweet stream, expanding this dataset would also improve results. Many news-related tweets also contain useful links to longer articles, images and video. Currently, the model is not capable of retrieving this kind of content.

The quality of the word vectors depends mostly on the size of the training set, and our approach requires a relatively fast training time in order to generate a timely model. There is a trade-off between quality and ability to adapt, that is dependent on the number of unique words, so a further improvement might be gained by maintaining a foreground model that's updated frequently, as well as a background model built on a much larger dataset.

The SumBasic summarization step does not make use of any information from the retrieval models, and a better summarization approach that explicitly accounts for diversity and novelty could take better advantage of the DSM approaches.

4.2.10 Conclusion

Distributional semantic models trained on Twitter data have the ability to capture both the semantic and syntactic similarities in tweet text. Creating vector representations of all terms used in tweets enables us to effectively compare words with account mentions and hashtags, reducing the need to pre-process entities and perform query expansion to maintain high recall. The compositionality of learnt vectors lets us combine terms to arrive at a similarity measure between individual tweets.

Retraining the model using fresh data in a sliding window approach allows us to create an adaptive way of measuring tweet similarity, by generating new representations of terms in tweets and queries at each time window.

Experiments on real-world events suggest that this approach is effective at filtering relevant tweets for many types of rapidly evolving breaking news stories, offering a useful supporting tool for journalists curating liveblogs and constructing timelines of events.

4.3 Detecting Attention Dominating Moments Across Media Types

In this chapter, based on work in [91] we address the problem of identifying attention dominating moments in online media. We are interested in discovering moments when everyone seems to be talking about the same thing.

We investigate one particular aspect of breaking news: the tendency of multiple sources to concentrate attention on a single topic, leading to a collapse in the diversity of content for a period of time. In this work, we show that diversity at a topic level is effective for capturing this effect in blogs, in news articles, and on Twitter. The phenomenon is present in three distinctly different media types, each with their own unique features. We describe the phenomenon using case studies relating to major news stories from September 2015.

4.3.1 Introduction

The problem of detecting breaking news events has inspired a host of approaches, extracting useful signals from activity on social networks, newswire, and other types of media. The online communication platforms that have been adopted allow these events to persist in some form. These *digital traces* can never fully capture the original experience, but offer us an opportunity to revisit significant phenomena with different points of view, or help us to characterise and learn something about the processes involved.

Many different forms of news media attempt to record and disseminate information deemed important enough to communicate, and as the barriers to broadcasting and sharing information are removed, attention becomes a scarce commodity.

We define the problem of detecting *attention dominating moments* across different media types, as a collapse in diversity in the content generated by a set of online sources in a topic during a given time period. *Media types* here include mainstream news articles, blog posts, and tweets. These media types differ in both the category of topics covered [191], and their use of language [192].

In the context of Twitter, we define *sources* as unique user accounts. For mainstream news and blogs, sources refer to individual publications or outlets. Publications may have different numbers of authors, but as unique author information is not available, we treat each unique blog or news outlet as a single source.

In Section 4.3.3, we describe the two stages of our proposed event detection procedure. In the first stage, content generated by the news, blog and tweet sources is grouped into broad topical categories, through the application of matrix factorization to the content generated by these sources. In the second stage, we examine the variation in the similarity between content generated by sources within a given topic during a given time period, in order to identify a collapse in diversity which corresponds to an attention dominating moment. In Section 4.3.5, we evaluate this procedure on a collection of one million news articles and blog posts from September 2015, along with a parallel corpus of tweets collected during the same time period.

Rather than formulating the problem as tracking the evolution of topics themselves, we consider the diversity of content within a specific topic over time. The motivation is that, for instance, a collapse in diversity around a major sporting event will be strongly evident in certain news sources, but not evident in others. The distinction is important, as this approach is more suited to retrospective analysis, when the entire collection of documents of interest is available. The topics do not change over time, as opposed to a real-time setting where topics must be updated as new documents arrive [193]. The information need is guided by two major questions. Firstly, when have significant collapses in diversity occurred in a topic of interest? Secondly, are there differences between media types when these events occur?

Our main contributions here are: 1) a diversity-based approach to detecting attention dominating news events; 2) a comparison between traditional news sources, blogs, and Twitter during these events. 3) a parallel corpus of newsworthy tweets for the NewsIR dataset.

4.3.2 Related Work

In previous work, attention dominating news stories have been described as *media explosions* [150] or *firestorms* [66]. The idea of combining signals from multiple sources for detecting or tracking the evolution of events proved effective in the past. Osborne et al. [58] used signals from Wikipedia page views, together with Twitter to improve "first story detection". Concurrent Wikipedia edits were used as a signal for breaking news detection in [59].

Topic modeling applied to parallel corpora of news and tweets has been previously explored by a number of researchers [194, 195, 196]. Extensions to LDA to account for tweet specific features have been proposed [191]. A comparison between Twitter and content from newswires was explored in [46]. A Non-negative Matrix Factorization (NMF) approach is used for topic detection in [197].

How offline phenomena link to bursty behaviour online is discussed in [151] and [152]. In [152] Shannon's Diversity Index was used to detect a "contraction of attention" in a tweet stream by measuring the diversity of hashtags. In contrast, we employ a different measure of diversity based on document similarity, applying it to streams from different media types segmented by topic. Methods for automatically detecting anomalies or significant changes in a time series are discussed in [198]. In [199] a change-point detection approach is applied to time series constructed from Tweet keyword frequencies.

As a broad overview, the common components involved in detecting high-impact, attention dominating news stories include: selecting relevant subsets of documents; representation and feature extraction; constructing time series from features; event detection and analysis. In this work, we concentrate on a single key feature of breaking news: a collapse in content diversity within a fixed time window.

4.3.3 Document Diversity Across Media Types

Our objective is to detect when multiple articles in a topical stream become less diverse, signalling the emergence of an attention dominating news story. We consider attention to a phenomenon as the main driving force behind the decision to produce or broadcast a message. Using the diversity of content within a time window, we attempt to characterise instances where a particular piece of information becomes dominant. Concretely, for each type of media, NMF is used to assign topics to documents; for documents in a topic, we calculate diversity between documents in a time window. This type of analysis allows us to examine the extent to which the onset of an important breaking news event is accompanied by a collapse in textual content diversity, both within a group of news sources and across different media types.

Preprocessing

For each *media source*, we build a *tf-idf* weighted term-document matrix and use this as input to NMF and for calculating diversity. Stopwords and words occurring in fewer than 10 documents are removed, as well as words occurring in more than 95% of documents. Stemming was not applied. While stemming can reduce the size of the vocabulary it does not help topic coherence and can produce topic terms that are difficult to interpret [200]. Different preprocessing steps may be more appropriate for non-English text.

Finding Topics

We apply a Non-negative Matrix Factorization (NMF) topic modeling approach to extract potentially interesting topics from a stream of tweets or a set of articles. We also considered LDA to infer topics in these datasets. The choice of NMF over LDA was primarily due to computation time. LDA was significantly more computationally expensive than NMF with NNDSVD [201] initialisation. NMF also tends to produce more coherent topics [200].

Different topic models and parameters can be applied to each individual *media type*, but results in Section 4.3.5 kept the same initialisation and 30 topics per media type.

Measuring Diversity

The same *tf-idf* representation used for topic modeling is used in diversity calculations. Each article, blog post or tweet is a *tf-idf* vector. A separate document-term matrix is built for each *media type*.

To measure diversity, we calculate the mean cosine similarity between all unique pairs of articles within a topic for a fixed time window. Given a set of documents D in a time window, the diversity is:

$$diversity(D) = -\frac{\sum_{i,j \in D, i \neq j} cosSim(D_i, D_j)}{\sum_{i=1}^{|D|-1} i}$$

Where $cosSim(D_i, D_j)$ is the cosine similarity of *tf-idf* vectors of documents i and j in a time window. In practice, calculating similarities between all pairs of documents can be efficiently performed in parallel, and can be calculated in a matter of seconds.

Longer time windows consider more document pairs, which naturally result in smoother trends. In contrast, shorter time windows are more sensitive to brief attention dominating events, but also false positive spikes—where a small number of articles happen to be similar in content, but do not constitute an attention dominating story.

An alternative to content diversity is also considered. Ignoring document content, and just considering the sources of articles, diversity is calculated with Shannon's Diversity Index:

$$H' = -\sum_{i=1}^{R} p_i \ln p_i$$

Where p_i is the proportion of documents produced by the ith source in a time window of interest, R is the total number of sources in a given media type.

Both diversity measures produce a single diversity value per time window, generating a univariate time series. Changes in diversity that are 2 standard deviations away from the mean are naively considered to be important enough to warrant attention. Exploring more robust and well-established methods for change point detection such as [199, 198] is left for future work.

For the case studies described in Section 4.3.5, the window length was set to 8 hours. While the fast-paced "24/7 news cycle" is described as a constant flood of information, we find that all three mediums largely follow a more traditional publishing cycle, with prominent spikes in the number of published articles on weekday mornings, and low numbers of articles published outside of normal office hours. A more detailed analysis of publishing times and characteristics will be explored in future work.

4.3.4 Datasets

To explore attention dominating news stories, we apply the method described above to three media sources: mainstream news, blogs, and tweets. For the first two sources, the NewsIR dataset[8] is used. For the final source, we use our own parallel corpus collected from Twitter[9]. In contrast to previous work [194, 196] where tweets are retrieved based on keywords extracted from news articles, the parallel corpus was derived from a large set of newsworthy sources, curated by journalists [89]. Journalists on Twitter curate lists[10] of useful sources by location or general topic of interest—for example, "US Politics" may contain accounts of US politicians and other journalists who tend to cover US politics related stories.

Gathering all members of such lists covering different countries and topics follows the *expert-digest* strategy from [130]. A tweet dataset collected independently of news and blog articles preserves Twitter-specific features and topics. Source and document counts are summarized in Table 4.9.

Of the original 1 million articles provided, 15,878 were filtered as non-English[11] or outside the date range of interest (*i.e.* created between 2015-09-01 and 2015-09-31). Tweet language filtering was performed using meta-data provided in the tweet.

[8] Available from: `http://research.signalmedia.co/newsir16/signal-dataset.html`

[9] Data: `https://dx.doi.org/10.6084/m9.figshare.2074105`

[10] Examples of such lists are available `https://twitter.com/storyful/lists/` and `https://twitter.com/syflmid/lists`

[11] `https://github.com/optimaize/language-detector` was used for language detection. Interestingly, language detection proved effective for filtering "spammy" articles containing obfuscated text, large numbers of URLs, or containing tabular data.

Media Type	*Sources*	*Documents*	*Docs. per 24h*
News	18,948	730,634	8,177
Blogs	73,403	253,488	23,568
Tweets	30,448	3,274,089	125,568

Table 4.9: Summary of overall source and document counts by media type after filtering, and average number of documents in a 24-hour window.

4.3.5 Attention Dominating Events

In order to compare the same topics across different media types, we compare the top 10 terms representing the topics from different models. Specifically, when topics from two different models have overlapping top term lists (using Jaccard similarity > 0.3), this indicates that similar events were discussed in both media types.

Topics in a model that do not have any overlapping terms with topics in other models suggest that content unique to a platform is prominent. For example, the "*live, periscope, follow, stream, updates*" topic in the tweet corpus has no equivalent among the news or blog topics. This reflects the fact that the Periscope app became popular with journalists for broadcasting short live video streams and Twitter is the main platform where these streams are announced. The "*music, album, song, video, band*" topic is prominent in the blogs and Twitter but is not present in news. This may reflect the fact that most Twitter accounts and blogs are far more personal in nature.

An indicative, but not a necessary feature of attention domination news is the presence of a similar topic on multiple platforms. To illustrate the phenomenon of topical diversity collapse, we now describe three case studies.

For each case study, we present the following: Top 10 topic terms for a topic in a media type, and a plot of diversity over time, where:

- Solid lines show the diversity of documents over time.
- Dashed lines show Shannon Diversity of sources.
- Highlighted time periods are when major developments occurred—based on Wikipedia Current Events Portal[12] for September 2015.
- Dot and Triangle markers indicate periods when diversity drops 2 standard deviations below the mean.

[12] `https://en.wikipedia.org/wiki/Portal:Current_events/September_2015`

European Refugee Crisis

The European crisis began in 2015, as increasing numbers of refugees from areas in Syria, Afghanistan, and Western Balkans [202] sought asylum in the EU. Figure 4.7 shows a plot of diversity for the documents assigned to this topic in each 8 hour time window, for the three media types. To help with visualisation, raw diversity values are standardised with z-scores on the y axis, while the x axis grid separates days.

Media	*Top 10 Topic Terms*
Blogs	refugees, syria, syrian, war, president, government, military, europe, russia, iran
News	refugees, migrants, border, hungary, eu, europe, european, refugee, asylum, germany
Tweets	refugees, syrian, hungary, help, migrants, europe, border, germany, austria, asylum

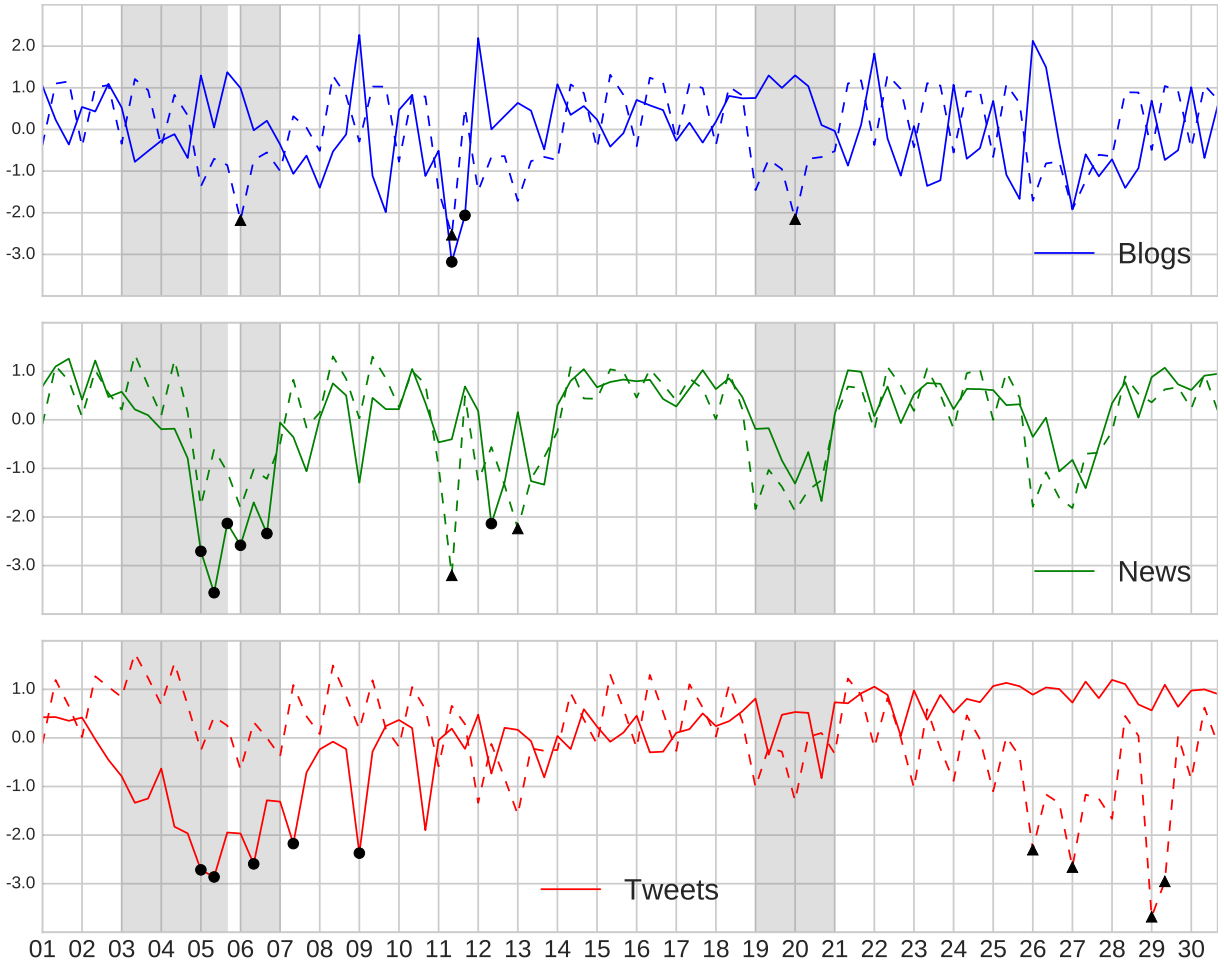


Figure 4.7: Standardised diversity scores for the European refugee crisis topic during September 2015, across three media types.

The downward trend in diversity between September 3rd and 5th in the refugee crisis topic can be explained by the death of Alan Kurdi. News of his drowning quickly spread online and made global headlines.

This was a particularly far-reaching story, dominating news coverage until an announcement on relaxing controls on the Austro-Hungarian border by Chancellors Faymann of Austria and Merkel of Germany. Both Twitter and mainstream news streams experienced a diversity collapse, while Blogs maintained a more diverse set of articles. Between 19th and 21st, smaller drops in diversity coincide with Pope Francis' visit, where the issue of refugees was a prominent topic of discussion.

Donald Trump Presidential Campaign

Donald Trump's presidential campaign has attracted considerable attention across all types of media[13]. Positions on issues of immigration and religion are particularly polarising, frequently causing controversies in mainstream media.

Media	*Top 10 Topic Terms*
Blogs	trump, donald, republican, presidential, debate, gop, president, candidates, candidate, bush
News	trump, republican, presidential, donald, debate, clinton, bush, fiorina, candidates, campaign
Tweets	trump, im, love, donald, going, debate, happy, gop, president, think

Significant events marked around 12th, 17th, 21st in Figure 4.8 relate to: Trump's comments on Senator Rand Paul on Twitter which was discussed on mainstream news around 12th, but not as prominently on blogs. On the 16th-17th coverage of a Republican presidential debate hosted by CNN; and 21st—mainstream news coverage of reactions to events on 17th: during a town hall meeting in Rochester, Donald Trump declined to correct a man who said that President Obama is a Muslim.

The statement prompted a significant drop in the diversity of stories across all platforms. On the 25th, during a speech given to conservative voters in Washington, Trump called fellow Republican presidential candidate Marco Rubio "a clown". Based on the data, it appears that the reaction to the latter on Twitter was not as pronounced as among journalists and bloggers.

Pope Francis visits North America

The visit of Pope Francis spanned 19 to 27 September 2015, where the itinerary included venues in both Cuba and the United States. This event is a good illustrative example as it was widely documented[14], and highlights a case where a collapse in diversity did not occur at the same time on different media platforms.

[13] https://en.wikipedia.org/wiki/Donald_Trump_presidential_campaign,_2016
[14] https://en.wikipedia.org/wiki/Pope_Francis'_2015_visit_to_North_America

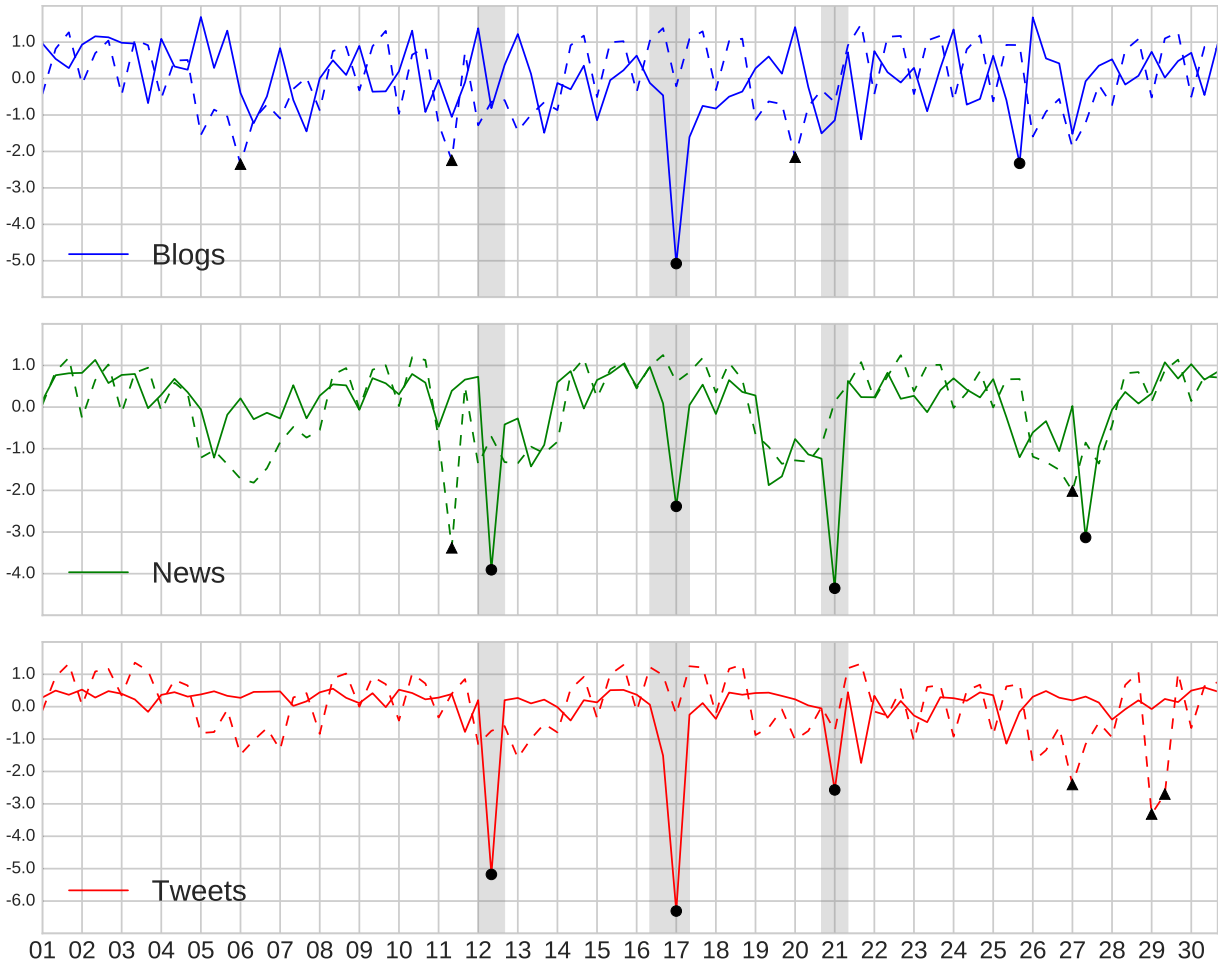


Figure 4.8: Standardised diversity scores for Donald Trump Presidential Campaign topic

Media	*Top 10 Topic Terms*
Blogs	pope, francis, church, catholic, visit, cuba, popes, climate, philadelphia, vatican
News	pope, francis, catholic, church, philadelphia, popes, cuba, united, vatican, visit
Tweets	pope, francis, visit, house, congress, popeindc, cuba, white, popeinphilly, philadelphia

In the case of news publishers, the largest drop in diversity coincided with the beginning of the Pope's visit to Havana. Twitter users and bloggers reacted more on September 23rd and 24th when the Pope met with Barack Obama and became the first Pope to address a joint session of US Congress.

In the Twitter stream, the notable event around 16th-17th is due to large numbers of similar tweets as preparations for the visit were being discussed, and #TellThePope trended briefly.

Earlier in the month, we see evidence of overlapping attention dominating events. Between 6th and 7th September, the Pope announced the Vatican's churches will welcome families of refugees.

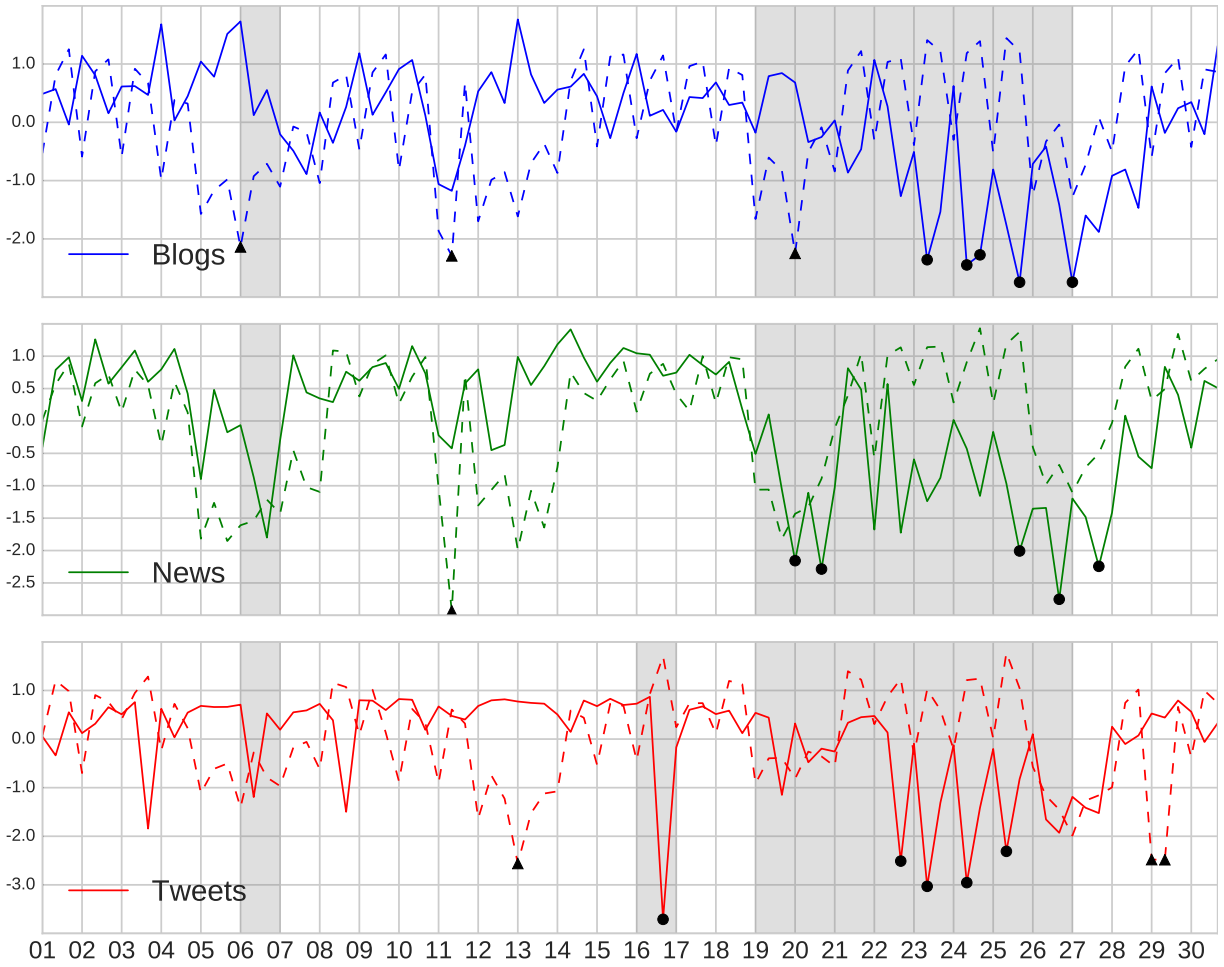


Figure 4.9: Standardised diversity scores for the Papal visit topic during September 2015.

This announcement followed a significant development in the ongoing European refugee crisis: around 6,500 refugees arrived in Vienna following Austria's and Germany's decision to waive asylum system rules. This suggests that an attention dominating news event in one topic can trigger events in other topics, especially where prominent public figures are involved.

4.3.6 Discussion

Consuming and participating in sharing news is a socially-engaging and socially-driven activity [203]. Measuring the diversity of text produced, as an indicator of what people are paying attention to, relies on people's tendency to form communities around news stories [45] and to participate in news sharing, often to positively influence their social capital [204]. In particular, the more engaging a news story is, the more likely it is to be shared by many people [103, 205], leading to more people talking about the same topic.

While the diversity measure we propose is relatively simple, it can be easily augmented to account for other factors. In the simplest form, every similarity value between a unique pair of articles within a time window carries equal weight in the diversity calculation, implying that a strong similarity between two highly influential publishers is just as important as between two inconsequential publishers with a small audience. However, this weight could be tuned, either manually or automatically using external information (*e.g.* Alexa rankings). Accounting for social context [206] could also be achieved by augmenting the topic modeling stage of the process. Instead of using a classic *tf-idf* vector space model, alternative representations that capture more semantic similarity between documents can be used. We aim to explore extensions to this measure in future work.

The sequence of events in the European refugee crisis and papal visit case studies suggest that it may be possible to identify and track major developments with global impact by linking attention dominating moments across multiple topics, as well as across sources on different platforms. Social media communities both influence and are influenced by traditional news media [196]. Stories break both on Twitter and through traditional news publishers. Tracking or linking instances of diversity collapse to explain the direction of influence between the different media types is also a potential avenue for future work.

CHAPTER

FIVE

DISCOURSE COMMUNITIES IN NEWS

Reporting on significant political events, such as elections and referendums, naturally involves the study of groups of people with different political views. Activities of political groups and individuals are of particular interest to journalists [207]. When new legislation is proposed in parliaments it often receives media attention. For professional journalists and citizen journalists alike, access to voting records offers an opportunity to examine an individual or a party group as a whole, and to study how their behaviour changes over time [208].

As Critical Discourse Analysis (CDA) is politically and socially motivated, it can be placed at the intersection of journalism and political science. Outcomes and analysis from CDA can both contextualise, and be informed by analysis of voting behaviour. Analysis of voting records from politicians allows journalists to examine both what people in positions of power promise in manifestos and statements of social media (*i.e.* the textual discourse), and how they act (*e.g.* vote on laws).

In this chapter, we first briefly examine the European Parliament. As a law-making body and venue for public debate of EU politics, the European Parliament holds significant influence. The decisions made by representatives can have far-reaching effects and often generates important debates in the media. In Section 5.1 *Dimensionality Reduction and Visualisation for Voting Records*, the decisions made by politicians are investigated using techniques that have previously been applied to text corpora, producing novel visualisations that group ideologically similar legislators in clusters.

For analysing communities central to significant news events and how they express their ideologies through text, we propose exploratory steps to reveal the *discursive strategies* used in order to advance and reinforce ideological positions. In Section 5.2 *Analysing Discourse Communities with Distributional Semantic Models*, we use distributional semantic models to capture associations between words specific to each community.

5.1 Dimensionality Reduction and Visualisation for Voting Records

Recorded votes in legislative bodies are an important source of data for journalists and political scientists. Voting records can potentially be a catalyst for breaking news stories for journalists. For political scientists, voting records can be used to describe parliamentary processes, identify ideological divides between members, and reveal the strength of party cohesion [209].

Both journalists and political scientists are interested in policy, and how ideologically opposing groups interact. Examining votes and party cohesion [210] requires some expert knowledge, and while visualisations cannot explain anything substantive about how "close" politicians and groups are to one another, they can create a useful visual summary that can generate new lines of inquiry.

In this section, based on [92] we explore the problem of visualising votes from ideologically diverse groups, using popular dimensionality reduction techniques and cluster validation methods, as an alternative to more traditional *scaling* techniques. We present results of dimensionality reduction techniques applied to votes from the 6th and 7th European Parliaments, covering activity from 2004 to 2014.

5.1.1 Introduction

As a law making body, votes passed in the European Parliament (EP) can have a significant influence on citizens across the European Union. Members of the European Parliament (MEPs) hold power over the majority of EU legislation, as well as decisions on budgets and spending. Analysis of votes is not only of interest to researchers, but many interest groups and industries operating within the EU. To produce insights into legislation and party politics computational approaches are highly dependant on latent variable models—using point estimates to make sense of and test theories using voting records [211], speeches [212], party manifestos [213], expert surveys [214], and more recently social media data [215].

A common theme in these models is the low dimensional reconstruction of high-dimensional data. Roll call votes, where the vote of each member is recorded are typically represented as a matrix of legislators with *for* and *against* votes, treating abstentions as missing values. Legislators, in this case, MEPs, are represented as vectors in d dimensions, where each dimension encodes a vote in some way. *Scaling* methods are then applied to recover point estimates or produce visualisations.

Scaling methods essentially perform dimensionality reduction, transforming data in a high-dimensional space to a space with fewer dimensions—an n-dimensional space $\mathbb{R}^n$ where $n << d$, typically 2 or 3 dimensions are used to produce interpretable visualisations.

While established methods for inductive scaling of roll call votes exist, there are many other potential alternatives that remain unexplored. We describe four such alternatives in Section 5.1.3, and formulate a cluster quality-based evaluation approach, highlighting the advantages and drawbacks of each method.

Data for 6th and 7th EU parliaments and Python code to reproduce the approaches on different sets of voting records are made available online[1], so that political science researchers can explore these alternative approaches when analysing vote data.

5.1.2 Related Work

The NOMINATE [216] family of multidimensional scaling approaches are the most widely adopted methods for estimating ideal points from roll call data, and have been applied to European Parliament roll call vote data in [211] where the main policy dimensions based on this data reveal a dominant left-right dimension, as well as evidence of a pro-/anti-Europe dimension. The results of scaling are often used as features for downstream tasks, such as [214] where ideal points are used as features in estimating party influence. In [217] roll call votes are compared to survey responses.

Scaling using text from speeches [212] can be related to the broader task of dimensionality reduction [218]. Popular scaling methods include Wordfish [219], and Wordscores [220]. The Wordfish model is applied to EP debates in [212]. While strong evidence for left-right ideology was not found in the speeches, the results suggest that legislators express ideology differently through speaking and voting. In [221] the voting records are combined with text contained in US House and Senate data, with ideal points estimated for topics such as health, military, and education.

What all these approaches share is a strong domain-specific focus: scaling approaches like W-NOMINATE [222] are developed specifically to deal with roll call votes and not any other kind of data.

We propose adapting dimensionality reduction methods which are not commonly used with roll call data, but have been previously shown to be effective elsewhere and are widely used across many other domains.

[1] `https://github.com/igorbrigadir/vote2vec`

5.1.3 Dimensionality Reduction for Roll Call Votes

We cast the problem of roll call vote analysis as a dimensionality reduction problem. We apply four methods (described below) to roll call voting records from the 6th and 7th European Parliament, testing alternative ways of encoding the vote data with different methods.

Voting in the EU Parliament

MEPs in the parliament are organised into transnational political groups. Group membership is based on ideological preferences of members from different countries, for example: Conservatives in one country will have more policies in common with conservatives in other countries, than with liberals in their own country.

These groups work together to divide the workload of drafting legislation, researching policy and other activities. The groups delegate experts to work on different issues, and agree to follow their instructions on the best voting strategy.

Given this organisation, MEPs have strong incentives to follow the voting patterns of their group [210]. The groups and their broad ideologies are summarized in Table 5.1.

MEPs do not always follow group voting decisions, but have strong incentives to do so, as the groups control the allocation of resources and committee positions.

Name	*Abbreviation*	Seats	*Ideology*
7th Term 2009–2014			
European People's Party (Christian Democrats)	EPP	274	Conservative
Progressive Alliance of Socialists and Democrats	S&D	195	Socialist
Alliance of Liberals and Democrats for Europe	ALDE	85	Liberal
European Conservatives and Reformists Group	ECR	56	Eurosceptic
Greens / European Free Alliance	G-EFA	58	Green
Group of the European United Left / Nordic Green Left	EUL-NGL	35	Radical Left
Europe of Freedom and Direct Democracy Group	EFD	33	Eurosceptic
Non-attached Members	NI	30	Various
6th Term 2004-2009			
European People's Party (Christian Democrats)	EPP-ED	288	Conservative
Socialist Group in the European Parliament	PES	217	Socialist
Alliance of Liberals and Democrats for Europe	ALDE	104	Liberal
Union for Europe of the Nations Group	UEN	40	Nationalist
Greens / European Free Alliance	G/EFA	43	Green
Group of the European United Left / Nordic Green Left	EUL/NGL	41	Radical Left
Independence / Democracy Group	IND/DEM	22	Eurosceptic
Non-attached Members	NI	30	Various

Table 5.1: Group names, seats, and ideologies for the 6th and 7th parliamentary terms. Number of seats doesn't reflect the number of MEPs active over the entire term, as some retire, or are substituted.

Encoding Vote Data

The EP plenary votes are publicly available and published regularly[2]. Before applying techniques to roll call votes, we construct the vote matrix X: the high-dimensional representation of votes—where an entry contains a binary value for *Yes*, *No*, and optionally *Abstain*, on each vote by an individual MEP.

A small example representing this encoding for two roll call votes for three different MEPs is shown in Figure 5.1.

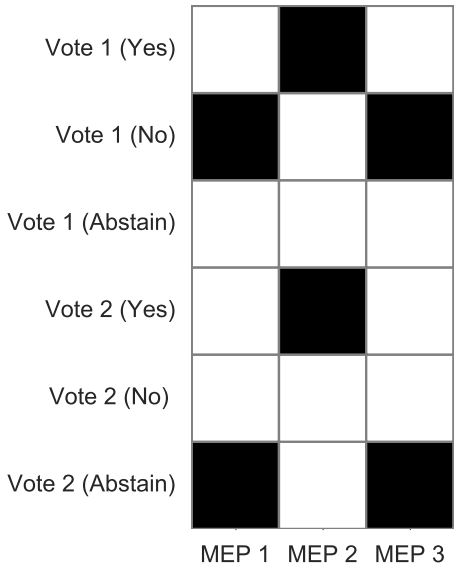


Figure 5.1: Example vote matrix: MEPs 1 and 3 voted *No* on Vote 1, and abstained on Vote 2. MEP 2 Voted Yes for both.

Other potential encodings, given vote metadata and method choice are possible: a count matrix is produced by merging votes using title similarity, or policy area or committee. Detailed vote metadata is available for the 6th parliament[3] from [210], but is incomplete for the 7th parliament. Results are reported for vote encoding using individual votes.

MEPs who switch groups [223] during the term present a data consistency challenge for roll call analysis using our proposed evaluation measure. MEPs who follow group voting procedure of one group for a period of the term, and then switch will be correctly clustered with the group most similar to them, but mislabelled during evaluation, as voting records remain, while group affiliation can change.

Every effort has been made to correct inconsistencies with data such as removing duplicate vote records and matching roll call records with MEP profiles to ensure MEPs represent the correct group at the time of the vote, but some inconsistencies may remain.

Dimensionality Reduction

W-NOMINATE: The Weighted Nominal Three-step Estimation approach [222] is an inductive scaling technique specifically designed for ideal point estimation of legislators using roll call data.

[2] http://www.europarl.europa.eu/plenary/en/votes.html
[3] http://personal.lse.ac.uk/hix/HixNouryRolandEPdata.htm

While the method is ubiquitous, a number of drawbacks are highlighted in [224]. Specifically: thresholds that exclude some votes, which results in poorer discrimination among extremist MEPs, and excluding MEPs with short voting histories. In the 7th Parliament dataset 5 of 853 MEPs and 460 of 6961 votes are excluded with the recommended settings.

The methods we propose do not exclude any MEPs or votes, and do not require setting vote or MEP specific thresholds, however, they do introduce their own method specific parameters and initialisation strategies that can impact results, and do not solve the problem of parameter tuning.

PCA: Principle Component Analysis [225] is a commonly used linear dimension reduction technique. PCA is performed using Singular Value Decomposition on the vote data matrix. Figures 5.4 and 5.5 show the resulting visualisations.

NMF: Given a non-negative matrix X, Non-negative Matrix factorization [226] approaches find two factor matrices W and H where the product of W and H approximates X. The dimensions of the factor matrices are significantly lower than the product. NMF is not commonly used for visualisation, but is a popular approach for clustering [227] and topic modeling.

t-SNE: t-Stochastic Neighbourhood Embedding is a popular dimensionality reduction and visualisation technique. Data is usually embedded in two or three dimensions, creating interpretable visualisations of high dimensional spaces.

The stochastic nature of the process can sometimes produce visualisations that are drastically different, or contain structure that could be over-interpreted.

For example, in a $2d$ plot, the x and y coordinates are not reliable values to use as point estimates in the same way as W-NOMINATE scores are—however, the clusters produced and relative positions of MEPs can be informative as MEPs with similar voting patterns will be clustered together.

SGNS with t-SNE: We explore a two-step process, where votes and MEPs are treated as co-occurrences—embedding votes and MEPs into a lower dimensional space with Stochastic Gradient Descent with Negative Sampling [228] and then applying t-SNE to further reduce dimensionality down to 2 or 3 for visualisation.

The two-step process tends to exaggerate distances between MEPs of the same group, however, this method introduces more parameters and instability, making qualitative analysis difficult and prone to over-interpretation—where visualisation artefacts can be interpreted as meaningful.

Evaluating Projections

While evaluating the usefulness of visualisations is often largely qualitative, it is useful to have an objective measure in mind when attempting to quantify the visual differences between alternative methods.

In order to evaluate the quality of the low dimensional projections of MEPs, we adopt Within-Group Scatter and Between-Group Scatter criteria, which have been widely used for the problem cluster validation [229]. Here we define our clusters as the parliamentary groups to which MEPs belong.

The between-group scatter quantifies differences in voting behaviour between-groups, while within-group scatter quantifies how cohesive a group is, or rather, how strongly party discipline dictates vote behaviour [210].

For group k, the within-group scatter is calculated as the within-group sum of squares, or $WGSS^{\{k\}}$:

$$WGSS^{\{k\}} = \sum_{i \in I_k} ||M_i^{\{k\}} - G^{\{k\}}||^2 \tag{5.1}$$

where G^k is the centroid of group k. The between-group scatter or $BGSS$ is

$$BGSS = \sum_{k=1}^{K} n_k ||G^{\{k\}} - G||^2 \tag{5.2}$$

where G^k is the centroid of group k, G is the centroid of all points (representing MEPs in a $2d$ space). Small $WGSS$ values indicate tight grouping of points in a cluster, or strong party discipline in the case of MEPs and votes. A large value for $BGSS$ indicates large differences between different groups.

5.1.4 Visualising 6th and 7th European Parliament

We now compare the outputs generated by W-NOMINATE and the alternative methods. Overall, in contrast to W-NOMINATE, the other methods have the advantage of significantly faster run times, but introduce method specific initialisations and parameters, which can affect visualisation output.

This is most pronounced in the case of t-SNE with random initialisation, where a cluster of MEPs may be placed "to the right" or "to the left" of another group depending on the run.

Initialising t-SNE with PCA produces stable arrangements of clusters in a $2d$ space, but the x and y values of individual MEPs are unsuitable for use as point estimates.

For $WGSS$ and $BGSS$ we exclude the non attached MEPs, as these are not members of any political group in the parliament. Ideology in the non-attached members ranges from communism, to populism, nationalism and neo-nazism.

Figures 5.2 and 5.3 show W-NOMINATE estimates that form our baseline: other approaches are compared to $WGSS$ and $BGSS$ scores derived from these results. Detailed scores by party group for the parliaments are shown in Tables 5.2 and 5.4 below.

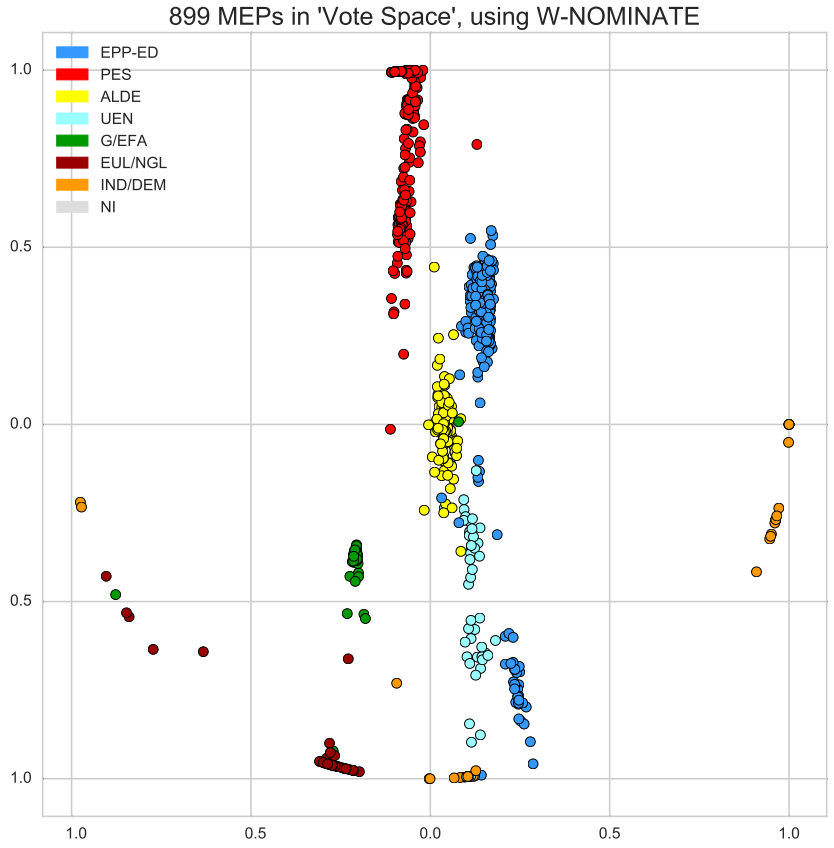


Figure 5.2: W-NOMINATE scales for the 6th Parliament.

In previous work, it has been suggested that the main policy dimensions in the European Parliament were a dominant left-right dimension, and a pro-/anti-Europe dimension [211].

In Figures 5.2 and 5.2, the x-axis is interpreted as the left/right dimension, with left-wing groups such as the European United Left / Nordic Green Left (EUL/NGL) placed on the left, and right-wing groups such as Europe of Freedom and Democracy (IND/DEM) on the right.

The y-axis is interpreted as capturing a pro/anti-EU integration dimension, with pro-EU groups assigned estimates close to 1 and Eurosceptic or anti-EU MEPs assigned point estimates close to -1.

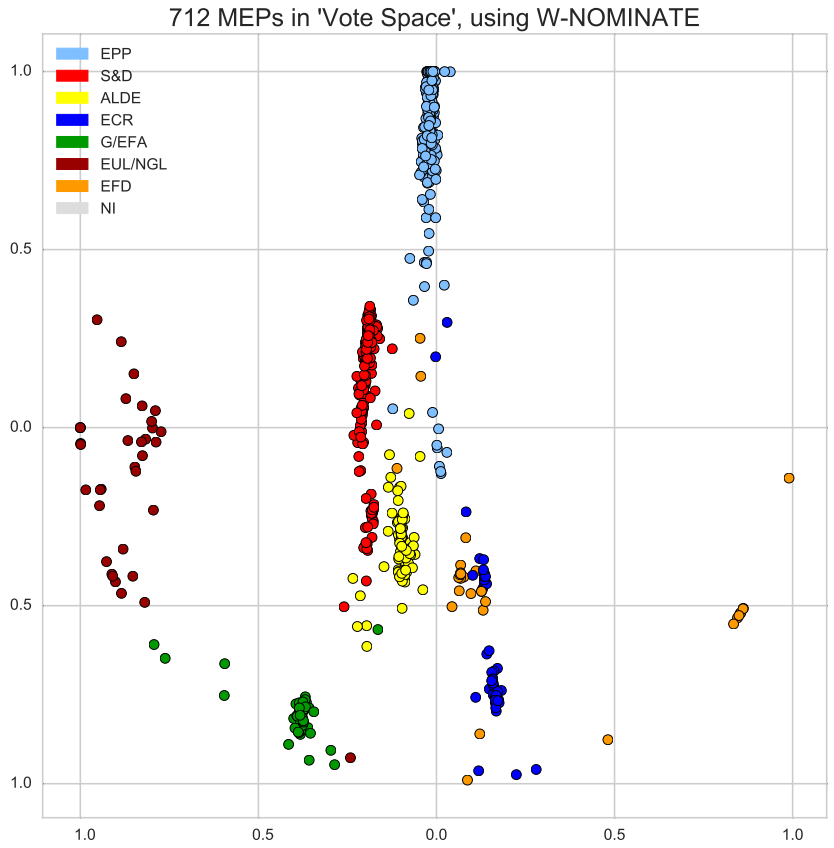


Figure 5.3: W-NOMINATE scales for the 7th Parliaments.

This interpretation for the dimensions is based entirely on face value, and should not be relied on as the only way to interpret an arrangement of points representing MEPs.

The same interpretation for left-right and pro-/anti-EU is less applicable to the other methods. For example, with t-SNE the positions of clusters can vary, *i.e.* a mirror image or rotation is just as valid.

The vote data specific W-NOMINATE approach enforces an orientation on the visualisation, while for general dimensionality reduction techniques this step must be performed manually.

The variation across different approaches is a good representation of the difficulty involved in drawing conclusions from voting records.

Figures 5.4 and 5.5 show an overview of all methods applied to the 6th and 7th parliamentary terms. In contrast to W-NOMINATE, the other methods have greater within-group scatter—exaggerating differences between MEPs in the same group.

While some groups are clustered more appropriately by the methods we explored, overall W-NOMINATE produces the best clustering of MEPs.

6th Term

The 6th parliamentary term began in July 2004, and lasted until June 2009. In total there are records for 899 MEPs. MEPs sometimes join the parliament at different times, retire, or are replaced. We include an MEP in a group if they have a record of a vote in the dataset.

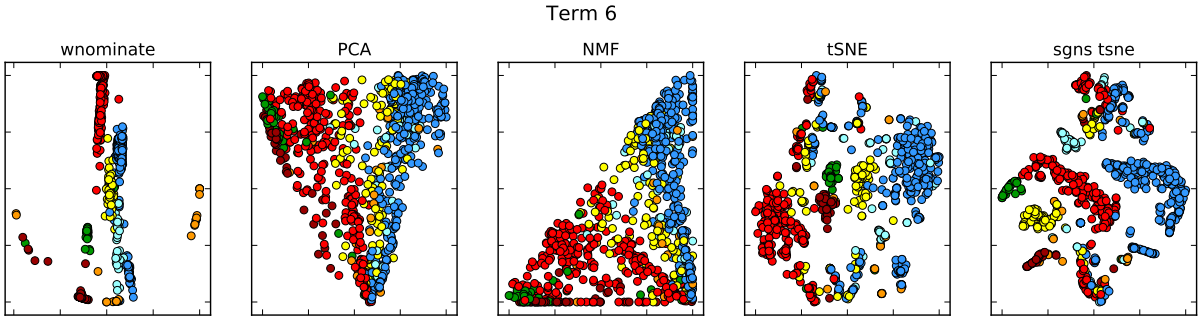


Figure 5.4: Overview of visualisations built on 6th Term voting records. Points are in clusters coloured by group.

Group	MEPs	WNOM.	PCA	NMF	t-SNE	SGNS
EPP-ED	340	43.22	139.52	108.88	144.38	101.81
PES	264	11.14	104.53	79.22	74.27	85.12
ALDE	125	1.22	50.96	39.07	32.10	40.65
UEN	51	3.32	16.66	13.73	16.51	12.67
EUL/NGL	48	2.17	16.24	14.39	5.79	11.70
G/EFA	44	1.08	10.47	9.52	2.54	5.71
IND/DEM	27	13.68	9.79	9.14	12.63	8.28
Overall	899	75.84	348.18	273.95	288.23	265.93

Table 5.2: $WGSS$: Within-Group Scatter, votes from 6th Term. Smaller values indicate that MEPs in a group are close to other group members in the vote space.

Group	MEPs	WNOM.	PCA	NMF	t-SNE	SGNS
EPP-ED	340	4.88	64.84	124.63	84.77	71.64
PES	264	106.88	45.62	88.63	104.96	6.75
ALDE	125	6.48	3.19	4.61	1.32	16.06
UEN	51	30.13	5.45	10.42	5.13	7.37
EUL/NGL	48	67.56	28.44	49.66	4.32	24.17
G/EFA	44	19.27	34.64	62.29	1.76	32.62
IND/DEM	27	18.92	3.35	2.08	4.57	5.10
Overall	899	254.13	185.53	342.32	206.84	163.71

Table 5.3: $BGSS$: Between-Group Scatter, using votes from the 6th Term. Larger values indicate greater separation between clusters of MEPs.

7th Term

The 7th parliament was elected in 2009 and finished in June 2014. Between the 6th and 7th parliaments, there were a number of changes made to groups, including new members and group affiliation switches by existing MEPs.

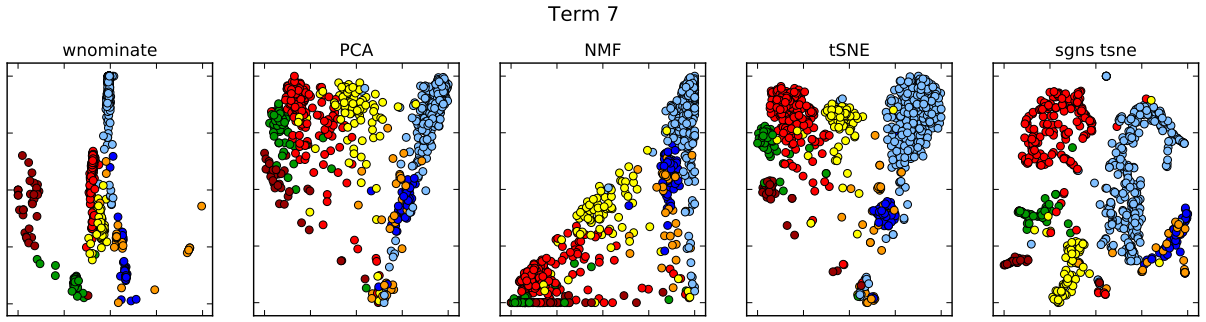


Figure 5.5: Overview of visualisations built on 7th Term voting records. Points are in clusters coloured by group.

Group	MEPs	WNOM.	PCA	NMF	t-SNE	SGNS
EPP	267	12.18	41.54	36.81	49.82	68.47
S&D	184	6.72	26.75	23.88	27.21	32.23
ALDE	85	0.96	15.55	13.91	12.36	14.33
G/EFA	56	0.70	7.89	7.06	6.88	2.99
ECR	54	3.17	3.80	2.83	2.97	3.20
EUL/NGL	35	2.59	6.45	5.96	5.31	4.60
EFD	31	5.85	6.97	3.91	6.50	4.22
Overall	712	32.17	108.95	94.35	111.06	130.03

Table 5.4: $WGSS$: Within-Group Scatter, votes from 7th Term. Smaller values indicate that MEPs in a group are close to other group members in the vote space.

Group	MEPs	WNOM.	PCA	NMF	t-SNE	SGNS
EPP	267	125.35	132.34	281.77	121.42	44.48
S&D	184	1.45	86.05	187.74	76.81	81.13
ALDE	85	22.32	2.64	4.05	2.11	40.16
G/EFA	56	57.96	44.33	92.59	42.30	25.37
ECR	54	39.57	28.56	18.36	29.45	44.77
EUL/NGL	35	23.11	35.09	43.00	31.59	37.77
EFD	31	16.52	20.80	7.84	20.19	25.22
Overall	712	286.29	349.81	635.33	323.86	298.89

Table 5.5: $BGSS$: Between-Group Scatter, using votes from the 7th Term. Larger values indicate greater separation between clusters of MEPs.

5.1.5 Discussion

While the methods we explore do not outperform the well established and widely used W-NOMINATE approach using a cluster validation based evaluation, there are a number of useful recommendations we can make when using different methods: NNDSVD initialisation strategy for NMF produces most stable results; PCA initialisation for t-SNE can help with stability of results. Even so, there is still a risk of over-interpreting the structure that t-SNE produces. UMAP [230] may offer a viable alternative to t-SNE, producing more reproducible visualisations.

Before drawing any conclusions from visualisations made with t-SNE, we recommend paying particular attention to the implementation and parameters, especially the learning rate used during optimisation. The SGNS approach allows most flexibility with encoding votes, but is the least stable method. The dimensions themselves from NMF, or t-SNE are not as useful for point estimates compared to W-NOMINATE, but the relative positions of cluster centroids offer a useful measure of similarity between-groups.

Many techniques are applicable if we treat roll call vote scaling as a dimensionality reduction problem. All methods that aim to project or embed high dimensional data in a low dimensional space introduce some uncertainty and instability.

Uncertainty in point estimates can come from many sources: from data quality issues and encoding schemes, to parameter and initialisation choices, to visualisation choices. Given these issues, one advantage that the alternative methods we explored have is their speed and efficiency: multiple runs under different settings can highlight errors in ideal point estimates more clearly.

In terms of evaluation, expert surveys [214] or coded party manifestos [213] may offer better benchmarks for differences between-groups and MEPs. Producing annotations and expert surveys is a costly task, and there are currently no expert judgements or annotations available for all votes for a full term.

5.1.6 Conclusion

We applied several commonly used dimensionality reduction techniques to voting records in the EU parliament. While all techniques tend to exaggerate distances between MEPs of the same group, they can perhaps be useful for quantifying within-party differences, or treating cluster centroids as points—similarities between-groups.

Applying similar methods to speeches and using point estimates derived from our proposed methods as alternatives in downstream tasks is ongoing, as well as comparisons of other projection techniques, applied to more recent data covering the current 8th parliamentary term.

5.2 Discourse Communities with Distributional Semantic Models

We now present a new corpus-driven approach applicable to the study of language patterns in social and political contexts, or Critical Discourse Analysis (CDA), which is based upon Distributional Semantic Models (DSMs). This approach considers changes in word semantics, both over time and between communities with differing viewpoints. The geometrical spaces constructed by DSMs or "word spaces" offer an objective, robust exploratory analysis tool for revealing novel patterns and similarities between communities, as well as highlighting when these changes occur. To quantify differences between word spaces built on different time periods and from different communities, we analyse the nearest neighbouring words in the DSM, a process we relate to analysing "concordance lines". This makes the approach intuitive and interpretable to practitioners. We demonstrate the usefulness of the approach with two case studies, following groups with opposing political ideologies in the 2014 Scottish Independence Referendum, and the US Midterm Elections 2014, based on work in [90].

5.2.1 Introduction

As discussed previously in Section 1.7, Discourse Analysis is concerned with the analysis of naturally occurring language use and patterns. Van Dijk [231] defined *Critical* Discourse Analysis (CDA) as:

> "...a type of discourse analytical research that primarily studies the way social power abuse, dominance, and inequality are enacted, reproduced, and resisted by text and talk in the social and political context."

Our focus here is on techniques for Critical Discourse Analysis applicable to opposing communities. A *discourse community* is a group of people sharing a set of basic values, assumptions and ways of communicating. Porter [232] offers a definition of a discourse community as:

> "...a local and temporary constraining system, defined by a body of texts (or more generally, practices) that are unified by a common focus. A discourse community is a textual system with stated and unstated conventions, a vital history, mechanisms for wielding power, institutional hierarchies, vested interests, and so on."

We are primarily concerned with the changes between, and within discourse communities over time. We explore how different political groups unified by a common focus (*i.e.* discourse communities) present themselves as defined by the body of text they generate on Twitter. This work proposes using word similarities from statistical semantics grounded in the Distributional Hypothesis, popularised by Firth [233] and adopts elements of the discourse-historical approach [76], a methodology that is problem-oriented, interdisciplinary, and recommends movement back and forth between theory and empirical data.

Current methods drawn from corpus linguistics that are used in critical discourse analysis, usually rely on keyword extraction and manual examination of "concordance lines" or "key words in context" (i.e. sorted and aligned lists of words with their surrounding contexts) but can also include topic modeling approaches. Emphasis is placed on generating insights into the ways in which the structures of text or speech relate to social and political contexts rather than on any particular approach.

The nature and volume of tweet text make these approaches challenging for several reasons. Firstly, poor sampling can lead to raw frequency counts of words to be skewed. Terse style and Twitter-specific use of user account mentions, hashtags and other media entities can also cause problems for methods that rely on frequency counts. Secondly, the sheer number of tweets available often makes close reading intractable, while distant reading techniques that look at the entire corpus can hide interesting periods of change and dynamics between communities.

The changes between and within communities with opposing political ideologies, manifest themselves as shifting distributional semantic similarities between words. We suggest that these changes can be quantified using Distributional Semantic Language Models (DSMs). The differences between semantic models derived from text produced by certain discourse communities can offer practitioners useful tools for CDA. These tools are more aligned with what Fairclough [234] calls *textually oriented discourse analysis*, examining how "the mode of language…identified as constitutive of power in modern society…is received and appropriated by those who are subjected to it".

Concretely, language models are used in this work as a supporting corpus-driven technique, providing entry points for further, more detailed analysis, allowing researchers to investigate how language use is shaped by political objectives.

As the primary contribution of this work, we propose a novel application of distributional semantic models for CDA, where constructing a DSM or word space model can be related to Key Words in Context (KWIC) analysis—a well-established, qualitative approach familiar in corpus-assisted CDA.

We evaluate the approach on two case studies from Twitter, each involving two distinct communities with differing political viewpoints—the 2014 Scottish Independence Referendum and the 2014 US Midterm Elections. As a further contribution, we also provide reusable datasets of tweets based on these case studies.

5.2.2 Related Work

A framework for using Corpus Linguistic methods for Critical Discourse analysis is presented by Baker et al. [235]. We take a similar position, arguing that since CDA lacks a concrete set of techniques for performing analysis, novel approaches can be made available to practitioners.

Social scientists and CDA practitioners are increasingly looking to social media as a rich source of data. Current corpus-based approaches and tools involve manual inspection of keyword frequency lists and reading concordance lines. Collocation analysis offers "a suitable vehicle for the discoursal presentation of a group" [236] but using plain frequency for collocation extraction yields general, uninteresting terms [237].

The methods proposed in this work are not related to Rhetorical Structure Theory (RST), a method for discourse *parsing* [75], concerned with coherence of multi-sentence texts. In contrast, we consider similarities at the word level, rather than sentence level.

Political Discourse on Twitter

An analysis of political discourse on Twitter by Zappavigna [238] suggests that users appear to bond around the act of collectively witnessing moments they perceive to be important to their cultural history, while politicians often use Twitter as a means of fostering engagement with others, offering positive evaluations of themselves and their parties. In our case studies, this promotional style adopted by official campaign accounts is also evident.

An in-depth study concentrating on politicians on Twitter is presented in [239]. Methods common in Information Retrieval have been applied to theoretical sociological constructs, deriving measures of "Cultural Similarity", Rank Biased Overlap measures for "Cultural Reproduction" and several others. The type of conversational practice (or discourse) examined included analysis of hashtags, retweets and mentions.

Political polarisation on Twitter is investigated by Conover et al. [240] through the analysis of mention and retweet interactions in the previous 2010 US Midterm elections.

The notion of "content injection" is also revealed using our proposed methods.

In our case studies, "content injection" is more pronounced in groups containing regular supporters of a particular ideology, rather than official function accounts such as campaign accounts of prominent politicians.

Related work that does not use Twitter data but deals with similar themes includes: summarizing contrastive views with augmented summarization techniques [241], performing comparative text mining and ideology classification with a topic modeling approach [242], and a network analysis approach for quantifying political polarity of individuals [243]. The problem of political alignment on policy issues, which is often cast as a classification task, is outside the scope of this work.

Distributional Semantic Models

Recently, *word2vec* [35] has been widely used to generate useful representations of words using a Neural Network Language Model (NNLM). This distributional semantic model offers efficient training times and performs well on a variety of semantic and syntactic word similarity tasks.

A comparison of distributional semantic models that involve context prediction and context counting is performed in [244]. Models were compared using a number of widely-used syntactic relatedness, synonym, concept categorisation and analogy tasks. Context-predicting models, such as *word2vec*, were shown to perform better than context-counting variants.

Linguistic Shift

Measuring linguistic shift with an information theoretic approach is explored by Juola [245]. Using a corpus of several decades of National Geographic publications, changes in language were not only perceptible algorithmically, but are also not uniform over time, suggesting that some periods of time are more actively changing than others.

Kulkarni et al. [246] developed a method for detecting significant linguistic shift in the meaning and usage of words, employing a DSM to construct a time series of word usage and a mean-shift change-point detection algorithm to estimate when this change occurs. In contrast to our work, the timespan involved is longer, covering two years for Twitter data and several decades for the Google n-gram set. However, a comparison between different clusters or communities is not considered by the authors.

In [247], a distributional similarity approach is compared to a relative frequency based approach, using two Google n-gram corpora from the 1960s and 1990s.

Another relevant approach that explores changing word meanings over a long period of time is described by Basile et al. [248]. Neighbourhoods of words are examined across several decades of Italian books and the ACL Anthology Network dataset.

Unlike previous work, in this study we focus on shorter time spans, seeking to identify changes between and within communities, as opposed to looking at changes across an entire corpus.

5.2.3 Exploring Discourse with Distributional Semantic Models

Common corpus-assisted techniques for discourse analysis include comparative word frequency lists, keyword extraction, and concordance lines or KWIC—showing the surrounding context of a keyword of interest (See Table 5.6). Typically, results are presen-

... #nothanks #indyref foremost authority on north sea **oil** throws doubt over snp prediction for the future of ...
... foremost expert felt he had to speak out to warn of **oil** depletion #indyref cant rely on oil to deliver public ..
... points out what john swimney said about volatility of **oil** @sygazette debate an injustice in one part of the uk ...
.. welfare health education and pensions costs 40 billion **oil** revenue 3 billion #no gb on the #nhs the ties that bind

Table 5.6: Example concordance lines, from Scottish Referendum tweets containing the word "oil".

ted as raw or normalised counts derived from the corpus, along with a qualitative assessment that involves a close reading of a selection of material. In collocation analysis [236] the most frequent co-occurrences may not be the most useful for CDA. To address the drawbacks of frequency-based approaches [237], we propose the use of a distributional semantic model that computes vector representations of words. The rationale here is that DSMs reveal different types of similarity and relatedness useful for CDA.

Motivated by results from Baroni et al. [244], we use a context-*predicting* distributional semantic model as opposed to a context-*counting* model. The task requires a good estimation of word similarity, as well as *association*. As a concrete example, the words "fields" and "oil" are not synonymous, describing two different concepts, but are *related* in the context of fossil fuels. Likewise, "oil" and "crude" are synonymous[4]. Both similarity and relatedness are useful to consider for CDA. It is important to note that DSMs have previously been evaluated with this distinction [249].

Figure 5.6 shows a toy example of two word spaces.

[4]The word "oil" or "crude" is frequently dropped from the phrase "crude oil", especially in length-restricted posts like Twitter.

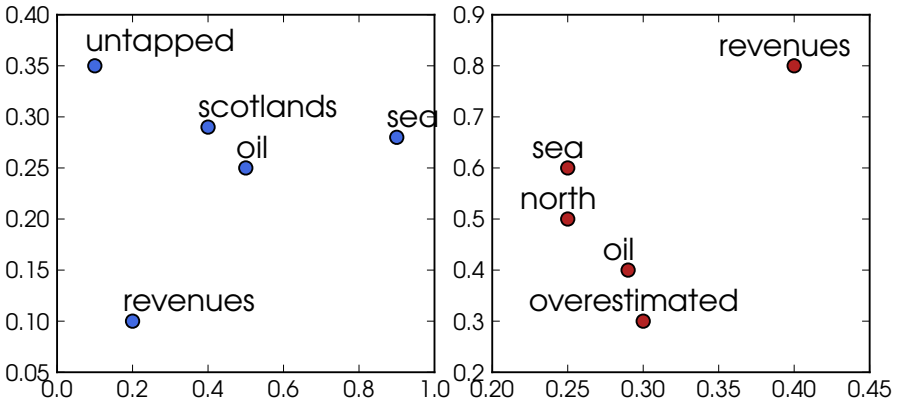


Figure 5.6: An illustrative example of word spaces with only 2 dimensions. Words are represented by vectors with 2 components, x and y values. In practice, trained word vectors have hundreds of components.

In the space on the left, the word "oil" is represented by a vector $[0.50, 0.25]$, and on the right $[0.29, 0.40]$. The components of the vectors do not represent actual counts or occurrences, but after training a DSM, words that are more *related* are closer to one another. We can compare the two spaces by looking at what words are closest to one another in each space. In practice, words are represented by hundreds of dimensions.

Discourse-historical Approach

A suitable framework for CDA is Wodak's discourse-historical approach (DHA)[76]. DHA emphasises the interpretation of discourse in its historical and cultural contexts. Four levels of context are suggested:

1. The immediate *co-text* for a particular linguistic feature found in the text: Involves analysis of the text itself. Our proposed approach provides most benefit at this level, suggesting entry points for further, more in-depth analysis.
2. Other texts and discourses that the text draws upon: In a corpus of tweets, this level would include analysis of media linked and referred to in tweets.
3. Conditions of text production, distribution and reception: This level is constrained by the platform, particularly in terms of distribution and reception with use of hashtags and mentions [238].
4. Wider social, political, economic and cultural contexts.

For analysis of groups with opposing political ideologies, DHA recommends six *discursive strategies* for identifying ideological positioning, summarized with questions below:

1. Nomination: Constructing in-groups and out-groups via membership categorisation. How do different groups categorise themselves and opposing groups? Do these change over time?
2. Predication: Labeling social actors positively or negatively. How are key individuals represented by different groups? How is this reproduced in tweet text?
3. Argumentation: Justifying positive or negative attributions, political inclusion or exclusion.
4. Framing: Expressing involvement through reporting, the narration of events and utterances.
5. Intensification / Mitigation: Modifying a proposition by intensifying or mitigating the force of utterances.

As a motivating example, consider discursive strategies used in the following tweets:

```
Why do Nats want Scotland to be one of Europes vulnerable, marginal economies?  We
truly are #BetterTogether #IndyRef

Nationalist lies over oil @YesScotland @UK_Together #idyref No Boom No Oil Bonanza
#ProjectFear
```

We can manually identify several interesting keywords in the text of these tweets: "nats"—nationalists (nomination, membership categorisation), "lies" (predication, labeling social actors negatively), "bonanza" (intensification).

Closely examining a large volume of tweets this way is impractical, and while some levels of context require a close reading, analysis of the text itself can be performed at scale, using corpus-driven approaches. A concrete example of where DSM approach can help DHA, is in exploring *discursive strategies* in racial, national and ethnic issues. Questions like "How are persons named and referred to linguistically?", "What traits, characteristics, qualities and features are attributed to them?" prescribed in DHA to explore "positive self" and "negative other" presentations can be answered with a combination of examining nearest neighbourhoods of words, and closer reading of selected tweets.

As a starting exploratory step for our analysis, we examine the different communities using a small selection of words representing topics of interest which are known *a priori*. We then examine some *discursive strategies* in the communities. This is firstly performed across communities over the entire period, and then in more detail, looking both within and across communities over shorter time periods. We then expand this set of words, by examining the k-nearest neighbouring words for the communities, in order to discover interesting commonalities or differences between them.

Restricting the nearest neighbour search to consider either words or hashtags alone could potentially provide alternative lines of inquiry [240]. However, we follow a more general approach, allowing for a mix of words, mentions and hashtags to appear, but excluding URLs which appear in tweets. The method is an iterative, word-level approach to critical discourse analysis that alternates between exploration, and close reading.

To summarize the prescribed process: 1) select initial candidate words of interest, 2) examine the word change "profile" visualized as a trend, 3) examine word neighbourhoods, 4) retrieve relevant tweets for a more qualitative, closer reading of texts, and 5) repeat the process armed with new keywords or hypotheses.

Distributional Semantic Approach

Mikolov et al. proposed a DSM that performs well on a variety of syntactic and semantic relatedness tasks [136]. The *skip-gram* training process learns word representations that are useful for predicting nearby words (the context). From a sequence of words $(w_1, w_2, \ldots w_T)$, the objective maximizes the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

where the context size c is the number of words before and after the target word w_t. It is interesting to note that the mechanics of "key words in context" analysis roughly maps to the training objective of the DSM approach.

When analysing *collocates* (*i.e.* words that co-occur more often than would be expected by chance), Mutual Information (MI) is a commonly used association measure. For instance, the popular *AntConc* concordance tool [250] uses Mutual Information to rank collocates.

Pointwise Mutual Information (PMI), proposed by Church and Hanks [251] is another widely-adopted association measure. If two outcomes x and y have probabilities $P(x)$ and $P(y)$, then their Pointwise Mutual Information $PMI(x, y)$ is defined as:

$$PMI(x; y) = \log \frac{P(x, y)}{P(x)P(y)}$$

Levy and Goldberg show that the *skip-gram* training process implicitly factorizes a word-context matrix, the cells of which are Pointwise Mutual Information (PMI) of word and context pairs, shifted by a global constant [228].

The *word2vec* model, trained with *skip-gram* is a good choice for the purpose of quantifying semantic & syntactic similarity between words. Similar models can also be used, but *word2vec* was chosen for having a good trade-off between performance and training time.

Comparison of KWIC & k-NN Lists

Manually examining KWIC lists will often be unfeasible for larger corpora, such as a collection of tweets. Sampling a selection can introduce bias, while a *close reading* of each and every document in a collection is impractical. Collocates are generally useful for "a semantic analysis of a word" [252].

Examining these collocation patterns can provide a corpus-driven tool for CDA, as used in an analysis of the representation of refugees, asylum seekers, and immigrants in the UK press [236]. Our approach can be used to discover similar patterns, where predicting a set of contexts given a word can be interpreted as an aggregation of concordance lines, drawing an analogy between the training objective and KWIC analysis familiar to practitioners.

Table 5.7 shows how different measures of association can provide different collocates. The *Frequency* column shows terms ordered by their raw frequency counts. *Mutual Information* is a measure of the strength of association, based on the number of times a word pair is observed together versus the number of times the words appear separately. The ordering in the *word2vec* column is based on the cosine similarity of the word vectors in the trained model.

Rank	*Frequency*	*Mutual Information*	*word2vec*
1	important	scotenergynews	untapped
2	#indyref	pegging	revenues
3	tank	kuwaits	pouring
4	oil	headlined	#clairridge
5	#scotdecides	@conhome	recoverable
6	#yes	kindness	300bn
7	#westcoastoil	exploration	rig
8	thousands	@yuillnoodz	bonanza
9	#voteyes	@wynnscottishsun	#northseaoil
10	north	@wulliekane	asianomics
	…	…	…

Table 5.7: Comparison of collocates considering 5 words before and after the word "oil" from 7 days of 'Yes' vote supporters based on raw frequency, Mutual Information, and *word2vec*.

Frequency-ranked terms produce what may be considered uninteresting associations, such as the word "oil" itself, and some frequently used hashtags.

Mutual Information generally surfaces technical terms, phrases, and any other terms where the frequency of co-occurrence is high but overall frequency of terms is not very frequent, *e.g.* the @mentions in the list for the word "oil".

The advantage of using a *word2vec* model, in this case, is that searching the trained model for nearest neighbouring words is extremely fast, and provides meaningful results, without over-promoting highly-rare or highly-frequent terms. A drawback to our approach is that training the model requires a relatively large corpus of text, and introduces extra hyperparameters to consider. However, applying these corpus-assisted techniques over a stream of documents can reveal more nuanced changes in discourse. These changes can potentially be related to external events or can serve to quantify the evolution of a discourse community over time.

Rather than examining the social network structure of different communities, k-nearest neighbour graphs (k-NNG) can be used to examine distinctive linguistic similarities and differences between discourse communities.

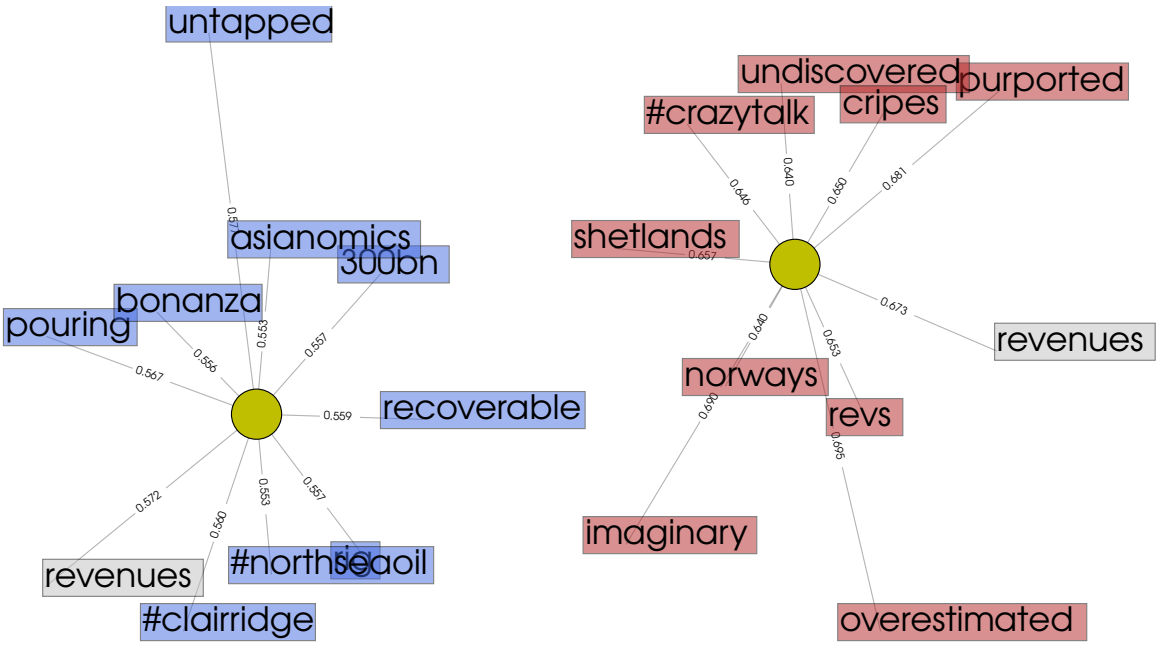


Figure 5.7: A sample k-nearest neighbour graph for the word "oil" for Scottish 'Yes' and 'No' voters. Words unique to a community are coloured, words in gray are common to both communities.

As an example, Figure 5.7 shows the word neighbourhoods of the word "oil" from two different communities in the Scottish Referendum campaign. Nodes are other related words, and edge lengths are inversely proportional to the cosine similarity of each word to "oil" in the community-specific word space.

Accounting for Change Over Time

Since discourse communities are temporary systems defined by a body of texts [232], we must account for time in the models. As *word2vec* does not account for a temporal dimension in texts, we propose splitting the dataset of each community into windows, each covering different time periods. The conversion of continuous data streams, such as content from social media platforms to discrete windows, has been a common strategy in the analysis of online communities [253].

Separate models are then trained for a number of fixed length windows of tweets, creating different models for each window and each community. Models are trained using the *skip-gram* architecture, with vector size 300, context window of 5 words, for 30 iterations on time windows spanning 7 days.

Due to the stochastic nature of *word2vec* training, and the different training sets, the various word spaces created are not directly comparable. However, we are not interested in the resultant model representation, but rather the relationships between words that can be interpreted by practitioners.

Therefore, for a given word, we retrieve its k nearest neighbours in the model, and present these for consideration. This is analogous to the way in which a KWIC analysis presents concordance lines. From this, quantifying the changes between time windows and communities can now be accomplished by comparing the *word neighbourhood* of a particular word in different spaces—*i.e.* the similarity of the word's k nearest neighbour lists in each model.

Average Jaccard for k-NN

To compare word neighbourhoods, we require a suitable similarity measure. The k-nearest neighbours of words in a word space, when viewed as rankings, are incomplete (*i.e.* not all words are covered), top-weighted (*i.e.* top-ranked words are more important), and indefinite (*i.e.* choice of k is arbitrary). A desirable measure should account for these properties.

The *Average Jaccard* (AJ) measure from Greene et al. [254] used for comparing ranked lists has the required properties. Though distinct, AJ can be related to *cultural reproduction* [239], as both measure a form of rank-biased overlap.

We calculate the AJ scores between the k nearest neighbouring words from two spaces. The two spaces can either be two time windows from the same community, or the same time window from two different communities.

As AJ is top-weighted, increasingly higher values of k have a decreasing influence on the overall score. The choice of k is largely influenced by the need for rankings to be examined by practitioners. In both case studies presented later, we set $k = 30$, but this parameter can be varied to consider more distant words.

Jaccard similarity between two sets is defined as the size of the intersection divided by the size of the union. The Average Jaccard (AJ) between two ranks A and B to depth k is defined as the average Jaccard scores between subsets of d top-ranked words in two rankings, where d is $d \in [1, k]$.

$$AJ(A, B) = \frac{1}{k} \sum_{d=1}^{k} J_d(A, B)$$

where $J_d(A, B) = \frac{|A_d \cap B_d|}{|A_d \cup B_d|}$ and A_d,B_d are the heads of lists up to depth k. See Table 5.8 for a worked example.

k	*Rank A*	*Rank B*	*Jaccard at k*	*AJ at k*
1	untapped	untapped	1.00	1.00
2	revenues	yada	0.33	0.66
3	pouring	reserves	0.20	0.51
4	#clairridge	bonanza	0.14	0.42
5	300bn	revenues	0.25	0.39
	...	...	...	...

Table 5.8: Average Jaccard values at different values of k, comparing the word neighbourhoods of "oil" from two consecutive weeks from 'Yes' voter tweets.

5.2.4 Datasets

Various selection criteria exist for gathering Twitter corpora for the study of politics and political discourse. A recent survey [255] offers a comprehensive overview of data sources and collection techniques. Gathering all tweets from a subset of users was shown to provide a much richer and trustworthy source of data as opposed to a random sample of tweets from all users [130]. Problems arising from monitoring social media based on the pre-selection of specific hashtags or keywords have also been discussed in the literature [41].

Rather than relying on keyword or hashtag searches, for our experiments, we gathered all available tweets for a fixed subset of users. For both the Scottish Referendum and US Midterm Elections, users were first selected by their "official function"—*i.e.* politicians, campaign accounts, political organisations. Additional accounts included in each set are detailed in 5.2.5 and 5.2.6.

During data collection, users are automatically notified when added to a Twitter list, and have the ability to remove themselves by "blocking" the account used to add them, or by making their account private. Several accounts were either deleted or made private during and after the data collection period.

For pre-processing, common stop words, those words occurring less than twice, and URLs are removed from tweet text. The default NLTK English stop word list[5] was expanded to include several Twitter-specific function words such as "ht", "via", "mt".

The dataset was post-processed to honour deletion requests and user privacy settings. A summary of the data is shown in Table 5.9.

Community	*Users*	*Tweets*	*Total Words*	*Date Range*
Scotland Yes	618	799,096	12,551,654	11-Aug to 19-Oct
Scotland No	610	570,024	8,957,721	11-Aug to 19-Oct
Democrat	942	89,296	1,404,737	10-Oct to 20-Nov
Republican	997	80,840	1,209,197	10-Oct to 20-Nov

Table 5.9: User, tweet and word counts for Scottish and US datasets. Date ranges are in 2014.

The sets of tweet IDs and users are available, together with tools to retrieve and reconstruct the dataset[6]. While classifying polarity and party affiliation is outside the scope of this study, this dataset potentially offers a useful ground truth for such tasks.

5.2.5 Case Study: 2014 Scottish Referendum

The Scottish Independence Referendum, which took place on 18th of September 2014, decided Scotland's membership in the United Kingdom political union. The single question posed by the referendum—"Should Scotland be an independent country?"—generated considerable debate on social media platforms in the weeks before the vote.

Both the official *Yes Scotland* and *Better Together* (No vote) campaigns were established in 2012, with the date of the referendum set in March 2013, and legislation passed in November 2014. While the lifetimes of these campaigns were long, the majority of activity occurred within weeks of the referendum. We consider tweets over a time span of 10 weeks (11 August to 19 October 2014), for communities of 'Yes' and 'No' supporters.

[5] `http://www.nltk.org/nltk_data/`

[6] `http://dx.doi.org/10.6084/m9.figshare.1430449`

Scottish Voter Communities

An initial seed list of Twitter accounts belonging to "registered campaigners" on the Scottish Independence Referendum Electoral Commission was built. As the number of these *official function* accounts was small, additional accounts were added to the set based on public Twitter lists the seed accounts were members of. Parody accounts, non-partisan organisations, and users with private accounts were removed.

To be included as a "Yes" or "No" supporter, users had to self-identify through prominent use of campaign profile banners (party logos and campaign icons in profile images were popular with both sides), explicitly stating an affiliation in their user descriptions (*e.g.* using `#BetterTogether`, `#iVotedYes` etc.), and actively engaging with referendum topics.

Data from Twitter showed the 'Yes' campaign was dominant in terms of volume and participation, skewing some predictions and online opinion polls in their favour[7]. Polls that relied on interviews showed more support for a 'No' vote[8]. Ultimately, Scotland remained part of the United Kingdom, the 'No' vote gathering 55.3% and 'Yes' 44.7%, with a turnout of 84.6%, one of the highest recorded for a referendum or election in the UK.

Key issues in the campaign included: EU membership and currency, health care, education and research funding, Scotland's renewable energy and north sea oil revenue, NATO membership, and the issue of British Trident nuclear missile system on Scottish territory.

Using the analysis methodology proposed in Section 5.2.3, an initial set of words relating to these issues was selected. This was followed by an exploration step, adding related words, and removing those words that did not feature prominently in either community.

While neighbour graphs such as Figure 5.7 can be illustrative for small examples, a network visualisation of larger word spaces will quickly become an uninterpretable "hairball". The differences between time windows are also not evident. As an alternative, we suggest a trend visualisation to compare the similarities between communities and time windows.

Figure 5.8 provides a sample comparison of word neighbourhoods between 'Yes' and 'No' voters over 10 weeks. A point in the trend is the AJ similarity between word neighbourhoods from different communities, for a window of a single week.

[7] `http://blog.twitter.com/en-gb/2014/indyref-at-the-polls`
[8] `http://survation.com/?s=Scottish+Referendum`

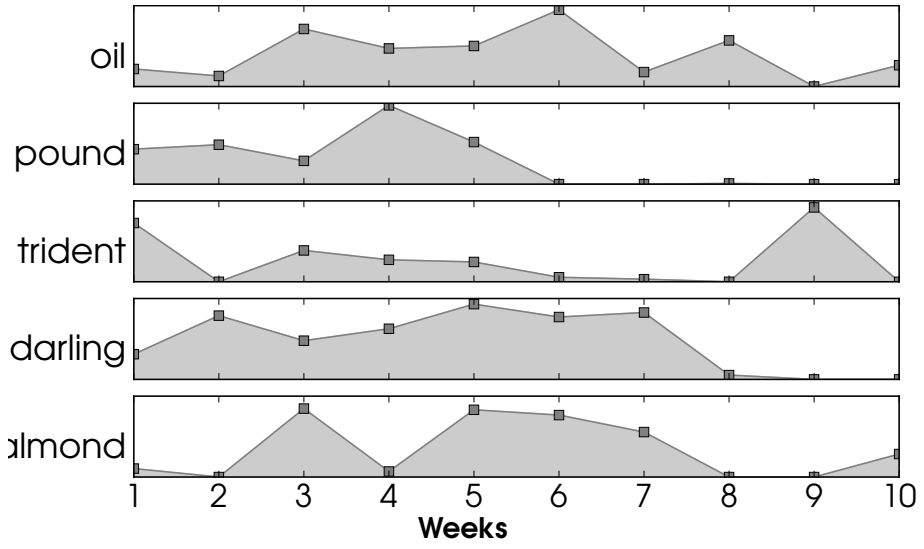


Figure 5.8: Trends illustrating the Average Jaccard similarity changes between word neighbourhoods for the 'Yes' and 'No' communities, over 10 weeks of the Scottish Referendum campaign. High similarity indicates agreement between communities, while low similarity indicates greater difference in how a word is used.

A more detailed understanding of a given point can be supported by looking at the corresponding word neighbourhoods for each community for that the time window, as illustrated by the ranked lists of words in Table 5.10. This can subsequently be used to retrieve the relevant tweets for a closer reading and analysis.

Yes Week 5	*No Week 5*	*Yes Week 6*	*No Week 6*
untapped	fields	norways	revenues
yada	rigs	revenues	ridge
reserves	rosy	reserves	**bonanza**
bonanza	revenues	147bn	sea
discoveries	downgrade	discoveries	4bn
bloddy	**bonanza**	chevron	offset
1billion	inflated	**bonanza**	estimates
…	…	…	…

Table 5.10: Yes and 'No' Voter word neighbourhood of "oil" corresponding to Figure 5.8 in weeks 5 and 6. Top 7 words are shown from the 30 used in analysis.

Discursive Strategies

Predication is an important discursive strategy with the objective of labeling social actors, used for reinforcing the construction of "us" and "them" between the 'Yes' and 'No' voter communities. These "positive self" and "negative other" presentations can be extracted from the nearest neighbouring terms used to refer to political leaders.

Table 5.11 shows a selection of nearest neighbouring words from 'Yes' and 'No' voters, for terms that refer to political figures central to the campaigns. The k nearest neighbours for Table 5.11 are derived from a model trained on tweets in the entire date range, before during and after the referendum.

Alex Salmond		**Alistair Darling**	
Yes Voters	*No Voters*	*Yes Voters*	*No Voters*
lucid	frantical	adversarial	commanding
authoritative	misdirection	bluffing	principled
statesman	fraudster	dismissive	quizzing

Table 5.11: Sample nearest neighbour words for "salmond", @alexsalmond, #alexsalmond, and "darling", @togetherdarling, #alistairdarling

The *Referential / Nomination* strategy in the discourse-historical approach is used for constructing in-groups and out-groups, and categorising memberships. A key advantage of using the DSM in this task, is that all tokens (individual words, hashtags, mentions) are in the same "word space" and their similarity can be compared—however, this process requires a practitioner to perform several searches: first to identify which nearest neighbouring terms are used to refer to a social actor ("salmond", "@alexsalmond", "#alexsalmond") and then retrieve some sample tweets for context.

In terms of *argumentation & framing*, there is evidence for *content injection* [240] in the time windows with highest similarity between the two groups. Our method suggests that this strategy is effectively reproduced in tweet text with hashtags, evidenced by the appearance of hashtags from the opposition in the nearest neighbouring term lists.

Twitter users would temporarily adopt hashtags popular with ideologically opposing groups, in order to spread and reinforce their political views. Below are examples of content injection from 'No' supporters, using `#yesscot`, and 'Yes' supporters using `#bettertogether`:

```
independence would bring a new wave of austerity for families in
scotland #indyref #yesscot #nothanks
```

```
why voting 'No' is a huge mistake #bettertogether #yesscotland
#indyref [link]
```

The debate around North Sea oil revenues featured frequently in Twitter discussions on both sides. In Figure 5.8, the "oil" row in Week 6 has a high AJ similarity. Both groups had "bonanza" in the word neighbourhoods, listed in Table 5.10. This revealed an interesting case, where 'Yes' voters were sharing an old article from 2013[9], while 'No' voters were quoting a correction to another news article from Prof. Alex Kemp, director of Aberdeen Centre for Research in Energy Economics:

[9] https://twitter.com/BizforScotland/status/510166055659270144

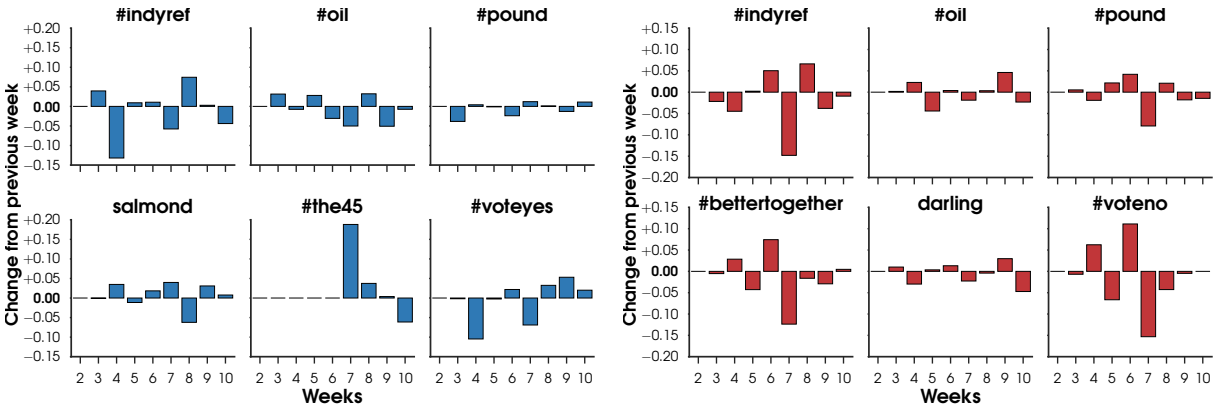
```
There will be 'No' oil bonanza - respected researcher Prof Alex Kemp
P&J 12/9/14 #indyref
```

Manually examining concordance lines for "oil", such as those illustrated in Table 5.6, would entail reading through thousands of entries, whereas the proposed method offers an immediately useful starting point for further exploration into how different groups appeal to authority in order to disseminate their ideas and exert power over one another.

Variation Within Communities Over Time

Looking at each community individually, we can begin to formulate an explanation for why and how these variations emerged in terms of social, political, internal and external influences.

Figure 5.9 shows temporal changes between word neighbourhoods for the two respective communities. These are derived by calculating the AJ similarity between neighbourhoods for each week with those from the previous week, for the same words and within the same community. Bars above 0.00 indicate increasing agreement within a community, while bars below 0.00 show a decrease in AJ scores between two consecutive weeks, indicating a larger difference between word neighbourhoods.



(a) "Yes" community word neighbourhoods.

(b) "No" community word neighbourhoods.

Figure 5.9: Temporal changes in within-community AJ similarity. This shows how much change there is in the word neighbourhoods over time.

The impact of the day of the vote, and the winning announcement can be seen within the communities between weeks 5 and 6 on the x-axis in Figure 5.9 a and b. Naturally, there is an upsurge in agreement for `#bettertogether` & `#voteno` within the community as 'No' supporters celebrated the result.

Zappavigna in [238] describes how Twitter users bond around a moment they perceive to be important to their cultural history.

After the vote results are announced, the changes within the 'No' community diminish, while the 'Yes' voters, form a brand new label (*nomination* strategy). The 'Yes' voters rapidly adopted `#the45` hashtag, rallying supporters around a new in-group. `#the45` refers to a rounded figure of 45% counted for the 'Yes' vote.

The 'Yes' voters were much more active and engaged on social media, but this activity did not seem to translate into a higher turnout for the 'Yes' campaign. In a meta-analysis of social media usage [256], while there may be a positive relationship between social media use and voter participation, whether or not this relationship is causal and transformative is questionable.

There are many other examples where interesting deviations in discourse between and within communities can serve as a guide for further, more qualitative interpretation.

5.2.6 Case Study: 2014 US Midterm Elections

Midterm elections in the United States are held near the midpoint of the four-year presidential elections. In 2014, elections were held on November 4th, involving seats being contested for the House of Representatives and Senate, along with governorships and a variety of local positions.

Several key topics dominated the elections: immigration, national debt, jobs and minimum wage, and fears of an Ebola outbreak in the US.

Midterm Elections Communities

Several official and unofficial sources listing Twitter accounts of incumbent and challenger campaigners were merged and segmented into Republican and Democrat groups. Third parties were not included in this case study.

Official sources included verified government accounts listed by the `@gov` Twitter account, and accounts linked from the House of Representatives[10] and Congress member pages[11]. Twitter accounts advertised on these pages were included in the set.

The majority of these accounts were verified by Twitter, and were either official campaign accounts of representatives run by staff, or their personal accounts which in many cases were also run by staff for the duration of the campaign.

[10] `http://house.gov/representatives`
[11] `https://www.congress.gov/members`

While Midterm elections do not involve the same level of activity as presidential elections, a qualitative analysis of Twitter feeds and interviews with campaign staff from the 2012 Presidential Election by Kreiss [257] offers an insight into the use of Twitter by campaign staff to frame an agenda and engage with supporters.

For initial exploration, words associated with issues outlined by The Brookings Institution[12] were used.

Figure 5.10 shows how similar Republican and Democrat word neighbourhoods were over time. For example, for "debt" both groups were more similar to one another in the first two weeks, then diverged.

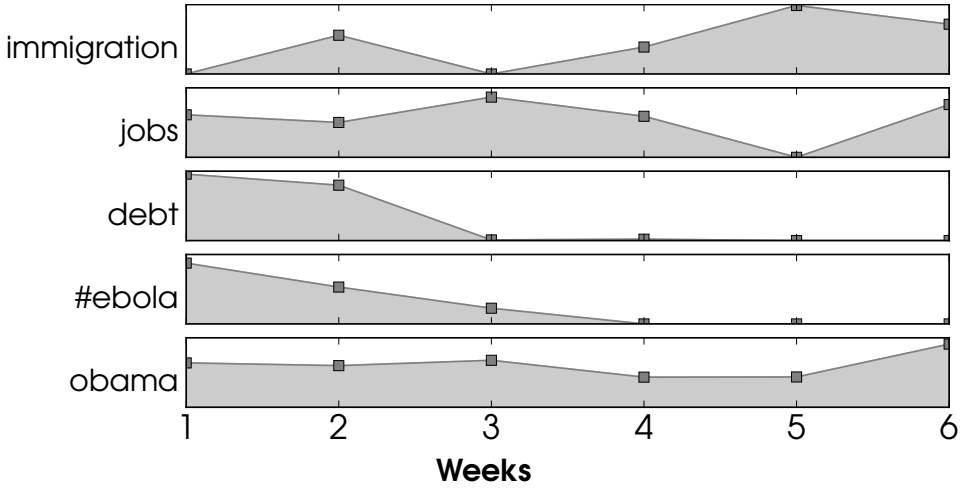


Figure 5.10: Trends illustrating AJ similarity changes between word neighbourhoods for Democrat and Republican communities. High similarity indicates agreement between communities, while low similarity indicates greater difference in how a word is used.

The issue of immigration revealed an important difference between Republican and Democrat candidates. Examining the nearest neighbours of "immigration", in word spaces built on Republican accounts, there were many more hashtags such as `#noamnesty` and `#amnesty` in contrast to Democrat word spaces, where "immigration" was associated with "reform" and "senate".

Discursive Strategies

The *framing* strategy was by far the most prominent in this case. Official campaign accounts rarely expressed or argued a stance on an issue, but did reference content elsewhere - manifestos on websites, interviews, etc.

[12] `http://www.brookings.edu/research/flash-topics/flash-topic-folder/2014-midterm-elections#state`

The majority of contentious conversation happened away from official accounts, among supporters and journalists. Finding this type of political discourse supports findings in [238], where politicians mainly use Twitter to foster engagement with their supporters, offering positive evaluations of themselves with a promotional style. In this study, we did not consider these other texts, restricting the corpus to tweets alone.

As an alternative to selecting words of interest, a full search through the word spaces using the *AJ* similarity measure can rank the most similar word neighbourhoods discussed by Republicans and Democrats during the election period. This has the advantage of discovering surprising instances of *mitigation* or intensifying utterances reproduced in text.

The top-ranked terms before and during elections included "#ebola", "jobs", and "halloween". Just after the elections, the most similar words included "birthday" associated with the 239th Birthday of the Marine Corps.

Neither community attempted to steer the conversation into some of the more contentious topics relating to veteran care or troops overseas around the time of the Marine Corps birthday celebrations. The *framing* strategy used by both sides amounted to sharing the same video messages and congratulations.

Variation Within Communities Over Time

Plots of the within-community temporal changes for word neighbourhoods (see Figures 5.11 and 5.12) show a large spike in similarity within the Republican community for "amnesty", and Democrat community for "immigration". This may be largely due to Obama's immigration reform speech that aired on November 20th.

Both communities, individually, expressed support for their official party line, as evidenced by a high similarity over time *within* each community.

Discussions around jobs and employment featured frequently in both Republican and Democrat campaign accounts (See Tables 5.13 and 5.12).

Both parties brought out announcements that thousands of new jobs need to be created, and frequently cited legislation on which they either voted, or will vote if re-elected.

Republicans tended to promote energy sector growth, while Democrats tended towards "entrepreneurs" in the context of creating jobs. Both are similar in using words like "approve" and "pass" referring to their party proposed legislation targeting job creation.

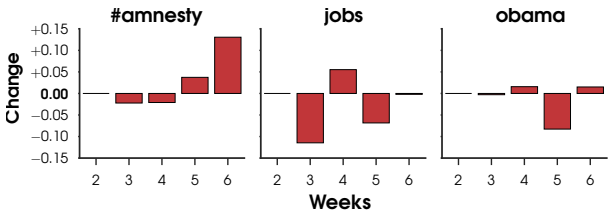


Figure 5.11: Temporal changes for selected Republican word neighbourhoods.

Week 3	*Week 4*	*Week 5*	*Week 6*
create	create	#energy	#jobs
textile	sector	independence	american
#madeinusa	private	lowers	create
kills	created	fewer	#energy
manufacturing	180	kill	project
creating	manufacturing	create	approve
remark	kill	gas	creating
amortization	scientific	lowering	#yes2energy
1k	generated	#jobs	supports
…	…	…	…

Table 5.12: Sample top words from word neighbourhoods for "jobs" in the Republican community, over 4 weekly time windows.

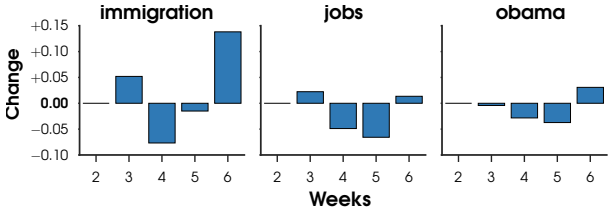


Figure 5.12: Temporal changes for selected Democrat word neighbourhoods.

Week 3	*Week 4*	*Week 5*	*Week 6*
creating	outsourced	000	creating
add	overseas	added	celebrate
manufacturing	paying	214	lets
overseas	created	economy	create
adding	#dayonenc	manufacturing	entrepreneurs
create	1943	adds	would
ship	300k	breaking	thousands
ca52	shipped	news	pass
bringing	rebuilding	214k	#kxl
…	…	…	…

Table 5.13: Sample top words from word neighbourhoods for "jobs" in the Democrat community, over 4 weekly time windows.

Overall, the majority of tweets by campaign accounts rarely discussed political issues on Twitter, instead the platform was utilised for general announcements and advertising positive feedback from supporters.

The campaign accounts of both Democrats and Republicans on Twitter steered away from contentious topics, opting to share generic calls to action or announcement updates about their campaigning activities. Some examples are highlighted below:

```
wow based on the turnout tonight voters are fired up
```

```
looking forward to discussing my work in congress on wnri tune in at
8 am [link]
```

```
this isnt just an election we can win its an election we must win
```

In general, for this case study, we observed that candidates did not directly engage in debates with one another, and typically kept expressions of their political positions to a minimum. This curiosity is perhaps explained by an overly cautious approach to Twitter as a medium of communication for politicians.

5.2.7 Conclusion

In this work, we have proposed an approach to Critical Discourse Analysis that uses distributional semantic models to explore variations between online communities, and over time, at a word level. The approach is applicable to large social media datasets, where frequency-based approaches fail to adequately capture discourse variations between communities and over time. We evaluated our approach using two quite different political case studies, each with distinct communities active on Twitter. These case studies illustrate that analysing discourse communities over short periods of time can highlight interesting dynamics both between and within communities, as they react to external influences that shape their discourse. While we have focused on cases involving two communities on Twitter, the approach naturally generalizes to scenarios involving multiple distinct communities, and longer pieces of text such as party manifestos, news articles, and blog posts.

In general, we suggest that DSMs offer CDA practitioners a useful exploratory tool that can be used in conjunction with existing qualitative and quantitative approaches. The ability of the DSM approach to produce word neighbourhoods with both semantic and syntactic similarities could also be employed in downstream applications, such as estimating party or candidate positions on certain key issues. The effectiveness of using features derived from DSMs for these tasks is currently being investigated.

The approach for comparing nearest neighbours of words in distributional semantic models can be applied in other scenarios, as it is not dependent on the method used to construct it. The top-weighted Jaccard Similarity and word trajectory visualisations are a generally useful and interpretable way to examine "word spaces" in other applications.

Potentially, multi-disciplinary methodologies can benefit from both quantitative and qualitative methods. A purely quantitative approach can be backed by a large body of social science and literary theory, while traditional qualitative approaches to discourse analysis can be guided and supported with quantitative techniques that are familiar in text mining, but remain underutilised by CDA practitioners.

More recently, however, more focus has been placed on applying close reading of social media posts, with guides on Social Media Critical Discourse Studies (SM-CDS) [258] and outlines of key theoretical and methodological aspects in applying DHA to social media data [259]. In [260] DHA is used to examine how U.S. President Donald Trump employs Twitter as a strategic instrument of power.

Our suggested approach has been subsquently adopted and applied to a corpus of White House Press Briefings from 1993 to 2014 [261]. Here Temporal Random Indexing (TRI) was used to construct the DSM, and two case studies focusing on discursive strategies used around *violence* and *terrorism* were examined.

Overall, our work indicates that the DSM-supported process of initial word selection, followed by expansion, and examination of word neighbourhoods can yield support to, or inform hypotheses about mechanisms for wielding power, hierarchies, vested interests and other aspects of critical discourse analysis.

CHAPTER

SIX

CONCLUSIONS

Chapter 2 outlined the importance of selecting *newsworthy* content and filtering. Chapters 3 and 4 addressed problems in detecting and tracking breaking news events. Finally, Chapter 5 examined groups with opposing ideologies. There are three main strategies, common in all chapters, that contributed most to our results:

1. **Ensuring Quality of Input Data:** Taking relevant platform-specific features and limitations into account. Relying on well-curated, human annotated ground truth data derived from real newsrooms.

2. **Flexible Text Representations:** Distributional Semantic Models proposed in Section 4.2 capture both semantic and syntactic similarities, which represent features that are important in analysing discourse in communities.

3. **Optimal Parameter Tuning:** As discussed in Section 3.2, with appropriate parameter settings, even simple approaches can provide an adequate level of effectiveness in decision support systems for journalists.

A prominent feature of working with Twitter datasets is dealing with platform specificity. Dealing with platform-specific features is necessary to perform effective analysis, but can result in approaches not being applicable to other platforms or other types of texts. Twitter-specific solutions may be more generally useful for other Twitter-like platforms such as Gab.ai, Sina Weibo, and Mastodon, but may not be as useful for other platforms such as Youtube or Facebook, where the structure and nature of user-generated content is significantly different.

Twitter is often mentioned in the same context as other sources of short, "noisy" texts, such as SMS and chat messages. However, the interactions of these are very different and are not directly comparable, as the speech act structure differs significantly [262].

Some components are more generally useful. For instance, DSMs representing text can be applied to text from any platform, but tokenizers and POS taggers trained on tweets are highly specific to Twitter and may need additional tuning for before being applied to other sources of text.

Other platform features, such as hashtags, are used in substantially different ways on alternative social media platforms. While techniques and assumptions for studying hashtag use may not be directly transferable, some reuse may be possible. For instance, tokenizers built to handle tweet text can work just as well for Instagram text.

6.1 Recent Advancements

While each article listed in Chapter 1.9 is self-contained with related work, the following section lists significant developments relevant to presented research which have been subsequently published:

Filtering Newsworthy Sources:
In the datasets used in our experiments, we relied heavily on human-curated sources of ground truth. While the system in Section 2.1 recommends newsworthy accounts, journalists manually verify sources and construct lists. Automatically adding and removing sources depending on the type of monitoring required is still an active area of research. The 2014 RepLab [263] competitive evaluation campaign for online reputation management systems focused on the problems of reputation dimensions classification and author profiling. While not specific to the news domain, the outcomes of the *Author Categorisation* and *Author Ranking* tasks revealed that a solution to these problems is feasible, but it is unclear as to which techniques and features are most effective.

Retrospective Event Detection:
Journalists, political scientists, and historians frequently examine large quantities of text from archives. For retrospective event detection, we proposed exploring attention dominating moments in Section 4.3 using a content diversity measure examining news, blogs and tweets. A *significant event detection* approach based on topic models is introduced in [129]. Using a corpus of U.S. State Department cables from the 1970s, the approach discovered both well-known, and obscure but significant events.

Combining Tweets and News Articles:
Approaches for dealing with tweets and headlines from Section 3.1 were adapted for a recommendation system for streaming news in [264] and [265]. More up-to-date models for Tweet specific feature extraction [266] have been developed in 2015 shared tasks on noisy user-generated text [267].

Timeline Summarization:
In Section 4.2 we explored multi-document summarization in a stream of news updates, for the purpose of generating event timelines. Since publication, Twitter has opened up *collections* to the public via TweetDeck[1], but these have not been widely adopted by journalists for curating breaking news events. Instead, a more streamlined *Moments* feature was introduced in 2015. *Moments* are essentially collections of tweets, with an emphasis on embedded media content. A promising alternative to adapting representations of text in streams is a Burst Information Network (BINet) [268]. Here text in a stream of news updates is represented as nodes and edges, where a node is a word with a timespan, and edges between nodes indicate relationships between words.

Scalability:
In Section 4.1 we explore the problem of dealing with streaming data. Storm [269], a real-time fault-tolerant distributed stream data processing system became an Apache top level project in September 2014. Major improvements have been made to the system for performance and reliability, as well as changes to the API. The most useful addition for online learning applications is the introduction of a native windowing API[2]. These improvements simplify the development of sliding window approaches common in online learning settings, such as those in Sections 4.2 and 3.2. Since publication, other frameworks, such as Heron [270] and MLLib [271], have been introduced. The stream processing landscape is extremely active and has changed significantly. Frameworks have matured and now offer more stable platforms to build on. However, increased availability of faster machines with more resources means that processing data streams need not involve distributed systems—even relatively large datasets can be efficiently processed on single machines, simplifying systems and reducing complexity.

News and Discourse Communities:
In Section 5.2 we explored elements of the Discourse-Historical Approach in CDA[3] applied to social media data. A recent discussion of key theoretical and methodological aspects in discourse studies is presented in [259]. The book chapter highlights the importance of sub-sampling social media data, media and genre-specific contexts, and the changes in format and functions of data over time. A further useful development is a collection of *discursive news values* [272], which have been proposed to support the examination of how news discourse is constructed. Automating elements of this framework is a potentially interesting area for future work.

[1] `https://tweetdeck.twitter.com/`
[2] `http://storm.apache.org/2016/04/12/storm100-released.html`
[3] Critical Discourse Analysis (CDA) is more recently called Critical Discourse Studies (CDS)

In another recent study comparing the language use of two ideologically opposing groups [273], entropy-based measures were used. The two discourse communities were Black Lives Matter movement [274] supporters and supporters of the "*#AllLivesMatter*" counter-protest.

Incorporating Parliamentary Activity:
Visualising high-dimensional spaces for exploratory analysis is a difficult problem with many different approaches. Our work in Section 5.1 proposes dimensionality reduction methods for visualising groups of politicians, based on their voting records. An alternative formulation for the problem treats politicians and votes as a network, with nodes and edges representing politicians and voting patterns [275]. Evaluating the quality and usefulness of such visualisations remains a difficult task, as there is no standard collection or agreement on appropriate measures of association between political figures.

Event Detection Evaluation:
We explored evaluation issues when detecting breaking news events in Section 3.2. One drawback with our evaluation is its reliance on a collection of documents and ground truth set of events that will not be updated, as the Reportedly project has ceased.

As the collection gets older, the risk of errors from information "from the future" is likely to increase. For example, a Named Entity Extraction system trained on a newer corpus may already include people or organisations that would have been unknown at the time when the events in the evaluation were occurring. As a result, improvements in associated detection results may be misleading.

In event detection tasks, it is beneficial to evaluate systems in real-time, but this is often not possible. EMBERS (Early Model Based Event Recognition using Surrogates) project [276] is part of the Advanced Research Projects Activity (IARPA) Open Source Indicators (OSI) program. The system monitors news, blogs, tweets, machine coded events, currency rates, food prices and implements a robust event detection evaluation in real-time, scored independently of the system authors. The key disadvantage of this evaluation approach is the prohibitive cost in maintaining the human-annotated ground truth from an external party.

With these new developments in mind, potential new applications and extensions to existing work are discussed in 6.2 below.

6.2 Future Work

For journalists, social media has become more than just a means of disseminating reports. Journalists are spending more time sourcing user-generated content and eyewitness accounts of news events. Verification of user-generated content has become a central part of social newsgathering, presenting both significant challenges and possibilities for new systems.

Following the 2016 U.S. Elections, interest in verification and fact-checking increased significantly. In the media, the trend is broadly referred to as the *"fake news"* problem. Different kinds of content tends to be grouped under this label ranging from satire, advertising masquerading as news, poor quality or provocative "clickbait", to highly partisan opinions. Different content types require different solutions. As highlighted in *Filtering* (1.4), verification is a challenging application area with many potential applications and diverse requirements [277].

Drawing on the output of a real newsroom, and soliciting relevancy judgements from journalists, has revealed gaps in what current retrieval systems optimise for, and what journalists prefer to see. Qualitative evaluations of results also demonstrated that an annotator's prior knowledge of a topic or event can influence relevancy judgements.

Our results also highlight the need for a more personalised filtering of duplicate information. Approaches to personalised news *filtering and summarization* [278] exist, but equivalents for *event detection* have not been widely adopted by journalists.

Recently, there has been a renewed interest in reproducibility, with dedicated efforts to maintain reproducible baseline systems [279]. Unfortunately, several issues make this difficult with the kinds of corpora involved in breaking news. Effective archiving during important events is hampered by platform restrictions, and the ephemeral nature of tweets can make standard collections decay over time.

It is estimated that about 11% of content shared is lost within one year [280]. The work presented in this thesis is no exception. Data becomes unavailable due to platforms removing users for violating their terms of service, or users themselves removing their content. Accounting for, and dealing with this decay is still an open problem.

Finally, the notion of *newsworthiness* remains surprisingly vague in the fields of Information Retrieval and Machine Learning. Emphasis is often placed on techniques and algorithms, as opposed to the task definition and implications.

Questions addressing what should and should not be considered as news are rarely explored in detail.

Formalising *newsworthiness* by defining linguistic devices used in constructing news, such as those proposed in *Discursive News Values Analysis* [281], presents numerous application areas and new tasks that can potentially be supported by techniques from information retrieval and natural language processing.

BIBLIOGRAPHY

[1] Bruns, A. *Gatewatching: Collaborative online news production*, volume 26. Peter Lang, 2005. (Cited on page 1)

[2] Bruns, A. Real-Time Applications (Twitter). In H. Friese, G. Rebane, M. Nolden, and M. Schreiter, editors, *Handbuch Soziale Praktiken und Digitale Alltagswelten*, pages 1–9. Springer Fachmedien Wiesbaden, Wiesbaden, 2016. ISBN 978-3-658-08460-8. doi:10.1007/978-3-658-08460-8_8-1. (Cited on page 1)

[3] Boyd, D. *Twitter: "pointless babble" or peripheral awareness + social grooming?*, 2009. Available at `http://www.zephoria.org/thoughts/archives/2009/08/16/twitter{_}pointle.html`. (Cited on page 1, 5)

[4] Lasorsa, D. L., Lewis, S. C., and Holton, A. E. Normalizing Twitter: Journalism practice in an emerging communication space. *Journalism studies*, 13(1):19–36, 2012. (Cited on page 1)

[5] Kamps, H. J. *Who Are Twitter's Verified Users?*, May 2015. Available at `https://medium.com/@Haje/who-are-twitter-s-verified-users-af976fc1b032`. (Cited on page 1)

[6] Hermida, A., Lewis, S. C., and Zamith, R. Sourcing the arab spring: a case study of Andy Carvin's sources on twitter during the Tunisian and Egyptian revolutions. *Journal of Computer-Mediated Communication*, 19(3):479–499, 2014. (Cited on page 2, 14, 27)

[7] Aday, S., Farrell, H., Lynch, M., Sides, J., Kelly, J., and Zuckerman, E. Blogs and bullets: New media in contentious politics. *United States Institute of Peace*, (65), 2010. (Cited on page 2)

[8] Tufekci, Z. and Freelon, D. Introduction to the special issue on new media and social unrest. *American Behavioral Scientist*, page 0002764213479376, 2013. (Cited on page 2)

[9] Allan, J. *Topic Detection and Tracking: Event-based Information Organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 0-7923-7664-1. (Cited on page 3, 11, 22, 37, 38, 55, 75)

[10] Ounis, I., Macdonald, C., Lin, J., and Soboroff, I. Overview of the TREC-2011 Microblog Track. In *In Proceedings of TREC 2011*. 2011. (Cited on page 3)

[11] Weller, K., Bruns, A., Burgess, J., Mahrt, M., and Puschmann, C. *Twitter and Society*. Peter Lang. ISBN 978-1-4541-9992-2. (Cited on page 4)

[12] Kwak, H., Lee, C., Park, H., and Moon, S. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600. ACM, New York, NY, USA, 2010. ISBN 978-1-60558-799-8. doi:10.1145/1772690.1772751. (Cited on page 4, 5)

[13] Gottfried, J. and Shearer, E. News Use Across Social Media Platforms 2016. Survey, Pew Research Center, May 2016. Available: `http://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/`. (Cited on page 5)

[14] Rosenstiel, T., Sonderman, J., Loker, K., Ivancin, M., and Kjarval, N. Twitter and the News: How people use the social network to learn about the world. Technical report, American Press Institute, January 2015. Available `https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-news/`. (Cited on page 5)

[15] Newman, M. E. and Park, J. Why social networks are different from other types of networks. *Physical Review E*, 68(3):036122, 2003. (Cited on page 5)

[16] Huberman, B., Romero, D. M., and Wu, F. Social networks that matter: Twitter under the microscope. *First Monday*, 14(1), 2008. (Cited on page 5)

[17] Thomas, K., Grier, C., Song, D., and Paxson, V. Suspended Accounts in Retrospect: An Analysis of Twitter Spam. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 243–258. ACM, New York, NY, USA, 2011. ISBN 978-1-4503-1013-0. doi:10.1145/2068816.2068840. (Cited on page 5)

[18] Galán-GarcÍa, P., De La Puerta, J. G., Gómez, C. L., Santos, I., and Bringas, P. G. Supervised machine learning for the detection of troll profiles in twitter social network: application to a real case of cyberbullying. *Logic Journal of IGPL*, 24(1):42–53, 2016. doi:10.1093/jigpal/jzv048. (Cited on page 5)

[19] Gupta, A., Lamba, H., Kumaraguru, P., and Joshi, A. Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on World Wide Web*, pages 729–736. ACM, 2013. (Cited on page 5)

[20] Gayo-Avello, D. "I Wanted to Predict Elections with Twitter and all I got was this Lousy Paper" - A Balanced Survey on Election Prediction using Twitter Data. *CoRR*, abs/1204.6441, 2012. (Cited on page 6)

[21] Liang, H. and Fu, K.-w. Testing Propositions Derived from Twitter Studies: Generalization and Replication in Computational Social Science. *PLOS ONE*, 10(8):1–14, August 2015. doi:10.1371/journal.pone.0134270. (Cited on page 6)

[22] Liu, Y., Kliman-Silver, C., and Mislove, A. The Tweets They Are a-Changin: Evolution of Twitter Users and Behavior. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* 2014. (Cited on page 6)

[23] Garber, M. Fort Hood: A First Test of Twitter Lists. *Columbia Journalism Review online. Web*, 2, 2009. (Cited on page 7)

[24] Boyd, D., Golder, S., and Lotan, G. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *43rd Hawaii International Conference on System Sciences (HICSS)*, pages 1–10. IEEE, 2010. (Cited on page 7)

[25] Garimella, K., Weber, I., and De Choudhury, M. Quote RTs on Twitter: Usage of the New Feature for Political Discourse. In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, pages 200–204. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-4208-7. doi:10.1145/2908131.2908170. (Cited on page 7, 66)

[26] Gorrell, G. and Bontcheva, K. Classifying Twitter favorites: Like, bookmark, or Thanks? *Journal of the Association for Information Science and Technology*, 67(1):17–25, 2016. ISSN 2330-1643. doi:10.1002/asi.23352. (Cited on page 7, 8)

[27] Meier, F., Elsweiler, D., and Wilson, M. L. More than Liking and Bookmarking? Towards Understanding Twitter Favouriting Behaviour. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* 2014. (Cited on page 7, 8)

[28] Petrovic, S., Osborne, M., and Lavrenko, V. RT to Win! Predicting Message Propagation in Twitter. In *ICWSM*. 2011. (Cited on page 7)

[29] Finn, S., Mustafaraj, E., and Metaxas, P. T. The Co-retweeted Network and Its Applications for Measuring the Perceived Political Polarization. In *Proceedings of the 10th International Conference on Web Information Systems and Technologies*, pages 276–284. 2014. ISBN 978-989-758-023-9. (Cited on page 7)

[30] Sousa, D., Sarmento, L., and Mendes Rodrigues, E. Characterization of the twitter replies network: are user ties social or topical? In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 63–70. ACM, 2010. (Cited on page 8)

[31] Nizam, N., Watters, C., and Gruzd, A. Link sharing on twitter during popular events: Implications for social navigation on websites. In *2014 47th Hawaii International Conference on System Sciences*, pages 1745–1754. IEEE, 2014. (Cited on page 8)

[32] Zubiaga, A. A longitudinal assessment of the persistence of twitter datasets. *Journal of the Association for Information Science and Technology*, 2018. doi:10.1002/asi.24026. (Cited on page 8)

[33] Lin, J., Efron, M., Wang, Y., and Sherman, G. Overview of the TREC-2014 Microblog Track. Technical report, DTIC Document, 2014. (Cited on page 8, 62)

[34] Harris, Z. S. Distributional structure. *Word*, 10(2-3):146–162, 1954. (Cited on page 12)

[35] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013. (Cited on page 12, 90, 99, 138)

[36] Zhang, Y., Rahman, M. M., Braylan, A., Dang, B., Chang, H., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., McDonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., and Lease, M. Neural Information Retrieval: A Literature Review. *CoRR*, abs/1611.06792, 2016. (Cited on page 12)

[37] Gallant, S. I., Caid, W. R., Carleton, J., Hecht-Nielsen, R., Qing, K. P., and Sudbeck, D. HNC's MatchPlus system. In *ACM SIGIR Forum*, volume 26, pages 34–38. ACM, 1992. (Cited on page 12)

[38] Maaten, L. v. d. and Hinton, G. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. (Cited on page 12)

[39] Nenkova, A., Vanderwende, L., and McKeown, K. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580. ACM, 2006. (Cited on page 13)

[40] Allan, J., Lavrenko, V., and Jin, H. First Story Detection in TDT is Hard. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, CIKM '00, pages 374–381. ACM, New York, NY, USA, 2000. ISBN 1-58113-320-0. doi:10.1145/354756.354843. (Cited on page 13, 16, 17, 57, 62)

[41] Tufekci, Z. Big Questions for Social Media Big Data: Representativeness, Validity and Other Methodological Pitfalls. In *Proceedings of the International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA*. 2014. (Cited on page 13, 146)

[42] Morstatter, F., Pfeffer, J., and Liu, H. When is It Biased?: Assessing the Representativeness of Twitter's Streaming API. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 555–556. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2745-9. doi:10.1145/2567948.2576952. (Cited on page 13)

[43] Morstatter, F., Pfeffer, J., Liu, H., and Carley, K. M. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *Proceedings of the Seventh International Conference on Weblogs and Social Media, ICWSM 2013, Cambridge, Massachusetts, USA, July 8-11, 2013*. 2013. (Cited on page 13)

[44] Zafar, M. B., Bhattacharya, P., Ganguly, N., Gummadi, K. P., and Ghosh, S. Sampling Content from Online Social Networks: Comparing Random vs. Expert Sampling of the Twitter Stream. *TWEB*, 9(3):12, 2015. doi:10.1145/2743023. (Cited on page 14, 26, 86)

[45] Lehmann, J., Castillo, C., Lalmas, M., and Zuckerman, E. Transient News Crowds in Social Media. In *Proc. 7th Int. Conf. on Weblogs and Social Media*. 2013. (Cited on page 14, 89, 121)

[46] Petrovic, S., Osborne, M., McCreadie, R., Macdonald, C., Ounis, I., and Shrimpton, L. Can Twitter Replace Newswire for Breaking News? In *Proc. 7th International Conference on Weblogs and Social Media, ICWSM*. 2013. (Cited on page 14, 37, 113)

[47] Wardle, C., Dubberley, S., and Brown, P. Amateur Footage: A Global Study Of User-generated Content In Tv And Online-news Output. Report 1, Tow Center for Digital Journalism, April 2014. Available `http://towcenter.org/wp-content/uploads/2014/04/80458_Tow-Center-Report-WEB.pdf`. (Cited on page 14)

[48] Allan, S. *Citizen witnessing: Revisioning journalism in times of crisis*. John Wiley & Sons, 2013. (Cited on page 14)

[49] Lauricella, T., Stewart, C. S., and Ovide, S. Twitter hoax sparks swift stock swoon. *The Wall Street Journal*, 23, 2013. (Cited on page 14)

[50] Eisenstein, J. What to do about bad language on the internet. In *Proceedings of NAACL-HLT*, pages 359–369. 2013. (Cited on page 15)

[51] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. 2002. (Cited on page 15)

[52] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 42–47. Association for Computational Linguistics, Stroudsburg, PA, USA, 2011. ISBN 978-1-932432-88-6. (Cited on page 15, 45)

[53] Denny, M. J. and Spirling, A. Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It. *Political Analysis*, 26(2):168–189, 2018. doi:10.1017/pan.2017.44. (Cited on page 15)

[54] Ferro, N. and Silvello, G. The CLEF Monolingual Grid of Points. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5-8, 2016, Proceedings*, pages 16–27. 2016. doi:10.1007/978-3-319-44564-9_2. (Cited on page 15)

[55] Petrović, S., Osborne, M., and Lavrenko, V. Streaming First Story Detection with Application to Twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 181–189. Association for Computational Linguistics, Stroudsburg, PA, USA, 2010. ISBN 1-932432-65-5. (Cited on page 16, 38, 57)

[56] Allan, J., Lavrenko, V., Malin, D., and Swan, R. Detections, Bounds, and Timelines: UMass and TDT-3. In *In Proceedings of Topic Detection and Tracking Workshop (TDT-3*. 2000. (Cited on page 16)

[57] Vuurens, J. B. and de Vries, A. P. First Story Detection Using Multiple Nearest Neighbors. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 845–848. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-4069-4. doi:10.1145/2911451.2914761. (Cited on page 16)

[58] Osborne, M., Petrovic, S., McCreadie, R., Macdonald, C., and Ounis, I. Bieber no more: First story detection using Twitter and Wikipedia. In *SIGIR 2012 Workshop on Time-aware Information Access*. 2012. (Cited on page 16, 76, 113)

[59] Steiner, T., van Hooland, S., and Summers, E. MJ no more: using concurrent wikipedia edit spikes with social network plausibility checks for breaking news detection. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013, Companion Volume*, pages 791–794. 2013. (Cited on page 16, 76, 77, 113)

[60] Myers, S. A. and Leskovec, J. The bursty dynamics of the twitter information network. In *Proceedings of the 23rd international conference on World wide web*, pages 913–924. ACM, 2014. (Cited on page 16)

[61] Vázquez, A., Oliveira, J. a. G., Dezsö, Z., Goh, K.-I., Kondor, I., and Barabási, A.-L. Modeling bursts and heavy tails in human dynamics. *Phys. Rev. E*, 73:036127, March 2006. doi:10.1103/PhysRevE.73.036127. (Cited on page 16)

[62] Weng, J. and Lee, B. Event Detection in Twitter. In *Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011*. 2011. (Cited on page 16, 77)

[63] Guille, A. and Favre, C. Mention-anomaly-based event detection and tracking in twitter. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 375–382. IEEE, 2014. (Cited on page 16, 77)

[64] Weiler, A., Grossniklaus, M., and Scholl, M. H. An evaluation of the run-time and task-based performance of event detection techniques for Twitter. *Inf. Syst.*, 62:207–219, 2016. doi:10.1016/j.is.2016.01.003. (Cited on page 16)

[65] Weiler, A., Grossniklaus, M., and Scholl, M. H. Evaluation measures for event detection techniques on twitter data streams. In *British International Conference on Databases*, pages 108–119. Springer, 2015. (Cited on page 16)

[66] Lamba, H., Malik, M. M., and Pfeffer, J. A Tempest in a Teacup? Analyzing Firestorms on Twitter. In *Proc. International Conference on Advances in Social Networks Analysis and Mining*, pages 17–24. 2015. (Cited on page 17, 76, 113)

[67] Law, E. and Ahn, L. v. Human computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(3):1–121, 2011. (Cited on page 17)

[68] Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. Power to the people: The role of humans in interactive machine learning. *AI Magazine*, 35(4):105–120, 2014. (Cited on page 17)

[69] Lin, J., Efron, M., Wang, Y., Sherman, G., and Voorhees, E. Overview of the TREC-2015 Microblog Track. In *Proceedings of The Twenty-Fourth Text REtrieval Conference, TREC 2015, Gaithersburg, Maryland, USA, November 17-20, 2015*. 2015. (Cited on page 17)

[70] Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. Learning in nonstationary environments: a survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015. (Cited on page 18)

[71] Hamilton, W. L., Leskovec, J., and Jurafsky, D. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501. Association for Computational Linguistics, Berlin, Germany, August 2016. (Cited on page 18)

[72] Hicks, M. H.-R., Dardagan, H., Bagnall, P. M., Spagat, M., and Sloboda, J. A. Casualties in civilians and coalition soldiers from suicide bombings in Iraq, 2003–10: a descriptive study. *The Lancet*, 378(9794):906–914, 2011. (Cited on page 19)

[73] Li, S., Wang, L., Cao, Z., and Li, W. Text-level Discourse Dependency Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35. Association for Computational Linguistics, Baltimore, Maryland, June 2014. (Cited on page 20)

[74] Ji, Y. and Eisenstein, J. Representation Learning for Text-level Discourse Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24. Association for Computational Linguistics, Baltimore, Maryland, June 2014. (Cited on page 20)

[75] Mann, W. C. and Thompson, S. A. Rhetorical Structure Theory: Toward a functional theory of text organisation. *Text - Interdisciplinary Journal for the Study of Discourse*, 3(8):234–281, 1988. (Cited on page 20, 137)

[76] Wodak, R. The Discourse-Historical Approach. In *Methods of critical discourse analysis*, pages 63–94. 2001. (Cited on page 20, 136, 140)

[77] Carta, C. and Wodak, R. Discourse analysis, policy analysis, and the borders of EU identity. *Journal of Language and Politics*, 14(1):1–17, 2015. (Cited on page 21)

[78] Bednarek, M. Corpora and discourse: A three-pronged approach to analyzing linguistic data. In *Proceedings of the 2008 HCSNet Workshop on Designing the Australian National Corpus*. 2009. (Cited on page 21)

[79] Baker, P., Gabrielatos, C., and McEnery, T. Sketching Muslims: A corpus driven analysis of representations around the word "Muslim" in the British press 1998–2009. *Applied Linguistics*, 34(3):255–278, 2013. (Cited on page 21)

[80] Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., and Kalai, A. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *CoRR*, abs/1607.06520, 2016. (Cited on page 21)

[81] Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., and Kalai, A. T. Quantifying and Reducing Stereotypes in Word Embeddings. *CoRR*, abs/1606.06121, 2016. (Cited on page 21)

[82] Panagiotou, N., Katakis, I., and Gunopulos, D. Detecting Events in Online Social Networks: Definitions, Trends and Challenges. In *Solving Large Scale Learning Tasks. Challenges and Algorithms - Essays Dedicated to Katharina Morik on the Occasion of Her 60th Birthday*, pages 42–84. 2016. doi:10.1007/978-3-319-41706-6_2. (Cited on page 22)

[83] Baena-García, M., Carmona-Cejudo, J. M., Castillo, G., and Morales-Bueno, R. TF-SIDF: Term frequency, sketched inverse document frequency. In *11th International Conference on Intelligent Systems Design and Applications (ISDA)*, pages 1044–1049. IEEE, 2011. (Cited on page 22)

[84] Tratz, S. and Hovy, E. H. Summarization Evaluation Using Transformed Basic Elements. In *Proceedings of the First Text Analysis Conference, TAC 2008, Gaithersburg, Maryland, USA, November 17-19, 2008*. 2008. (Cited on page 22, 105)

[85] Wachsmuth, H. Text Analysis Pipelines. In *Text Analysis Pipelines*, pages 19–53. Springer, 2015. (Cited on page 23)

[86] Brigadir, I., Greene, D., and Cunningham, P. A system for twitter user list curation. In *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, pages 293–294. 2012. doi:10.1145/2365952.2366019. (Cited on page 23, 26, 28)

[87] Brigadir, I., Greene, D., Cunningham, P., and Sheridan, G. Real Time Event Monitoring With Trident. In *RealStream: Real-World Challenges for Data Stream Mining workshop at European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013), Prague, September 23th to 27th, 2013*. 2013. (Cited on page 23, 83)

[88] Ifrim, G., Shi, B., and Brigadir, I. Event Detection in Twitter using Aggressive Filtering and Hierarchical Tweet Clustering. In *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014.*, pages 33–40. 2014. (Cited on page 23, 36)

[89] Brigadir, I., Greene, D., and Cunningham, P. Adaptive Representations for Tracking Breaking News on Twitter. *CoRR*, abs/1403.2923, 2014. (Cited on page 24, 81, 90, 116)

[90] Brigadir, I., Greene, D., and Cunningham, P. Analyzing Discourse Communities with Distributional Semantic Models. In *Proceedings of the ACM Web Science Conference, WebSci 2015, Oxford, United Kingdom, June 28 - July 1, 2015*, pages 27:1–27:10. 2015. doi:10.1145/2786451.2786470. (Cited on page 24, 135)

[91] Brigadir, I., Greene, D., and Cunningham, P. Detecting Attention Dominating Moments Across Media Types. In *Proceedings of the First International Workshop on Recent Trends in News Information Retrieval co-located with 38th European Conference on Information Retrieval (ECIR 2016), Padua, Italy, March 20, 2016.*, pages 15–20. 2016. (Cited on page 24, 68, 112)

[92] Brigadir, I., Greene, D., Cross, J. P., and Cunningham, P. Dimensionality Reduction and Visualisation Tools for Voting Records. In *Proceedings of 24th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'16)*. 2016. (Cited on page 24, 124)

[93] Brigadir, I., Cunningham, P., and Greene, D. An Investigation into the Effectiveness of Breaking News Event Detection, 2016. In review: Natural Language Engineering. (Cited on page 24)

[94] Brigadir, I., Greene, D., and Cunningham, P. *Dataset: Analyzing Discourse Communities with Distributional Semantic Models*, May 2015. doi:10.6084/m9.figshare.1430449.v1. Available at `http://dx.doi.org/10.6084/m9.figshare.1430449.v1`. (Cited on page 25)

[95] Brigadir, I. *Detecting Attention Dominating Moments Across Media Types - Tweet Stream*, August 2016. doi:10.6084/m9.figshare.2074105.v5. Available at `http://dx.doi.org/10.6084/m9.figshare.2074105.v5`. (Cited on page 25)

[96] Martinez, M., Kruschwitz, U., Kazai, G., Hopfgartner, F., Corney, D., Campos, R., and Albakour, D. Report on the 1st International Workshop on Recent Trends in News Information Retrieval (NewsIR16). *SIGIR Forum*, 50(1):58–67, June 2016. ISSN 0163-5840. doi:10.1145/2964797.2964807. (Cited on page 25)

[97] Greene, D., Reid, F., Sheridan, G., and Cunningham, P. Supporting the Curation of Twitter User Lists. *CoRR*, abs/1110.1349, 2011. (Cited on page 26, 32)

[98] Schwartz, R., Naaman, M., and Teodoro, R. Editorial Algorithms: Using Social Media to Discover and Report Local News. In *Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM*, pages 407–415. 2015. (Cited on page 27)

[99] Ghosh, S., Sharma, N. K., Benevenuto, F., Ganguly, N., and Gummadi, P. K. Cognos: crowdsourcing search for topic experts in microblogs. In *The 35th International ACM SIGIR conference on research and development in Information Retrieval, SIGIR '12, Portland, OR, USA, August 12-16, 2012*, pages 575–590. 2012. doi:10.1145/2348283.2348361. (Cited on page 27)

[100] Lehmann, J., Castillo, C., Lalmas, M., and Zuckerman, E. Finding news curators in twitter. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 863–870. ACM, 2013. (Cited on page 27)

[101] Diakopoulos, N., De Choudhury, M., and Naaman, M. Finding and Assessing Social Media Information Sources in the Context of Journalism. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2451–2460. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2208409. (Cited on page 27)

[102] Greene, D., Sheridan, G., Smyth, B., and Cunningham, P. Aggregating content and network information to curate twitter user lists. In *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, pages 29–36. ACM, 2012. (Cited on page 30)

[103] Leskovec, J., Backstrom, L., and Kleinberg, J. Meme-tracking and the dynamics of the news cycle. *Proc. 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, page 497, 2009. (Cited on page 31, 92, 121)

[104] Papadopoulos, S., Corney, D., and Aiello, L. M. SNOW 2014 Data Challenge: Assessing the Performance of News Topic Detection Methods in Social Media. In *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014.*, pages 1–8. 2014. (Cited on page 34, 36, 39, 40, 48, 75)

[105] Jordaan, M. Poke me, I'm a journalist: The impact of Facebook and Twitter on newsroom routines and cultures at two South African weeklies. *Ecquid Novi: African Journalism Studies*, 34(1):21–35, 2013. (Cited on page 37)

[106] Schifferes, S., Newman, N., Thurman, N., Corney, D., Goker, A., and Martin, C. Identifying and verifying news through social media: Developing a user-centred tool for professional journalists. *Digital Journalism*, 2014. doi:10.1080/21670811.2014.892747. (Cited on page 37)

[107] Aiello, L. M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Göker, A., Kompatsiaris, I., and Jaimes, A. Sensing Trending Topics in Twitter. *IEEE Transactions on Multimedia*, 15(6):1268–1282, October 2013. ISSN 1520-9210. doi:10.1109/TMM.2013.2265080. (Cited on page 37, 38, 44, 45, 46, 75)

[108] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003. ISSN 1532-4435. (Cited on page 38)

[109] Petrović, S., Osborne, M., and Lavrenko, V. Using paraphrases for improving first story detection in news and Twitter. In *Proc. Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 338–346. 2012. (Cited on page 38, 92)

[110] Martin, C., Corney, D., and Göker, A. Finding Newsworthy Topics on Twitter. *IEEE Computer Society Special Technical Community on Social Networking E-Letter*, 2013. (Cited on page 38)

[111] Matuszka, T., Vinceller, Z., and Laki, S. On a keyword-lifecycle model for real-time event detection in social network data. In *IEEE International Conference on Cognitive Infocommunications*. 2013. (Cited on page 39)

[112] Guzman, J. and Poblete, B. On-line relevant anomaly detection in the Twitter stream: an efficient bursty keyword detection model. In *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, pages 31–39. ACM, 2013. (Cited on page 39)

[113] Muellner, D. fastcluster: Fast Hierarchical, Agglomerative Clustering Routines for R and Python. *Journal of Statistical Software*, 53(9):1–18, May 2013. ISSN 1548-7660. (Cited on page 41, 44)

[114] Bird, S. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*. 2006. (Cited on page 45)

[115] Bird, S., Klein, E., and Loper, E. *Natural language processing with Python*. O'Reilly Media, Inc., 2009. (Cited on page 45)

[116] Li, C., Weng, J., He, Q., Yao, Y., Datta, A., Sun, A., and Lee, B.-S. Twiner: named entity recognition in targeted twitter stream. In *ACM SIGIR*. 2012. (Cited on page 45)

[117] Newman, D., Karimi, S., and Cavedon, L. External evaluation of topic models. In *Australasian Document Computing Symposium (ADCS)*, pages 11–18. 2009. (Cited on page 47)

[118] Martín-Dancausa, C. J. and Göker, A. Real-time Topic Detection with Bursty N-grams. In *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014.*, pages 9–16. 2014. (Cited on page 50)

[119] Burnside, G., Milioris, D., and Jacquet, P. One Day in Twitter: Topic Detection Via Joint Complexity. In *Proceedings of the SNOW 2014 Data Challenge co-located with 23rd International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014.*, pages 41–48. 2014. (Cited on page 50)

[120] Jones, K. S. Automatic language and information processing: rethinking evaluation. *Natural Language Engineering*, 7(01):29–46, 2001. (Cited on page 54)

[121] Stray, J. What do Journalists do with Documents? Field Notes for Natural Language Processing Researchers. In *Computation Journalism Symposium*. 2016. (Cited on page 55)

[122] Khare, P., Torres, P., and Heravi, B. R. What just happened? A Framework for Social Event Detection and Contextualisation. In *International Conference on System Sciences (HICSS)*, pages 1565–1574. IEEE, 2015. (Cited on page 55)

[123] Tan, L., Lin, J. J., Roegiest, A., and Clarke, C. L. A. The Effects of Latency Penalties in Evaluating Push Notification Systems. *CoRR*, abs/1606.03066, 2016. (Cited on page 55)

[124] Lin, J. and Efron, M. Evaluation As a Service for Information Retrieval. *SIGIR Forum*, 47(2):8–14, January 2013. ISSN 0163-5840. doi:10.1145/2568388.2568390. (Cited on page 55)

[125] Orellana-Rodriguez, C., Greene, D., and Keane, M. T. Spreading the News: How Can Journalists Gain More Engagement for Their Tweets? In *Proc. ACM Web Science 2016*. 2016. (Cited on page 56)

[126] Lanagan, J. and Smeaton, A. F. Using Twitter to Detect and Tag Important Events in Sports Media. In *Fifth International AAAI Conference on Weblogs and Social Media*. 2011. (Cited on page 56)

[127] McMinn, A. J., Moshfeghi, Y., and Jose, J. M. Building a Large-scale Corpus for Evaluating Event Detection on Twitter. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 409–418. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505695. (Cited on page 57, 75)

[128] Fiscus, J. G. and Doddington, G. R. Topic Detection and Tracking. In J. Allan, editor, *Topic Detection and Tracking*, chapter Topic Detection and Tracking Evaluation Overview, pages 17–31. Kluwer Academic Publishers, Norwell, MA, USA, 2002. ISBN 0-7923-7664-1. (Cited on page 57)

[129] Chaney, A. J., Wallach, H. M., Connelly, M., and Blei, D. M. Detecting and Characterizing Events. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1142–1152. 2016. (Cited on page 57, 159)

[130] Ghosh, S., Zafar, M. B., Bhattacharya, P., Sharma, N., Ganguly, N., and Gummadi, K. On Sampling the Wisdom of Crowds: Random vs. Expert Sampling of the Twitter Stream. In *Proceedings of the International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 1739–1744. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2263-8. doi:10.1145/2505515.2505615. (Cited on page 59, 116, 146)

[131] Martinez-Alvarez, M., Kruschwitz, U., Kazai, G., Hopfgartner, F., Corney, D., Campos, R., and Albakour, D. *First International Workshop on Recent Trends in News Information Retrieval (NewsIR'16)*, pages 878–882. Springer International Publishing, Cham, 2016. ISBN 978-3-319-30671-1. doi:10.1007/978-3-319-30671-1_85. (Cited on page 61)

[132] Atefeh, F. and Khreich, W. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015. (Cited on page 62, 76)

[133] Carterette, B. A. Multiple Testing in Statistical Analysis of Systems-based Information Retrieval Experiments. *ACM Transactions on Information Systems*, 30(1):4:1–4:34, March 2012. ISSN 1046-8188. doi:10.1145/2094072.2094076. (Cited on page 65, 75)

[134] Fung, G. P. C., Yu, J. X., Yu, P. S., and Lu, H. 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. ISBN 1-59593-154-6. (Cited on page 66)

[135] Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. CrisisLex: A Lexicon for Collecting and Filtering Microblogged Communications in Crises. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* 2014. (Cited on page 67)

[136] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119. 2013. (Cited on page 68, 96, 98, 99, 142)

[137] Lavin, A. and Ahmad, S. Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 38–44. December 2015. doi: 10.1109/ICMLA.2015.141. (Cited on page 68)

[138] Hole, K. J. Anomaly Detection with HTM. In *Anti-fragile ICT Systems*, pages 125–132. Springer, 2016. (Cited on page 68)

[139] Schneider, M., Ertel, W., and Ramos, F. T. Expected Similarity Estimation for Large-Scale Batch and Streaming Anomaly Detection. *CoRR*, abs/1601.06602, 2016. (Cited on page 68)

[140] Robertson, S. E. and Kanoulas, E. On Per-topic Variance in IR Evaluation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 891–900. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1472-5. doi:10.1145/2348283.2348402. (Cited on page 73, 75)

[141] Sanderson, M. *Test collection based evaluation of information retrieval systems*. Now Publishers Inc, 2010. (Cited on page 75)

[142] Kelly, D. and Sugimoto, C. R. A systematic review of interactive information retrieval evaluation studies, 1967–2006. *Journal of the American Society for Information Science and Technology*, 64(4):745–770, 2013. (Cited on page 75)

[143] Armstrong, T. G., Moffat, A., Webber, W., and Zobel, J. Improvements That Don'T Add Up: Ad-hoc Retrieval Results Since 1998. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 601–610. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646031. (Cited on page 75)

[144] Fang, H., Tao, T., and Zhai, C. Diagnostic evaluation of information retrieval models. *ACM Transactions on Information Systems (TOIS)*, 29(2):7, 2011. (Cited on page 75)

[145] Boytsov, L., Belova, A., and Westfall, P. Deciding on an adjustment for multiplicity in IR experiments. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 403–412. ACM, 2013. (Cited on page 75)

[146] Ferro, N. and Silvello, G. A General Linear Mixed Models Approach to Study System Component Effects. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, pages 25–34. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-4069-4. doi:10.1145/2911451.2911530. (Cited on page 75)

[147] Tran, G. B. and Alrifai, M. Indexing and Analyzing Wikipedia's Current Events Portal, the Daily News Summaries by the Crowd. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pages 511–516. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2745-9. doi:10.1145/2567948.2576942. (Cited on page 76)

[148] Shilliday, A. and Lautenschlager, J. Data for a Worldwide ICEWS and Ongoing Research. *Advances in Design for Cross-Cultural Activities*, page 455, 2012. (Cited on page 76, 78)

[149] Leetaru, K. and A. Schrodt, P. GDELT: Global data on events, location, and tone. *ISA Annual Convention*, 2013. (Cited on page 76, 78)

[150] Boydstun, A. E. *Making the news: Politics, the media, and agenda setting*. University of Chicago Press, 2013. (Cited on page 76, 113)

[151] Gandica, Y., Carvalho, J., dos Aidos, F. S., Lambiotte, R., and Carletti, T. On the origin of burstiness in human behavior: The wikipedia edits case, 2016. (Cited on page 76, 114)

[152] Jungherr, A. and Pascal, J. Forecasting the pulse: how deviations from regular patterns in online data can identify offline phenomena. *Internet Research*, 23(5):589–607, 2013. (Cited on page 76, 114)

[153] Cordeiro, M. and Gama, J. Online Social Networks Event Detection: A Survey. In *Solving Large Scale Learning Tasks. Challenges and Algorithms*, pages 1–41. Springer, 2016. (Cited on page 76)

[154] Weiler, A., Grossniklaus, M., and Scholl, M. H. Editorial: Survey and Experimental Analysis of Event Detection Techniques for Twitter. *The Computer Journal*, 2016. (Cited on page 76)

[155] Imran, M., Castillo, C., Diaz, F., and Vieweg, S. Processing Social Media Messages in Mass Emergency: A Survey. *CoRR*, abs/1407.7071, 2014. (Cited on page 76)

[156] Weiler, A., Grossniklaus, M., and Scholl, M. H. *Run-Time and Task-Based Performance of Event Detection Techniques for Twitter*, pages 35–49. Springer International Publishing, Cham, 2015. ISBN 978-3-319-19069-3. doi:10.1007/978-3-319-19069-3_3. (Cited on page 76)

[157] Uysal, A. K. and Gunal, S. The Impact of Preprocessing on Text Classification. *Inf. Process. Manage.*, 50(1):104–112, January 2014. ISSN 0306-4573. doi:10.1016/j.ipm.2013.08.006. (Cited on page 77)

[158] Pomikalek, J. and Rehurek, R. The Influence of Preprocessing Parameters on Text Categorization. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 1(9):504–507, 2007. (Cited on page 77)

[159] Thurman, N. Social Media, Surveillance, and News Work. *Digital Journalism*, 6(1):76–97, 2018. doi:10.1080/21670811.2017.1345318. (Cited on page 77)

[160] Sprugnoli, R. and Tonelli, S. One, no one and one hundred thousand events: Defining and processing events in an inter-disciplinary perspective. *Natural Language Engineering*, pages 1–22, October 2016. doi:10.1017/S1351324916000292. (Cited on page 78)

[161] Piskorski, J. and Yangarber, R. *Information Extraction: Past, Present and Future*, pages 23–49. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-28569-1. doi:10.1007/978-3-642-28569-1_2. (Cited on page 78)

[162] Beieler, J. Generating Politically-Relevant Event Data. *ArXiv e-prints*, 2016. (Cited on page 78)

[163] Schrodt, P., Beieler, and Idris, M. Three'sa Charm?: Open Event Data Coding with EL: DIABLO, PETRARCH, and the Open Event Data Alliance. In *Proceedings of the Internvarianational Studies Association meetings*. 2014. (Cited on page 78)

[164] Ritter, A., Etzioni, O., Clark, S., et al. Open Domain Event Extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM, 2012. (Cited on page 78)

[165] Kunneman, F. and Van Den Bosch, A. Open-domain extraction of future events from Twitter. *Natural Language Engineering*, 22(5):655–686, September 2016. (Cited on page 78)

[166] Hartigan, J. A. *Clustering Algorithms*. John Wiley & Sons, Inc., 1975. (Cited on page 83, 84)

[167] Marz, N. Storm: Distributed and fault-tolerant realtime computation, 2012. (Cited on page 84)

[168] Berendsen, R., Tsagkias, M., Weerkamp, W., and de Rijke, M. Pseudo Test Collections for Training and Tuning Microblog Rankers. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 53–62. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-2034-4. doi:10.1145/2484028.2484063. (Cited on page 86)

[169] Järvelin, K. and Kekäläinen, J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. (Cited on page 87)

[170] Chong, F. and Chua, T. Automatic Summarization of Events From Social Media. In *Proc. 7th International AAAI Conference on Weblogs and Social Media (ICWSM'13)*. 2013. (Cited on page 89, 92)

[171] Kanerva, P., Kristoferson, J., and Holst, A. Random Indexing of Text Samples for Latent Semantic Analysis. In *In Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 103–6. Erlbaum, 2000. (Cited on page 90)

[172] Wang, T. Time-dependent Hierarchical Dirichlet Model for Timeline Generation. *arXiv preprint arXiv:1312.2244*, 2013. (Cited on page 92)

[173] Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., and Zhang, Y. Evolutionary Timeline Summarization : a Balanced Optimization Framework via Iterative Substitution. In *Proc. 34th SIGIR Conference on Research and development in Information Retrieval*, pages 745–754. 2011. ISBN 9781450307574. (Cited on page 92)

[174] Li, J. and Cardie, C. Timeline Generation : Tracking individuals on Twitter. *arXiv preprint arXiv:1309.7313*, 2013. (Cited on page 92)

[175] Shou, L. Sumblr: Continuous Summarization of Evolving Tweet Streams. In *Proc. 36th SIGIR conference on Research and Development in Information Retrieval*, pages 533–542. 2013. ISBN 9781450320344. (Cited on page 92)

[176] Jurgens, D. and Stevens, K. Event Detection in Blogs Using Temporal Random Indexing. In *Proceedings of the Workshop on Events in Emerging Text Types*, eETTs '09, pages 9–16. Association for Computational Linguistics, Stroudsburg, PA, USA, 2009. ISBN 978-954-452-011-3. (Cited on page 92)

[177] Wang, Y. and Lin, J. The Impact of Future Term Statistics in Real-Time Tweet Search. In M. de Rijke, T. Kenter, A. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, editors, *Advances in Information Retrieval*, volume 8416 of *Lecture Notes in Computer Science*, pages 567–572. Springer International Publishing, 2014. ISBN 978-3-319-06027-9. doi:10.1007/978-3-319-06028-6_58. (Cited on page 95)

[178] Vanderwende, L., Suzuki, H., Brockett, C., and Nenkova, A. Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing & Management*, 43(6):1606–1618, 2007. (Cited on page 97, 103)

[179] Inouye, D. and Kalita, J. K. Comparing Twitter Summarization Algorithms For Multiple Post Summaries. In *Privacy, Security, Risk And Trust (passat), 2011 Ieee Third International Conference On And 2011 IEEE Third International Conference On Social Computing (socialcom)*, pages 298–306. IEEE, 2011. (Cited on page 97, 103)

[180] Jones, K. S., Walker, S., and Robertson, S. E. A Probabilistic Model of Information Retrieval: Development and Comparative Experiments. *Inf. Process. Manage.*, 36(6):779–808, November 2000. ISSN 0306-4573. doi:10.1016/S0306-4573(00)00015-7. (Cited on page 97)

[181] Ferguson, P., OHare, N., Lanagan, J., Smeaton, A. F., Phelan, O., McCarthy, K., and Smyth, B. CLARITY at the TREC 2011 Microblog Track. *Proceeddings of the 20th Text REtrieval Conference (TREC 2011)*, 2011. (Cited on page 97)

[182] Sahlgren, M. An introduction to random indexing. In *Semantic Indexing Methods and applications workshop, TKE*, volume 5. 2005. (Cited on page 100)

[183] Kanerva, P. *Sparse distributed memory*. MIT press, 1988. (Cited on page 100)

[184] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990. (Cited on page 100)

[185] Chatterjee, N. and Sahoo, P. K. Effect of Near-orthogonality on Random Indexing Based Extractive Text Summarization. *International Journal of Innovation and Applied Studies*, 3(3):701–713, 2013. ISSN 2028-9324. (Cited on page 100)

[186] Frankl, P. and Maehara, H. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988. (Cited on page 100)

[187] Cohen, T., Schvaneveldt, R., and Widdows, D. Reflective Random Indexing and Indirect Inference: A Scalable Method for Discovery of Implicit Connections. *Journal of Biomedical Informatics*, 43(2):240–256, April 2010. ISSN 1532-0464. doi: 10.1016/j.jbi.2009.09.003. (Cited on page 101)

[188] Lin, C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries. In S. S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81. Association for Computational Linguistics, Barcelona, Spain, July 2004. (Cited on page 103)

[189] Owczarzak, K., Conroy, J. M., Dang, H. T., and Nenkova, A. An Assessment of the Accuracy of Automatic Evaluation in Summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9. Association for Computational Linguistics, Stroudsburg, PA, USA, 2012. (Cited on page 104)

[190] Nenkova, A. and Passonneau, R. J. Evaluating Content Selection in Summarization: The Pyramid Method. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, pages 145–152. 2004. (Cited on page 109, 110)

[191] Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011. (Cited on page 112, 113)

[192] Hu, Y., Talamadupula, K., and Kambhampati, S. *Dude, srsly?: The surprisingly formal nature of Twitter's language*, pages 244–253. AAAI press, 2013. (Cited on page 112)

[193] Zhai, K. and Boyd-Graber, J. Online Latent Dirichlet Allocation with Infinite Vocabulary. In *Proc. 30th International Conference on Machine Learning*, pages 561–569. 2013. (Cited on page 113)

[194] Gao, W., Li, P., and Darwish, K. Joint topic modeling for event summarization across news and social media streams. In *Proc. 21st ACM international conference on Information and knowledge management*, pages 1173–1182. ACM, 2012. (Cited on page 113, 116)

[195] Hu, Y., John, A., Wang, F., and Kambhampati, S. ET-LDA: Joint Topic Modeling for Aligning Events and their Twitter Feedback. In *AAAI Conference on Artificial Intelligence*. 2012. (Cited on page 113)

[196] Hua, T., Chen, F., Lu, C.-T., and Ramakrishnan, N. Topical analysis of interactions between news and social media. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016. (Cited on page 113, 116, 122)

[197] Vaca, C. K., Mantrach, A., Jaimes, A., and Saerens, M. A time-based collective factorization for topic discovery and monitoring in news. In *Proceedings of the 23rd international conference on World wide web*, pages 527–538. ACM, 2014. (Cited on page 113)

[198] Esling, P. and Agon, C. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012. (Cited on page 114, 116)

[199] Liu, S., Yamada, M., Collier, N., and Sugiyama, M. Change-Point Detection in Time-Series Data by Relative Density-Ratio Estimation. *ArXiv e-prints*, March 2012. (Cited on page 114, 116)

[200] O'Callaghan, D., Greene, D., Carthy, J., and Cunningham, P. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657, 2015. ISSN 0957-4174. doi:http://dx.doi.org/10.1016/j.eswa.2015.02.055. (Cited on page 114, 115)

[201] Boutsidis, C. and Gallopoulos, E. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4), 2008. (Cited on page 115)

[202] Giulio Sabbati, E.-M. P. and Saliba, S. Asylum in the EU: Facts and Figures. *European Parliamentary Research Service*, (PE 551.332), March 2015. (Cited on page 118)

[203] Purcell, K., Rainie, L., Mitchell, A., Rosenstiel, T., and Olmstead, K. Understanding the participatory news consumer. *Pew Internet and American Life Project*, 1:19–21, 2010. (Cited on page 121)

[204] Gil de Zúñiga, H., Jung, N., and Valenzuela, S. Social media use for news and individuals' social capital, civic engagement and political participation. *Journal of Computer-Mediated Communication*, 17(3):319–336, 2012. (Cited on page 121)

[205] Berger, J. Arousal Increases Social Transmission of Information. *Psychological Science*, 22(7):891–893, 2011. doi:10.1177/0956797611413294. PMID: 21690315. (Cited on page 121)

[206] Kalyanam, J., Mantrach, A., Saez-Trumper, D., Vahabi, H., and Lanckriet, G. Leveraging Social Context for Modeling Topic Evolution. In *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 517–526. 2015. (Cited on page 122)

[207] McNair, B. *An introduction to political communication*. Routledge, 2017. (Cited on page 123)

[208] Joshi, D. and Rosenfield, E. MP transparency, communication links and social media: A comparative assessment of 184 parliamentary websites. *The Journal of Legislative Studies*, 19(4):526–545, 2013. (Cited on page 123)

[209] Attina, F. The voting behaviour of the European Parliament members and the problem of the Europarties. *European Journal of Political Research*, 18(5):557–579, 1990. (Cited on page 124)

[210] Hix, S., Noury, A., and Roland, G. Voting Patterns and Alliance Formation in the European Parliament. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1518):821–831, 2009. ISSN 0962-8436. (Cited on page 124, 126, 127, 129)

[211] Hix, S., Noury, A., and Roland, G. Dimensions of Politics in the European Parliament. *American Journal of Political Science*, 50(2):494–511, 2006. (Cited on page 124, 125, 130)

[212] Proksch, S.-O. and Slapin, J. B. Position Taking in European Parliament Speeches. *British Journal of Political Science*, 40(03):587–611, 2010. (Cited on page 124, 125)

[213] Braun, D., Mikhaylov, S., and Schmitt, H. European Parliament Election Study 2009, Euromanifesto Study, 2010. doi:10.4232/1.10204. (Cited on page 124, 134)

[214] McElroy, G. and Benoit, K. Policy positioning in the European Parliament. *European Union Politics*, 13(1):150–167, 2012. (Cited on page 124, 125, 134)

[215] Barbera, P. Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data. *Political Analysis*, 23(1):76–91, 2014. doi:10.1093/pan/mpu011. (Cited on page 124)

[216] Poole, K. T. and Rosenthal, H. *Congress: A Political-Economic History of Roll Call Voting*. Oxford University Press, 2000. (Cited on page 125)

[217] Gabel, M. and Hix, S. From Preferences to Behaviour: Comparing MEPs Survey Response and Roll-Call Voting Behaviour. In *Tenth Biennial Conference of the European Union Studies Association*. Citeseer, 2007. (Cited on page 125)

[218] Lowe, W. There's (Basically) Only One Way to Do it. *Available at SSRN 2318543*, 2013. (Cited on page 125)

[219] Lo, J., Proksch, S.-O., and Slapin, J. B. Ideological Clarity in Multiparty Competition: A New Measure and Test Using Election Manifestos. *British Journal of Political Science*, FirstView:1–20, December 2014. ISSN 1469-2112. doi:10.1017/S0007123414000192. (Cited on page 125)

[220] Laver, M., Benoit, K., and Garry, J. Extracting Policy Positions from Political Texts Using Words as Data. *The American Political Science Review*, 97(2):311–331, 2003. (Cited on page 125)

[221] Gu, Y., Sun, Y., Jiang, N., Wang, B., and Chen, T. Topic-factorized Ideal Point Estimation Model for Legislative Voting Network. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 183–192. ACM, 2014. ISBN 978-1-4503-2956-9. (Cited on page 125)

[222] Poole, K., Lewis, J., Lo, J., and Carroll, R. Scaling Roll Call Votes with wnominate in R. *Journal of Statistical Software*, 42:1–21, 2011. (Cited on page 125, 127)

[223] Evans, A. M. and Vink, M. P. Measuring Group Switching in the European Parliament: Methodology, Data and Trends (1979-2009). *Análise social*, pages 92–112, 2012. (Cited on page 127)

[224] Clinton, J., Jackman, S., and Rivers, D. The Statistical Analysis of Roll Call Data. *American Political Science Review*, 98(2):355–370, 2003. (Cited on page 128)

[225] Fodor, I. K. A Survey of Dimension Reduction Techniques. Technical report, Lawrence Livermore National Lab., CA (US), 2002. (Cited on page 128)

[226] Lin, C.-J. Projected Gradient Methods for Nonnegative Matrix Factorization. *Neural Computation*, 19(10):2756–2779, 2007. ISSN 0899-7667. (Cited on page 128)

[227] Ding, C. Nonnegative Matrix Factorizations for Clustering: A Survey. *Data clustering: Algorithms and Applications*, page 148, 2013. (Cited on page 128)

[228] Levy, O. and Goldberg, Y. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2177–2185. 2014. (Cited on page 128, 142)

[229] Desgraupes, B. Clustering Indices. *University of Paris Ouest-Lab Modal'X*, 1:34, 2013. (Cited on page 129)

[230] McInnes, L. and Healy, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018. (Cited on page 134)

[231] Van Dijk, T. A. Critical Discourse Analysis. *The handbook of discourse analysis*, 18:352, 2003. (Cited on page 135)

[232] Porter, J. *Audience and rhetoric: an archaeological composition of the discourse community*. Prentice Hall studies in writing and culture. Prentice Hall, 1992. ISBN 9780130506672. (Cited on page 135, 145)

[233] Firth, J. A Synopsis of Linguistic Theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford, 1957. Reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow. (Cited on page 136)

[234] Fairclough, N. *Critical Discourse Analysis: Papers in the Critical Study of Language*. Language in social life series. Longman, 1995. ISBN 9780582028845. (Cited on page 136)

[235] Baker, P., Gabrielatos, C., Khosravinik, M., Krzyżanowski, M., McEnery, T., and Wodak, R. A useful methodological synergy? Combining critical discourse analysis and corpus linguistics to examine discourses of refugees and asylum seekers in the UK press. *Discourse & Society*, 19(3):273–306, 2008. (Cited on page 137)

[236] Baker, P., McEnery, T., and Gabrielatos, C. Using collocation analysis to reveal the construction of minority groups: The case of refugees, asylum seekers and immigrants in the UK press. In *Corpus Linguistics*. 2007. (Cited on page 137, 139, 143)

[237] Thanopoulos, A., Fakotakis, N., and Kokkinakis, G. Comparative Evaluation of Collocation Extraction Metrics. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002, May 29-31, 2002, Las Palmas, Canary Islands, Spain*. 2002. (Cited on page 137, 139)

[238] Zappavigna, M. *Discourse of Twitter and Social Media: How We Use Language to Create Affiliation on the Web*. Bloomsbury Academic, UK, 1st edition, 2012. ISBN 9781441141866. (Cited on page 137, 140, 152, 154)

[239] Lietz, H., Wagner, C., Bleier, A., and Strohmaier, M. When Politicians Talk: Assessing Online Conversational Practices of Political Parties on Twitter. In *Proceedings of the International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA*. 2014. (Cited on page 137, 145)

[240] Conover, M., Ratkiewicz, J., Francisco, M. R., Gonçalves, B., Menczer, F., and Flammini, A. Political Polarization on Twitter. In *Proceedings of the International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain*. 2011. (Cited on page 137, 142, 150)

[241] Paul, M. J., Zhai, C., and Girju, R. Summarizing Contrastive Viewpoints in Opinionated Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 66–76. Stroudsburg, PA, USA, 2010. (Cited on page 138)

[242] Chen, C., Buntine, W. L., Ding, N., Xie, L., and Du, L. Differential Topic Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):230–242, 2015. doi:10.1109/TPAMI.2014.2313127. (Cited on page 138)

[243] Akoglu, L. Quantifying Political Polarity Based on Bipartite Opinion Networks. In *Proceedings of the International Conference on Weblogs and Social Media, ICWSM, Ann Arbor, Michigan, USA*. 2014. (Cited on page 138)

[244] Baroni, M., Dinu, G., and Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 238–247. 2014. (Cited on page 138, 139)

[245] Juola, P. The time course of language change. *Computers and the Humanities*, 37(1):77–96, 2003. (Cited on page 138)

[246] Kulkarni, V., Al-Rfou, R., Perozzi, B., and Skiena, S. Statistically Significant Detection of Linguistic Change. In *Proceedings of the International Conference on World Wide Web*, WWW '15, pages 625–635. International World Wide Web Conferences Steering Committee, Geneva, Switzerland, 2015. ISBN 978-1-4503-3469-3. doi: 10.1145/2736277.2741627. (Cited on page 138)

[247] Gulordava, K. and Baroni, M. A distributional similarity approach to the detection of semantic change in the Google Books Ngram corpus. In *Proceedings of the GEMS 2011 Workshop on Geometrical Models of Natural Language Semantics, EMNLP '11*, pages 67–71. Association for Computational Linguistics, 2011. (Cited on page 138)

[248] Basile, P., Caputo, A., and Semeraro, G. Analysing Word Meaning over Time by Exploiting Temporal Random Indexing. In *CLIC 2014, The Italian Conference on Computational Linguistics*, pages 38–42. 2014. (Cited on page 139)

[249] Hill, F., Reichart, R., and Korhonen, A. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *ArXiv e-prints*, August 2014. (Cited on page 139)

[250] Anthony, L. AntConc: A Learner and Classroom Friendly, Multi-Platform Corpus Analysis Toolkit. In *Proceedings of IWLeL 2004: An Interactive Workshop on Language e-Learning*. 2005. (Cited on page 142)

[251] Church, K. W. and Hanks, P. Word Association Norms, Mutual Information, and Lexicography. 16(1):22–29. (Cited on page 142)

[252] Sinclair, J. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford, 1991. (Cited on page 143)

[253] Sulo, R., Berger-Wolf, T., and Grossman, R. Meaningful Selection of Temporal Resolution for Dynamic Networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, MLG '10, pages 127–136. New York, NY, USA, 2010. ISBN 978-1-4503-0214-2. doi:10.1145/1830252.1830269. (Cited on page 145)

[254] Greene, D., O'Callaghan, D., and Cunningham, P. How Many Topics? Stability Analysis for Topic Models. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD '14, Nancy, France*, pages 498–513. 2014. doi:10.1007/978-3-662-44848-9_32. (Cited on page 145)

[255] Jungherr, A. Twitter use in election campaigns: A systematic literature review. *Journal of Information Technology & Politics*, 13(1):72–91, 2016. doi:10.1080/19331681.2015.1132401. (Cited on page 146)

[256] Boulianne, S. Social media use and participation: a meta-analysis of current research. *Information, Communication & Society*, 18(5):524–538, 2015. doi:10.1080/1369118X.2015.1008542. (Cited on page 152)

[257] Kreiss, D. Seizing the moment: The presidential campaigns' use of Twitter during the 2012 electoral cycle. *New Media & Society*, 2014. doi:10.1177/1461444814562445. (Cited on page 153)

[258] Khosravinik, M. *Handbook of Critical Discourse Studies*, chapter Social media critical discourse studies (SM-CDS), pages 582–596. Routledge, 2017. (Cited on page 157)

[259] Unger, J., Wodak, R., and KhosraviNik, M. *Critical Discourse Studies and Social Media Data*. Sage, 4th edition. ISBN 9781473916579. (Cited on page 157, 160)

[260] Kreis, R. The "Tweet Politics" of President Trump. *Right-Wing Populism in Europe & USA*, 16(4):607–618, June 2017. doi:10.1075/jlp.17032.kre. (Cited on page 157)

[261] Esposito, F., Esposito, E., and Basile, P. White House Under Attack: Introducing Distributional Semantic Models for the Analysis of US Crisis Communication Strategies. In N. C. Lauro, E. Amaturo, M. G. Grassia, B. Aragona, and M. Marino, editors, *Data Science and Social Research*, pages 175–183. Springer International Publishing, Cham, 2017. ISBN 978-3-319-55477-8. (Cited on page 157)

[262] Ritter, A., Cherry, C., and Dolan, B. Unsupervised modeling of twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180. Association for Computational Linguistics, 2010. (Cited on page 158)

[263] Amigó, E., Carrillo de Albornoz, J., Chugur, I., Corujo, A., Gonzalo, J., Meij, E., de Rijke, M., and Spina, D. Overview of RepLab 2014: Author Profiling and Reputation Dimensions for Online Reputation Management. In *Information Access Evaluation. Multilinguality, Multimodality, and Interaction - 5th International Conference of the CLEF Initiative, CLEF 2014, Sheffield, UK, September 15-18, 2014. Proceedings*, pages 307–322. 2014. doi:10.1007/978-3-319-11382-1_24. (Cited on page 159)

[264] Shi, B., Ifrim, G., and Hurley, N. Learning-to-Rank for Real-Time High-Precision Hashtag Recommendation for Streaming News. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 1191–1202. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 2016. ISBN 978-1-4503-4143-1. doi:10.1145/2872427.2882982. (Cited on page 159)

[265] Shi, B., Ifrim, G., and Hurley, N. *Insight4News: Connecting News to Relevant Social Conversations*, pages 473–476. Springer Berlin Heidelberg, Berlin, Heidelberg,

2014. ISBN 978-3-662-44845-8. doi:10.1007/978-3-662-44845-8_38. (Cited on page 159)

[266] Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 380–390. 2013. (Cited on page 159)

[267] Baldwin, T., Kim, Y.-B., de Marneffe, M. C., Ritter, A., Han, B., and Xu, W. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. *ACL-IJCNLP*, 126:2015, 2015. (Cited on page 159)

[268] Ge, T., Cui, L., Chang, B., Li, S., Zhou, M., and Sui, Z. News Stream Summarization using Burst Information Networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 784–794. Association for Computational Linguistics, Austin, Texas, November 2016. (Cited on page 160)

[269] Toshniwal, A., Taneja, S., Shukla, A., Ramasamy, K., Patel, J. M., Kulkarni, S., Jackson, J., Gade, K., Fu, M., Donham, J., Bhagat, N., Mittal, S., and Ryaboy, D. Storm at Twitter. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD 14, pages 147–156. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2376-5. doi:10.1145/2588555.2595641. (Cited on page 160)

[270] Kulkarni, S., Bhagat, N., Fu, M., Kedigehalli, V., Kellogg, C., Mittal, S., Patel, J. M., Ramasamy, K., and Taneja, S. Twitter Heron: Stream Processing at Scale. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 239–250. ACM, New York, NY, USA, 2015. ISBN 978-1-4503-2758-9. doi:10.1145/2723372.2742788. (Cited on page 160)

[271] Meng, X., Bradley, J., Yuvaz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al. Mllib: Machine learning in apache spark. *JMLR*, 17(34):1–7, 2016. (Cited on page 160)

[272] Bednarek, M. Investigating evaluation and news values in news items that are shared through social media. *Corpora*, 11(2):227–257, 2016. doi:10.3366/cor.2016.0093. (Cited on page 160)

[273] Gallagher, R. J., Reagan, A. J., Danforth, C. M., and Dodds, P. S. Divergent discourse between protests and counter-protests: #BlackLivesMatter and #AllLivesMatter. *PLOS ONE*, 13(4):1–23, April 2018. doi:10.1371/journal.pone.0195644. (Cited on page 161)

[274] Freelon, D. G., McIlwain, C. D., and Clark, M. D. Beyond the hashtags:#Ferguson,#Blacklivesmatter, and the online struggle for offline justice. *Available at SSRN: https://ssrn.com/abstract=2747066*, 2016. (Cited on page 161)

[275] Lee, M. and On, B. Effective social graph visualization techniques using friend and hierarchical matching. *Intelligent Data Analysis*, 20(2):417–438, 2016. doi: 10.3233/IDA-160812. (Cited on page 161)

[276] Muthiah, S., Butler, P., Khandpur, R. P., Saraf, P., Self, N., Rozovskaya, A., Zhao, L., Cadena, J., Lu, C.-T., Vullikanti, A., Marathe, A., Summers, K., Katz, G., Doyle, A., Arredondo, J., Gupta, D. K., Mares, D., and Ramakrishnan, N. EMBERS at 4 years: Experiences Operating an Open Source Indicators Forecasting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 205–214. ACM, New York, NY, USA, 2016. ISBN 978-1-4503-4232-2. doi:10.1145/2939672.2939709. (Cited on page 161)

[277] Rath-Wiggins, L., Bouwmeester, R., Spangenberg, J., Strobel, G., Dorchain, M., Jaho, E., and Sarris, N. REVEAL FP7-610928: Requirements analysis and specifications. Report 1, FP7-610928, May 2014. Available `http://revealproject.eu/wp-content/uploads/D1.1-Requirements-Analysis-and-Specifications.pdf`. (Cited on page 162)

[278] Wu, X., Xie, F., Wu, G., and Ding, W. Personalized news filtering and summarization on the web. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 414–421. IEEE, 2011. (Cited on page 162)

[279] Arguello, J., Diaz, F., Lin, J., and Trotman, A. SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1147–1148. ACM, 2015. (Cited on page 162)

[280] SalahEldeen, H. M. and Nelson, M. L. Losing my revolution: how many resources shared on social media have been lost? In *International Conference on Theory and Practice of Digital Libraries*, pages 125–137. Springer, 2012. (Cited on page 162)

[281] Potts, A., Bednarek, M., and Caple, H. How can computer-based methods help researchers to investigate news values in large datasets? A corpus linguistic study of the construction of newsworthiness in the reporting on Hurricane Katrina. *Discourse & Communication*, 9(2):149–172, 2015. (Cited on page 163)


sfi
science foundation ireland
fondúireacht eolaíochta éireann

UCD
DUBLIN

Insight