algorithms

MDPI

Article

An Automatic Participant Detection Framework for Event Tracking on Twitter

Nicholas Mamo *, **Joel Azzopardi** and **Colin Layfield**

Faculty of ICT, University of Malta, MSD 2080 Msida, Malta; joel.azzopardi@um.edu.mt (J.A.); colin.layfield@um.edu.mt (C.L.)
* Correspondence: nicholas.mamo.14@um.edu.mt

Abstract: Topic Detection and Tracking (TDT) on Twitter emulates human identifying developments in events from a stream of tweets, but while event participants are important for humans to understand what happens during events, machines have no knowledge of them. Our evaluation on football matches and basketball games shows that identifying event participants from tweets is a difficult problem exacerbated by Twitter's noise and bias. As a result, traditional Named Entity Recognition (NER) approaches struggle to identify participants from the pre-event Twitter stream. To overcome these challenges, we describe Automatic Participant Detection (APD) to detect an event's participants before the event starts and improve the machine understanding of events. We propose a six-step framework to identify participants and present our implementation, which combines information from Twitter's pre-event stream and Wikipedia. In spite of the difficulties associated with Twitter and NER in the challenging context of events, our approach manages to restrict noise and consistently detects the majority of the participants. By empowering machines with some of the knowledge that humans have about events, APD lays the foundation not just for improved TDT systems, but also for a future where machines can model and mine events for themselves.

Keywords: information retrieval; automatic participant detection; twitter; event understanding; topic detection and tracking; event modelling

check for updates

Citation: Mamo, N.; Azzopardi, J.; Layfield, C. An Automatic Participant Detection Framework for Event Tracking on Twitter. *Algorithms* **2021**, *14*, 92. https://doi.org/10.3390/a14030092

Academic Editor: Frank Werner

Received: 26 February 2021
Accepted: 16 March 2021
Published: 18 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

For many people, the idea of a football match is a well-defined concept: Two teams of 11 players, playing each other for 90 min of football. Thousands of spectators use social networks to transform what was a traditionally social experience into a lively discussion, and it is not just a matter of football or sports events. Politics, natural disasters and tragedies all attract a lot of attention, and Topic Detection and Tracking (TDT) research pounced on the opportunity, using tweets to build timelines of events in near real-time. However, these timelines remain far below the standards of the news media, partially because machines do not understand events like humans.

One of the most basic semantic components of events is their participants: Events either affect participants, or participants affect them. For example, the players directly influence the outcome of a football match, and candidates shape elections. While it is essential for humans to recognize the participants to thoroughly understand the event, few TDT systems give them their due consideration. The traditional definition of an event excludes participants, and even approaches that focus on them [1–3] never formalize what qualifies an entity to be a participant.

Understanding the role of participants in events, and identifying who or what is participating in an event, can be instrumental for TDT and event modelling systems to produce more useful and accurate information about events. For example, Shen et al. [1], McMinn and Jose [2], and Huang et al. [3], construct separate timelines for named entities captured by off-the-shelf Named Entity Recognition (NER) tools. The three systems report a positive impact of this participant-centric view of events on TDT's results, highlighting the value of participants. However neither approach distinguishes between incidental mentions of named entities and actual participants with closer ties to the event. Here, we show that this distinction is not just important, but also necessary when tracking events on Twitter.

In this article we extend our previous work [4], where we argued that the definition of events should include participants, and describe Automatic Participant Detection (APD) as a problem to identify the event's participants before the event starts. With this knowledge before the event has even started, our system could collect tweets mentioning the event's participants, leading to a visibly broader coverage, as shown in Figure 1. However, we emphasise that event knowledge is a tool with wide applicability not just in TDT, but also related tasks, such as to summarise events [5], to create explainable events [6], and to model and mine events [7]. In this article we elaborate on the initial idea of APD with the following contributions:

- We define event participants and propose a six-step framework for APD. Differently from existing approaches that consider participants [1–3], APD extracts participants before the event starts, not while it is ongoing. This framework first confirms which named entities are relevant to the event. Then it looks for other participants that Twitter users are not discussing, and hence which NER cannot identify;
- We explore what named entities Twitter users talk about before sports events, with a focus on football matches and basketball games. We show that although NER toolkits are relatively capable of navigating Twitter's informal tweeting habits, many of the named entities that users mention before events are spurious and not directly-related to the event. As a result, NER cannot be a stand-alone replacement for APD;
- We implement the APD framework based on the hypothesis that event participants are tightly-interconnected on Wikipedia. Our evaluation on football matches and basketball games shows that our approach is capable of identifying almost twice as many participants as traditional NER approaches. At the same time, our approach almost halves NER noise, which usually consists of irrelevant named entities that are only tangentially-related to events, and which therefore cannot be qualified as event participants.

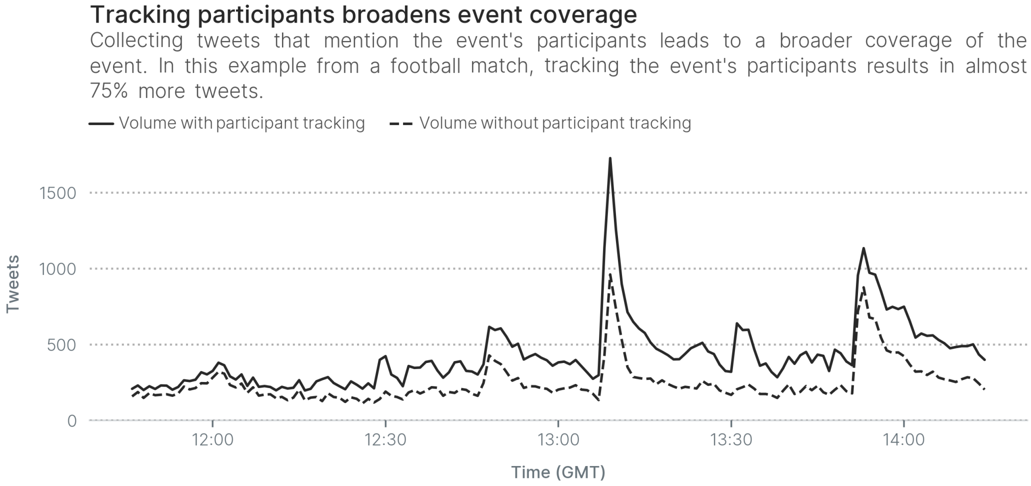


Figure 1. Collecting tweets that mention the event's participants broadens coverage.

The rest of this article is organised as follows. We outline related research in Section 2. Following our previous work [4], we describe APD as an approach that borrows from query expansion and entity set expansion. Based on this research, we propose a framework for APD and present our own implementation in Section 3. In Section 4 we explore what named entities Twitter users discuss before football matches and basketball games, and evaluate our approach on several sports events. We conclude the article with plans for future work and ideas for applications of APD in Section 5.

2. Related Work

TDT was proposed late in the 20th century as an initiative to discover stories from news corpora and link them together. Early efforts focused on identifying multiple simultaneous events from a document stream, which is today known as unspecified event detection [8]. In this context, the key to TDT's accuracy was the ability for algorithms to distinguish between events. One prevailing idea was that an event could be understood in terms of what, where, and when it happens, and, crucially, who is involved [9,10]. Since a person or a place is normally only active in one event at a time, unspecified event detection used such named entities to distinguish between events [9–11].

When Twitter launched, it contributed several benefits to TDT. In spite of its characteristically-short and informal tweets, the popular social network generates a lot of data, and most of it is publicly-available through Twitter's API [12]. As a result, Twitter made it possible to follow a particular topic or specific events, allowing what is known as specified event detection. Since there is no need to contrast events with each other in specified event detection, the commonly-accepted definition of an event is very simple [8]:

Definition 1. *An event is a real-word occurrence that happens at a particular place and at a particular time.*

However, neither the long years nor the launch of Twitter changed what was important in events. From some perspectives, it is not just important, but also necessary that a machine is aware of who is involved in events to be able to clearly describe events to humans [6]. Moreover, understanding is the required preamble to more advanced applications that can model and reason about events [7]. How could a machine do that without truly understanding who is participating in an event?

In this article, we focus on how we can identify a specified event's participants before it starts. We note that modern TDT works only consider participants when the approaches revolve around them [1–4]. The simplest uses for participants are approaches that, like Event TimeLine Detection (ELD) [4], broaden coverage by collecting tweets that mention them, but these applications usually require the participant names as manual input [13]. Another simple use case improved summarisation by prioritising tweets that mentioned participants [5].

Other approaches present more complex applications for participants. For example, instead of building one event timeline, Shen et al. [1], McMinn and Jose [2], and Huang et al. [3] build one timeline for each participant. These systems use off-the-shelf NER techniques to identify participants, and disregard the challenges of recognising named entities from tweets [14]. In addition, they never discern between trivial named entities and actual participants that affect or are affected by the event.

Similarly to our previous work [4], we liken APD to query expansion because it receives a short representation of the event—the query—and expands it with the participants to generate a more expressive description. However, looking for participants in the pre-event stream is a difficult task, as we show in Section 4. Therefore in this article we complement query expansion with entity set expansion. To the best of our knowledge, the idea of APD has not been developed in any previous research, so in the rest of this section we look at existing literature in query expansion and entity set expansion.

2.1. Query Expansion

Query expansion on Twitter tackles two issues. The first problem is vocabulary mismatch because tweets are so short, it is very easy to miss relevant tweets that do not mention the query keywords [12,15]. The second issue is that most queries are brief since it is an abstract representation of the user's needs [15]. Without understanding the query, a search engine interprets it literally.

Query expansion compensates for the machine's lack of understanding of the user's needs by expanding the query with additional terms that capture the user's intentions [16]. The expansion process usually starts by extracting candidate terms, and then scores and ranks them to retain only those keywords that are semantically-related to the query [16–18].

Query expansion can be as simple as expanding the query terms with synonyms from thesauri [12] or structured data, such as DBpedia [16]. Synonyms minimise the risk of changing the query's meaning, but this method fails to explore other closely-related keywords. Instead, many extraction approaches use pseudo-relevance feedback to identify candidate terms. This process submits the query to a search engine and assumes that the top results are relevant, and thus contain good terms for query expansion. For example, Albishre et al. [17] use pseudo-relevance feedback to extract the latent topics in the retrieved documents and build a new language model for the query.

In the context of Twitter, methods based on tweets retain the social network's language and style, but the brevity of tweets makes it difficult to diversify vocabulary. Therefore external resources are one alternative to expand and vary the query. In this way, Zingla et al. [16] look for nouns that are semantically-related to the query on Wikipedia.

2.2. Entity Set Expansion

In Section 4 we show how Twitter makes APD a challenging problem for two reasons. First, Twitter users talk about many named entities before an event starts, and not all of them are directly-relevant to the event. Second, Twitter's users are very biased, and they generate many tweets about a few popular participants. To overcome the latter problem, in our research we combine query expansion with entity set expansion to find the missed participants.

Entity set expansion receives a seed set of entities as its input. It considers these entities as examples from a broader class and looks for other instances that belong to this class [19,20]. To discover new entities, these approaches commonly exploit the web for its widely-available expertise on practically every subject [19]. Like query expansion, entity set expansion methods first extract candidate entities and then rank them according to their relevance to the seed set [19,20].

Semantics play an important role in entity set expansion to avoid deviating from the seed set of entities as the set grows—semantic drift [21]. Zhang et al.'s approach [21] considers semantics explicitly by extracting the entities' attributes from the web. They connect entities with their attributes in a bipartite graph and extract entities that have typical attributes.

A simpler and more common assumption about semantics is that entities that appear in the same context are semantically similar. For instance, Wang and Cohen's Set Expander for Any Language (SEAL) [22] and Iterative SEAL (iSEAL) [23] identify extraction patterns from pages that mention the seed set entities. Then, both approaches use these patterns to identify other entities mentioned in similar contexts. The approach by Letham et al. [19] makes a similar assumption, but instead it looks for candidates that share the same HTML tags, such as `<b>` or `<li>`, as the entities in the seed set.

3. Materials & Methods

Although APD resembles query expansion, the scope of the expansion terms differs slightly. In this section we describe the new Information Retrieval (IR) task of APD and explain how it is different from query expansion. Mindful of these differences, we propose a framework to identify participants before the event starts and present our own implementation. To conclude this section, we discuss an evaluation methodology for APD.

Before describing APD, we formally define event participants in the context of events and TDT. Participants are a part of events, so it is only intuitive that the properties of event participants are closely-related to those of events. Definition 1 of events considers only where and when the event is taking place, but we modify it to include participants:

Definition 2. *An event is a real-word occurrence that happens at a particular place and at a particular time, and that involves any number of participants.*

Participants are related to an event because they participate in it—they either affect the event or the event affects them. Therefore participants inherit the spatial and temporal properties of events:

Definition 3. *A participant is a real-world concept that affects or is affected by the event while the event is ongoing.*

This definition describes two characteristics: the discriminative and temporal properties of participants. The discriminative property binds participants to be active in the event since they either affect or are affected by it. For a participant to be truly discriminative, normally it is only active in one event at a time. For example, while a debate involving American presidential candidates affects the USA, the presidential debate is not the only event in the USA. Therefore the country itself is not discriminative with regards to the debate, unlike the presidential candidates who are involved in it.

The temporal property of participation requires that participants are active at the same time as the event. This excludes entities that are not relevant while the event is taking place. In the presidential debate example, election candidates that dropped out and who are not part of the debate are not temporally relevant. This definition considers the presidential candidates and the debate's host to be participants because they are directly affecting the debate while it is ongoing.

Moreover, the definition is flexible enough such that additional assumptions can restrict or relax these properties. For example, in the analysis of Section 4 we do not consider injured and suspended football players to be participants as they are not actively participating in the football match. However, unavailable players could be perceived as participants because their absence still has consequences on the event, such as by influencing the coach's decisions or the match's outcome.

Furthermore, we acknowledge that normally, participants describe named entities: Persons, organisations, or locations. We follow this assumption in the rest of this article. However, while we assume that all participants are named entities, not all named entities are participants; according to Definition 3, only named entities that are discriminative and temporally-relevant are participants. This is what makes APD different from other participant-centric TDT approaches [1–3]. Based on these definitions, we present the APD framework and our implementation of it next.

3.1. Framework

APD shares query expansion's objective to describe the input query, or the event description, in more detail. However, query expansion is insufficient in isolation because Twitter users do not only talk about relevant participants before the event starts. Therefore in this section, we present a framework that combines query expansion and entity set expansion.

The APD framework receives as input a corpus of tweets collected shortly before the event. This period covers the time when users are likely to start discussing the event more intensely, and can be used to generate machine-readable information about the event in advance. The details about the datasets used in this research are shown in Table 1. APD processes these datasets in six steps to finally output a list of participants that could be involved in the event once it starts:

1. Extract named entities, or candidate participants, from the corpus;
2. Score and rank the candidates;
3. Filter out the low-scoring candidates;
4. Resolve valid candidates to a semantic form so they become participants;
5. Extrapolate the seed set of participants; and
6. Post-process the participants.

The first three steps correspond roughly to pseudo-relevance feedback in query expansion. The next steps overcome the challenges of APD that we observe in Section 4: Twitter's discussions mention many named entities that are not relevant to the event while missing many valid participants. Step 4 confirms which named entities are likely to be valid participants, and Step 5 is an entity set expansion technique that looks for the missed participants. Figure 2 shows the flow of this process from the collection of tweets, which are the algorithm's input, to the list of participants, or the output.

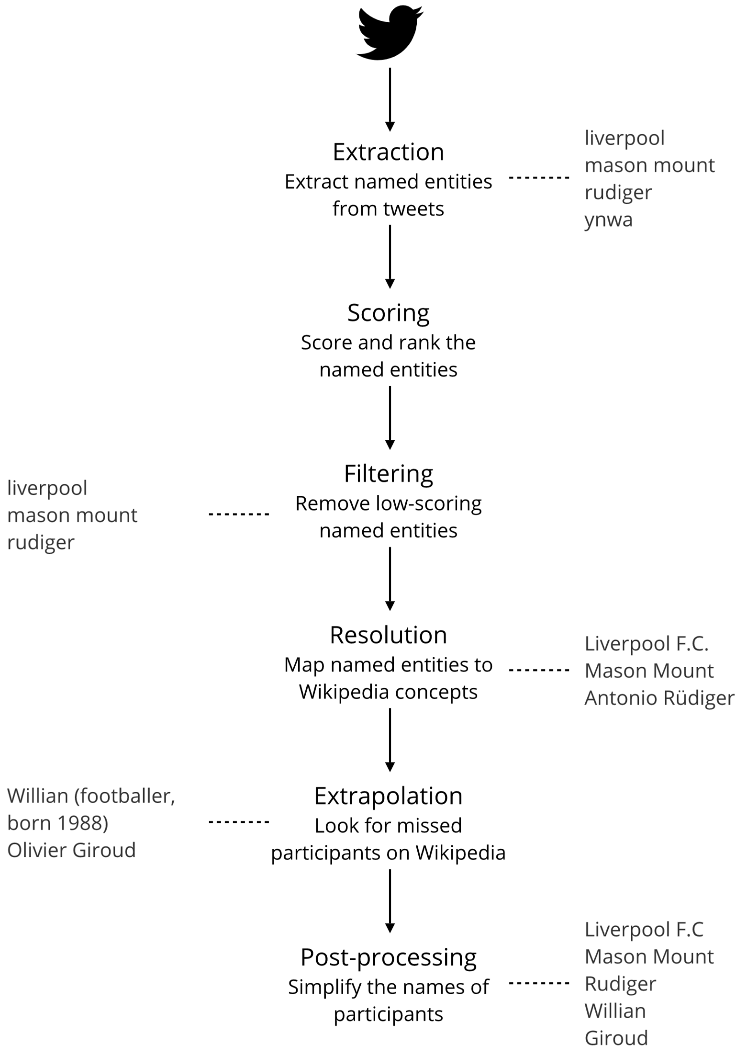

Extraction
Extract named entities from tweets
liverpool
mason mount
rudiger
ynwa
Scoring
Score and rank the named entities
Filtering
Remove low-scoring named entities
liverpool
mason mount
rudiger
Resolution
Map named entities to Wikipedia concepts
Liverpool F.C.
Mason Mount
Antonio Rüdiger
Extrapolation
Look for missed participants on Wikipedia
Willian (footballer, born 1988)
Olivier Giroud
Post-processing
Simplify the names of participants
Liverpool F.C
Mason Mount
Rudiger
Willian
Giroud

Figure 2. The process of the APD (Automatic Participant Detection) framework takes as input a corpus of tweets collected before an event and outputs a list of event participants.

Table 1. The datasets used in the evaluation.

Event Query	Collected (UTC)	Tweets
#BarcaAtleti, Barca, Barcelona, Atleti, Atletico	30 June 2020 18:45–19:45	6889
#WOLARS, Wolves, Arsenal	4 July 2020 15:15–16:15	26,467
#AVLMUN, Villa, Manchester United	9 July 2020 18:00–19:00	16,044
#LIVBUR, Liverpool, Burnley	11 July 2020 12:45–13:45	9491
#ARSLIV, Arsenal, Liverpool	15 July 2020 18:00–19:00	26,498
#TOTLEI, Tottenham, Leicester	19 July 2020, 13:45–14:45	4006
#LIVCHE, Liverpool, Chelsea	22 July 2020 18:00–19:00	32,360
Nets, Warriors	22 December 2020 23:00–00:00	14,333
Lakers, Clippers	23 December 2020 02:00–03:00	48,141
Celtics, Nets	25 December 2020 21:00–22:00	7030
Lakers, Mavericks	26 December 2020 00:00–01:00	9508

We describe our implementation of this framework next. Due to the time-consuming nature of our evaluation, described in Section 4, we set the values of our implementation's parameters empirically. All tools and algorithms implemented for this article are available on GitHub [24].

3.1.1. Extraction

Since we assume that all participants are named entities, extraction is analogous to identifying named entities from the collected tweets. In this article, we use the Natural Language Toolkit (NLTK) (http://nltk.org, last accessed on 17 February 2021) and TwitterNER, and the NER approach trained on tweets [14]. Although we use off-the-shelf libraries, unlike other participant-centric approaches [1–3] we do not rely entirely on them, with the next steps overcoming the NER techniques' limitations.

To extract more information from tweets, we pre-process them by replacing Twitter mentions with the users' display names. For example, *@ManUtd* becomes *Manchester United*. This step makes tweets resemble natural language more closely, and thus makes it easier to extract named entities. Although retweets may introduce bias, we retain them in our approach since we assume that they reflect authoritative content about the event.

3.1.2. Scoring

Our scoring strategy assumes that the most frequent named entities are more likely to be participants. The scoring step creates a frequency-based ranking of the named entities extracted in the previous step. This ranking is used in the next step to retain the most credible named entities.

3.1.3. Filtering

As we show in Section 4, the pre-event discussions on Twitter are very noisy. Generally, the tweets mention many named entities, and not all of them are participants. The top named entities are likely to be discriminative and temporally-relevant since they are mentioned frequently before the event. Therefore the filtering step retains the k most frequent named entities from the ranking produced in the previous step.

We found 50 to be a good value for k in football matches and basketball games because it is close to the number of participants in both types of events. Furthermore, we note that the rankings become far noisier after the first 50 items, with many spurious named entities that could mislead the rest of the APD process. At this stage, the retained named entities are not participants, but candidates that are accepted or rejected in the next step: Resolution.

3.1.4. Resolution

As the results in Section 4 clearly show, the tweeting habits before events make it infeasible for NER to detect all of an event's participants while simultaneously discarding non-participants. The resolution step allows APD to rely less on NER, and instead confirm which candidates are discriminative and temporally-relevant, thereby making them participants.

Our approach exploits Wikipedia to this end. Apart from its extensive coverage, Wikipedia represents each concept as a single article, and we use this aspect to assess the suitability of named entities to be participants. The algorithm fetches Wikipedia articles that could represent named entities by submitting each named entity separately as a query to the MediaWiki API (https://mediawiki.org/wiki/API:Main_page, last accessed on 17 February 2021) and retrieving the top five related articles. We exclude articles with a year in the title, such as "2019–20 Premier League", unless the year is used to disambiguate the article.

To assess whether a Wikipedia article is fit to be a participant, we compare the corpus of tweets with the first sentence in the article. We focus only on its first sentence because it succinctly describes the Wikipedia concept in its present state, and therefore this sentence can be an approximation for the candidate's temporal relevance to the event. For example, the first sentence in Alexandre Lacazette's Wikipedia article describes what he does at present:

> Alexandre Lacazette [...] is a French professional footballer who plays as a forward for Premier League club Arsenal and the France national team. He is known for his pace, hold-up play, and work-rate. [...]

Before computing a score for each article, we vectorise all tweets and the Wikipedia sentences by applying stemming and stopword removal. We also normalise repeating letters, which are more common on Twitter than on Wikipedia. For example, we replace the word *gooo* with the shorter *go*.

Our approach weights vectors using Term Frequency-Inverse Corpus Frequency (TF-ICF). This weighting scheme is similar to Term Frequency-Inverse Document Frequency (TF-IDF), but it approximates IDF using a different, general corpus [25]. As a result, this term-weighting scheme boosts terms that appear more often in the event domain than in general. As a general corpus for the ICF component, we used Twitter's Sample API to collect a large corpus of English tweets over 12 hours. Then, our approach scores the five articles collected earlier for each candidate as follows:

$$score_{c,a} = cos(c, a_{title}) \cdot cos(C, a_{first}). \tag{1}$$

These two components measure how relevant the Wikipedia article is to the named entity and to the event's domain respectively. We measure relevance using cosine similarity due to its common application to assess document similarity. $cos(c, a_{title})$ is the cosine similarity between the candidate's name c and the article's title, a_{title}; intuitively, an article about the candidate mentions them in the title. The second component, $cos(C, a_{first})$, measures the similarity between the article's first sentence, a_{first}, and the event's domain, which is the centroid of the collected corpus C.

Our approach accepts a candidate as a participant if at least one Wikipedia article exceeds a low threshold, empirically set to 0.05. In practice, this threshold filters out participants whose Wikipedia concepts are hardly related to the event. The algorithm assumes that the highest-scoring Wikipedia article represents the participant. The accepted participants become an automatically-generated seed set for the extrapolation step, analogous to entity set expansion.

3.1.5. Extrapolation

Extrapolation is analogous to entity set expansion. This step is necessary in APD because, as we show in Section 4, the resolution step inherits the bias of Twitter: Users discuss a few, popular participants and barely mention the rest. Extrapolation minimises the effects of bias by looking for the missing participants, identifying other concepts that are similar to the resolved participants.

Our approach is based on the hypothesis that any missed participants are tightly-connected with the resolved participants. In reality, this assumption holds true in many domains. For example, the Wikipedia pages of Germany and France have a section with links to the articles of other European Union member states. Therefore we build a graph where the nodes represent Wikipedia articles, and undirected edges between them indicate that one article has an outgoing link to the other.

The algorithm fetches linked articles twice. First, the algorithm fetches outgoing links from the seed set of participants that were resolved in the previous step, and retains the 100 most linked articles. Our approach creates weighted edges by comparing the first sentence of pairs of articles, similarly to before. Since articles retrieved in the first iteration are semantically closer to the seed set, the graph retains edges between them if their similarity is non-zero.

Second, the algorithm fetches articles linked from these new pages and retains the 500 most linked articles. This time, the graph retains edges between articles if their cosine similarity is at least 0.5. We note that the more links the algorithm collects, the more candidate participants it considers. Accumulating more links, however, also means more API calls, which brings down the performance of APD to the detriment of real-time applications.

While constructing the graph, our APD approach again excludes concepts with a year in their title, unless the year is a disambiguation aid. We also exclude "List of" articles, such as List of Premier League seasons, on the basis that they rarely represent participants, if at all. By being so selective with the Wikipedia articles it adds to the graph, the algorithm restricts the semantic drift.

Eventually, the links between articles form clusters of Wikipedia concepts in the graph. We make the structure of the graph clearer using the Girvan–Newman algorithm [26]. We repeatedly remove the most central edge until the number of communities is fewer than the square root of the order of the graph. This step isolates noisy articles, but retains semantically-related concepts as part of the same cluster. Since extrapolation has a similar role to entity set expansion, we assume that graph components with fewer than four nodes are unlikely to represent a class of entities, and we discard them.

To rank the new candidate participants, we score articles in the remaining graph components using Equation (1) by comparing the first sentence of each Wikipedia article with the corpus of tweets collected before the event. The algorithm again retains articles whose similarity scores exceed 0.05 as participants, thereby excluding concepts that barely overlap with the event domain. Our implementation ranks the remaining participants in descending order of similarity. We note that this step relies on the assumption of interconnectivity for semantic similarity and it only approximates the discriminative and temporal properties from the articles' first sentences.

3.1.6. Post-Processing

The post-processing step concludes the framework by cleaning the participant names. This step depends on the designated application for participants, but since we map participants to Wikipedia, it would generally be desirable to remove text in parentheses because it serves as a disambiguation detail. For example, the article title of Barcelona's goalkeeper, Neto, is Neto (footballer, born 1989) due to his relatively common surname. Post-processing reduces this title to just Neto.

We also note that sports domains, which we focus on in this article, are filled with informal orthography. In particular, it is common practice to refer to participants by their surnames, and on Twitter they are seldom written with accents. Therefore we remove accents and reduce the names of persons to their surnames, unless the surname is an English lexeme. To decide whether a participant is a person, we look for a date of birth in their Wikipedia page.

The final ranking concatenates the lists of resolved and extrapolated participants. We place the resolved participants first since the algorithm captured them directly from the pre-event stream, and thus we can be more certain that they are discriminative and temporally-relevant. Before analyzing this final ranking, we introduce our evaluation methodology for APD.

3.2. Evaluation Metrics

APD's output, a list of participants, can be analysed in two ways: The number of participants in it, and the quality of the ranking's order. Both aspects are important; the number of participants describes the algorithm's ability to identify as many entities as possible, while the order describes how well the technique distinguishes between real participants and named entities that are unrelated to the event. We suggest standard IR metrics to evaluate these two qualities.

In many events, like the football matches and basketball games that we follow in the next section, the participants are easily-enumerable. Precision and recall, defined as follows, can be used to evaluate APD's retrieval performance:

$$precision = \frac{tp}{tp + fp} \tag{2}$$

$$recall = \frac{tp}{tp + fn}. \tag{3}$$

The true positives and the false positives, tp and fp, refer to the number of captured participants and the number of incorrect named entities in the ranking respectively. The false negatives, fn, are the missed participants. Together, precision and recall assess the APD algorithm's ability to capture as many participants as possible without introducing noise. The two can be combined into the harmonic mean, or the F-measure [27].

To evaluate the order of the ranking, we propose Mean Average Precision (MAP), or the mean of the Average Precision (AP) over all datasets. AP is the mean precision score of a list of items, considering only precision values at ranks where the item is correct [28]:

$$AP = \frac{1}{r} \sum_{k=1}^{n} P@k \cdot rel_k. \tag{4}$$

In this equation, r is the number of correct participants in the ranking, n is the number of elements in the ranking, and $P@k$ is the precision at rank k. rel_k's role is to consider only those elements that are relevant; it is set to 1 if item k in the ranking is relevant and 0 otherwise. The R-precision, equivalent to $P@r$, is another suitable metric [28]. In the next section, we evaluate our approach using precision, recall and MAP—all bound between 0 and 1.

4. Results

In this section, we evaluate our APD algorithms on football matches and basketball games, and contrast our implementations with standard NER techniques. Our evaluation shows how NER techniques do not suffice to detect event participants, but that APD can significantly improve performance. Before we discuss these results, we describe our evaluation set-up, including the datasets, ground truth, and baselines.

4.1. Evaluation Set-up

4.1.1. Datasets

Like many specified event detection approaches, we base our evaluation on sports events [1,3,13,29]. We iterate that neither APD nor its framework are limited to sports events, and both could be applied in other domains. However, just like sports events are popular in TDT evaluations due to their rigid structures, they are also suited to and facilitate APD's analyses. To demonstrate the portability of our APD algorithm, in this evaluation we follow football matches and basketball games. Both sports are immensely popular, with easily-enumerable participants and widely-available ground truth.

Since we were unable to find appropriate corpora for APD's evaluation and Twitter restricts data sharing, we collected our own datasets using the Tweepy (https://www.tweepy.org/, last accessed on 17 February 2021) library. The event queries, along with statistics from the datasets, are available in Table 1; the tweet IDs from these corpora are available on GitHub [24]. In all cases, we collected English tweets before the events started, at a time when we expected more discussion about the match and its participants.

The datasets cover different scenarios; some events oppose equally-popular teams, like the match between Arsenal and Liverpool, while others are mismatched, like the match between Aston Villa and Manchester United. The dataset sizes vary as well, which allows us to understand how the number of tweets before an event influences APD's ability to identify its participants. While not all tweets have the same quality, we leave the analysis of filtering strategies for future work.

4.1.2. Ground Truth

Based on Definition 3, we consider that football matches have 45 or 51 participants: 1 stadium, 2 teams, 2 coaches, 22 players, and 18 or 24 substitutes (all matches allowed 9 substitutes per team, except the match between Barcelona and Atlético Madrid, which permitted 12 substitutes). We obtained the ground truth from LiveScore.com (https://livescore.com, last accessed on 17 February 2021).

The number of participants in NBA can vary between games, depending on the number of players who are available. However, in all of our basketball datasets, the games' participants are: 1 arena, 2 teams, 2 coaches, 10 starting players, and 18 bench players. We obtained the ground truth from ESPN.com (https://espn.com/nba, last accessed on 17 February 2021) and annotated the algorithms' rankings manually. Since we are interested in capturing as many participants as possible, we only mark the first incidence of a participant as correct if it has several variations.

4.1.3. Baselines

To the best of our knowledge, the problem of APD has not been previously explored in TDT literature. Although several approaches [1–3] use participants, they are only a means to an end: Improving the performance of the over-arching TDT algorithm. In addition, these approaches address a different problem than APD: They identify named entities during—not before—the event using off-the-shelf NER tools. For these reasons, we cannot use any of these systems as baselines.

Instead, we compare our APD algorithm with two NER libraries: NLTK's NER implementation and TwitterNER [14]. NLTK is a general-purpose Natural Language Processing (NLP) library, whereas TwitterNER is a NER method trained on tweets. In both cases, we follow the first three steps of the framework described in Section 3: We extract named entities using the NER library, score them based on frequency, and filter the ranking to retain only the top 50 most common entities.

We also evaluate our APD algorithm twice: Once by extracting named entities using NLTK and again using TwitterNER. We refer to the two approaches as APD_{NLTK} and $\text{APD}_{\text{TwitterNER}}$ respectively. Both approaches resolve as many participants as possible from the top 50 extracted named entities and use extrapolation to find missing participants. Like the baselines, at the end we retain the top 50 participants from the final ranking. The results of the baselines and of our APD algorithms after resolution are presented in Table 2. Table 3 shows the results of our APD algorithms after extrapolation. We discuss these results next.

Table 2. The precision and recall of the rankings produced by the baselines and by our APD approaches after resolution.

	Baseline				Resolution			
	NLTK		TwitterNER		APD$_{\text{NLTK}}$		APD$_{\text{TwitterNER}}$	
Match	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
Barcelona-Atlético Madrid	0.16	0.16	0.14	0.14	0.73	0.16	0.70	0.14
Wolves-Arsenal	0.20	0.22	0.14	0.16	0.56	0.11	0.67	0.13
Aston Villa-Manchester United	0.32	0.36	**0.42**	**0.47**	**0.93**	**0.31**	**0.95**	0.40
Liverpool-Burnley	**0.34**	**0.38**	0.38	0.42	0.82	**0.31**	0.48	0.24
Arsenal-Liverpool	**0.34**	**0.38**	0.28	0.31	0.76	0.29	0.80	0.27
Tottenham-Leicester	0.30	0.33	0.36	0.40	0.71	0.22	0.56	0.20
Liverpool-Chelsea	0.20	0.22	0.26	0.29	0.50	0.20	0.53	0.22
Nets-Warriors	0.18	0.27	0.28	0.42	0.69	0.27	0.78	**0.42**
Lakers-Clippers	0.16	0.24	0.20	0.30	0.62	0.24	0.67	0.30
Celtics-Nets	0.10	0.15	0.20	0.30	0.63	0.15	0.67	0.24
Lakers-Mavericks	0.16	0.24	0.18	0.27	0.71	0.15	0.78	0.21
Average	0.22	0.27	0.26	0.32	0.70	0.22	0.69	0.25

Table 3. The precision and recall of our APD approaches' final rankings after extrapolation.

	APD$_{\text{NLTK}}$		APD$_{\text{TwitterNER}}$	
Match	**Precision**	**Recall**	**Precision**	**Recall**
Barcelona-Atlético Madrid	0.46	0.45	0.43	0.35
Wolves-Arsenal	**0.58**	0.47	0.65	0.49
Aston Villa-Manchester United	0.54	0.60	**0.70**	0.78
Liverpool-Burnley	0.48	0.53	0.48	0.53
Arsenal-Liverpool	**0.58**	0.64	0.58	0.64
Tottenham-Leicester	0.36	0.40	0.46	0.51
Liverpool-Chelsea	**0.58**	0.64	0.66	0.73
Nets-Warriors	0.46	**0.70**	0.54	**0.82**
Lakers-Clippers	0.47	0.61	0.46	0.64
Celtics-Nets	0.17	0.24	0.40	0.61
Lakers-Mavericks	0.36	0.55	0.45	0.52
Average	0.46	0.53	0.53	0.60

4.2. Discussion

In this discussion, we analyse how people talk about events on Twitter before events start, and how our APD algorithms and off-the-shelf NER libraries perform in this environment. A glance at Figure 3 makes it immediately clear that the discussion starts well before the event itself. The figure shows a sharp spike in tweeting volume in our football datasets around the time when clubs release their line-ups for the upcoming match. This behaviour reflects the importance that users attribute to participants, as evidenced by how announcing the starting line-ups propels discussion.

However, not all participants are equally important. The best way to understand how Twitter users talk about events is by examining the rankings produced by NLTK and TwitterNER. The low precision and recall results of these baselines stand out in Table 2, only around a quarter of NLTK's and TwitterNER's top 50 participants are valid. Intuition would have it that NER tools return many false positives because of Twitter's informal nature, but that is an inaccurate assessment. The poor results of the baselines do not reflect poorly on the quality of NLTK and TwitterNER, which are relatively capable of overcoming Twitter's challenges, but on the way Twitter users talk about events.

In fact, discussions on Twitter go beyond the event and consider its broader context; the result of one football match may affect other teams, so Twitter users often talk about these effects. Other contexts are more unique. The passing of ex-Boston Celtics player K.C. Jones shortly before the NBA game opposing Boston Celtics and Brooklyn Nets generated many tributes, which led to all algorithms mistaking him as a participant.

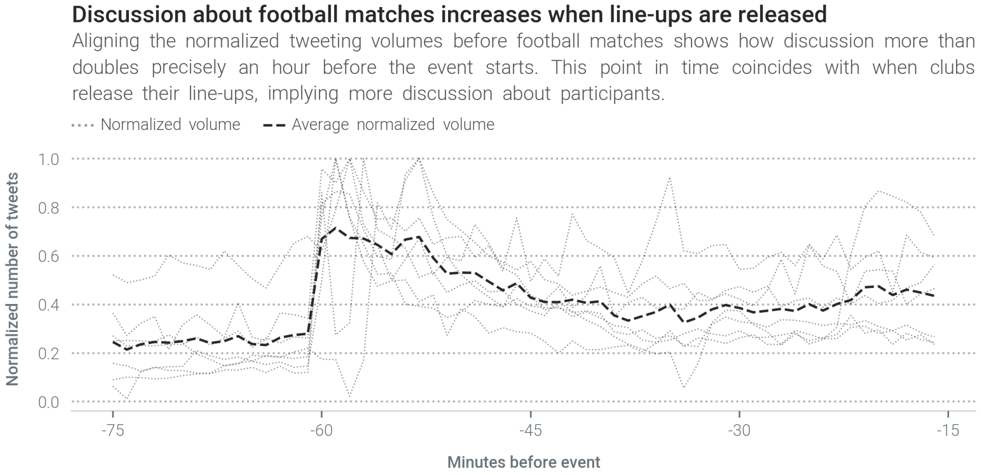


Figure 3. Discussion increases immediately after football clubs release their line-ups.

Another explanation for the poor precision is that there are several ways of referring to participants, many of them colloquial and redundant. For example, TwitterNER captured three different references to Liverpool in the top 10 ranking of the match between Liverpool and Burnley: Liverpool, the reds, and LFC.

The tweeting behaviour does not only impact precision, but recall too. Logically, the fact that football clubs release the line-ups on Twitter should make it easier to capture participants. However, it has become common practice to announce the players using images or videos, which NER techniques cannot parse. In NBA, the starting five players are published much later, so sometimes the participants remain uncertain until the very start of the game, making for an even more challenging task.

There are also inequalities among participants, with star players generally attracting far more attention than the others. For example, out of 6889 tweets in the corpus of the match between Barcelona and Atlético Madrid, only 4 mention Atlético Madrid's defender, José Giménez.

All of these factors explain why NER is inadequate to solve the APD problem on its own, no matter how accurate the library is at identifying named entities. APD's two steps that go beyond NER, resolution and extrapolation, solve these challenges by rejecting non-participant named entities and looking for missed participants respectively.

The resolution step in our APD approach is relatively successful in overcoming the low precision values, as shown in Table 2. Naturally, the recall values at the resolution stage cannot exceed the baseline's results, but precision increases drastically at a small expense to recall. In other words, our APD implementation correctly resolves most valid participants while simultaneously filtering out noisy named entities.

While the role of resolution is to automatically-generate a seed set of participants with high precision, extrapolation looks for the missing participants to improve recall. The final rankings of our APD algorithms is the list of resolved participants followed by the list of extrapolated participants. We retain the top 50 participants in this ranking, and Table 3 shows how our APD algorithms' final rankings see a sharp increase in recall with a smaller drop in precision.

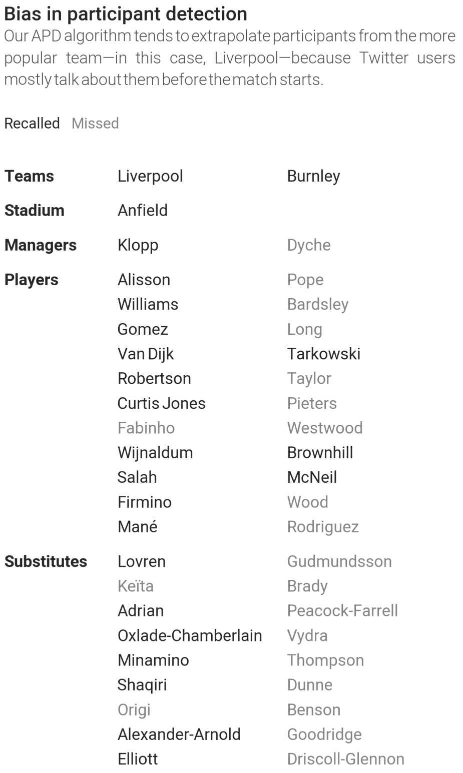
Extrapolation achieves its goal of improving recall, often needing very few examples of participants to identify others. At the same time, while precision decreases from resolution, it remains much higher than the baselines' precision values. We used the one-tailed paired samples t-test to evaluate the improvements of our APD approaches over the baselines' results of Table 2. With p-values well below 0.01, the improvements in precision and recall are all statistically-significant at the 99% confidence level.

We attribute the drops in precision to two factors. First, extrapolation captures several entities that would normally be participants. For example, Ousmane Dembélé plays regularly for Barcelona, but he was injured for the match against Atlético Madrid. Even his absence, Dembélé could be interpreted as being an indirect participant, as described in Section 3. Regardless of the interpretation, since our APD approach was unaware of his status, it identified him as a participant along the other Barcelona players.

Second, extrapolation captures several tangential concepts. For example, the teams and players in English football matches are related to the Premier League. Therefore when extrapolation could not find more relevant players, it continued listing concepts related to the Premier League and English teams in general. We also note that the three lowest precision values in the football matches came from the smallest datasets, indicating that the more comprehensive the pre-event domain is, the better the results.

A peculiar behaviour of our APD approach is that it often recalls around half of all participants, both in football and basketball. This value is not incidental. When one team is more popular than the other, NER and resolution predominantly capture players from that team. Consequently, extrapolation looks for participants that also belong to the more popular team. When this happens, our APD approach captures almost all of the participants from the popular team, but few from the other team, bringing the recall values to around 50% or 60%.

This observation exposes extrapolation's reliance on resolution, and is confirmed in other datasets. Of particular interest are the matches between Liverpool and Burnley, and between Liverpool and Chelsea. These two events had similar recall values after resolution, but the latter did not succumb to bias. This happened because most of the resolved participants in the first match were Liverpool players, as shown in Figure 4, so extrapolation sought other Liverpool players. Conversely, APD resolved a mix of Liverpool and Chelsea players in the second match, and extrapolation could detect participants from both sides.



Bias in participant detection

Our APD algorithm tends to extrapolate participants from the more popular team—in this case, Liverpool—because Twitter users mostly talk about them before the match starts.

Recalled Missed

Teams	Liverpool	Burnley
Stadium	Anfield	
Managers	Klopp	Dyche
Players	Alisson	Pope
	Williams	Bardsley
	Gomez	Long
	Van Dijk	Tarkowski
	Robertson	Taylor
	Curtis Jones	Pieters
	Fabinho	Westwood
	Wijnaldum	Brownhill
	Salah	McNeil
	Firmino	Wood
	Mané	Rodriguez
Substitutes	Lovren	Gudmundsson
	Keïta	Brady
	Adrian	Peacock-Farrell
	Oxlade-Chamberlain	Vydra
	Minamino	Thompson
	Shaqiri	Dunne
	Origi	Benson
	Alexander-Arnold	Goodridge
	Elliott	Driscoll-Glennon

Figure 4. Participant detection inherits Twitter's bias and predominantly captures participants from the more popular team.

Finally, the MAP results confirm that our APD approaches rank participants higher than unrelated concepts. Whereas NLTK and TwitterNER obtain MAP results of 0.36 and 0.46 respectively, our APD implementations based on NLTK and TwitterNER obtain MAP results of 0.68 and 0.69 respectively.

5. Conclusions

TDT systems do not have the same understanding that humans have, but in this paper we showed how APD empowers machines to generate some of that understanding automatically. Although our interpretation of participants, inspired by TDT literature and described in Definition 3, is that they are named entities, our analysis in Section 4 showed how NER was insufficient in identifying an event's participants.

In this article, we built APD on the principles outlined in our previous work [4] as a way of identifying event participants before the event starts. Our implementation, based on a mix of query expansion and entity set expansion, was largely successful in overcoming many of NER's challenges. We also showed how APD could detect several participants from the pre-event stream using relatively small datasets. Although APD cannot rely on Twitter alone to capture the majority of participants, our approach was able to overcome this problem with Wikipedia.

In spite of the improvements over traditional NER approaches, our APD technique still suffers from bias, so we suggest a closer look at how APD can overcome it. Moreover, in this article we focused on APD's applicability in sports, where participants are easily-enumerable. We suggest that future work in APD looks at its applicability in other areas where participants are neither as numerous nor easily-enumerable.

More importantly, APD's participants are machine-readable knowledge about events. Future work in this area cannot stop at improving how APD generates this knowledge. It should also explore ways how APD and participants can contribute to event-related research areas, such as TDT, and event modelling and mining.

Author Contributions: Conceptualisation, N.M., J.A. and C.L.; Methodology, N.M., J.A. and C.L.; Software, N.M.; Validation, N.M.; Formal Analysis, N.M., J.A. and C.L.; Investigation, N.M.; Resources, N.M., J.A. and C.L.; Data Curation, N.M.; Writing—Original Draft Preparation, N.M., J.A. and C.L.; Writing—Review & Editing, N.M., J.A. and C.L.; Visualisation, N.M.; Supervision, J.A. and C.L.; Project Administration, J.A. and C.L.; Funding Acquisition, N.M. All authors have read and agreed to the published version of the manuscript.

Funding: 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"*.

Data Availability Statement: Owing to Twitter's Developer Policy (https://developer.twitter.com/en/developer-terms/policy, last accessed on 17 February 2021), we are only able to share the tweet IDs from the datasets used in this research. The data and tools used in this study are openly available in a dedicated GitHub repository [24].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shen, C.; Liu, F.; Weng, F.; Li, T. 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, Atlanta, GA, USA, 9–14 June 2013; Association for Computational Linguistics: Atlanta, GA, USA, 2013; pp. 1152–1162.
2. McMinn, A.J.; Jose, J.M. Real-Time Entity-Based Event Detection for Twitter. In Proceedings of the 6th International Conference of the Cross-Language Evaluation Forum for European Languages, Toulouse, France, 8–11 September 2015; Springer: Toulouse, France, 2015; pp. 65–77.

3. Huang, Y.; Shen, C.; Li, T. Event Summarization for Sports Games using Twitter Streams. *World Wide Web* **2018**, *21*, 609–627. [CrossRef]
4. Mamo, N.; Azzopardi, J.; Layfield, C. ELD: Event TimeLine Detection—A Participant-Based Approach to Tracking Events. In Proceedings of the HT '19: 30th ACM Conference on Hypertext and Social Media, Hof, Germany, 17–20 September 2019; ACM: Hof, Germany, 2019; pp. 267–268.
5. Kubo, M.; Sasano, R.; Takamura, H.; Okumura, M. 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), Atlanta, GA, USA, 17–20 November 2013; IEEE Computer Society: Atlanta, GA, USA, 17 November 2013; Volume 1, pp. 527–534.
6. Panagiotou, N.; Katakis, I.; Gunopulos, D. *Detecting Events in Online Social Networks: Definitions, Trends and Challenges*; Lecture Notes in Computer Science; Springer: Cham, Germany, 2016; Volume 9580, pp. 42–84.
7. Chen, X.; Li, Q. Event Modeling and Mining: A Long Journey Toward Explainable Events. *VLDB J.* **2020**, *29*, 459–482. [CrossRef]
8. Atefeh, F.; Khreich, W. A Survey of Techniques for Event Detection in Twitter. *Comput. Intell.* **2015**, *31*, 132–164. [CrossRef]
9. Allan, J.; Papka, R.; Lavrenko, V. On-Line New Event Detection and Tracking. In Proceedings of the SIGIR '98: 21st Annual ACM/SIGIR International Conference on Research and Development in Information Retrieval, Melbourne, Australia, 24–28 August 1998; ACM: Melbourne, Australia, 1998; pp. 37–45.
10. Makkonen, J.; Ahonen-Myka, H.; Salmenkivi, M. Simple Semantics in Topic Detection and Tracking. *Inf. Retr.* **2004**, *7*, 347–368. [CrossRef] [PubMed]
11. Li, B.; Li, W.; Lu, Q.; Wu, M. Profile-Based Event Tracking. In Proceedings of the SIGIR '05: The 28th ACM/SIGIR International Symposium on Information Retrieval 2005, Salvador, Brazil, 15–19 August 2005; ACM: Salvador, Brazil, 2005; pp. 631–632.
12. Nakade, V.; Musaev, A.; Atkison, T. Preliminary Research on Thesaurus-Based Query Expansion for Twitter Data Extraction. In Proceedings of the ACM SE '18: Southeast Conference, Richmond, KY, USA, 29–31 March 2018; ACM: Richmond, KY, USA, 2018; pp. 1–4.
13. Corney, D.; Martin, C.; Göker, A. Spot the Ball: Detecting Sports Events on Twitter. In Proceedings of the ECIR 2014: Advances in Information Retrieval, Amsterdam, The Netherlands, 13–16 April 2014; Springer: Amsterdam, The Netherlands, 2014; pp. 449–454.
14. Mishra, S.; Diesner, J. Semi-Supervised Named Entity Recognition in Noisy-Text. In Proceedings of the WNUT 2016: The 2nd Workshop on Noisy User-generated Text, Osaka, Japan, 11–16 December 2016; The COLING 2016 Organizing Committee: Osaka, Japan, 2016; pp. 203–212.
15. Yang, Z.; Li, C.; Fan, K.; Huang, J. Exploiting Multi-Sources Query Expansion in Microblogging Filtering. *Neural Netw. World* **2017**, *27*, 59–76. [CrossRef]
16. Zingla, M.A.; Chiraz, L.; Slimani, Y. Short Query Expansion for Microblog Retrieval. *Procedia Comput. Sci.* **2016**, *96*, 225–234. [CrossRef]
17. Albishre, K.; Li, Y.; Xu, Y. Effective Pseudo-Relevance for Microblog Retrieval. In Proceedings of the ACSW 2017: Australasian Computer Science Week 2017, Geelong, Australia, 31 January–3 February 2017; ACM: Geelong, Australia, 2017; pp. 1–6.
18. Massoudi, K.; Tsagkias, M.; de Rijke, M.; Weerkamp, W. Incorporating Query Expansion and Quality Indicators in Searching Microblog Posts. In *Advances in Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6611, pp. 362–367.
19. Letham, B.; Rudin, C.; Heller, K. Growing a List. *Data Min. Knowl. Discov.* **2013**, *27*, 372–395. [CrossRef]
20. Sarmento, L.; Jijkuon, V.; de Rijke, M.; Oliveira, E. "More Like These": Growing Entity Classes from Seeds. In Proceedings of the CIKM '07: Conference on Information and Knowledge Management, Lisboa, Portugal, 6–7 November 2007; ACM: New York, NY, USA, 2007; pp. 959–962.
21. Zhang, Z.; Sun, L.; Han, X. A Joint Model for Entity Set Expansion and Attribute Extraction from Web Search Queries. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; AAAI Press: Phoenix, AZ, USA, 2016; pp. 3101–3107.
22. Wang, R.C.; Cohen, W.W. Language-Independent Set Expansion of Named Entities Using the Web. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, USA, 28–31 October 2007; IEEE: Omaha, NE, USA, 2007; pp. 342–350.
23. Wang, R.C.; Cohen, W.W. Iterative Set Expansion of Named Entities Using the Web. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; IEEE: Pisa, Italy, 2008; pp. 1091–1096.
24. Mamo, N. APD: The tools and data used in the article 'An Automatic Participant Detection Framework for Event Tracking on Twitter'. Available online: https://github.com/NicholasMamo/apd/ (accessed on 29 January 2021).
25. Reed, J.W.; Jiao, Y.; Potok, T.E.; Klump, B.A.; Elmore, M.T.; Hurson, A.R. TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams. In Proceedings of the 5th International Conference on Machine Learning and Applications, Orlando, FL, USA, 14–16 December 2006; IEEE: Orlando, FL, USA, 2006; pp. 258–263.
26. Girvan, M.; Newman, M.E.J. Community Structure in Social and Biological Networks. *Proc. Natl. Acad. Sci. USA* **2002**, *99*, 7821–7826. [CrossRef] [PubMed]
27. Hripcsak, G.; Rothschild, A.S. Agreement, the F-Measure, and Reliability in Information Retrieval. *J. Am. Med Informatics Assoc. JAMIA* **2005**, *12*, 296–298. [CrossRef] [PubMed]

28. Buckley, C.; Voorhees, E. Evaluating evaluation measure stability. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens Greece, 24–28 July 2000; Association for Computing Machinery: Athens, Greece, 24 July 2000; pp. 33–40.
29. Löchtefeld, M.; Jäckel, C.; Krüger, A. TwitSoccer: Knowledge-Based Crowd-Sourcing of Live Soccer Events. In Proceedings of the MUM ’15: 14th International Conference on Mobile and Ubiquitous Multimedia, Linz, Austria, 30 November–2 December 2015; ACM: Linz, Austria, 2015; pp. 148–151.