Identifying Evolutionary Topic Temporal Patterns Based on Bursty Phrase Clustering

Yixuan Liu[1], Zihao Gao[2] and Mizuho Iwaihara[3]

[1,2,3] Graduate School of Information, Production and Systems, Waseda University, Fukuoka 808-0135, Japan
liuyixuan@ruri.waseda.jp,
gao_jihao@suou.waseda.jp,iwaihara@waseda.jp

Abstract. We discuss a temporal text mining task on finding evolutionary patterns of topics from a collection of article revisions. To reveal the evolution of topics, we propose a novel method for finding key phrases that are bursty and significant in terms of revision histories. Then we show a time series clustering method to group phrases that have similar burst histories, where additions and deletions are separately considered, and time series is abstracted by burst detection. In clustering, we use dynamic time warping to measure the distance between time sequences of phrase frequencies. Experimental results show that our method clusters phrases into groups that actually share similar bursts which can be explained by real-world events.

Keywords: Topic evolution, Temporal pattern, Burst detection, DTW, Clustering

1 Introduction

Over the past decade, numerous online collaboration systems have appeared and thrived on the Internet. Prime examples include Wikipedia, Yahoo! Answers, Mechanical Turkbased systems [3]. They enlist a large number of people to supply information and solve problems. Users become the main strength of contributors. Wikipedia is an open, multilingual Internet encyclopedia written collaboratively by volunteers around the world [11], and end users can also edit articles. Each edit revision is saved and all the revisions are available to the public. We can utilize the revision history to discover trends of topics.

In this paper, we particularly focus on how representative phrases change their frequencies in revisions, to find whether bursty edits occur phrases. We further present a method for clustering phrases by similarity on bursty time series, where we expect that topics in one cluster share similar temporal patterns in edit history. One cluster may contain multiple real-world events which are related each other. Such fine-grained temporal relationship is difficult to be found on topic models over a static corpus or a temporal series of corpora.

Unlike traditional text clustering works, we solve the problem by focusing on the changes of phrase frequencies in revisions. We also discuss extraction of candidate phrases that have significant temporal features, where additions and deletions of a phrase are separately evaluated.

In Wikipedia edit history, articles of events or persons are edited over years, and edits of articles can be bursty or gentle, and the peak time can be shifted. Therefore we need to adopt a flexible function to measure temporal similarities between phrases. Major contributions of this paper are summarized below:

- We define a number of scoring functions to find candidate phrases that are having significant temporal features, and define the cumulative phrase frequency, which is effective to obtain time series of edit activities of a phrase in articles.
- We apply burst detection on the time series of each phrase to reduce minor details and simply the time series. Then we apply k-means to cluster phrases, where similarity is defined by burst patterns, and dynamic time warping (DTW) [2] is utilized.

2 Related Work

Kalogeratos et al. [5] proposed an algorithm to improving text clustering algorithm by term coburstiness. The algorithm first constructs a bursty term correlation graph, then applying graph partitioning technique to find clusters based on bursts and inter-burst relationships. The fundamental problem in [5] is that it is irrespective of bursts of edits. Kleinberg [6] proposed a burst model where a burst is defined as a rapid increase of a term's frequency of occurrences. If the term frequency is encountered at an unusual high rate, then the term is labeled as 'bursty'. The work is used to identify bursts in text stream and produce state labels of bursts. Tran et al [10] engaged in temporal text mining domain and proposed to represent an event by entities, which is instructive for event representation during burst event detection in Wikipedia. However, though a number of research activities on burst, it is difficult to compare methods because of a lack of common procedures today. Subašic [9] build up an evaluation system for temporal text mining methods, which makes it possible to measure the effectiveness of temporal text mining for news.

3 Selecting historically significant phrases

To find evolution of topics in a Wikipedia category, we need to monitor a relatively long period of edit history. We utilize key phrases to represent candidate topics. We discuss how to detect phrases that have bursty surge of edits in an article and a collection of articles. Wikipedia articles are edited repeatedly to reflect a chain of new events. Phrases occurring in such burst edits shall be detected as bursty phrases. We discuss selecting phrases that are semantically representative and having significant edit activities in their history.

3.1 POS Tagging and filtering

In order to cluster shifty topics, we first detect key phrases that could represent topics in one article. In this step, POS (part of speech) Tagger and Chunker provided by Apache openNLP [4] is used to apply POS tagging and chunking function upon the revisions of Wikipedia articles, to produce various POS labels to each chunk.

3.2 Decay phrase frequency

Next we introduce a number of temporal features to refine candidate phrases. Revisions of one article are created over time, where various phrases are added or deleted, and phrase frequencies, namely the times one phrase occur in one revision, dynamically change over time. Traditional TF-IDF does not capture such temporal factors of document revisions.

In order to give weights to phrases through the whole edit history, one way is to compute the term frequency with a decay factor, like Aji [1]. However, their model ignores revision quality, and decay factor is applied based on revision counts, instead of time interval. We should give a higher weight to phrases appearing in long-lived revisions because that long-lived revisions are accepted by editors as trustworthy and high-quality. Let $t(r)$ denote the time point of revision r and $f(p,r_i)$ denote the frequency of phrase p in i^{th} revision. We propose our timespan-weighted phrase frequency over history as:

$$PFH = \frac{f(p,r_i)}{i^{\rho}} \cdot \frac{t(r_{i+1}) - t(r_i)}{t(r_n) - t(r_1)} \quad (1)$$

Here, $\rho > 0$ is a decay factor, and r_i is the i^{th} revision of the article, $i=1,..,n$.

We evaluate how widely a term occurs in the considered article set as below:

$$rf(p, D) = \frac{|r \in D : p \in r| + 1}{N} \quad (2)$$

Here, N is the total number of articles in the corpus D, $|r \in D : p \in r|$ is the number of revisions in which phrase p occurs. In case p is not in the corpus, we add one to the numerator to avoid division by zero.

3.3 Survival rate

We define the survival rate of a phrase p in an article a as:

$$SR(p) = e^{\frac{scale(p)}{|R|}} \cdot \frac{contain(p)}{scale(p)} \quad (3)$$

Here, $|R|$ is the number of revisions of article a, $scale(p)$ is the number of revisions of article a in the period between p appears first and p appears last, and $contain(p)$ is the number of revisions of article a containing p. The first part measures how many revisions the phrase survived over the history, and the second part measures how long

the phrase appears without interruptions during its lifespan. The survival rate is independent from text length. Phrases having long and non-interrupting lifespans are highly scored by the survival rate.

Combining the features we proposed so far, we define the following weight function for selecting historically-significant phrases:

$$W_{p_i} = PFH(p_i, R_x) \cdot SR(p_i) \cdot rf(p_i, R_x) \tag{4}$$

4 Abstracting time series of bursty phrases

4.1 Time series modeling

Let $S = [s_1, \ldots, s_n]$ be a corpus, which is a sequence of target articles $s_1, \ldots, s_n$. The corpus can be chosen based on certain topics, such as from a Wikipedia category.

The *revision frequency* of a phrase is the times the phrase occurs in one revision. As new revisions are created over time, certain phrases increase their revision frequency, while others keep stable, or decrease, and sometimes fluctuate. To detect edit activities, we should focus only on changes of revision frequency, which is caused by additions and deletions of a phrase. Let r_i and r_{i-1} be two adjacent revisions of an identical article. The frequency difference of phrase p between revision r_i and r_{i-1} is given by:

$$\delta(p, r_i) = pf(p, r_i) - pf(p, r_{i-1}) \tag{5}$$

Here *pf* is the revision frequency of phrase p in the article. When $\delta(p, r_i) > 0$ holds, additions of p are more than deletions of p in creating revision r_i from r_{i-1}.

Usually, the length of one articles grows over time, and the frequency of a phrase increases as well. On the other hand, deleting a phrase p can happen when p is overwritten by new contents, due to obsoleteness, error corrections, new facts, new concepts, etc. Also, edit fights between editors can cause deletions of a phrase. Furthermore, a real-world event can trigger a rush of edits, causing fluctuations of phrase frequencies. Thus additions and deletions of a phrase indicate distinct phenomena, so that we should introduce a burst detection which treats additions and deletions separately. We define the adding effect af of phrase p from revision ri to revision rj as:

$$af(p, r_i, r_j) = \begin{cases} \delta(p, r_i) \cdot e^{-\lambda_1 (j-i)}, & \delta(p, r_i) > 0 \ \ and \ \ j > i \\ 0, & \delta(p, r_i) \leq 0 \ \ or \ \ j \leq i \end{cases} \tag{6}$$

We also define the *deleting effect df* of phrase p from revision r_i to revision r_j as:

$$df(p, r_i, r_j) = \begin{cases} \delta(p, r_i) \cdot e^{-\lambda_2 (i-j)}, & \delta(p, r_i) < 0 \ \ and \ \ j < i \\ 0, & \delta(p, r_i) \geq 0 \ \ or \ \ j \geq i \end{cases} \tag{7}$$

Here, we assume that additions and deletions of preceding revisions i are affecting revision j with an exponential decay, based on revision number difference j-i, with the decay parameters $\lambda_1 > 0$ and $\lambda_2 > 0$. By having these separate decay parameters, we can control relative weighting between adding and deleting revisions over a phrase.

Finally, we define *cumulative phrase frequency Pf*(p, r_j) below, which sums up adding and deleting effects of revisions before j.

$$\begin{gathered} PF(p,S) = [Pf(p,r_{11}), \ldots, Pf(p,r_{nm})] \\ Pf(p,r_j) = \delta(p,r_{11}) + \sum_{i=1}^{n}[af(p,r_i,r_j) + df(p,r_i,r_j)] \end{gathered} \tag{8}$$

Here, r_{mn} is the n^{th} revision of the m^{th} article in corpus S. We take the sum of cumulative phrase frequencies for every article containing p and for every revision in a time unit of one week, into one element $Pf(p, r_{ij})$, and construct the time series $PF(p, S)$ for temporal clustering.

4.2 Burst detection and time series abstraction

Cumulative phrase frequency captures trends of edit activities, but its time series $PF(p, S)$ still contains noises and spikes, which makes difficult to cluster similar edit histories. To overcome this problem, we apply Kleinberg's burst detection algorithm [6], to convert the cumulative phrase frequencies into burst levels. The burst detection algorithm finds a state sequence where each symbol b_i corresponds to a burst level of non-negative integers. After burst detection, we obtain a time series $TS_p = [(s_1, x_1), \ldots, (s_n, x_n)]$ on phase p, where s_i is the burst level at time point x_i. After this time series abstraction, we can easily detect bursts co-occurring in phrases.

4.3 Temporal clustering by dynamic time warping measure

In order to uncover the temporal dynamics of key phrases in Wikipedia, we apply the temporal clustering method dynamic time warping (DTW) [1] to the abstracted time series of edit histories. Since the time sequences on phrases have sparse and random bursts, and their durations are diverse, time shift on peaks of frequencies is necessary. The classical Euclidean distance metric which compares at exact time points is not suitable in this situation.

For a pair of phrases p and q, let TS_1 and TS_2 be their time sequences of cumulative phrase frequencies. We define the distance $d(i, j)$ between the i^{th} component (s_{1i}, x_{1i}) of TS_1 and the j^{th} component (s_{2j}, x_{2j}) of TS_2 as $d(i, j) = \sqrt{((s_{1i} - s_{2j})^2 + (x_{1i} - x_{2j})^2}$. Based on the local cost matrix by $d(i, j)$, dynamic time warping paths are calculated by dynamic programming, which yield DTW distances. We carry out k-means clustering [12], to cluster phrases having similar burst patterns. Since burst positions and durations are varying, in k-means clustering we adopt the DTW metric, instead of the Euclidean metric.

5 Experiments

To confirm the effect of cumulative phrase frequency, we present in Figure 1 the result on phrase "airstrikes," which is in the candidate phrases selected by the weighting function in Section 3, from article "American-led intervention in Syria."

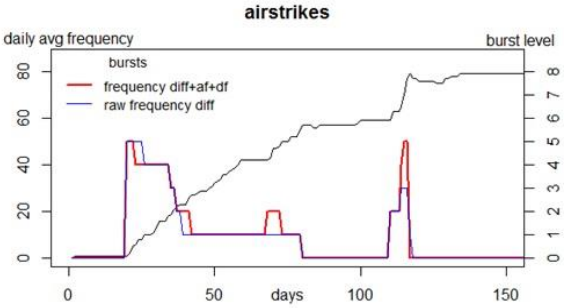

Fig. 1. Detected bursts of "airstrikes" with both effects in "American-led intervention in Syria"

The red curve in Figure 1 shows burst levels detected on cumulative phrase frequencies, where additions and deletions are evaluated separately with decay parameters, while the blue curve shows burst levels detected on changes of raw phrase frequencies. We can find that the red curve has more detailed burst levels and burst intervals, responding to sudden changes of phrase frequencies. On the other hand, although the blue curve has reduced noises, a number of bursts are not detected. Note that we can control the burst sensitivity by changing the decay parameters of additions and deletions.

Table 1. Sample of dataset articles in experiments

Title	Time span
Barack Obama	2004/9/30-2016/2/13
Democratic Party (United States)	2001/9/21-2014/12/4
Donald Trump presidential campaign, 2016	2014/8/4-2016/3/22
Donald Trump	2011/8/28-2016/3/3
Hillary Clinton	2007/1/24-2016/5/6
United States Presidential election, 2016	2014/9/28-2016/3/12
United States Presidential election, 2012	2010/6/15-2014/7/19
United States Presidential election, 2008	2004/12/15-2011/6/7
Republican Party (United States)	2010/6/14-2015/10/7
Republican Party presidential primaries, 2016	2010/6/14-2015/10/7

Table 2. Clustering results on phrases by cummlative phrase frequencies

Cluster	Phrases	Number of phrases
1	Trump, Rand Paul, Ted Cruz, his campaign, reuters	5
2	Illegal immigrant, ISIS, Syria, Mac Cain…	40
3	Washington post, Huffington post, the United States, news max, fox news, blogs…	9
4	Actors, campaign, crime…	46

In order to evaluate our proposing method, we select ten different categories in Wikipedia. In each category, articles are filtered by the following criteria: English language, having more than 200 revisions.

After filtering, we manually review the articles and remove articles with repeated contents or irrelevant contents. Finally, in each category 10 articles are filtered as datasets. A subset is shown in **Table 1**. Then we retrieved all the revisions of each article.

We selected historically significant phrases by the method described in Section 3, and produced time sequences by their cumulative phrase frequencies. After simplifying the time sequences by burst detection, temporal clustering was carried out.

Fig. 2 and **Table 2** show a part of phrases in each cluster of the temporal clustering, where phrases having similar burst patters are grouped into the same cluster.

We choose a week as a unit of time series, which gives us reasonable revision counts per unit time. Temporal clustering is applied over revisions created in the 167 weeks from Aug.1st, 2012 to Oct.6th, 2015. We tried several times and four clusters were observed. The burst detection step realizes removal of most of spikes and noises, so that only notable bursts are observed. In cluster 1, the patterns can be linked to real-world events. For example, in November 2012, Ted Cruz won general election. He is the first Hispanic American to serve as a U.S. Senator representing Texas. Then in 2015, he declared to join the election again so the phrase bursts at 2015 summer. Cluster 2 only has three peaks, which are related to phrases about isolated events, including illegal immigrants. In Sept. 2015, a new policy for immigration was in effect, and the relationship between this event and the burst is obvious. Cluster 3 is similar to cluster 1, but in cluster 3 two main bursts share the same level, which can be linked to political news in 2012 and 2015 on presidential elections. In cluster 4, there is no obvious burst, and most phrases in this cluster are common nouns which are quite frequent in articles.

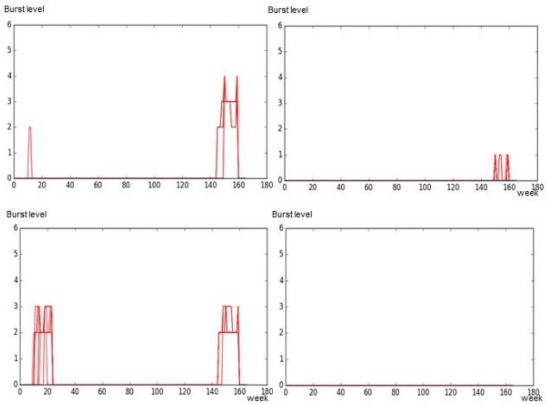

Fig. 2. Temporal patterns of evolutionary topics

We compared temporal clustering results without burst detection, and clustering by Euclidian distance. Due to space limitation, we show overviews of the results. When the burst detection step was omitted, the time sequences contained sharp peaks and noises. Also, when the Euclidean metric was used instead of DTW, time shifting ability

was lost. In both cases, cluster qualities were inferior than **Table 2**, and resulting clusters contain phrases that are difficult to link to real-world events. We therefore conclude that both burst detection and DTW steps are necessary in our scheme.

6 Conclusion and future work

In this paper, we proposed a novel approach for capturing topical trends, by analyzing changes of phrase frequencies in edit revisions of articles. We combine burst detection and DTW for temporal clustering of phrases, where phrases that were edited around the same time periods were grouped. Bust detection of phrases are utilized to simplify phrase time series. Our experimental results show that the proposed method produces meaningful clusters of phrases that share similar burst patterns. In future, we will focus on improving clustering qualities and efficiency.

References

1. A. Aji, Y. Wang, E. Agichtein, et al.: Using the past to score the present: Extending term weighting models through revision history analysis. Proc. 19th ACM Int. Conf. Information and Knowledge Management, pp. 629-638, 2010.
2. S. Adwan, and H. Arof: On improving Dynamic Time Warping for pattern matching. Measurement, 2012, 45(6): 1609-1620.
3. Doan, A., R. Ramakrishnan, and A. Y. Halevy: Crowdsourcing systems on the World-Wide Web. Commun. ACM 54, 4, 86-96, 2011.
4. http://opennlp.apache.org/
5. A. Kalogeratos, P. Zagorisios and A. Likas: Improving Text Stream Clustering using Term Burstiness and Co-burstiness. Proc. 9th Hellenic Conf. Artificial Intelligence. ACM, pp. 16, 2016.
6. J. Kleinberg: Bursty and Hierarchical Structure in Streams. Data Mining and Knowledge Discovery, 7(4): 373-397, 2003.
7. Y. Liu, Z. Gao, and M. Iwaihara: Identifying Evolutionary Topic Temporal Patterns Based on Bursty Phrase Clustering, DEIM Forum C5-1, Mar. 2017.
8. A Press. Wikipedia and Artificial Intelligence: An Evolving Synergy.
9. I Subašic, B Berendt. From bursty patterns to bursty facts: The effectiveness of temporal text mining for news. Proc. ECAI. 2010.
10. T. Tran, A. Ceroni, M. Georgescu, et al: Wikipevent: Leveraging wikipedia edit history for event detection. International Conference on Web Information Systems Engineering. Springer International Publishing, pp, 90-108, 2014.
11. Wikipedia: http://en.wikipedia.org/wiki/Wikipedia
12. J. Yang and J. Leskovec: Patterns of Temporal Variation in Online Media. Proc. 4th ACM Int. conf. Web Search and Data Mining. pp. 177-186, 2011.