Analyzing Feature Trajectories for Event Detection

Qi He
qihe@pmail.ntu.edu.sg

Kuiyu Chang
ASKYChang@ntu.edu.sg

Ee-Peng Lim
ASEPLim@ntu.edu.sg

School of Computer Engineering
Nanyang Technological University
Block N4, Nanyang Avenue, Singapore 639798

ABSTRACT

We consider the problem of analyzing word trajectories in both time and frequency domains, with the specific goal of identifying important and less-reported, periodic and aperiodic words. A set of words with identical trends can be grouped together to reconstruct an event in a completely unsupervised manner. The document frequency of each word across time is treated like a time series, where each element is the document frequency - inverse document frequency (DFIDF) score at one time point. In this paper, we 1) first applied spectral analysis to categorize features for different event characteristics: important and less-reported, periodic and aperiodic; 2) modeled aperiodic features with Gaussian density and periodic features with Gaussian mixture densities, and subsequently detected each feature's burst by the truncated Gaussian approach; 3) proposed an unsupervised greedy event detection algorithm to detect both aperiodic and periodic events. All of the above methods can be applied to time series data in general. We extensively evaluated our methods on the 1-year Reuters News Corpus [3] and showed that they were able to uncover meaningful aperiodic and periodic events.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

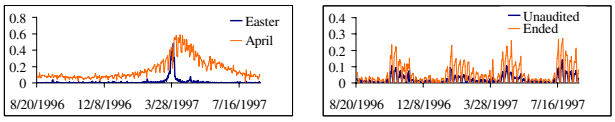
General Terms: Algorithms, Experimentation.

Keywords: feature categorization, event detection, DFT, Gaussian

1. INTRODUCTION

There are more than 4,000 online news sources in the world. Manually monitoring all of them for important events has become difficult or practically impossible. In fact, the topic detection and tracking (TDT) community has for many years been trying to come up with a practical solution to help people monitor news effectively. Unfortunately, the holy grail is still elusive, because the vast majority of TDT solutions proposed for event detection [20, 5, 17, 4, 21, 7, 14, 10] are either too simplistic (based on cosine similarity [5]) or impractical due to the need to tune a large number of parameters [9]. The ineffectiveness of current TDT technologies can be easily illustrated by subscribing to any of the many online news alerts services such as the industry-leading Google News Alerts [2], which generates more than 50% false alarms [10]. As further proof, portals like Yahoo take a more pragmatic approach by requiring all machine generated news alerts to go through a human operator for confirmation before sending them out to subscribers.

Instead of attacking the problem with variations of the same hammer (cosine similarity and TFIDF), a fundamental understanding of the characteristics of news stream data is necessary before any major breakthroughs can be made in TDT. Thus in this paper, we look at news stories and feature trends from the perspective of analyzing a time-series word signal. Previous work like [9] has attempted to reconstruct an event with its representative features. However, in many predictive event detection tasks (i.e., retrospective event detection), there is a vast set of potential features only for a fixed set of observations (i.e., the obvious bursts). Of these features, often only a small number are expected to be useful. In particular, we study the novel problem of *analyzing feature trajectories for event detection*, borrowing a well-known technique from signal processing: identifying distributional correlations among all features by spectral analysis. To evaluate our method, we subsequently propose an unsupervised event detection algorithm for news streams.



(a) aperiodic event (b) periodic event

Figure 1: Feature correlation (DFIDF:time) between a) *Easter* and *April* b) *Unaudited* and *Ended*.

As an illustrative example, consider the correlation between the words *Easter* and *April* from the Reuters Corpus[1]. From the plot of their normalized DFIDF in Figure 1(a), we observe the heavy overlap between the two words *circa* 04/1997, which means they probably both belong to the same event during that time (*Easter feast*). In this example, the hidden event *Easter feast* is a typical important aperiodic event over 1-year data. Another example is given by Figure 1(b), where both the words *Unaudited* and *Ended*

[1]Reuters Corpus is the default dataset for all examples.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.

exhibit similar behaviour over periods of 3 months. These two words actually originated from the same periodic event, *net income-loss reports*, which are released quarterly by publicly listed companies.

Other observations drawn from Figure 1 are: 1) the bursty period of *April* is much longer than *Easter*, which suggests that *April* may exist in other events during the same period; 2) *Unaudited* has a higher average DFIDF value than *Ended*, which indicates *Unaudited* to be more representative for the underlying event. These two examples are but the tip of the iceberg among all word trends and correlations hidden in a news stream like Reuters. If a large number of them can be uncovered, it could significantly aid TDT tasks. In particular, it indicates the significance of mining correlating features for detecting corresponding events. To summarize, we postulate that: 1) An event is described by its representative features. A periodic event has a list of periodic features and an aperiodic event has a list of aperiodic features; 2) Representative features from the same event share similar distributions over time and are highly correlated; 3) An important event has a set of active (largely reported) representative features, whereas an unimportant event has a set of inactive (less-reported) representative features; 4) A feature may be included by several events with overlaps in time frames. Based on these observations, we can either mine representative features given an event or detect an event from a list of highly correlated features. In this paper, we focus on the latter, i.e., how correlated features can be uncovered to form an event in an unsupervised manner.

1.1 Contributions

This paper has three main contributions:

- To the best of our knowledge, our approach is the first to categorize word features for heterogenous events. Specifically, every word feature is categorized into one of the following five feature types based on its power spectrum strength and periodicity: 1) HH (high power and high/long periodicity): important aperiodic events, 2) HL (high power and low periodicity): important periodic events, 3) LH (low power and high periodicity): unimportant aperiodic events, 4) LL (low power and low periodicity): non-events, and 5) SW (stopwords), a higher power and periodicity subset of LL comprising stopwords, which contains no information.
- We propose a simple and effective mixture density-based approach to model and detect feature bursts.
- We come up with an unsupervised event detection algorithm to detect both aperiodic and periodic events. Our algorithm has been evaluated on a real news stream to show its effectiveness.

2. RELATED WORK

This work is largely motivated by a broader family of problems collectively known as Topic Detection and Tracking (TDT) [20, 5, 17, 4, 21, 7, 14, 10]. Moreover, most TDT research so far has been concerned with clustering/classifying documents into topic types, identifying novel sentences [6] for new events, etc., without much regard to analyzing the word trajectory with respect to time. Swan and Allan [18] first attempted using co-occuring terms to construct an event. However, they only considered named entities and noun phrase pairs, without considering their periodicities. On the contrary, our paper considers all of the above.

Recently, there has been significant interest in modeling an event in text streams as a "burst of activities" by incorporating temporal information. Kleinberg's seminal work described how bursty features can be extracted from text streams using an infinite automaton model [12], which inspired a whole series of applications such as Kumar's identification of bursty communities from Weblog graphs [13], Mei's summarization of evolutionary themes in text streams [15], He's clustering of text streams using bursty features [11], etc. Nevertheless, none of the existing work specifically identified features for events, except for Fung et al. [9], who clustered busty features to identify various bursty events. Our work differs from [9] in several ways: 1) we analyze every single feature, not only bursty features; 2) we classify features along two categorical dimensions (periodicity and power), yielding altogether five primary feature types; 3) we do not restrict each feature to exclusively belong to only one event.

Spectral analysis techniques have previously been used by Vlachos et al. [19] to identify periodicities and bursts from query logs. Their focus was on detecting multiple periodicities from the power spectrum graph, which were then used to index words for "query-by-burst" search. In this paper, we use spectral analysis to classify word features along two dimensions, namely periodicity and power spectrum, with the ultimate goal of identifying both periodic and aperiodic bursty events.

3. DATA REPRESENTATION

Let T be the duration/period (in days) of a news stream, and F represents the complete word feature space in the classical static Vector Space Model (VSM).

3.1 Event Periodicity Classification

Within T, there may exist certain events that occur only once, e.g., *Tony Blair elected as Prime Minister of U.K.*, and other recurring events of various periodicities, e.g., *weekly soccer matches*. We thus categorize all events into two types: aperiodic and periodic, defined as follows.

DEFINITION 1. (Aperiodic Event) *An event is aperiodic within T if it only happens once.*

DEFINITION 2. (Periodic Event) *If events of a certain event genre occur regularly with a fixed periodicity $P \leq \lceil T/2 \rceil$, we say that this particular event genre is periodic, with each member event qualified as a periodic event.*

Note that the definition of "aperiodic" is relative, i.e., it is true only for a given T, and may be invalid for any other $T' > T$. For example, the event *Christmas feast* is aperiodic for $T \leq 365$ but periodic for $T \geq 730$.

3.2 Representative Features

Intuitively, an event can be described very concisely by a few discriminative and representative word features and vice-versa, e.g., "hurricane", "sweep", and "strike" could be representative features of a Hurricane genre event. Likewise, a set of strongly correlated features could be used to reconstruct an event description, assuming that strongly correlated features are representative. The representation vector of a word feature is defined as follows:

DEFINITION 3. (Feature Trajectory) *The trajectory of a word feature f can be written as the sequence*

$$y_f = [y_f(1), y_f(2), \ldots, y_f(T)],$$

where each element $y_f(t)$ is a measure of feature f at time t, which could be defined using the normalized DFIDF score[2]

$$y_f(t) = \frac{DF_f(t)}{N(t)} \times log(\frac{N}{DF_f}),$$

where $DF_f(t)$ is the number of documents (local DF) containing feature f at day t, DF_f is the total number of documents (global DF) containing feature f over T, $N(t)$ is the number of documents for day t, and N is the total number of documents over T.

4. IDENTIFYING FEATURES FOR EVENTS

In this section, we show how representative features can be extracted for (un)important or (a)periodic events.

4.1 Spectral Analysis for Dominant Period

Given a feature f, we decompose its feature trajectory $y_f = [y_f(1), y_f(2), ..., y_f(T)]$ into the sequence of T complex numbers $[X_1, \ldots, X_T]$ via the discrete Fourier transform (DFT):

$$X_k = \sum_{t=1}^{T} y_f(t) e^{-\frac{2\pi i}{T}(k-1)t}, \ k = 1, 2, \ldots, T.$$

DFT can represent the original time series as a linear combination of complex sinusoids, which is illustrated by the inverse discrete Fourier transform (IDFT):

$$y_f(t) = \frac{1}{T} \sum_{k=1}^{T} X_k e^{\frac{2\pi i}{T}(k-1)t}, \ t = 1, 2, \ldots, T,$$

where the Fourier coefficient X_k denotes the amplitude of the sinusoid with frequency k/T.

The original trajectory can be reconstructed with just the dominant frequencies, which can be determined from the power spectrum using the popular periodogram estimator. The periodogram is a sequence of the squared magnitude of the Fourier coefficients, $\|X_k\|^2$, $k = 1, 2, \ldots, \lceil T/2 \rceil$, which indicates the signal power at frequency k/T in the spectrum. From the power spectrum, the dominant period is chosen as the inverse of the frequency with the highest power spectrum, as follows.

DEFINITION 4. (Dominant Period) *The dominant period (DP) of a given feature f is $P_f = T/\arg\max_k \|X_k\|^2$.*

Accordingly, we have

DEFINITION 5. (Dominant Power Spectrum) *The dominant power spectrum (DPS) of a given feature f is*

$$S_f = \|X_k\|^2, \ \ with \ \ \|X_k\|^2 \geq \|X_j\|^2, \ \ \forall j \neq k.$$

4.2 Categorizing Features

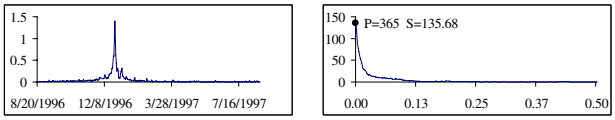
The DPS of a feature trajectory is a strong indicator of its activeness at the specified frequency; the higher the DPS, the more likely for the feature to be bursty. Combining DPS with DP, we therefore categorize all features into four types:

- HH: high S_f, aperiodic or long-term periodic ($P_f > \lceil T/2 \rceil$);
- HL: high S_f, short-term periodic ($P_f \leq \lceil T/2 \rceil$);
- LH: low S_f, aperiodic or long-term periodic;
- LL: low S_f, short-term periodic.

The boundary between long-term and short-term periodic is set to $\lceil T/2 \rceil$. However, distinguishing between a high and low DPS is not straightforward, which will be tackled later.

Properties of Different Feature Sets

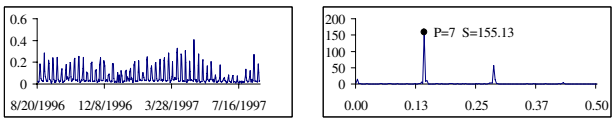
To better understand the properties of HH, HL, LH and LL, we select four features, *Christmas*, *soccer*, *DBS* and *your* as illustrative examples. Since the boundary between high and low power spectrum is unclear, these chosen examples have relative wide range of power spectrum values. Figure 2(a) shows the DFIDF trajectory for *Christmas* with a distinct burst around Christmas day. For the 1-year Reuters dataset, "Christmas" is classified as a typical aperiodic event with $P_f = 365$ and $S_f = 135.68$, as shown in Figure 2(b). Clearly, the value of $S_f = 135.68$ is reasonable for a well-known bursty event like Christmas.



(a) Christmas(DFIDF:time) (b) Christmas(S:frequency)

Figure 2: Feature "Christmas" with relative high S_f and long-term P_f.

The DFIDF trajectory for *soccer* is shown in Figure 3(a), from which we can observe that there is a regular burst every 7 days, which is again verified by its computed value of $P_f = 7$, as shown in Figure 3(b). Using the domain knowledge that soccer games have more matches every Saturday, which makes it a typical and heavily reported periodic event, we thus consider the value of $S_f = 155.13$ to be high.



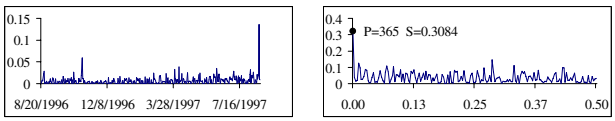
(a) soccer(DFIDF:time) (b) soccer(S:frequency)

Figure 3: Feature "soccer" with relative high S_f and short-term P_f.

From the DFIDF trajectory for *DBS* in Figure 4(a), we can immediately deduce *DBS* to be an infrequent word with a trivial burst on 08/17/1997 corresponding to *DBS Land Raffles Holdings plans*. This is confirmed by the long period of $P_f = 365$ and low power of $S_f = 0.3084$ as shown in Figure 4(b). Moreover, since this aperiodic event is only reported in a few news stories over a very short time of few days, we therefore say that its low power value of $S_f = 0.3084$ is representative of unimportant events.

The most confusing example is shown in Figure 5 for the word feature *your*, which looks very similar to the graph for *soccer* in Figure 3. At first glance, we may be tempted to group both *your* and *soccer* into the same category of HL or LL since both distributions look similar and have the same dominant period of approximately a week. However, further

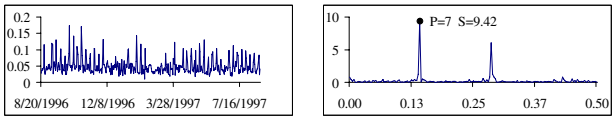
[2]We normalize $y_f(t)$ as $y'_f(t) = y_f(t)/\sum_{i=1}^{T} y_f(i)$ so that it could be interpreted as a probability.



(a) DBS(DFIDF:time) (b) DBS(S:frequency)

Figure 4: Feature "DBS" with relative low S_f and long-term P_f.

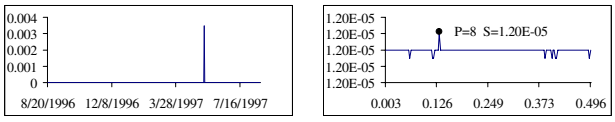
analysis indicates that the periodicity of *your* is due to the differences in document counts for weekdays (average 2,919 per day) and weekends[3] (average 479 per day). One would have expected the "periodicity" of a stopword like *your* to be a day. Moreover, despite our DFIDF normalization, the weekday/weekend imbalance still prevailed; stopwords occur 4 times more frequently on weekends than on weekdays. Thus, the DPS remains the only distinguishing factor between *your* ($S_f = 9.42$) and *soccer* ($S_f = 155.13$). However, it is very dangerous to simply conclude that a power value of $S = 9.42$ corresponds to a stopword feature.



(a) your(DFIDF:time) (b) your(S:frequency)

Figure 5: Feature "your" as an example confusing with feature "soccer".

Before introducing our solution to this problem, let's look at another LL example as shown in Figure 6 for *beenb*, which is actually a confirmed typo. We therefore classify *beenb* as a noisy feature that does not contribute to any event. Clearly, the trajectory of *your* is very different from *beenb*, which means that the former has to be considered separately.



(a) beenb(DFIDF:time) (b) beenb(S:frequency)

Figure 6: Feature "beenb" with relative low S_f and short-term P_f.

Stop Words (SW) Feature Set

Based on the above analysis, we realize that there must be another feature set between HL and LL that corresponds to the set of stopwords. Features from this set has moderate DPS and low but known dominant period. Since it is hard to distinguish this feature set from HL and LL only based on DPS, we introduce another factor called average DFIDF ($\overline{DFIDF}$). As shown in Figure 5, features like *your* usually have a lower DPS than a HL feature like *soccer*, but have a much higher $\overline{DFIDF}$ than another LL noisy feature such as *beenb*. Since such properties are usually characteristics of stopwords, we group features like *your* into the newly defined stopword (SW) feature set.

Since setting the DPS and $\overline{DFIDF}$ thresholds for identifying stopwords is more of an art than science, we proposed a heuristic HS algorithm, Algorithm 1. The basic idea is to only use news stories from weekdays to identify stopwords.

[3]The "weekends" here also include public holidays falling on weekdays.

The SW set is initially seeded with a small set of 29 popular stopwords utilized by Google search engine.

Algorithm 1 Heuristic Stopwords detection (HS)

Input: Seed SW set, weekday trajectories of all words

1: From the seed set SW, compute the maximum DPS as UDPS, maximum $\overline{DFIDF}$ as UDFIDF, and minimum of $\overline{DFIDF}$ as LDFIDF.
2: **for** $f_i \in F$ **do**
3: Compute DFT for f_i.
4: **if** $S_{f_i} \leq UDPS$ and $\overline{DFIDF_{f_i}} \in [LDFIDF, UDFIDF]$ **then**
5: $f_i \rightarrow SW$
6: $F = F - f_i$
7: **end if**
8: **end for**

Overview of Feature Categorization

After the SW set is generated, all stopwords are removed from F. We then set the boundary between high and low DPS to be the upper bound of the SW set's DPS. An overview of all five feature sets is shown in Figure 7.

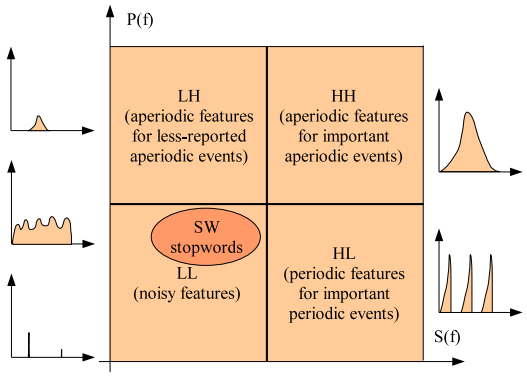


Figure 7: The 5 feature sets for events.

5. IDENTIFYING BURSTS FOR FEATURES

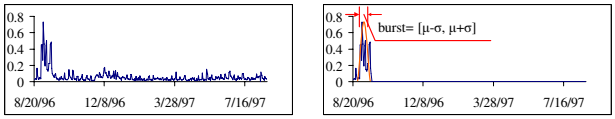
Since only features from HH, HL and LH are meaningful and could potentially be representative to some events, we pruned all other feature classified as LL or SW. In this section, we describe how bursts can be identified from the remaining features. Unlike Kleinberg's burst identification algorithm [12], we can identify both significant and trivial bursts without the need to set any parameters.

5.1 Detecting Aperiodic Features' Bursts

For each feature in HH and HL, we truncate its trajectory by keeping only the bursty period, which is modeled with a Gaussian distribution. For example, Figure 8 shows the word feature *Iraq* with a burst *circa* 09/06/1996 being modeled as a Gaussian. Its bursty period is defined by $[\mu_f - \sigma_f, \mu_f + \sigma_f]$ as shown in Figure 8(b).

5.2 Detecting Periodic Features' Bursts

Since we have computed the DP for a periodic feature f, we can easily model its periodic feature trajectory y_f using



(a) original DFIDF:time (b) identifying burst

Figure 8: Modeling *Iraq*'s time series as a truncated Gaussian with $\mu = 09/06/1996$ and $\sigma = 6.26$.

a mixture of $K = \lfloor T/P_f \rfloor$ Gaussians:

$$f(y_f = y_f(t)|\theta_f) = \sum_{k=1}^{K} \alpha_k \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{1}{2\sigma_k^2}(y_f(t)-\mu_k)^2},$$

where the parameter set $\theta_f = \{\alpha_k, \mu_k, \sigma_k\}_{k=1}^{K}$ comprises:

- α_k is the probability of assigning y_f into the k^{th} Gaussian. $\alpha_k > 0$, $\forall k \in [1, K]$ and $\sum_{k=1}^{K} \alpha_k = 1$;
- μ_k/σ_k is mean/standard deviation of the k^{th} Gaussian.

The well known Expectation Maximization (EM) [8] algorithm is used to compute the mixing proportions α_k, as well as the individual Gaussian density parameters μ_k and σ_K. Each Gaussian represents one periodic event, and is modeled similarly as mentioned in Section 5.1.

6. EVENTS FROM FEATURES

After identifying and modeling bursts for all features, the next task is to paint a picture of the event with a potential set of representative features.

6.1 Feature Correlation

If two features f_i and f_j are representative of the same event, they must satisfy the following necessary conditions:

1. f_i and f_j are identically distributed: $y_{f_i} \sim y_{f_j}$.
2. f_i and f_j have a high document overlap.

Measuring Feature Distribution Similarity

We measure the similarity between two features f_i and f_j using discrete KL-divergence defined as follows.

DEFINITION 6. (feature similarity) *$KL(f_i, f_j)$ is given by* $\max(KL(f_i|f_j), KL(f_j|f_i))$, *where*

$$KL(f_i|f_j) = \sum_{t=1}^{T} f(y_{f_i}(t)|\theta_{f_i}) log \frac{f(y_{f_i}(t)|\theta_{f_i})}{f(y_{f_j}(t)|\theta_{f_j})}. \quad (1)$$

Since KL-divergence is not symmetric, we define the similarity between between f_i and f_j as the maximum of $KL(f_i|f_j)$ and $KL(f_j|f_i)$. Further, the similarity between two aperiodic features can be computed using a closed form of the KL-divergence [16]. The same discrete KL-divergence formula of Eq. 1 is employed to compute the similarity between two periodic features,

Next, we define the overal similarity among a set of features R using the maximum inter-feature KL-Divergence value as follows.

DEFINITION 7. (set's similarity) $KL(R) = \max_{\forall f_i, f_j \in R} KL(f_i, f_j)$.

Document Overlap

Let M_i be the set of all documents containing feature f_i. Given two features f_i and f_j, the overlapping document set containing both features is $M_i \cap M_j$. Intuitively, the higher the $|M_i \cap M_j|$, the more likelty that f_i and f_j will be highly correlated. We define the degree of document overlap between two features f_i and f_j as follows.

DEFINITION 8. (Feature DF Overlap) $d(f_i, f_j) = \frac{|M_i \cap M_j|}{min(|M_i|, |M_j|)}$.

Accordingly, the DF Overlap among a set of features R is also defined.

DEFINITION 9. (Set DF Overlap) $d(R) = \min_{\forall f_i, f_j \in R} d(f_i, f_j)$.

6.2 Unsupervised Greedy Event Detection

We use features from HH to detect important aperiodic events, features from LH to detect less-reported/unimportant aperiodic events, and features from HL to detect periodic events. All of them share the same algorithm. Given bursty feature $f_i \in HH$, the goal is to find highly correlated features from HH. The set of features similar to f_i can then collectively describe an event. Specifically, we need to find a subset R_i of HH that minimizes the following cost function:

$$C(R_i) = \frac{KL(R_i)}{d(R_i) \sum_{f_j \in R_i} S_{f_j}}, \; R_i \subset HH. \quad (2)$$

The underlying event e (associated with the burst of f_i) can be represented by R_i as

$$y(e) = \sum_{f_j \in R_i} \frac{S_{f_j}}{\sum_{f_u \in R_i} S_{f_u}} y_{f_j}. \quad (3)$$

The burst analysis for event e is exactly the same as the feature trajectory.

The cost in Eq. 2 can be minimized using our unsupervised greedy UG event detection algorithm, which is described in Algorithm 2. The UG algorithm allows a feature to be contained in multiple events so that we can detect several events happening at the same time. Furthermore, trivial events only containing *year*/*month* features (i.e., an event

Algorithm 2 Unsupervised Greedy event detection (UG).

Input: HH, document index for each feature.

```
Sort and select features in descending DPS order: S_{f_1} ≥ S_{f_2} ≥ ... ≥ S_{f_|HH|}.
k = 0.
for f_i ∈ HH do
    k = k + 1.
    Init: R_i ← f_i, C(R_i) = 1/S_{f_i} and HH = HH − f_i.
    while HH not empty do
        m = arg min_m C(R_i ∪ f_m).
        if C(R_i ∪ f_m) < C(R_i) then
            R_i ← f_m and HH = HH − f_m.
        else
            break while.
        end if
    end while
    Output e_k as Eq. 3.
end for
```

only containing 1 feature *Aug* could be identified over a 1-year news stream) could be removed, although such events will have inherent high cost and should already be ranked very low. Note that our UG algorithm only requires one data-dependant parameter, the boundary between high and low power spectrum, to be set once, and this parameter can be easily estimated using the HS algorithm (Algorithm 1).

7. EXPERIMENTS

In this section, we study the performances of our feature categorizing method and event detection algorithm. We first introduce the dataset and experimental setup, then we subjectively evaluate the categorization of features for HH, HL, LH, LL and SW. Finally, we study the (a)periodic event detection problem with Algorithm 2.

7.1 Dataset and Experimental Setup

The Reuters Corpus contains 806,791 English news stories from 08/20/1996 to 08/19/1997 at a day resolution. Version 2 of the open source Lucene software [1] was used to tokenize the news text content and generate the document-word vector. In order to preserve the time-sensitive past/present/future tenses of verbs and the differences between lower case nouns and upper case named entities, no stemming was done. Since dynamic stopword removal is one of the functionalities of our method, no stopword was removed. We did remove non-English characters, however, after which the number of word features amounts to 423,433. All experiments were implemented in Java and conducted on a 3.2 GHz Pentium 4 PC running Windows 2003 Server with 1 GB of memory.

7.2 Categorizing Features

We downloaded 34 well-known stopwords utilized by the Google search engine as our seed training features, which includes *a, about, an, are, as, at, be, by, de, for, from, how, in, is, it, of, on, or, that, the, this, to, was, what, when, where, who, will, with, la, com, und, en* and *www*. We excluded the last five stopwords as they are uncommon in news stories. By only analyzing news stories over 259 weekdays, we computed the upper bound of the power spectrum for stopwords at 11.18 and corresponding $\overline{DFIDF}$ ranges from 0.1182 to 0.3691. Any feature f satisfying $S_f <= 11.18$ and $0.1182 <= \overline{DFIDF_f} <= 0.3691$ over weekdays will be considered a stopword. In this manner, 470 stopwords were found and removed as visualized in Figure 9. Some detected stopwords are *A* ($P = 65$, $S = 3.36$, $\overline{DFIDF} = 0.3103$), *At* ($P = 259$, $S = 1.86$, $\overline{DFIDF} = 0.1551$), *GMT* ($P = 130$, $S = 6.16$, $\overline{DFIDF} = 0.1628$) and *much* ($P = 22$, $S = 0.80$, $\overline{DFIDF} = 0.1865$). After the removal of these stopwords, the distribution of weekday and weekend news are more or less matched, and in the ensuing experiments, we shall make use of the full corpus (weekdays and weekends).

The upper bound power spectrum value of 11.18 for stopwords training was selected as the boundary between the high power and low power spectrum. The boundary between high and low periodicity was set to $\lceil 365/2 \rceil = 183$. All 422,963 ($423433 - 470$) word features were categorized into 4 feature sets: HH (69 features), HL (1,087 features), LH (83,471 features), and LL (338,806 features) as shown in Figure 10. In Figure 10, each gray level denotes the relative density of features in a square region, measured by $log_{10}(1 + D_k)$, where D_k is the number of features within the k-th square region. From the figure, we can make the

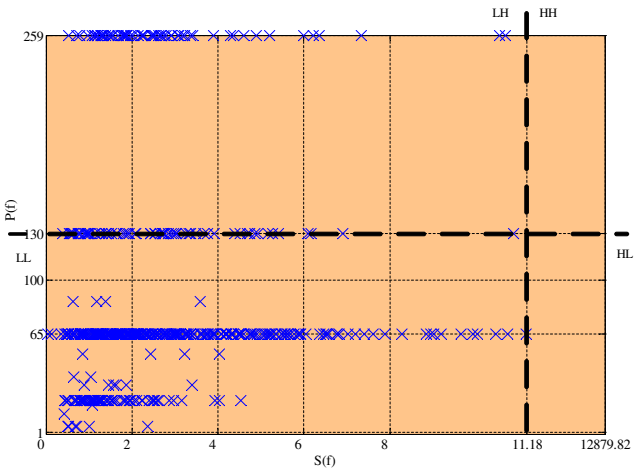


Figure 9: Distribution of SW (stopwords) in the HH, HL, LH, and LL regions.

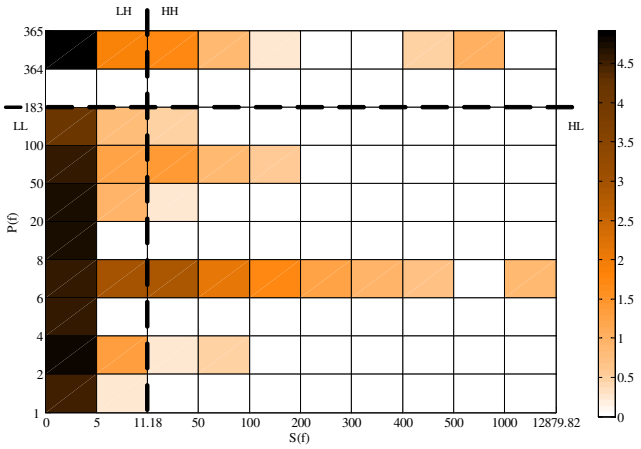


Figure 10: Distribution of categorized features over the four quadrants (shading in log scale).

following observations:

1. Most features have low S and are easily distinguishable from those features having a much higher S, which allows us to detect important (a)periodic events from trivial events by selecting features with high S.

2. Features in the HH and LH quadrants are aperiodic, which are nicely separated (big horizontal gap) from the periodic features. This allows reliably detecting aperiodic events and periodic events independently.

3. The (vertical) boundary between high and low power spectrum is not as clearcut and the exact value will be application specific.

By checking the scatter distribution of features from SW on HH, HL, LH, and LL as shown in Figure 9, we found that 87.02%(409/470) of the detected stopwords originated from LL. The LL classification and high $\overline{DFIDF}$ scores of stopwords agree with the generally accepted notion that stopwords are equally frequent over all time. Therefore, setting the boundary between high and low power spectrum using the upper bound S_f of SW is a reasonable heuristic.

7.3 Detecting Aperiodic Events

We shall evaluate our two hypotheses, 1)important aperiodic events can be defined by a set of HH features, and 2)less reported aperiodic events can be defined by a set of LH features. Since no benchmark news streams exist for event detection (TDT datasets are not proper streams), we evaluate the quality of the automatically detected events by comparing them to manually-confirmed events by searching through the corpus.

Among the 69 HH features, we detected 17 important aperiodic events as shown in Table 1 ($e_1 - e_{17}$). Note that the entire identification took less than 1 second, after removing events containing only the *month* feature. Among the 17 events, other than the overlaps between e_3 and e_4 (both describes the same hostage event), e_{11} and e_{16} (both about company reports), the 14 identified events are extremely accurate and correspond very well to the major events of the period. For example, the defeat of Bob Dole, election of Tony Blair, Missile attack on Iraq, etc. Recall that selecting the features for one event should minimize the cost in Eq. 2 such that 1) the number of features span different events, and 2) not all features relevant to an event will be selected, e.g., the feature *Clinton* is representative to e_{12} but since *Clinton* relates to many other events, its time domain signal is far different from those of other representative features like *Dole* and *Bob*. The number of documents of a detected event is roughly estimated by the number of indexed documents containing the representative features. We can see that all 17 important aperiodic events are popularly reported events.

After 742 minutes of computation time, we detected 23, 525 less reported aperiodic events from 83,471 LH features. Table 1 lists the top 5 detected aperiodic events ($e_{18} - e_{22}$) with respect to the cost. We found that these 5 events are actually very trivial events with only a few news reports, and are usually subsumed by some larger topics. For example, e_{22} is one of the rescue events in an airplane hijack topic. One advantage of our UG Algorithm for discovering less-reported aperiodic events is that we are able to precisely detect the true event period.

7.4 Detecting Periodic Events

Among the 1,087 HL features, 330 important periodic events were detected within 10 minutes of computing time. Table 1 lists the top 5 detected periodic events with respect to the cost ($e_{23} - e_{27}$). All of the detected periodic events are indeed valid, and correspond to real life periodic events. The GMM model is able to detect and estimate the bursty period nicely although it cannot distinguish the slight difference between every Monday-Friday and all weekdays as shown in e_{23}. We also notice that e_{26} is actually a subset of e_{27} (soccer game), which is acceptable since the Sheffield league results are announced independently every weekend.

8. CONCLUSIONS

This paper took a whole new perspective of analyzing feature trajectories as time domain signals. By considering the word document frequencies in both time and frequency domains, we were able to derive many new characteristics about news streams that were previously unknown, e.g., the different distributions of stopwords during weekdays and weekends. For the first time in the area of TDT, we applied a systematic approach to automatically detect important and less-reported, periodic and aperiodic events.

The key idea of our work lies in the observations that (a)periodic events have (a)periodic representative features and (un)important events have (in)active representative features, differentiated by their power spectrums and time periods. To address the real event detection problem, a simple and effective mixture density-based approach was used to identify feature bursts and their associated bursty periods. We also designed an unsupervised greedy algorithm to detect both aperiodic and periodic events, which was successful in detecting real events as shown in the evaluation on a real news stream.

Although we have not made any benchmark comparison against another approach, simply because there is no previous work in the addressed problem. Future work includes evaluating the recall of detected events for a labeled news stream, and comparing our model against the closest equivalent methods, which currently are limited to the methods of Kleinberg [12] (which can only detect certain type of bursty events depending on parameter settings), Fung et al. [9], and Swan and Allan [18]. Nevertheless, we believe our simple and effective method will be useful for all TDT practitioners, and will be especially useful for the initial exploratory analysis of news streams.

9. REFERENCES

[1] Apache lucene-core 2.0.0, http://lucene.apache.org.

[2] Google news alerts, http://www.google.com/alerts.

[3] Reuters corpus, http://www.reuters.com/researchandstandards/corpus/.

[4] J. Allan. *Topic Detection and Tracking. Event-based Information Organization*. Kluwer Academic Publishers, 2002.

[5] J. Allan, V. Lavrenko, and H. Jin. First story detection in tdt is hard. In *CIKM*, pages 374–381, 2000.

[6] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *SIGIR*, pages 314–321, 2003.

[7] T. Brants, F. Chen, and A. Farahat. A system for new event detection. In *SIGIR*, pages 330–337, 2003.

[8] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[9] G. P. C. Fung, J. X. Yu, P. S. Yu, and H. Lu. Parameter free bursty events detection in text streams. In *VLDB*, pages 181–192, 2005.

[10] Q. He, K. Chang, and E.-P. Lim. A model for anticipatory event detection. In *ER*, pages 168–181, 2006.

[11] Q. He, K. Chang, E.-P. Lim, and J. Zhang. Bursty feature reprensentation for clustering text streams. In *SDM, accepted*, 2007.

[12] J. Kleinberg. Bursty and hierarchical structure in streams. In *SIGKDD*, pages 91–101, 2002.

[13] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. In *WWW*, pages 159–178, 2005.

[14] G. Kumaran and J. Allan. Text classification and named entities for new event detection. In *SIGIR*, pages 297–304, 2004.

[15] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *SIGKDD*, pages 198–207, 2005.

[16] W. D. Penny. Kullback-liebler divergences of normal, gamma, dirichlet and wishart densities. *Technical report*, 2001.

[17] N. Stokes and J. Carthy. Combining semantic and syntactic document classifiers to improve first story detection. In *SIGIR*, pages 424–425, 2001.

[18] R. Swan and J. Allan. Automatic generation of overview timelines. In *SIGIR*, pages 49–56, 2000.

[19] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *SIGMOD*, pages 131–142, 2004.

[20] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *SIGIR*, pages 28–36, 1998.

[21] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *SIGKDD*, pages 688–693, 2002.

Table 1: All important aperiodic events ($e_1 - e_{17}$), top 5 less-reported aperiodic events ($e_{18} - e_{22}$) and top 5 important periodic events ($e_{23} - e_{27}$).

Detected Event and Bursty Period	Doc #	True Event
e_1(Sali,Berisha,Albania,Albanian,March) 02/02/1997-05/29/1997	1409	Albanian's president Sali Berisha lost in an early election and resigned, 12/1996-07/1997.
e_2(Seko,Mobutu,Sese,Kabila) 03/22/1997-06/09/1997	2273	Zaire's president Mobutu Sese coordinated the native rebellion and failed on 05/16/1997.
e_3(Marxist,Peruvian) 11/19/1996-03/05/1997	824	Peru rebels (Tupac Amaru revolutionary Movement) led a hostage siege in Lima in early 1997.
e_4(Movement,Tupac,Amaru,Lima,hostage,hostages) 11/16/1996-03/20/1997	824	The same as e_3.
e_5(Kinshasa,Kabila,Laurent,Congo) 03/26/1997-06/15/1997	1378	Zaire was renamed the Democratic Republic of Congo on 05/16/1997.
e_6(Jospin,Lionel,June) 05/10/1997-07/09/1997	605	Following the early General Elections circa 06/1997, Lionel Jospin was appointed Prime Minister on 06/02/1997.
e_7(Iraq,missile) 08/31/1996-09/13/1996	1262	U.S. fired missile at Iraq on 09/03/1996 and 09/04/1996.
e_8(Kurdish,Baghdad,Iraqi) 08/29/1996-09/09/1996	1132	Iraqi troop fought with Kurdish faction circa 09/1996.
e_9(May,Blair) 03/24/1997-07/04/1997	1049	Tony Blair became the Primary Minister of the United Kingdom on 05/02/1997.
e_{10}(slalom,skiing) 12/05/1996-03/21/1997	253	Slalom Game of Alpine Skiing in 01/1997-02/1997.
e_{11}(Interim,months) 09/24/1996-12/31/1996	3063	Tokyo released company interim results for the past several months in 09/1996-12/1996.
e_{12}(Dole,Bob) 09/09/1996-11/24/1996	1599	Dole Bob lost the 1996 US presidential election.
e_{13}(July,Sen) 06/25/1997-06/25/1997	344	Cambodia's Prime Minister Hun Sen launched a bloody military coup in 07/1997.
e_{14}(Hebron) 10/15/1996-02/14/1997	2098	Hebron was divided into two sectors in early 1997.
e_{15}(April,Easter) 02/23/1997-05/04/1997	480	Easter feasts circa 04/1997 (for western and Orthodox).
e_{16}(Diluted,Group) 04/27/1997-07/20/1997	1888	Tokyo released all 96/97 group results in 04/1997-07/1997.
e_{17}(December,Christmas) 11/17/1996-01/26/1997	1326	Christmas feast in late 12/1997.
e_{18}(Kolaceva,winter,Together,promenades,Zajedno, Slobodan,Belgrade,Serbian,Serbia,Draskovic,municipal, Kragujevac) 1/25/1997	3	University students organized a vigil on Kolaceva street against government on 1/25/1997.
e_{19}(Tutsi,Luvengi,Burundi,Uvira,fuel,Banyamulenge, Burundian,Kivu,Kiliba,Runingo,Kagunga,Bwegera) 10/19/1996	6	Fresh fighting erupted around Uvira between Zaire armed forces and Banyamulengs Tutsi rebels on 10/19/1996.
e_{20}(Malantacchi,Korea,Guy,Rider,Unions,labour, Trade,unions,Confederation,rammed,Geneva,stoppages, Virgin,hire,Myongdong,Metalworkers) 1/11/1997	2	Marcello Malantacchi secretary general of the International Metalworkers Federation and Guy Rider who heads the Geneva office of the International Confederation of Free Trade Unions attacked the new labour law of South Korea on 1/11/1997.
e_{21}(DBS,Raffles) 8/17/1997	9	The list of the unit of Singapore DBS Land Raffles Holdings plans on 8/17/1997.
e_{22}(preserver,fuel,Galawa,Huddle,Leul,Beausse) 11/24/1996	3	Rescued a woman and her baby during a hijacked Ethiopian plane that ran out of fuel and crashed into the sea near Le Galawa beach on 11/24/1996.
e_{23}(PRICE,LISTING,MLN,MATURITY,COUPON, MOODY,AMT,FIRST,ISS,TYPE,PAY,BORROWER) Monday-Friday/week	7966	Announce bond price on all weekdays.
e_{24}(Unaudited,Ended,Months,Weighted,Provision,Cost, Selling,Revenues,Loss,Income,except,Shrs,Revs) every season	2264	Net income-loss reports released by companies in every season.
e_{25}(rating,Wall,Street,Ian) Monday-Friday/week	21767	Stock reports from Wall Street on all weekdays.
e_{26}(Sheffield,league,scoring,goals,striker,games) every Friday, Saturday and Sunday	574	Match results of Sheffield soccer league were published on Friday, Saturday and Sunday 10 times than other 4 days.
e_{27}(soccer,matches,Results,season,game,Cup,match, victory,beat,played,play,division) every Friday, Saturday and Sunday	2396	Soccer games held on Friday, Saturday and Sunday 7 times than other 4 days.