Studying the Temporal Dynamics of Word Co-Occurrences: An Application to Event Detection

Daniel Preoţiuc-Pietro[1], **Srijith P.K.**[2], **Mark Hepple**[2], **Trevor Cohn**[3]
[1]Computer and Information Science, University of Pennsylvania, USA
[2]Department of Computer Science, The University of Sheffield, UK
[3]Computing and Information Systems The University of Melbourne, Australia
danielpr@sas.upenn.edu, {pk.srijith, m.r.hepple}@sheffield.ac.uk, t.cohn@unimelb.edu.au

Abstract

Streaming media provides a number of unique challenges for computational linguistics. This paper studies the temporal variation in word co-occurrence statistics, with application to event detection. We develop a spectral clustering approach to find groups of mutually informative terms occurring in discrete time frames. Experiments on large datasets of tweets show that these groups identify key real world events as they occur in time, despite no explicit supervision. The performance of our method rivals state-of-the-art methods for event detection on F-score, obtaining higher recall at the expense of precision.

Keywords: Topic Detection & Tracking, Information Extraction, Information Retrieval, Text Mining

1. Introduction

Algorithms based on word co-occurrences have a long tradition in NLP and have been used successfully for applications ranging from sentiment analysis to thesaurus learning, collocation extraction and discovering multiword expressions (Turney, 2002; Curran, 2004; Sag et al., 2002; Evert, 2005). However, in most cases, these scores have been computed using underlying static corpora and ignoring any temporal variation.

In this paper we study the changing behaviour of word co-occurrences over time using social media data. We hypothesise that co-occurrences between words will change over time as a response to real world events. Increased social media usage enables us to extract for analysis large scale streaming data, previously largely unavailable to researchers. Data arising from these sources, specifically Twitter, has been shown to reflect real world events in a timely fashion (Sakaki et al., 2010). Using this data we extract and analyse co-occurrence statistics over time.

To illustrate the temporal evolution of word co-occurrences, consider the example of the word 'riot'. Using static corpora, e.g. newswire or Wikipedia entries, the highest co-occurring words will be syntactically or semantically related words (e.g. 'city', 'police', 'riots'). However, if we study the change of frequently co-occurring words with 'riot' over the course of 2011, we notice that in January 'Egypt' and 'Lebanon' commonly co-occur (due to the riots in both countries and the abundance of news and opinions about these events), while in August 2011, 'riot' co-occurs more with words like 'U.K.', 'London' or 'Hackney' where a series of riots took place. Moreover, in small time frames, we observe increases of co-occurrences of other seemingly unrelated words (e.g. 'Bieber') because of popular opinions or viral messages ('If Bieber wins, we riot').

Our application is the problem of discovering and clustering words specific of events – newsworthy happenings (McCreadie et al., 2013) – using their mutual word co-occurrence scores. For computing these scores we use Normalised Pointwise Mutual Information (NPMI) (Bouma, 2009). We group tweets in time windows and treat the co-occurrence score in that time window as a clustering similarity measure. We develop an efficient unsupervised spectral clustering algorithm that uncovers clusters of co-occurring words which can be related to events in the dataset. Using information extracted from the data, we can measure the magnitude of an event and, using a cluster centrality measure, automatically select relevant messages that can be used as labels when presenting the clusters to end users.

Our results on event detection tasks using tweets, show that our method rivals state-of-the-art message based event detection techniques. Our method is especially useful for downstream applications where higher recall is desirable, such as time dependent information retrieval. The data from this study is freely available.[1]

2. Related Work

The study of word co-occurrences has a long tradition in Natural Language Processing. Measures of co-occurrence have been studied by Fano (1961) and Dunning (1993). In NLP, they have been used for finding collocations or multiword expressions in documents (Sag et al., 2002; Evert, 2005), for weighting vectors for measuring distributional semantic similarity (Turney and Pantel, 2010) or for finding the sentiment polarity of words (Turney, 2002). More related to this study, Newman et al. (2010) shows that the best performance for measuring topic coherence is obtained using the Pointwise Mutual Information co-occurrence metric.

Spectral clustering is a state-of-the-art clustering method that has been used for various tasks like image segmentation (Shi and Malik, 2000) or detecting communities in networks (Newman, 2006). The application of spectral clustering methods in NLP has been limited because of increased storage space and runtime when faced with large-scale text datasets (Lin and Cohen, 2010).

[1] http://www.sas.upenn.edu/~danielpr/clusters.html

Cluster analysis and modelling events over time have been studied in different contexts. Hall et al. (2008) studies the evolution and trends of topics by using topic modeling and matching the topics obtained independently at different time intervals. Wang and McCallum (2006) develop a topic model that explicitly embeds time as an observed variable. Several other approaches have integrated time into a probabilistic graphical model of text (Al Sumait et al., 2008; Gohr et al., 2009; Wang et al., 2008). These papers used datasets of long and well structured documents on a restricted set of topics (e.g. conference proceedings or political addresses).

In social media, event detection represents an extremely challenging task due to the large volume and heterogeneity of the data. Methods have been developed for identifying events in general (Becker et al., 2011a), for finding new emergent topics (Petrović et al., 2010) or focusing on particular events such as earthquakes (Sakaki et al., 2010). For newswire, event ('topic') detection was first researched in the context of 'Topic Detection and Tracking' (Allan, 2002). Methods for topic detection can be grouped into document and feature based approaches. The former aims to cluster documents into events and then extract features from the clusters (Brants et al., 2003). The later category is based on identifying and clustering features that are representative of events (Kleinberg, 2002). This work presents an approach of the later category, adopting a word-level view to extract relevant clusters. We deal with social media peculiarities (e.g. non-event related messages) by removing non-event related clusters and make the method scalable to millions of messages by clustering words (which are of fixed size) rather than messages.

3. Pointwise Mutual Information

A standard method of uncovering associations between words is by computing the Pointwise Mutual Information (PMI). The PMI value is usually computed over a large but static corpus, such as Wikipedia. In this study, we experiment with computing association scores for each word pair in separate time intervals by splitting our dataset based on the timestamp of the texts.

PMI is an information theoretic measure that indicates which words tend to often co-occur in a context. It measures the relative difference between observed word co-occurrences, and their expected co-occurrence assuming independence,

$$\mathrm{PMI}(X,Y) = \alpha \cdot \log \frac{P(x,y)}{P(x) \cdot P(y)}\,, \qquad (1)$$

where α is a normalisation factor, here set to $\alpha = -\log P(x,y)$ following (Bouma, 2009) to address issues with interpretability and sensitivity to low-count events in regular PMI (where $\alpha = 1$). (Church and Hanks, 1990). This normalised variant of PMI (NPMI) is bounded in the $[-1,1]$ interval and can be easily interpreted: word pairs with a negative NPMI co-occur less often than expected under independence, a positive NPMI means more often, and 0 denotes equality. The maximum value NPMI=1 implies that both words exclusively appear together. We use NPMI as our word co-occurrence measure and use this as a similarity measure in our clustering approach.

Word1	Word2	NPMI	Type
arrests	yemen	0.699	news
publish	trailers	0.678	news
bestfriends	forming	0.678	news
g-slate	spotted	0.675	news
activist	arrests	0.674	news
china's	stealth	0.674	news
blake	griffin	0.672	proper name
magazines	merchandise	0.669	news
activist	yemen	0.667	news
actors	showcase	0.667	news
cameras	g-slate	0.664	news
angeles	los	0.662	proper name

Table 1: Top NPMI values for 23 Jan 2011, 9-10am. Word1, Word2 are in alphabetical order.

4. Clustering

We test our hypothesis of word co-occurrence changing with time using the downstream task of event detection. An event represents a timely real world story – similar to 'topics' in the TDT framework (Allan, 2002). Given that tweet content is timely, our events are rarely seminal and usually die out in hours.

The word co-occurrence measure is regularly indicative either of semantically related words or collocations. However, given data from a narrow time frame, we hypothesise that word co-occurrence also indicates an underlying event that triggers words to appear together across multiple texts authored at similar times. By clustering words based on co-occurrences we expect to find the terms that can reliably and uniquely characterise an event. Table 1 presents a list of the highest word similarity pairs in a one hour time window. These are either collocations or terms indicative of timely news stories.

Our clustering goals are to group pairs of highly co-occurring words and their local neighbours. Also, clusters should include words that may not co-occur much, but are distributionally similar (e.g. 'recall' and 'recalls'; 'Moscow' and 'Domodedovo'). We experiment with two algorithms: the widely known K-means clustering (MacQueen, 1967) and spectral clustering (Shi and Malik, 2000).

4.1. Spectral Clustering

Spectral clustering (Shi and Malik, 2000; Ng et al., 2002) is a state-of-the-art method for clustering that solves a graph partition problem on the similarity graph. It has a solid theoretical foundation in spectral graph theory (Chung, 1997) and is designed for situations when the clusters are non-convex and can not be identified by the use of a spherical metric. The algorithm is known to be particularly useful when assumptions cannot be made about the shape of the clusters and is suited for the goal of preserving local neighbourhoods. The performance of the algorithm is dependent on the underlying similarity graph. In our case, the similarity graph can be directly linked to a graph where the vertices are the words in the vocabulary and the edges between ver-

tices are weighted using the computed NPMI in the given time frame.

The algorithm works as follows. We define W as the similarity matrix, D as the diagonal matrix with the elements corresponding to the row sums of W and L the graph Laplacian chosen to suit the graph optimisation problem. In our case, we use $L_{\text{RW}} = I - D^{-1}W$, corresponding to the problem of finding a graph partitioning such that a random walk in the similarity graph seldom changes cluster memberships. This is well suited to identifying clusters of different sizes. Because the solution to the optimisation problem is NP-hard, spectral clustering solves a relaxed version:

$$Z_{\text{RW}} = \arg\min_{Z} \text{tr}(Z^\top LZ) \text{ s.t. } Z^\top Z = I \quad (2)$$

The optimal solution to Equation 2 consists of the matrix Z containing the eigenvectors corresponding to the k smallest eigenvalues of L (ignoring the 0 values). The original data points are then clustered with a run of a standard clustering algorithm – here k-means – on the matrix Z. We refer the interested reader to Von Luxburg (2007) for an in-depth tutorial on spectral clustering.

A large dataset leads to a high dimensional similarity matrix. The spectral clustering algorithm needs to solve an eigenvalue problem and thus its performance is determined by the structure of the similarity matrix. Fortunately, our similarity graph can be represented as a sparse matrix, as each word will only be associated with a limited subset of words with non-zero values. Moreover, many of these co-occurrence scores are insignificantly small and can be safely ignored.

5. Data

We collected and used data from Twitter for all our experiments. We used this data source because the content is streaming, time dependent and it reflects real world events in a timely fashion (Becker et al., 2011a). We have tokenised all the tweets and filtered out all the non-English tweets using the Trendminer pipeline (Preoţiuc-Pietro et al., 2012). We also removed duplicate tweets, as these messages bias the co-occurrence values, artificially inflating the counts for the terms therein.

For computing the NPMI we consider that two words co-occur if they belong to the same message. Commonly, co-occurrences were computed over a word window. We chose our method because tweets are short (avg. 6 tokens) and we can assume that they refer only to one topic. The vocabulary consists of the most frequent 50,000 words.

Gardenhose Dataset We use the Twitter Gardenhose stream which is a representative sample of 10% of the entire Twitter stream. The collection interval is 23 January – 8 February 2011. In total, after the processing and filtering described above, our dataset totals around 150 million unique English language tweets with an average of about 400,000 tweets/hour over 17 days.

First Story Detection Dataset This dataset consists of labeled events used for evaluating the performance of the first story detection (FSD) system from (Petrović et al., 2012). The FSD corpus consists of 27 events occurred during the period June 2011 to September 2011. There are 2,228 annotated tweets assigned to these events. We augment these tweets with background tweets belonging to the same time period, resulting in a corpus of around 85,000 tweets.

London Riots Dataset This dataset consists of tweets related to the riots taking place in London in 2011. It consists of tweets gathered during 10 days starting from August 6, 2011. The dataset consists of 2.5 million tweets with around 10,000 of them being labeled as belonging to one of the 7 events associated with the London riots. This dataset is more challenging than the FSD data because the tweets forming the background data also are about the London riots and thus share similar vocabulary.

6. Experiments

We start with a quantitative evaluation. This helps us establish the best clustering method and a way to identify optimal parameters based on internal and external cluster evaluation. We then directly evaluate our method on the event detection task and compare its performance to a state-of-the-art approach. We further perform and extensive qualitative analysis of our method. We analyse the output of our method by examining a few known events and show how to automatically infer the magnitude of an event and select relevant labels.

6.1. Quantitative Evaluation

First, we perform an automatic quantitative comparison of clustering methods and analyse the sensitivity to different parameter settings. We use the first day of the Gardenhose dataset split into hourly intervals. We perform 24 individual evaluations (one for each hour) and present the average scores across all hours.

We compare K-means (denoted **K**) and spectral (denoted **S**) clustering as well as random partitioning (denoted **R**). The most important parameter to tune is the number of clusters (**n**), which we vary from 50 to 1000 (due to space constraints we only present a subset of results).

Beyond this, also important for clustering is the underlying similarity matrix/graph, which should be sparse for efficiency reasons. We experiment with a few setups, each building on the previous. These are denoted with suffixes to the clustering method e.g. K-f is K-means with the first setup:

- **-f:** Initially, we discard all the NPMI values below a threshold (here 0.3) and from the resulting graph, keeping the largest connected component. This way, we remove both common words that are uninformative for any event and words that are poorly correlated with others. For comparison purposes, this reduced vocabulary is used in all experiments for the respective hour.
- **-r:** From the resulting graph, we build a mutual k-nearest-neighbourhood graph with $k = 50$, the lowest value that keeps the graph connected as suggested in (Von Luxburg, 2007).
- **-s:** We experiment with 'spreading' the values in the $[0, 1]$ interval by applying a Gaussian similarity function.[2]

[2] $s(x) = 1 - \exp\left(\frac{-x}{2\sigma^2}\right)$

n	R	K	K-f	K-r	K-s	S-f	S-r	S-s
50	0.005	**0.015**	0.013	0.012	0.012	0.010	0.010	0.010
100	0.005	**0.023**	0.020	0.020	0.020	0.016	0.016	0.016
200	0.005	**0.035**	0.034	0.033	0.033	0.030	0.030	0.030
500	0.004	0.066	0.064	0.064	0.064	0.087	0.087	**0.088**

Table 2: Average word coherence. Bold numbers show best performance.

n	K	K-f	K-r	K-s	S-f	S-r	S-s
50	0.20 ↑ 11%	0.13↑ 11%	0.11↑ 11%	0.11↑ 9%	0.10↑ 9%	0.12↑ 9%	0.12↑ 8%
100	0.32 ↑ 9%	0.23↑ 10%	0.20↑ 9%	0.19↑ 7%	0.23↑ 11%	0.25↑ 11%	0.25↑ 11%
200	0.48 ↑ 8%	0.37↑ 9%	0.33↑ 8%	0.32↑ 8%	0.42↑ 15%	0.43↑ 14%	0.43↑ 14%
500	0.68 ↑ 6%	0.59↑ 7%	0.55↑ 7%	0.56↑ 7%	0.64↑ 16%	0.65↑ 16%	0.65↑ 16%

Table 3: Purity on labeled tuples. ↑ $x\%$ is the relative improvement over a controlled random baseline.

Internal Cluster Evaluation As a means of internal cluster evaluation, we compute for each word an average coherence score with respect to the words in its assigned cluster using the full original similarity matrix (even if we use the reduced version in clustering). The coherence score is calculated as

$$Q_w = \frac{\sum_{w,v \in c} \text{NPMI}(w, v)}{|c|}. \quad (3)$$

The average word coherence scores are presented in Table 2. We note that scores are not comparable across different number of clusters, as larger clusters a-priori lead to lower scores (as clusters are larger). We discover first that K-means has better results for lower number of clusters, but this changes in favour of spectral clustering when increasing the number of clusters above 300. This is somewhat expected: the K-means algorithm collapses many words into large clusters ($n = 500, \sigma = 131$) because they have similar values (close to 0) for the majority of dimensions. Large clusters have poor interpretability and are very likely to contain words relevant to multiple clusters/events. Spectral clustering avoids this problem ($n = 500, \sigma = 16$) by performing clustering on a reduced space that provides a better separation. Under this measure, the three setups do not have very different results, with **(s)** having the advantage of a shorter runtime.

External Cluster Evaluation For external evaluation, we need access to a large set of gold standard pairs of words that should be together in the same cluster. As collecting these pairs for timely events is very hard, we consider that a word, its equivalent hashtag and its plural should always appear in the same cluster (e.g. 'packer', '#packer', '#packers'). These words usually have small NPMI values, as they are used in place of each other and rarely co-occur in the same tweet. In our vocabulary there are in total 2009 pairs and 114 triples (due to pruning by largest connected component, not all are present in every clustering). We consider the purity measure (Manning et al., 2008) and present the score relative to a random baseline that keeps the cluster sizes fixed and randomises the assignments similarly to (Bamman et al., 2013). The scores are presented in Table 3, showing that all our models obtain better performance on this challenging task, with spectral clustering with $n = 500$ clusters achieving the best relative improvement. All results compared to the random baseline are statistically significant (t-test, $p < 0.01$). Results follow a similar pattern for the Variation of Information metric (Manning et al., 2008) – not shown here.

Event Detection Our event detection method clusters words, rather than messages. We assign tweets to a event cluster as follows. For each tweet and cluster c, a score is computed as the sum of the word centralities for the words in the tweet part of cluster c. The tweet is assigned to the cluster with the largest score. The word centrality measure is computed as:

$$C_w(c) = \frac{\sum_{x \in c} \text{NPMI}(w, x)}{|c| - 1}.$$

The centrality measures the tokens that are most representative for a cluster, indicated by high co-occurrence values with all the other tokens in the cluster.

We use our best performing method on previous evaluations (**S-s**) – here denoted **SCT** (Spectral Clustering with Time partitioning) – and compare it to the state-of-the-art approach of Petrović et al. (2010). This method uses document similarity and Locality Sensitive Hashing (**LSH**) to create clusters of tweets related to events. We also compare to our method when using co-occurrence information from the entire dataset without partitioning into time slices (**SC**). Clusters detected by the approaches are evaluated against the known tweets associated with the events. Since the approaches are unsupervised, the clusters discovered by them are not aligned to the events. For every event, the alignment is done by finding the cluster that has maximum number of tweets from that event. Evaluation is done with respect to this cluster for the event. We report the performance using the micro-averaged measures of recall and precision over all the events, due to the inconsistencies in size of the tweets associated with the events. We also report the micro-averaged F-score which is the harmonic mean of micro-averaged precision and recall.

We divide the London Riots dataset into 50 partitions with approximately 50,000 tweets each based on their timestamp. In the FSD dataset, we consider 9 partitions with approximately 10,000 tweets each. Results on the FSD and London Riots corpus are presented in Table 4. The approaches are compared on precision, recall and F-score, micro-averaged all events in the dataset. We report the best results obtained in terms of F-score for different parameter settings of the approaches. We also provide the number of

Method	Precision	Recall	F-Score	clusters	Time
SC	0.15	0.76	0.25	200	550 sec
SCT	0.24	0.59	0.34	600	204 sec
LSH	0.81	0.24	0.37	1,500	511 sec

(a) First Story Detection dataset.

Method	Precision	Recall	F-Score	clusters	Time
SC	0.02	0.54	0.05	2,000	10 hr
SCT	0.45	0.25	0.33	5,000	1.5 hr
LSH	0.49	0.22	0.33	45,000	4 hr

(b) London Riots dataset.

Table 4: Event detection results comparing Spectral Clustering (SC), Spectral Clustering with Time partitioning (SCT) and Locality Sensitive Hashing (LSH).

clusters generated by each method to achieve this score.
We observe that both spectral clustering approaches provide a better recall than the LSH approach. The SCT approach is found to provide a performance comparable to that of LSH with respect to F-score. Recall is higher for our method as the clusters are more balanced in the number of words which leads to clusters with similar number of tweets. The LSH approach tends to over generate clusters (especially singleton clusters) for tweets that are different to others. It produces a larger number of small sized clusters leading to an improved precision but low recall. SCT is also faster than LSH in clustering the tweets mainly because it operates on words rather than messages. Thus, SCT represents an effective approach to perform real time event detection in Twitter.

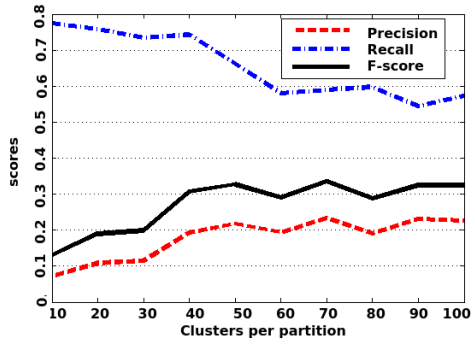


Figure 1: Variation in precision, recall and F-score on increasing the number of clusters per partition for the FSD dataset.

We also analyse the variation in precision, recall and F-score metrics of the SCT approach as we vary the number of clusters in each partition. Figure 1 plots this in the case of FSD dataset. We observe that upon increasing the number of clusters, the precision metric improves while recall degrades. Thus, one can trade off recall for a better precision by choosing a larger number of clusters.

6.2. Qualitative Evaluation

We now switch to qualitatively analyse the results of our method. We focus on a single model that yielded best results on the Gardenhose dataset, i.e. spectral clustering with 500 clusters on the reduced matrix and with similarity function (**S-s**).

Event related	Partially related	Not related	Spam
58	27.5	24	21.5

Table 5: Cluster quality judgements.

Cluster Quality Analysis To get a sense of the information contained in the clusters, we chose a date and hour randomly from our dataset (24 January 2011, 9-10pm G.M.T.) and asked 2 independent annotators to judge how relevant each cluster is to an event on that hour. We only present the clusters with an average word coherence above the threshold of 0.2. The Inter-Annotator agreement (IAA) is 0.67 and results are presented in Table 5 showing that most of the clusters are indeed related to events. Some spam is discovered mostly in the form of slightly altered messages, many of which are the result of automatic tools (e.g. mobile Twitter apps). On average, the number of clusters in each hour with a coherence score above the 0.2 threshold is 175.6 with a standard deviation of 41.

behind	weather	cheaper	works	sea
john	chill	replacement	manager	pure
tweetdeck	across	rubber	bears	boat
medium	canada	originally	coach	pushed
keen	recent	doh	general	breeze
techcrunch	brave		smith	depth
mogul	ch		jerry	probe
tctv	canadians		extension	flowing
	coldest		angelo	curiosity
	warnings		lovie	
0.77	**0.61**	**0.61**	**0.53**	**0.53**

Table 6: Most coherent clusters and coherence score for 24 January 2011 9-10pm GMT.

down	half	free	come	believe
shut	sister	card	other	tried
50	oprah	gift	doing	huge
@50cent	meets	secret	bored	lee
cent	reveal	picked	chat	shooting
wshh	adoption	receive	each	guilty
shutting	oprah's	$1,000	http://tinychat.com	murder
shuts	winfrey	victoria's	tinychat	mass
worldstarhiphop	@oprah	chosen	chatroom	arizona
shuttin	patricia	selected		suspect
	half-sister	drawn		accused (...)
304	**82**	**62**	**47**	**42**

Table 7: Most important clusters by magnitude score for 24 January 2011 9-10pm GMT.

The top clusters in terms of coherence are presented in Table 6 and in terms of magnitude are displayed in Table 7. The magnitude of a cluster is the average number of co-occurrences between all word pairs that belong to the cluster. Notice that clusters are formed by words which co-occur together for three main reasons: **a)** they are representative for an event (e.g. 'weather'; 'down' ; 'works' – clusters denoted by topmost word) **b)** frequent idioms (e.g. 'cheaper') and **c)** slightly altered automated messages (e.g. 'come'; 'free'). However, the number of the latter category is smaller. Our intention of capturing distributionally similar words is illustrated by the presence in the same cluster of pairs of words (e.g. 'wshh' and 'worldstarhiphop',

Query: Taco Bell filling lawsuit
Label: Taco Bell defends its mixture of seasoned meat http://bit.ly/efIzP3
Coherence: 0.53, Magnitude: 73
Date: 25 Jan 2011, 10-11pm

Query: Moscow airport bombing
Label: Suicide bomber kills 35 at Moscow airport http://ind.pn/idWMJj
Coherence: 0.37, Magnitude: 214
Date: 24 Jan 2011, 5-6pm

Query: Kubica crash
Label: Formula 1 driver Robert Kubica injured in rally crash http://ow.ly/3R7IQ
Coherence: 0.47, Magnitude: 140
Date: 6 Feb 2011, 12-1pm

Query: Oprah Winfrey half-sister
Label: Oprah Winfrey has a half-sister. http://bit.ly/i7NNjs
Coherence: 0.29, Magnitude: 43
Date: 24 Jan 2011, 9-10pm

Query: Toyota recall
Label: Toyota recalls nearly 1.7 million vehicles http://lsnlw.com/t/132876715/
Coherence: 0.62, Magnitude: 230
Date: 26 Jan 2011, 6-7am

Query: US Unemployment
Label: Unemployment 9.0% #unemployment #economy
Coherence: 0.22, Magnitude: 108
Date: 4 Feb 2011 2-3pm

Figure 2: Original TREC Microblog query, the most relevant tweet, words in the cluster with font size defined by centrality, coherence, magnitude and date of the 6 events

'techcrunch' and 'tctv') which very rarely co-occur but are representative of the same event (users will use the terms interchangeably but rarely at the same time).

Event Analysis For an in-depth analysis we chose a number of known real-world events that occurred in the Gardenhose dataset. For objectivity, we used a subset of events extracted from the queries of the TREC Microblog track 2011.[3] We discarded single word queries or those that had less than 10 occurrences in any hour for any pair of query words. This is because these pairs could have been discarded in our filtering steps which are necessary for adjusting the NPMI values. Most of the TREC events are lower interest (e.g. 'the release of the Rite') or static over time (e.g. 'global warming and weather'). Having the entire Twitter stream as input would increase recall for these smaller events. An alternative would be using larger time buckets. Out of the remaining set of 13 events, we randomly chose 6 clusters to present in Figure 2. The cluster was chosen automatically as the cluster that contained the majority of the query terms and the time as the peak value of the sum of co-occurrences between all pairs of terms in the query.

We can reliably discover all the events indicated by our queries. A qualitative inspection shows that the clusters contain most of the relevant words to that event. For example, in the 'Oprah' cluster we see that the related event[4] is about the revealing that she has a half-sister that was given to adoption. Her name ('Patricia') together with Oprah's surname, hashtag and username are also present. In the 'Kubica' example, we observe that Robert Kubica was a driver that was badly injured in a rally crash in Italy.[5] Notice other words that describe his activity as a Formula 1 driver, like 'f1', '#f1', 'formula' and the team for which he was racing ('Renault'). The centrality measure emphasises correctly the concepts important to the cluster (e.g. 'Oprah, 'Winfrey', 'sister', 'half-sister' or 'Kubica', 'Robert', 'crash', 'injured'). Moreover, our method finds related words even if they are very frequent in the dataset, which a method based on a tf-idf metric will discard because of high idf (e.g. 'sister' in the 'Oprah' cluster or 'Italy' in the 'Kubica' cluster). We again highlight that our event detection method is unsupervised in that it discovers these events with no supervision or manual tuning.

Method	Average score
Our method	4.33
TREC judged 'Very relevant'	3.61
TREC judged 'Relevant'	3.13
TREC judged 'Not relevant'	1.53

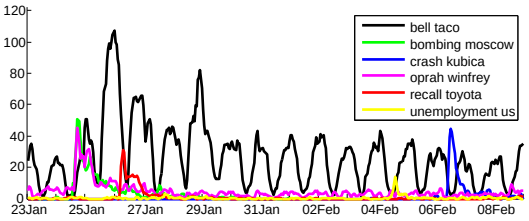
Table 8: Relevance judgement results

Labeling Clusters To label a cluster, we use the tweet with the highest membership score (see the **Event Detection** section). In order to remove short tweets that are not suited for our purpose we keep only tweets with more than 3 tokens. Becker et al. (2011b) shows that, even though using a different measure of similarity and centrality, when

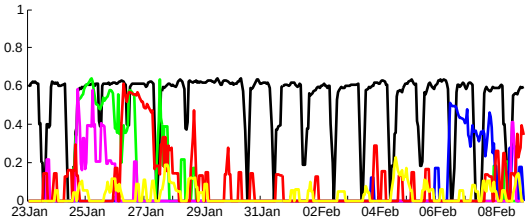
[3] http://trec.nist.gov/data/microblog.html

[4] http://www.bbc.co.uk/news/entertainment-arts-12274349

[5] http://news.bbc.co.uk/sport1/hi/motorsport/formula_one/9388940.stm



(a) Temporal variation of co-occurrence counts



(b) Temporal variation of NPMI values

Figure 3: Temporal variation of co-occurrences

finding representative tweets in a collection, weighting by the centrality of words performs well. Examples of relevant tweets for their cluster are shown as cluster labels in Figure 2.

We find that the most representative tweet to a cluster is a well written piece of text that describes that event. In order to evaluate relevance, we have asked 3 independent annotators to perform the following experiment. They were presented with a URL giving background information about the event and the following tweets: 3 random tweets for each 'highly relevant', 'relevant' and 'not relevant' category as judged as part of the 2011 TREC Microblog task evaluation and the top 3 most relevant tweets as found by our method on the same underlying data. The question they had to answer was: 'On a scale from 1-5, how appropriate is the tweet as a label for that event'. Each annotator judged all the 13 events. The average variance across judges is 0.38. IAA is computed as the average Spearman's ρ between the scores given by the annotator and the average ratings given by all other annotators. The average IAA across all events was 0.875. The results presented in Table 8 show that the words in our clusters identify the correct events and provide a good match for the queries. The relevant tweets were preferred by the human judges against a strong baseline of tweets judged as very relevant to each event.

7. Word Co-occurrence with Time

Recall our initial hypothesis that the co-occurrence distribution of words changes in time based on real world events. We first analyse pairs of words relating to the previous events. We show the temporal evolution of co-occurrence counts and NPMI values of six word pairs in Figure 3. From Figure 3a we observe that pairs like 'Taco Bell' and 'Toyota recall' co-occur often in a daily pattern. For 'Taco Bell' this happens mostly because U.S. users comment on the company's products every day. The other four pairs mostly co-occur around events related to them, with a decay as the relevance of the event diminishes. In the 'Taco Bell' case, this trend is combined with the daily pattern.

8. Conclusions

We have studied the dynamics of word co-occurrences over time. We have shown these change over time as a response to events and can be used to identify them given timely data sources such as Twitter. We have demonstrated our research hypothesis by developing a spectral clustering method based on similarities computed by the NPMI co-occurrence metric. Results have shown that our unsupervised method reliably finds clusters of good quality. Automatic evaluation on event detection datasets shown results competitive to current state-of-the-art. Our method is particularly useful if higher recall is desirable. Further applications, such as extracting cluster labels for events were judged by humans to be very accurate.

Future improvements are possible in our framework. The association measure, here NPMI, can be replaced with those based on word embeddings (Mikolov et al., 2013; Pennington et al., 2014). The temporal dimension of our data can be better modelled either by performing evolutionary clustering (Chakrabarti et al., 2006) where clusters are linked over time or by using streaming methods over a varying time window.

Acknowledgements

This work was partially supported by the European Union under grant agreement no. 611233 PHEME, and the Australian Research Council (Future Fellowship, project no. FT130101105).

9. Bibliographical References

Al Sumait, L., Barbará, D., and Domeniconi, C. (2008). On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *International Conference on Data Mining*, ICDM, pages 3–12.

James Allan, editor. (2002). *Topic detection and tracking: Event-based information organization*. Kluwer Academic Publishers.

Bamman, D., O'Connor, B., and Smith, N. (2013). Learning latent personas of film characters. In *Proceedings of the 51st annual meeting of the Association for Computational Linguistics*, ACL, pages 352–361.

Becker, H., Naaman, M., and Gravano, L. (2011a). Beyond trending topics: Real-world event identification on Twitter. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, ICWSM, pages 438–441.

Becker, H., Naaman, M., and Gravano, L. (2011b). Selecting quality Twitter content for events. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, ICWSM, pages 442–445.

Bouma, G. (2009). Normalized (Pointwise) Mutual Information in collocation extraction. *Proceedings of German Society for Computational Linguistics & Language Technology Conference*, pages 31–40.

Brants, T., Chen, F., and Farahat, A. (2003). A system for new event detection. In *Proceedings of the 26th annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR, pages 330–337.

Chakrabarti, D., Kumar, R., and Tomkins, A. (2006). Evolutionary Clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD, pages 554–560.

Chung, F. (1997). *Spectral Graph Theory*, volume 92. American Mathematical Society.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Curran, J. (2004). From distributional to semantic similarity. *PhD Thesis, University of Edinburgh.*

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Evert, S. (2005). The statistics of word cooccurrences. *Phil. Diss. Stuttgart*.

Fano, R. (1961). *Transmission of information: A statistical theory of communications*. The MIT Press.

Gohr, A., Hinneburg, A., Schult, R., and Spiliopoulou, M. (2009). Topic evolution in a stream of documents. In *Proceedings of the SIAM Data Mining Conference*, SDM, pages 859–870.

Hall, D., Jurafsky, D., and Manning, C. D. (2008). Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 363–371.

Kleinberg, J. (2002). Bursty and hierarchical structure in streams. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD, pages 91–101.

Lin, F. and Cohen, W. (2010). A very fast method for clustering big text datasets. In *Proceedings of the 19th European Conference on Artificial Intelligence*, ECAI, pages 303–308.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

McCreadie, R., Macdonald, C., Ounis, I., Osborne, M., and Petrovic, S. (2013). Scalable distributed event detection for Twitter. In *Proceedings of IEEE International Conference on Big Data*, pages 543–549.

Mikolov, T., Yih, W.-T., and Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 746–751.

Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 100–108.

Newman, M. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3).

Ng, A., Jordan, M., and Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1532–1543.

Petrović, S., Osborne, M., and Lavrenko, V. (2010). Streaming first story detection with application to Twitter. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL, pages 181–189.

Petrović, S., Osborne, M., and Lavrenko, V. (2012). Using paraphrases for improving first story detection in news and Twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL, pages 338–346.

Preoţiuc-Pietro, D., Samangooei, S., Cohn, T., Gibbins, N., and Niranjan, M. (2012). Trendminer: An architecture for real time analysis of social media text. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media, Workshop on Real-Time Analysis and Mining of Social Streams*, ICWSM, pages 38–43.

Sag, I. A., Baldwin, T., Bond, F., Copestake, A. A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing, pages 1–15.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW, pages 851–860.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Turney, P. D. and Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188.

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on Association for Computational Linguistics*, ACL, pages 417–424.

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wang, X. and McCallum, A. (2006). Topics over Time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD, pages 424–433.

Wang, C., Blei, D., and Heckerman, D. (2008). Continuous time dynamic topic models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, UAI.