ELD:
Event TimeLine Detection

A Participant-Based Approach to Tracking Events

NICHOLAS MAMO

Supervised by Dr Joel Azzopardi

Co-supervised by Dr Colin Layfield

Department of Artificial Intelligence

Faculty of ICT

University of Malta

October, 2019

A dissertation submitted in partial fulfilment of the requirements for the degree of M.Sc. AI.

WWW.UM.EDU.MT

First edition, October 28, 2019

Statement of Originality

I, the undersigned, declare that this is my own work unless where otherwise acknowledged and referenced.

Candidate Nicholas Mamo

Signed 

Date October 23, 2019

Acknowledgements

I would like to thank the lecturers at the University of Malta, and especially those in the department of Artificial Intelligence, who prepared me for this dissertation. Most importantly, ELD would not have become what it is today without the guidance and expertise of my supervisor, Dr. Joel Azzopardi, and my co-supervisor, Dr. Colin Layfield.

Special thanks goes out to Hervé Boisdé and Jorge Rodrigues for their help in the evaluation of the USA Midterm Elections of 2018, and to Christian Camilleri for contributing his knowledge of UFC in the corresponding analysis. I would also like to express my gratitude to Paul Doyle, football writer at The Guardian, who found time in his schedule to give an interview for the qualitative analysis.

Finally, I would like to express my appreciation for the continuous support of family members and friends throughout this dissertation.

The research work disclosed in this publication is partially funded by the Endeavour Scholarship Scheme (Malta). Scholarships are part-financed by the European Union - European Social Fund (ESF) - Operational Programme II – Cohesion Policy 2014-2020 "*Investing in human capital to create more opportunities and promote the well-being of society*".

European Union – European Structural and Investment Funds
Operational Programme II – Cohesion Policy 2014-2020
"Investing in human capital to create more opportunities and promote the well-being of society"
Scholarships are part financed by the European Union - European Social Funds (ESF)
Co-financing rate: 80% EU Funds;20% National Funds

Abstract

People all over the world watch and talk as events unfold, but what does it take for a machine to truly track an event through this crowd-sourced narration?

Event TimeLine Detection (ELD) is a real-time Topic Detection and Tracking (TDT) solution to track events using Twitter with the hypothesis that it takes a deeper understanding of the event's domain for a machine to describe its evolution thoroughly. We test this hypothesis predominantly on football matches as we propose the novel concept of Automatic Participant Detection (APD) to extract an event's participants before it even starts.

We use the detected participants to track events in the TDT component, in which we contribute a novel feature-pivot algorithm. Seeking to explain developments within events, TDT hands our two novel summarization algorithms not only the corpus they are meant to describe, but also the topical keywords within.

Our evaluation demonstrates APD's potential to discover participants early on while exposing the algorithm's dependency on well-defined environments. As a result, TDT boosts its event coverage through the newly-found participants.

TDT's understanding of developments and their make-up also leads to improved expressiveness in summaries when compared to authoritative content from Twitter and the mainstream media. In fact, an interview with Paul Doyle, a football writer at The Guardian, reveals that ELD's improvements shift the focus of event tracking from TDT to summarization.

Through ELD and its contributions we show that although machines may not have the same understanding that humans accrue over time, they can gain it. In turn, this comprehension permits them to track events closely.

Contents

List of Figures

List of Tables

List of Abbreviations

ELD Event TimeLine Detection vi

FIRE FIRE: Finding Important News REports 2

TDT Topic Detection and Tracking vi

APD Automatic Participant Detection vi

VSM Vector Space Model 10

BOW Bag of Words 11

TF-IDF Term Frequency-Inverse Document Frequency 11

IR Information Retrieval 49

IDF Inverse Document Frequency 12

TF-ICF Term Frequency-Inverse Corpus Frequency 12

ICF Inverse Corpus Frequency 30

NER Named Entity Recognition 17

NLTK Natural Language Toolkit 49

MMR Maximal Marginal Relevance 26

P@k Precision at k 66

AP Average Precision 66

MAP Mean Average Precision 66

R-precision Ranked-Precision 67

FMMR Fragmented MMR 4

DGS Document Graph Summarizer 4

1

Introduction

Late in 2010, a fruit-seller by the name of Mohamed Bouazizi set himself on fire. Few could have predicted what would follow. Bouazizi's despaired flame spread like wildfire from the small city of Sidi Bouzid to the whole of Tunisia and beyond [1, 2].

The Arab Spring forever changed the faces of countries like Tunisia, Egypt and Libya. Thousands took to the streets to voice their discontent over inflation and growing political oppression. However, the revolution did not only happen in those streets - it extended beyond the towns, grew bigger than countries, and played out globally on social media [1].

During the Arab Spring, Facebook and Twitter contributed to organise activists and coordinate protests in Northern Africa[1], so much so that the demonstrations became known as the Facebook and Twitter Revolutions. Today, those labels sound hyperbolic. Even a short time later, research described the social networks as instrumental, but not necessarily catalytic [2]. Nevertheless, that was a turning point for Twitter, which now assumes the role of a news outlet for many of its users [3, 4].

Over time, Twitter has helped narrate anything from terrorism-related news [3, 5, 6] to natural disasters [7, 8]. Literature too looked to Twitter to understand what is happening around the world, with several systems looking broadly for news that is breaking worldwide [9, 10, 11]. However, Twitter does not only talk about the news that shocks or disturbs, but also about that which thrills and impassions people.

For years Twitter has been used to discuss sports. In 2018, tweets about the FIFA World Cup alone were viewed billions of times; people reacted, speculated and celebrated on the social network[2]. Over the years, algorithms appeared in literature that

[1] https://web.archive.org/web/20180814184054/https://advox.globalvoices.org/2011/04/27/mena-journalists-cyber-activists-in-the-line-of-fire/, last accessed on March 2, 2019

[2] https://web.archive.org/web/20190210100547/https://blog.twitter.com/en_us/topics/

University of Malta
L-Università ta' Malta

focus narrowly on these kinds of events, including American football [12], basketball [13] and, most predominantly, football [14, 15].

Nevertheless, while everybody contributes to the global conversation, the content can rapidly spiral out of control. Even back in 2012, Twitter's conversation about Hurricane Sandy peaked at 16,000 tweets per minute [7]. Provide these microblogs to a citizen and they will be none the wiser, only over-burdened.

It is for this reason that focus has gradually shifted from retrieving information to using Artificial Intelligence to understand this data and explain it back to the consumer. Event TimeLine Detection (ELD) is our proposed solution in this research area, taking on the information overload problem and instead allowing users to follow the dialogue, rather than be buried underneath it.

1.1 Proposed Solution

Event TimeLine Detection (ELD) is predominantly a Topic Detection and Tracking (TDT) solution that aims to leverage the velocity, volume and low latency of Twitter to consume information and instead provide its users with news. Our approach is inspired and based on our previous work – FIRE: Finding Important News REports (FIRE) [9].

FIRE's model looks for breaking news around the world using Twitter, splitting the TDT approach into two steps. Firstly, it clusters incoming tweets to identify what people are talking about. Then, it looks at how these topics are being discussed to verify that they are indeed emerging. This set-up allowed FIRE to capture breaking news very early and with a fine granularity [9]. In ELD, we build on this model, but we turn our attention towards specific events.

Throughout this dissertation, we think of an event similarly to the rest of literature – a definition that revolves around two principal notions. Most studies think of an event as something that happens in a certain time window in a particular location [8, 16, 17, 18, 19, 20, 21, 22, 23]. We follow a similar definition, but we do not exclude that in a planned event, a number of participants may be instrumental in its developments.

Definition 1 (Event) *A series of semantically-related developments that happen over a fixed or indefinite time window and in a particular space, possibly driven by any number of participants.*

Each such event may comprise any number of small developments, or topics, that shape the event. This is implicit in most studies, but a general understanding of a topic within an event is a surge in discussion of a subject [16, 24]. Allan et al. describe this facet

`events/2018/2018-World-Cup-Insights.html`, last accessed on February 10, 2019

in similar terms, citing the importance of a topic being unexpected [21]. Our definition of a development is in line with this research:

Definition 2 (Development) *A significant moment, or a semantic topic, within an event.*

The last piece of the puzzle is the participant - an actor that shapes the event's course. Sakaki et al. and Zhao et al. both consider them to be the users tweeting about an event [8, 20]. Others interpret them as the actors in the unfolding event. Few studies consider the importance of participants in this way, but those which do share similar definitions.

A number of studies track football matches using not only team names, but also the names of the players [13, 15, 25, 26] and the coaches [13]. Similarly, Han et al. tracked Superbowl XLVII using the names of coaches and key players [27]. Along the same vein, McMinn and Jose's system looks for named entities in tweets, considering their importance within events like political debates [18].

We adopt a similar interpretation of participants to enrich the definition of an event. In this way, an event is not described solely in terms of its temporal and spatial effects, but also as a product of its participants' contributions:

Definition 3 (Participant) *An animate or inanimate entity that actively influences an ongoing event.*

Following these definitions, ELD takes as inputs a short description and the time window of the event, and follows proceedings on Twitter to create a timeline that caters to particular interests. In ELD, we focus extensively on football matches since they are events that generate a lot of discussion and are simple to evaluate. However, we postulate that our approach can be applied not only to other types of events, but may also be applicable on other microblogging platforms that provide sufficient coverage of events. We identify three desirable characteristics that ELD should exhibit.

Firstly, it should minimize latency in delivering news. Twitter is the peak representation of fast streams, with the brevity of the microblogging environment contributing not only to volume, but also immediate publication times. Nonetheless, the insistence of literature on using batch or near-real-time workflows introduces latency.

Secondly, a successful system maximizes the granularity of detected topics. The coarse beauty of Twitter's scale is that each distinct voice adds its own character to an evolving story. Many existing systems compromise on granularity [25, 28, 29], resulting in timelines that are limited to the inescapable facts without enriching the narrative.

Thirdly, the returned updates should be human-readable and understandable. TDT systems like ELD harbour a lot of potential to reduce inundating streams to consumable

pieces of information. To serve this purpose, these systems need to go beyond detecting important topics; they need to understand moments and explain them back.

In this dissertation, we make contributions in various areas of Artificial Intelligence, including big data, text mining and TDT. In fact, ELD's central hypothesis is that TDT systems need to understand the domain if they are to achieve the above goals. It is from this point that ELD departs.

A machine's understanding differs fundamentally from a human's understanding. In ELD, we think of a machine's understanding as learning about the domain, which ultimately yields tools with which to achieve the broader goal of TDT better. The process of understanding starts with Automatic Participant Detection (APD) even before the event itself starts.

The novel concept of APD is placed at the beginning of ELD's process and allows our system to find the main participants that drive an event. It does so from an initial Twitter stream, before extrapolating other participants from Wikipedia. Once APD concludes, this understanding phase yields a list of participants, such as the players in a football match. ELD considers them as tools gleaned from the understanding of the domain and uses them to broaden coverage of an event by tracking them during the proceedings.

The second phase builds on FIRE's novel document-pivot and feature-pivot combination to identify important topics as they emerge throughout the event. In this area, we present our own novel, feature-pivot algorithm that facilitates the transition towards a real-time system. TDT too is a process that goes beyond detecting developments and guides summarization through understanding.

The document-pivot algorithm uses clustering to find the documents responsible for the salient moments. Then, our feature-pivot approach identifies the topical keywords that lead to a breaking development and quantifies their importance. In this way, the document-pivot technique provides the documents that discuss a topic and the feature-pivot approach extracts from these tweets the topical keywords. It is this in-depth understanding of each individual moment that fuels our two summarization approaches.

The third and final summarization phase transforms the machine-readable results of TDT into legible summaries to form a timeline. The two novel approaches - the Fragmented MMR (FMMR) and the Document Graph Summarizer (DGS) - seek to explain developments in the same multi-faceted way that Twitter users describe the moments; the former picks documents iteratively, and the latter uses graphs to extract the different conversation perspectives.

It is around these novelties that we formalize our hypotheses for ELD. We look to create a system that continuously tries to understand an evolving event to finally explain

its developments in detail. The next section takes these contributions and uses them to formulate the aims and objectives of ELD.

1.2 Aims and Objectives

Machines have an understanding of a domain that is not only different from a human's, but also essentially simpler. ELD's methodology is our attempt to find an answer to the hypothesis that a deeper understanding of an event's domain leads to improved results in the subsequent processes. Therefore to achieve a global understanding of ELD's strengths and weaknesses, we work towards the following objectives:

- Determine how well participants can be identified before an event has started;
- Explore the ways in which APD contributes to topic detection;
- Identify the effects of the combined document- and feature-pivot approach on topic detection in a narrow stream;
- Examine the link between the fragmentation of topic detection and summarization, and how TDT can contribute to summarization; and
- Analyse summarization's performance in describing developments.

Although literature points towards certain benefits of identifying participants within an event, the concept of APD as presented in this dissertation is novel. The first of our objectives is to explain this research problem in terms of its motivation and benefits. Moreover, we aim to explore its viability and challenges in the scope of microblog streams.

Our solution to extract participants from a stream affects the TDT component. Thus it is important to quantify APD's contribution to topic detection algorithms to understand the potential of this area and how it could be applied in future research.

TDT is at the core of ELD. In our proposed approach, we bring the novel combination of document-pivot and feature-pivot algorithms to a narrow stream. We perform an analysis of whether the positive results noted in FIRE [9] carry over to narrow Twitter streams. Moreover, we contribute a new feature-pivot algorithm. We evaluate its effects on ELD's TDT component and its link to summarization.

With the help of the TDT algorithm, the two summarization algorithms consider topic fragmentation to be an asset, rather than a liability. The last objective of ELD is

to explore this idea and analyse its results through quantitative and qualitative evaluations.

We evaluate our APD novelties in the domains of sports and politics. We pursue our evaluation by following popular literature in focusing predominantly on the domain of football. As enumerable environments, football matches allow for an easily-quantifiable analysis of ELD's struggles and qualities.

This quantitative analysis demonstrates the potential of APD to accurately identify the main participants before the event starts. However, it also brings out its fragility in loosely-defined contexts.

Football's popularity also permits for an understanding of how our TDT approach performs in different scenarios. By using widely-available ground truth, we perform an analysis that shows how our TDT approach is impervious to tweeting habits.

Moreover, the coverage that football enjoys allows an analysis of the summaries generated by our algorithms. We perform an evaluation that reveals improvements when comparing our summaries with tweets by authoritative Twitter accounts and reports in the mainstream media. We round up the analysis by conducting a semi-structured interview with Paul Doyle, a football writer at The Guardian.

In this dissertation, we demonstrate that TDT approaches struggle when they are not allowed to comprehend the domains of events. Through an initial understanding, ELD exhibits benefits throughout its process, although its timelines leave room for improvements to be comparable with media reports. The rest of this dissertation revolves around how we answer the questions posed by these aims and objectives.

1.3 Dissertation Overview

Next, Chapter 2 provides a brief overview of background principles behind ELD and Chapter 3 looks at existing literature. Chapter 4 shifts the focus from existing research to ELD, describing the proposed system's workflow and the motivations behind our decisions. We move on to the implementation details in Chapter 5 before evaluating ELD and discussing its results in Chapter 6. Chapter 7 concludes this dissertation, including how the developed system can benefit from future work.

2

Background

For much of its process, ELD is either dealing with Twitter or the content derived from it. This chapter examines the characteristics that have made Twitter the standard base for many TDT projects. Next, we present the link that transforms human-readable information into a machine-readable representation.

2.1 Twitter

It is difficult to look at a microblog and imagine how a few hundred characters can tell a story. In the world of blogs and long articles, tweets are the antithesis of narrative. Consider, however, the democratization of information dissemination that results in the small contributions of individuals all over the world to form one whole picture.

Far from the long status updates of Facebook[1] and the forum-like structure of Reddit[2], the concept of microblogging has been gaining popularity. Shedding lengthy comments in favour of terse messages, normally referred-to as microblogs, these networks provide various benefits to its users.

This family of platforms encompasses some of the largest social networks in existence. Although unpopular in Europe and in the United States of America, Sina Weibo[3] has amassed a huge following in China [30]. Tumblr[4], a blogging-oriented platform has rivalled with Twitter[5] in the past [31].

More than any other network, Twitter has become a popular platform not only for regular netizens, but also for the scientific community. Its huge user base and the incre-

[1] `http://facebook.com`, last accessed on February 17, 2019
[2] `http://reddit.com`, last accessed on February 17, 2019
[3] `https://weibo.com/login.php`, last accessed on February 17, 2019
[4] `https://tumblr.com/`, last accessed on February 17, 2019
[5] `http://twitter.com`, last accessed on February 17, 2019

mental changes refined the microblogging recipe. Although other modern social networks, like Mastodon[6], have tried to emulate its success, Twitter remains one of the most commonly-used social networks.

Up until recently, Twitter was synonymous with its 140-character microblogs, called tweets [4, 32, 33, 34]. All of the text, user mentions, hyperlinks and media had to share these 140 characters, nibbling away from users' expressive freedom. Since September of 2017, Twitter has relaxed these limitations to permit 280 characters, but users venture only cautiously beyond the historical restrictions[7].

The platform also developed its social aspect built atop a graph structure. Twitter allows its users to follow other users to get updates on their timelines. This results in a directed mesh that connects all users based on their follower-followee relationship [1, 4, 32]. Users can then interact with each other and their content in various ways.

One popular conversation tool is the ability to tag other users by using the @ symbol, followed by their handle, or username. Twitter connects together replies to tweets, gradually constructing a chain of microblogs to form a conversation among users [4, 16, 32, 33]. Content interaction is another important facet of conversation on Twitter.

Users can like microblogs and retweet, or share, tweets on their own timeline for their own followers to see. Over time, this facility has climbed to become a predominant means of conversation. Furthermore, users can add their own comments to these retweeted microblogs [4, 16, 32, 33].

Retweets are particularly useful in the context of news dissemination, as was the case during the Arab Spring [1]. However, whereas retweeting is a convenient way of sharing content, spreading tweets requires a mix of considerable following and luck. A Twitter-born solution is the hashtag convention, which tags tweets using the hash symbol (#) followed by a keyword [16, 35].

This convention has found many uses on Twitter, from building social movements to categorizing tweets, thereby facilitating their discovery [1, 32, 36]. Hashtags also pack a lot of semantics into a single keyword, countering the relative brevity of tweets [36]. In turn, the short nature of microblogs elicits two main advantages that have attracted researchers as much as users.

Firstly, the short messages take less time to write than an elaborate blog post, for example [4, 37, 38]. Zhao et al. examined real-time events and concluded that users comment about developments within 13 seconds of them happening [12]. Secondly, users post more often than they do on traditional mediums [4, 6, 34, 37, 39].

[6] `https://mastodon.social/about`, last accessed on February 21, 2019

[7] `https://web.archive.org/web/20180823140339/https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html`, last accessed on March 2, 2019

The sheer volume of tweets published at a rapid velocity create a big data problem that has to be managed carefully. Nonetheless, the prospect of this extensive, low latency knowledge has made Twitter an alluring venture for those researchers that found ways to navigate it properly. However, managing the inundating stream of tweets is only the start of a list of challenges.

As expected of a real-time social network, Twitter is teeming with emotions, and this sentiment works its way into the tweets [16, 17, 29, 34]. This proved problematic in studies like Löchtefeld et al.'s, which looks at sport events - environments that are especially emotionally-charged and riddled with colloquialism [29].

Emotion is only the tip of the iceberg when it comes to colloquialism. Social media's open nature means that anyone can contribute to any subject and language can be quite incoherent. Twitter's brevity encourages this behaviour, such as through abbreviations [17, 34, 40].

This growing understanding of Twitter has been facilitated since tweets are predominantly published publicly [32, 40]. Although this is a desirable feature, publicity also breeds spam and noise. Even early in Twitter's life, Java et al. observed that the majority of the tweets are what they called "daily chatter", interesting to very few users [4]. Other times, bots exploit Twitter's openness to serve their own agenda, in the meantime creating a lot of spam [9, 28].

A more recent problem is fake news, which gained prominence during the US Presidential Election Campaign of 2016, and again a few months later, while Britain was voting to leave the European Union [41]. Research into verifying online information is still in its infancy, but it does point towards an ever-increasing need of wariness when mining social media.

Researchers who brave these challenges discover an immense source of knowledge about the world. Twitter's API makes all of the public microblogs available for collection - albeit with some limits - which puts a lot of data in the hands of researchers. Through this set of tools, prospective miners of the Twitter stream can listen to general discussions or focus on a narrower stream [37].

Most prominently, Twitter has been used to commentate events of historical importance from all over the world, from major disasters [6] to political upheavals [1]. To this end, systems like FIRE [9] and TwitterMonitor [10] use Twitter's API to extract major breaking news from all over the world. Other, more specific projects, like Marcus et al.'s TwitInfo, focus on singular events, like football matches [42].

Twitter has also taken an important role in emergency situations. As early as in 2010, Vieweg et al. monitored conversations that centred around flooding and wildfire emergencies to build an overview of what was happening [43]. Perhaps driven by the

disastrous implications of earthquakes, they have also been the focus of many research projects, often permitting detection within a few minutes [6, 8, 42, 44].

In such events, beyond recounting and reporting news, it could be argued that Twitter becomes a microcosm of the world itself. The changes in sentiment [16] and communication habits [26, 45] reflect the real-life dynamics, but who is Twitter?

As FIRE looks to extract important news by mining Twitter's conversations, the definitions too reflect the meaning of importance. Mamo and Azzopardi reason that what is significant to Twitter may not always be representative of what the world thinks is important. For this reason, FIRE adopts a narrow definition of importance that is relative only to what the social network finds important [9].

Apart from being a networking platform, Kwak et al. find that Twitter serves the function of being a news outlet to many of its users [33]. It is therefore unsurprising that Twitter's user base also includes many journalists [46] who seek to simultaneously push news and learn about it from sources present in the heart of the action [1].

It is for these reasons that research, rather than stopping at understanding the digital world of Twitter, is attempting to figure out the real world through this social network [26]. Many systems have considered Twitter to be more than simply a social network, but a window into the world [6]. The next section takes a more technical view of the machine representations of tweets.

2.2 Vector Space Model

Like most data on the web, tweets represent information created by and aimed at humans. In spite of all the content that they harbour, these messages are useless in their raw form to an algorithm. To make sense of human-readable information, the natural approach is to convert it into a more adept representation.

In most cases, the go-to solution is the Vector Space Model (VSM), which has its basis in vectors. As mathematical constructs, vectors deal more comfortably with numbers than with the ambiguity of words. The model represents documents in a multi-dimensional space. Each dimension is a feature with a weight that is proportional to its importance in the document. In this way, the mathematical representation allows comparisons between documents based on their feature make-up [47, 48].

For this reason, the choice of features depicts a caricature of the content, bringing out the most important characteristics according to the system's interpretation. In their system, Sarmento et al. were interested in how entities interact with each other, which they represented as co-occurrence features [47].

Petkos et al. and Ratinov et al. consider the semantics of words by harnessing the knowledge embedded in WordNet and Wikipedia [49, 50]. Ultimately, Petkos et al. abandoned this representation in favour of a simpler approach, resembling the popular Bag of Words (BOW) representation [49].

The BOW's dimensional space is synonymous to that of the words [50]; each word, or token, in the document is represented with its own dimension, with a weight that is normally proportional to the number of times that it occurs. Implicitly, since the vector space has no ordering, this model also ignores the order of words.

Although the BOW model is fairly simple, it is also flexible. To reduce dimensionality, one common preprocessing step is to coalesce words with similar morphological origins using word normalization. Mamo and Azzopardi adopt the wildly popular suffix stripping approach introduced by Porter [51], commonly referred to as a Porter Stemmer [9].

Petkos et al. go in a different direction by reducing a word to its lemma [49]. This approach has the advantage of reducing a word to another real term, but it is also over-reliant on a dictionary [52, 53]. Although stemming may not necessarily map words to real terms, it minimizes dimensionality more than lemmatization [53]. Moreover, whereas both approaches have been shown to improve results [53], various studies noted the complexities of lemmatization, which weighs on the efficiency [52, 53, 54].

Additional preprocessing steps depend on the application. One distinction of note is the approach taken by both Ozdikis et al. and Löchtefeld et al., who deal with sports commentary on Twitter. In these cases, developments often result in emotions, which manifest themselves in repeated characters, like *GOOOAAAL*. Both studies overcome these issues by removing repetition in tokens [17, 29].

Stopword removal is another regular occurrence in VSM models, filtering out terms like *is* and *when* [9]. Seen in a wider context, the decision to ignore common words is akin to assigning them a weight of zero. In fact, it is common for terms to have different weightings.

Various authors have surveyed term weighting schemes and explain how feature values are made up of three components - the local weight within the document, the global importance of a keyword, and a normalization factor [48, 55].

The local weight measures the importance of words within one document, and it is usually combined with the global context to put the importance of a token in a broader perspective. The Term Frequency-Inverse Document Frequency (TF-IDF) scheme has become one of the most commonly-used methods to weight terms [56].

The TF component assigns a value that is proportional to the frequency of a term in the content [55]. Other variations were introduced over time, including the augmented

normalized term frequency of Salton and Buckley [55]. The IDF part deals with the discrimination value of terms [48, 55].

As early as in 1972, Sparck Jones argued that infrequent terms contribute to their specificity [57]. From this argument sprang the idea that those terms which are frequent in the local context, but do not appear often globally, characterize the document. This comprises the Inverse Document Frequency (IDF) [48, 55], which appears in many systems [49, 58, 59].

Lastly, the normalization factor has to be understood within the context of domains where the length of documents varies greatly. This prompted Salton and Buckley to introduce popular measures so as not to promote long content, including vector normalization [55].

Although by definition, microblogs limit the discrepancy in length among documents, they present another problem. Brevity often limits the frequency of terms to at most one per document, in this way transforming the IDF component into the inverse term frequency. Moreover, the entire corpus may not be available beforehand to create the IDF table.

Reed et al. propose that a static corpus can approximate the distribution of a set of documents. The scheme, which they call Term Frequency-Inverse Corpus Frequency (TF-ICF), does not make any reference to microblogs at all [60]. However, FIRE uses it to overcome the challenges of a microblogging environment. By collecting an external corpus of tweets, the authors could measure the impact of keywords with knowledge of past usage [9].

2.3 Summary

Twitter being a stalwart component of TDT projects is not merely anecdotal. The social network's benefits are evident in the slew of projects in literature that uses its data. This chapter explored Twitter's possibilities and possible representations of tweets. The next part of this dissertation explores systems in literature that incorporate any of ELD's characteristics.

3

Literature Review

The world of possibilities afforded by Twitter has been explored from several angles, and many researchers preceded ELD in their ventures. This chapter starts by giving an overview of systems with a similar scope to our proposed solution.

Many of these systems blend together various approaches to reach their end-goal. ELD itself is an amalgamation of three research problems. Throughout the chapter we explore the various methodologies associated with these problems, and how the state-of-the-art approaches are relevant to this project.

3.1 Similar Systems

The sheer volume of tweets and the immediacy of their publication has made Twitter a thriving area of study. This section looks at existing systems in literature that sift through the content generated on the social network to extract meaningful topics.

The abundant microblogs, complemented with copious data about the interactions between users do more than fulfil human communication needs. With the Twitter API putting all this data at researchers' fingertips, many studies view Twitter as a running commentary of the world's events and try to bring order to this unstructured stream, reducing it to a more manageable narrative [61].

FIRE [9], TwitterMonitor [10], NewsStand [11] and others exploit a sample of the public Twitter stream to identify the major breaking news items around the world. Then, these prototypes compress thousands of messages into a more meaningful representation of what is happening around the world, whether as a timeline [9, 10, 62] or as an annotated map [11].

Unfortunately, these systems are inhibited by the number of tweets provided by Twitter API's and the noisy environment of the social network. In fact, the API pro-

vides only a random sample that represents anywhere between 1% and 2% of all public tweets. As a result, projects that rely on the public Twitter stream also find news in the general discourse. Owing to these limitations, this research area branches out into another family of solutions that focuses on specific areas of interest.

In such cases, the goal - and in many cases, the approach - is similar to the former class of Twitter use cases. However, instead of mining the entire public Twitter stream, the approaches use Twitter's filtering API to exploit a flow of tweets centred around particular topics [42, 63, 64, 65]. These artefacts can operate in different areas, such as politics [63, 66, 67, 68], sports [42, 63, 67], yearly events [63] and emergency situations [42, 66, 67, 68].

Both families of solutions share common components and face similar challenges. Building atop the Twitter API, these artefacts need to handle an emotionally-charged environment where spam and noise thrives [25] while identifying the salient talking points. Moreover, it is essential that such systems effectively deal with the velocity and volume of Twitter publications [69].

The general workflow of both types of artefacts starts with streaming, filtering and preprocessing incoming tweets before extracting salient developments. The detected sub-topics are presented to users in a human-readable form, commonly as textual summaries or ordered in a descriptive timeline. As these solutions diverge and move from the general public Twitter stream to more specific streams, so do they come with their own, specific challenges, starting with the very first step - defining the topic of interest.

The starting point of these systems is describing the subject, or how the Twitter stream should be filtered. These tracking keywords are often manually-defined, but the end-goal is to have a list of filtering keywords to cover the event. The simplest approach to track a topic is to select a small set of keywords that approximate the event.

For example, when Shou et al. sought breaking news about Chelsea F.C., they looked for tweets mentioning *Chelsea* [63]. However, such an approach on its own does not represent semantics adequately. While tracking typhoons and earthquakes, Sakaki et al. explain that polysemy could pose problems, and highlight the importance of filtering tweets according to their semantics [8].

Studies like [28], [29], [64] and [65] adopt a sounder solution by embracing hashtags as a classification tool defined and driven by Twitter users [16, 25, 33, 36]. Gillani et al.'s study, for example, employs hashtags like *#Euro2012* to collect tweets about the UEFA EURO 2012 football competition [64].

Nonetheless, a study of how the tweetosphere engages in conversation by Zhao et al. claims that only 11% of messages contain hashtags [12]. This is the reason why many studies do not depend solely on hashtags as a means to filter the Twitter stream [8, 26,

42]. Instead, studies consider what constitutes an event.

Some approaches describe the participants that drive the event, rather than simply the name of the event itself. For example, Corney et al.'s system, which focuses on football matches, tracks the teams and their players for the entirety of the event [15]. Shen et al. follow a similar approach when working in the domain of basketball [13].

These tracking solutions are not mutually-exclusive. Löchtefeld et al.'s main method of collecting football matches' corpora is by using hashtags, but they follow this up by extracting participants using a knowledge base [29].

Natural Language Processing (NLP) tools substitute the knowledge base in Shen et al.'s and McMinn and Jose's research, but like Löchtefeld et al., they do not exploit information about participants to broaden coverage of an event [13, 18]. Instead, they seek only to obtain a better understanding of the stream.

Naturally, the scope of an event has to be described adequately before it can be tracked, but what happens when a user is not knowledgeable enough about the event? Many researchers skirt this issue, implicitly or explicitly assuming that users have in-depth knowledge and define keywords that provide sufficient coverage [26].

After defining the starting point, these solutions proceed to reduce an endless stream of messages into a few salient topics. Some studies tightly couple their approaches with a class of events, or even a single event due the choice of algorithms.

Gillani et al. use the K-Means clustering algorithm to detect the most important developments during one football match [64]. Khan et al. [65] and Gao et al. [70] relax this heuristic by estimating the number of topics using various automated algorithms.

Timing is another important consideration during ongoing events. Batch systems are the most predominant, with algorithms like FIRE [9] and Sumblr [63] accumulating tweets in time windows before processing them. One downside of these approaches is that they introduce latency into the system, keeping these approaches from exploiting Twitter's velocity.

Corney et al. alleviate this problem by using smaller batches of a few hundred tweets at a time and only then processing them [15]. Conversely, Shen et al. create timed bins, processing tweets every ten seconds [13], and Nichols et al.'s algorithm uses a one-minute window [14]. Very few go a step further, applying their algorithms in real-time.

One of these real-time systems is Zhao et al.'s artefact [12], which uses a sliding time window to detect sub-events from American football matches less than a minute after they happened. Kubo et al.'s approach analyses the number of tweets per second, comparing it with a normal state [25].

Closely linked with the timing mechanism is the granularity, which very often depends on the underlying topic detection approach. Some studies, like [12], [28] and

[71], limit the types of detectable sub-events by training classifiers. This is a common occurrence in sport summarization systems since these kinds of events adhere to rigid rules. These approaches ensure that more machine-readable knowledge is available to the respective systems, but classification cannot handle unexpected developments

Others focus on the evolution of tweet volume. Just like Kubo et al.'s approach [25], several studies extract salient events by monitoring changes in the volume of tweets. In many cases, this simple approach only provides the incontestable sub-events - a phenomenon that is clearly observable during sporting events.

Low-profile developments, like yellow cards in football, are difficult to capture. This is because minor moments do not evoke the same emotions, and nor do they generate as many tweets [12, 14, 29, 42, 64, 71, 72]. Fetching topics with a finer granularity calls for more sophisticated means, such as by examining bursts in term usage [9, 73].

The final step in many of these systems is a description of the event in a form that end-users can understand. The simplest approach extracts the tweets that are at the heart of an event. [62] takes a very simplistic interpretation of summarization, returning only the first relevant tweet. Other variations promote tweets that contain entities [25], or the most representative documents [9, 13].

Since events are constantly evolving, some projects promote their temporal notion by representing them as a timeline. For example, in their study, Alonso et al. detect emerging news revolving around hashtags, and add contextual information to a timeline [67]. Lin et al. consider not only the temporal notion, but also how different sub-events link together to form a storyline [74].

Many of these systems' summarization modules revolve around the same concepts, but some take different routes. Certain approaches explain the sub-events that they unearth by constructing a graph made up of tweets or words. This graph is then used to extract tweets [68], or construct brand new messages from individual words or phrases [40].

Summarization concludes the workflow that starts from tweet filtering before detecting and describing emerging topics. The rest of this chapter will take a deeper look at these three main components of the systems that relate to ELD in some way. Section 3.2 lays out some basic definitions, and looks at the most influential approaches when it comes to expanding a list with similar entities, with the end-goal of providing broader coverage to an event.

Next, Section 3.3 examines how a stream of tweets can be reduced to a few salient topics. Section 3.4 concludes this chapter by outlining major contributions in the area of summarization, and explaining how different methods explain sub-events in such a way that is understandable to human users.

3.2 Entity Set and Query Expansion

As alluded to in the previous section, systems that seek to understand a specific stream share common ground - they must all define a starting point. In Twitter, this translates to a seed set of keywords that filter out voices that are not discussing one narrow subject or event. The aim of the stream filtering step, facilitated through the Twitter API, is to obtain coverage of an event. In ELD we hypothesise that by tracking participants, we can maximize the coverage of an event.

To the best of our knowledge, existing systems do not automatically extract participants before the event starts. Instead, they use a few select keywords to define an event or identify participants manually [13]. Thus, literature about this methodology, which we coin as Automatic Participant Detection (APD), is lacking. However, the importance of such an approach is implicit in a few systems.

Within the context of a football match, Shen et al. reason that a search is not only concerned with the teams, but also with their players and coaches [13]. McMinn and Jose detect participants later during the tracking phase using Named Entity Recognition (NER) techniques and then look for entity-specific developments [18].

These systems do not always extract entities automatically. However, they indicate that for the search to cover an event, it should also include the participants that are directly relevant. In lack of relevant literature, APD can be thought of as a function of two other popular areas of study - entity set expansion and query expansion.

The simpler of the two is entity set expansion. The idea behind this approach is to work up from a small set of items to an enlarged list of entities that belong to the same class as the initial group. In this way, the seed set is thought to contain examples of a class of elements, and the approach tries to complete this class by finding missing items [47, 75, 76, 77].

Although some concepts carry over from entity expansion to APD, our proposed method is arguably closer to the second approach - query expansion. The goal of query expansion is to fine-tune a search to reflect the information that a user really needs [37, 50].

In their study, Efron et al. explain how the brevity of Twitter could mean that tweets do not have enough space for all relevant terms. In this case, query expansion could capture more valuable content [78]. In a later study, Nakade et al. further add that although Twitter generates a lot of data, events that are either small or not in English may still suffer from a scarcity of tweets. The authors see query expansion as a means to maximize coverage by listening to all facets of an event [37].

APD aims to bring these advantages to real-time topic detection by maximizing in-

formation flow from Twitter. Nevertheless, APD differs from these two approaches. Unlike the research area of entity set expansion, in APD the seed set provided by the user before the event does not necessarily serve as a list of examples of a class that needs to expanded. It may be a simple reference to an event, like a hashtag. Moreover, unlike general query expansion, the described workflow is only interested in fetching participants of events, and not simply any related terms.

Thus, this approach is akin to a more specific case of query expansion, but understanding of the domain is necessary to filter out keywords that are not participants. In fact, as Gabrilovich and Markovitch explain, domain understanding is imperative in any query expansion task [79].

Query expansion usually starts off with understanding the concept behind the original query, in a broad sense. In this context, Ratinov et al. describe conventional searches as conceptual searching; the terms provided to this search do not necessarily cover the whole concept, but they describe its main aspects [50].

To obtain a general idea of the concept, query expansion normally starts with an initial search, regardless of the scope of the query [39, 50, 75, 77, 80, 81]. Zingla et al. outline the three ways of performing this look-up - using the local or global contexts, or using an external source altogether [81].

The local context simply performs a search within the corpus itself, using the returned documents to understand the query and grow the seed set. The global context uses the entire corpus to try and understand the domain. The alternative is an external source - usually a large knowledge base - to gain a more global understanding of the query beyond what is given in the corpus [81].

The authors' query expansion method for Twitter complements the local context with Wikipedia [81]. In fact, this encyclopaedia has soared to become one of the most popular choices in these kinds of algorithms, owing no little to its enormous size [82].

Gabrilovich and Markovitch describe its massive size and openness as two virtues that make Wikipedia an ideal source of information [79]. Egozi et al.'s system looks beyond the text for the meaning behind words, constructing a representation of documents based on the vector space of all Wikipedia concepts [83]. Furthermore, the structures of pages and the links among articles themselves are two options that have been exploited for their semantics in research studies, including to expand entity sets [35, 39, 47, 79, 81, 82].

For example, Zingla et al. explain how articles can give the general idea of a page without necessarily using all of the text [81]. Conversely, Milne et al. use the links connecting articles as the basis to build a thesaurus that would later serve to boost searches [82]. Wikipedia is also ideal for its organization in the way that it dedicates one page for

each such concept or named entity [47].

If structure is the priority, formal knowledge bases like DBpedia [81] and Freebase [84] are preferred to expand queries, and to understand and rank documents better respectively. An alternative for less sophisticated search queries are thesauri, which Nakade et al. use to back up searches [37].

Time-sensitive queries have also been explored in literature. In their study, Massoudi et al. consider that as topics and events evolve, so does Twitter's language. Thus, the authors opt to assign a higher score to keywords that are temporally-closer to the query [85].

Whatever the route that is taken, this additional knowledge is conceptually represented as a set of candidate keywords that can be used to expand the query. Naturally, both entity set and query expansion methods often result in an inundating list of candidates. To prune candidate keywords that are not related to the user's intended query, these methods conclude with a ranking and filtering stage [47, 81].

The results of entity set expansion and query expansion are often ranked in the end - a sort of assessment of the semantic similarity between the original query and the candidates. To this end, Milne et al.'s solution represents Wikipedia concepts using their outgoing links and bases their semantic relatedness on this overlap [82].

Various other similarity measures have been used depending on the application, ranging from cosine similarity [47] to graph-based approaches [39, 75, 80]. A threshold on the similarity can then filter out unrelated keywords, or the top candidates make up the final list.

Naturally, extending the candidate set also widens the scope of a search task that is already overwhelming on its own. TDT makes sense of the stream to reduce it to a few updates about the event. The next section explains this problem by exploring popular approaches to detect significant changes in a stream of messages.

3.3 Topic Detection and Tracking

Twitter's standard public API limits consumers to 50 tweets per second [12]. It is a generous limit, and one that is very rarely reached when dealing with narrow streams. However, it still represents momentous volumes that researchers have to curb.

Before being a problem for Twitter-based systems, this volume is an insurmountable step for the end-users themselves, for whom the immense data is humanly-impossible to consume. For this reason, the TDT track has proved essential in reducing a slew of microblogs into manageable pieces of information that describe an ongoing event.

The objective of TDT algorithms is to analyse a stream of documents, identifying the defining moments that indicate a change in discourse. By consuming this stream and reducing it to a list of developments, this methodology hides all the intricacies and retains only the important moments [19].

In one way or another, evolutions are the result of unusual developments. In his seminal study, Kleinberg represents topics that originate from unexpected patterns in conversations as an automaton. In its most basic form, this automaton has two states representing the two extremities of topic developments [86].

The automaton's first state represents the topic in its normal disposition. The second state of the automaton is the topic in an agitated position, indicating a change in language that abstracts a real-life development. As time passes and the nature of the event evolves, the automaton moves from one state to the other. Tracking is implicit in this workflow when the automaton remains in the same state through time [86].

This movement across states over time represents one of the two major approaches to TDT - feature-pivot methods. Central to this methodology is the temporal nature of the stream; features change with time, and a significant movement signals an important development [19, 26].

The second major family of TDT solutions is document-pivot algorithms. These approaches are usually centred around clustering methods. Although the cluster algorithms themselves consider the features of documents, the ideology of document-pivot approaches is that the resulting clusters each represent a single topic [26, 19].

An important concept of feature-pivot approaches is the measurement of a development's unexpectedness, which indicates a possible salient topic within an event. Kleinberg is among those who formalized the notion of 'burst', or an unnatural surge in usage [86].

Cataldi et al.'s feature-pivot method is one of the approaches that hinges on burst. Their algorithm calculates burst over 'nutrition' - a term that was later also adopted in FIRE [9]. Cataldi et al. understand nutrition to be a function of a term's usage and the authority of the bloggers that contributed to the associated documents [73].

In its process, Cataldi et al.'s algorithm calculates the burst value of each term from all documents in one window. Then, it ranks the features according to their relative importance. The approach automatically detects a cut-off point where the burst - based on nutrition - dips [73].

FIRE adapts Cataldi et al.'s approach to achieve a finer granularity and to make it more democratic by considering all authors as being equal [9]. In their own way, systems like Olariu et al.'s [40] too calculate the burst measure for each term. However, the majority of approaches calculate the surges in one simple feature - volume.

Dealing with the massive content that is generated online requires meticulous planning and difficult decisions to curb the number of documents. A simpler alternative does not look at each document individually, but assumes that when a significant topic change occurs, it leads to an overall increase in discussion volume [86].

Nichols et al.'s research confirms the notion that key moments in football matches, such as goals and red cards, prompt more discussions. However, they also make an unsettling revelation - moments that are less emotive, like yellow cards, do not spark quite the same surge in overall discussion [14]. Numerous other studies confirm these findings [12, 29, 64, 71, 72].

These results also say a lot about the nature of Twitter. No matter how one looks at Twitter, it remains a social network. TDT algorithms in this area depend on its users and the saliency of extracted topics has to be understood within this context. As we previously explained in FIRE's definition of importance, the topics in the stream are those that Twitter users find most interesting [9].

Other studies go the way of statistics to seek out developments. Sakaki et al. use Twitter to detect earthquakes in Japan, considering users as sensors that give signals when the ground shakes. Using the number of published tweets similarly to tweet volume, they approximate the probability that an earthquake really occurred [8].

Like Sakaki et al., other studies use statistical tests [87, 88] and other probabilistic approaches [18, 89, 90, 91] to analyse the significance of appearances in terms. These methods mean that these systems do not have to deal with arbitrary thresholds, or with the unbounded burst values that appear with the aforementioned methods [9, 73].

As events change and new topics emerge, it is only fair to expect that the language used to discuss them changes as well. Although Sumblr is predominantly a document-pivot method, the system compares the language models of successive summaries to determine whether new topics have surfaced [63]. Shen et al. put language models on the forefront, foregoing clustering altogether [13].

An alternative methodology for feature-pivot topic detection is based on graph theory. Weng and Lee [92], and Aiello et al. [26] describe approaches that create a graph linking together keywords using the strength of their association in the corpus. Using graph partitioning techniques, sets of terms can be found and considered to be the event's topics.

A central theme of these approaches is how they revolve around the usage of features, whether terms or the overall volume. The outcome gives little to no context to the words apart from knowing that they experienced a significant evolution. This problem is usually rectified by grouping together co-occurring breaking terms [40, 73, 87].

Document-pivot techniques are a contrasting sight with their idealistic reasoning

that if documents deal with the same topic, then they would share enough similarities to be naturally grouped together. This is in line with the general idea behind clustering algorithms, and indeed these approaches are the backbone of document-pivot topic detection [19, 26, 49].

Traditional algorithms, like K-Means have occasionally been the subject of experimentation [19, 93]. Nonetheless, such approaches often require heuristic knowledge, such as the number of clusters, or topics that are in the stream [19]. Naturally, this necessitates in-depth knowledge of the domain. For this reason, these algorithms are very rarely used in literature except as baselines.

Graph-based algorithms make an appearance in document-pivot methods as well. Instead of having nodes represent terms, the documents themselves assume the role of vertices, with the edges between them representing some sort of pairwise similarity [20, 49]. Similarly to before, graph partitioning, or clustering, finds document groups, akin to topics.

Hierarchical approaches have also gained footing in literature because they allow control over the granularity of topics by limiting the hierarchy's depth. Some approaches require prior knowledge about the number of clusters that exist [94], but others prune the hierarchy automatically [15, 95]. Nevertheless, a case could be made that hierarchical clusters are not well-armed to deal with the immense data of social networks [19].

It is this increasing demand of growing data streams that has pushed for approaches to sacrifice quality to cope. The demands of real-time streams that use Twitter have made incremental clustering the default choice [9, 13, 17, 19, 26, 49, 56, 96, 97].

Although the specifics vary from one method to the other, the underlying concept is almost identical; the algorithm receives documents sequentially and adds them to the most similar cluster if the similarity exceeds a certain threshold. If no such group is found, a brand new cluster is formed [17, 19, 98].

Sometimes, to limit the number of comparisons, clusters are allowed to die out after a period of inactivity. This could be interpreted in terms of temporal idleness [17], or based on the number of documents that have been processed since a cluster was last updated [98].

Nonetheless, incremental approaches are heavily-parametrized. In lack of sufficient research, these parameters are usually set empirically based on validation data [19, 26, 49]. Still, fragmentation and merging emerge as two major issues, incorrectly separating or combining topics respectively [19, 26, 49].

The crux of document-pivot algorithms is being able to assign relevant tweets to the same group. Twitter's brevity shackles clustering since short messages often have little overlap, even if they talk about the same topic, thereby fragmenting discussion. For this

reason, Phuvipadawat and Murata boosted the term weights of hashtags and pronouns as they saw them as components that are more identifiable and topical than simpler terms [99].

In reality, even the theoretical ideation of document-pivot algorithm rarely holds true. Daily rambling is the norm on social networks like Twitter [4], and thus, many of the generated clusters may not be actual developments, but noise or spam black holes [19].

To avoid this problem, most systems include a filtering stage. FIRE, for example, removes clusters that are likely to contain spam using a general neural network [9]. In similar fashion, Becker et al. classify clusters as on-topic or not [96], whereas Abdelhaq et al. score groups and retain only the top clusters as legit topics [97].

Both feature-pivot and document-pivot approaches have their fair share of usage in literature, and both families of solutions have their application. Although many feature-pivot algorithms bypass fragmentation, their reliance on co-occurrences may lead to unreliable results [26]. On the other hand, it is unusual for feature-pivot methods to require parameters barring the essentials. By avoiding clustering, they do not rely on an external algorithm's performance [6].

In general, TDT algorithms either involve a lot of calculations - especially in the case of feature-pivot methods - or a lot of comparisons and processing - notably in the document-pivot approaches. Since TDT algorithms are the cornerstone of many systems cited in Section 3.1, these challenges are usually behind the applications' major limitations.

The majority of systems that rely too heavily on the volume of the input stream suffer from very coarse granularity. Moreover, analysing document volume or keyword usage across time windows forces most feature-pivot methods to abandon real-time processing. Their dependency on time windows is a double-edged sword, promoting reliability and precision, but restricting timeliness.

Real-time topic detection within voluminous streams greatly limits the available options. It rules out algorithms that are not scalable, such as traditional approaches and hierarchical methods [19]. Due to all of these considerations, most of the systems presented in Section 3.1 are either batch or near real-time processing applications.

Systems that are not time-sensitive can venture beyond the data that Twitter provides and use supporting knowledge bases to improve results. De Maio et al. look to Wikipedia to perform a process that they term *Wikification*. The authors use Wikification to represent tweets not with keywords, but with the richer Wikipedia concepts [68]. Wikipedia has also been used as a source of confirmation for the news captured on Twitter [100] and to help link semantically-similar terms together [101].

Historically, feature-pivot and document-pivot methods were two opposing and mutually-exclusive approaches, as noted in any literature overview of the TDT track [19, 26, 49]. Nonetheless, scattered throughout literature are examples of systems that point towards the benefits of integrating these two classes of approaches.

Clustering is not central to feature-pivot algorithms, and neither is the grouping mechanism associated with the documents. However, studies are aware that terms do not provide enough context in isolation, thus requiring some form of grouping [40, 73, 87].

Elsewhere, Shen et al. [13], and by McMinn and Jose [18] heap importance on the participants, or named entities, when plotting the evolution of events. Their approaches are both founded on feature-pivot methods. However, their algorithms split streams according to the named entities that they mention.

Arguably, this stream-splitting approach does not necessarily fit the conventional definitions of document-pivot algorithms. In contrast with the idea behind these techniques, the resulting groupings are splits in the streams, not candidate topics. Nevertheless, the loose forms of clustering improve results thanks to a finer granularity of detected topics [13, 18]. FIRE consolidated these findings [9].

To the best of our knowledge, FIRE is the only system that combines the two distant approaches together in one. The approach clusters documents using an incremental algorithm and considers each resulting group as a candidate topic. Then, an adapted version of Cataldi et al.'s [73] algorithm finds any breaking terms in each cluster [9].

Apart from a finer granularity, FIRE achieves a high precision score because the combination of techniques adds another layer of confirmation to the process. Moreover, the feature-pivot technique itself yields knowledge about which terms are breaking since the features come from a single cluster with a common semantic basis [9].

These topic detection approaches reduce streams into a handful of terms or documents. Nonetheless, the result is often robbed of context, making it difficult for human readers to make sense of it. The next section presents an overview of summarization techniques that transform machine-readable knowledge into a breakdown of events that can be more easily consumed by human users.

3.4 Summarization

A Twitter stream for an event is impossible for human readers to consume in its raw form. The TDT track is providential in this regard because it reduces the abundant documents into its salient talking points. Nonetheless, the output of these algorithms is still only machine-readable.

Whereas TDT extracts important developments from a stream, summarization is the extra mile that transforms topics into a human-readable form. Summaries can take various forms, from timelines to snippets of text, but their objective is to concisely describe the most important information within a document or a corpus [102].

To achieve their goal, summaries should be fluent and legible [102] without overwhelming readers with the copious information that such systems aim to curb in the first place. To this end, existing literature outlines some desirable features of the generated summaries; namely, machine excerpts should be made up of relevant content [7, 103], maximize coverage [7, 102, 104] while minimizing redundancy [7, 102, 104], and be grammatically-correct [7, 102].

Historically, summarization operated within the local context of a single document, reducing it to a concise version. With the internet generating so much data, recent research has shifted its focus towards multi-document summarization. The analogous task creates a short write-up from the individual tidbits in a collection of documents [102, 105, 106].

Multi-document summarization adds another side to the problem because corpora often include overlap, which is practically non-existent in a single document [105, 106, 107, 108]. Furthermore, the variation of sources likely results in structural incoherency, unless ordering is somewhat enforced [102].

Two predominant summarization approaches have emerged in literature - abstractive and extractive summarization. The former creates summaries using the words in the corpus, possibly creating sentences that do not even appear in the original documents. Extractive methods form summaries by picking among the key sentences in documents [63, 102, 109, 110].

Of the two approaches, abstractive summarization is the least popular. One contributing factor is that it encompasses a number of sub-problems. Yao et al. [102], and Erkan and Radev [110] both explain how abstractive summarization can be split into numerous other complex research tracks tasked with understanding the existing content and generating the new summary.

With research progress in these smaller areas still in its infancy, and a limited selection of popular methods, there is little guidance to prospective new studies [102, 110]. In fact, Yao et al.'s survey of existing approaches notes that most summarization systems focus on narrow, domain-specific subjects. Word or sentence graphs are commonly used as the basis of abstractive summarization. Edges connecting words or phrases represent a scoring function that boosts weight when one component follows another [102].

One popular approach, used among others by Nichols et al. [14], Olariu [40], and Mane and Kulkarni [111], is Sharifi et al.'s Phrase Reinforcement Algorithm [112]. Al-

though the authors describe it as an extractive approach in [113], the algorithm is often the starting point for most abstractive approaches [102].

Sharifi et al.'s algorithm uses a directed graph built around a topical keyword and connects adjacent words together. The strength of the directed associations is proportional to the frequency of the two words appearing in the given order in the corpus. The one-sentence summary starts from the topical keyword and works its way back to the beginning, and then to the end of the sentence, retaining the highest-scoring path [112].

Nonetheless, due to the absence of sufficient research, abstractive summarization techniques give way to extractive algorithms. Since humans write the content themselves, extractive summaries ensure, to a certain degree, legibility and correct grammar. Extractive summarization is a three-piece approach whose goal is to extract the sentences that describe a corpus most succinctly.

The first component scores and ranks sentences according to their importance relative to the corpus. The second component - the extractor - uses these scores to choose a set of sentences that minimize redundancy while maximizing coverage and legibility. The optional third component is especially useful when the chosen fragments come from different documents or authors as it reformulates sentences to improve the flow of the summary [102].

Sentence scoring is the necessary preamble to extraction. The majority of TDT systems, which do not focus extensively on summarization, prefer a simple route. Clustering often simplifies the process. MEAD, a summarization system created by Radev et al. back in 2001, is a popular procedure that incorporates cluster centrality when choosing representative content [114]. Others follow in selecting documents or sentences that incorporate important keywords [71, 96, 102, 107, 115].

Some systems exploit the brevity of Twitter to choose entire tweets, rather than split microblogs into individual sentences. For instance, Zubiaga et al.'s topic detection system picks the tweets that contain the most integral event terms [72]. When dealing with more specific domains, some authors favour instead tweets that contain domain-specific discourse, such as numbers or participant names [7, 25, 38].

Others still, like Alonso et al., incorporate Twitter metrics in their sentence scoring mechanism. They consider retweets, for example, as endorsements and score tweets in proportion with their popularity and their authors' following [67, 70].

Another common approach is Carbonell and Goldstein's Maximal Marginal Relevance (MMR) model. This model extracts sentences progressively, choosing those that are most relevant to the query, while simultaneously dissimilar to already-chosen fragments. In case of summarization, which has no query, relevance focuses instead on the key terms in the document [103, 116].

More complex extractive algorithms exist to exploit other research domains' capabilities. Like in abstractive summarization, graphs make significant appearances in sentence extraction as well. One such algorithm, by Wan and Yang, creates an affinity graph of sentences. This mesh structure also computes the worth and novelty of information in each sentence [105].

Two of the most influential document-graph approaches were originally presented by Erkan and Radev in 2004 - LexRank [110] and LexPageRank [117]. One notable introduction in these systems is the concept of prestige, or the importance of nodes based on eigenvector centrality. Erkan and Radev explain how their solutions can isolate sentences that are subsumed by others [110, 117].

LexRank and LexPageRank have been used frequently in the context of microblogs [63, 118, 119]. Once more, systems that deal with tweets, like Shou et al.'s Sumblr, represent each microblog in this graph as a node without splitting it into sentences. Shou et al. later combine the approach with Carbonell and Goldstein's MMR to select a few microblogs and create a summary [63].

Other graph-based methods ingrain the social structure of Twitter within the graph itself. Duan et al.'s method, for example, creates a three-tiered graph connecting tweets, uni-grams and authors together. The algorithm uses this graph to score and rank tweets, before finally using MMR to construct a summary [89]. Celikyilmaz and Hakkani-Tur use a tree-based approach [120].

Most literature considers abstractive and extractive summarization to be apart, but one notable, recent effort by Rudra et al. attempts to join the two. Rudra et al.'s approach identifies a subset of qualitative tweets that describe what is happening and uses them to generate brand new content [7]. Nonetheless, to date, extractive summarization is the predominant choice in Twitter and similar microblogging environments [102, 109].

The concept of readability is important even when it comes to extractive summarization. A lot of effort has gone into identifying quality content [7, 119] and the approaches vary wildly, with some scoring sentences based on their temporal features [121]. The peculiarities of microblogs present their own challenges, like colloquialism [102].

Naturally, multi-document summarization is the de facto approach when operating in social networks like Twitter, where individual messages are already short [63, 109]. Song et al. consider the evolutionary effects of the Twitter timeline. They suggest that summarization algorithms can also set their sights on the temporal nature and network context [104].

Ng et al. consider that as a timeline evolves, so does its language. To reflect this evolution, they present an alternative to MMR [103], which they coin TIMEMMR. Their model goes beyond lexical similarity and considers the publication time of sentences

since temporally-distant documents have different contexts [121].

This problem is similar to that of the TDT track - both tackle the problem of novelty in a corpus. Similarly to McCreadie [122], Chua et al. deal with perplexity [123], and Ren et al. use novelty as a feature [30]. This problem constitutes incremental summarization.

Incremental, or update, summarization assumes that the user already knows what has happened earlier, and that they are only interested in new information [102]. Due to its nature, it is most appropriate in event summarization, as introduced by McCreadie et al. [122] and by Wang et al. [124].

This section concludes a long road that sets off to process and understand human-readable tweets. Summarization is the last stage that serves to transform the machine-readable information of TDT back into a cohesive, informative and non-redundant form that can be understood easily by end-users.

3.5 Summary

Like this chapter, ELD's process takes a representation of a topic and uses Twitter to build knowledge about an event and understand its progression. Naturally, the methodologies presented in this chapter greatly influence the direction that ELD takes. Chapter 4 explains how the outlined literature shapes the design of ELD. We also explain how our approach diverges from other systems to reach the objectives laid out for it.

4

Design

4.1 General Workflow

Literature has examined the basic components of ELD from different facets. In our proposed solution, we consider the virtues and flaws of existing systems to model an event and pursue its development. The point of departure is what it means to follow an event.

Definition 1 on page 2 reduces the complexities of an event to four components - the semantics, the space and the temporal dynamics, and the participants. ELD operates in the context of Twitter, and thus it is safe to ignore the locality of an event. In contrast, the semantics and the temporal backdrop of an event need describing.

The temporal notion describes when the event will exist. Although an event may go on for an indefinite period of time, in this dissertation we focus on events that are bound to rigidly-defined time windows. What we consider to be the most fundamental aspect of an event is its semantic composition, which translates into what ELD would track. This is not dissimilar to what Ratinov et al. describe as a conceptual search, or the idea that people do not look for a particular set of terms, but a richer concept [50].

For many people, a handful of terms, like the names of two teams, gives enough of an understanding because they had spent years absorbing knowledge from the world around them. On the other hand, a machine has no such upbringing.

ELD uses the conceptual search as the foundation on which to build a deeper semantic comprehension of an event. APD is our primary mechanism to nurture this understanding by looking beyond keywords, and instead seeing events as the products of actors interacting with their environments.

The conceptual search terms are the necessary preamble to follow an event, and thus they are required inputs. So is the time window, which cannot be easily extracted automatically [28]. Whether explicitly stated [26] or implied in the workflow [42], existing

systems use these two inputs as the instruments to track an event. We follow this idea in ELD, accepting three inputs - a seed set of keywords and two time window lengths.

The two time windows reflect the system's two roles, as shown in Figure 4.1. First, the understanding phase builds a semantic comprehension that fills in any voids in user knowledge about the subject. The second phase takes this enhanced understanding and uses it to track the event and identify its developments.

The semantic decomposition happens before the event starts. It is desirable for this period to be extensive to capture as much of the domain as possible, and for the ensuing understanding to be reliable. ELD bases its comprehension on a corpus of tweets that it would have gathered using the user's initial seed set of keywords. The resulting dataset serves two purposes.

Firstly, ELD uses this corpus to build an understanding of the domain's vocabulary, inspired by Reed et al.'s TF-ICF scheme to maintain the IDF component of the TF-IDF term-weighting scheme. Reed et al.'s experiments demonstrate that the corpus should share the domain with the application [60]. It is for this reason that we create the Inverse Corpus Frequency (ICF) table using the event's domain, not from a general corpus.

Secondly, the tweets expose semantics that the user might not be expected to describe in the seed set, which is also ELD's principal deviation from literature. Unlike existing systems [26], we make no assumptions about the users and their comprehension of the event's domain. At best, it is unreasonable to expect users to provide a comprehensive description of an event. At worst, users themselves may not have a solid grasp of the domain and are unable to be thorough. This is where APD steps in.

Like Shen et al. [13] and McMinn and Jose [18], with our concept of APD we recognize the importance of participants in understanding the event. Our method uses a combination of what Zingla et al. describe as the local and external sources [81].

The local context is the sample of the event conversations collected in the beginning using the user's seed set. We look for participants in the tweets, but consider them to be candidates to be confirmed by the external source. Like Zingla et al. [81], we choose Wikipedia as the supporting external knowledge base to prune irrelevant candidates.

The corpus explains a domain, but only from the very narrow perspective of Twitter users. Mamo and Azzopardi evoke the definition of important content, finally concluding that like an echo chamber, Twitter distorts the definition to fit the interest of the masses [9]. We similarly assume that tweets do not give the full picture of an event.

In all likelihood, the candidates that are found are the prominent or controversial persons that attract attention. APD overcomes this problem by exploiting Wikipedia's link structure to look for the missing participants. At the end of this process, detailed in

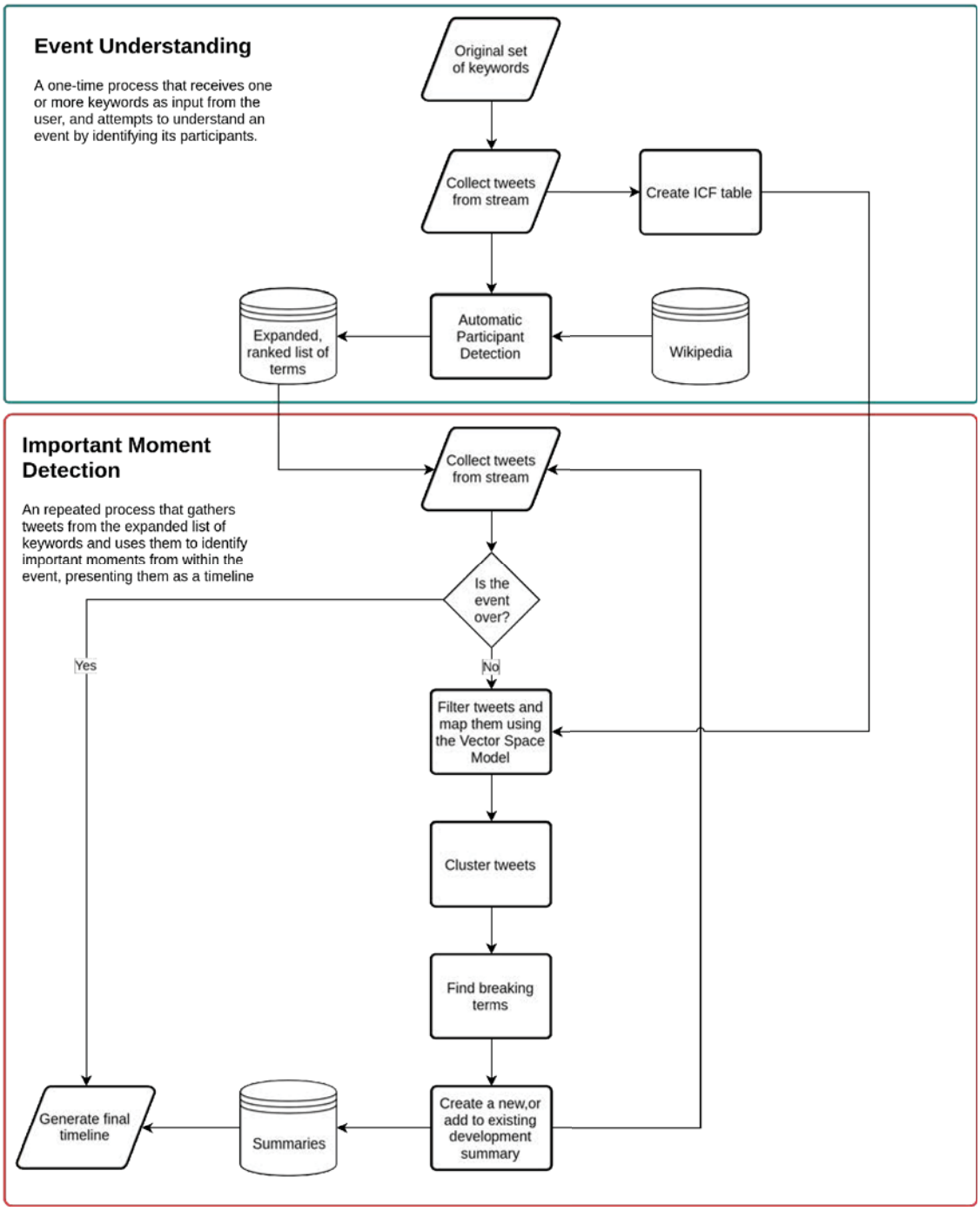

Figure 4.1: ELD's workflow is split into two non-overlapping processes - an understanding phase, and the topic tracking phase.

Section 4.2, lie waiting the individual participants that populate the new, enlarged set of keywords to track the event.

The actual topic tracking stage starts a few minutes before the event. As tweets arrive, ELD uses the ICF table to convert them into vectors in the VSM by using the BOW model. The algorithm calculates the weight $w_{k,t}$ of keyword k in tweet t as follows:

$$w_{k,t} = f_{k,t} \cdot log_{10}(\frac{|C_u|}{|\{d \in C_u : k \in d\}|}) \tag{4.1}$$

$f_{k,t}$ is the number of times that keyword k appears in tweet t - the *TF* component. The logarithm is the *ICF* component. $|C_u|$ is the size of the corpus that ELD collected during the understanding period. $|\{d \in C_u | k \in d\}|$ is the number of tweets in this same corpus in which keyword k appears. We subsequently normalize the document vectors. In the rest of this dissertation, we refer to tweets and documents interchangeably.

By incorporating TF-ICF, we diminish terms that are consistently popular within the domain, instead promoting those that gain traction throughout the event. During the first minutes of tracking, the core system of ELD establishes a baseline of the event's discourse based on this representation.

We construct the initial baseline using Cataldi et al.'s definition of nutrition [73]. We follow in the footsteps of FIRE [9] and unlike Cataldi et al. [73] we exclude user authority from the nutrition calculation. As the event progresses, ELD forgets this baseline, slowly replacing it with a fresher memory of an ever-evolving environment.

This nutrition is the basis upon which topic detection is performed. Our algorithm, detailed in Section 4.3, calculates the nutrition of each term, and saves it after every short time window. Rather than use a sliding window mechanism, these periodic checkpoints serve as approximations, thus detecting developments in real-time.

Following the results of [9], we combine document-pivot and feature-pivot approaches to detect topics with a very fine granularity. Promising groups that emerge from the document-pivot algorithm undergo further processing by the feature-pivot module to determine the validity of topics. The TDT component passes on the detected developments to the summarization algorithm.

Throughout its lifetime, news develops and changes. We observe that this is also true of short-lived news items, such as the rapid developments during an event. To capture this element without delaying reporting, the summarization approach must also be flexible to adapt to the dynamic nature of news.

When called upon, our summarization algorithms generate a summary using all the clusters added to the active topic. In the meantime, all the clusters remain active and

collecting tweets. As a result, the algorithms keep receiving new documents to polish the summary, explaining developments without sacrificing timeliness.

The rest of this chapter explains each component in more detail. Section 4.2 outlines the reasons why we adapt the definition of an event to include participants and formalize APD. In Section 4.3, we present the principles to create a topic tracking system that operates not only in real-time, but also with a fine granularity. Finally in Section 4.4, we introduce our contributions to the field of summarization.

4.2 Automatic Participant Detection

Late in the match Alexandre Lacazette's, clean strike equalized for Arsenal F.C. against Liverpool F.C. in the Emirates Stadium[1]. In this scenario, it is almost unquestionable that Lacazette is a participant, but other aspects are far from clear. Lacking flesh and bones, is the Emirates Stadium a participant? For that matter, are the teams active participants?

In Definition 3 on page 3 we specified that participants may be either animate or inanimate. Within the loose bounds of this definition, in ELD we consider the teams, their players, coaches, and even the stadium to be participants. On the other hand, although integral to football, topical words like *goalkeeper* and *striker* do not make the cut for reasons that we will explain shortly. It is this split between the two sets that symbolizes the two restrictions on participants.

Consider a football match between Olympique Lyonnais and Saint-Étienne. The heated rivalry is normally played on a Sunday evening. During this time window, the football bonanza is alive beyond France. On such evenings, what sets apart this particular match from all the others is that Lyon captain Nabil Fekir is only playing in that one game, and that there is only one Rhône Derby.

Placed in contrast, keywords like *goalkeeper* and *striker* pale in comparison. We see these terms as describing the general domain of football. Conversely, the player names discriminate among singular instances of the class of players. Therefore the first requirement of participants is that they must be discriminative.

Not all discriminative keywords are participants though. In July 2018, France met Croatia in the FIFA World Cup final. The phrase *FIFA World Cup 2018 final* will forever refer to that single stage of a one-time competition. Years from now, France and Croatia will meet again, but on those occasions, the phrase *FIFA World Cup 2018 final* will not be relevant. This characteristic shapes the second restriction - temporal relevance.

[1] `https://web.archive.org/web/20190217101043/https://www.theguardian.com/football/live/2018/nov/03/arsenal-v-liverpool-premier-league-live`, last accessed on February 17, 2019

Query expansion very rarely has any frivolous restrictions, but we introduce them because they are meaningful to our end-goal. A query expansion of the keywords *France* and *Croatia* could include both the terms *football* and *FIFA World Cup 2018 final*. The lax problem definition of query expansion does away with the discriminative and temporal characteristics.

It is because of these two restrictions that we consider APD to be a distinct research area from query expansion, even if it is a specialization of it. Through the discriminative and temporal characteristics, APD alters the direction of the expanded seed set.

We assume that the user's seed set is sufficiently focused. With a specific focus on real-time event tracking, we regard Twitter as a tool that facilitates this integration of restrictions. Already, the temporal relevance is likely ingrained in this stream.

Consider the case where the initial seed set contains only the terms *Croatia* and *France*. If the two were meeting each other in a football match, discussion before the event starts would likely revolve around that match, not the nations. Even if at any point APD captures the phrase *FIFA World Cup 2018 final*, discussion during the match would be more focused on what is happening.

Moreover, it is reasonable to expect that some participants start showing semantic value at that point in time. In such real-time environments, focusing on the discrimination aspect of participation is far more meaningful. We simplify this problem by considering what constitutes participants.

Among the few systems that we encountered with mentions of participants, McMinn and Jose detect entities dynamically from the documents that they receive using NER [18]. Shen et al. focus on sport events, and they too look for proper nouns as indicators of participants [13]. Finally, Corney et al. do not look for participants automatically, but they track football matches using player and team names [15].

What exists in this literature yields an apparent pattern about what a participant is - a named entity. Therefore we make the simplifying assumption that participants are either persons, organizations or locations. When NER has seemingly found its place in similar systems, where does APD fit in?

NER implementations are normally designed for written content that is far more formal than Twitter and its unruly orthography. We consider APD to be ELD's response to the issues that plague Twitter, although NER still has its use in this noisy medium.

The corpus collected during the understanding period revolves around a very narrow concept; usually, a hashtag or a general description of an event suffices. Unlike Shen et al., who provide a more comprehensive tracking set [13], there is little machine-readable knowledge about the event. Notwithstanding the limited scope of the corpus, it represents Twitter's initial knowledge about the event.

Like McMinn and Jose [18], and Shen et al. [13], we start by extracting named entities from the batch of tweets. The orthographic deficiencies of social media present a challenging context to NER, but we rely on the crowd's uncoordinated contributions to overwhelm the noise.

At this stage, the list likely contains several false positives. To this end, like Shen et al. [13], we filter out infrequent participants. What little literature exists in this area halts at this point. It does not consider that, as we discussed in FIRE [9], Twitter talks about the participants that it finds important, ignoring unpopular entities.

In these cases, NER is helpless to detect these participants because they do not exude importance in the pre-event chatter. The corpus' local context is insufficient to resolve the problem because it created the issue itself. Our attention turns instead to Zingla et al.'s external context [81] - Wikipedia.

Like Twitter, Wikipedia is voluminous, updated rapidly, and the fruit of a collaborative effort of thousands of users. However, Wikipedia's structure also complements Twitter. It is organized into articles, each representing a single concept [79]. Commonly, articles are written by several contributors that not only fact-check, but also ensure high quality. These virtues make up for Twitter's faults.

Like Egozi et al. [83], and Gabrilovich and Markovitch [79] before, we consider Wikipedia as an encyclopaedia that can explain candidates in formalisms that Twitter seldom includes. The unique structure of the encyclopedia, accompanied by a robust, free API, allows for candidate resolution.

The process continues where NER left off by looking for the page of each identified candidate, comparing its article with the local context. Candidates without a page, or whose articles are not relevant to the event are filtered out immediately. ELD considers the top remaining entities to be actual participants. At the same time, it maps them to Wikipedia concepts.

Our APD algorithm enters the next phase with a set of entities that were deemed important by Twitter users. Yet Twitter's narrow focus is compounded by bias, leaving a lacklustre set with only a slight improvement over the initial seed. The enlarged set is a form of understanding that APD uses to find those entities that NER could not identify.

If enough information is available in the dataset, a few participants should have been found at this point, and in turn they can serve as examples for entity set expansion. The general concept of APD sets it apart from query expansion, but the environment also thwarts entity set expansion.

Consider the previous example of the football match between Olympique Lyonnais and Saint-Étienne. The participants detected during the first phase would likely contain the two team names, and a few players from one or the two sides. Taken in isolation, the

two teams make up a class of French football teams. The players of each team make up two distinct classes - the players of each team. Clearly, this mixture does not fit entity set expansion's assumption that the examples represent a single class.

On top of confirming NER's results, the first part resolves participants to Wikipedia concepts. The encyclopaedia connects articles together in a network of concepts [47]. Outgoing and incoming links from articles create interconnections that literature exploits [35]. We use the mesh supporting Wikipedia's structure to circumvent the problems when presented with multiple classes.

Starting from the resolved set of Wikipedia concepts, the expansion phase gathers all outgoing links from their respective Wikipedia articles. The algorithm keeps a count of incoming links for each Wikipedia article. To restrain the exponential growth, our APD algorithm retains only the highest-linked pages using a predefined cut-off point.

The strenuous process repeats a second time, but using the enlarged set of Wikipedia concepts as the new seed set. After the article gathering stage concludes for the second time, the algorithm will have collected a large number of new candidates.

Similarly to Wikipedia's conceptual representation of the encyclopaedia and Ferragina et al.'s research [35], our algorithm represents this network of articles using a graph of its own. An edge is created between two candidates if either one links to the other.

However, not all edges are equal. We distinguish between a mention of a concept made in passing and a closely-associated entity. For example, Donald Trump's Wikipedia page mentions the concept *witch hunt*[2]. However, concepts like his political affiliation are more relevant to him.

Therefore we convert Wikipedia articles into vectors in the VSM using the textual representation of their articles, as explained in more detail in Section 5.2. The edge strength between the two Wikipedia pages is calculated using the cosine similarity of the two vectors, retaining only heavily-weighted edges.

Whatever the semantic class of concepts, they come together to form a single graph. What distinguishes the classes is their connectedness. We make the reasonable assumption that if two concepts belong to the same class after this process, then their articles link with each other. This assumption translates well in practice.

Concepts that belong to the same class are often tightly-knit with navigation links to other entities in the same group. This assumption imagines the graph to be made up of communities characterizing the different classes of participants. When our APD implementation dissects this graph, weak links break up to expose the underlying communities. It stands to reason that the larger communities represent the conceptual classes.

[2] `https://web.archive.org/web/20190221194910/https://en.wikipedia.org/wiki/Donald_Trump`, last accessed on February 21, 2019

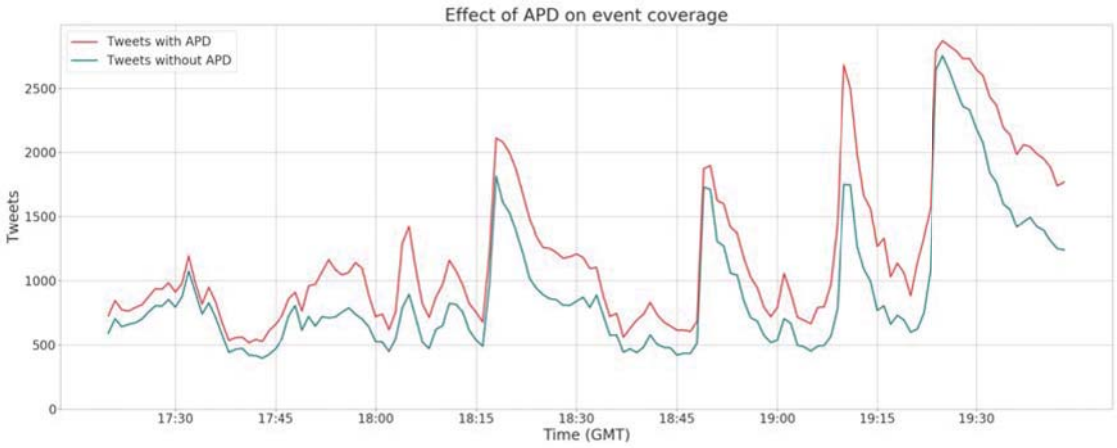

Figure 4.2: APD increases the volume of tweets that are collected during an event.

Our APD methodology strips away isolated weakly-connected nodes and small communities in favour of larger groups. From these sets, the technique picks the concepts that are most related to the local context captured in the understanding phase. The method returns the top entities as participants, joining those captured in the first phase.

ELD adds these participants to the original seed set provided by the user to broaden the search during the event. Other literature showcases more elaborate use cases for this automated methodology, but we leave this for future work. Nevertheless, extending coverage may sound frugal; what are 50,000 additional tweets to an already 100,000-strong corpus?

Figure 4.2 shows the volume of tweets every minute during a football match with and without APD. Certain developments, like goals, alight discussion more than others. In these moments, we hypothesize that a boost in tweets results in earlier reporting. During other periods, APD boosts volume, making it easier for TDT to pick up on the subtle moments.

There is also the issue of spam and noise, so abundant on social media [9, 96]. In ELD, we take a conservative approach to filtering, but it is not uncommon for many tweets to be classified as low-quality or outright spam. Even at its simplest, APD's ability to extend coverage can compensate for this constant bleeding out of content. Later on, the added documents are a richer source of information for summarization.

The enhanced knowledge of the event through its participants in the form of an enlarged corpus is merely the raw material - TDT must fill in the gaps itself. We describe this process in the next section, explaining how our TDT approach harnesses this wealth of information to make sense of an event and its evolution.

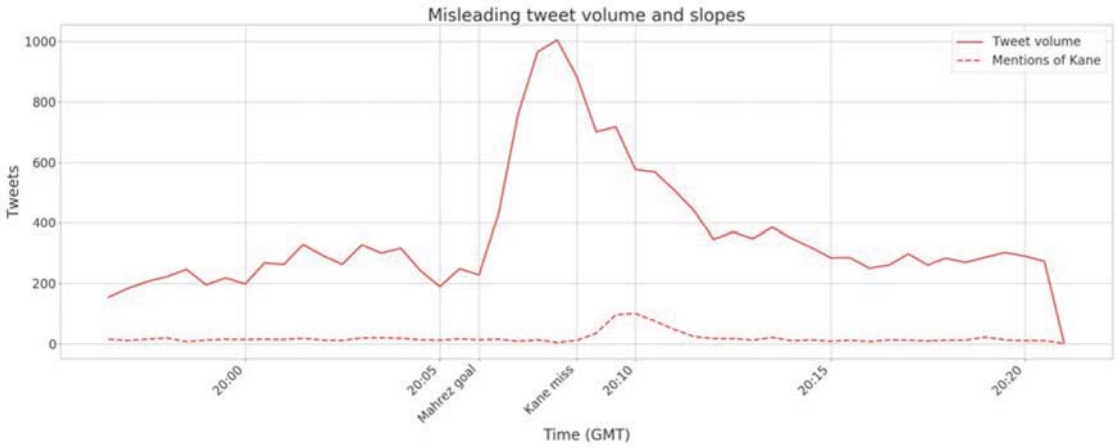

Figure 4.3: An emotional event, like a goal, can dwarf other less emotional moments in close temporal proximity.

4.3 Topic Detection and Tracking

Tottenham Hotspur had scored the second goal against Manchester United. A few minutes later, the Tottenham crowd started jeering opposing manager José Mourinho. But you might not have known it by following the game using many of the existing systems.

When techniques like Zhao et al.'s [12] follow an event through the tweet volume, they rely on developments so momentous that they force a shift in the public discourse. The simplification sounds alluring, but it also glosses over the intricate emotions that give the event its unique personality. Nichols et al.'s focus on volume slopes translated into tangible improvements, but even this is not enough in some cases [14].

Figure 4.3 shows the first minutes of the game between Tottenham Hotspur F.C. and Manchester City F.C. At a glance, it represents a single obviously significant moment - a goal by Riyad Mahrez. What the naked eye misses is a surge of tweets that mention Tottenham's reaction to conceding - a Harry Kane shot that came close to equalizing.

The hundred-tweet increase is buried amid the euphoria of an early goal, and makes no glaring dent in the slope. A multitude of scenarios contribute the same effect - a substitution follows a goal, or a team scores two goals in a matter of minutes. It stands to reason that to describe an event with a finer granularity, so must the algorithm take a more delicate approach.

Yet while the level of detail increases, neither should the technique open up to spam and noise. Mamo and Azzopardi's FIRE satisfies these requirements. FIRE's TDT approach, which we adopt, combines the two distinct families of TDT solutions by embedding a feature-pivot method in a more traditional document-pivot algorithm [9].

The document-pivot approach operates in the global context, consisting of all the documents that are entering the stream. Due to time constraints, we adopt an incremental approach to clustering. Like in popular literature, documents join an existing cluster if their similarity exceeds a certain threshold; otherwise, the algorithm creates a new group [18, 19, 26].

However, the approach does not assume that the clusters represent actual topics [9]. Unlike McMinn and Jose, who finalized topic detection by retaining sizeable clusters as topics [18], FIRE's premise takes a more tolerant approach. It allows smaller clusters to pass the filter only to undergo in-depth verification by the feature-pivot algorithm [9].

The document- and feature-pivot algorithms operate in vastly different environments. Clustering considers all streamed documents, and thus resides in the global context. In contrast, the feature-pivot algorithm considers only the documents within the local context of a cluster. The feature-pivot technique compares the local and global contexts to understand how the importance of each keyword has changed [9].

In ELD we present a new feature-pivot algorithm that harnesses this separation of contexts. We focus our efforts on making the algorithm interpretable, which starts by formalizing the distinction between the two contexts. Literature represents the global context as routine snapshots, whether over volume [14] or individual terms [9, 73]. Existing systems create these snapshots at every time-window around the TDT approach's basic features, like nutrition. Over time, the snapshots represent a stream's progression.

Whereas batch systems use snapshots, the analogous snapshot procedure for real-time applications is to establish a sliding, or dynamic [12], time-window. However, this approach piles on the calculations. Therefore ELD replaces the snapshot technique by the routine checkpoints of batch systems. We refer to snapshots and checkpoints interchangeably.

Like in our previous work [9], we represent feature importance using nutrition, but we base it on the term-weighting scheme alone, setting aside user popularity. We calculate nutrition for keyword k at time window t using the following equation:

$$nutr_{k,t} = \sum_{d \in C_t} w_{k,d} \tag{4.2}$$

The nutrition $nutr_{k,t}$ sums the weights $w_{k,d}$ of the keyword in question, calculated as shown in Equation 4.1 on page 32, over all documents in the snapshot's corpus C_t. Differently from FIRE [9], each of these checkpoints are subsequently rescaled as follows:

$$nutr_{k,t} = \frac{nutr_{k,t}}{\max(nutr_{o,t} | o \in v)} \tag{4.3}$$

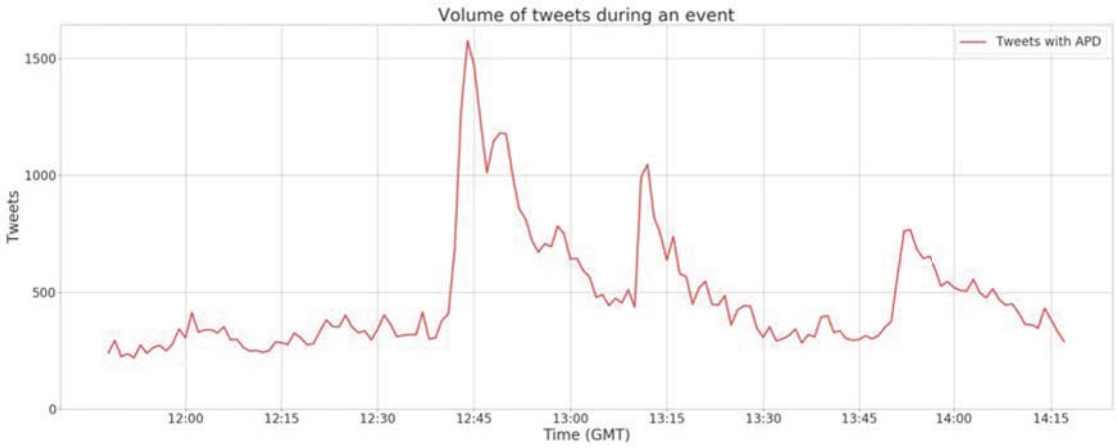

Figure 4.4: The volume of tweets during a football match paints a rough picture of the event's progression.

The objective is to bind the keyword nutritions between 0 and 1, where the highest and lowest values are the most common and least frequent tokens respectively in the snapshot's vocabulary v. As explained later on, this property contributes to the interpretable nature of our feature-pivot TDT approach.

With the knowledge about past conversation habits stored in these checkpoints, the feature-pivot algorithm can validate incoming clusters. Similarly to previous research [12, 14, 18], TDT observes the evolution of an event over snapshots to detect deviations from normal conversation.

The feature-pivot algorithm operates within clusters with the knowledge that their topics were popular enough to be discussed by several people in close temporal proximity. Each cluster's topic revolves around a small set of keywords, which shared enough context to form a single group. Therefore for the topic to be a breaking development, the validation step needs to ensure that the keywords at the core of the cluster should also have been subjected to a burst in usage.

Mamo and Azzopardi approached this validation step using filtering [9]. However, our proposed feature-pivot algorithm ingrains this notion in the burst calculation. The algorithm's validation task is reduced to confirming or rejecting the hypothesis of whether the cluster's distinguishing keywords are valuable in the global context.

Verification starts by converting it into a representation that is comparable with the knowledge accumulated in the historical checkpoints. Unlike in FIRE [9], we create a snapshot from the cluster's local context, which is also rescaled to permit a comparison between the candidate topic and the entire stream. ELD's novel feature-pivot algorithm

can then calculate burst by comparing the local and global contexts.

The central concept of burst is in line of general expectations of a breaking topic [9, 14, 73]. Even without knowing much about the match between Liverpool F.C. and Fulham F.C., the spikes in tweet volume in Figure 4.4 betray the passionate moments. The concept of burst quantifies these spikes by observing sudden changes in the event's landscape.

Like Cataldi et al. [73] and our previous work [9], we model ELD's feature-pivot algorithm to compute the burst for individual keywords. This is the reason why we create checkpoints over all the received keywords. Equation 4.4 presents the calculation of burst $burst_{k,t}$ of term k in time window t, which compares the cluster's local context with the previous s checkpoints in the global context:

$$burst_{k,t} = \frac{\sum_{c=t-s}^{t-1}((nutr_{k,l} - nutr_{k,c}) \cdot \frac{1}{\sqrt{e^{t-c}}})}{\sum_{c=1}^{s} \frac{1}{\sqrt{e^{c}}}} \tag{4.4}$$

We purposefully skip the latest time-window t since it could overlap with documents in the local context. Burst calculations compare the nutrition of each keyword k in the local context - the cluster - $nutr_{k,l}$, with its nutrition in the global context of each past checkpoint c, $nutr_{k,c}$.

Old snapshots get exponentially less importance with the inclusion of a decay factor in the numerator, based on Euler's number e. This allows calculations to consider only a handful of checkpoints s. As a result, the decay factor can help to recognize that two spikes are different, for example when quick-fire goals are scored during a football match. Beyond the basics in the burst calculations, we also felt the need to create an algorithm whose results would be interpretable.

Cataldi et al. [73], and Mamo and Azzopardi [9], present approaches with unbounded burst. This characteristic means that different developments are only understandable to the algorithms. Therefore the denominator of Equation 4.4 places bounds by replicating the decay factor. By restraining the local and global contexts to stand between 0 and 1, the burst calculation inherits the upper bounds. In turn, the burst represents the intensity of a spike.

Figure 4.5 exhibits the changing burst of Manchester City F.C.'s Riyad Mahrez, which soars upon scoring against Tottenham Hotspur F.C. When nutrition plateaus, burst decreases and goes into negative values, reflecting the consistency of nutrition in previous time windows. When the player becomes a protagonist a few minutes later, the burst recovers along with the rising nutrition.

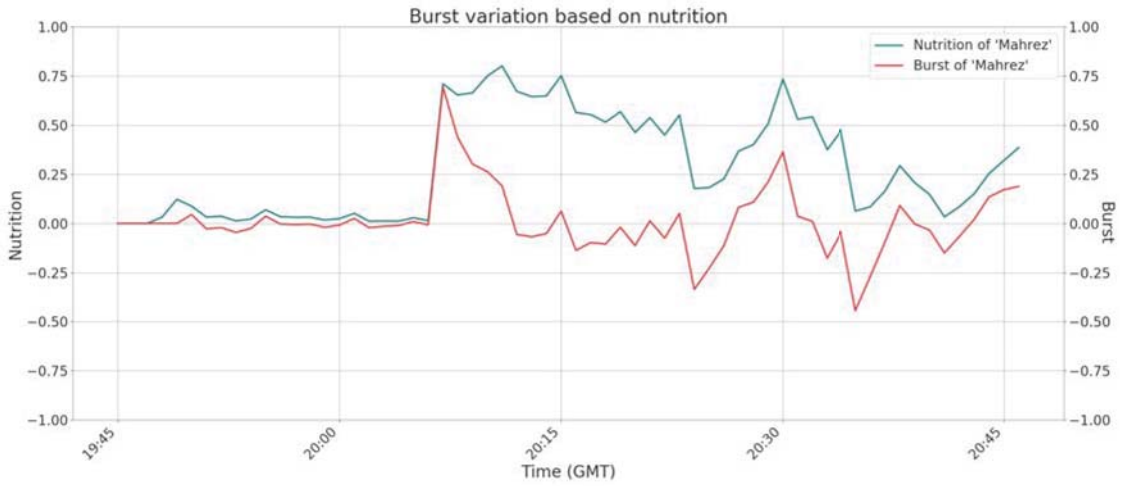

Figure 4.5: Burst responds to the usage pattern of a term, reflecting not only surges, but also downward trends and constancy.

Negativity in the burst values is no anomaly. The internal $nutr_k^l - nutr_{k,c}$ calculation does not only analyse upward trends in keyword usage. If the local context's nutrition of term k is consistently less than that of the global context, then the burst runs into the negative numbers. Effectively, burst is bound between -1 and 1, indicating not only when a development starts, but also when it ends. We do not exploit this property in ELD, but we postulate that in future work, it could be used to improve topic tracking.

With modern computing power, the burst computations are almost negligible. This means that an arbitrary number of past checkpoints may be considered to maximize precision, although this becomes less useful with each added snapshot. One parameter that is more delicate than the number of prior checkpoints is the time-window length.

The snapshots taken at the end of every time-window have an implicit smoothing effect on topic detection, as observed previously [12, 42]. In Figure 4.6, a short window improves timeliness of reporting, but is far too sensitive and impulsive. On the other hand, a longer time window is slow in its reaction. The decision thus comes down not only to the preference for accuracy, but also the expectation over the length of an average development during an event.

Though common individually, chained together in this way, the document-pivot and feature-pivot algorithms keep each other in check and permit a more adventurous approach to TDT. Unlike McMinn and Jose [18], ELD may consider clusters with as few as two documents, achieving the sought fine granularity. The principal trade-off is that this model inherits the parameters of both its components. Most glaringly, the arbitrary thresholds of incremental clustering control fragmentation and specificity [19, 26].

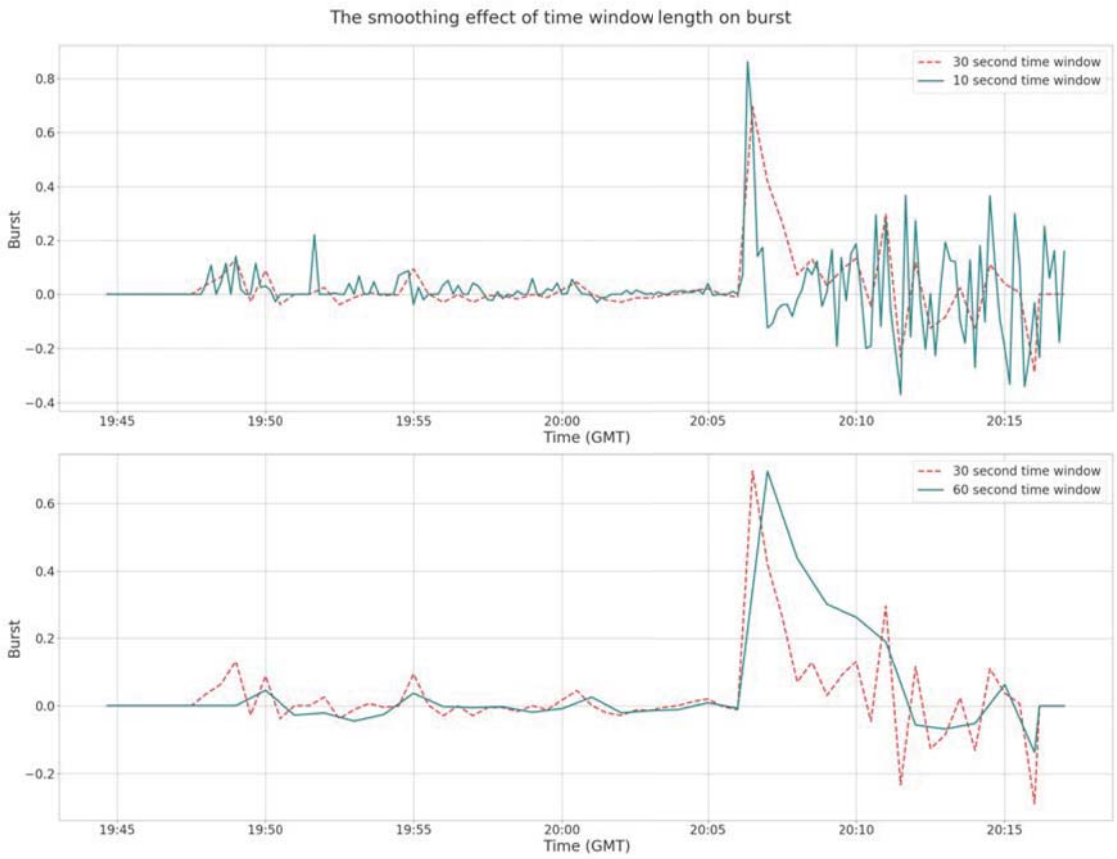

Figure 4.6: The length of the time window affects the sensitivity of the algorithm.

In ELD we do not explore these two issues in depth, but our novel summarization approach considers fragmentation as a natural occurrence in environments that are teeming with emotions. The document-pivot approach described above provides the documents that describe a development through clustering, while the feature-pivot algorithm extracts the topical keywords. We pair up TDT and summarization, guiding our techniques to capture the most important facets of a moment, as presented next.

4.4 Summarization

A development is seen through countless eyes and re-told by as many voices. However simple the moment is, the language used to report it can vary wildly. While humans create this lexical diversity, machines struggle to catch up.

Pulled from one side towards efficiency to perform in real-time, and from the other

side to understand semantics, incremental clustering tries to balance between the two extremes. When it fails to reach the utopia of documents segregated perfectly according to semantics, it shatters developments into fragments.

Historically, fragmentation has been seen as the bane of incremental clustering [19, 26]. Fragmentation fails to reconcile all pieces to form one whole story, but we take a more optimistic point of view - it splits a story into different perspectives.

The summarization module takes in sets of documents put together by the incremental clustering algorithm and the keywords that drive those developments. These two inputs serve as the foundation for summarization, whose end-goal is to create individual summaries that when chained together form a timeline.

Literature evokes two forks in summarization solutions, as outlined in Section 3.4. Abstractive summarization relies on copious and informative documents - something that fine granularity does not contribute. This constrains ELD to an extractive approach. We suggest that by chaining tweets together, we can create a narrative that describes developments. To this end, we make one fundamental assumption.

In our summarization approaches, we assume that clusters that emerged as developments in close temporal proximity all describe the same moment, similarly to Shen et al. [13]. If a summary is inactive for a long time, or an incoming cluster is wildly different from the current one, then ELD considers the development concluded.

The assumption holds under normal circumstances. It is uncommon for multiple developments to occur in the same minute. Moreover, we reason that even if two developments occur in a short timespan, there is little harm done in combining them. For example, if a player fouls an opponent and gives away a penalty, it would make sense for the kick's outcome to be bundled with the initial decision.

By making this assumption, we present two novel algorithms that take the fragmented pieces of a development and combine them into a single summary. We see this general approach as a route to maximizing coverage while simultaneously reducing redundancy - two points that other authors raised before us [7, 102, 104].

The theoretical goal of clustering - that of collecting together semantically-similar documents - captures relevance. As explained in this section, we also work redundancy minimization into our algorithms, while keeping in mind literature's warnings that emotions risk compromising coherency and grammatical correctness [16, 17, 29, 40].

Our first of two approaches is the Fragmented MMR (FMMR), based on Carbonell and Goldstein's popular MMR [103, 116]. ELD's clustering approach in TDT supplies the documents that are relevant to a development. Furthermore, the feature-pivot algorithm yields the terms that describe a development and their intensities. These key-

words are most relevant to the summary that the algorithm needs to create, in similar fashion to Sharifi et al.'s topical keywords [113].

For this reason, we do not create a query based on the core components of all the documents, as imagined by Goldstein and Carbonell [116]. Instead, the query revolves around the breaking terms, thereby ensuring that the summary captures the essence of the development. We weight the breaking terms according to their burst, in this way favouring keywords that emerged more intensely than others.

Moreover, in ELD we promote terms from popular clusters, as indicated by their sizes. We create a query from each group and add it to a new cluster whose normalized centroid constitutes the keywords weighted by importance in the topic. In the rest of the FMMR approach, we input the query and a selection of documents from all clusters into Carbonell and Goldstein's MMR [103, 116], presented in Equation 4.5:

$$MMR = \underset{D_i \in R \setminus S}{\operatorname{argmax}} [\lambda(Sim_1(D_i, Q)) - (1 - \lambda) \max_{D_j \in S} Sim_2(D_i, D_j)] \tag{4.5}$$

The MMR repeatedly picks among all documents R that are not already part of the summary - the set of documents S. The algorithm gives preference to those which are simultaneously similar to breaking terms - the query Q - and dissimilar to already-picked tweets. Gradually, it creates the summary set of documents S [103, 116]. ELD factors in a quality score and picks three tweets at most, or stops when the MMR score of the next-best document is not positive.

Our second approach takes a different view of the collection of documents that make up a development. Unlike TDT, the summarization algorithms are neither tightly constrained by time, nor shackled by overwhelming volume. Therefore the Document Graph Summarizer (DGS) considers the corpus of documents anew, and tries to split it into new facets.

Similarly to Erkan and Radev [110, 117], and Wan and Yang [105], DGS transforms the clusters' documents into a graph. We follow existing methods in exploiting Twitter's brevity, creating a graph made up of entire tweets, rather than sentences [9, 25, 38, 70, 72]. The edges connecting two microblogs is proportional to their similarity. As a result, this operation connects documents that clustering could not reconcile.

From this graph, one or more communities are likely to arise, as shown in Figure 4.7. One is expected to surface more prominently than others, describing the essence of the development, while smaller groups describe secondary angles. We identify these structures using community detection, retaining only sizeable groups. The last step is to extract one document from each community that best describes the facet.

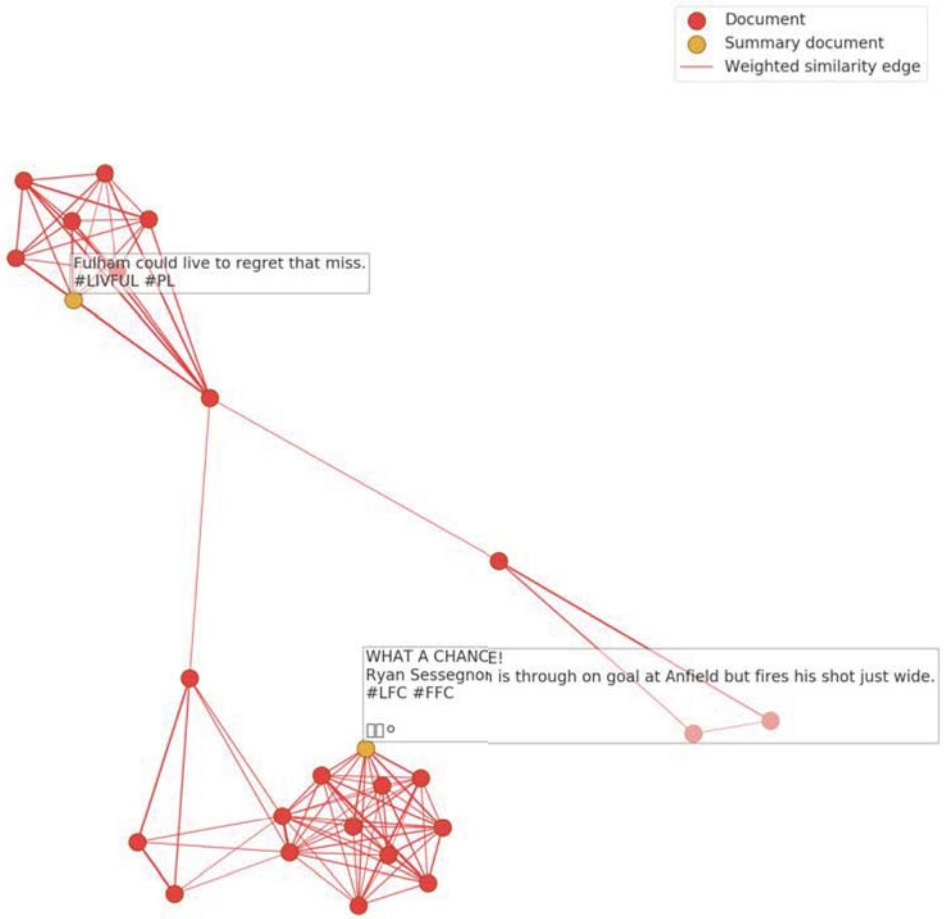

Figure 4.7: The facets of a development cluster together in DGS's underlying graph.

We base this choice on the concept of centrality, previously adopted by Erkan and Radev [110, 117] and Marujo et al. [125]. Freeman describes central vertices as those nodes that best allow the flow of information [126]. Thus, we assume that the central documents in the communities capture the spirit of each group.

Like Erkan and Radev [110, 117], we opt for eigenvector centrality to find central nodes, but we apply it for each community. The communities are likely to be tightly-knit, as shown in Figure 4.7. Thus, DGS combines centrality with the microblog's quality and its relevance to the development.

The former score promotes coherency by analyzing lexical features, as explained in more detail in Section 5.4. We quantify relevance using the same query that was used in FMMR. Chained together, the highest scoring documents from each community combine to create a comprehensive view of developments.

The specifics of the environment in which APD operates give an additional responsibility to the summarization module. The granularity of the TDT task and the prevalence of retweets give ELD an added sensitivity to repeated moments. For these reasons, the summarization algorithms must be wary of capturing old developments and guard against repeating unnecessary information - the concept of update summarization.

Summarization shares this challenge with the TDT component. Topic tracking is implicit in the document-feature pivot algorithm, but the lifetime of clusters eventually expires. From a long-term perspective, the responsibility of identifying repeated updates falls instead to the summarization algorithms. The novelty checks described in Section 5.4 compare new developments with old topics before accepting them.

Another characteristic of these summarization algorithms arises from ELD's real-time incremental setting. As clusters form, they are regularly checked for any breaking terms that they may harbour. These checks halt once a cluster is deemed to be emerging, but it remains active and accruing microblogs.

This property of clusters means that when the TDT component detects a new development, the summarization algorithms can create a summary right away and add it to the timeline. As the fervour dies down and more tweets are posted, the algorithms get a wider choice of documents, transforming a machine-readable representation of topics into a human-readable summary.

4.5 Summary

From a handful of keywords, ELD moves on to an idea of an event that considers the participants that drive its developments. As these actors influence the ongoing event, the system listens to their actions through Twitter's re-telling. The workflow concludes with ELD taking breaking developments and summarizing them in Twitter's own terms. The next chapter explores the practical considerations of the individual components and how when connected, they lead to one coherent system.

5

Implementation

5.1 Architecture

The previous chapter detailed a workflow fixated on delivering a real-time solution to the TDT problem. The design spells the direction of ELD to ensure the timeliness of the approach. The implementation takes the broad theoretical basis of the design and considers its minute details to create a working artefact.

ELD's design choices make for a very lightweight solution; the system requires only sufficient RAM to handle the incoming voluminous tweets, store historical nutrition checkpoints and keep track of active clusters. We use a two-core CPU to ensure that we miss no tweets, as explained further down.

Beyond the architecture, performance depends on the design choices explained in the previous chapter. ELD is programmed in Python 3 and interfaces with Twitter using the Tweepy library[1]. We tested the described algorithms and supporting structures using unit tests and experimentation during the implementation.

At its core, the system operates using this minimal set of tools, and the user-provided seed set of keywords to describe the event and two time window lengths. During the understanding period, Tweepy filters the Twitter stream. Any tweets that are in English and which, in some way, mention at least one of the keywords in the seed set are added to a queue. The collected corpus defines the event's domain from Twitter's perspective.

The first intended use of this dataset is to create an ICF table. Later on the crux of the term-weighting scheme, the ICF uses the exact same tokenizer that ELD uses to tokenize tweets during the actual event processing. In this way, all microblogs share the same vocabulary.

[1] `https://web.archive.org/web/20181024111635/https://github.com/tweepy/tweepy/`, last accessed on March 2, 2019

Throughout the entire workflow, unless otherwise specified, we adopt a uniform document representation. Like many Information Retrieval (IR) applications, we use a BOW approach to represent documents in the vector space. We perform various operations that are common in literature to transform textual documents into vectors.

We opt for a Porter Stemmer [51] over lemmatization due to the performance requirements of ELD. ELD uses Python's Natural Language Toolkit (NLTK) library[2] to stem tokens and to remove stopwords. Furthermore, we add extra preprocessing steps based on Twitter's language to reduce the dimensionality of the VSM [34]. For example, we consider user mentions (such as *@NicholasMamo*) and URLs to be uninformative, and thus exclude them from the VSM.

During data collection and experimentation, we also observed two types of behaviour when tweeting during live events, especially football matches. The first characteristic concerns a popular Twitter convention - the use of hashtags, which often include named entities chained together. We split them based on capitalization, transforming hashtags like *#ManchesterUnited* into *Manchester United*. In this way, ELD splits these kinds of hashtags into normal tokens – *Manchester* and *United*.

The second behaviour is prevalent during emotional spurts, especially in sports events. In these cases, it is common to see words elongated by repeating characters – a makeshift way to convey sentiment. We reduce letters that are repeated more than twice to a single occurrence. This step nullifies the emotional effect on words like *goooooaaaal* by transforming them into a simpler token – *goal*.

ELD stores the ICF table in a Python associative array, or a dictionary, to exploit the underlying hashing for rapid look-ups. The ICF table is constructed using Equation 5.1, associating each token k with the number of documents in the understanding corpus C_u that contain it. A special key stores the total number of documents considered. Later on, it is used to create vectors from tweets, as shown in Equation 4.1 on page 32.

$$ICF_k = |\{d \in C_u | k \in d\}| \tag{5.1}$$

Although the ICF table is crucial in the TDT component to calculate, among other things, the nutrition on which burst is based, the APD component does not use it. We discuss APD in more detail in Section 5.2, but it is worth noting that the term-weighting scheme used to detect participants uses a different table.

APD looks for terms that distinguish themselves within the event's domain, and thus it measures keyword significance against the general discourse. To this end, we

[2] `https://web.archive.org/web/20181024111714/https://www.nltk.org/`, last accessed on March 2, 2019

use Go et al.'s *sentiment140* dataset[3] - a 1.6 million-tweet corpus with tweets from various domains [34]. ELD uses this dataset to create a general IDF table to elevate domain keywords that appear commonly in the event's stream, but seldom outside of it. Simultaneously, it dims the value of terms that are also common in general discourse.

By the time APD concludes, the small seed set that described the event expands to include participants. The data collection task restarts in identical fashion, using Tweepy to interface with the Twitter API. ELD listens to mentions of the keywords in the original seed set and of the detected participants for the duration of the event.

Although ELD is a real-time system, the processing itself takes some time. The TDT consumer is a cyclical process that gathers any newly-collected tweets and processes them. ELD first converts microblogs into tokenized documents. The TDT component clusters these documents and examines any groups that hint at breaking developments. Emerging moments are later added to the timeline.

Clearly, emotional situations risk hogging the CPU with lengthy processing. Therefore ELD forks into two processes during the main event execution cycle. ELD uses Python's *multiprocessing* library[4] to create these processes. One process collects tweets, and the other follows the TDT procedure to analyse microblogs. Regardless of the TDT's load, microblog collection can resume unperturbed on the other core, with the two processes communicating using a shared queue structure.

ELD filters out noisy and spam tweets to facilitate the TDT task. We base some of the criteria on FIRE's [9]. For instance, we exclude non-English tweets and microblogs with more than two hashtags. Like in FIRE [9], we remove tweets by users who never liked another microblog or who have no more than one follower for every thousand tweets that they published.

ELD also introduces two new filters. We observe that users who have an empty biography are usually career spammers, rendering their tweets harmful. Moreover, we reason that tweets with more than a single URL are premeditated and thus do not discuss current developments. We remove these microblogs.

The tweets that make it through the filters fuel the TDT task, and eventually the summarization algorithms. However, when ELD represents tweets as documents in the vector space, the tokens lose a lot of the context. Without stopwords, word order and all the other nuances of language, these documents lose semantic value. Therefore ELD saves the original tweet and the full text alongside the document.

[3] `https://web.archive.org/web/20181024111745/https://www.kaggle.com/kazanova/sentiment140/`, last accessed on March 2, 2019

[4] `https://web.archive.org/web/20181024111838/https://docs.python.org/3.4/library/multiprocessing.html`, last accessed on March 2, 2019

The rest of this chapter delves into more detail about the individual responsibilities of the three main components that make up ELD. Section 5.2 describes our modular approach to detect participants. In Section 5.3, we explain how the checkpoint procedure contributes to extract salient moments. We conclude the implementation with an in-depth explanation of summarization and how ELD maximizes the information of its summaries in Section 5.4.

5.2 Automatic Participant Detection

The general concept of APD is largely uncharted in literature; to the best of our knowledge, no similar systems have been proposed. Therefore we use this section to present a modular approach with six components, with the corresponding pseudocode in Appendix A.1.

APD commences after the understanding period has concluded. Each tweet is converted into a document that is made up of tokens and the original text. APD uses the dataset that ELD collected using the user's seed set of keywords. We refer to this corpus as the local context, which is the basis for the first module.

The first component extracts all candidate participants, no matter how unlikely they are to affect the proceedings. Secondly, APD assigns each of these potential actors a score to represent their worth in the local context. The third module simply selects the best scored candidates and performs simple pre-processing.

The fourth stage resolves all candidates to an alternative, semantic representation, considering them to be participants. The fifth stage deems these participants as examples of various entity classes and extrapolates them to create a larger participant set. Post-processing concludes the workflow, adapting the representation of these entities to the event's domain. The rest of this section explores each of these six phases of APD.

5.2.1 Extraction

Extraction's results intertwine with the entire process, outlining the main actors elected by the Twitter user base, and later on influencing the extrapolated participants. In Section 4.2, we assumed that named entities are the actors that drive an event. With this in mind, candidate extraction is as simple as identifying the named entities in the microblogs.

We use NLTK for NER, but like other natural language libraries, it struggles with Twitter's erratic orthography. Mentions, hashtags, abbreviations and other parts of speech in tweets are born out of Twitter's constraints. However, the fact that the un-

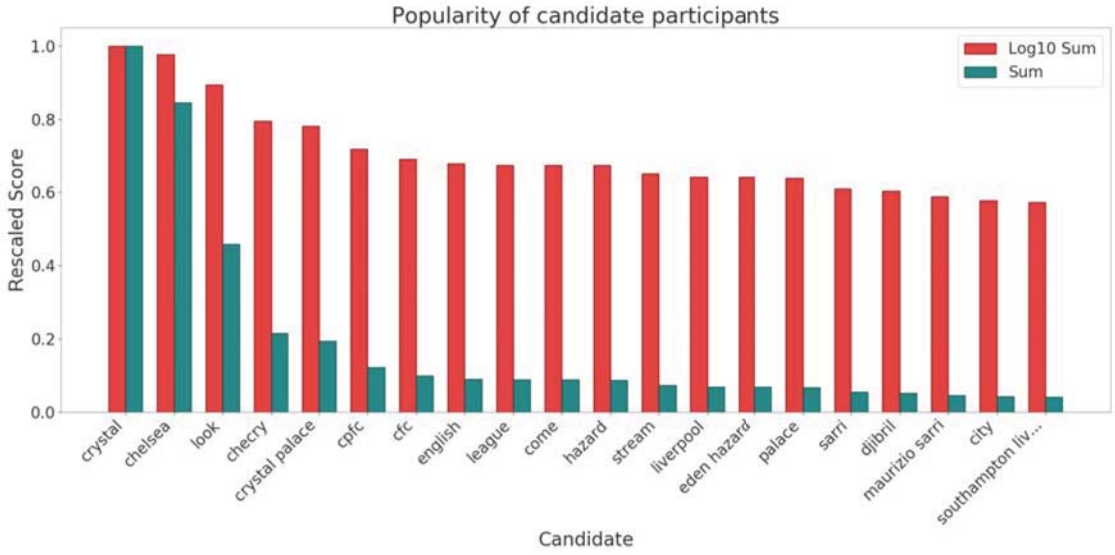

Figure 5.1: Twitter conversations often exhibit a clear bias towards certain candidates.

derstanding phase happens before any emotional developments occur diminishes the impact of these conversation habits.

All the same, the extractor cleans microblogs as much as possible, including by removing mentions and splitting hashtags on capitalization. Subsequently, it passes on these documents to NLTK to extract candidates. By the end of the process, microblogs become a bag-of-candidates representation, where each candidate is a named entity, concluding the extractor process.

5.2.2 Scoring

The scorer receives documents described only by the candidates that they contain to build an overall ranking of candidates. While APD relies on user contributions to understand the domain, Twitter itself is inherently flawed by humans' predisposition to bias.

Figure 5.1 shows how the initial distribution for the game between Chelsea F.C. and Crystal Palace F.C. revolves around the two teams. This is predominantly a reflection of the tracking terms - the team names. However, bias manifests itself even among the individual participants. For example, star strikers, like Eden Hazard, attract much more attention than goalkeepers due to their ability to change games.

The polarizing popularity among the classes of entities does not tell much of a story. At one point, additional mentions of the same candidate do not contribute to APD; an

increase of participant references from 100 to 110 would not change our opinion about their importance, but an equivalent difference between 5 and 15 mentions is telling.

$$score_c = \log_{10} \sum_{d \in C} f_{c,d} \tag{5.2}$$

In these cases, simple summations would not be useful in separating common candidates from uncommon ones. It would only serve to widen the divide and lump relevant candidates with irrelevant ones. Using Equation 5.2, we count the number of times that a candidate c appears in each document - $f_{c,d}$ - and sum these frequencies across all documents in corpus C. Then, we take the sum's logarithm, toning down vast mismatches. The final scores are rescaled between 0 and 1.

5.2.3 Filtering

The simple filtering stage is primarily aimed at retaining the best-scoring candidates. The participant detector retains only those candidates whose score exceeds a pre-defined threshold. Those that remain are likely to be valid candidates by virtue of the large corpus that is normally collected.

Before retaining the top candidates we assume that if one candidate subsumes the other, the longer version may be retained. For example, two of the candidates could be the named entities *Umtiti* and *Samuel Umtiti*. This step merges the shorter version and its score into the longer name. As a result, candidates are less ambiguous. The top-scoring k candidates become the resolver's input.

5.2.4 Candidate Resolution

Candidate resolution represents the shift from the local to the external context, providing a better understanding of the event in general. The resolver verifies the validity of the candidates, transforming those that pass the test into full-fledged participants that ELD tracks for the duration of the event.

We use the MediaWiki API[5] to resolve the named entities, or candidate participants, to Wikipedia concepts. The API provides the option of fetching the concept directly through the title, but this route poses two problems.

Firstly, surnames are shared by many people, making them ambiguous. Secondly, some named entities may be incorrect. For example, *Arsenal F.C.* supporters refer to the club simply as *Arsenal*. However, this word has a different meaning altogether, which

[5] `https://web.archive.org/web/20181024111939/https://www.mediawiki.org/wiki/API:Main_page`, last accessed on March 2, 2019

is distinct from Wikipedia's *Arsenal F.C.* concept. To ensure accuracy, we opt for a more general search.

Since we focus on the most mentioned entities, they are likely to be popular enough to appear in the highest-ranked pages. We look up entities using the MediaWiki API's search functionality, retaining the top five results. From these hits, the resolver needs to identify the one that is most likely to be the participant described by the candidate.

We approach this problem by considering that the participant should share a domain with the event. The initial understanding corpus contains the event's domain, which we represent as a single normalized vector v. The weight of dimension k in this document is the sum of the dimension across all vectors in the understanding corpus C_u:

$$v_k = \sum_{d \in C_u} d_k \tag{5.3}$$

A candidate's domain is more complicated because Wikipedia articles do not only contain the concept's present state. For example, Donald Trump's Wikipedia page does not only describe him as the President of the USA, but lists his biography.

The article's introduction is a slight improvement as it captures the essential, but it may still contain tangential details. We observe that the first sentence of the article is made up of the fundamental facts. More importantly, it is an account that focuses on the concept's present state.

We consider the first sentence to be a concentrated summary of the candidate. Since we are only interested in the candidate's domain to verify its relevance, we remove any mentions of the candidate. We also exclude the date of birth because it is a personal detail. We tokenize the processed sentence similarly to before.

The algorithm uses cosine similarity to compare the page's content with the domain of the event. We choose the concept with the highest similarity to act as the participant if it exceeds a specified threshold; otherwise, we simply ignore the candidate. This selection of concepts is the input to the extrapolator.

5.2.5 Extrapolation

With the knowledge that the accepted participants are also Wikipedia concepts, extrapolation looks for entities mentioned by the new seed set. We keep looking in Wikipedia, which means that many of the principles that governed the candidate resolution process carry over to extrapolation.

The extrapolator recursively fetches links twice, translating into extensive use of the MediaWiki API's links endpoint. The outgoing links grow exponentially from one

iteration to the next. To improve efficiency, we discard pages that are unpopular after the first iteration.

We count the number of incoming links of each article and retain the 100 Wikipedia concepts with the most references. In the second iteration, we fetch the outgoing concepts from these pages and again prune pages based on their popularity. This time, the stopping condition is when the popularity threshold admits fewer than 1000 pages.

All of the returned articles populate a local NetworkX[6] graph that is similar to Wikipedia's own structure. The articles become nodes, connected with undirected edges if at least one of them links to the other concept.

We create stronger edges between pages if they are highly similar, based on the cosine similarity of their respective first sentences. The underlying shape of the graph is an interconnected, almost chaotic mesh of concepts, but the structure holds a wealth of information.

To make this structure more apparent, we exclude weak links. Edges from the first iteration need to have a non-zero similarity, whereas links from the second iteration require an empirically-set similarity of more than 0.5. We use NetworkX's Girvan-Newman algorithm to partition the resulting graph, in this way breaking it up into classes of concepts.

We consider communities with at least 4 nodes to be semantic concepts, treating the inundation of candidates similarly to the rest of the APD process. Like candidate extraction and resolution, extrapolation attempts to find the intersection where the local and external contexts meet.

We follow Gabrilovich and Markovich's approach, excluding candidates that contain a year in their title [79] since they usually represent historic events. The k concepts whose first sentences are most similar to the local context and which exceed the user-provided threshold are the extrapolated participants.

5.2.6 Postprocessing

The last stage embraces the conversation behaviour of Twitter by post-processing the participants one last time to exploit their potential when extending coverage. When conversing, people assume that those who are listening have domain knowledge. Twitter users do not talk with glee about *Corentin Tolisso* scoring, but instead celebrate *Tolisso*. The surnames on the back of players' kits become the names on people's tongues.

[6] `https://web.archive.org/web/20181024112011/https://networkx.github.io/`, last accessed on March 2, 2019

We use Wikipedia's adherence to syntax in the first sentence, which includes the date of birth of players, to detect human participants. Mimicking human behaviour, the algorithm uses just the surname to track the participant. The decision increases the number of collected tweets, but it also introduces some risks.

For various reasons, players like *Memphis Depay*, *Kepa Arrizabalaga* and *Rafael da Silva* are best known by their first names. Others are known by a nickname, at times used almost exclusively - *Javier Hernández* is commonly known as *Chicharito* - and on other times sporadically. Others still are known by virtue of the uniqueness of their profession, such as *President of the United States*. However, these are exceptions.

A more serious problem is the fact that this technique introduces ambiguity. For example, *Joe Gomez*'s name would be reduced to just *Gomez*, which also happens to be the surname of *Selena Gomez*. Therefore tracking Joe Gomez by his surname would also follow Selena Gomez. We tolerate this ambiguity since it is unlikely that multiple people with the same surname become topics of conversation during the event. However, we do make an exception when the last name is in any language's vocabulary.

For example, *Martin Terrier*'s surname refers to a breed of dogs. Listening to his last name in the stream would introduce a lot of noise to the stream. Instead, ELD tracks his full name. Finally, we remove accents from participants since many Twitter users do not write them.

Due to APD being in its infancy, any resolved or extrapolated team names are left in their full form - *Chelsea F.C.* is retained instead of *Chelsea*. In this way, incorrect team names do not interfere as much in the stream. We assume that colloquial references to the event's teams are given in the seed set.

The resolved and extrapolated participants become the newly-recognized instruments of discussion. ELD automatically adds them to the user's seed set and tracks them for the duration of the event. The added microblogs become fuel for the core processing of ELD, starting with TDT.

5.3 Topic Detection and Tracking

The numbers climb from 30,000 messages to 100,000 and beyond for popular events. The volume scales, but the problem remains constant. Where do you draw the line? Which messages discuss a development? ELD's TDT approach, whose pseudocode is presented in Appendix A.2, answers these questions.

Evaluating individual documents on their own merit is impractical in many events, and infeasible in real-time systems. More practically, if a development is noteworthy,

the audience would discuss it widely. Therefore ELD shifts its focus towards analysing groups of documents.

We choose a method that is popular in such scenarios to group together documents with lexical overlap - the incremental clustering algorithm. Simple in nature, the approach adds a document to a cluster if they are sufficiently similar, as determined by an empirical threshold. If the document is dissimilar from anything else that came before it, a new cluster is created.

The technique creates clusters that revolve around keywords with a time complexity that grows proportionally with the number of clusters in activity. Although it is simpler than other clustering algorithms, its complexity still needs controlling. To this end, incremental clustering depends on two parameters - the threshold for new document inclusion and the freeze period.

Since ELD, like FIRE, binds together two TDT approaches, the threshold needs to be carefully fine-tuned as it dictates the rest of the process. A low threshold turns clusters into microblog 'blackholes', absorbing any document that is remotely similar. A high threshold is desirable because it contributes to specificity, but it fragments topics to such a fine point that they could never be considered as potential developments.

The freeze period is less sensitive because it can vary based on the volatility of an event if backed up by architectural capabilities. In fast-paced events, moments fade quickly as newer developments take over. Analogously, the clusters that represent these moments become irrelevant with time. Each old cluster spells more comparisons and an increasing possibility of combining different developments that share topical keywords.

For this reason, we freeze inactive clusters out of circulation, in similar vein to what Azzopardi et al. did [98]. The only difference is that we base the freeze period based on time, like Ozdikis et al. [17], rather than the number of received documents. In this way, the time complexity sways to the volume of incoming microblogs.

In practice, a longer freeze period may be desirable to collate enough documents in one group to create a tentative development. Once frozen, groups serve no purpose unless they constitute an actual development, except to hog RAM. Therefore we unlink all frozen clusters from memory, unless they represent developments.

At this point, the clusters are unconfirmed developments because ELD's feature-pivot algorithm still needs to verify these candidate topics. Twitter gives way to crowd-sourced journalism, with all the advantages, but also disadvantages of such an open model, including spam and noise.

The entire purpose of TDT is defeated if detail is only interspersed among noise. Before answering the question of what constitutes a real development, we filter out

the speculative content. Although single documents can be newsworthy, ELD seeks confirmation by necessitating that developments are discussed by multiple people.

Following this logic, singleton clusters are immediately ignored. Beyond this restriction, the minimum size can be as low as two documents, although the filtering rules come down stricter on small clusters. The concept of newsworthiness is inviting to another set of filtering rules.

When the users themselves become the purveyors of news, the intrinsic worth of the item can be approximated from the size of the crowd that is creating the information, not just relaying it. Like FIRE [9], ELD double-checks developments by capping the maximum intra-similarity of clusters - the average cosine similarity of each document with its cluster's centroid. If people are regurgitating information, such as by retweeting, the development would have a high intra-similarity, indicating that the moment is not eliciting enough varied conversations. Therefore the cluster is not considered.

Once a development happens, there is little time to ponder - impulse kicks in and people talk about it. However, career spammers write regularly with total disregard to proceedings while including spam links. During the implementation, we observed that these microblogs tend to cluster together. Therefore in ELD we count the number of external or media links in each cluster across all of its documents. If the average is greater than one, the algorithm rejects the cluster. The cluster is also rejected if more than half the messages are replies.

The clusters that remain after filtering are potential new developments. The TDT algorithm routinely polls clusters until they are found to contain breaking news or they are frozen. Intuitively, developments represent an evolution that changes the state of an event, and this is what the feature-pivot algorithm seeks.

The build-up to the feature-pivot technique happens before clustering starts. Microblogs stream into a buffer which ELD clears routinely to create global checkpoints. Seen next to each other, these routine snapshots record the progression of the event in terms of individual keywords, and they play a pivotal role in detecting evolutions within it.

ELD stores checkpoints in a nutrition store. The associative array relates the snapshot's timestamp with another Python dictionary of nutritions for all keywords observed during the time window that it covers. In order to make all checkpoints comparable, the TDT algorithm rescales the values of each snapshot between 0 and 1.

The decay rate employed in the feature-pivot approach reduces the relevance of old checkpoints so much so that they can be safely removed alongside their memory footprint. However, the timespan covered by each checkpoint is a compromise between timeliness and sensitivity, as shown in Figure 4.6 on page 43.

Nonetheless, we note that the length of time windows is reminiscent of the assumption of development continuity of summaries explained in Section 4.4. The length of checkpoints should be comparable with that of an average development. If this assumption holds true, comparisons with previous snapshots show a divergence from old developments to mark new progressions.

Once ELD accumulates enough checkpoints, the feature-pivot algorithm can start analysing clusters. The embedded TDT technique puts clusters in the context of an evolving development. The algorithm calculates the nutrition of each term in the cluster's documents to create a checkpoint for the local context.

Then, it rescales the cluster's snapshot between 0 and 1, as shown in Equation 4.3 on page 39 and calculates the burst of all of its terms. Combined with Equation 4.4 on page 41, the previous rescaling step binds the burst of keywords between -1 and 1.

Since the highest nutrition value in a cluster is bound to be 1, the feature-pivot algorithm tends to assign the most commonly-used keyword a high burst value. A notable exception is when the topic emerged very recently with quasi-identical terms, which tones down the burst. These two properties reduce the problem to whether the core keywords of a candidate development are emerging, and if so, to what extent.

Nonetheless, the technique's most important parameter is the minimum acceptable burst. The rescaling step regularly leads to high burst values that do not necessarily reflect the worth of a development. Thus, the threshold needs to be sufficiently high to shut down any false positives.

We empirically set this threshold at 0.5, excluding terms whose burst is lower. ELD considers clusters as breaking developments if at least three keywords have burst values that exceed this threshold, or if the average burst of the emerging terms is higher than 0.9. The associated clusters and their documents are the summarizer's input, alongside a list of tuples of emerging keywords and their burst.

By grouping together documents, ELD not only avoids any performance issues. The embedded feature-pivot algorithm verifies the validity of developments, allowing for a fine granularity while blocking spam and noise. The implementation concludes with summarization, described next, which wraps the TDT information in context.

5.4 Summarization

Often, there is a very thin line separating subsequent developments. Intuition would have it that TDT would be responsible for delineating switches in progressions. TDT does assume this role to a certain extent, with clusters taking on the role of trackers.

However, preferably these groups should also be relatively ephemeral in order not to hinder performance or incorrectly merge developments.

Conversely, Figure 4.4 on page 40 shows just how long it takes for hype surrounding developments to die down. Therefore TDT and summarization tackle this issue together, with the latter focusing on the long-term aspect, or the problem of update summarization.

The algorithms have the added role of ensuring that summaries contain only novel content that the user had not seen earlier in the timeline. The pseudocode of the workflow and the algorithms as described in this section are presented in Appendix A.3.

As update summarization algorithms, FMMR and DGS generate summaries, which we consider to be the nodes on the timeline. The latest node on the timeline is the work-in-progress summary, which is committed after a period of inactivity or a shift in discourse. The timeline structure allows the algorithms to minimize repetition.

Each timeline node comprises documents from the development's clusters. When breaking clusters and their emerging keywords arrive from the TDT component, the summarization methods first check the age of the current timeline node. If it has been inactive for a long time, then it is committed and a new node is added to the timeline.

Then, the algorithm uses cosine similarity to compare the centroid of the breaking documents of each old node with the tweets of the incoming cluster. If the highest similarity among these comparisons is higher than a predefined threshold, empirically set at 0.6, FMMR and DGS decide that the new development is likely a resurfacing moment. Therefore the corresponding node absorbs the cluster and the breaking terms to boost its own information.

Clusters that pass the initial test prompt a choice among two routes. If the latest summary has been active for a long time, a second test compares the similarity with the new cluster similarly to before. If the similarity is sufficiently-low - a threshold that we set at 0.1 - the algorithm creates a new summary on the timeline to reflect a change in the event's discourse. Conversely, if the latest node is relatively new, it immediately absorbs the new cluster.

Two parameters play an important role in our update summarization approaches - the maximum inactivity of a node before committing it and creating a new one, and the minimum time coverage of a summary. The shorter the maximum inactivity, the more singular the focus of the timeline's summaries. However, it also adds unnecessary strain on the algorithms' absorption practice in order not to repeat developments.

The minimum coverage of a summary is a compromise between these two extremes. It enforces a time period when any new clusters are unconditionally added to the cur-

rent summary. Once the development exceeds a certain age, a diversion is permitted if the new cluster varies enough.

The last common ground between FMMR and DGS is the way in which they represent the summary query. Both FMMR and DGS construct this query using only the breaking terms, thereby guiding the summary to include the most topical keywords.

As per Equation 5.4, a query document is constructed for each cluster c. The equation weights component k proportionally to its burst in the corresponding cluster - $burst_{c,k}$ - and the cluster size $|c|$. Each such group creates its own query, which is not normalized. These individual queries come together in a new cluster. Consequently, the bigger clusters influence the centroid more than smaller groups because they contribute a stronger vector. The cluster's normalized centroid becomes the summarization query.

$$query_{k,c} = burst_{k,c} \cdot |c| \tag{5.4}$$

Both FMMR and DGS compute similarities among documents to reduce repetition. To control the duration of execution, the algorithms retain the best 25 documents according to a quality scoring mechanism described further down. Then, the algorithms pick a set of documents to make up a summary in conjunction with the query representation.

The FMMR implementation is the simplest of the two approaches. The technique starts by constructing a similarity matrix that incorporates the chosen documents and the query. The first iteration picks the document with the highest similarity to the development's representation and quality score.

In subsequent iterations, our derivative MMR implementation chooses documents that are simultaneously similar to the query, dissimilar to picked microblogs, and which score highly in quality - a metric that we describe further down.

Normally, MMR stops after reaching a predefined length, but in ELD the documents have a very narrow focus and repetition abounds. We alter the stopping condition to exploit the score that MMR computes at each iteration. The algorithm stops looking to add documents whenever the score drops below a pre-defined threshold or a number of documents - which we empirically set at three - have already been picked.

DGS is similar in many respects. Instead of a similarity matrix, it represents the interlinks between documents as edges weighted according to their lexical similarity in a NetworkX graph. DGS segregates nodes similarly to the APD algorithm; the Girvan-Newman method partitions the graph into the different angles of discussion. DGS considers communities with at least three nodes. If it fails to find one that is large enough, it uses the largest partition.

As in Equation 5.5, a single document d_p is chosen from each partition p. The decision takes three factors into consideration - the document's eigenvector centrality c_d, query relevance $sim(d_p, q)$, based on cosine similarity, and quality score s_d.

$$d_p = \text{argmax}(c_d \cdot sim(d_p, q) \cdot s_d) \tag{5.5}$$

Emotions are evoked throughout literature, influenced ELD's design and finally round off the implementation. Commentators assume that readers have domain knowledge, and this behaviour peaks during the most emotional moments. Quality scoring attempts to promote those documents that make the fewest assumptions, or which are the most descriptive.

The score is the product of two components. Firstly, brevity often works against expressiveness. Therefore the scorer includes a penalty inspired by Papineni et al.'s evaluation metric - BLEU [127]. The brevity penalty is calculated as shown in Equation 5.6. It compares the number of keywords r in a document with a desired length c, empirically set at 10.

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{5.6}$$

Secondly, emotion best manifests itself in capitalization, such as writing made up exclusively of upper-case letters, like *GOOOOAAAAAL*. We penalize messages which make abundant use of capitalization, thus avoiding the inclusion of tweets written in heated situations. This score is simply the fraction of lowercase letters in the document.

The chosen documents make up summary objects on the timeline, but to improve coherency among the different tweets, the microblogs go through a cleaning process. Processing starts by stripping new-lines and removing dangling spaces at either end.

We also strip away the embellishments that lose all meaning when taken out of Twitter's context. These include URLs, some emojis, ALT-codes and the *RT* prefix prepended automatically to retweets. The cleaner also splits hashtags with camel-case notation, but retains mentions in the likelihood that they refer to named entities.

The last processing step collapses consecutive spaces to a single one, and suffixes documents that do not end in punctuation with a period. The microblogs are ordered chronologically and concatenated to form the final summary node on the timeline.

Owing to the the popularity of retweets, authoritative tweets appear across different summaries - a give-away of developments that are still being discussed. Thus, we cross-check moments with older descriptions, removing duplicate documents, and possibly summaries altogether, from the timeline.

With the cleaning step, FMMR and DGS meet again at a common point. They take fragments of a story, bring them to order and make sense of them. Through a meticulous process, the approaches ensure coverage and refine the deliverable to maximize coherency.

5.5 Summary

Tens of thousands of documents stream in, and one by one, they undergo stringent filtering. Then, ELD processes and clusters them to form pieces of a larger picture. Finally, the contributions of thousands of users become a succinct timeline.

In this chapter we described a process that does not simply consume an event, but first and foremost understands it as a function of its participants. ELD's process is heavily-parametric, but it is one that aims to understand and explain events better. It is in this setting that we evaluate ELD in terms of its individual components, and then as an entity made up of its collaborating components.

6

Evaluation

ELD's architecture was created to track an event by understanding its domain. The initial depiction of the event is all that APD gets as it goes on to influence the rest of the proceedings. The TDT algorithm works directly on the content that ELD captures through the individual participants to find the developments that characterize the event.

The output of the TDT component becomes the input to a summarization module that transforms the machine-extracted knowledge into a more descriptive form. ELD stacks these three components upon each other, and this chapter explores their individual capabilities and their combined strength.

Section 6.1 looks at the evaluation methodologies that similar research endeavours adopted. ELD builds on this literature, basing its own evaluation plan on existing systems. Our methodology is explained in Section 6.2. The quantitative results follow next in Section 6.3, where we also present a qualitative analysis with a football writer at The Guardian. This chapter concludes in Section 6.4.

6.1 Evaluation in Literature

6.1.1 Automatic Participant Detection

In IR, evaluation approaches can often be reduced to a single question - how accurate are the results that the algorithm returns? Query and entity set expansion vary wildly in their scope, but they are essentially IR problems that attempt to improve search results; the former by refining a query to perfect a user's search, and the latter by extrapolating the query itself.

The two fields have a thin, but clear separation from APD. Thus, in spite of their differences, and in the absence of any APD approaches in literature, query expansion

and entity set expansion are the only basis for ELD's evaluation of its first component.

Nakade et al. take a very simplistic approach that borders on the naive. The authors put their thesaurus-based query expansion algorithm to the test by counting the number of additional retrieved tweets [37]. The approach is arguably too simple because in reality, quality is as important as quantity.

This is why evaluation approaches prefer to turn towards the quality of the returned items. Conventionally, IR thinks of quality in terms of precision and recall, shown in Equations 6.1 and 6.2 respectively [44, 128].

$$precision = \frac{tp}{tp + fp} \tag{6.1}$$

$$recall = \frac{tp}{tp + fn} \tag{6.2}$$

The self-descriptive precision measure compares the true positives tp against the entirety of the returned collection, including any false positives fp. The recall measures how many of the relevant documents were returned, measuring the true positives against the set of false negative documents fn that were not retrieved [44].

The precision and recall metrics exhibit complementarity because they assess how well an algorithm identifies relevant content and how much it misses. It can be tempting to maximize either of the two, but such an approach takes a one-sided view. The harmonic mean, shown in Equation 6.3, combines the two in one simple metric that penalizes systems that favour one over the other.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6.3}$$

Query expansion finds its basis in these approaches because it is directly concerned with refining search terms to improve document retrieval. The most notable hitch is that in most cases, the recall metric is dependent on a labelled corpus that assigns relevance judgements to documents.

In fact, manually creating enormous labelled corpora to simulate a search engine's performance is a time-consuming task to the point that it becomes infeasible. In the lack of such a dataset, Milne et al. engaged human judges to assess the three metrics outlined above [82].

The alternative is a compromise that excludes recall. Though extreme, this route is not nonsensical either. It takes a practical view of what query expansion means in real applications - a ranking mechanism to bring forth certain documents while relegating

others. In this viewpoint, what matters is not the global relevance of the system's output as much as the point when it answers the query [128].

The Precision at k (P@k) metric, formally defined as shown in Equation 6.4, considers that rankings are meant to deliver relevant content as high up as possible. Therefore the measure cuts off the top k documents and evaluates precision on this subset. r_i is a boolean membership function that equates to 1 if the document is relevant, and 0 otherwise [47, 128].

$$P@k = \frac{\sum_{i=1}^{k} r_i}{k} \tag{6.4}$$

From this metric appear other common ones that find a place in query expansion. Average Precision (AP), shown in Equation 6.5 averages the P@k at each relevant document [128]. More frequently, multiple queries are made to assess the algorithm's performance across different settings. The more popular Mean Average Precision (MAP), shown in Equation 6.6, handles this case [128].

$$AP = \frac{\sum_{i=1}^{k} r_i \cdot P@i}{\sum_{i=1}^{k} r_i} \tag{6.5}$$

$$MAP = \frac{\sum_{i=1}^{q} AP_i}{q} \tag{6.6}$$

The MAP metric averages the AP values over the q rankings that result from each query [47, 128]. Turpin and Scholer note that MAP also considers recall, albeit very lightly, since incorrectly-retrieved documents contribute a zero-score to AP [128]. MAP has been the predominant route to evaluate query expansion techniques built to refine rankings. As in the research undertaken by Turpin and Scholer [128], users are mainly interested that a correct result is shown as early as possible.

In this way, Egozi et al. [83] base their evaluation on a TREC dataset. Others adopt similar approaches, often using annotated datasets [50, 78, 81] or manually-collected and tagged corpora [85]. Although other metrics make an appearance in some of these studies, including P@k [78, 81, 85], MAP dominates as the metric of choice.

The trend continues in entity set expansion. Gabrilovich and Markovich's evaluation looks at the break-even point between precision and recall [79]. More commonly, the class entities are returned with decreasing confidence, priming them for a ranked evaluation.

Although precision and recall make spurious appearances [76], they are more often discarded in favour of other metrics that measure the quality of rankings. The most obvious method of evaluation is again the P@k metric, present in research by Letham et

al. [77] and Zhang et al. [80]. Similarly to query expansion, MAP is commonly adopted when the evaluation covers several classes [47, 75].

Unlike the innumerable environment of query set expansion, concept instances can sometimes be limited to an irrefutable set. Among others, Sarmento et al., Pantel et al. and Letham et al. take class instances to mean those elected by contributors in Wikipedia *List of* articles; any entity not in the list is simply irrelevant [47, 76, 77].

This knowledge is inviting to another type of evaluation - the Ranked-Precision (R-precision). When the number of instances in the class are known in advance, the cut-off point of the ranked list is set at this figure, and the precision is calculated at this rank [76]. Dalvi et al. [75] and Pantel et al. [76] use this metric in their analyses.

If evaluation methods in research show anything, it is that APD shares characteristics with both query expansion and entity set expansion. It attempts to understand a query, and extrapolate it with additional participants. These effects are subsequently felt in TDT, whose background in evaluation follows next.

6.1.2 Topic Detection and Tracking

At face value TDT seems like the antithesis of APD. Instead of extrapolating information, it looks inwards at a corpus to detect its salient developments. However, just like APD, TDT is an IR task, and it is considered as one in popular literature. Systems that are similar to ELD tackle the evaluation of TDT from various angles.

Ghelani et al. perform somewhat of an empirical evaluation that compares detection times with Twitter's Moments functionality [62]. Although Gao et al.'s system focuses on summarization, the authors measure development segmentation by calculating the similarity between subsequent moments [70]. As comfortable as they are in their simplicity, these approaches are unconventional.

Instead, TDT builds an evaluation foundation on more common IR measures. What sets apart TDT from query expansion and entity set expansion is that TDT's value lies in its ability to capture the topics that make up events. For this reason, recall is as important as precision.

The existence of a ground truth corpus would be attractive for a comparison between similar systems, but this is hardly possible owing to the Twitter API's policy[1]. The social network only allows the publication of tweet IDs, which requires microblogs to be downloaded anew. Tweets are often deleted and irretrievable, greatly hindering

[1] `https://web.archive.org/web/20180819144731/https://developer.twitter.com/en/developer-terms/agreement-and-policy`, last accessed on 2 March, 2019

dataset re-use. Therefore it is usually on manually-collected datasets that evaluations are carried out.

The point of departure is that a TDT algorithm that fails to isolate developments and instead echoes noise is only a slight improvement over more traditional methods. Therefore Nichols et al. and others incorporate precision in their quantitative evaluation [14, 29, 42, 65]. Aiello et al., whose algorithm surfaces a long list of keywords, look at P@k as an alternative to precision [26]. Recall often accompanies precision, with research showing the usefulness of clear environments for such an analysis.

The list of existing similar systems in literature that focus on football matches is long, and with reason [14, 26, 28, 29, 64, 72, 129]. With football's most important developments, like goals, being unquestionable, not only are key moments enumerable, but ground truth is also widely-available in mainstream media. However, an element of interpretation prevails in literature about what constitutes the ground truth, sometimes only explained as "interesting" moments [15].

Literature takes a wide view of important developments in an event. The solutions vary from one extreme to the other; Löchtefeld et al. focus narrowly on goals and cards [29], while Van Oorschot et al. incorporate substitutions [28]. No real consensus surfaces, but literature demonstrates a clear preference for irrefutable facts. This includes game stoppages and resumptions [14, 26, 64, 72], goal situations [14, 26, 28, 29, 64, 72, 129], punishable offences [14, 26, 28, 29, 64, 72, 129] and scarcely, substitutions [28].

The evaluations look at the systems' abilities to capture these events. Since the analysis is often a manual one, many studies focus on a narrow selection of events. Corney et al. follow two matches [15], whereas others follow three [26, 42, 64, 129]. Authors of classification systems afford to evaluate on more datasets. For example, Zubiaga et al. collected 26 football matches [72]. Van Oorschot and Zhao et al. used 61 football matches and 33 NFL games respectively [12, 28].

Notwithstanding the allure of a clearly-defined environment, football, like many other domains, is not only about the facts. Emotions weigh heavy in more subjective actions. Khan et al.'s way out is to forego recall completely and focus uniquely on precision, altering the metric's definition to include narration [65].

Eager to stick to the recall mechanism, Corney et al. shift their attention towards evaluating the recall of "interesting" developments in two FA Cup Finals [15]. This decision opens the evaluation to moments with a subjective truth value since interest is not easily quantifiable. The issue comes back to the choice of what constitutes a development in a football setting.

For instance, substitutions are essentially facts, but they are unpopular among supporters because they do not immediately influence the proceedings. In fact, only Van

Oorschot et al. include them in their evaluation [28]. This problem constitutes the level of development detail, which is particularly close to ELD due to its fine granularity.

Granularity is not a commonly-evoked problem in similar systems, mainly because sometimes they struggle even with the major facts, let alone subtler moments. One close area that suffers from this issue is the more general TDT field. Algorithms like FIRE operate in the public stream, consuming content from various events simultaneously [9]. The broad nature of these streams render it impossible to enumerate every single development [9, 73, 92] and ground truth is normally unavailable [23, 94].

Some researchers innovate; Brants et al. evaluate their system on the probability of missing a story [56]. However, the most popular solution is deceptively-simple. Like query expansion and entity set expansion, calculating recall is infeasible, and thus the mere possibility is abandoned. Instead, many systems that operate in Twitter's public stream focus solely on precision [69, 73, 92, 94] or variants of it, like P@k [130].

Though admittedly challenging, recall can find a place in the evaluation of such broad systems. This is usually approached with caution by using a labelled corpus to control the domain [20, 22, 44, 131, 132, 133]. Lappas et al. evaluate their search system by enumerating the major events listed on Wikipedia [134].

Twitter Trends, which are topics detected by the social network itself, have also been adopted as ground truth. Since Twitter extracts these events from the network itself, they offer the benefit of having similar input to that of systems built on tweets [9, 24]. Nonetheless, the nature of Twitter Trends makes them viable only in broad streams.

Otherwise, a slightly different definition of recall can limit the domain just enough to permit it. This is the case in Krumm et al.'s research, who refer to "relative recall" as detected topics that also originate from a particular geographical location [23].

Seen in isolation, precision and recall can work against each other. Therefore in practice, many studies that operate in clear domains strive for a balance between precision and recall, reflected in the harmonic mean [8, 13, 28, 64, 72].

These evaluation techniques present the two worlds of TDT. On the one hand, narrow streams present a bounded environment. On the other hand, these same formalisms can serve as a shackle. We borrow from these two areas of TDT evaluation to quantify the performance of ELD.

6.1.3 Summarization

Summarization is a thorny field. It acts as a middleman between some form of knowledge produced by a machine and a human reader. As an algorithmic component, sum-

marization has a limited understanding of the content it is expected to transform into a coherent message.

The evaluation of summarization is also split between these two worlds. These algorithms are designed to cater to human readers, but human evaluation is expensive and inviting to subjectivity. Beyond the struggle of choosing a middle ground between the automatic and manual evaluation methods, the assessment itself is in a mature state.

An automated evaluation comes with many advantages, not least its simplicity. Long et al., who summarize events using keywords, evaluate the system using the P@k metric. However, this approach is uncommon. In fact, the authors only adopt it in lack of a labelled dataset [94].

Two automated approaches exist in literature - BLEU and ROUGE. Created more than a decade ago, the two metrics found their basis in IR. More specifically, BLEU and ROUGE allude to the precision and recall metrics respectively.

BLEU was originally proposed by Papineni et al. in 2003 as an alternative, automated evaluation approach for summarization. The metric compares machine-generated summaries with those created by human writers analogously to the precision metric of Equation 6.1 on page 65. However, it is also adapted to handle n-grams [127].

The inclusion of n-grams works towards approaching the assessment to human interests. Papineni et al. describe unigrams as working towards adequacy, or comprehensiveness. Bi-grams and n-grams of higher order contribute to fluency by considering the coherency and overlap with the human summaries [127].

The BLEU metric also contributes a brevity penalty, which serves as the inspiration for ELD's own document quality assessor in Section 5.4 [127]. Despite BLEU's applicability, few studies include it in their analyses [14]. Instead, the metric mostly serves as a springboard for another measure - ROUGE - that focuses more actively on coverage.

Lin presented ROUGE two years after BLEU, taking inspiration from it to create the recall counterpart to BLEU. In this way, the metric focuses on the relevant value that a machine-generated summary captures [135]. ROUGE borrows many of the ideas from BLEU, and it has since become the de facto automated evaluation approach in literature. In fact, the distinguishing feature between studies is commonly only the source for ground truth.

Many researchers create their own gold standard, either by picking model documents [13, 30, 38, 63, 74, 89] or by having annotators generate a summary themselves [112, 113, 115]. In other cases, researchers seek a gold standard, such as those provided by TAC [121, 136, 137] and DUC [105, 106, 107, 110, 117, 120, 138, 139, 140].

Other public sources, most notably online resources, have also been explored in cases where existing corpora did not meet the study's requirements. Chua et al. map

captured events to Wikipedia articles, which fuel their quantitative analysis. Later, they complement this analysis with a qualitative one [123]. More commonly, researchers adopt news reports as authoritative, human-generated summaries.

The mainstream media has proven to be useful in literature when up-to-date summaries are required [64, 141]. Takamura et al. use online text reports in their ROUGE evaluation [61]. Intuitively, the popularity of certain events leads to media coverage that is simultaneously of high-quality and broad, especially in sports.

Nichols et al. construct their ground truth football summaries using ESPN's and FIFA's own updates [14]. Kubo et al. reference Yahoo! News, but since their algorithm operates on Twitter data, they supplement these summaries using tweets from authoritative users [25]. Elsewhere, Liu et al. use Twitter's trending topics and linked articles to build reference summaries [142].

Although studies have shown that ROUGE and BLEU correspond with user evaluations [127, 135, 143], the lexical variations befuddle these metrics. These shortcomings can only be overcome by a human evaluator [140]. Although subjectivity is an issue when human evaluators are employed, it is not uncommon for research studies to incorporate both an automated evaluation and a human assessment [113, 123, 140, 142].

Literature evaluations have given machine-generated summaries to readers to evaluate them on more practical aspects. The assessed properties vary from one paper to the other, but they aim to capture the desirable characteristics of summaries as described in Section 3.4.

Sharifi et al.'s content metric is one simple measure that they adopt from DUC 2002. The metric captures the semantic similarity between an automatic summary and a reference description [113]. Becker et al. take a more specific approach by asking human judges to assess the quality, relevance and usefulness of summaries. Using these metrics, the authors could understand the description's expressiveness [119].

Other research assesses quality, but hones in on particular aspects. Twice, Olariu et al. looked at grammatical correctness and the summary's coverage [40, 109]. Nichols et al. too consider completeness and grammaticality in the generated summaries, substantiating it with a readability metric [14].

Rudra et al. look at four different characteristics - coverage, redundancy, novelty and readability [7]. Dealing in sequential summarization, Gao et al. include sequence assessment in addition to novelty and readability [70].

Alonso et al. go in a completely different direction by incorporating A-B testing. Instead of comparing a generated summary with a referential one, the researchers presented five human judges with pairs of algorithmic timelines side-by-side and asked

them to choose the better one, if any stood out [67]. De Maio et al. follow a similar approach, asking readers to pick the optimal configurations [68].

The limited number of human judges is a limiting factor. Chua et al. sought a larger pool of judges in Amazon Mechanical Turk[2], which gives access to paid workers all around the globe. Chua et al. paid some to participate in a qualitative evaluation that is similar to De Maio et al.'s [68] preference-based assessment [123].

The approach is likeable enough, but Chua et al. ended up paying around $420 to their 100 workers [123]. Perhaps it is because of this financial barrier that Amazon Mechanical Turk is uncommon in literature. Even less common are specific qualitative analyses like Marcus et al.'s, who evaluated their system by presenting it to a journalist, shedding light on the system's usefulness and applicability [42].

Literature presents clear cases for evaluation approaches, not only for summarization, but also for APD and TDT. These methodologies are discussed in the next section, which lays out ELD's own evaluation strategy.

6.2 Evaluation Plan

6.2.1 Datasets

ELD's process is a seamless one that transitions from APD to TDT, and then to summarization. We build gradually, moving from one component onto the next and permitting an analysis of the influence that each decision has on the ensuing process. To this end, the analyses of this chapter are all based on the same datasets.

The TDT literature outlined in Section 6.1.2 culminates in one clear conclusion - a controlled environment facilitates the evaluation. Constrained by Twitter's policy, we follow in the steps of other researchers in constructing our own datasets, largely based on pre-planned, rigidly-defined sports events. Due to the demands of TDT, described in Subsection 6.2.4, we base most of the quantitative evaluation on football matches.

The sport enjoys an immense popularity worldwide, which has cemented its place as one of the most followed disciplines in existence. With many popular leagues, most notably in Europe, big matches abound. Furthermore, the masses do not hold back in discussing them, leading to frequent and voluminous discussions.

Nonetheless, ELD has a general scope so that the individual components are portable to other systems and potentially applied in different fields altogether. For this rea-

[2] `https://web.archive.org/web/20181128142549/https://www.mturk.com/`, last accessed on March 2, 2018

son, we include in our evaluation datasets from other areas, including planned political events.

In total, we collect six football matches with different degrees of popularity, which leads to a comparison of results among datasets of varying sizes. This number is around double than what is usually taken in literature when the task is not a classification one. The low figure reflects the time-consuming task of manually marking the algorithms' results.

In addition to these matches, we also collect datasets from one UFC (mixed-martial arts) main event, the Brazil presidential election of 2018, and the USA's midterm elections of 2018. Naturally, the evaluation procedures adapt to the different environments, as explained in the rest of this section.

We use the football match datasets throughout in the quantitative evaluation because they are adequate for the precision and recall analysis of TDT, as described in Subsection 6.2.4. The APD evaluation is the only one that uses the UFC, election and football datasets. We use these datasets to examine how our APD algorithm performs in different scenarios. Since the APD evaluation focuses only on participant detection, it uses only the corresponding understanding corpora.

We collect the datasets using the Twitter API and the Tweepy library for Python 3 using a manually-provided, case-insensitive seed set. The seed set varies according to the event, but the general rule is to assume simple knowledge about the event.

Similarly to Aiello et al. [26], we initially track events using their principal parties and hashtags, whenever available. This minimal set of keywords serves as the seed set. For example, to collect the dataset for the football match between Liverpool F.C. and Manchester United F.C., we used the seed set containing *#LIVMUN*, *Liverpool* and *Manchester United*.

We collect each event three times using two different Twitter application credentials. These datasets are the basis for the evaluation of one of the most fundamental hypotheses in ELD - that an understanding of the event results in improved results.

The first dataset uses only the bare seed set, representing a person that has only superficial knowledge of an event, or who chooses not to describe it in detail. In the rest of this chapter, we refer to this dataset as the baseline corpus. The second dataset incorporates the automated process of identifying participants, which are tracked for the duration of the event - the APD corpus.

The third dataset collects data about every single participant in the event, provided manually. This represents the best possible performance of APD. In this way, we also permit an evaluation that is independent of APD's imperfections. We refer to this corpus as the manual or full dataset.

We note that two of the six datasets contain inadvertent noise with the inclusion of players who were not substitutes and did not participate in the match. However, we show that TDT resists against this noise in Subsection 6.3.2.

Each APD dataset that results from this algorithm is split in two. We collect the first corpus of tweets during the understanding period, which also represents the local context of an event. The APD algorithm operates on this dataset to extract the participants for the event. These entities became tracking keywords for the remainder of the event - a time period that ultimately builds the boosted second APD dataset.

Like the initial seed set, the length of the understanding period varies from one type of event to the other to maximize the information gain. For sports events, data collection starts 75 minutes before their scheduled start and ends an hour later. This window overlaps with the time when football line-ups are published, increasing discussion about participants. Stopping a quarter of an hour before the game starts allows APD ample time to perform its task without infringing on the event itself.

It should be noted that since APD's execution time is unpredictable, the datasets start being collected during slightly different times. However, by starting collection a quarter of an hour early, all datasets capture the entire event. We follow a similar procedure for events with an undetermined time slot.

The rest of the collection ends half an hour after the scheduled finish, when known. This results in two and a half hour collection windows for football matches. This time window applies to both APD and normal data collection tasks. Once more, since the APD algorithm takes a few minutes to conclude, the participant-boosted datasets extend until after the other corpora have already concluded.

Other events that are unpredictable by nature, like elections, necessitate a slightly different approach. With conversations plentiful, a shorter understanding period is affordable. After APD concludes, we provide an exaggerated window, but cut it short as soon as the event ends.

An overview of the datasets used in this evaluation is in Table 6.1. We present the full details in Appendix B, alongside complete information about their collection processes. The rest of this section describes how we make use of these datasets to evaluate the performance of ELD.

6.2.2 Ground Truth

An automated system depends on ground truth to evaluate its performance. In ELD's evaluation, we construct our reference output for all three components using widely-available sources.

Event Type	Event
Football Match	Manchester United F.C. - Arsenal F.C.
	Liverpool F.C. - S.S.C. Napoli
	Valencia CF - Manchester United F.C.
	Liverpool F.C. - Manchester United F.C.
	Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.
	Crystal Palace F.C. - Chelsea F.C.
Mixed Martial Arts Event	UFC 232
Election	Brazil Presidential Election 2018
	USA Midterm Elections 2018

Table 6.1: The datasets used in the evaluation.

APD's evaluation is perhaps the simplest in terms of ground truth availability. UFC's website contains the full line-up from UFC 232[3], whereas election participants and related concepts can be verified using Google searches. We use LiveScore.com[4] to establish the ground truth participants for the football matches. In all cases, we look for participants that are discriminative and temporally-relevant. We discuss what this constitutes in more detail in Subsection 6.2.3.

LiveScore.com also contains the ground truth for the entirety of TDT's recall-based evaluation, whose distinct enumerable developments are detailed in Subsection 6.2.4. For the precision-based analysis, we use a combination of Twitter searches and The Guardian[5] to assess the veracity of reported developments.

The Guardian, a media outlet based in England, reports on football matches using timelines in real-time, achieving the same objective as ELD. Therefore on top of being a valid ground truth for TDT's precision, it also serves as one of summarization evaluation's references.

Analyses that compare machine summaries with mainstream media reports have been followed in literature [14, 25]. However, like Kubo et al. [25], we also supplement

[3] `https://web.archive.org/web/20190218105721/https://www.ufc.com/event/ufc-232`, last accessed on February 18, 2019

[4] `http://web.archive.org/web/20181224124905/https://www.livescore.com/`, last accessed on December 24, 2018

[5] `https://web.archive.org/web/20181202202320/https://www.theguardian.com/uk`, last accessed on December 2, 2018

this evaluation by comparisons with authoritative Twitter accounts.

Particularly in sports events, franchises and teams alike curate accounts that commentate proceedings as they happen. We collect the tweets published by these accounts during the event to act as an alternative ground truth for the summarization evaluation.

This combination also permits an indirect analysis into just how different Twitter's conversations are from the masterful content of publishing houses. In our evaluation, we consider the English Twitter accounts of the teams involved and, if available, the competition's own account. We explain how we use these datasets and the related ground truth in the next subsections.

6.2.3 Automatic Participant Detection

In the literature review section, we showed how APD is a novel concept that borrows from two other fields. Since we were unable to find any projects that incorporated approaches that are similar to APD, we are given a certain liberty to define the evaluation approach. We associate this approach with ELD's scope, exploiting the rigidity of football matches.

Entity set expansion research often looks at Wikipedia *List of* articles, such as *Lists of Nobel Prizes*, as loosely-defined classes. We refer to these classes as such because even in literature, studies are wary of considering them to contain all instances of the corresponding class. This consideration is what drove Sarmento et al., among others, to focus solely on precision [47]. However, the TDT literature contributes to this cause.

In its pursuit of events with rigid rules, TDT yields environments that are welcoming to APD's evaluation. Contrary to the tentative *List of* Wikipedia pages, football matches' participants are clear and unwavering. Departing from the fact that these participants follow a rigid structure, then recall can also be calculated. Naturally, the definition of participants has to be in line with the two rules that we laid out in Section 4.2 - they have to be simultaneously discriminative and temporally-relevant.

Although literature has not explored the APD task to our best knowledge, we look to existing research for a fitting definition of participants in sports events. On top of the teams themselves, evaluation approaches consider players to be crucial in the proceedings and track them to construct datasets [13, 15, 25, 26]. Shen et al. consider coaches as well [13], just like Han et al., who tracked key players and team coaches during Superbowl XLVII [27].

Players and coaches fit our definition of participation, permitting a recall evaluation. We include not only players in the starting line-up, but also those who are on the bench since they could become active participants later on. On top of these entities, we also

include the stadium name. Other agents are technically participants, such as the referees and assistant coaches. However, we observe that they are very rarely mentioned by name and we exclude them.

Thus, for each football match dataset we create a factual ground truth that includes the names of the involved teams, players, coaches and the name of the stadium. Similarly for the UFC 232 event, we construct a factual ground truth that includes the fighters participating in the event's main card section.

Other environments do not present such a rigid environment, rendering a recall analysis impractical. In the USA's midterm elections of 2018, for example, the participants could have been infinitely small. The major political parties fielded candidates in smaller elections across a number of states. In cases such as this one, the stream was broad in a relative sense - we tracked the USA election, rather than its individual, smaller state-specific elections.

This decision poses difficulties that are not dissimilar to those encountered in the entity set expansion evaluations, where a recall-oriented analysis is often infeasible. To this end, we skip the recall evaluation in the elections datasets. Instead, we carry out a precision-based analysis for all events, including not only the elections datasets, but also the football matches and UFC datasets.

APD's results determine the ensuing data collection process. All of the participants that the algorithm identifies become tracking keywords for the event, no matter their order. Therefore the precision over this set is the first metric that we calculate.

In football matches, we consider the participants mentioned above to be precise, whereas in the UFC dataset we look only for the main card fighters. Thus, in these cases we calculate precision and recall by manually matching the output with the ground truth. In the less clear environments of elections, any concept that is relevant to the proceedings and that adheres to the bounds of participation contributes to precision. In the elections datasets, we mark the precision of participants manually by checking whether they are discriminative and temporally-relevant.

Our algorithms contribute more than a few suggestions for participation. In practice, we set an empirical threshold of candidates that is approximately the number of participants in the event. Nonetheless, moving this threshold around transforms the problem into a ranking one - the higher an entity's rank, the more likely it is to make the cut as a participant.

The resolution and extrapolation phases described in Section 5.2 provide two lists of participants, which we concatenate in this order. Resolved entities rank higher than extrapolated actors on the basis that participants from the local context are discussed in direct relation to the event, and thus the system is more certain of their validity. We

adopt P@k and MAP to analyse how APD's results change as this certainty fades away.

Choosing a baseline is far more problematic than determining the evaluation metric. Query expansion techniques improve coverage through terms, rather than participants. On the other hand, entity set expansion techniques require a manually-provided seed set.

Having failed to find existing APD algorithms, and seeing as these two related areas are far too different, we propose a new baseline for APD. We compare the results against a frequency-based ranking of the named entities collected in the same dataset.

In ELD, the identified participants are just the ingredients that serve a higher purpose - that of boosting the associated corpus with more tweets. The next subsection details the plan to analyse how APD affects the generation of timelines through TDT.

6.2.4 Topic Detection and Tracking

A TDT system is expected to consume an often-voluminous stream and make sense of it. This breaks down not only to surfacing the most important facets, but also to reduce the noise. Our evaluation plan focuses on the football datasets to assess ELD's centrepiece component based on these two desirable characteristics.

The evaluation uses a custom-built simulator that feeds tweets from each dataset to ELD's implementation as if the event is happening at that moment. All consumption tasks use the understanding corpus collected in the APD workflow to build the TF-ICF table. We split the quantitative analysis in two parts like other IR systems [14, 42].

The first part of our evaluation assesses ELD's ability to capture the most important enumerable developments during an event, analogous to recall. The second part of our evaluation takes granularity to heart and looks at precision to analyse how well ELD detects and removes noise.

Recall cannot be easily calculated - at least, not without making a compromise. In literature, that trade-off translates into the consensus of researchers that drifted towards enumerable sporting events. We follow this convention in our evaluation, collecting datasets from football matches as described in Subsection 6.2.1.

Enumeration necessitates objectivity, which literature seeks in fact-based moments that leave no room for doubt. Game stoppages, goal situations, punishable offences and substitutions cover these possibilities, and have been adopted in the research mentioned in Subsection 6.1.2 for a recall-based evaluation.

Our recall evaluation excludes developments like offsides and fouls; they are common, have minimal impact on the proceedings and they are not considered in literature

to the best of our knowledge. Instead, we consider all of the fact-based developments that are present in literature:

- Stoppages or continuations
  - Kick-off
  - Half-time
  - Second-half kick-off
  - Full-time
- Goal situations
  - Goals
  - Own goals
  - Penalties
  - Disallowed goals
- Punishable offences
  - Yellow cards
  - Red cards
- Substitutions

Our TDT and summarization approaches are closely linked together. For this reason, we base our evaluation on the generated timelines. We consider any of the above developments to be captured if the summaries refer to them within a few minutes of them happening. This leniency compensates for the fine granularity, which sometimes results in too few descriptive tweets to describe a development in detail.

We hypothesize that a fine granularity contributes to detect the more forgettable developments. Other subjective developments too evoke emotions, but they cannot be easily demarcated. To this end, we adopt Marcus et al.'s approach since it eliminates subjectivity; the authors evaluate using both precision and recall, reverting to precision when developments are not easily enumerable [42].

The enumeration challenge is reminiscent of issues in broad streams - describing the moments that attract interest - no matter how minimal - is hardly feasible. Therefore we evaluate ELD's timelines more generally by looking at whether or not a summary mentions a development that happened in reality.

Furthermore, we observe that an event is not simply about what happens, but also about what is happening. We refer to Khan et al.'s approach, which considers narration. The authors explain that narrative comments illustrate the event's setting, and thus contribute to precision [65] - a decision that we adopt ourselves.

Precision and recall only describe the first role of TDT - detection. Tracking is a responsibility that is shared between this component and summarization. The two summarization methods that we presented naturally vary in the timelines that they generate. As a result, this affects the update summarization algorithms' tracking component.

Although TDT shares this update track with summarization, we evaluate its results alongside TDT. Thus, we repeat each experiment twice - once with each summarization algorithm. Each time, we evaluate how well ELD manages to remove redundancy by invalidating precision if a summary adds no new substance to the event timeline.

We use similar configurations across all football datasets. We established all specifications empirically during the experimentation leading up to the evaluation, but we fine-tune ELD's parameters to adapt to the varying levels of discussion among datasets. We list the full specifications in Appendix C.

We compare our TDT component with Zhao et al.'s algorithm [12]. Already used as a baseline by Zubiaga et al. [72], Zhao et al.'s TDT approach is a traditional, volume-based algorithm. Zhao et al.'s method, which can work in real-time, uses a dynamic time-window solution [12].

Their algorithm starts with a time-window of 10 seconds, building up to 20, 30 and 60-second time-windows. The algorithm dissects each window into two halves, and analyses the post-rate change between the two. If the second half has at least 1.7 times as many tweets as the first half, then the window represents a breaking development according to the algorithm [12].

It should be noted that Zhao et al.'s algorithm was originally a lexicon-based classifier for the main NFL developments [12]. Since this lexicon is not public and unrelated to football, and ELD itself prioritises granularity, we do not include this step. Instead, we summarize the tweets using MMR to understand what the development represents.

The end-product - a timeline - represents the objective of TDT. It reduces corpora of thousands of documents into a succinct representation of the highlights of an event. Yet a timeline loses its purpose if it is incapable of communicating clearly with users. The next subsection outlines our approach to evaluate the quality of the timelines.

6.2.5 Summarization

In summarization, evaluation is man against machine. And ironically, in a field that is so close in contact with humans, the efforts multiply to find an automatic way of assessment. Though idealistic, human evaluation is subjective, and eliminating that human blemish is unrealistically expensive.

Though sub-par in qualitative terms, an automated evaluation brings research in agreement. ROUGE is the metric in the majority of cases, and not without reason. Studies reveal that ROUGE correlates with human judgements, de-emphasizing the flaws of a machine evaluating a human-readable summary [135, 143].

Though ROUGE does not capture some human attributes, like cohesion, we follow the majority of research studies in adopting the metric. Our evaluation plan adopts the recall-based metric to measure our algorithms' capability to capture the most important aspects of developments. Since our summaries vary in length, like Nichols et al. [14] we incorporate BLEU and the harmonic mean in our evaluation.

We use a ready-made Python library for the ROUGE evaluation, which pipes the input to a standard *perl* implementation[6]. The package comes with default options for the evaluation, but we append flags to remove common words and stem keywords.

Common words, such as stopwords, are excluded due to the variability in summary length. This approach avoids inflating ROUGE scores with the inclusion of stopwords by virtue only of a longer summary. Instead, we assign more importance to uncommon, topical keywords.

We perform the evaluation on the football datasets that tracked all the participants since they achieve the best results in Subsection 6.3.2. We compare the resulting timelines with The Guardian's timelines and tweets by authoritative accounts, as described in Subsection 6.2.2. Nonetheless, we cannot ensure total overlap between ELD's summaries and those written on The Guardian and Twitter.

Narration, for instance, is seldom discussed by official Twitter accounts. Therefore we skip the evaluation of machine summaries that do not have a reference description. The inverse applies too - some reference summaries discuss a moment that none of the machine summaries do. We only include such comparisons if at least one machine summary discusses the reference summary's moment.

Whenever we make these comparisons, we include all of the machine summaries that happened at approximately the same time, even if some machine summaries do not capture the development. In this way, the evaluation does not only analyse the quality

[6] `http://web.archive.org/web/20181224162622/https://github.com/bheinzerling/pyrouge`, last accessed on December 24, 2018

of summaries when they are correct, but also how well the approaches can pinpoint the documents that are relevant to the development.

ROUGE's correlation with human judgements has not been discussed. However, it may not be purely incidental. The n-gram set-up, though in no way a replacement for human attributes, can facilitate an approximation. The concatenation of lexemes can approximate grammatical validity and thus we perform tests using bi-grams.

ELD's two summarization approaches generate different timelines. Therefore to allow for a direct comparison with a baseline, we borrow Nichols et al.'s strategy. When Nichols et al.'s algorithm finds a new moment, it proceeds to identify the documents that best represent the development. Their solution to make summaries comparable is to use the same documents as the input to the baseline [14]. We follow Nichols' et al. approach to calculate the baseline.

The timeline constitutes a number of nodes, each made up of documents scattered across clusters. We re-create ELD's real-time timelines using the final clusters and their documents in each node to create retrospective summaries. The same process is followed for each timeline, but this time using the baseline technique - MEAD - which is also adopted as a comparison algorithm by Wan [137].

Radev et al.'s MEAD picks documents that are close to the corpus' centroid [114]. Since the FMMR and DGS timelines vary, we dispatch MEAD twice, and it operates individually over each node in the timeline to generate its own description.

By default, MEAD reduces corpora to 20% of the original size. However, we find that voluminous moments, though describing one or two minutes, produce long summaries. For this reason, we limit summaries to around 50 words, which is approximately the average length of ELD's generated descriptions. We use a standard MEAD package for summary generation[7].

MEAD gives us an idea of how FMMR and DGS compare with a basic summarization model. Nonetheless, an automated evaluation stops at the superficial, ignoring the human qualities of a summary. Therefore to understand these subtleties, we perform a qualitative analysis, as described in the next subsection.

6.2.6 Qualitative Analysis

ELD builds timelines to describe an event to humans, and similarly it takes a human to evaluate its fine details. Whereas the previous subsections described how we will

[7] `http://web.archive.org/web/20181224163118/http://www.summarization.com/mead/`, last accessed on December 24, 2018

evaluate the individual components, the qualitative analysis that we outline next looks at the bigger picture of ELD.

The timelines that ELD produces have a certain journalistic quality to them; like reporters, ELD follows an event, albeit from a second-hand view, and describes it. Therefore it stands to reason that we seek the feedback of a journalist to understand how well ELD performs this task.

The process to engage a journalist was a simple outreach effort. A public tweet called for journalists for a short interview, with a few reporters tagged in a reply. Of these, one journalist contacted us - Paul Doyle, football writer for The Guardian[8]. Initially, a former managing editor of FourFourTwo also reached out. However, since The Guardian journalists write timelines for football matches, we selected Paul Doyle.

In the reviewed literature, only one study conducted a similar interview - Marcus et al. [42]. The questions are naturally different, but we follow with a semi-structured interview. Shedding a rigid question-and-answer structure, such an interview is more akin to a discussion. Although we prepared a few questions, by letting the interviewee speak his mind, he was free to voice his concerns and steer the conversation to focus on ELD's shortcomings.

We split the interview into two distinct sections - an experiment, and an ensuing discussion, as described below. To accommodate the journalist's schedule, we sent all of the described timelines in advance, giving him ample time to go over them at his convenience. On the day, we conducted the telephonic interview using WhatsApp, obtaining prior permission to record the call to transcribe it later.

The qualitative analysis has two distinct purposes, translating into its two parts. The first aim is to get an idea of how ELD compares with curated systems. The second goal is to understand the direction that ELD and TDT in general needs to work towards.

To achieve the first objective, we ask Paul Doyle to compare ELD directly with a mainstream media website. We choose the match between Liverpool F.C. and S.S.C. Napoli, generating the real-time timeline for it using the DGS algorithm based on its full dataset. We compare ELD with the corresponding timeline by This Is Anfield[9].

This Is Anfield is a popular Liverpool F.C. fan website with a writing style that betrays an undeniable bias towards the English club, not unlike ELD. Although it contains no profanity or Twitter fragments, like mentions, it is still an informal blog with liberal use of emojis.

[8] `https://web.archive.org/web/20190209132410/https://www.theguardian.com/profile/pauldoyle`, last accessed February 9, 2019

[9] `https://www.thisisanfield.com/2018/12/live-liverpool-vs-napoli-follow-the-reds-crucial-champions-league-clash-here/`, last accessed on February 9, 2019

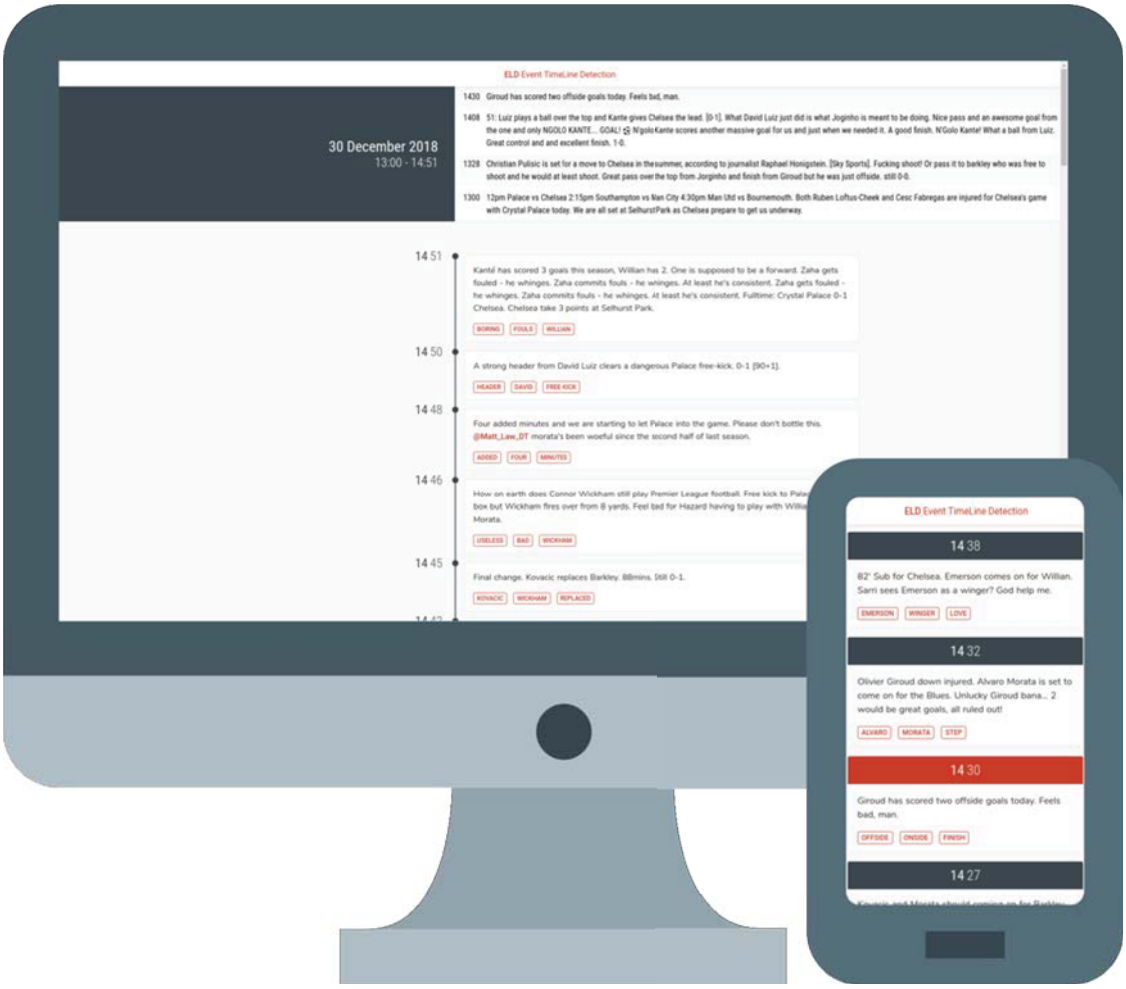

Figure 6.1: The frontend with the timeline that we gave to Paul Doyle.

We sent the two unedited textual timelines to Paul Doyle side-by-side in a spreadsheet in advance. During the experiment, we asked him whether he could tell the timelines apart before revealing the correct answer.

In the second part of the interview, we show him a timeline from the match between Crystal Palace F.C. and Chelsea F.C., constructed in similar fashion to the experiment. We host the second timeline on a frontend website, shown in Figure 6.1. We use these two timelines to understand how ELD performs in three different areas.

The first set of questions seek to understand how well the timelines provide coverage of the events' landscapes, analogous to TDT. With the second set of questions, we look at how well the individual developments are described by the summarization components. The final set of questions revolve around the applicability of systems like

ELD to journalists, thereby tying all the components together.

This subsection concludes the evaluation plan, with the described strategy enacted in the next section. We present and discuss the results that arise from this plan, thus exploring the strengths and weaknesses of ELD in the context of our aims and objectives.

6.3 Results

6.3.1 Automatic Participant Detection

It can be difficult to understand an event before it has even started, but that is the task that APD takes on through an understanding of the event's domain. In this subsection, a focus on the environments of various domains serves to answer how possible this is, even though the dependency on user participation shows through.

Football is one very narrow field, but the context surrounding each and every match is unique. The first results in Table 6.2 show the varying degrees of success for different events. Precision and recall fluctuate together in APD's first results. We attribute a big chunk of the algorithm's failings in precision to players that are not participating, perhaps ruled out due to suspension or injury. More commonly, the algorithm captured adjacent concepts, bringing down the precision score.

In spite of its shortcomings, our APD algorithm showed improvements over the baseline. The poor precision and recall scores are a symptom of an ailing NER performance over the conversations of Twitter. We test the statistical significance of the results on the football datasets using one-tailed Paired Sample T-Tests using the following hypotheses, applied analogously to precision and recall at a confidence level of 95%:

$$H_0 : \textit{results remain the same}$$
$$H_1 : \textit{results improve with the APD algorithm}$$

It turns out that our APD algorithm's improvements are statistically-significant in both precision and recall, with p-values of 0.04 and 0.006 respectively. In fact, the only case when the baseline out-performed APD was when the understanding period yielded confounding information, as described further down.

What is interesting in the results on football datasets is APD's recall score, often just below the 0.5 mark. This figure is anything but incidental. Capturing slightly less than half the participants in an event is a reflection of Twitter's conversations. Like the

	Precision		Recall	
Event	**ELD**	**Baseline**	**ELD**	**Baseline**
Manchester United F.C. - Arsenal F.C.	0.4286	0.3800	**0.4878**	0.3171
Liverpool F.C. - S.S.C. Napoli	**0.4762**	0.2400	0.4634	0.2683
Valencia CF - Manchester United F.C.	0.3778	0.3000	0.4146	0.3171
Liverpool F.C. - Manchester United F.C.	0.4681	0.3400	0.4634	0.3171
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	0.1905	0.2600	0.1951	0.2195
Crystal Palace F.C. - Chelsea F.C.	0.3400	0.2200	0.4146	0.2195
UFC 232	0.0698	0.1200	0.2000	0.4000
Brazil Presidential Election 2018	0.3824	0.1800		
USA Midterm Elections 2018	0.2321	0.3800		

Table 6.2: Base performances of ELD's APD component and the baseline in terms of precision and recall.

rest of the event, the discussions that preceded events were unreservedly biased, which influenced the extraction and extrapolation phases of APD.

The initial, resolved participants gave a biased example set to the extrapolation phase. This led to extrapolated actors that covered almost the entirety of the majority side, leaving little space for the minority team. This was also the case when both teams were behemoths, as was the case in the match between Liverpool F.C. and Manchester United F.C. - one side consistently came out on top.

The algorithm conceded to Twitter's bias, but it did signal the potential of the APD algorithm. When the context was clear, participation resolution yielded enough examples to come close to completing one class. This indicates that a shift towards identifying this split in bias to process the classes separately could bridge this gap.

Table 6.3, which breaks down the evolution of precision, demonstrates the importance of the extraction and resolution stage. Amid a trend of ever-decreasing precision, the baseline was especially weak in the first 10 results. Multiple factors contributed to these results, ranging from Twitter's conversation style to a struggling NER algorithm.

Table 6.4 confirms that our APD approach ranked valid participants higher than the baseline. The algorithm's results were consistently an improvement over the baseline, even in the match between Tottenham Hotspur F.C. and Wolverhampton F.C. - a match

	P@10		P@20		P@30	
Event	**ELD**	**Baseline**	**ELD**	**Baseline**	**ELD**	**Baseline**
Manchester United F.C. - Arsenal F.C.	0.5000	0.3000	0.6000	0.4500	0.5667	0.4667
Liverpool F.C. - S.S.C. Napoli	0.6000	0.5000	0.5500	0.3000	0.5333	0.2667
Valencia CF - Manchester United F.C.	0.5000	0.5000	0.4500	0.3000	0.4333	0.3333
Liverpool F.C. - Manchester United F.C.	0.7000	0.3000	0.5500	0.3500	0.5333	0.3667
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	0.5000	0.4000	0.4000	0.3500	0.2667	0.2667
Crystal Palace F.C. - Chelsea F.C.	0.6000	0.3000	0.4000	0.3000	0.4000	0.3000

Table 6.3: The evolution of precision in the rankings that the APD techniques produced.

that performed poorly overall, both in precision and recall.

As this match demonstrated, APD was not always successful, even in detecting the majority class. The match between Tottenham Hotspur F.C. and Wolverhampton Wanderers F.C. was at the same time as four other matches. It kicked off early in the afternoon, and it was not the most eagerly awaited fixture of the week-end, resulting in only mild buzz.

Although the small understanding corpus is comparable to the one from the match between Valencia CF and Manchester United F.C., these factors influenced the results. In the Manchester United F.C. case, many English-speaking supporters focused almost uniquely on the Champions League match. On the other hand, the Tottenham game was seen in the context of multiple matches being played at the same time.

In the end, the resolver captured the teams involved elsewhere and misunderstood the context to be the Saturday matchday. The extrapolated results built on this notion and the majority of returned participants were other English teams. It is not only in football that we observed this behaviour.

The dataset collected during the UFC 232 event focused specifically on the main card. The high profile section of fights followed two other phases, each with their own

	Average Precision (AP)	
Event	**ELD**	**Baseline**
Manchester United F.C. - Arsenal F.C.	0.6571	0.4507
Liverpool F.C. - S.S.C. Napoli	0.5656	0.5248
Valencia CF - Manchester United F.C.	0.5200	0.4788
Liverpool F.C. - Manchester United F.C.	0.6081	0.3944
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.6958**	0.4173
Crystal Palace F.C. - Chelsea F.C.	0.5560	0.3762
Mean Average Precision (MAP)	0.6004	0.4404

Table 6.4: Average Precision of automatically-detected participants in football matches.

fights. Thus, the understanding phase coincided with the phase that came before it - the preliminary card. The captured fighters reflected this confused context.

As a result, ELD's APD approach captured all the participants from the preliminary card, but only two of the ten fighters from the main card. In contrast, as shown in Table 6.2 on page 86, the baseline kept reading beyond APD's cut-off point in the local context. By considering more named entities, it correctly identified four main card fighters. The improvement is slight, but it indicates that in some cases, the understanding period itself may not suffice to understand an event.

The results do not just expose the insufficiencies of the understanding period in developing contexts, but it also admonishes the extrapolation phase. In spite of having captured and resolved UFC fighters, the extrapolation step failed. Instead of surfacing additional athletes, it returned related events.

More than just a misinterpretation of the event, this fault revealed the flaws of an algorithm that depends on a tightly-connected underlying mesh. Unlike football players, inter-connectedness between fighters on Wikipedia is much less common, leading to struggles to identify communities, and thus other athletes.

The peculiarities of an ill-defined domain are observable in other scenarios as well. The relatively simple structure of the Brazilian election saw Jair Bolsonaro and Fernando Haddad contest the presidency alongside their running mates. The results from ELD's APD algorithm were an echo of media coverage in the English-speaking world.

Bolsonaro was the eventual winner and source of worry for many dissidents worldwide, and our APD algorithm captured him with ease, unlike his opponent. The base-

line did manage to identify Haddad, but his name only cropped up in the 24th position. The results demonstrate the polarizing character and policies of Bolsonaro, which ultimately dictated conversations and biased the understanding phase.

In the USA, the complex structure of multiple elections that were held simultaneously was a more trying task. The specificity of certain regional elections was overshadowed by the broad coverage of the seed set. In their stead, the resolver recognized the main actors that characterized election night of 2018.

The midterms too bore testament to the bias on a large-scale basis. Although the Republicans suffered heavy defeats, ELD's resolver captured the Republican Party, President Donald Trump, and other individuals for whom the result could have spelt trouble. In such cases, extrapolation missed candidates because of the wide coverage of the event. Instead, the extrapolator looked for adjacent concepts.

In the Brazil election, the extrapolator identified concepts like *President of Brazil* and *Vice President of Brazil*. The fragmented space of the USA midterm elections complicated matters as shown in Table 6.2 on page 86. Nonetheless, the extrapolator was still capable of capturing concepts linked to the election, such as *Republican Party of Wisconsin*.

The results achieved by ELD's APD algorithm expose a clear sensitivity to the way Twitter views events themselves. Although prone to bias, the results from the football domain also demonstrate APD's potential. Other areas have to be understood further. Most evident are the algorithm's struggles in more confusing environments, which indicates that the approach might need to be tailored for different applications. Next, we analyse the effects of APD and participants in events through the evaluation of TDT.

6.3.2 Topic Detection and Tracking

There is a lot more that goes on during an average football match than just the fact-based moments. It is also an emotionally-charged context in which the TDT algorithm operates, and it shows. This subsection demonstrates just how much bearing this environment has, and how a changing landscape of discussion influences TDT techniques.

The precision and recall metrics give a general indication of the performance of the algorithms. The FMMR and DGS timeline construction techniques both started from the same places, but the results in Table 6.5 show how they went their own ways when it came to describing developments.

What stands out immediately is the small dip in precision when APD was factored into the corpus, both in the FMMR and DGS timelines. Depending on the distribution of differences, we performed one-tailed Paired Sample T-Tests and Wilcoxon Signed-Rank

Tests to analyse the statistical significance of the drops in precision at a confidence level of 95% with the following hypotheses:

$$H_0 : \textit{precision remains the same}$$
$$H_1 : \textit{precision dips in-between corpora}$$

Comparisons between the full and APD corpora reject the null hypothesis for the FMMR timeline, but not for the DGS approach, with p-values of 0.0099 and 0.0604 respectively. The inverse is true when comparing the baseline corpus with the APD dataset. The increase without APD is not statistically-significant in FMMR, but significant in DGS with p-values of 0.1011 and 0.0034 respectively.

We note that although precision does dip in some cases, it remains consistently high, dipping below 0.75 just once in ELD's approaches. Even though the precision of the APD algorithm was consistently less than 0.5, TDT managed to guard against the resulting noise. One possible explanation is that the erroneous participants seldom broke out during an event.

In other cases, just one out of every five developments were redundant, which we contrast with the baseline. Zhao et al.'s algorithm struggled to come close to ELD. Its dynamic window approach captured the same development multiple times when it broke, even though there was not enough novelty to justify it.

Other times, an overly-sensitive approach included too much noise. This observation was also made by Zubiaga et al., who used Zhao et al.'s technique as a baseline. During phases of low activity, Zhao et al.'s volume-based approach was prone to detecting false developments. In these cases, the only indication that pointed to a breaking development was an even calmer period preceding a tentative moment [72].

In spite of the high precision values, we observe that Twitter commentary did not always have a specific temporal focal point. Narration is one of the features that consistently appeared in timelines, even in the baseline. That is not to say that these updates were pointless because they described the general activity in the match.

For example, during the match between Valencia CF and Manchester United F.C., the English supporters discussed the poor performance in the context of the then-dismal situation at the club. At other times, the performances of individual players emerged repeatedly during an event.

Bias is often associated with the differences among the teams themselves, such as the contrast between an English and a foreign team. However, the evaluation also exposed

Method	Precision	Recall	F1
$FMMR_{Full}$	**0.8238**	0.6136	0.7033
$FMMR_{APD}$	0.7415	0.635	0.6841
$FMMR_{Base}$	0.7992	0.4738	0.5949
DGS_{Full}	0.8144	**0.6739**	**0.7375**
DGS_{APD}	0.7612	0.6663	0.7106
DGS_{Base}	0.8309	0.5236	0.6424
$Baseline_{Full}$	0.4843	0.3278	0.3910
$Baseline_{APD}$	0.3620	0.2410	0.2894
$Baseline_{Base}$	0.4169	0.4574	0.4362

Table 6.5: Macro-average precision and recall, and F1 results of the APD techniques across all football matches based on the various datasets.

certain actors that hogged attention. Whether a controversial foul or a glaring mistake, some actions kept finding a way back into Twitter discussions.

What set the three algorithms apart was the recall, which measured the method's ability to extract the salient moments from among the narration. Generally, the FMMR and DGS-based methods out-performed the baseline. The only exceptions came in corpora that were not boosted. However, we note that in this case, a very low precision score accompanied the baseline's high recall.

A steep improvement in recall was evident in our own techniques as the corpus grew and became more refined. More interestingly, the recall value hit a plateau with the APD corpus. We analysed the significance of the evolution of recall results among the corpora's construction methods similarly to before, using the hypotheses:

$$H_0 : \textit{recall remains the same}$$
$$H_1 : \textit{recall increases in-between corpora}$$

The paired statistical tests between the FMMR timelines based on the full and APD datasets failed to reject the null hypothesis with a p-value of 0.3274. The results were similar for the DGS algorithm - a p-value of 0.3368 failed to show any significant difference between the two datasets' results. The results flipped when comparing the APD

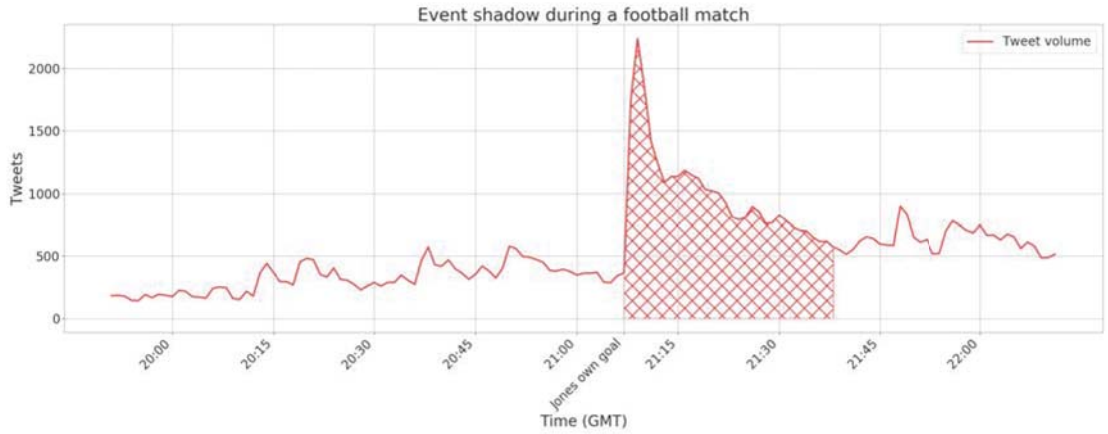

Figure 6.2: Jones' own goal sparked high-volume discussions that took a long time to subside.

and baseline corpora. This time, both the FMMR and DGS timelines rejected the null hypothesis with p-values of 0.0069 and 0.0099 respectively.

In other words, whereas tracking the participants found by the APD algorithm improved results, adding the missing actors to these entities did not contribute significantly. This climb in results confirms that the bias observed in APD persists in TDT.

The APD algorithm presented in ELD consistently captured most of the majority class while ignoring the minority class. The minimal improvement in recall when we considered the smaller class to build the full corpus points towards a gulf between the teams. The difference in discussions among the two teams seemed too far to reconcile just by tracking all participants, possibly necessitating a different TDT approach.

Interestingly, Zhao et al.'s approach saw reversed results, with the best recall value coming with the baseline corpus. This was partly due to a phenomenon wherein emotional developments kept volume high for an extended period of time, described by Lanagan and Smeaton as an "event shadow" [129].

For example, Phil Jones' own goal in Manchester United F.C.'s match against Valencia CF raised volume in its aftermath, as shown in Figure 6.2. It took almost half an hour for the volume to reach the previously-observed maximum rate.

This phenomenon kept the volume relatively stable on these occasions and complicated matters for volume-based approaches, such as Zhao et al.'s. The tweeting rates did not exhibit as sharp a rise amid the background hubbub and noise, unlike the defined spikes in Nichols et al.'s datasets from almost a decade ago [14].

Small developments during this event shadow did not alter the dynamics enough for the baseline to capture them. This is why the smaller corpora with more volatile

Match	Precision	Recall
Manchester United F.C. - Arsenal F.C.	0.8095	0.6364
Liverpool F.C. - S.S.C. Napoli	**0.8494**	0.6961
Valencia CF - Manchester United F.C.	0.7852	0.4216
Liverpool F.C. - Manchester United F.C.	0.8253	**0.7604**
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	0.7356	0.3922
Crystal Palace F.C. - Chelsea F.C.	0.7661	0.6795

Table 6.6: Macro-average of the precision and recall metrics of ELD's models across all datasets separated by football match.

tweeting rates led to a better performance, and why volume-based approaches may be outdated in today's Twitter.

On the other hand, our TDT approaches were impervious to this baseline of tweet volume that appeared consistently throughout the event. Instead, its flaw came out in its struggle to keep up when there were far too few tweets to create sizeable clusters.

In fact, results are highly-dependent on corpus richness. Table 6.6 shows how ELD's top three precision results across all algorithms coincided with the three biggest corpora. Two of the best three recall results also came from this selection of corpora - an abrupt reminder that ultimately, it is difficult to take away TDT's dependency on Twitter's description of events.

Volume is not the only factor that affects recall. Literature shows its weakness in certain types of actions that make up events. Table 6.7 shows how this trend is ever-present in ELD. Our algorithms captured almost every single goal situation. The only exceptions were disallowed goals and one of two very similar goals in the match between Liverpool F.C. and Manchester United F.C.

Stoppages performed slightly worse, with the first- and second-half kick-offs performing the worst in this section. These occasions followed quiet moments, and perhaps Twitter users had nothing to recap, as opposed to the ends of both halves.

Punishable offences and substitutions were the two types of events that benefited the most from the full corpora when compared to the baseline datasets. The FMMR algorithm captured twice as many yellow cards, whereas DGS saw a boost of 50%. Substitutions saw increases of 44% and 111% in the FMMR and DGS approaches respectively.

Notwithstanding these improvements, the two types of moments lagged behind

Method	Stoppages	Goal Situations	Offences	Substitutions
$FMMR_{Full}$	0.7500	0.9130	**0.4348**	0.4063
$FMMR_{APD}$	0.6667	**1.0000**	0.3913	0.5000
$FMMR_{Base}$	0.7083	0.9130	0.2174	0.2813
DGS_{Full}	0.7917	0.9565	0.3913	**0.5938**
DGS_{APD}	**0.8333**	0.9565	**0.4348**	0.4688
DGS_{Base}	0.6667	0.9565	0.2609	0.2813
$Baseline_{Full}$	0.3750	0.6087	0.1739	0.1250
$Baseline_{APD}$	0.2500	0.5217	0.0435	0.1250
$Baseline_{Base}$	0.6250	0.7826	0.1304	0.2500

Table 6.7: Recall breakdown by development type using the different APD techniques.

other developments. The results confirm those found elsewhere in popular literature - some developments are quite unpopular - but we make some observations that are responsible for these trends.

Primarily, not all yellow cards are equal. A dangerous or controversial offence was more likely to fuel conversation. Within the limited scope of these six matches, we note that even the time of the offence seemed to influence results.

Generally, late substitutions and yellow cards had lower recall rates than those that happened earlier during the game, and it is only intuitive. Yellow cards and substitutions towards the end of the game were unlikely to affect proceedings beyond a temporary pause in play.

For example, in the dying moments of the match between Liverpool F.C. and S.S.C. Napoli, two players picked up yellow cards. Although both were part of the majority class, ELD failed to detect the developments. Instead, the impending end of the match and the Champions League qualification that came with it overshadowed these actions.

More intuitively, the recall of developments from corpora boosted by APD exhibited a dependency on the detected participants. Focusing only on the APD corpora, in Table 6.8 we present a breakdown of recall results according to how many of the participants were being tracked in the APD corpus. We split yellow and red cards, as well as substitutions into the number of their participants that were being tracked in the APD corpus.

One participant at a time is shown a yellow or a red card. Thus, we split these

	Offences		Substitutions		
Method	All	None	All	Partial	None
$FMMR_{APD}$	0.4545	0.3333	0.6364	0.5000	0.4000
DGS_{APD}	0.6364	0.3333	0.7273	0.3333	0.3333

Table 6.8: Recall breakdown in two development types based on whether our APD algorithm detected all, some or none of the involved participants.

moments into whether that participant was being tracked – *All* – or not – *None*. Since substitutions involve two participants, we split these moments into whether or not the two participants were being tracked – *All* – or whether just one of the two was being tracked – *Partial* coverage – or whether none of them were being tracked – *None*.

Table 6.8 confirms that developments were more likely to be captured if their participants were detected. This also applied when one of the two participants involved in a substitution was captured by APD. In other words, when APD was able to extract participants and track them for the duration of the event, it was more likely that related moments would be captured.

In many cases APD exacerbated the bias towards the majority side by tracking its participants. Nonetheless, we note that the bias towards the majority side contributed indirectly to the minority class, though to a lesser extent. This arose because many situations involved participants from both teams.

For example, Davinson Sánchez was one of the few participants who were captured by our APD algorithm during the match between Tottenham Hotspur F.C. and Wolverhampton Wanderers F.C. This led to ELD producing the following summary:

> 65' Wolves break through Cavaleiro but Sanchez manages to stop the dangerous counter-attack. 1-0. Great defending from Sanchez, he's looked sharp today!

ELD improved Wolves' coverage through Sánchez's contribution, even though he was not behind the attack. The bias is still visible in the way that that summary elevates the defender's contributions above anything else. However, we highlight the fact that bias implicitly improved the overall quality of TDT, while transferring the challenge of impartiality to summarization.

TDT systems are pointless if they're not capable of describing developments. The next subsection looks beyond what moments ELD is capable of extracting, and turns its attention towards how well it explains the salient developments of an event.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0958	0.1205	0.1006	0.1423
Liverpool F.C. - S.S.C. Napoli	0.1246	0.1556	0.1202	0.1786
Valencia CF - Manchester United F.C.	0.0636	0.0961	0.0623	0.0882
Liverpool F.C. - Manchester United F.C.	0.1264	0.1328	0.1373	0.1299
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.2569**	**0.2817**	**0.2553**	**0.3658**
Crystal Palace F.C. - Chelsea F.C.	0.2062	0.2235	0.2315	0.2796
Macro-Average	0.1456	0.1684	0.1512	**0.1974**

Table 6.9: ROUGE-1 F1 results based on authoritative Twitter accounts.

6.3.3 Summarization

In ELD, we tie TDT and summarization closely together, with the former guiding the latter in what the description should include. This subsection looks at the quality of summaries generated by FMMR and DGS. For brevity, we present only F1 results here, with the full precision and recall results reported in Appendix D.

The results in both Tables 6.9 and 6.10 demonstrate that the FMMR and DGS algorithms led to improvements over the baseline in terms of unigrams. Throughout this subsection, we use the following hypotheses to test the significance of improvements using Wilcoxon Signed-Rank Tests and Paired Sample T-Tests:

$$H_0 : \textit{F1 score remains the same}$$
$$H_1 : \textit{F1 score increases}$$

All of the F1 improvements were statistically significant at a confidence level of 95%. FMMR's and DGS' boosts over their respective MEAD comparisons in the evaluation that considered The Guardian's timelines came away with p-values of 0.0109 and 0.0022 respectively. When compared to authoritative Twitter accounts' tweets, FMMR and DGS obtained p-values of 0.0011 and 0.0168 respectively.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.1224	0.1599	0.1225	0.1659
Liverpool F.C. - S.S.C. Napoli	0.1243	0.1651	0.1249	0.1868
Valencia CF - Manchester United F.C.	0.1431	0.1860	0.1408	0.2064
Liverpool F.C. - Manchester United F.C.	0.1673	0.1702	0.1654	0.1848
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.2688**	**0.2698**	**0.3109**	**0.3269**
Crystal Palace F.C. - Chelsea F.C.	0.1796	0.2072	0.1899	0.2330
Macro-Average	0.1676	0.1930	0.1757	**0.2173**

Table 6.10: ROUGE-1 F1 results based on mainstream media reports.

A look at the recall results on Twitter and mainstream media reports in Tables D.5 and D.6 on pages 123 and 124 respectively yields a contrast between conversation styles. ELD's output is much more similar to Twitter's way of conversing than it is to mainstream media. From the lack of length restrictions to more formal writing, the media talks completely different from Twitter, and these aspects showed through in ELD. Nonetheless, media timelines, longer than tweets, resulted in higher precision and F1 for ELD and MEAD.

Interestingly, ELD struggled in terms of recall when compared to MEAD in the last two matches, which comprised two of the smallest datasets. We observe that the lack of attention paid to these games promoted retweets from these accounts to the point that MEAD picked them over other tweets.

The same observation explains the high recall values achieved by MEAD and ELD's approaches. In contrast, bigger matches generated diverse conversations that dwarfed retweets. Faced with this indecision, MEAD opted for microblogs that did not describe the development as well as ELD's chosen tweets.

TDT's granularity also shifted more responsibility to summarization to describe the fine moments in detail. The inclusion of clusters with as few as 2 documents meant that algorithms struggled to find descriptive content to explain developments. At times, this was simply down to the moment being vague; other times it was too innocuous.

This observation on its own is not enough to explain the drop between high- and

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0239	0.0304	0.0258	0.0317
Liverpool F.C. - S.S.C. Napoli	0.0329	0.0299	0.0316	0.0435
Valencia CF - Manchester United F.C.	0.0235	0.0374	0.0232	0.0246
Liverpool F.C. - Manchester United F.C.	0.0488	0.0600	0.0556	0.0498
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.1320**	**0.1102**	**0.1268**	**0.148**
Crystal Palace F.C. - Chelsea F.C.	0.1076	0.0935	0.1232	0.1156
Macro-Average	0.0615	0.0602	0.0644	**0.0689**

Table 6.11: ROUGE-2 F1 results based on authoritative Twitter accounts.

low-activity matches. In fact, token recall did not increase simply by virtue of the voluminous discussions of high-profile matches.

Although longer and supposedly more informative, the machine summaries struggled with recalling all of the important tidbits as reported by authoritative accounts or the media itself. Once again, a shift in discussion could be the reason behind this drop.

Unlike authoritative accounts or the media, supporters are neither bound by objectivity nor formalism. Subjectivity was evident when reading the timelines generated by ELD, but greatly limited in media and curated Twitter accounts. Similarly, emotions are sparse in formal writing, but Twitter users need not adhere to these standards.

MEAD occasionally out-performed FMMR and DGS in recall by virtue of constantly constructing summaries with around 50 words. In this way, MEAD included a lot of repetition, which greatly brought down the precision score, even in the small datasets of Tables D.1 and D.2 on pages 121 and 122. In contrast, FMMR and especially DGS could decide better when to stop adding documents to a summary.

More popular matches generated enough content for the DGS and FMMR algorithms to explore different facets of discussions. However, when few accounts were tweeting about developments, the low volume reduced conversation to a narrow scope, which led to shorter summaries than MEAD's. In turn, FMMR's and DGS' precision results shot up, with the latter being much stricter in reducing redundancy.

The results carried over into ROUGE-2. Naturally, both Twitter- and media-based evaluations, presented in Tables 6.11 and 6.12 respectively, exhibited a drop in F1 score

when considering bi-grams, as opposed to uni-grams.

The small number of datasets combined with the closeness of results meant that none of the differences were statistically significant when using the above hypotheses. FMMR's and DGS' performances in the Twitter-based evaluations when compared to MEAD received p-values of 0.422 and 0.1798 respectively. Similar results followed in the mainstream media-based evaluation, with FMMR's and DGS' differences from MEAD getting p-values of 0.2282 and 0.2348 respectively.

Beyond the general trend of drops from unigram results, many of the observations remain the same. This time, a stricter approach when evaluating word order saw the algorithms under-performing in recall on average when compared to MEAD. The results in low-activity matches were largely responsible for the drop in average recall of Tables D.7 and D.8 on pages 124 and 125, but ELD recovered in more popular matches and obtained comparable scores with MEAD.

This seems to further confirm MEAD's bias towards picking highly-retweeted content over diversity. On the other hand, the evaluation on media reports used a completely different medium and it did not promote this bias. Both of ELD's algorithms struggled when content was scarce, but outdid MEAD as volume grew.

Nevertheless, the improvements in ROUGE-2 recall results on small datasets betrayed a predilection for quality content. This observation points towards a need for document scoring with preference for content posted by more authoritative users and whose mannerisms are closer to those of the media.

Interestingly, DGS largely out-performed the FMMR algorithm in the media-based evaluation, with boosts of 1.38% and 0.98% in the ROUGE-1 and ROUGE-2 recall results of Tables D.6 and D.8 on pages 124 and 125. The gaps in precision were even wider at 4.66% for uni-grams and 1.7% for bi-grams, as shown in Tables D.2 and D.4 on pages 122 and 123. The improvements in DGS indicate that there could be a case for summarization to spend more time understanding documents to improve coverage.

Above all, the one aspect that separates DGS and FMMR from MEAD is their query formulation. Rather than provide a context-less corpus to the summarizer, the TDT algorithm dictated to it what topical keywords are important to a development.

The results in this subsection cautiously point towards the benefits of this decision on machine summaries. This was especially true when the context was muddled up as a side-effect of voluminous corpora. Simultaneously, the results showed the importance of a granular TDT approach to not only recognize that a development has occurred, but also understand and explain *why* it occurred.

The next subsection moves on from the quantitative evaluation and looks at the timelines' qualities from a more grounded perspective. We examine the totality of ELD's

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0310	0.0249	0.0317	0.0368
Liverpool F.C. - S.S.C. Napoli	0.0211	0.0202	0.0211	0.0293
Valencia CF - Manchester United F.C.	0.0430	0.0621	0.0432	0.0629
Liverpool F.C. - Manchester United F.C.	0.0440	0.0420	0.0426	0.0419
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.0787**	**0.0635**	**0.0948**	**0.0798**
Crystal Palace F.C. - Chelsea F.C.	0.0585	0.0348	0.0606	0.0651
Macro-Average	0.0461	0.0413	0.0490	**0.0526**

Table 6.12: ROUGE-2 F1 results based on mainstream media reports.

performance by performing a qualitative evaluation with Paul Doyle to examine its practical shortcomings and values.

6.3.4 Qualitative Analysis

Do thousands of Twitter users compare to news agencies? When we contacted Paul Doyle to conduct an interview, we wanted to understand how close ELD comes to describing events. The answer comes not in terms of numbers, but an expert opinion with the high standards of a professional. We present the entire transcript in Appendix E.

In the first experiment, we sought the answer to a very simple question to start off the interview - which timeline was generated by ELD, and which one was written by the staff at This Is Anfield? In spite of the erratic discussions of Twitter, when Paul Doyle was presented with the unlabelled write-ups, he could not discern between the two.

The answer indicates that, at least for the single match we showed him, ELD comes close enough to be indistinguishable from This Is Anfield and its human-curated timeline. Nonetheless, he identified critical faults in both of them, ultimately qualifying them as being "as bad as each other."

In light of this answer and the perceived similarity between This Is Anfield and ELD, we asked Paul Doyle to consider our system to be an intern, or a trainee reporter. In the rest of the interview, we could understand better what he was looking for that both timelines missed so sorely.

It turned out that the issues were ingrained in the mannerisms when describing developments. In fact, when we questioned him about the event coverage, the TDT component seemed to be ELD's principal quality. Very few missing details seemed to jump out at him. He only singled out one node on the timeline that mentioned an unrelated participant due to surname ambiguity.

However, overall he seemed to think that ELD mapped the evolution of the event correctly. Paul Doyle explained that was able to "see a general sense of overall understanding of the subtleties of how the match changed." This was thanks no little to ELD's ability to capture the highlights of the match. Rather than seeing faults in TDT, ELD's failings were in summarization.

As a journalist, Paul Doyle was more interested in the in-depth movements of a football match. "I did not see much tactical analysis in there. I did not see much indication of how substitutions may have changed matches," he explained. From this perspective, the issues were more about how moments were described than whether or not they were captured.

This aspect came out especially when he focused on the individual nodes on the timeline. He found that the summaries seemed like the personal musings of users. He speculated that the incomplete information was a failure of Twitter that made it into ELD. "On Twitter most people are assuming that people can see the action. They don't give a detailed description of exactly what happened," he told us.

The one thing that stuck out most during our interview was the subjectivity in some of the nodes - perhaps the one thing which makes This Is Anfield and ELD so similar. In this segment, Paul Doyle contrasted our system with more robotic systems, which he described as "blunt and boring," but also accurate.

Conversely, he said a timeline that is made up of subjective judgements with very little objectivity means that "you have no idea whether it is accurate or not." In his opinion, such a timeline is even more useless than robotic summaries in terms of accuracy.

Elaborating on this point, he explained that the fault of subjective timelines is not knowing who is writing. In Paul Doyle's opinion, this bias would be fine, but the authors need to be known to make it easier to assign value to their judgements. Beyond subjectivity, he reduced other characteristics of the timeline to style decisions.

Paul Doyle explained that he is not opposed to emotion and profanity. Whereas the former is normally acceptable, he warned of "slightly incoherent emotions" in some of ELD's summaries. These gave away the fact that tweets from different people were stitched together. He found profanity similarly acceptable if used sparingly to preserve its impact, while encouraging narration in timelines to "enrich the commentary" and describe the context.

We also quizzed Paul Doyle on how ELD could be useful to journalists. He expressed concern that such systems could abolish journalists' jobs, and thus could see little use in the timelines. On the other hand, when we explained that ELD's descriptions of events are a reflection of what Twitter finds important, he reasoned that ELD could be useful as an analysis tool.

Nonetheless, similar to the conclusions of Marcus et al.'s [42] interview, Paul Doyle was reluctant to this idea due to Twitter's nature. He expressed concern that the social network is not representative of the community at large and is too extreme. According to him, that may be Twitter's biggest fault of all - being "a forum where people are deliberately seeking attention."

6.4 Summary

The general improvements that we presented in this chapter indicate that ELD's model to understand events before explaining them is not only desirable, but also possible. Far from being premature, we showed how the time before an event starts can lead to an understanding about the domain if it is well-defined.

This process showed its effects in the ensuing TDT task, which benefited greatly, improving granularity without compromising on precision. Small progress was also visible when the topic detection task was allowed to explain developments in conjunction with the proposed FMMR and DGS algorithms.

The qualitative analysis demonstrated that ELD has made meaningful progress in identifying the salient points of events. Nonetheless, the discussion heaped importance on constructing transparent and credible timelines - tasks for summarization. Going off these findings, the next chapter summarizes this dissertation and lays down proposals for future work.

7

Conclusion

When Kleinberg proposed the idea of burst in his seminal 2003 paper [86], it seemed like an elegant solution to the TDT research area. The simplicity of a two-state automaton that travelled back and forth between one normal and another agitated state was a sophisticated representation that explained the problem tersely.

Over the years, scientific research took Kleinberg's proposal almost too literally. The studies presented over the years reduced the TDT problem to the speed and resilience of noticing state changes to the breaking state. But their bids focused far too much on *when* the change happens, and not enough on *how* it happens.

ELD set out to create a TDT approach that could operate in real-time and with a fine granularity, exploiting all that microblogging has to offer. By the end ELD challenged the traditional definition of an event that ignores the intricacies of events and does not permit machines the possibility of understanding for themselves.

In the rest of this chapter, we first take a look at how ELD achieved its aims and the extent of their success based on the quantitative and qualitative evaluations. Subsequently, we suggest ways forward to improve ELD's performance before concluding the dissertation with the final remarks.

7.1 Achieved Aims and Objectives

The principle that drove ELD was that a richer understanding is not only desirable, but also necessary to sufficiently cover an event. We confirmed this hypothesis in Chapter 6, demonstrating that a football match is not confined to the period between two whistles. Supporters start thinking about their team well before the ball is kicked off and follow their teams arduously during the entirety of the match. The hypothesis held true across different domains, including politics.

It is for this reason that ELD seeks to understand events before they even start. The novel concept of APD demonstrates how the conversations leading up to an event paint a rough picture of the hours that would follow. ELD looks at the discussions as a preface to the event. In the rest of this section, we describe how ELD achieved the aims and objectives outlined in Section 1.2.

7.1.1 Determine how well participants can be identified before an event has started

Though the first goal was an ambitious one, in Subsection 6.3.1 we showed how our APD algorithm is able to gain an insight into the key participants of an event before it has even started. However, our method, so dependent on Twitter conversations, is also too easily misled.

When it worked, APD inherited the bias of Twitter, succumbing to coverage that favoured the majority side. When it did not work, it opened up to extraneous noise. Nonetheless, even with our first presentation of APD, we showed that Wikipedia can make up for the shortcomings of the insufficient Twitter stream.

7.1.2 Explore the ways in which APD contributes to topic detection

Using the found participants, our implementation of FIRE's TDT model, helped on by our novel feature-pivot algorithm, could pick out the fine details from event streams. This was evident in the improved recall results of Subsection 6.3.2. The same results showed that although APD displayed bias and imprecision, it impacted TDT negatively only marginally.

On the contrary, the TDT algorithm operating over biased APD datasets routinely climbed to the same heights as when using the full datasets. It is as encouraging a sign as it is eye-opening - APD in its current state can improve results to an extent that is similar to a manually-defined set of keywords. On the other hand, it points to a need for a sharper TDT approach that overcomes the bias of Twitter streams.

7.1.3 Identify the effects of the combined document- and feature-pivot approach on topic detection in a narrow stream

Our evaluation in Subsection 6.3.2 also demonstrated that unlike simpler methods, ELD is impervious to the conversation habits of Twitter. While volume-based approaches oscillate between the overly-sensitive and outright powerless when faced with odd tweeting habits, ELD looks for the saliency amid the noise.

TDT's contributions extend beyond development detection. The document-pivot method identifies the documents responsible for a shift towards an agitated state within the event. Our novel feature-pivot algorithm drills down even deeper to find the topical keywords around which the development revolves. Like APD, TDT seeks to understand developments to explain them better.

7.1.4 Examine the link between the fragmentation of topic detection and summarization, and how TDT can contribute to summarization

In ELD we reasoned that summarization builds atop of TDT, and thus the two should not be considered separate. The understanding of TDT becomes the core around which FMMR and DGS construct summaries that look at the documents as a mixture of perspectives.

By exploiting this enforced link between TDT and summarization in the evaluation in Subsection 6.3.3, our novel algorithms showed encouraging results when compared to the baseline. The analysis revealed a rift between the writing habits of regular Twitter users, authoritative accounts and the mainstream media.

This aspect also resulted in the most glaring downfall of our summarization approaches, further confirmed in the qualitative analysis in Subsection 6.3.4 - subjectivity. Though still succumbing to bias, ELD exhibited promise to detect developments even when data is scant, but the same cannot be said for summarization.

7.1.5 Analyse summarization's performance in describing developments

The final quantitative evaluation in Subsection 6.3.3 and the qualitative evaluation in Subsection 6.3.4 raised the unexpected question of what happens when descriptive content is scarce or unavailable. The interview heaped more importance on summarization's capabilities to explain developments in an impartial and transparent manner. For TDT systems to move closer to the mainstream media could mean looking beyond Twitter and its distinctive mannerisms. We explore avenues for future research in these areas in the next section.

7.2 Future Work

In this dissertation, we confirmed the principal hypothesis that a deeper understanding of an event presents opportunities for TDT and summarization to explain developments better. Although the quantitative and qualitative analyses confirmed this hypothesis, they also revealed shortcomings.

Deeply ingrained in Twitter, and ultimately in humans, bias is the most prominent issue of APD, albeit not as detrimental as feared. The resounding split between classes is not entirely negative. It means that when the resolved participants are predominantly one-sided, capturing the totality of this one class is within reach. Recognizing this divide and repeating the process for each side is one possible way of overcoming bias.

APD as presented in this dissertation is a concept with a lot of untapped potential. In ELD, we shielded TDT from the knowledge of APD, but the works of Shen et al. [13], and McMinn and Jose [18] demonstrate models that ingrained participants in their core workflows.

Even considered individually, our novel feature-pivot TDT algorithm was not exploited thoroughly in this dissertation. Literature often considers topic tracking only implicitly, yet we postulate that our novel feature-pivot TDT solution can be used in this regard. Its bounded burst opens up opportunities for tracking to be considered more explicitly by looking at drops in burst as an indication of a development's end.

Although bias in TDT remains a challenge that needs to be tackled, the interview with Paul Doyle indicated that Twitter's content is too superficial to bridge the gap with human-curated timelines. When Twitter fails to contribute the right material for timelines, where should systems look instead?

One avenue that has not been sufficiently explored in summarization literature is big data. External sources, such as media reports themselves, could not only serve as more comprehensive corpora, but they could also be the solution to more objective timelines.

7.3 Final Remarks

Before France and Croatia walked out onto the pitch for the 2018 World Cup final, there was a road crafted by the players. Those same players fought for World Cup glory - they scored, some celebrated and others wept.

While they played, the world talked about them - people talked because they understood what was happening, but an automated system does not. In ELD we demonstrated that perhaps machines and humans are not so far apart. In their own peculiar way, they both require more than instructions - an understanding to first reason about, and then explain events.

A

Algorithm Pseudocode

A.1 Automatic Participant Detection

Listing A.1: Extractor

```
Input: set of documents C
Output: set of documents as bag-of-candidates

document_entities = {}
for each document d in C do
    d = preprocess(d)
    entities = named entities in d
    document_entities.insert(entities)

return document_entities
```

Listing A.2: Scorer

```
Input: set of documents as bag-of-candidates C
Output: set of scored candidates

entity_scores = {}
for each document d in C do
    for each named entity ne in d do
        if ne in entity_scores then
            entity_scores[ne] += 1
        else
            entity_scores[ne] = 1

for each named entity, score (ne, s) in entity_scores do
```

```
13      entity_scores[ne] = log10(s + 1)
14
15  return entity_scores
```

Listing A.3: Filter

```
1   Input: set of scored candidates S
2          minimum score min_score
3          number of candidates k
4   Output: list of candidates
5
6   candidates = {}
7
8   S = candidates in S with score >= min_score
9
10  for each candidate c in S do
11      for each candidate other in S do
12          if other is a subset of c then
13              S[c] += S[other]
14              S.remove(other)
15
16  sort candidates in descending order
17  return top k candidates
```

Listing A.4: Resolver

```
1   Input: list of candidates E
2          corpus C
3          minimum score min_score
4   Output: list of resolved participants
5
6   resolved = {}
7
8   local_context = concatenate(C)
9   for each candidate c in E do
10      articles = searchWikipedia(c)
11      similarities = {}
12      for each article wikipedia_concept in articles
13          sentence = getFirstSentence(wikipedia_concept)
14          similarity = cosine(local_context, sentence)
15          similarities[wikipedia_concept] = similarity
16
```

```
17      candidate_concept = argmax(similarities)
18      if similarities[candidate_concept] > min_score
19          resolved.insert(candidate_concept)
20
21  return resolved
```

Listing A.5: Extrapolator

```
1   Input: list of Wikipedia concepts of participants P
2          corpus C
3          minimum score min_score
4          number of candidates k
5   Output: list of extrapolated participants
6
7   extrapolated = {}
8
9   local_context = concatenate documents in C
10  graph = new Graph
11
12  // first iteration
13
14  link_popularity = {}
15  first_edges = {}
16  for each Wikipedia concept p in P do
17      add node p to graph
18      articles = articles linked from p
19      first_edges[p] = articles
20
21      for each article a in articles do
22          if a in link_popularity then
23              link_popularity[a] += 1
24          else
25              link_popularity[a] = 1
26
27  popular_articles = top 100 most popular links
28  for each participant source in P do
29      target_edges = first_edges[source]
30      target_edges = edges in target_edges if in popular_articles
31      for each Wikipedia concept target in target_edges do
32          article_content = Wikipedia article of target
33          sentence = first sentence in article_content
34          similarity = cosine(local_context, sentence)
```

```
35
36          add node target to graph
37          if similarity > 0 then
38              add undirected edge source-target
39                          with weight similarity
40
41  // second iteration
42
43  link_popularity = {}
44  second_edges = {}
45  for each Wikipedia concept a in articles do
46      articles = articles linked from a
47      second_edges[a] = articles
48
49      for each article other in articles do
50          if other in link_popularity then
51              link_popularity[other] += 1
52          else
53              link_popularity[other] = 1
54
55  sort link_popularity in descending order
56  cut_off = popularity of concept at rank 1000
57  popular_articles = articles in link_popularity if count >= cut_off
58  for each article source in second_edges do
59      target_edges = second_edges[source]
60      target_edges = edges in target_edges if in popular_articles
61      for each Wikipedia concept target in target_edges do
62          article_content = Wikipedia article of target
63          sentence = first sentence in article_content
64          similarity = cosine(local_context, sentence)
65
66          add node target to graph
67          if similarity > 0.5 then
68              add undirected edge source-target
69                          with weight similarity
70
71  // extract participants
72
73  communities = extract communities from graph
74  communities = communities with at least 4 articles
75  articles = {}
```

```
for each community c in communities do
    for each article a in communities do
        article_content = Wikipedia article a
        sentence = first sentence in article_content
        similarity = cosine(local_context, sentence)

        if similarity >= min_score then
            articles[a] = similarity

return top k articles in articles
```

Listing A.6: Postprocessor

```
Input: list of participants P
Output: list of postprocessed participants

participants = {}

for each participant full_name in P do
    article = Wikipedia article of full_name
    if article has a DOB
        surname = last word in full_name
        if surname is not an English word then
            participants.insert(surname)
        else
            participants.insert(full_name)

return participants
```

A.2 Topic Detection and Tracking

Listing A.7: Temporal No-K-Means

```
Input: document d
       clusters C
       similarity threshold min_similarity
       freeze period freeze

// remove inactive clusters first
time = current time
for each cluster c in C do
```

```
    if time - c.last_updated > freeze then
        remove c from C

// compare the document with each cluster
similarities = {}
for each cluster c in C do
    similarity = cosine(d, c.centroid)
    similarities[c] = similarity

// cluster attribution
cluster, max_similarity = argmax(similarities)
if max_similarity >= min_similarity then
    cluster.add(d)
    cluster.last_updated = time
else
    cluster = new Cluster
    cluster.last_updated = time
    cluster.add(d)
    clusters.insert(cluster)
```

Listing A.8: Burst calculation

```
Input: keyword k
       keyword nutrition nutrition
       nutrition store store
       number of sets to consider s
       decay rate decay_rate

nutrition_sets = s recent nutrition sets
                    in reverse chronological order

denominator = 0
for i between 1 and s do
    denominator += (1 / math.exp(i) ^ decay_rate)

burst = 0
for i between 1 and s do
    historic_burst = nutrition_sets[i - 1].keyword
    decay = (1 / math.exp(i) ^ decay_rate)
    burst += (nutrition - historic_burst) * decay

burst /= denominator
```

```
21 return burst
```

A.3 Summarization

Listing A.9: New breaking development

```
1  Input: breaking cluster C
2         tuples of breaking terms and their scores t
3         timestamp time
4         time_window window
5         wait_period wait
6
7  current_node = last node on the timeline
8  if time - current_node.last_updated > window then
9      create new timeline node
10
11 max_similarity, closest_development = 0, none
12 for each node n in timeline do
13     development = new Cluster(n.breaking_documents)
14     similarity = cosine(C.centroid, development.centroid)
15     if similarity > max_similarity then
16         max_similarity = similarity
17         closest_development = n
18
19 if max_similarity > 0.6 then
20     add (C, t) to closest_development
21
22 if max_similarity <= 0.6
23     and (time - current_node.last_updated > wait) then
24
25     current_development =
26         new Cluster(current_node.breaking_documents)
27     similarity = cosine(C.centroid, current_development.centroid)
28     if similarity < 0.1 then
29         create new timeline node
30
31 current_node = last node on the timeline
32 if max_similarity <= 0.6 then
33     if current_node.clusters.length == 0 then
34         create timeline node anew
35
```

```
36      current_node.clusters.insert((C, t))
37      current_node.breaking_documents.insert(C.documents)
38      current_node.last_updated = time
```

Listing A.10: FMMR

```
1  Input: document set D
2         breaking clusters (with bursty terms and their burst) C
3  Output: summary object
4
5  // query formulation
6
7  query_cluster = new Cluster
8  for each cluster c in C do
9      size = number of documents in c
10     (terms, scores) = breaking terms in c
11
12     document = new Document
13     for each (term, score) in (terms, scores) do
14         document.term = score * size
15     query_cluster.add(document)
16
17 query = query_cluster.centroid.normalize()
18
19 return MMR(D, query)
```

Listing A.11: DGS

```
1  Input: document set D
2         breaking clusters (with bursty terms and their burst) C
3  Output: summary object
4
5  // query formulation
6
7  query_cluster = new Cluster
8  for each cluster c in C do
9      size = number of documents in c
10     (terms, scores) = breaking terms in c
11
12     document = new Document
13     for each (term, score) in (terms, scores) do
14         document.term = score * size
```

```
15      query_cluster.add(document)
16
17  query = query_cluster.centroid.normalize()
18
19  // graph creation
20
21  graph = new Graph
22  for each document d in D do
23      add d to graph
24
25      for each node n in graph do
26          similarity = cosine(d, node.document)
27          if similarity > 0.3 then
28              add undirected edge d-node with weight similarity
29
30  communities = extract communities from graph
31  if there are communities with at least 3 nodes
32      communities = communities with at least 3 nodes
33  else
34      communities = biggest community
35
36  summary = new Summary
37  for each community c in communities do
38      centrality_scores = eigenvector_centrality(c)
39
40      node_scores = {}
41      for each node n in c do
42          document = n.document
43
44          quality_score = score(document)
45          query_score = cosine(document, query)
46          centrality_score = centrality_scores[n]
47          node_scores[n] = quality_score *
48                                  query_score *
49                                  centrality_score
50
51      chosen_document = argmax(node_scores)
52      add chosen_document to summary
```

B

Evaluation Datasets

Event	Seed Set
Manchester United F.C. - Arsenal F.C.	*#MUNARS, Manchester United, Arsenal*
Liverpool F.C. - S.S.C. Napoli	*#LIVNAP, Liverpool, Napoli*
Valencia CF - Manchester United F.C.	*#VALMUN, Valencia, Manchester United*
Liverpool F.C. - Manchester United F.C.	*#LIVMUN, Liverpool, Manchester United*
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	*#TOTWOL, Tottenham, Wolves*
Crystal Palace F.C. - Chelsea F.C.	*#CRYCHE, Crystal Palace, Chelsea*
UFC 232	*#UFC232, UFC 232*
Brazil Presidential Election 2018	*Brazil election, Brasil election*
USA Midterm Elections 2018	*#MidtermElections2018, #ElectionDay, #ElectionNight, midterm*

Table B.1: The seed sets that we used to collect the different datasets.

Event	Understanding Dataset Collection (UTC)	Event Dataset Collection (UTC)
Manchester United F.C. - Arsenal F.C.	18:45-19:45 on 5 December, 2018	19:45-22:15 on 5 December, 2018
Valencia CF - Manchester United F.C.	18:45-19:45 on 12 December, 2018	19:45-22:15 on 12 December, 2018
Liverpool F.C. - S.S.C. Napoli	18:45-19:45 on 13 December, 2018	19:45-22:15 on 13 December, 2018
Liverpool F.C. - Manchester United F.C.	14:45-15:45 on 16 December, 2018	15:45-18:15 on 16 December, 2018
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	13:45-14:45 on 29 December, 2018	14:45-17:15 on 29 December, 2018
Crystal Palace F.C. - Chelsea F.C.	10:45-11:45 on 30 December, 2018	11:45-14:15 on 30 December, 2018
UFC 232	1:45-2:45 on 30 December, 2018	2:45-7:15 on 30 December, 2018
Brazil Presidential Election 2018	18:45-19:45 on 28 October, 2018	
USA Midterm Elections 2018	22:00-22:30 on 6 November, 2018	

Table B.2: The dates and times when we collected the datasets.

	Number of Tweets			
Event	Understanding	Base	APD	Full
Manchester United F.C. - Arsenal F.C.	41,586	121,329	183,459	212,729
Valencia CF - Manchester United F.C.	3,005	33,156	95,662	79,419
Liverpool F.C. - S.S.C. Napoli	15,841	93,761	153,927	165,132
Liverpool F.C. - Manchester United F.C.	29,785	186,116	290,696	303,982
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	3,563	67,726	95,547	93,223
Crystal Palace F.C. - Chelsea F.C.	8,937	36,968	75,770	63,891
UFC 232	15,457	195,519	223,273	249,805
Brazil Presidential Election 2018	841			
USA Midterm Elections 2018	41,668			

Table B.3: The number of tweets in the datasets that we collected.

C

Topic Detection and Tracking Evaluation Configuration

Match	Document-Pivot (Clustering)			Feature-Pivot	
	Minimum size	Threshold	Freeze period (s)	Time window (s)	Minimum burst
Manchester United F.C. - Arsenal F.C.	3	0.5	20	30	0.5
Liverpool F.C. - S.S.C. Napoli	3	0.5	20	30	0.5
Valencia CF - Manchester United F.C.	3	0.5	20	30	0.5
Liverpool F.C. - Manchester United F.C.	3	0.5	20	30	0.5
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	2	0.5	20	30	0.5
Crystal Palace F.C. - Chelsea F.C.	2	0.5	20	30	0.5

Table C.1: The configurations that we used to evaluate ELD's TDT component.

D

Summarization Results

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0694	0.0872	0.0735	0.1147
Liverpool F.C. - S.S.C. Napoli	0.0893	0.1258	0.0866	0.1570
Valencia CF - Manchester United F.C.	0.0401	0.0620	0.0388	0.0585
Liverpool F.C. - Manchester United F.C.	0.0926	0.0926	0.0992	0.0980
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.1822**	**0.2254**	**0.1816**	**0.3417**
Crystal Palace F.C. - Chelsea F.C.	0.1595	0.1692	0.1760	0.2490
Macro-Average	0.1055	0.1270	0.1093	**0.1698**

Table D.1: ROUGE-1 precision results based on authoritative Twitter accounts.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.1144	0.1455	0.1177	0.1841
Liverpool F.C. - S.S.C. Napoli	0.1055	0.1393	0.1064	0.1807
Valencia CF - Manchester United F.C.	0.1439	0.2172	0.1620	0.2208
Liverpool F.C. - Manchester United F.C.	0.1764	0.1648	0.1707	0.2054
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.2428**	**0.2321**	**0.2765**	**0.3539**
Crystal Palace F.C. - Chelsea F.C.	0.1623	0.1897	0.1688	0.2231
Macro-Average	0.1576	0.1814	0.1670	**0.2280**

Table D.2: ROUGE-1 precision results based on mainstream media reports.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0175	0.0211	0.0188	0.0242
Liverpool F.C. - S.S.C. Napoli	0.0223	0.0242	0.0214	0.0358
Valencia CF - Manchester United F.C.	0.0143	0.0230	0.0138	0.0156
Liverpool F.C. - Manchester United F.C.	0.0354	0.0430	0.0395	0.0365
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.0929**	**0.0882**	0.0895	**0.1378**
Crystal Palace F.C. - Chelsea F.C.	0.0862	0.0668	**0.0966**	0.1094
Macro-Average	0.0448	0.0444	0.0466	**0.0599**

Table D.3: ROUGE-2 precision results based on authoritative Twitter accounts.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0303	0.0223	0.0328	0.0485
Liverpool F.C. - S.S.C. Napoli	0.0169	0.0154	0.0169	0.0284
Valencia CF - Manchester United F.C.	0.0393	**0.0662**	0.0386	0.0671
Liverpool F.C. - Manchester United F.C.	0.0472	0.0410	0.0457	0.0467
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.0622**	0.0538	**0.0749**	**0.0824**
Crystal Palace F.C. - Chelsea F.C.	0.0512	0.0278	0.0521	0.0559
Macro-Average	0.0412	0.0378	0.0435	**0.0548**

Table D.4: ROUGE-2 precision results based on mainstream media reports.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.1705	0.2326	0.1763	0.2555
Liverpool F.C. - S.S.C. Napoli	0.2235	0.2257	0.2139	0.2410
Valencia CF - Manchester United F.C.	0.1646	0.2488	0.1841	0.2520
Liverpool F.C. - Manchester United F.C.	0.2236	0.2513	0.2542	0.2535
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.4491**	**0.3965**	**0.4424**	**0.4166**
Crystal Palace F.C. - Chelsea F.C.	0.3553	0.3501	0.4143	0.3595
Macro-Average	0.2644	0.2842	0.2809	**0.2964**

Table D.5: ROUGE-1 recall results based on authoritative Twitter accounts.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.1746	0.2154	0.1742	0.2500
Liverpool F.C. - S.S.C. Napoli	0.1904	0.2288	0.1904	0.2341
Valencia CF - Manchester United F.C.	0.1583	0.1995	0.1605	0.2120
Liverpool F.C. - Manchester United F.C.	0.1873	0.2011	0.1892	0.2188
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.3510**	**0.3344**	**0.4119**	0.3166
Crystal Palace F.C. - Chelsea F.C.	0.2809	0.2967	0.3084	**0.3271**
Macro-Average	0.2238	0.2460	0.2391	**0.2598**

Table D.6: ROUGE-1 recall results based on mainstream media reports.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0414	0.0638	0.0447	0.0586
Liverpool F.C. - S.S.C. Napoli	0.0650	0.0440	0.0624	0.0689
Valencia CF - Manchester United F.C.	0.0677	0.1094	0.0777	0.093
Liverpool F.C. - Manchester United F.C.	0.0843	0.1060	0.1050	0.0940
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	**0.2333**	0.1534	0.2225	**0.1678**
Crystal Palace F.C. - Chelsea F.C.	0.1884	**0.1680**	**0.2243**	0.1373
Macro-Average	0.1134	0.1074	**0.1228**	0.1033

Table D.7: ROUGE-2 recall results based on authoritative Twitter accounts.

Match	$\text{MEAD}_{\text{FMMR}}$	FMMR	MEAD_{DGS}	DGS
Manchester United F.C. - Arsenal F.C.	0.0395	0.0348	0.0403	0.0559
Liverpool F.C. - S.S.C. Napoli	0.0349	0.0312	0.0349	0.0368
Valencia CF - Manchester United F.C.	0.0548	0.0787	0.0553	0.0647
Liverpool F.C. - Manchester United F.C.	0.0441	0.0457	0.0433	0.0479
Tottenham Hotspur F.C. - Wolverhampton Wanderers F.C.	0.1077	0.0798	0.1297	0.0808
Crystal Palace F.C. - Chelsea F.C.	**0.1284**	**0.0914**	**0.1388**	**0.1342**
Macro-Average	0.0682	0.0603	**0.0737**	0.0701

Table D.8: ROUGE-2 recall results based on mainstream media reports.

E

Qualitative Analysis Transcript

Nicholas Mamo: First of all, thanks a lot for accepting to conduct this interview with me. I realize that there were some issues that you couldn't understand what I was saying. Just to give you a little bit of background about the system. Basically, I am a postgraduate student at the University of Malta and I am studying AI - Artificial Intelligence. In my area, I am creating a system that tracks events and creates timelines for them. So for example, we give it the keywords Crystal Palace and Chelsea and the system, the AI, automatically collects tweets that mention Crystal Palace or Chelsea, reads them on its own and creates its own timeline from the tweets.

The reason I wanted to have an interview in the dissertation was to have a sort of qualitative analysis to understand how well it does its job because as journalists, you cover matches. I wanted to get feedback from a journalist by asking a few questions. Is that clear? Do you understand?

Paul Doyle: Just a question. For example, in the second of the things that you sent me, it was a Chelsea match against Crystal Palace. Are you saying that the timeline was a compilation of tweets from all over the internet?

NM: Yes, that is why there were some inconsistencies. That is why there was profanity, for example. It was created by an AI. Just to confirm. You work at The Guardian, right? What is your job title, so to speak?

PD: Football writer.

NM: Is it okay if I mention you by name in the dissertation?

PD: If you want.

NM: Thank you. Do you want to start with the questions, or do you have any questions? Is something unclear?

PD: Yeah, the fundamental thing that was unclear to me. You want to know how useful this would be to journalists. I thought that the whole purpose of systems like this

was to abolish journalists, for machines to do the job instead. How could it possibly be useful to journalists?

NM: I think it's almost impossible to abolish journalism using systems such as this. The reason being that, for example, for the Chelsea against Crystal Palace match that you read, there were only like 60,000 tweets. It is very, very difficult to create a compilation when the numbers are even smaller. For example, I remember I fed it [the system] a game between - I think - Southampton and West Ham. The system struggled a lot because there were only like 20,000 tweets. In its current stage, and I think within the next ten years, it will not be able to come close to journalists in terms of comprehensiveness. The question that I wanted to ask - how it could help journalists - is knowing that the timeline was created based on what people found important and interesting. If you see the timeline for the Crystal Palace-Chelsea match, or the Liverpool-Napoli match, you realize that there is a lot of bias, perhaps. Does it help knowing what the topics of conversation are among the Twitterbase?

PD: In that case, it's more of an analysis of what people think on Twitter more than what is happening in the match. I guess if you're a journalist interested on what's happening on Twitter, it would be useful. We know that only a still very small proportion of people use Twitter, so you're never going to get a representative understanding. The Twitter users are a distinct group, so maybe the system will help you get an opinion of what people on Twitter are thinking of about the match or a particular player. From that point of view I suppose it could be useful. Personally, I wouldn't ever use it.

NM: Why not?

PD: A) I don't care what people think on Twitter because I think that it is not representative of what most people think. B) People are [...] more extreme on Twitter. It is a forum where people are deliberately seeking attention. If you're looking at accuracy, it's not interesting. I consider Twitter to be the "toilet walls" of the twenty-first century. People would write graffiti on toilet walls to make their opinions known. Now, they use Twitter. It's not something that's interesting or useful to journalists.

NM: Thanks for your honesty. Let's take it from a different perspective then. Imagine that the system was an intern and you were supervising. Do you remember the spreadsheet that I sent you? The first part. *PD*: The comparison of the two, yes.

NM: Could you tell which one was written by our system, and which one was written by the other blog - the other blog was ThisIsAnfield.com.

PD: Right, no, I couldn't tell. They were both as bad as each other.

NM: Alright, that's something - thanks. The second one was actually ours. There was some profanity, some [Twitter] mentions. I was surprised that This is Anfield uses emojis in their text, but that's another story. Let's move on to the actual timeline. How

well do you consider the timeline to be comprehensive, if it was an intern. Do you think that the timeline covers the most important moments? Or does it leave out certain parts?

PD: I didn't see that match so I can't say for sure. I didn't study every entry in the timeline because it was not good reading, but I did not see much tactical analysis in there. I did not see much indication of how substitutions may have changed matches. I can see a general sense of overall understanding of the subtleties of how the match changed. The main chances etcetera seemed to be covered from what I could make out, that was about it.

NM: You mentioned that it was bad reading. In what way? Was it incohesive? Is it the fact that there was a lot of emotion. What made it "bad reading"?

PD: There was a lot of emotion, which is not necessarily bad. There was slightly incoherent emotions, which is a lot of contradictory biases. You could tell that it was a mish-mash of various people's thoughts. There was congestion from that point of view.

The profanities, that's a question of style for the individual reader and individual publication. I always think that there is a time to use profanities, and it's when you are describing a particularly extreme situation because profanity is a rather extreme language. If you use it willy-nilly, you use it to describe relatively-mundane events, then how do you describe extreme situations? Honestly there was a lot of slang, which again is fine, but there was inconsistent slang. I was puzzled that there was a woman in it - Marie something [referring to Maine Mendoza] - not sure what she was talking about.

NM: That was one of the failings of the system. It was Mendoza, I think? Something like that.

PD: Yeah, I don't know what was happening there.

NM: We tracked Marcos Alonso, and actually his [full] surname is Marcos Alonso Mendoza, so I think it was Maine Mendoza or something like that. If that was it, we captured some ambiguity with someone else named Mendoza, that's what happened. That's a very good point, it's something we're actively working on [with APD]. You mentioned that it struggles with subtleties and tactics. It's a very fair point, it also struggles with substitutions sometimes.

Let's focus on what it gets rights - as you said, the big moments, such as Van Dijk's yellow card. How well do you think that the timelines describe what happened in the big moments? The big moments are naturally easier to capture for the system because they are very popular.

PD: I didn't think that it described them well, presumably for the reason that on Twitter most people are assuming that people can see the action. They don't give a detailed description of exactly what happened. So they will say, for example, "Oh Giroud

missed a [...] chance, he should have scored." They don't tell you that the ball was crossed from the right and he took it down onto his chest and he [...] a pass [...] and then curled the shot over the bar, or whatever. You don't get a detailed description of exactly what happened. You just get one person's impression or judgement.

NM: Fair enough.

PD: Similar to the yellow card. I don't remember that exact entry, but I imagine that the person might have said that "Van Dijk got a yellow card for fouling such a player. I think that was harsh" or "I think he deserved it" or "I think it should have been red," but probably don't describe exactly what happened.

NM: Yes, it struggles with emotions a lot and that is why it misses tactics and subtle points that you mentioned earlier before.

PD: Yeah, and as far as the timelines that are done by machines certainly read like they're done by machines or some kind of software, things like that you sometimes see on the BBC, which are very very bald data. It will say "corner kick to Chelsea." Or "throw-in to Liverpool." Incomplete, something like that. Because it is so blunt and boring, it probably is accurate. At least, it has that in its favour - the fact that it's accurate. When you have one [timeline] that is just judgements, but has very little objective data in it, then you have no idea whether it is accurate or not, really. That's what those people tweeting are. [...] It is even more useless than those sort of BBC things.

NM: In your opinion, objectivity is much more important than expressing the emotions associated with actions. You're just interested in what happens, not how people react.

PD: I think that the detailed description has to be accurate. Absolutely. That is important both for the people that are following in real-time, but also in the future. We very often have people who read back over the coverage, the next day or two days later, or even weeks and months later to get an accurate record of what happened. The details have to be correct.

You could also put in subject of commentary. So describe the action, you could then say "I think that should have been a red card" or "I think he should have scored that chance" or whatever. And then, it would be up to readers to decide over time whether they agree with the writer's judgement or not. However, in this system, where it's a compilation of various tweets, you don't even know who the people tweeting are so you don't know how reliable their [...] is.

NM: So on top of subjectivity, you can't even assess how reliable they are.

PD: If you know who the subject is, you can decide how much value to attach to the subjectivity. If you don't even know who the subject is, then the subjectivity has to be worthless.

NM: Okay, understood. Just one last question. There is a lot of narration and bias in the timeline. Bias, as you said, in terms of subjectivity, but also narration, such as facts. For example, Milner had assisted nine, or something like that. Do you think that bias and narration are something that should be in the timeline, or are they things that you would rather not see?

PD: First of all, in terms of the bias, I [...] to my previous point. I don't mind, I like bias, I like subjective judgements if I know who the subject is. Then I am free to decide how much value I attach to that subject, that person's judgements. If I have no idea who the person is, then it's worthless.

In terms of narration, if that helps, that can help us understand context. That's absolutely fine. This narration is just basic data, statistics, which is kind of the most basic form of narration. At least, assuming that they're giving the correct statistics, at least it has the value of being objectively-correct. By all means, including that can enrich the commentary.

NM: Thanks a lot, I think you covered all the questions that I had to ask, so thanks a lot. I think I'm done on my end, unless you have any questions or anything else.

PD: No, I don't have any more questions.

References

[1] G. Lotan, E. Graeff, M. Ananny, D. Gaffney, I. Pearce, and D. Boyd, "The arab spring| the revolutions were tweeted: Information flows during the 2011 tunisian and egyptian revolutions," *International Journal of Communication*, vol. 5, no. 0, 2011.

[2] H. H. Khondker, "Role of the new media in the arab spring," *Globalizations*, vol. 8, no. 5, pp. 675–679, 2011.

[3] E. C. Tandoc Jr. and E. Johnson, "Most students get breaking news first from twitter," *Newspaper Research Journal*, vol. 37, no. 2, pp. 153–166, 2016.

[4] A. Java, X. Song, T. Finin, and B. Tseng, "Why we twitter: Understanding microblogging usage and communities," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, WebKDD/SNA-KDD '07, (New York, NY, USA), pp. 56–65, ACM, 2007.

[5] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma, "Breaking news on twitter," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, (New York, NY, USA), pp. 2751–2754, ACM, 2012.

[6] T. Hashimoto, D. Shepard, T. Kuboyama, and K. Shin, "Event detection from millions of tweets related to the great east japan earthquake using feature selection technique," in *Proceedings of the 2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, ICDMW '15, (Washington, DC, USA), pp. 7–12, IEEE Computer Society, 2015.

[7] K. Rudra, S. Banerjee, N. Ganguly, P. Goyal, M. Imran, and P. Mitra, "Summarizing situational tweets in crisis scenario," in *Proceedings of the 27th ACM Conference on Hypertext and Social Media*, HT '16, (New York, NY, USA), pp. 137–147, ACM, 2016.

[8] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), pp. 851–860, ACM, 2010.

[9] N. Mamo and J. Azzopardi, "Fire: Finding important news reports," in *Semantic Keyword-Based Search on Structured Data Sources* (J. Szymański and Y. Velegrakis, eds.), (Cham), pp. 20–31, Springer International Publishing, 2018.

[10] M. Mathioudakis and N. Koudas, "Twittermonitor: Trend detection over the twitter stream," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, (New York, NY, USA), pp. 1155–1158, ACM, 2010.

[11] B. E. Teitler, M. D. Lieberman, D. Panozzo, J. Sankaranarayanan, H. Samet, and J. Sperling, "Newsstand: A new view on news," in *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '08, (New York, NY, USA), pp. 18:1–18:10, ACM, 2008.

[12] S. Zhao, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "Human as real-time sensors of social and physical events: A case study of twitter and sports games," *arXiv e-prints*, June 2011.

[13] C. Shen, F. Liu, F. Weng, and T. Li, "A participant-based approach for event summarization using twitter streams," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152–1162, 2013.

[14] J. Nichols, J. Mahmud, and C. Drews, "Summarizing sporting events using twitter," in *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, IUI '12, (New York, NY, USA), pp. 189–198, ACM, 2012.

[15] D. Corney, C. Martin, and A. Göker, "Spot the ball: Detecting sports events on twitter," in *Advances in Information Retrieval* (M. de Rijke, T. Kenter, A. P. de Vries, C. Zhai, F. de Jong, K. Radinsky, and K. Hofmann, eds.), (Cham), pp. 449–454, Springer International Publishing, 2014.

[16] T. Mike, B. Kevan, and P. Georgios, "Sentiment in twitter events," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 2, pp. 406–418, 2011.

[17] O. Ozdikis, P. Senkul, and H. Oguztuzun, "Semantic expansion of tweet contents for enhanced event detection in twitter," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12, (Washington, DC, USA), pp. 20–24, IEEE Computer Society, 2012.

[18] A. J. McMinn and J. M. Jose, "Real-time entity-based event detection for twitter," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. Jones, E. San Juan, L. Capellato, and N. Ferro, eds.), (Cham), pp. 65–77, Springer International Publishing, 2015.

[19] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Comput. Intell.*, vol. 31, pp. 132–164, Feb. 2015.

[20] Q. Zhao, P. Mitra, and B. Chen, "Temporal and information flow based event detection from social text streams," in *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pp. 1501–1506, AAAI Press, 2007.

[21] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 37–45, ACM, 1998.

[22] Y. Yang, T. Pierce, and J. Carbonell, "A study of retrospective and on-line event detection," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 28–36, ACM, 1998.

[23] J. Krumm and E. Horvitz, "Eyewitness: Identifying local events via space-time signals in twitter feeds," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '15, (New York, NY, USA), pp. 20:1–20:10, ACM, 2015.

[24] J. Benhardus and J. Kalita, "Streaming trend detection in twitter," *Int. J. Web Based Communities*, vol. 9, pp. 122–139, Jan. 2013.

[25] M. Kubo, R. Sasano, H. Takamura, and M. Okumura, "Generating live sports updates from twitter by finding good reporters," in *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01*, WI-IAT '13, (Washington, DC, USA), pp. 527–534, IEEE Computer Society, 2013.

[26] L. M. Aiello, G. Petkos, C. Martin, D. Corney, S. Papadopoulos, R. Skraba, A. Göker, I. Kompatsiaris, and A. Jaimes, "Sensing trending topics in twitter," *IEEE Transactions on Multimedia*, vol. 15, pp. 1268–1282, Oct 2013.

[27] Y. Han, B. Hong, and K. K. Kim, "Super bowl live tweets: The usage of social media during a sporting event," in *Proceedings of the 8th International Conference on Social Media & Society*, #SMSociety17, (New York, NY, USA), pp. 37:1–37:5, ACM, 2017.

[28] G. Van Oorschot, M. Van Erp, and C. Dijkshoorn, "Automatic extraction of soccer game events from twitter," *Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012)*, vol. 902, pp. 21–30, 2012.

[29] M. Löchtefeld, C. Jäckel, and A. Krüger, "Twitsoccer: Knowledge-based crowdsourcing of live soccer events," in *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, MUM '15, (New York, NY, USA), pp. 148–151, ACM, 2015.

[30] Z. Ren, O. Inel, L. Aroyo, and M. de Rijke, "Time-aware multi-viewpoint summarization of multilingual social text streams," in *Proceedings of the 25th ACM Inter-*

national on Conference on Information and Knowledge Management, CIKM '16, (New York, NY, USA), pp. 387–396, ACM, 2016.

[31] Y. Chang, L. Tang, Y. Inagaki, and Y. Liu, "What is tumblr: A statistical overview and comparison," *SIGKDD Explor. Newsl.*, vol. 16, pp. 21–29, Sept. 2014.

[32] M. Naaman, H. Becker, and L. Gravano, "Hip and trendy: Characterizing emerging trends on twitter," *J. Am. Soc. Inf. Sci. Technol.*, vol. 62, pp. 902–918, May 2011.

[33] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, (New York, NY, USA), pp. 591–600, ACM, 2010.

[34] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Report, Stanford*, vol. 1, no. 12, 2009.

[35] P. Ferragina, F. Piccinno, and R. Santoro, "On analyzing hashtags in twitter," in *Proceedings of the Ninth International AAAI Conference on Weblogs and Social Media*, ICWSM '15, pp. 110–119, AAAI Press, American Association for the Advancement of Science, 2015.

[36] T. Declerck and P. Lendvai, "Processing and normalizing hashtags," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 104–109, 2015.

[37] V. Nakade, A. Musaev, and T. Atkison, "Preliminary research on thesaurus-based query expansion for twitter data extraction," in *Proceedings of the ACMSE 2018 Conference*, ACMSE '18, (New York, NY, USA), pp. 14:1–14:4, ACM, 2018.

[38] K. Rudra, S. Ghosh, N. Ganguly, P. Goyal, and S. Ghosh, "Extracting situational information from microblogs during disaster events: A classification-summarization approach," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, (New York, NY, USA), pp. 583–592, ACM, 2015.

[39] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B.-S. Lee, "Twiner: Named entity recognition in targeted twitter stream," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, (New York, NY, USA), pp. 721–730, ACM, 2012.

[40] A. Olariu, "Hierarchical clustering in improving microblog stream summarization," in *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 2*, CICLing'13, (Berlin, Heidelberg), pp. 424–435, Springer-Verlag, 2013.

[41] J. Kim, B. Tabibian, A. Oh, B. Schölkopf, and M. Gomez-Rodriguez, "Leveraging the crowd to detect and reduce the spread of fake news and misinformation," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, (New York, NY, USA), pp. 324–332, ACM, 2018.

[42] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller, "Twitinfo: Aggregating and visualizing microblogs for event exploration," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, (New York, NY, USA), pp. 227–236, ACM, 2011.

[43] S. Vieweg, A. L. Hughes, K. Starbird, and L. Palen, "Microblogging during two natural hazards events: What twitter may contribute to situational awareness," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, (New York, NY, USA), pp. 1079–1088, ACM, 2010.

[44] B. Robinson, R. Power, and M. Cameron, "A sensitive twitter earthquake detector," in *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13 Companion, (New York, NY, USA), pp. 999–1002, ACM, 2013.

[45] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: A survey," *ACM Comput. Surv.*, vol. 47, pp. 67:1–67:38, June 2015.

[46] M. Bagdouri, "Journalists and twitter: A multidimensional quantitative description of usage patterns," in *Tenth International AAAI Conference on Web and Social Media*, ICWSM '16, 2016.

[47] L. Sarmento, V. Jijkuon, M. de Rijke, and E. Oliveira, “"more like these": Growing entity classes from seeds,” in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM ’07, (New York, NY, USA), pp. 959–962, ACM, 2007.

[48] E. Chisholm and T. G. Kolda, “New term weighting formulas for the vector space method in information retrieval,” *Computer Science and Mathematics Division, Oak Ridge National Laboratory*, vol. 10, p. 5698, 1999.

[49] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris, “Two-level message clustering for topic detection in twitter.,” in *Proceedings of the SNOW 2014 Data Challenge*, pp. 49–56, 2014.

[50] L. Ratinov, D. Roth, and V. Srikumar, “Conceptual search and text categorization,” tech. rep., UIUC, CS Dept, 2008.

[51] M. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[52] M. Toman, R. Tesar, and K. Jezek, “Influence of word normalization on text classification,” *Proceedings of InSciT*, vol. 4, pp. 354–358, 2006.

[53] P. Han, S. Shen, D. Wang, and Y. Liu, “The influence of word normalization in english document clustering,” in *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on*, vol. 2, pp. 116–120, IEEE, 2012.

[54] A. G. Jivani *et al.*, “A comparative study of stemming algorithms,” *Int. J. Comp. Tech. Appl*, vol. 2, no. 6, pp. 1930–1938, 2011.

[55] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[56] T. Brants, F. Chen, and A. Farahat, “A system for new event detection,” in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR ’03, (New York, NY, USA), pp. 330–337, ACM, 2003.

[57] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[58] A. Jackoway, H. Samet, and J. Sankaranarayanan, "Identification of live news events using twitter," in *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '11, (New York, NY, USA), pp. 25–32, ACM, 2011.

[59] Y. S. Yilmaz, M. F. Bulut, C. G. Akcora, M. A. Bayir, and M. Demirbas, "Trend sensing via twitter," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 14, pp. 16–26, Sept. 2013.

[60] J. W. Reed, Y. Jiao, T. E. Potok, B. A. Klump, M. T. Elmore, and A. R. Hurson, "Tf-icf: A new term weighting scheme for clustering dynamic data streams," in *Proceedings of the 5th International Conference on Machine Learning and Applications*, ICMLA '06, (Washington, DC, USA), pp. 258–263, IEEE Computer Society, 2006.

[61] H. Takamura, H. Yokono, and M. Okumura, "Summarizing a document stream," in *Advances in Information Retrieval* (P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, and V. Mudoch, eds.), (Berlin, Heidelberg), pp. 177–188, Springer Berlin Heidelberg, 2011.

[62] N. Ghelani, S. Mohammed, S. Wang, and J. Lin, "Event detection on curated tweet streams," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, (New York, NY, USA), pp. 1325–1328, ACM, 2017.

[63] L. Shou, Z. Wang, K. Chen, and G. Chen, "Sumblr: Continuous summarization of evolving tweet streams," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, (New York, NY, USA), pp. 533–542, ACM, 2013.

[64] M. Gillani, M. U. Ilyas, S. Saleh, J. S. Alowibdi, N. Aljohani, and F. S. Alotaibi, "Post summarization of microblogs of sporting events," in *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, (Re-

public and Canton of Geneva, Switzerland), pp. 59–68, International World Wide Web Conferences Steering Committee, 2017.

[65] M. A. H. Khan, D. Bollegala, G. Liu, and K. Sezaki, "Multi-tweet summarization of real-time events," in *Proceedings of the 2013 International Conference on Social Computing*, SOCIALCOM '13, (Washington, DC, USA), pp. 128–133, IEEE Computer Society, 2013.

[66] B. O'Connor, M. Krieger, and D. Ahn, "Tweetmotif: Exploratory search and topic summarization for twitter," in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, ICWSM '10, pp. 384–385, American Association for the Advancement of Science, 2010.

[67] O. Alonso, S.-E. Tremblay, and F. Diaz, "Automatic generation of event timelines from social data," in *Proceedings of the 2017 ACM on Web Science Conference*, WebSci '17, (New York, NY, USA), pp. 207–211, ACM, 2017.

[68] C. De Maio, G. Fenza, V. Loia, and M. Parente, "Time aware knowledge extraction for microblog summarization on twitter," *Inf. Fusion*, vol. 28, pp. 60–74, Mar. 2016.

[69] C. Zhang, G. Zhou, Q. Yuan, H. Zhuang, Y. Zheng, L. Kaplan, S. Wang, and J. Han, "Geoburst: Real-time local event detection in geo-tagged tweet streams," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, (New York, NY, USA), pp. 513–522, ACM, 2016.

[70] D. Gao, W. Li, X. Cai, R. Zhang, and Y. Ouyang, "Sequential summarization: A full view of twitter trending topics," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, pp. 293–302, Feb. 2014.

[71] D. Chakrabarti and K. Punera, "Event summarization using tweets," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, vol. 11 of *ICWSM '11*, pp. 66–73, American Association for the Advancement of Science, 2011.

[72] A. Zubiaga, D. Spina, E. Amigó, and J. Gonzalo, "Towards real-time summarization of scheduled events from twitter streams," in *Proceedings of the 23rd ACM*

Conference on Hypertext and Social Media, HT '12, (New York, NY, USA), pp. 319–320, ACM, 2012.

[73] M. Cataldi, L. D. Caro, and C. Schifanella, "Personalized emerging topic detection based on a term aging model," *ACM Trans. Intell. Syst. Technol.*, vol. 5, pp. 7:1–7:27, Jan. 2014.

[74] C. Lin, C. Lin, J. Li, D. Wang, Y. Chen, and T. Li, "Generating event storylines from microblogs," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, (New York, NY, USA), pp. 175–184, ACM, 2012.

[75] B. Dalvi, J. Callan, and W. Cohen, "Entity list completion using set expansion techniques," tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Language Technologies Inst, 2011.

[76] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas, "Web-scale distributional similarity and entity set expansion," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, (Stroudsburg, PA, USA), pp. 938–947, Association for Computational Linguistics, 2009.

[77] B. Letham, C. Rudin, and K. A. Heller, "Growing a list," *Data Mining and Knowledge Discovery*, vol. 27, pp. 372–395, Nov 2013.

[78] M. Efron, P. Organisciak, and K. Fenlon, "Improving retrieval of short texts through document expansion," in *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, (New York, NY, USA), pp. 911–920, ACM, 2012.

[79] E. Gabrilovich and S. Markovitch, "Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pp. 1301–1306, AAAI Press, 2006.

[80] Z. Zhang, L. Sun, and X. Han, "A joint model for entity set expansion and attribute extraction from web search queries," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 3101–3107, AAAI Press, 2016.

[81] M. A. Zingla, L. Chiraz, and Y. Slimani, "Short query expansion for microblog retrieval," *Procedia Computer Science*, vol. 96, pp. 225 – 234, 2016. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016.

[82] D. N. Milne, I. H. Witten, and D. M. Nichols, "A knowledge-based search engine powered by wikipedia," in *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, (New York, NY, USA), pp. 445–454, ACM, 2007.

[83] O. Egozi, E. Gabrilovich, and S. Markovitch, "Concept-based feature generation and selection for information retrieval," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pp. 1132–1137, AAAI Press, 2008.

[84] C. Xiong, J. Callan, and T.-Y. Liu, "Bag-of-entities representation for ranking," in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ICTIR '16, (New York, NY, USA), pp. 181–184, ACM, 2016.

[85] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *Advances in Information Retrieval* (P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, and V. Mudoch, eds.), (Berlin, Heidelberg), pp. 362–367, Springer Berlin Heidelberg, 2011.

[86] J. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, pp. 373–397, Oct 2003.

[87] R. Swan and D. Jensen, "TimeMines: Constructing Timelines with Statistical Models of Word Usage," in *ACM SIGKDD 2000 Workshop on Text Mining*, pp. 73–80, 2000.

[88] R. Swan and J. Allan, "Automatic generation of overview timelines," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, (New York, NY, USA), pp. 49–56, ACM, 2000.

[89] Y. Duan, Z. Chen, F. Wei, M. Zhou, and H.-Y. Shum, "Twitter topic summarization by ranking tweets using social influence and content quality," *Proceedings of COLING 2012*, pp. 763–780, 2012.

[90] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu, "Parameter free bursty events detection in text streams," in *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pp. 181–192, VLDB Endowment, 2005.

[91] Q. He, K. Chang, and E.-P. Lim, "Analyzing feature trajectories for event detection," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, (New York, NY, USA), pp. 207–214, ACM, 2007.

[92] J. Weng and B.-S. Lee, "Event detection in twitter," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, vol. 11 of *ICWSM '11*, pp. 401–408, American Association for the Advancement of Science, 2011.

[93] M. S. C. Sapul, T. H. Aung, and R. Jiamthapthaksin, "Trending topic discovery of twitter tweets using clustering and topic modeling algorithms," in *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–6, July 2017.

[94] R. Long, H. Wang, Y. Chen, O. Jin, and Y. Yu, "Towards effective event detection, tracking and summarization on microblog data," in *Proceedings of the 12th International Conference on Web-age Information Management*, WAIM'11, (Berlin, Heidelberg), pp. 652–663, Springer-Verlag, 2011.

[95] G. Ifrim, B. Shi, and I. Brigadir, "Event detection in twitter using aggressive filtering and hierarchical tweet clustering," in *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*, ACM, 2014.

[96] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," *Icwsm*, vol. 11, no. 2011, pp. 438–441, 2011.

[97] H. Abdelhaq, C. Sengstock, and M. Gertz, "Eventweet: Online localized event detection from twitter," *Proc. VLDB Endow.*, vol. 6, pp. 1326–1329, Aug. 2013.

[98] J. Azzopardi, C. Staff, and C. Layfield, "Extended no-k-means for search results clustering," in *2nd International Symposium on Web Algorithms (iSWAG), Deauville, France*, 2016.

[99] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3, pp. 120–123, IEEE, 2010.

[100] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis, "Bieber no more: First story detection using twitter and wikipedia," in *Sigir 2012 workshop on time-aware information access*, 2012.

[101] S. Ishikawa, Y. Arakawa, S. Tagashira, and A. Fukuda, "Hot topic detection in local areas using twitter and wikipedia," in *ARCS 2012*, pp. 1–5, IEEE, 2012.

[102] J.-G. Yao, X. Wan, and J. Xiao, "Recent advances in document summarization," *Knowl. Inf. Syst.*, vol. 53, pp. 297–336, Nov. 2017.

[103] J. Carbonell and J. Goldstein, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, (New York, NY, USA), pp. 335–336, ACM, 1998.

[104] L. Song, P. Zhang, Z. Bao, and T. Sellis, "Continuous summarization over microblog threads," in *Database Systems for Advanced Applications*, (Cham), pp. 511–526, Springer International Publishing, 2017.

[105] X. Wan and J. Yang, "Improved affinity graph based multi-document summarization," in *Proceedings of the Human Language Technology Conference of the NAACL,*

Companion Volume: Short Papers, NAACL-Short '06, (Stroudsburg, PA, USA), pp. 181–184, Association for Computational Linguistics, 2006.

[106] X. Wan and J. Yang, "Multi-document summarization using cluster-based link analysis," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, (New York, NY, USA), pp. 299–306, ACM, 2008.

[107] W.-t. Yih, J. Goodman, L. Vanderwende, and H. Suzuki, "Multi-document summarization by maximizing informative content-words," in *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, (San Francisco, CA, USA), pp. 1776–1782, Morgan Kaufmann Publishers Inc., 2007.

[108] I. Mani and E. Bloedorn, "Multi-document summarization by graph search and matching," in *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, pp. 622–628, AAAI Press, 1997.

[109] A. Olariu, "Efficient online summarization of microblogging streams," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pp. 236–240, 2014.

[110] G. Erkan and D. R. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Int. Res.*, vol. 22, pp. 457–479, Dec. 2004.

[111] G. Mane and A. Kulkarni, "Twitter event summarization using phrase reinforcement algorithm and nlp features," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 5, pp. 427–430, 2015.

[112] B. Sharifi, M.-A. Hutton, and J. Kalita, "Summarizing microblogs automatically," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, (Stroudsburg, PA, USA), pp. 685–688, Association for Computational Linguistics, 2010.

[113] B. Sharifi, M.-A. Hutton, and J. K. Kalita, "Experiments in microblog summarization," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pp. 49–56, IEEE, 2010.

[114] D. R. Radev, S. Blair-Goldensohn, and Z. Zhang, "Experiments in single and multi-document summarization using mead," *Ann Arbor*, vol. 1001, p. 48109, 2001.

[115] N. Alsaedi, P. Burnap, and O. Rana, "Temporal tf-idf: A high performance approach for event summarization in twitter," in *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pp. 515–521, IEEE, Oct 2016.

[116] J. Goldstein and J. Carbonell, "Summarization: (1) using mmr for diversity - based reranking and (2) evaluating summaries," in *Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-15, 1998*, TIPSTER '98, (Stroudsburg, PA, USA), pp. 181–195, Association for Computational Linguistics, 1998.

[117] G. Erkan and D. R. Radev, "Lexpagerank: Prestige in multi-document text summarization," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[118] Q. Wu, J. Lv, and S. Ma, "Continuous summarization for microblog streams based on clustering," in *Neural Information Processing* (S. Arik, T. Huang, W. K. Lai, and Q. Liu, eds.), (Cham), pp. 371–379, Springer International Publishing, 2015.

[119] H. Becker, M. Naaman, L. Gravano, *et al.*, "Selecting quality twitter content for events," in *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, ICWSM '11, American Association for the Advancement of Science, 2011.

[120] A. Celikyilmaz and D. Hakkani-Tur, "A hybrid hierarchical model for multi-document summarization," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, (Stroudsburg, PA, USA), pp. 815–824, Association for Computational Linguistics, 2010.

[121] J.-P. Ng, Y. Chen, M.-Y. Kan, and Z. Li, "Exploiting timelines to enhance multi-document summarization," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 923–933, 2014.

[122] R. McCreadie, C. Macdonald, and I. Ounis, "Incremental update summarization: Adaptive sentence selection based on prevalence and novelty," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, (New York, NY, USA), pp. 301–310, ACM, 2014.

[123] F. C. T. Chua and S. Asur, "Automatic summarization of events from social media," in *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, ICWSM '13, pp. 81–90, 2013.

[124] D. Wang, L. Zheng, T. Li, and Y. Deng, "Evolutionary document summarization for disaster management," in *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, (New York, NY, USA), pp. 680–681, ACM, 2009.

[125] L. Marujo, R. Ribeiro, A. Gershman, D. M. de Matos, J. P. Neto, and J. Carbonell, "Event-based summarization using a centrality-as-relevance model," *Knowledge and Information Systems*, vol. 50, no. 3, pp. 945–968, 2017.

[126] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[127] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, (Stroudsburg, PA, USA), pp. 311–318, Association for Computational Linguistics, 2002.

[128] A. Turpin and F. Scholer, "User performance versus precision measures for simple search tasks," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, (New York, NY, USA), pp. 11–18, ACM, 2006.

[129] J. Lanagan and A. F. Smeaton, "Using twitter to detect and tag important events in live sports," *Artificial Intelligence*, vol. 29, no. 2, pp. 542–545, 2011.

[130] F. Alvanaki, S. Michel, K. Ramamritham, and G. Weikum, "See what's enblogue: Real-time emergent topic identification in social media," in *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, (New York, NY, USA), pp. 336–347, ACM, 2012.

[131] Q. He, K. Chang, and E. P. Lim, "Using burstiness to improve clustering of topics in news streams," in *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pp. 493–498, Oct 2007.

[132] Q. He, K. Chang, E.-P. Lim, and J. Zhang, "Bursty feature representation for clustering text streams," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 491–496, SIAM, 2007.

[133] A.-M. Popescu, M. Pennacchiotti, and D. Paranjpe, "Extracting events and event descriptions from twitter," in *Proceedings of the 20th International Conference Companion on World Wide Web*, WWW '11, (New York, NY, USA), pp. 105–106, ACM, 2011.

[134] T. Lappas, B. Arai, M. Platakis, D. Kotsakos, and D. Gunopulos, "On burstiness-aware search for document sequences," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, (New York, NY, USA), pp. 477–486, ACM, 2009.

[135] C.-Y. Lin, "Rouge: a package for automatic evaluation of summaries," in *"Text Summarization Branches Out"*, July 2004.

[136] J.-Y. Delort and E. Alfonseca, "Dualsum: A topic-model based approach for update summarization," in *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, (Stroudsburg, PA, USA), pp. 214–223, Association for Computational Linguistics, 2012.

[137] X. Wan, "Update summarization based on co-ranking with constraints," *Proceedings of COLING 2012: Posters*, pp. 1291–1300, 2012.

[138] A. Haghighi and L. Vanderwende, "Exploring content models for multi-document summarization," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, (Stroudsburg, PA, USA), pp. 362–370, Association for Computational Linguistics, 2009.

[139] A. Nenkova and L. Vanderwende, "The impact of frequency on summarization," *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005*, vol. 101, 2005.

[140] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion," *Information Processing & Management*, vol. 43, no. 6, pp. 1606 – 1618, 2007. Text Summarization.

[141] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang, "Evolutionary timeline summarization: A balanced optimization framework via iterative substitution," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, (New York, NY, USA), pp. 745–754, ACM, 2011.

[142] F. Liu, Y. Liu, and F. Weng, "Why is "sxsw" trending?: Exploring multiple text sources for twitter topic summarization," in *Proceedings of the Workshop on Languages in Social Media*, LSM '11, (Stroudsburg, PA, USA), pp. 66–75, Association for Computational Linguistics, 2011.

[143] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using n-gram co-occurrence statistics," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, (Stroudsburg, PA, USA), pp. 71–78, Association for Computational Linguistics, 2003.