Building a Large-scale Corpus for Evaluating Event Detection on Twitter

Andrew J. McMinn
School of Computing Science
University of Glasgow
Glasgow, Scotland
a.mcminn.1@research.gla.ac.uk

Yashar Moshfeghi
School of Computing Science
University of Glasgow
Glasgow, Scotland
yashar.moshfeghi@glasgow.ac.uk

Joemon M. Jose
School of Computing Science
University of Glasgow
Glasgow, Scotland
joemon.jose@glasgow.ac.uk

ABSTRACT

Despite the popularity of Twitter for research, there are very few publicly available corpora, and those which are available are either too small or unsuitable for tasks such as event detection. This is partially due to a number of issues associated with the creation of Twitter corpora, including restrictions on the distribution of the tweets and the difficultly of creating relevance judgements at such a large scale. The difficulty of creating relevance judgements for the task of event detection is further hampered by ambiguity in the definition of event. In this paper, we propose a methodology for the creation of an event detection corpus. Specifically, we first create a new corpus that covers a period of 4 weeks and contains over 120 million tweets, which we make available for research. We then propose a definition of event which fits the characteristics of Twitter, and using this definition, we generate a set of relevance judgements aimed specifically at the task of event detection. To do so, we make use of existing state-of-the-art event detection approaches and Wikipedia to generate a set of candidate events with associated tweets. We then use crowdsourcing to gather relevance judgements, and discuss the quality of results, including how we ensured integrity and prevented spam. As a result of this process, along with our Twitter corpus, we release relevance judgements containing over 150,000 tweets, covering more than 500 events, which can be used for the evaluation of event detection approaches.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: Collection; H.3.3 [**Information Search and Retrieval**]: Clustering; Information Filtering

Keywords

Test Collection; Twitter Corpus; Event Detection; Social Media; Mechanical Turk; Crowdsourcing; Reproducibility

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2505695.

1. INTRODUCTION

With the phenomenal growth of Social Media, a large amount of real-time data has become available about ongoing real-world events. Twitter is perhaps the most popular microblogging platform in the world, with over 200 million active users, and 400 million tweets posted each day[1]. The real-time nature and massive volume of data has focused the attention of many researchers on Twitter. Sakaki et al. [17] were able to use Twitter as a social-sensor to detect the size and direction of earthquakes in real-time, notifying users of incoming earthquakes much faster than even the Japan Meteorological Agency. Hu et al. [9] demonstrated the effectiveness of Twitter as a medium for breaking news, and found that news of Osama bin Laden's death not only broke on Twitter, but informed millions of his death before the official announcement.

Despite the interest in Twitter, there are only a small number of corpora available, none of which are suitable for the large-scale evaluation of event detection approaches due to their small size or small number of events. This is partially due to the massive scale of Twitter, which makes the creation of corpora difficult, time-consuming and expensive. Furthermore, Twitter's Terms of Service restrict the distribution of tweets, and do not allow the content of tweets to be distributed as part of a corpus[2]. As a result, there are very few publicly available Twitter corpora, and in some cases, corpora from other medium are used in place of a Twitter corpus [2, 13, 14]. However, it is not clear that effectiveness on a non-Twitter corpus is comparable to effectiveness on a Twitter corpus, with some evidence suggesting that this is not always the case [14].

In order to tackle this problem, we have created a large collection of 120 million tweets, which we make available in a manner similar to the TREC Microblog Track [11], releasing only User ID and Tweet ID pairs which can be used to crawl Twitter. The corpus we present is an order of magnitude larger than currently available Twitter corpora, and contains relevance judgements for over 150,000 tweets, covering more than 500 events – considerably more than the largest Twitter event corpus currently available [14].

Event Detection is one of the most commonly studied tasks on Twitter [2, 5, 21, 19, 6, 13, 10]. Twitter provides a real-time stream of updates, opinions, and first-hand re-

[1] `https://blog.twitter.com/2013/celebrating-twitter7`

[2] `https://dev.twitter.com/terms/api-terms` Section I.4.A.: *"If you provide downloadable datasets of Twitter Content [...] you may only return IDs (including tweet IDs and user IDs)."*

ports of what is happening – something newswire documents simply cannot compete with. This makes it very desirable to develop systems that are able to detect and track events from Twitter streams. However, there is disagreement on the definition of event, which makes comparison of different event detection approaches very difficult – one system may consider something to be a single event, while others may break it into multiple events. Additionally, there is no standard corpus which is used, and most works require the creation of a bespoke corpus which is often not made available for use by others due to the reasons mentioned above. This means that time and resources are wasted, motivating the need for a corpus which can be used for the evaluation and comparison of event detection approaches.

To solve this, we propose a new definition of event which better fits how Twitter is used to discuss events. We then study the problem of creating a set of relevance judgements for the evaluation of event detection approaches, which uses our new definition of event. To do so, we use a number of existing event detection approaches and the Wikipedia Current Events Portal to generate a set of candidate events and associated tweets. We then use crowdsourcing to evaluate if a candidate event fits our definition of event, and create relevance judgements for each of the events. Finally, we analyze the quality of the resulting judgements, and make the corpus and judgements available for research and further development.

This paper has a number of novel contributions: (i) we create a large-scale test collection for Twitter (ii) we examine the conflicting definitions of "event" and give a concrete definition which fits the characteristics of Twitter (iii) we propose a novel methodology for the creation of relevance judgements using state-of-the-art event detection approaches, Wikipedia, and Mechanical Turk and (iv) we study the characteristics of the corpus and the generated relevance judgements.

The remainder of the paper is as follows. In section 2, we describe existing Twitter corpora and discuss why they are unsuitable for the large-scale evaluation of event detection systems on Twitter. We also propose a new definition for "event", and describe the event detection task. In Section 3, we describe the methodology used for the creation of the corpus, including the selection of events and the creation of relevance judgements. In Section 4, we describe the characteristics of the corpus, evaluate the effectiveness of our methodology, and examine the quality of the results. Finally, in Section 5, we conclude and discuss future work.

2. BACKGROUND

In this section, we examine events on Twitter. First, we examine existing corpora with an emphasis on their use for the detection and analysis of events, and describe the issues involved with the creation of a corpora specifically for this purpose. Secondly, we discuss the definition of *event*, the issues associated with past and present definitions, and propose a new definition of *event* which better fits the characteristics of event-based discussion on Twitter. Finally, we examine event detection on Twitter, and more precisely define the task of event detection.

2.1 Existing Twitter Copora

In this section, we examine currently available Twitter corpora, paying particular attention to their suitability for the evaluation and analysis of event-based systems. We also discuss some of the challenges associated with the creation of a Twitter-sized corpus.

TREC ran a Microblog Track in 2011 with an ad-hoc retrieval task, and again in 2012 with the addition of a filtering tasks. The Tweets2011 [11] corpus was used both years, with new topics and relevance judgements being generated each year. The collection was the first publicly available, large-scale Twitter corpus, containing 16 million tweets covering a period of 2 weeks. However, the corpus contains tweets in all languages, and once non-english tweets have been removed, only around 4 million tweets remain. Furthermore, the corpus is designed specifically for ad-hoc retrieval, and as such, the topics and relevance judgements are unsuitable for event-based analysis. The track is running again in 2013, however it is moving to an experimental track-as-a-service approach, where the corpus will be hosted by TREC and participants query it using an API.

Becker at al. [5] produced what we believe was the first Twitter collection of events, however it only contains tweets posted by users in New York. This clearly introduces geographical bias and restricts the type of events available. The collection itself is also relatively tiny, containing only 2.6 million tweets.

Petrović et al. [14] created a corpus aimed at First Story Detection, and while their collection contains a relatively high 50 million tweets from the beginning of July 2011 until mid-September 2011, they identified only 27 events. This means that large-scale comparisons are difficult as there is only a small sample of events and failure to detect only a small number of these could result in unsubstantiated and misleading results.

Although these collections have been made available, none appear suitable for the analysis of events and comparison of event detection approaches. One reason for the lack of comparative corpora may be the difficulty and expense of creating one. A reasonable sized Twitter corpus will contain tens of millions of documents – performing a manual search on a corpus of that magnitude is simply impossible. To overcome this, Petrović et al. [14] used a procedure similar to NIST, where expert annotators read the description of an event and use keywords to search for relevant documents. However, this approach means that events must be carefully identified in advance, annotation requires expensive experts, and it does not scale well past a certain size.

This demonstrates the need for a method of creating large-scale corpora which can be used for the comparison of event detection approaches, and which is not extremely expensive or time consuming. We believe that by using crowdsourcing, we can reduce the time and cost required to produce large-scale corpora.

2.2 Definition of Event

Despite the significant interest in events, there is little consensus on the exact definition of *event*. This leads to issues when evaluating and comparing event-based systems.

The *Topic Detection and Tracking* (TDT) project defines an *event* as something that happens at some specific time and place, and the unavoidable consequences [1]. Specific elections, accidents, crimes and natural disasters are examples of events under the TDT definition. They also define an *activity* as a connected set of actions that have a common focus or purpose. Specific campaigns, investigations,

and disaster relief efforts are examples of activities. Furthermore, in TDT, a *topic* is defined as a seminal event or activity, along with all directly related events and activities.

Aggarwal and Subbian [2] define a *news event* as being any event (something happening at a specific time and place) of interest to the (news) media. They also consider any such event as being a single episode in a larger story arc. For example, a speech at a rally might be an event, but it is an episode in a larger context: a presidential election. They use the term *episode* to mean any such event, and *saga* to refer to the collection of events related within a broader context.

Becker et al. [6] define *event* in a much more formal, but still entirely subjective manner. They define an event as a real-world occurrence e with (1) an associated time period T_e and (2) a time-ordered stream of Twitter messages, of substantial volume, discussing the occurrence and published during time T_e. Other definitions, such as that used by Weng et al. [19], define an event simply as a burst in the usage of a related group of terms.

Clearly there is a consensus that events are temporal, as time is a reoccurring theme within all definitions. However, the consensus appears to end there. Whilst Aggarwal and Subbian, and the TDT definition show a parallel in their hierarchical organization of events (events and topics, news events and sagas), this is less common in other definitions where a distinction between events and topics is not made. This makes comparisons very difficult; one definition may break an election into many events, while another could consider the election as a single event, or not as an event at all.

To solve these issues, we take the most basic definition of event (something that happens at some specific time and place), and introduce the requirement that an event should be *significant*. By requiring that an event is significant, we are able to filter out every-day personal and trivial events which are extremely common on Twitter.

Definition 1. An *event* is a **significant** thing that happens at some specific time and place.

It was reasonable to assume that all events in the TDT datasets were significant events due to the use of newswire documents, something which is not true in the case of Twitter. Given this, we model our definition of significance so that an event under our definition would be of similar significance to those found in the TDT datasets, given the disparity between the two sources.

Definition 2. Something is *significant* if it may be discussed in the media. For example, you may read a news article or watch a news report about it.

It is important to note that something does not necessarily have to be discussed by the media in order for it to be an event, we simply use this as an indication of the level of significance required for something to be considered an event. Whilst this is still somewhat subjective, we believe that it is impossible to further restrict significance whilst keeping our definition of event generalizable. Given this definition of event, our goal is to create a collection of significant events which have been discussed on Twitter, and a set of relevance judgments for tweets which discuss the events.

2.3 Event Detection

Event detection was extensively researched as part of the Topic Detection and Tracking (TDT) project, which dealt with the event-based organization of newswire stories. Event detection in microblogs is conceptually very similar to the clustering task [3] (more commonly referred to as *detection*) from the TDT project. In both, a system is presented with a continuous stream of time ordered documents, and must place each of them into the most appropriate event-based cluster. The only difference between the two tasks is the type and volume of documents in the stream – however in practice this makes a great deal of difference, and event detection in microblog streams is considerably more challenging for a number or reasons. Firstly, the volume of documents is several orders of magnitude greater in microblogs, which means that event detection systems must be extremely efficient to run in real-time. Secondly, the majority of microblog posts discuss mundane, every day occurrences which are not noteworthy or of interest. These documents must be filtered out so that only interesting real-world events are detected – an issue which was not present for participants of the TDT evaluations. Furthermore, microblog posts tend to be very noisy, of very limited length (tweets are restricted to 140 characters) and frequently contain spelling or grammar errors. These differences mean that approaches developed for the TDT project tend to be far too slow for real-time application, and extremely vulnerable to the noise found in microblog streams.

Petrović et al. [13] make use of Locality Sensitive Hashing (LSH), which places similar documents into the same bucket of a hash table. Using their method it is possible to reduce the size of the candidate set to a fixed size which contains the nearest neighbour with a high probability. Clustering can then be performed in $O(1)$ time, using a variance reduction technique to improve clustering performance if no neighbour is found within a certain distance. This is one of the state-of-the-art approaches used in the work, and is described in more detail in Section 3.

Aggarwal and Subbian [2] use a fixed number of clusters and cluster summaries to reduce the number of comparisons required for document clustering. They use a novel similarity score which exploits the underlying, graph-based structure of Twitter to create a similarity metric which improves upon content-only similarity measures. Events are detected by tracking the growth rate of clusters, and marking bursty clusters as events.

Weng et al. [19] transform term statistics into wavelets, and then calculate the cross-correlation for each term, mapping changes in the term's usage over time. They create a graph of terms with large correlation values, and perform graph partitioning to create clusters of terms which discuss the same event. However, their approach seems to be very sensitive to parameter tuning, with very small changes having a significant impact on effectiveness.

Becker et al. [6] use a clustering approach proposed as part of the TDT project by Yang et al. [20]. They then use a manually trained classifier to identify event clusters, looking at features such as retweets and hasthags to identify the most likely events.

Of these, only [13] was evaluated on a publicly available Twitter dataset, however this was done after its initital publication and on a collection which we argue is not suitable for the comparison of event detection approaches (see Section 2.1 for details). Factors which affect difficulty of event detection, such as variations in the volume of tweets used and the period of time covered, means that a fair comparison be-

tween these approaches is impossible. Furthermore, because of the lack of a coherent definition of event, it is not even clear if these approaches are attempting to detect the same type of event. This further motivates the need for a corpus designed specifically for the evaluation and comparison of event detection approaches.

3. BUILDING THE CORPUS

We collected tweets using the Twitter Streaming API for 28 days, starting on the 10th of October 2012 and ending on the 7th of November 2012. This period was chosen specifically because it covers a number of interesting and significant events, including Hurricane Sandy, and the U.S. Presidential Elections. Language based filtering was performed using a language detection library for Java[3] to remove non-English tweets. Further filtering was performed to remove common types of Twitter spam (i.e., tweets which contain more than 3 hashtags, more than 3 mentions, or more than 2 URLs, empirically chosen due to [7]). After spam and language filtering, we were left with 120 million tweets.

Of the 120 million tweets in the corpus, around 30% (40 million) are *retweets*. A retweet is a copy of someone else's tweet which was broadcast by a second user to their followers. In the context of Twitter, retweets are a very useful method of spreading information. However, retweets are commonly associated with the spread of spam [8], and because they are an unmodified copy of someone else's tweet, they do not generally add any new information. Given this, and in order to reduce the complexity of creating relevance judgements, we chose not to include retweets in the relevance judgements (however they remain in the corpus)[4].

The remainder of this section details the approach used to generate a list of events and relevance judgements for the corpus. We begin by describing the approach used to generate a set of candidate events and associated tweets. We then describe the crowdsourced evaluation and discuss a number of spam detection and prevention techniques. Finally, we show how events from different sources are merged to create the final set of relevance judgements.

3.1 Candidate Event Generation

Rather than create our own list of events, we chose to use a number of existing event detection approaches and the Wikipedia Current Events Portal[5] to create a pool of events. In the remainder of this paper, we refer to the event detection approaches as *detection approaches*, and the use of the Wikipedia Current Events Portal as the *curated approach*.

We choose to use 2 detection approaches, namely the Locality Sensitive Hashing approach proposed by Petrović et al. [13] and the Cluster Summarization approach proposed by Aggarwal and Subbian [2]. These were chosen based upon a number of desirable characteristics. Firstly, both approaches produce clusters of documents, rather than clusters of terms. While there are a large number of event detection approaches, the many product clusters of *terms*, which is considerably less useful in our case. Secondly, we believed that clusters produced by these approaches could be easily combined due to somewhat similar levels of event granularity. Finally, both approaches are fast, and we were confident that they would finish in a reasonable time frame (i.e., days rather than weeks or months).

While it would have been desirable to implement additional detection approaches, the time taken to implement, run and evaluate each approach is prohibitive. Wikipedia maintains a Current Events Portal, which provides a curated list of events from the around the world. The use of Wikipedia offers a number of advantages over the use of more detection approaches. Firstly, each of the events listed on the current events portal is substantiate by a link to a news article from a reputable news source. This allows a high level of confidence that the events are accurate and significant under our definition. Secondly, much of the work has already been done for us by unpaid editors and is of a high quality, ensured by Wikipedia's editorial guidelines. This means that we do not have to pay workers to evaluate non-events, reducing the cost and time taken to produce judgements. Additionally, because Wikipedia provides a broad set of events covering many topics and categories, it helps to increase event coverage (discussed in Section 4).

The remainder of this section details both detection approaches and the curated approach, and describes how they were used to generate a set of candidate events.

3.1.1 Locality Sensitive Hashing (LSH)

Petrović et al. [13] make use of Locality Sensitive Hashing (LSH), which places similar documents into the same bucket of a hash table. Documents are hashed using a certain type of hash function:

$$S(x) = \{y : h_{ij}(y) = h_{ij}(x), \exists i \in [1..L], \forall j \in [1..k]\}$$

where hash functions h_{ij} are defined as:

$$h_{ij}(x) = sgn(\mathbf{u}_{ij}^T x)$$

with the random hyperplanes $\mathbf{u}_{ij}$ being drawn independently from a Gaussian distribution for each i and j.

Using multiple hash tables it is possible to reduce the size of the candidate set to a fixed size which contains the nearest neighbour with a high probability. Clustering can then be performed in $O(1)$ time. In cases where no neighbour is found within a certain distance, a variance reduction technique is used which has been shown to vastly improve clustering effectiveness [13].

Shannon entropy is used to measure the amount of information in the cluster, and clusters with small entropies (< 2.5) are moved to the back of the list of possible events. When ranking events, they found that ranking by the number of unique users gives better performance than other measures, such as number of Tweets.

We ran the algorithm over our corpus using parameters very similar to those used by Petrović et al. [13]. More precisely, 13 bits per key, a maximum distance of 0.45 and 70 hash tables. However, we chose to measure the fastest growing clusters on a hourly basis, rather than every 100,000 tweets as used in the original paper. We made this decision due the fact that 100,000 tweet covers only a short period of time in our collection (around 30 minutes) because of its high density. This would have generated many more candidate events without necessarily increasing the number of actual events, making it prohibitively more expensive to generate judgements.

[3] https://code.google.com/p/language-detection/

[4] Information on how to obtain the corpus can be found on the University of Glasgow Multimedia IR group website http://mir.dcs.gla.ac.uk/resources/

[5] http://en.wikipedia.org/wiki/Portal:Current_events

For each hour long time period, the clusters were ranked by the number of unique users, and clusters with low entropy were moved to the back of the list. Simply taking this list would have yielded a prohibitively high number of clusters, however by removing clusters outside the optimal entropy range (3.5 - 4.25)[13], the list is reduced to a manageable size of 1340 candidate events.

3.1.2 Cluster Summarization (CS)

Aggarwal and Subbian [2] use a fixed number of clusters and cluster summaries to reduce the number of comparisons required for document clustering. Each cluster summary contains (i) a node-summary, which is a set of users and their frequencies and (ii) a content-summary, which is a set of words and their TF-IDF weighted frequencies. By combining these two summaries, they suggest a novel similarity score which exploits the underlying structure of the social network and improves upon content-only measures. A sketch-based technique is used to maintain node statistics and calculate structural similarity at a significantly reduced cost. The similarity score between document D and cluster C is given by:

$$Sim(D, C) = \lambda \cdot SimS(D, C) + (1 - \lambda) \cdot SimC(D, C)$$

Where $SimS$ and $SimC$ are structure and content similarity measures respectively, and λ is a balancing parameter in the range (0,1).

Each incoming document is assigned to its closest cluster unless its similarity score is significantly lower than that of other documents. A similarity score is considered significantly lower if it falls below $\mu - 3 \cdot \sigma$, where μ is the mean of all previous similarity scores, and σ is the standard deviation [15].

We ran the CS algorithm [2] with an input size of 1200 clusters. We selected this number of clusters as it gave a reasonable runtime of approximately 4 days for the entire collection on the hardware available to us. Retweets were not used as input to the algorithm as we found, in line with the literature [8], that they tend to cause more spam and meme clusters to be identified as events. We used a λ value of 0.0 (i.e., full weight was given to text when calculating similarity) as we were unable to obtain user information due to rate limiting of the Twitter API.

Similar to the LSH approach, we removed clusters with less than 30 tweets, and those with α values smaller than 12 (i.e., slow growth rates) [2]. Empirically we found that very few clusters with an α value below 12 discussed an event, and by removing these clusters we were able to significantly reduce the number of candidate events to a manageable size of 1097.

3.1.3 Wikipedia Current Events Portal (CEP)

The Wikipedia Current Events Portal[6] provides a detailed list of significant events from around the world. Each event is given as a short description (usually around 1 sentence in length), a category, and a link (or links) to a relevant news article. An example event is shown below:

- **Date:** October 25, 2012
- **Category:** Business and economics
- **Description:** Official [[GDP]] figures indicate the [[2012 Summer Olympic]] helped the [[UK economy]] emerge from recession in the three months from July to September, with growth of 1.0%.
- **Reference:** `http://www.bbc.co.uk/news/business-20078231`

Note: Terms enclosed by [[and]] are links to other Wikipedia pages.

This approach is very different from the detection approaches as we already already have a list of events and need to retrieve relevant tweets. To do this, we indexed the corpus using Lucene 4.2[7]. Stop words and URLs were removed, porter stemming was applied, and prefixes (#, @ characters) were remove from hashtags and mentions.

We then used the description from each of the Wikipedia events as an initial query to retrieve tweets which potentially discuss the event. Query expansion was performed to decease lexical mismatch and was used by some of the best performing approaches in the TREC Microblog track for 2011 [4] and 2012 [18]. In particular, we expand links to other Wikipedia pages to the full title of that page (e.g., "UK economy" -> "Economy of the United Kingdom"), and expand/contract acronyms (e.g., "U.K." -> "United Kingdom", "United States" -> "U.S."). Furthermore, terms used as links to other pages were given double weighting as they are generally the most important and contextual terms in the description. Divergence from Randomness using Inverse Document Frequency as the basic randomness model was used for retrieval as experimentation using the TREC11 corpus showed that, of the retrieval models included with Lucene 4.2, it gave the best retrieval performance.

For each of the 468 events on the Wikipedia Current Events Portal listed between the dates covered by the corpus, we retrieved the top 2000 tweets from a window of 72 hours, centered around the day of the event (i.e., for an event on the 16th of October, retrieval was restricted to tweets posted between the 15th and 17th of October inclusively).

3.2 Generating Relevance Judgements

This section describes the methodology behind our crowdsourced evaluation using Amazon Mechanical Turk. The aim of the evaluation was to decide which of the candidate events fit our definition, and generate a set of relevance judgements for each of the events. We also wanted to gather descriptions and category information for each event, both of which are used at later stages to merge events from different sources (discussed in Section 3.3) and to evaluate the coverage of the collection.

The following relevance statement was used for all evaluations:

> Anything that discusses the described event is considered relevant, even if the information is now out-of-date or does not necessarily match that given in other tweets (e.g., the number of deaths is different). However, care should be taken to mark any untrustworthy or obviously false statements as non-relevant. Tweets which give a user's opinion of an event and are obviously discussing the event but do necessarily describe what happened are still considered relevant.

The definition was intentionally very open as we wanted to capture as many tweets about each event as possible.

[6] `http://en.wikipedia.org/wiki/Portal:Current_events`

[7] `http://lucene.apache.org/`

Specifically, as well as objective tweets, we wanted to include subjective tweets (i.e., the opinion of users) as they are one of the most defining characteristics of Twitter, and one of the reasons why Twitter is so popular for event detection.

The remainder of this section is split into two. First, we describe the methodology used for the detection approaches, including a description of the pilot evaluations and changes these motivated to the final evaluation. Secondly, we describe the methodology used for the curated approach, including the pilot evaluations, and the rational behind our incremental approach to generating relevance judgements.

3.2.1 Evaluation of Detection Approaches

As we wanted to gather category information for each of the events, this meant that we had to decide on a set of categories. Although the Wikipedia Current Events Portal does assign categories, there is a very large number of categories, each of which is very specific (e.g., History, Literature, Spirituality). Rather than forcing annotators to choose between a large number of very specific categories, we chose to use the categories defined by the TDT project [1]. The 13 categories defined by the TDT project cover a wide range of topics, with a Miscellaneous category for events which do not fit elsewhere. The full list is shown in Table 1. The choice of 5 annotators per candidate event was made so that the minimum majority would not be less than 3, so that each tweet would be judged a minimum of 3 times.

Pilot Evaluations We ran 2 sets of pilot evaluations, each using 20 carefully selected candidate events covering many different categories and with varying degrees of difficult or ambiguity. Several candidate events were selected specifically because they were difficult to judge and fell between event and non-event (i.e., very subjective), while other candidates were selected because they were particularity unclean (i.e., event based discussion mixed with discussion of an unrelated topic). Other candidates were selected because they were difficult to categorize (e.g., debates between presidential candidates).

We found that our evaluation performed as expected, and judgements given by annotators matched closely with our assessment of the candidates. We noted that some annotators were submitting responses very quickly, and were clearly not reading the tweets in detail, if at all. To solve this, we added a minimum time limit of 20 seconds, and prevented the annotators from submitting before the time had elapsed. We also noticed that agreement between annotators when selecting categories for the events was low, which motivated us to include example events for each category, taken from the TDT annotation guidelines [1].

Although it would have been desirable to have each annotator judge a large number of tweets, it is unreasonable to expect crowdsourced workers to judge a large number of tweets without becoming fatigued. To prevent fatigue, we chose to keep the time taken to perform an evaluation under 1 minute. Initially we chose to use 30 tweets per evaluations as this had been the number estimated by TREC Microblog Track [12] as the number of tweets a single user would be willing to read. However, this caused the time taken to read the tweets and make a judgement about the event as a whole (i.e., does the candidate fit our definition of event?) to be considerably over 1 minute on average. To solve this, we reduced the number of tweets shown per evaluation to 13.

Full Evaluation For each candidate event we asked 5 annotators to perform a questionnaire. Each annotator was asked to read 13 tweets (selected at random however kept constant between evaluations) from a single candidate event. They were then asked: "Do the majority of tweets discuss the same real-life topic?" If they responded with "no" then the evaluation was complete and they were allowed to submit the evaluation. If they answered "yes", then a further question was posed: "Do the tweets discuss an event?" At this point, they were also reminded of our definition of event. Again, annotators who answered "no" were allowed to submit the evaluation. However, if they answered "yes" (signaling this the candidate is an event), then they were asked to re-read the tweets and mark any non-relevant tweets as so. They were then asked to briefly describe the event and select the category which fits the event. Assuming they had completed all of the above, they were allowed to submit the evaluation.

3.2.2 Evaluation of Curated Approach

Unlike the detection approaches, we already know that each of the candidates from the curated approach is an event, and are only lacking relevance judgements for tweets, allowing us to have more tweets judged in out 1 minute time limit. This allowed us to present tweets in batches of 30, ordered by their rank as given by the DFR retrieval model. Rather than have all of the batches annotated simultaneously, we chose to use an incremental approach, inspired by the methodology used by the TDT project. Once relevance judgements for a batch had been obtained, an automatic decision is made based upon the number of relevant tweets. If at least 50% of the tweets in a batch are marked as relevant, then the next batch is suitable for annotation. On the other hand, if less than 50% of tweets are marked as relevant, then annotation is stopped and the event is marked as complete. This process is repeated until all events have been marked as complete.

Pilot Evaluations In order to determine if our stopping point was effective, we created a pilot study where annotators were shown tweets which were ranked below the automatic cutoff point (i.e., where there were very few or no relevant tweets). Interesting, the number of tweets marked as relevant by annotators was generally very high, often above 50%. We believe that the majority of annotators actually became confused by the lack of relevant tweets, and created their own pseudo-topic based upon the tweets being shown to them. For example, where the event described a mass shooting in Nigeria, all 3 annotators seemed to switch to another, unrelated event, leaving only tweets discussing a bombing at a church (also in Nigeria) as relevant. This indicates that continuing to ask for annotations after our cutoff point would actually harm the accuracy of results, rather than improve them.

We also ran a number of small pilots to test the best method of gathering judgements (i.e., mark relevant, mark non-relevant, or select relevant/non-relevant for each tweet). We found that all 3 options gave similar results, with selecting relevant/non-relevant for each tweet giving very slightly more accurate results. However, selecting relevant/ non-relevant for each tweet is significantly more work than selecting only the relevant or non-relevant tweets, and fatigues annotators much faster than the other methods. Of the two remaining methods, (i.e., selecting relevant or selecting non-relevant), we chose to use the selection of non-relevant

tweets, as our honey-pot spam detection technique requires it (described in section 3.2.3).

Full Evaluation For each batch, we asked 3 annotators to read the Wikipedia description of the event. A link to a relevant news article (also taken from Wikipedia) was shown to the worker as an additional source of information should the description unclear or for verifying information found in tweets. They were then asked to briefly describe the event in their own words, as if they were entering a query into a search engine. Finally, they were asked to read the tweets, marking any non-relevant tweets as so.

3.2.3 Detecting and Preventing Spam

For all evaluations, a 20 second minimum time limit was enforced to deter low-quality submission for easy money, the intuition being that a worker who was only interested in making quick money would not be willing to wait between successive HITs. We also developed several methods of detecting spam submissions so that spam evaluations could be removed and re-run. We employed a honey-pot technique to detect workers who were not correctly classifying tweets as relevant or non-relevant. We inserted a tweet from a preselected set of spam tweets which were known not to discuss an event. Because the worker had already indicated that the tweets discussed an event, we could be sure that the spam tweet was non-relevant. If the worker did not identify this tweet as being non-relevant then their evaluation was marked as being spoiled and re-run. Of those evaluations submitted as events, 999 (out of 22,114) were marked as being spoiled (i.e., the worker failed to identify the honey-pot).

For the detection approaches, we applied user-level spam detection, and attempted to remove annotations by workers who were submitting large numbers of low-quality evaluations. By examining the ratio of evaluations performed to the number of clusters marked as events, we were able to identify outliers to who had identified either significantly more or significantly less events than the average worker. We removed workers who had performed over 75 evaluations and had given more than 90% "yes" answers, or over 90% "no" answers. This resulted in the removal of 12 workers who had performed 4560 evaluations in total. This amounts to around 35% of the total number of evaluations for the detection approaches. Interestingly, we noted that of the 12 workers removed due to spam, 9 appeared in the top 10 workers by number of evaluations performed. This suggests that limiting the number of evaluations which a single worker can perform could be an effective method of reducing noise and spam.

3.3 Combining Events from Multiple Sources

Each of the methods produces a different set of events, however these are not disjoint sets, and there is a significant overlap in results which are produced. For example, the third U.S. Presidential debate is included in the results of all methods, meaning that there are at least 3 sets of tweets which all discuss the same event. Additionally, each method can produce multiple results for the same event. For example, the LSH algorithm appears to produce no fewer than 40 sets of tweets for the third U.S. Presidential debate, and although each of these could potentially be mapped down to specific sub-events within the debate (such as individual questions, quotes, etc.), it would be desirable to cluster all of these together so that they fit under our definition of event. Given this, we attempt to cluster events together, both from different sources (e.g., the LSH and CS algorithms) and the same sources (i.e., 2 sets of tweets from the LSH algorithm), such that they fit our definition of event as closely as possible.

3.3.1 Clustering Features

There are a number of features which could be used to cluster of events. This section describes the features, their advantages, and any issues that could arise from their use for combining events.

Category Features Categorisation is somewhat problematic for the detection approaches as there was a large amount of disagreement between annotators. Going back to our example of the third U.S. Presidential Debate, we noted that the LSH approach produced over 40 sets of results for that single event, many of which referred to specific sub-events within the debate. Many of the subjects of the debate were related to economics, war, business, and international relations. This is an issue because annotators often categorised results based upon the topic of discussion (e.g., New Laws, Political and Diplomatic Meetings), rather than based on the event which caused the debate to take place (the Election). Despite it being clear at a higher level that the debate should be categorised as Election, it is not so clear at the level of sub-events and specific moments within the debate – the categorisation changes as we change the level of granularity. This makes the use of categorisation difficult for clustering as it is not immediately clear how War and Conflicts could be related to Election, and means that we cannot simply say that different categories mean that the events are different.

Additionally, because Wikipedia defines its own set of categories, we must create a mapping between the TDT categories and the Wikipedia categories if we wish to measure category similarity. Creating a direct mapping between TDT categories and Wikipedia categories would solve the mapping problem but not increase agreement between the detection approaches. Thus, we created a new set of categories, each of which covers a much broader range than either the TDT or Wikipedia categories. Table 1 shows the new categories and the corresponding categories from the TDT project and Wikipedia. These categories greatly increase agreement, which is discussed in detail in Section 4.2.

Temporal Features Clearly temporal features are going to be extremely important in the clustering of events – results which have a significant period of time between them are unlikely to be related to the same seminal event. For example, all 3 U.S. Presidential Debates are all likely to be very similar in terms of category and content-based features. The largest defining factor is the specific time at which each debates took place. On the other hand, events which share similar characteristics in terms of both category and content-based features are still relatively common in the same temporal region. Sports events are the best example of this type of event – it is not uncommon for two football matches to occur simultaneously, such as World Cup qualifying matches. These share the same category, the same temporal region, and likely share similar content features, making them particularly difficult to distinguish between.

Empirically, we found that the use of temporal proximity as a feature for measuring similarity was harmful, and re-

Combined Categories	TDT Categories	Wikipedia Categories
Business and Economy	Financial News	Business, Economics
Law and Politics	Elections, Political and Diplomatic Meetings, Legal/Criminal Cases, New Laws, Scandals/Hearings	International relations, Human rights, Law, Crime, Politics, Elections
Science and Technology	Science and Discovery News	Exploration, Innovation, Science, Technology
Arts, Culture and Entertainments	Celebrity and Human Interest News	Arts, Culture, Literature, Religion, Spirituality
Sports	Sports News	Sports
Disasters and Accidents	Accidents, Natural Disaster	Accidents, Disasters
Armed Conflicts and Attacks	Acts of Violence or War	Armed conflicts, Attacks
Miscellaneous	Miscellaneous News	*Anything which is not listed above. e.g., Heath, Transport*

Table 1: The categories used in our approach (the combined categories), shown with their corresponding TDT and Wikipedia Categories.

sulted in a large number of false matches, a very undesirable property. Given this, rather than using temporal proximity as an indication of a relationship, we do the converse, and say that events which are temporally dissimilar are unlikely to be related.

Content-based Features Content based features are some of the most distinguishing features of each event, which can make them difficult to use for event clustering. This problem is most pronounced when comparing results from a detection approach as content based features have already been used to perform clustering at a document (tweet) level. This means that each of the clusters are generally dissimilar in terms of content based features, even when discussing the same event, and often contain very distinct sets of terms. This means that content based similarity measures are generally ineffective when clustering results from the same approach. For example, returning to the U.S. Presidential Debates as an example, the quote "Well, Governor, we also have fewer horses and bayonets, because the nature of our military's changed." was extremely popular with Twitter users, and very relevant to one of the debates. However, matching this back to the election is very difficult if the context is not known, and using only the content to match it to other results from the debate is very difficult, if not impossible. This means that we cannot rely on the content of tweets alone when performing clustering.

In additional to tweets, we also have the event descriptions given by worker from the crowdsourced evaluation. These tend to be much higher level and seem to be a more viable feature for clustering events. Additionally, the descriptions often contain named entities (such as people or places), which are generally constant features regardless of the event granularity.

3.3.2 Clustering Algorithm

For each candidate event e, our algorithm calculates its similarity against every other event e' within a time window T_{time}. The similarity is calculated as:

$$S_{con} = max(escore(e, e'), dscore(e, e'))$$

$$S_{cat} = cscore(e, e')$$

$$S_{tweet} = tscore(e, e')$$

$$S_{full} = 0.3 * S_{cat} + 0.4 * S_{con} + 0.3 * S_{tweet}$$

where *dscore* gives the similarity between event descriptions as given by annotators and *escore* gives the similarity between named entities, also from the event descriptions. *cscore* gives the similarity between the categories assigned to the event, and *tscore* gives the similarity between the top 10 most frequent terms in relevant tweets for both events. In every case, cosine similarity is used. The weighting parameters were chosen empirically so that no one feature would be enough to cause a match, reducing the chance of a false match. Features based upon the descriptions given by annotators were given slightly more weight than other features because of its high-level nature,

If two events are found to have an S_{full} value above threshold T_{sim} then they are clustered together. If both e and e' already belong to a cluster then the clusters are merged. The algorithm is shown as pseudo-code in Algorithm 1.

ALGORITHM 1: Event Clustering Approach

```
done = emptyset;
foreach event e in candidates do
    add e to done;
    foreach event e' in candidates but not in done do
        S_con = max{escore(e,e'), dscore(e,e')};
        S_cat = cscore(e,e');
        S_tweet = tscore(e,e');
        S_full = 0.3 * S_cat + 0.4 * S_con + 0.3 * S_tweet;
        if S_full >= T_sim and time_diff(e,e') <= T_time then
            if neither e nor e' in cluster then
                create new cluster containing e and e'
            else
                add e or e' to existing cluster
            end
        end
    end
end
```

3.3.3 Experimentation

Candidates from the detection approaches are considered to be an event if more than 50% of annotators marked it as so and it has a tweet precision greater than 0.9, ensuring that only high-quality events are used. This resulted in 382 events for the LSH approach, and 53 for the CS approach. Candidates from the curated approach were considered an event if they produced at least 1 relevant tweet, resulting in

361 events. In total, this produced 796 events before clustering. Individual tweets are regarded as relevant if more than 50% of annotators agreed. Table 3 shows the distribution of tweets broken down by both approach used and type (implicit and explicit). This gave 4,009 explicit and 93,398 implicit judged tweets for the LSH approach, 465 explicit and 15,098 implicit judgments for the CS approach, and 39,980 explicit judgments (with no implicit judgments) for the curated approach. Although the use of implicit judgments will have introduced some noise to the relevance judgments, because we remove candidates with low precision we are able to minimize noise whilst increasing the number of judgments by over 200%.

We then ran our clustering algorithm with T_{sim} set to 0.5, which we empirically found gave the best results. We used a maximum time difference (T_{time}) of 6 hours, which was picked specifically to allow for a reasonably lag between events, whilst still giving a reasonable guarantee that the events generated will fit our definition of event.

Categories were assigned to events based on the combined categories defined in Table 1. For events where multiple categories were given, the category with the highest frequency was used. In cases where there was a tie between the categories, an author gave the deciding vote.

4. RESULTS & DISCUSSION

In this section, we describe the results of our evaluation. We begin by describing characteristics of the corpus, including the number of events, their distribution across different categories, and the event coverage. We then discuss the annotator agreement, and discuss possible reasons for differences in agreement between the two approaches. Finally, we examine the quality of our event clustering approach.

4.1 Corpus Characteristics

Combined Categories	Clus.	LSH	CS	Wiki
Armed Conflicts & Attacks	98	3	1	95
Arts, Culture & Entertainment	53	26	3	34
Business & Economy	23	2	1	22
Disasters & Accidents	29	16	4	23
Law, Politics & Scandals	140	124	12	128
Miscellaneous	21	26	6	3
Science & Technology	16	10	2	11
Sports	126	175	24	26
Total	**506**	**382**	**53**	**342**

Table 2: The distribution of events across the 8 different categories, broken down by method used. The LSH, CS and Wiki columns show numbers of events *before* clustering, while the Clus. column shows the number of events *after* clustering has been performed.

After clustering, 506 *top-level events* (i.e. events created by combining events from different sources) were produced. In total, 367 events were clustered to create 77 top-level events, and a further 429 top-level events were created using individual events.

The detection approach seems to closely reflect to the types of event most commonly discussed on Twitter [22], while the Wikipedia approach gives a more realistic representation of real-world events. Both approaches produced very different distributions of results, which seem to complement each other well, giving better coverage across each of the categories. For example, the detection approaches contribute a large number of *Sports* events, whereas the Wikipedia approach contributes only a few. In contrast, the Wikipedia approach contributes a large number of events about *"Armed Conflicts and Attacks"* and *"Law, Politics and Scandals"*, where the detection approaches do not contribute many. Table 2 shows the events broken down by category and the type of approach used to generate the event.

Approach	Explicit	Implicit	Total
LSH	4,009	93,398	97,407
CS	465	15,098	15,563
Wikipedia	39,980	0	39,980
Total	**44,454**	**108,496**	**152,950**

Table 3: The distribution of relevance judgments across the different approaches.

The detection method contributed to 186 top-level events, while the Wikipedia approach contributed to 342, almost double that of the detection method. However, the detection approaches contribute over 110,000 of the 150,000 relevance judgments in the corpus, with an average of 259 tweets per event cluster. The Wikipedia approach contributes just under 40,000 of the relevance judgments, at an average of only 110 tweets per event. This difference could be for a number of reasons, such as different volumes of discussion for different types of event, or due to the removal of events with less than 30 tweers from the detection approaches. However, combination of the two approaches allows their different characteristics to complement each other, producing a much more robust corpus than would have been produced had a single approach been used.

4.2 Annotator Agreement

The choice of 5 annotators for the evaluation of the detection approaches was useful for a number of reasons. Firstly, event agreement increases significantly when 5 annotators are used as opposed to 3 (0.91 and 0.82 respectively using Free-marginal multirater kappa [16]). Secondly, it guarantees that at least 3 annotators will judge each tweet. However, tweet agreement remains almost unaffected between 5 and 3 annotators (0.91 and 0.90 respectively). This suggests that 3 annotators would have been enough for the evaluation of detection approaches, however would have resulted in many cases where fewer than 3 annotators judged the tweets for a candidate event.

In the case of the Wikipedia approach, tweet agreement was significantly lower at 0.72, although this still shows very strong agreement between annotators. We hypothesize the slight drop in agreement is because low-quality and lazy workers simply skipped judging tweets for the detection approaches by answering "no" to the first question. However, this is not possible for the Wikipedia approach, meaning that low-quality workers performing evaluations have a detrimental effect on annotator agreement.

Annotator agreement was substantial across the TDT categories (0.76). However, agreement was further improved when the combined categories were used, giving near-perfect agreement (0.81). This shows that our combined categories not only helped to create a category mapping between the different approaches, but also helped to improve agreement, and thus the categorization of events.

4.3 Event Clustering

Empirical evaluation of the clusters generated show that the vast majority of clusters have been created very precisely, and there appears to be very few false matches. Furthermore, there are only 16 cases when events from the Wikipedia approach were clustered together, which is very promising as the granularity of the Wikipedia Current Events Portal is very similar to the level of granularity which we hoped to achieve.

There appear to be a small number of missed-matches for events from the detection approaches. This was to be expected as the clusters had already been shown to be dissimilar by the detection approach which created it. This suggests that our clustering approach has some room for improvement, particularity when clustering events from the same detection approach.

It is interesting to note that, although we could have used the number of shared tweets as a feature for clustering (since shared tweets suggest discussion of a shared event), it would have made no difference to the resulting clusters. Out of 41 cases where events share more than 10 tweets, there is only a single case where they do not have a similarity score above our threshold, and the events are subsequently clustered through shared similarity to a 3rd event. This helps to demonstrate the effectiveness and robustness of our event clustering technique.

5. CONCLUSION

In this paper, we examined the creation of a new corpus for evaluation of event detection systems on Twitter. We examined current definitions of event, and proposed a new definition which better fits the characteristics of Twitter. We proposed a methodology for the creation of a large-scale event detection corpora using state-of-the-art event detection approaches, and the Wikipedia Current Events Portal to create a pool of events. We then use crowdsourcing to generate relevance judgments for the pool of events. We then propose a method of merging events from different sources, so that the final events fit our definition of event. We discuss the quality of the results obtained, and note a number of areas which merit further investigation. We make the corpus, which contains 120 million Twitter, and relevance judgments for 150,000 tweets covering more than 500 events, available for further research and development.

5.1 Future Work

As noted in Section 4, we do not believe that the event clustering is perfect, and it merits further investigation to evaluate the effectivenesses of the clustering approach used. Additionally, with the TREC 2013 Microblog corpus soon to be released, it would be interesting to apply our methodology to their corpus.

6. ACKNOWLEDGEMENT

This work was supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 288024 (LiMoSINe project).

7. REFERENCES

[1] TDT 2004: Annotation manual, 2004.

[2] C. C. Aggarwal and K. Subbian. Event detection in social streams. In *SDM'12*, pages 624–635, 2012.

[3] J. Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer, 2002.

[4] G. Amati, G. Amodeo, M. Bianchi, A. Celi, C. Nicola, M. Flammini, C. Gaibisso, G. Gambosi, and G. Marcone. Fub, iasi-cnr, univaq at trec 2011. In *TREC 2011*. 2011.

[5] H. Becker, D. Iter, M. Naaman, and L. Gravano. Identifying content for planned events across social media sites. In *WSDM'12*, pages 533–542, 2012.

[6] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. In *ICWSM'11*, pages 438–441, 2011.

[7] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *CEAS'10*, volume 6, 2010.

[8] D. Boyd, S. Golder, and G. Lotan. Tweet, tweet, retweet: Conversational aspects of retweeting on twitter. In *HICSS'10*, pages 1–10, 2010.

[9] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking news on twitter. In *CHI'12*, pages 2751–2754, 2012.

[10] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang. Tedas: A twitter-based event detection and analysis system. *Data Engineering, International Conference on*, 0:1273–1276, 2012.

[11] R. McCreadie, I. Soboroff, J. Lin, C. Macdonald, I. Ounis, and D. McCullough. On building a reusable twitter corpus. In *SIGIR'12*, pages 1113–1114, 2012.

[12] I. Ounis, C. Macdonald, J. Lin, and I. Soboroff. Overview of the trec-2011 microblog track. In *TREC 2011*, 2011.

[13] S. Petrović, M. Osborne, and V. Lavrenko. Streaming first story detection with application to twitter. In *HLT'10*, pages 181–189, 2010.

[14] S. Petrović, M. Osborne, and V. Lavrenko. Using paraphrases for improving first story detection in news and twitter. In *NAACL HLT'12*, pages 338–346, 2012.

[15] F. Pukelsheim. The three sigma rule. *The American Statistician*, 48(2):pp. 88–91, 1994.

[16] J. J. Randolph. Free-Marginal Multirater Kappa (multirater K[free]): An Alternative to Fleiss' Fixed-Marginal Multirater Kappa. In *Joensuu University Learning and Instruction Symposium*, 2005.

[17] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW'10*, pages 851–860, 2010.

[18] T. H. Van Duc, T. Demeester, J. Deleu, P. Demeester, and C. Develder. Ugent participation in the microblog track 2012. In *TREC 2012*, 2012.

[19] J. Weng and B.-S. Lee. Event detection in twitter. In *ICWSM'11*, 2011.

[20] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *SIGIR'98*, pages 28–36, 1998.

[21] Q. Zhao, P. Mitra, and B. Chen. Temporal and information flow based event detection from social text streams. In *AAAI'07*, pages 1501–1506, 2007.

[22] X. Zhao and J. Jiang. An empirical comparison of topics in twitter and traditional media. *Singapore Management University School of Information Systems Technical paper series.*, 10:2011, 2011.