Sub-story detection in Twitter with hierarchical Dirichlet processes

P.K. Srijith[a,*], Mark Hepple[a], Kalina Bontcheva[a], Daniel Preotiuc-Pietro[b]

[a] *Department of Computer Science, The University of Sheffield, UK*
[b] *Computer & Information Science, University of Pennsylvania, USA*

ABSTRACT

Social media has now become the de facto information source on real world events. The challenge, however, due to the high volume and velocity nature of social media streams, is in how to follow all posts pertaining to a given event over time – a task referred to as story detection. Moreover, there are often several different stories pertaining to a given event, which we refer to as *sub-stories* and the corresponding task of their automatic detection – as *sub-story detection*. This paper proposes hierarchical Dirichlet processes (HDP), a probabilistic topic model, as an effective method for automatic sub-story detection. HDP can learn sub-topics associated with sub-stories which enables it to handle subtle variations in sub-stories. It is compared with state-of-the-art story detection approaches based on locality sensitive hashing and spectral clustering. We demonstrate the superior performance of HDP for sub-story detection on real world Twitter data sets using various evaluation measures. The ability of HDP to learn sub-topics helps it to recall the sub-stories with high precision. This has resulted in an improvement of up to 60% in the F-score performance of HDP based sub-story detection approach compared to standard story detection approaches. A similar performance improvement is also seen using an information theoretic evaluation measure proposed for the sub-story detection task. Another contribution of this paper is in demonstrating that considering the conversational structures within the Twitter stream can bring up to 200% improvement in sub-story detection performance.

© 2016 Published by Elsevier Ltd.

1. Introduction

Online social networks play a major role in generating and disseminating information. They provide a platform for people to voice their opinion and viewpoints. Moreover, social media provide main stream media, governments, companies, and citizens the opportunity to obtain real time information about events happening around the world. The challenge, however, due to the high volume and velocity nature of social media streams, is in how to follow all posts pertaining to a given event over time – a task referred to as *story detection* Petrović, Osborne, and Lavrenko (2010).

Story detection is a specific form of Topic Detection and Tracking (TDT) (Allan, 2002), which is concerned with discovering posts pertaining to real-world stories as they unfold over time. This paper, in particular, is concerned with tracking event-related posts within the micro-blogging social network, Twitter.

* Corresponding author.
E-mail addresses: pk.srijith@sheffield.ac.uk, pk.srijith@gmail.com (P.K. Srijith), m.r.hepple@sheffield.ac.uk (M. Hepple), k.bontcheva@sheffield.ac.uk (K. Bontcheva), danielpr@sas.upenn.edu (D. Preotiuc-Pietro).

When considering posts related to a given real-world event (e.g. the Ferguson unrest), researchers (e.g. Procter, Crump, Karstedt, Voss, and Cantijoch (2013); Zubiaga, Liakata, Procter, Wong Sak Hoi, and Tolmie (2016)) have found that there are often different *sub-stories* pertaining that event (e.g. 'Ferguson police shot Brown', 'Ferguson police confiscated all video evidence'). In this paper we address the question how such substories can be identified automatically. This is particularly beneficial for mainstream media, governments, law enforcing agencies, companies, and other organisations who are increasingly faced with the challenge of tracking and responding to emerging sub-stories, rumours, and misinformation.

Automatic sub-story detection is a harder task than story detection as sub-stories on the same event have a lot of vocabulary in common. For instance, all the posts related to the Ferguson unrest share words such as 'Ferguson', 'police', 'Brown'. Sub-stories tend to overlap considerably in time and are often associated with low tweet rates. Also, they are not necessarily true (e.g. different rumours emerged simultaneously during the England riots (Procter et al., 2013)) and thus may not have a corresponding sub-story in the real world. Standard story detection (also known as event detection) approaches fail to take these into account and do not perform well on sub-story detection task.

This paper investigates a sub-story detection approach based on hierarchical Dirichlet processes (HDP). This hierarchical topic model can detect sub-topics associated with a main topic and is thus well suited to the sub-story detection task. For instance, in the Ferguson unrest example, the approach can detect 'Ferguson police' as the main topic, while 'shot Brown' and 'video evidence' as different sub-topics. The effectiveness of the proposed approach is established by comparing it with state-of-the-art story detection approaches based on locality sensitive hashing (LSH) Petrović et al. (2010) and spectral clustering (SC) Preotiuc-Pietro et al. (2013) (see Section 6 for details).

Sub-story detection in Twitter streams is particularly challenging, since reply tweets often do not share sufficient vocabulary with the tweets they reply to. For instance, in one of the tweet data set used for experiments here (Zubiaga et al., 2016), there is a sub-story claiming that Fox News is not covering the Ferguson protests. In particular, one of the earlier tweets has text 'Currently the #FoxNews website has zero, repeat, ZERO coverage of the #Ferguson protests'. A subsequent reply tweet, however, states "Too busy bitching about POTUS for sure", i.e. does not mention neither Ferguson, nor Fox News explicitly. Since many reply tweets exhibit similar lack of linguistic overlap with the originating tweet, this makes it difficult to cluster them as pertaining to the same sub-story cluster. To address this problem, our approach takes into account the conversational structure, in addition to linguistic features, in order to improve sub-story detection performance.

Sub-story detection performance is evaluated on real world Twitter data sets. Twitter is ideal for this task because it provides a large source of publicly available posts on major events, e.g. the 2014 Ferguson unrest and the 2011 London riots. In particular, we experiment with five different Twitter data sets. Three of these arise from rumour detection research (Procter et al., 2013; Zubiaga et al., 2016) and come annotated with events (2011 London riots, the 2014 Ferguson unrest, and the 2014 Ottawa shooting) and tweets grouped by sub-story within each of the events. The fourth public data set was created for first story detection (FSD) with Locality Sensitive Hashing (LSH), which is one of our baselines (Petrović et al., 2010). The fifth public data set (FAcup (Aiello et al., 2013)) contains tweets about various events occurring during a football match. The properties of the data sets are discussed further in Section 5 and comparative evaluation results for all methods and data sets are detailed in Section 7.

We propose to use a mutual information based evaluation measure, adjusted mutual information (AMI) (Vinh, Epps, & Bailey, 2009), in addition to the standard precision-recall metrics which avoids agreements by chance by considering the number of clusters produced. The experimental results establish the superior performance of HDP for sub-story detection task. The approach could detect sub-story specific topics which helps journalists and government agencies to monitor the evolution of new topics associated with a story. The runtime performance of HDP is comparable to established story detection approaches and can be used to perform real-time detection of sub-stories.

The main contributions of the paper are summarized as follows:

- Introduces the sub-story detection task and proposes a hierarchical Dirichlet process based approach to solve the problem of sub-story detection.
- Provides rigorous experimental comparison of the proposed sub-story detection approach with state-of-the-art story detection approaches, establishing the effectiveness of HDP for sub-story detection.
- Proposes a mutual information based metric for evaluating the performance of sub-story detection approaches.
- Demonstrates the usefulness of conversational structure in improving sub-story detection performance.

2. Related work

A number of techniques have been employed for detecting and tracking stories in social media streams (Allan, 2002). Story detection is typically done by extending traditional clustering algorithms to a streaming data setting (Aggarwal, 2014). A comprehensive survey of the literature on story detection techniques in Twitter data is given in (Farzindar & Wael, 2015).

Story detection in Twitter for a particular topic such as 'earthquake' is studied in (Sakaki, Okazaki, & Matsuo, 2010). Becker et al. (2011) use an online clustering algorithm to detect stories and distinguish real vs. non-real stories using a classification method. Twevent (Li, Sun, & Datta, 2012) is a story detection approach which clusters bursty segments in Twitter data. A fast and efficient approach based on locality sensitive hashing (LSH) is first used in (Petrović et al., 2010) to detect the emergence of new stories (first story detection) in Twitter. Locality sensitive hashing reduced the computational complexity associated with nearest neighbour search and detected clusters of documents in constant space and time. Later,

they extended this approach to counter lexical variations in documents by using paraphrases (Petrović, Osborne, & Lavrenko, 2012). An alternative approach to detect new events by storing the contents of already seen documents in a single hash table is proposed in (Wurzer, Lavrenko, & Osborne, 2015b). Further, LSH based techniques are also developed to handle topic streams emerging in Twitter (Wurzer, Lavrenko, & Osborne, 2015a). Here, topics are also hashed into a bucket in addition to the tweets. A multi-view clustering algorithm is proposed in (Fang, Zhang, Ye, & Li, 2014) to detect hot topics in Twitter by considering semantic relations, social tag relations and temporal relations. A multi-layer event detection algorithm is proposed in (Abramson et al., 2014) to detect global and local events occurring in Twitter. An ontology based semantic clustering approach, where hash-tags are mapped to Wordnet terms, is used in (Vicient & Moreno, 2015) to discover topics in Twitter. Lately, a formal concept based approach is introduced in (Cigarran, Castellanos, & Garca-Serrano, 2016) to perform topic detection in Twitter by considering the relationship between tweet terms and tweets.

Topic models are also used to detect stories in Twitter, for instance latent Dirichlet allocation (LDA) (Diao, Jiang, Zhu, & Lim, 2012) to detect bursty topics by considering the user and time at which a post is published. Twitter-LDA (Zhao et al., 2011) models a tweet to consist of a single topic and a background topic. Later, this was extended to an online setting by Sasaki, Yoshikawa, and Furuhashi (2014). A non-parametric topic model based on Dirichlet process is used in (Wang, Zhu, Jiang, & Li, 2013) to detect newsworthy stories in Twitter, where topics are shared among tweets from consecutive time periods. Topicsketch (Xie, Zhu, Jiang, Lim, & Wang, 2013) uses a novel sketch based topic model to detect bursty topics from millions of tweets. Latent Dirichlet allocation is combined with Concepnet in (Vo, Hai, & Ock, 2015) to learn distinguishing terms among tweets from different events in Twitter.

A spectral clustering based approach is developed in (Preotiuc-Pietro, Srijith, Hepple, & Cohn, 2016) to address the task of story detection. The approach uses a mutual information based metric to represent the similarity matrix in spectral clustering. Real world events happen at different scales of time and space. Multi-scale event detection (Dong, Mavroeidis, Calabrese, & Frossard, 2015) aims to detect stories evolving at different pace and spanning different geographic locations by using the properties of wavelet transform. Supervised machine learning techniques such as support vector machines and logistic regression are used to detect events corresponding to specific topics such as those related to traffic (D'Andrea, Ducange, Lazzerini, & Marcelloni, 2015), lifestyle and wellness (Akbari, Hu, Nie, & Chua, 2016) and uprisings (Boecking, Hall, & Schneider, 2015). These event detection techniques will not be able to distinguish different sub-stories associated with a main story due to content overlap.

Whilst story detection has received considerable attention, less attention has been paid to sub-story detection task. Aiello et al. (2013) discuss tasks similar to sub-story detection like finding important events happening over time in a main event such as a football match. They used standard story detection approaches to find events on their tasks. There exists approaches (Nichols, Mahmud, & Drews, 2012; Zubiaga, Spina, Amigó, & Gonzalo, 2012) which rely on tweet rates in an interval to detect major moments in a game. Chakrabarti and Punera (2011) use a modified hidden Markov model combining tweet rate and text features to summarize events in a game. Shen et al. (2013) use a time-content mixture model which effectively combines burstiness and cohesiveness to detect key moments in a story. Chierichetti, Kleinberg, Kumar, Mahdian, and Pandey (2014) use non-textual features based on tweet rate and communication pattern among users to detect points in time where an important event happens within a story. An approach based on graph-of-words to represent sequence of tweets was used in (Polykarpos, Giannis, Francois, Yannis, & Michalis, 2015) to detect important events happening within a football match. Recently, a mutually generative LDA (Xing, Wang, Liu, Huang, & Ma, 2016) is proposed for the same task by considering the effect of hash-tags in Twitter. These approaches will not be effective for detecting sub-stories which overlaps considerably in time and have low tweet rates.

There exists a hitherto unaddressed problem – finding sub-stories related to a particular real world event. These sub-stories may or may not correspond to real-world events (e.g. false rumours about the London riots (Procter et al., 2013) do not), they tend to overlap in time (i.e. tweets on more than one sub-story circulate simultaneously), and share significant common vocabulary (Zubiaga et al., 2016). As demonstrated in the rest of this paper, state-of-the-art approaches for story detection do not perform well on this type of task.

3. Research objective

The following are our main research objectives:

1. Introduce the task of sub-story detection in Twitter. Sub-story detection differs from story detection and we discuss the properties specific to sub-story detection which makes it a harder task than story detection.
2. Propose hierarchical Dirichlet processes as an effective approach for sub-story detection. Unlike story detection approaches, HDP can learn sub-topics associated with sub-stories which makes it particularly useful for modelling sub-story detection task.
3. Verify experimentally the effectiveness of HDP for sub-story detection task. We compare HDP with standard story detection approaches based on locality sensitive hashing and spectral clustering on real world Twitter data sets to establish the fitness of HDP for sub-story detection.
4. Show the usefulness of conversational structure in Twitter for improving the performance of sub-story detection task. By considering conversational structure, reply tweets which does not share a topical similarity with the source tweets gets clustered along with the source tweet.

Table 1
Description of major stories in the Ferguson data set.

Sub-story id	Description	# Tweets	# Source
1	M. Brown was involved in a robbery before being shot	2312	89
2	Ferguson police are leading a smear campaign or character assassination of M. Brown	623	18
3	Initial contact between police officer and M. Brown was not related to the robbery	553	28
4	Ferguson police to release name of police officer who shot M. Brown today	536	26
5	Ferguson police are lying about the circumstances leading up to M. Brown's death	426	23
6	Ferguson police once beat up a man and charged him for bleeding on their uniforms	369	24
7	M. Brown was stopped by police for walking in the middle of the street	236	10
8	Fox News is not covering the Ferguson protests	102	2
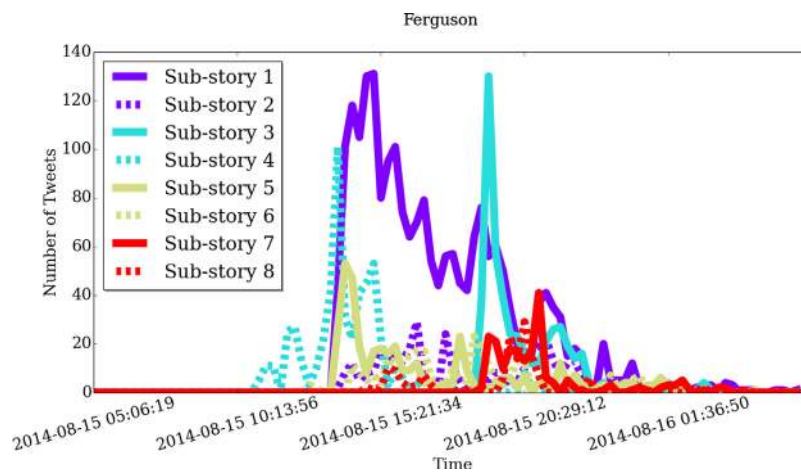
Fig. 1. Temporal profile of sub-stories in the Ferguson data set.

5. Introduces adjusted mutual information score as an effective metric to measure clustering performance in sub-story detection. Standard metrics based on precision typically favour clustering approaches which produces large number of small sized clusters. Such clustering approaches are not useful in practice and we propose to use AMI as an effective alternative metric which can take care of such problems.

4. Problem definition: sub-story detection

This paper addresses the problem of detecting *sub-stories* associated with a main story (collection of tweets related to a real world event) as they emerge in social media streams. The scope of the story could be any event of interest to the user (for e.g. Olympics or a sports category in Olympics). Sub-story detection separates tweets pertaining to the same real world-event into clusters related to different sub-stories. Sub-story detection poses additional challenges over story detection: the sub-stories have an overlapping vocabulary and tweet rates are comparatively lower.

Table 1 shows 8 major sub-stories related to the Ferguson unrest from one of our five data sets. All these sub-stories are related to the shooting of M. Brown by the Ferguson police and thus share words such as 'M. Brown', 'Ferguson', 'police' etc. Standard story detection approaches fail to produce good results in this setting where vocabulary is shared across the sub-stories because they look at tweet similarity or overlapping words to cluster tweets.

Sub-stories can have considerably larger lifespan, overlap in time, and are set within a broader over-arching story, that contains many thematically related sub-stories. The themes discussed in these sub-stories are referred to as *sub-topics*. For example, consider the temporal profile of sub-stories from the Ferguson data shown in Fig. 1. We can observe that these sub-stories overlap in time and have relatively low tweet rate. This is mainly due to the fact that there are multiple conversations within a sub-story, each evolving at a different point in time. A sub-story which becomes active at some point in time can become dormant temporarily and then re-activate again at a later time.

5. Data set description

The core of our experiments are carried out on three sub-story annotated data sets (Ferguson unrest, Ottawa shooting and London riots). The first two are very recent and have been collected and human annotated as part of a rumour analysis project (Zubiaga et al., 2016), while the London riots one arose from an earlier qualitative social science analysis of related tweets (Procter et al., 2013). All three data sets consist of tweets, grouped together into human annotated sub-stories related to the particular real-world event. Other tweets pertaining to the same event count as background data.

Table 2
Description of sub-stories for London riots data set.

Sub-story id	Description	# Tweets
1	Army deployed in Bank	192
2	Rioters attack children's hospital	1666
3	London Eye set on fire	657
4	Rioters cook food in McDonalds	218
5	Miss Selfridge set on fire	5581
6	Police beat girl	902
7	Rioters attack London zoo	937

We also consider two other publicly available data sets, FSD and FAcup, which have been created for story detection. Even though these data sets are not strictly suitable to the sub-story detection task, we use them for comparative purposes and have vocabulary overlap across some sub-stories.

Next we describe these data sets in more detail.

5.1. Ferguson

This data set consists of tweets collected between August and September 2014, all related to the unrest that took place in Ferguson, USA (Zubiaga et al., 2016). This data set is not only sub-story annotated, but also includes "reply-to" information, which connects together subsets of tweets into conversational graphs. We refer to *reply tweets* as those that reply to a tweet present in the data set, and *source tweets* as all those which do not have such a "parent". In other words, the data set is regarded as a collection of conversational threads, each of which has a single source tweet at its root.

As detailed in (Zubiaga et al., 2016), journalists categorised manually source tweets as belonging to one of 45 different sub-stories. A reply tweet is assumed automatically to belong to the sub-story to which its source tweet has been assigned (if any).

After discounting sub-stories with fewer than 10 tweets, the final data set consists of 6598 labelled tweets spread across 35 sub-stories and 18,650 tweets as background, i.e. not belonging to any of those sub-stories. Considering source tweets alone, there are 284 labelled source tweets and 899 background source tweets. Table 1 lists major sub-stories in the data (Zubiaga et al., 2016), illustrating how the sub-stories are very similar and have the Ferguson unrest as a common topic.

5.2. Ottawa

The Ottawa data set consists of tweets related to shootings at the parliament building in Ottawa during October 2014 (Zubiaga et al., 2016). Similar to the Ferguson data, it also has a conversational structure including source and reply tweets. The data set consists of 6414 tweets spread across 39 sub-stories and 5975 tweets as background.

Considering source tweets alone, there are 462 labelled tweets and 439 background tweets. Some major sub-stories associated with the Ottawa shooting event are 'Soldier shot dead is Nathan Cirillo', 'Soldier shot at War Memorial has died', 'Suspected shooter is dead', etc. All these sub-stories have a common theme of shooting and death which makes them an ideal candidate for the sub-story detection task. With respect to temporal patterns, the evolution of sub-stories is very similar to the patterns observed in the Ferguson data.

5.3. London riots

The London riots data set consists of 2.5 million tweets related to the riots that took place in London during August, 2011 (Procter et al., 2013). It includes 10,000 tweets that are labelled as belonging to 7 different sub-stories, all with a common background topic – the London riots. Table 2 provides a summary of number of tweets in each sub-story in this data set. Unlike Ferguson and Ottawa, the conversational structure was not made available by the researchers.

5.4. First story detection

This is a publicly available story detection data set (Petrović et al., 2012) with approximately 2400 tweets labelled as belonging to 27 stories, from the period June to September 2011. This is augmented with background tweets from the same period, to create a corpus of approximately 80,000 tweets. Originally this data set was created by Petrović et al. (2012) for evaluation of their first story detection (FSD) system. This FSD data set can be seen to represent the standard story detection task, in contrast to the sub-story task represented by the former three data sets. It should be noted, however, that there is some overlap of stories here as well, e.g. four of the stories are related to the London riots in 2011 and another four are about death of some celebrities. These commonalities make this data also applicable to sub-story detection, as well as enabling us to benchmark our methods on the related story detection problem.

5.5. FAcup

This is a publicly available data set (Aiello et al., 2013) with approximately 7000 tweets belonging to 13 different sub-stories associated with a football match story. These tweets represents sub-stories such as goals, fouls etc. in the 2012 Football Association (FA) final match between Chelsea and Liverpool. This data set is augmented with approximately 20,000 tweets related to the same football match as background. Due to the shared common story (football game), sub-story tweets in this data set share a common vocabulary and is useful for evaluating the proposed sub-story detection approach. However, they differ from sub-story data sets such as Ferguson and Ottawa in that the sub-stories in this data set are temporally separated.

6. Methods

The main sub-story detection method investigated in this paper uses hierarchical topic modelling. In particular, we experiment with hierarchical Dirichlet processes (HDP), a non-parametric Bayesian model, which can effectively model the sub-story detection task. HDP is also compared to two story detection state-of-the-art approaches: spectral clustering and locality sensitive hashing.

6.1. Hierarchical dirichlet process

Latent Dirichlet allocation (LDA) (Blei, Ng, & Jordan, 2003) and hierarchical Dirichlet processes (HDP) (Teh, Jordan, Beal, & Blei, 2006) have shown promising results in topic modelling due to their probabilistic interpretations. They model a document (i.e. tweet in our case) as a mixture of topics, where each topic has a distinct distribution over the words. These generative models can infer the latent topics associated with the tweets.

In this paper, we propose to use HDP for sub-story detection, since it can model the hierarchical structure underlying the topic distribution. As argued above, in sub-story detection we need to find sub-topics associated with the main story (e.g. the Ferguson unrest), and HDP is developed specifically to handle such kinds of tasks. HDP achieves this by extending the Dirichlet process mixture model (DPMM) Murphy (2012) to a hierarchical setting.

In more detail, the DPMM considers a tweet as consisting of words generated by a mixture of topics. The mixture distribution is modelled using a non-parametric prior based on a Dirichlet process (DP) (Hjort, Holmes, Mller, & Walker, 2010). A DP is parameterized by a base distribution H and a concentration parameter α and is denoted as $DP(\alpha, H)$. The base measure specifies the a-priori distribution over some parameter space $\boldsymbol{\theta}$ which is used to generate observed data.

In our case, $\boldsymbol{\theta}$ represents the parameters of a multinomial distribution over the words $\boldsymbol{w}$ in a tweet. A draw from $DP(\alpha, H)$ is a discrete probability measure G providing a distribution over $\boldsymbol{\theta}$. It can be represented as $G(\boldsymbol{\theta}) = \sum_{i=1}^{\infty} \pi_i \delta_{\theta_i}(\boldsymbol{\theta})$, where δ_{θ_i} is the Kronecker delta function which gives a value of 1 when the parameter takes value θ_i, θ_i is a draw from H and π_i is the probability mass associated with θ_i. The sequence of values π_i is obtained from α using a stick breaking process (Sethuraman, 1994)

$$\pi_i = \bar{\pi}_i \prod_{l=1}^{i-1} (1 - \bar{\pi}_l) \qquad \bar{\pi}_i \sim Beta(1, \alpha). \tag{1}$$

The process ensures that $\boldsymbol{\pi}$ represents a probability distribution *i.e.* $\sum_{i=1}^{\infty} \pi_i = 1$ and is often represented as $\boldsymbol{\pi} \sim GEM(\alpha)$. The concentration parameter α determines the probability mass associated with a topic π_i as a parameter in Beta distribution. A draw from G results in θ_i with probability π_i, with θ_i representing the parameters of a multinomial distribution associated with a topic i. Thus each topic i occurs in a tweet with probability π_i. Modelling tweets independently as a DPMM does not allow topics to be shared across tweets, which is needed in our task.

Hierarchical Dirichlet processes are developed to handle grouped data and share topics across the groups (Teh et al., 2006). We use them to model the tweet as consisting of a set of topics and to share topics across multiple tweets. HDP achieves this by drawing tweet specific probability distribution G_d for a tweet d from $DP(\gamma, G_0)$, where γ represents the concentration parameter and G_0 is the base distribution shared by all the tweets. The common base distribution G_0 is indeed a draw from $DP(\alpha, H)$. The common base distribution has the form $G_0(\boldsymbol{\theta}) = \sum_{i=1}^{\infty} \pi_{0i} \delta_{\theta_i}(\boldsymbol{\theta})$ and the tweet specific distribution has the form $G_d(\boldsymbol{\theta}) = \sum_{i=1}^{\infty} \pi_{di} \delta_{\theta_i}(\boldsymbol{\theta})$. Here, both the common base distribution and tweet specific distributions share the parameters θ_i (or the topics) with tweet specific mixture distribution $\boldsymbol{\pi}_d$ over the topics. Thus, the tweets modelled using HDP share the topics but with different probabilities. The mixing proportions $\boldsymbol{\pi}_d$ is generated as follows (Teh et al., 2006)

$$\boldsymbol{\pi_d} \sim DP(\gamma, \boldsymbol{\pi_0}) \qquad \boldsymbol{\pi_0} \sim GEM(\alpha). \tag{2}$$

Fig. 2 shows the graphical model representation of the HDP model. A word w_{dn} in a tweet d comes from a topic with parameter θ_{dn} drawn from the Dirichlet Process G_d associated with the tweet. The topics are shared across the tweets due to the hierarchical modelling of DPMM.

Since in sub-story detection tweets relate to the same real world event, HDP can model this effectively, coupled with the fact that individual tweets address different sub-topics (corresponding to the sub-stories). These sub-topics are characterized by words and each word is associated with a probability indicating the importance of the word in representing the sub-topic.
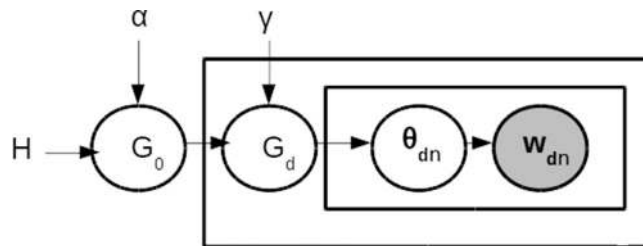
Fig. 2. Graphical model for HDP.

The identified sub-topics are used to cluster tweets based on the words common in the tweet and the sub-topics. For each tweet, we detect common words and calculate a similarity score to a sub-topic by summing the probability associated with these words in representing the sub-topic. The tweet is assigned to the sub-topic with the maximum similarity score. We use sub-topics for clustering the tweets as they can better discriminate the tweets associated with sub-stories.

6.2. Spectral clustering

Clustering techniques have been used widely to detect stories in social data streams (Aggarwal, 2014). Here, we discuss one based on spectral clustering using point-wise mutual information (Preotiuc-Pietro et al., 2013). The spectral clustering (SC) algorithm (Shi & Malik, 2000) has been shown to achieve the state-of-the-art performance for tasks such as community detection in graphs (White & Smyth, 2005). This method treats clustering as a graph partitioning problem. It projects the objects into a lower dimensional space by performing singular value decomposition of a similarity graph constructed over the objects. It then discovers clusters of objects which are maximally separated in this space using standard clustering techniques, such as k-means. A good spectral clustering relies on a good similarity graph which best reflects the connections between objects.

We apply spectral clustering to detect sub-stories in a stream of tweets (Preotiuc-Pietro et al., 2013). The approach represents a similarity graph by constructing a matrix which captures the similarity over words that appear in the data. It uses normalized point-wise mutual information (NPMI) (Bouma, 2009) to capture the word similarity. NPMI measures the probability of co-occurrence of words in the same tweet. The idea is that if two words appear consistently in the same tweet, then they are indicative of the same story. For example, the co-occurrence of words such as "Ferguson" and "police" over a period of time indicate there is a story related to Ferguson police.

The NPMI measure between words pairs x and y is calculated as

$$NPMI(x, y) = -\log p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{3}$$

where $p(x)$ denotes the probability of occurrence of a word x in a tweet, and $p(x, y)$ provides the probability of co-occurrence of words x and y in a tweet. We consider two words as co-occurring if they appear in the same tweet, which gives us a straightforward measure of co-occurrence frequencies. The NPMI measure takes values between -1 and 1 with positive values indicating a higher chance of co-occurrence and negative value indicating a lower chance of co-occurrence.

The spectral clustering algorithm proceeds by filtering out less frequent words and constructing a similarity graph over words using the NPMI measure. It ignores all NPMI values less than a threshold and keeps the largest connected component from the resulting graph. Singular value decomposition is performed over a graph Laplacian constructed from this similarity graph to obtain a representation of words in a lower dimensional space. A k-means algorithm then finds clusters of words in this reduced space.

The word clusters discovered by the spectral clustering algorithm represents a coherent topic. The words are associated with a score, which provides a measure of importance of the word in representing the topic. For each tweet, a similarity score is computed with respect to each topic, using the co-occurrence score of the words in the tweet. The tweets are then clustered by assigning them to the topic with the highest similarity score. Thus, tweets in the same cluster form a topically coherent cluster.

6.3. Locality sensitive hashing

The second state-of-the-art approach is locality sensitive hashing (LSH) Rajaraman and Ullman 2011), which was proposed originally for first story detection in Twitter (Petrović et al., 2010). LSH finds nearest neighbour tweets in constant time and keeps only a constant number of tweets in memory.

The LSH approach uses the nearest neighbour algorithm to find the tweet closest to the incoming tweet. The computational overhead of finding the nearest neighbour is overcome using locality sensitive hashing. LSH maps incoming tweets to buckets using a hashing function which maps similar tweets to the same bucket. The method then finds the nearest neighbour to the incoming tweet by searching the bucket to which it has been mapped. This greatly reduces the search space.

The approach clusters tweets based on the cosine similarity of the tweets which are hashed into the same bucket. It assigns an incoming tweet to the cluster of its nearest neighbour if the cosine similarity is greater than a particular threshold. Otherwise, it assigns the tweet to a new cluster.

In more detail, LSH uses a series of random hyperplanes sampled from a normal Gaussian distribution. These hyperplanes divide the space into subspaces and similar tweets fall into the same subspace. We consider k such hyperplanes. The number of hyperplanes k can be considered as a number of bits per key in this hashing scheme. Let $\boldsymbol{u}_i, i = 1, \ldots, k$ represent the hyperplanes and $\boldsymbol{x}$ be the *tf-idf* representation of the tweet. The hash value is considered to be a binary vector with k bits. We set the bit i to be 1 if $\boldsymbol{x}.\boldsymbol{u}_i > 0$ and 0 otherwise. The tweets falling in the same subspace have the same hash value in the hash table and is stored in a bucket of size b. The higher k is, the fewer collisions there will be in the buckets, but more time will be needed to compute the hash values. However, increasing k also decreases the probability of collision with the nearest neighbour, and hence multiple hash tables (h) are required to increase the chances of finding the nearest neighbour. Thus, a tweet is compared with the tweets belonging to the same bucket in multiple hash tables in order to find its nearest neighbour using cosine similarity. The nearest neighbour tweets with cosine similarity below a user specified threshold forms a cluster. This cluster represents tweets with some topical similarity, which helps one to detect stories evolving in Twitter.

7. Experimental results

This section reports on the comparative evaluation of HDP and all state-of-the-art baselines on the sub-story detection data sets. We follow a cluster-based approach, as it accounts for the varying popularity of sub-stories and the related user endorsement aspect. This also provides a fair comparison with LSH. Alongside this, we also consider an event extraction setting where major sub-stories detected by HDP and SC are described in terms of detected topics.[1] Results are reported using the standard metrics of precision, recall, F-score and adjusted mutual information (Vinh et al., 2009). The latter is included as it has certain advantages over the others with respect to cluster evaluation.

In particular, the experiments compare Hierarchical Dirichlet processes (HDP), spectral clustering (SC) and locality sensitive hashing (LSH) on the data sets introduced in Section 5 (Ferguson, Ottawa, London riots, FSD, and FA Cup).

The text of each tweet is pre-processed to remove unusual characters, tokens, and stop words, followed by stemming. In particular, the filtered tokens are: user-mentions (tokens starting with @), hashtags (words starting with #) and URLs. The rationale behind hashtag removal is that hashtags often tend to refer to the shared real world event (e.g. #Ferguson, #Londonriots) and are thus shared across sub-stories.

7.1. Method comparison using precision, recall and F-score

Our first comparative evaluation experiment uses the standard information retrieval metrics of precision and recall (Manning, Raghavan, & Schütze, 2008). Detected tweet clusters are evaluated against the gold standard tweets in the respective sub-stories. Since the approaches are unsupervised, the number of automatically discovered clusters does not always align to the sub-stories in the gold standard. Therefore, for each sub-story, we find the automatically produced cluster with the maximum overlap, in terms of number of tweets from that sub-story. It should be noted that multiple sub-stories may get aligned to a single cluster. In this case, precision measures how many of the retrieved tweets belong to the actual sub-story, while recall measures whether the system could retrieve all known tweets associated with the aligned sub-story. Performance is reported using micro-averaged precision and recall, due to the varying sizes of each sub-story (in terms of number of contained tweets).

More formally, let N be the total number of known sub-stories in a given data set. TP_i, FP_i, and FN_i are the true positives, false positives, and false negatives associated with a sub-story i. Then, micro-averaged precision and recall are calculated as:

$$P_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FP_i} \tag{4}$$

$$R_{micro} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FN_i} \tag{5}$$

We also report micro-averaged F-score, which is the harmonic mean of micro-averaged precision and recall. Approaches with a high F-score are preferred.

$$F_{micro} = \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}} \tag{6}$$

The free HDP parameters (e.g. the concentration parameters) are learnt from the data using Gibbs sampling. We put an upper bound on the number of topics produced by HDP (k) and this allows for a fair comparison with SC. The effective

---

[1] Please note that LSH does not assign topics to sub-stories, due to the nature of the algorithm.

Table 3
Results of HDP, SC, and LSH on Ferguson and Ottawa data sets for different parameter settings. Best results appear in bold letters.

Method	Ferguson			Ottawa		
	P_{micro}	R_{micro}	F_{micro}	P_{micro}	R_{micro}	F_{micro}
HDP (k200)	0.0536	0.0889	0.0668	0.1799	0.1431	**0.1594**
HDP (k300)	0.1366	0.1057	**0.1191**	0.2182	0.1249	0.1588
SC (k400)	0.0131	**0.1622**	0.0242	0.0519	**0.1821**	0.0807
SC (k2000)	0.0422	0.0861	0.0566	0.0873	0.1244	0.1025
LSH (k12h56b10)	**0.3441**	0.0301	0.0554	**0.4797**	0.0314	0.0589
LSH (k13h71b10)	0.3430	0.0407	0.0728	0.3768	0.0285	0.0529

Table 4
Results of HDP, SC, and LSH on London riots for different parameter settings. Best results are indicated in bold letters.

Method	London riots		
	P_{micro}	R_{micro}	F_{micro}
HDP (k50)	0.4188	**0.2759**	**0.3326**
HDP (k100)	0.4194	0.2013	0.2720
SC (k50)	0.1833	0.2666	0.2172
SC (k100)	0.4522	0.2539	0.3252
LSH (k12h56b10)	**0.5948**	0.2258	0.3273
LSH (k13h71b10)	0.4976	0.2323	0.3167

number of topics could be less and is determined by the concentration parameter learnt from the data. The spectral clustering approach depends primarily on the parameter *k*, which determines the number of clusters in the data set. The approach is run by filtering out words with an NPMI threshold of 0.1 and with word frequency threshold of 10. We perform experiments with different values of *k* for HDP and SC. Lastly, the LSH approach depends on the parameters *k* (number of bits), *h* (number of hash tables), and *b* (bucket size). The experiments are conducted with different values of these parameters. We present only the results obtained with the best two parameter settings (in terms of F-score) for each of the approaches.

As can be seen in Table 3, HDP is the best performing method on the Ferguson and Ottawa sub-story data sets. In particular, HDP's F-score is significantly better than the SC and LSH F-scores. With respect to precision, LSH performs best, while SC has the best recall. The low recall of LSH, however, is due to the fact that it generates a large number of very small tweet clusters, which is also the reason for its high precision. On the other hand, SC is not able to differentiate sufficiently between similar sub-stories and groups them together in a small number of very large clusters. While this increases recall, it leads to the observed low precision. In contrast, HDP can detect subtle differences in sub-stories, thanks to the sub-topics, which are then used to cluster the tweets. This leads to improved precision for HDP over SC and an ultimately higher F-score.

Next, Table 4 reports the experimental results on the much larger London riots data set. The methods here are executed by partitioning the data set into 50 sub-partitions with approximately 50,000 tweets in each. The table shows the number of clusters per partition for the HDP and SC approaches. As above, HDP has the best recall and F-score, while LSH has the highest precision.

Nevertheless, it should be noted that precision, recall and F-score are still very low on the Ferguson and Ottawa data sets, which is due to the presence of conversational threads within the sub-story clusters. As discussed in Section 1, some of the tweets within the conversational tweets tend to discuss completely unrelated topics. For instance, even if the source tweet mentions the sub-story explicitly, often reply tweets within the thread do not have significant word overlap with the source. Consequently, these reply tweets are not deemed topically similar to the source tweet and are assigned to a completely different cluster, which negatively impacts performance.

7.1.1. Conversational structure experiments

The effect of reply tweets in lowering the performance of the system is verified by conducting clustering experiments on the Ferguson and Ottawa data sets, using source tweets alone. As can be seen in Table 5, algorithm performance improves significantly, compared to the results reported in Table 3. In some cases, the improvement in performance is by an order of magnitude. Again we observe that HDP outperforms both LSH and SC, with similar precision and recall patterns as those observed in the full data sets.

The next experiment considers the sub-story assignment of entire conversational threads. The first step is to cluster only the source tweets, while reply tweets are assigned automatically to the cluster of their corresponding source tweet. This

Table 5

Results of HDP, SC, and LSH on Ferguson and Ottawa data sets (considering source tweets alone) for different parameter settings. Best results are indicated in bold letters.

	Ferguson			Ottawa		
	P_{micro}	R_{micro}	F_{micro}	P_{micro}	R_{micro}	F_{micro}
HDP (k200)	0.2578	0.3042	0.2790	0.4615	0.4172	0.4382
HDP (k300)	0.3482	0.3688	**0.3582**	0.4212	**0.5248**	**0.4673**
SC (k400)	0.0508	**0.3691**	0.0893	0.1574	0.3055	0.2077
SC (k2000)	0.1614	0.2617	0.1996	0.1373	0.3055	0.1894
LSH (k12 h56 b10)	**0.6083**	0.2402	0.3444	0.6417	0.2603	0.3703
LSH (k13 h71 b10)	0.5591	0.2508	0.3462	**0.6975**	0.2451	0.3627

Table 6

Results of HDP, SC, and LSH on Ferguson and Ottawa data sets (considering conversational structure) for different parameter settings. Bold letters indicate the best results among HDP, SC, and LSH.

Method	Ferguson			Ottawa		
	P_{micro}	R_{micro}	F_{micro}	P_{micro}	R_{micro}	F_{micro}
HDP (k200)	0.273	0.3674	0.3132	0.4749	0.4612	0.4679
HDP (k300)	0.3822	**0.4199**	**0.4001**	0.4398	**0.5691**	**0.4968**
SC (k400)	0.0722	0.4091	0.1227	0.1786	0.3581	0.2383
SC (k2000)	0.2149	0.3034	0.2515	0.1588	0.3143	0.2109
LSH (k12 h56 b10)	**0.5589**	0.3087	0.3977	0.5428	0.3038	0.3895
LSH (k13 h71 b10)	0.5079	0.3106	0.3854	**0.7777**	0.2877	0.4200
Baseline	1.0	0.2545	0.4057	1.0	0.1696	0.2900

Table 7

Results of HDP, SC, and LSH on most prominent stories in the Ferguson and Ottawa data sets.

Method	Sub-story 1			Sub-story 2			Sub-story 3		
	P	R	F	P	R	F	P	R	F
Ferguson									
HDP (k300)	0.96	**0.26**	**0.41**	0.45	**0.40**	**0.42**	**0.59**	0.74	0.66
SC (k400)	0.32	**0.26**	0.28	0.16	0.19	0.17	0.23	0.29	0.26
LSH (k12h56b10)	**0.99**	0.16	0.28	1	0.17	0.29	0.31	0.26	0.28
Ottawa									
HDP (k300)	**0.99**	0.65	0.78	0.82	**0.66**	**0.73**	0.46	0.47	0.46
SC (k400)	0.95	0.27	0.42	0.15	0.31	0.20	0.11	0.28	0.16
LSH (k13h71b10)	**0.99**	0.44	0.61	1	0.15	0.26	0.4	0.16	0.23

is a realistic setting on these data sets, and on unseen Twitter data in general, since the source-reply structure is readily available.

Table 6 shows that considering conversational threads achieves an order of magnitude improvement in recall and F-score, as compared to those in Table 3. Again, we observe that LSH has better precision, while HDP has better recall, and ultimately HDP has the best F-score.

This experiment also considered an additional baseline, which clusters tweets using only the conversational structure. Here, we consider each conversation as a cluster and the largest conversation as the representative of a substory. The aim here is to investigate whether the sub-story detection approaches can get a better recall, than this readily available baseline. By design, this approach has a precision of 1. This simple baseline has a recall of 0.2545 and 0.1696 for Ferguson and Ottawa respectively, which is lower than the recall of the other three methods. The story detection approaches are able to cluster together tweets from other conversations of the same sub-story. The F-score of the baseline is unusually high due to a precision of 1. However, it is not useful as it could detect only one conversation of a sub-story and generates a large number of clusters (same as the number of conversations).

In order to investigate the variation in performance across individual sub-stories, 3 major sub-stories are selected at random in the Ferguson and Ottawa data sets. As can seen in Table 7, performance patterns for LSH, SC, and HDP remain unchanged, i.e. LSH has the best precision, while HDP – the best recall and F-score. The latter is able to find most of the tweets associated with sub-stories with a good precision.

Table 8
Results of HDP, SC, and LSH on FSD for different parameter settings. Best results are indicated in bold letters.

Method	FSD		
	P_{micro}	R_{micro}	F_{micro}
HDP (k200)	0.3181	**0.7765**	0.4523
HDP (k300)	0.3558	0.7549	**0.4863**
SC (k200)	0.1564	0.7683	0.2598
SC (k400)	0.1529	0.5406	0.2383
LSH (k12h56b10)	**0.9792**	0.2279	0.3697
LSH (k13h71b10)	0.8128	0.2428	0.3739

Table 9
Results of HDP, SC, and LSH on FAcup for different parameter settings. Best results are indicated in bold letters.

Method	FAcup		
	P_{micro}	R_{micro}	F_{micro}
HDP (k100)	0.1441	**0.1683**	0.1552
HDP (k200)	0.3023	0.1438	**0.1949**
SC (k400)	0.0582	0.0947	0.0721
SC (k1000)	0.1281	0.0875	0.1039
LSH (k12h56b10)	0.4975	0.0881	0.1496
LSH (k13h71b10)	**0.4992**	0.0979	0.1636

7.1.2. Performance on FSD data

The next experiment compares the methods on the publicly available FSD data set. As noted before, FSD is a story detection data set with some stories having vocabulary overlap. We evaluate the performance of the HDP approach against the standard story detection approaches (see Table 8), which provide a good performance in this data set. Here, LSH has very high precision but low recall. HDP and SC outperform LSH in recall, while HDP precision is better than that obtained for SC. HDP is again found to have the highest F-score, indicating the dominance of HDP over SC and LSH on the FSD data set.

Similar to the sub-story data sets, LSH produces very small clusters, which split the tweets belonging to a particular story across multiple threads resulting in higher precision but low recall. For instance in the case of the story on "Death of Amy Winehouse" with 726 tweets, the corresponding LSH cluster contains 109 tweets mostly from that story. The precision for this story is thus 0.90, while recall is only 0.13.

In the FSD data, we observed that LSH produced around 1500 clusters in total, after ignoring clusters with fewer than 3 tweets. Spectral clustering, on the other hand, tends to cluster together tweets from related stories, resulting in few large story clusters. For example, some of the tweets from the two stories ('Death of Amy Winehouse' and 'Betty Ford dies') are put into the same cluster. In the case of 'Death of Amy Winehouse', the corresponding SC cluster has 821 tweets, with 0.59 precision and 0.67 recall. This is mainly due to SC clustering words rather than messages, and thus merging sub-stories sharing common vocabulary.

HDP provides a more balanced result with comparatively higher precision and recall. It is a more fine grained approach which can distinguish subtle differences in various stories, due to the hierarchical modelling of the topics with some shared vocabulary. In the case of 'Death of Amy Wine house', the corresponding HDP cluster has 660 tweets with 0.81 precision and 0.73 recall.

In conclusion, this experiment has demonstrated that HDP performs very well also on story detection data sets and tasks.

7.1.3. Performance on FAcup data

We study the performance of the approaches on another publicly available data set, FAcup. This data exhibits properties similar to Ferguson and Ottawa since all the tweets belongs to a common main event, i.e. a football match. However, it is more challenging due to the very high overlap in vocabulary across the sub-stories (most of the sub-stories are related to goals) and the occurrence of sub-stories in a small time frame.

Table 9 compares the performance HDP, SC, and LSH on the FAcup data. The methods struggle to separate the tweets into the different sub-story clusters, which leads to lower precision. Broadly speaking, the results obtained on the FAcup data are similar to those reported in Table 3 on the sub-story detection data sets. Again, HDP outperforms LSH and SC, thanks to superior recall and F-score, while LSH maintains the best precision. Overall, the performance is low as this data set is more challenging than FSD, Ferguson, and Ottawa.

Table 10

Adjusted mutual information scores for HDP, SC and LSH on the Ferguson, Ottawa, FSD and FAcup data sets. The best AMI scores obtained for different parameter setting of the approaches are reported and the best results are shown in bold.

Method	Ferguson	Ottawa	FSD	FAcup
HDP (k100)	0.46	0.59	**0.70**	0.10
HDP (k200)	0.46	0.55	0.67	**0.11**
HDP (k300)	**0.47**	**0.60**	0.65	0.10
SC (k200)	0.40	0.39	0.65	0.07
SC (k400)	0.38	0.43	0.45	0.07
SC (k2000)	0.39	0.42	0.31	0.08
LSH (k12 h56 b10)	0.40	0.46	0.23	0.08
LSH (k13 h71 b10)	0.40	0.47	0.24	0.09

7.1.4. Discussion

The experiments presented above demonstrated that LSH generally produces a large number of clusters with high precision but low recall. For instance, on the London riots data set, it produced around 45,000 clusters. In contrast, HDP and SC achieve similar performance with only 2500 and 5000 clusters, respectively.

In general, LSH tends to create numerous very small clusters (mostly containing re-tweets), which explains its very high precision. On the other hand, SC tends to cluster together similar categories, which lowers precision. HDP distinguishes subtle topical differences, resulting in more balanced precision and recall. Another noteworthy observation is that, in general, increasing the number of clusters in HDP and SC leads to improved precision but at the cost of recall. Thus, depending on application needs, HDP and SC make it possible to trade off some recall for better precision.

With respect to the metrics used above, precision, recall and F-score do not penalize methods, such as LSH, which produce a large number of small clusters, and thus the corresponding F-score is often high due to their high precision. Such methods, however, are not useful in practice as an user has to navigate over a large number of clusters, in search of important sub-stories. Therefore, in our final experiment we use adjusted mutual information (AMI) (Vinh et al., 2009), which takes cluster size and cluster numbers into account. The improvement in clustering quality due to HDP is clearly visible also with adjusted mutual information, which corrects for agreement by chance due to a larger number of clusters.

7.2. Adjusted mutual information experiments

The information theoretic, adjusted mutual information measure (AMI) (Vinh et al., 2009) is used to evaluate cluster quality. In prior work, information theoretic measures, such as mutual information, have been shown as being well suited to comparing the performance of clustering approaches (Banerjee, Dhillon, Ghosh, & Sra, 2005; Meilă, 2005). These measures are theoretically grounded and provide a better evaluation of cluster quality.

Mutual information (MI) between two clusterings $\mathbf{U} = \{U_1, \ldots, U_R\}$ (true clustering of tweets) and $\mathbf{V} = \{V_1, \ldots, V_C\}$ (generated clustering of tweets) quantifies the information shared among them and provides the reduction in uncertainty on $\mathbf{U}$ upon observing $\mathbf{V}$. The MI score between $\mathbf{U}$ and $\mathbf{V}$, is computed as

$$MI(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^{R} \sum_{j=1}^{C} p(i, j) \log \frac{p(i, j)}{p(i) p(j)}. \tag{7}$$

Here, $p(i)$ is the probability that tweets belong to the cluster U_i, $p(j)$ is the probability that tweets belong to the cluster V_j, and $p(i, j)$ is the probability that tweets belong to both the clusters U_i and V_j. When clusterings are identical, MI score takes a higher value upper bounded by $max\{H(\mathbf{U}), H(\mathbf{V})\}$, where $H(\mathbf{U}) = -\sum_{i=1}^{R} p(i) \log(p(i))$ is the entropy of the clustering $\mathbf{U}$. If the clusterings are disjoint, MI score is close to zero. One can also use a normalized MI (NMI) score, which normalizes the MI score to be between zero and one.

The shortcoming of the MI and NMI scores, however, is that they do not correct for clusters that occur by chance. They do not have a constant baseline value, i.e. the average value obtained for a random clustering of the data (Vinh et al., 2009). Consequently, these scores tend to be higher for results with larger number of clusters, or when the ratio of the total number of data points to number of clusters is small. In particular, they would produce a high score for an approach, which categorizes each tweet into a separate cluster.

Therefore, in our experiments we consider adjusted mutual information (AMI) (Vinh et al., 2009), which is corrected for chance by subtracting the expected mutual information score from both the numerator and denominator of the normalized mutual information score. The AMI score is calculated as follows

$$AMI(\mathbf{U}, \mathbf{V}) = \frac{MI(\mathbf{U}, \mathbf{V}) - \mathbb{E}\{MI(\mathbf{U}, \mathbf{V})\}}{max\{H(\mathbf{U}), H(\mathbf{V})\} - \mathbb{E}\{MI(\mathbf{U}, \mathbf{V})\}}. \tag{8}$$

Tables 10 and 11 provide the AMI scores obtained by HDP, SC and LSH on the different data sets. As can seen, HDP has the best performance, as measured by its AMI score. In this case, we also note that SC demonstrated improved performance

Table 11

Adjusted mutual information scores for HDP, SC and LSH on the London riots data set. The best AMI scores obtained for different parameter setting of the approaches are reported and the best results are shown in bold.

Method	LondonRiots
HDP (k25)	**0.32**
HDP (k50)	0.31
HDP (k100)	0.29
SC (k50)	0.31
SC (k100)	0.31
SC (k200)	0.28
LSH (k12 h56 b10)	0.29
LSH (k13 h71 b10)	0.30

Table 12

Top 5 Topics identified by HDP and SC on the Ferguson data set.

HDP		SC	
Shared	Police, Ferguson	Shared	None
Topic 1	Suspect, Robbery, Brown, Mike, Officer, Involved	Topic 1	Beat, Charged, Man, Property, Uniforms
Topic 2	Officer, Darren, Wilson, Shot, Brown, Michael	Topic 2	Brown, Darren, Michael, Officer, Shot, Wilson
Topic 3	Chief, Stopped, Robbery, Says, Street, Walking	Topic 3	Law, Live, Militarized, State, Town, Victim
Topic 4	Charged, Beat, Man, Bleeding, Uniforms, Property, Destruction	Topic 4	Before, Boy, Community, Dogs, Fergusonshooting
Topic 5	Store, Video, Stills, Surveillance, Robbery, Brown, Release, Michael	Topic 5	Mike, Name, Release, Police

Table 13

Running times of the approaches on London riots, Ferguson, Ottawa, FSD and FAcup data sets.

Method	LondonRiot (h)	Ferguson (s)	Ottawa (s)	FSD (s)	FAcup (s)
HDP	2	196	55	661	188
SC	1.5	183	52	550	128
LSH	4	151	35	511	152

and tends to be better than LSH on most data sets. As expected, the AMI score penalizes the LSH algorithm, which produces a very large number of clusters, since the expected mutual information score grows as the number of clusters increases. Note that the AMI scores of FSD is high and that of FAcup is low due to the same reasons mentioned in Sections 7.1.2 and 7.1.3, respectively.

7.3. Topic detection

Since HDP and SC are topic based and can describe a cluster through key terms, this is not the case for LSH. Therefore, in this section we examine topics within sub-stories, as identified automatically by these two methods.

In particular, Table 12 shows 5 major topics learnt by HDP and SC from the Ferguson data. We found that HDP learns topics corresponding to major stories in the Ferguson data set. For instance, Topic 1, Topic 2 and Topic 3 correspond to Story 1, Story 4 and Story 7 in the Ferguson data set, described in Table 1. The first two topics detected by SC correspond to Story 6 and Story 4 of the Ferguson data.

7.4. Runtime efficiency

We study the runtime of different approaches on the data sets and check their practical usability. Table 13 provide runtime comparisons of HDP, LSH and SC approaches on different data sets. The algorithms are run on a Linux computer having 4 core Intel CPU with 3.40 GHz speed and 16 GB RAM. In terms of run time, the performance of all the approaches are comparable in Ferguson, Ottawa, FSD and FAcup. In the case of London riots, LSH is found to have relatively higher runtime.

8. Discussion and implications

Social networks such as Twitter provide real time information on various events happening around the world. Sub-story detection in Twitter provides researchers, journalists, and government organizations the ability to track the evolution of various stories associated with a main story. For instance, it helps journalists to detect various stories associated with U.S. presidential elections and government to track stories arising around natural disasters such as earthquakes. The proposed

approach based on HDP could detect accurately most of the sub-stories associated with a main story in real time. HDP learns sub-topics which can distinguish subtle differences in the topics associated with the sub-stories. It will be useful for news agencies and governments to more accurately track the evolution of sub-stories and take appropriate remedial measures. The sub-topics learnt by HDP from the Twitter helps humans to understand the content of sub-stories without actually inspecting a large volume of tweets. The keywords extracted from sub-topics also implicitly offer a way of summarizing the sub-stories or can be used as input to more sophisticated summarization algorithms.

We take into account the conversational structure in Twitter which allows our model to more accurately track the evolution of sub-stories. This is particularly useful in applications such as rumour detection where early detection of rumours is important. Categorizing the conversational tweets into the cluster of the source tweet helps in debugging the truthfulness of a rumour mentioned in the source tweet. For instance, the presence of words such as 'incorrect' and 'unbelievable' in the reply tweets often indicate that the topic mentioned in the source tweet is not true.

We provide a better measure to evaluate the clustering quality in sub-story detection by using adjusted mutual information. We observed that standard story detection approaches such as LSH when applied to sub-story detection task, produced a large number of small sized highly accurate clusters. Standard metrics based on precision favor such clustering approaches but they may not be useful in practice. Though F-score consider recall as well, very high precision often leads to a good F-score. AMI takes into account number of clusters produced by the approach and penalizes those which produces large number of clusters. By correcting agreement between clusters due to chance, AMI measure better reflects the clustering quality of the approaches. We proposed to use it for comparing the quality of clusters produced in the sub-story detection task. HDP performed far better than other approaches in terms of AMI score which makes it an ideal candidate for sub-story detection.

The application of the proposed approach to streaming data requires one to partition the stream into intervals and apply the HDP algorithm to the textual content in each interval. The same procedure has to be followed for the SC algorithm, while LSH is directly applicable to the streaming data setting. Topics in the adjacent intervals could overlap and the proposed approach in the current form treat the topics across the intervals to be independent. In order to capture this dependence, one could consider an additional hierarchy over the base distributions generating topics in the adjacent intervals. We intend to develop this model for sub-story detection as a future work. Runtime experiments showed that HDP is as fast as LSH and is useful for real time sub-story detection. The HDP approach requires multiple passes over the data to learn the parameters. An improvement in the runtime performance could be achieved by considering an online HDP algorithm (Wang, Paisley, & Blei, 2011), which performs a single pass over the data to learn the parameters. Our preliminary experiments showed that this comes at the cost of deteriorated precision-recall performance in sub-story detection.

9. Conclusion

This paper introduced the sub-story detection task, which differs from the previously studied story detection task. Secondly, we proposed a probabilistic topic model (hierarchical Dirichlet processes (HDP)) for automatic sub-story detection. HDP performs hierarchical modelling of topics and is effective in modelling sub-stories by learning sub-topics associated with the common topic of the shared real-world event.

HDP performance was compared against spectral clustering and locality sensitive hashing on several sub-story detection and story detection data sets. In general, we found that SC provides good recall, while LSH provides good precision. HDP, on the other hand, was found to have balanced precision and recall and achieves the best F-scores on all data sets. This demonstrates that HDP can handle effectively the subtle differences in sub-stories, which leads to an improved clustering performance. The performance of HDP is better than the rest with respect to F-score on the best parameter settings, but not on all the parameter settings. The superior performance of HDP is substantiated by evaluating cluster quality via adjusted mutual information. Here, HDP is found to perform better than the rest of the approaches in all the parameter settings.

Lastly, our experiments also demonstrated that considering the conversational structure of tweet threads significantly improved performance of the sub-story detection approaches.

Future work will address sharing of the topics across adjacent intervals, adding one more layer of hierarchy in the HDP model. We will also address the task of automatic sub-story ranking, which will enable users, such as journalists or emergency responders, to identify and focus on the most important sub-stories within a large volume of tweets surrounding major world events.

Acknowledgements

References

Abramson, D., Lees, M., Krzhizhanovskaya, V., Dongarra, J., Sloot, P. M., Tan, Z., et al. (2014). A multi-layer event detection algorithm for detecting global and local hot events in social networks. *Procedia Computer Science, 29*, 2080–2089.

Aggarwal, C. C. (2014). Mining text and social streams: A Review. *SIGKDD Explorer Newsletters, 15*, 9–19.

Aiello, L. M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., … Jaimes, A. (2013). Sensing trending topics in twitter. *IEEE Transactions on Multimedia, 15*(6), 1268–1282.

Akbari, M., Hu, X., Nie, L., & Chua, T. (2016). From Tweets to Wellness: Wellness Event Detection from Twitter Streams. In *Proceedings of AAAI conference on artificial intelligence* (pp. 87–93).

Allan, J. (2002). Introduction to topic detection and tracking. In *Topic detection and tracking* (pp. 1–16). Kluwer Academic Publishers.

Banerjee, A., Dhillon, I. S., Ghosh, J., & Sra, S. (2005). Clustering on the unit hypersphere using von mises-Fisher distributions. *Journal of Machine Learning Research, 6*, 1345–1382.

Becker, H., Naaman, M., & Gravano, L. (2011). *Beyond trending topics: Real-world event identification on twitter*. International AAAI Conference on Weblogs and Social Media.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Boecking, B., Hall, M., & Schneider, J. (2015). Event prediction with learning algorithmsa study of events surrounding the egyptian revolution of 2011 on the basis of micro blog data. *Policy and Internet, 7*(2), 159–184.

Bouma, G. (2009). Normalized (Pointwise) Mutual Information in Collocation Extraction. In *Proceedings of the biennial GSCL conference* (pp. 31–40).

Chakrabarti, D., & Punera, K. (2011). Event summarization using tweets. In *Proceedings of international AAAI conference on weblogs and social media*.

Chierichetti, F., Kleinberg, J., Kumar, R., Mahdian, M., & Pandey, S. (2014). Event detection via communication pattern analysis. In *Proceedings of international AAAI conference on weblogs and social media*.

Cigarran, J., Castellanos, ngel, & Garca-Serrano, A. (2016). A step forward for topic detection in twitter: An FCA-based approach. *Expert Systems with Applications, 57*, 21–36.

D'Andrea, E., Ducange, P., Lazzerini, B., & Marcelloni, F. (2015). Real-Time detection of traffic from twitter stream analysis. *IEEE Transactions on Intelligent Transportation Systems, 16*(4), 2269–2283.

Diao, Q., Jiang, J., Zhu, F., & Lim, E.-P. (2012). Finding bursty topics from microblogs. In *Association for computational linguistics (ACL)* (pp. 536–544).

Dong, X., Mavroeidis, D., Calabrese, F., & Frossard, P. (2015). Multiscale event detection in social media. *Data Mining and Knowledge Discovery, 29*(5), 1374–1405.

Fang, Y., Zhang, H., Ye, Y., & Li, X. (2014). Detecting hot topics from twitter: A Multiview approach. *Journal of Information Science, 40*(5), 578–593.

Farzindar, A., & Wael, K. (2015). A survey of techniques for event detection in twitter. *Computational Intelligence, 31*(1), 132–164.

Hjort, N. L., Holmes, C., Mller, P., & Walker, S. G. (2010). *Bayesian nonparametrics*. Cambridge, New York: Cambridge University Press.

Li, C., Sun, A., & Datta, A. (2012). Twevent: Segment-based event detection from tweets. In *Proceedings of ACM international conference on information and knowledge management (cikm)* (pp. 155–164).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Meilă, M. (2005). Comparing clusterings: An axiomatic view. In *Proceedings of International conference on machine learning* (pp. 577–584).

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. The MIT Press.

Nichols, J., Mahmud, J., & Drews, C. (2012). Summarizing Sporting Events Using Twitter. In *Proceedings of ACM international conference on intelligent user interfaces* (pp. 189–198).

Petrović, S., Osborne, M., & Lavrenko, V. (2010). Streaming First Story Detection with Application to Twitter. In *North American chapter of the association for computational linguistics: Human language technologies* (pp. 181–189).

Petrović, S., Osborne, M., & Lavrenko, V. (2012). Using Paraphrases for Improving First Story Detection in News and Twitter. In *North American chapter of the association for computational linguistics : Human language technologies* (pp. 338–346).

Polykarpos, M., Giannis, N., Francois, R., Yannis, S., & Michalis, V. (2015). Degeneracy-based real-time sub-event detection in twitter stream. In *Proceedings of International AAAI conference on weblogs and social media* (pp. 248–257).

Preotiuc-Pietro, D., Samangooei, S., Lampos, V., Cohn, T., Gibbins, N., & Niranjan, M. (2013). Clustering models for discovery of regional and demographic variation. *Technical Report*.

Preotiuc-Pietro, D., Srijith, P. K., Hepple, M., & Cohn, T. (2016). Studying the temporal dynamics of word co-occurrences: An application to event detection. *Language Resources and Evaluation Conference*, 4380–4387.

Procter, R., Crump, J., Karstedt, S., Voss, A., & Cantijoch, M. (2013). Reading the riots: What were the police doing on twitter? *Policing and society, 23*(4), 413–436.

Rajaraman, A., & Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake shakes twitter users: Real-time event detection by social sensors. In *World Wide Web (WWW)* (pp. 851–860).

Sasaki, K., Yoshikawa, T., & Furuhashi, T. (2014). Online topic model for twitter considering dynamics of user interests and topic trends. In *Empirical methods in natural language processing* (pp. 1977–1985).

Sethuraman, J. (1994). A constructive definition of dirichlet priors. *Statistica Sinica, 4*, 639–650.

Shen, C., Liu, F., Weng, F., & Li, T. (2013). A Participant-based Approach for Event Summarization Using Twitter Streams. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1152–1162).

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation.. *Pattern Analysis Machine Intelligence, 22*, 888–905.

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association, 101*, 1566–1581.

Vicient, C., & Moreno, A. (2015). Unsupervised topic discovery in micro-blogging networks. *Expert Systems with Applications, 42*(17–18), 6472–6485.

Vinh, N. X., Epps, J., & Bailey, J. (2009). Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of International conference on machine learning* (pp. 1073–1080).

Vo, D.-T., Hai, V. T., & Ock, C.-Y. (2015). Exploiting language models to classify events from twitter. *Computational Intelligence and Neuroscience*, 401024:1–401024:11.

Wang, C., Paisley, J., & Blei, D. M. (2011). Online Variational Inference for the Hierarchical Dirichlet Process. In *Proceedings of International conference on artificial intelligence and statistics: 15* (pp. 752–760).

Wang, X., Zhu, F., Jiang, J., & Li, S. (2013). Real time event detection in twitter. In J. Wang, H. Xiong, Y. Ishikawa, J. Xu, & J. Zhou (Eds.), *Web-age information management* (pp. 502–513).

White, S., & Smyth, P. (2005). A Spectral Clustering Approach to Finding Communities in Graphs. *SIAM International Conference on Data Mining*, 76–84.

Wurzer, D., Lavrenko, V., & Osborne, M. (2015a). Tracking Unbounded Topic Streams. In *Association for computational linguistics* (pp. 1765–1773).

Wurzer, D., Lavrenko, V., & Osborne, M. (2015b). Twitter-scale New Event Detection via K-term Hashing. In *Empirical methods in natural language processing* (pp. 2584–2589).

Xie, W., Zhu, F., Jiang, J., Lim, E. P., & Wang, K. (2013). TopicSketch: Real-time bursty topic detection from Twitter. In *Proceedings of International conference on data mining* (pp. 837–846).

Xing, C., Wang, Y., Liu, J., Huang, Y., & Ma, W.-Y. (2016). Hashtag-based sub-event discovery using mutually generative LDA in Twitter. In *Proceedings of AAAI conference on artificial intelligence* (pp. 2666–2672).

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., & Li, X. (2011). Comparing Twitter and traditional media using topic models. In *Proceedings of European conference on advances in information retrieval* (pp. 338–349).

Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., & Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PLOS One, 11*(3), 1–29.

Zubiaga, A., Spina, D., Amigó, E., & Gonzalo, J. (2012). Towards Real-time Summarization of Scheduled Events from Twitter Streams. In *Proceedings of ACM conference on hypertext and social media* (pp. 319–320).