Never Abandon Minorities: Exhaustive Extraction of Bursty Phrases on Microblogs Using Set Cover Problem

Masumi Shirakawa[†‡] and **Takahiro Hara**[‡] and **Takuya Maekawa**[‡]
hapicom Inc., Japan[†]
Graduate School of Information Science and Technology, Osaka University, Japan[‡]
shirakawa@hapicom.jp
{hara, maekawa}@ist.osaka-u.ac.jp

Abstract

We propose a language-independent data-driven method to exhaustively extract bursty phrases of arbitrary forms (e.g., phrases other than simple noun phrases) from microblogs. The burst (i.e., the rapid increase of the occurrence) of a phrase causes the burst of overlapping N-grams including incomplete ones. In other words, bursty incomplete N-grams inevitably overlap bursty phrases. Thus, the proposed method performs the extraction of bursty phrases as the set cover problem in which all bursty N-grams are covered by a minimum set of bursty phrases. Experimental results using Japanese Twitter data showed that the proposed method outperformed word-based, noun phrase-based, and segmentation-based methods both in terms of accuracy and coverage.

1 Introduction

Background and motivation. Trends on microblogs reflect manifold real-world events including natural disaster, new product launch, television broadcasting, public speech, airplane accident, scandal and national holiday. To catch real-world events, not a few researchers and practitioners have sought ways to detect trends on microblogs. Trend detection often involves bursty phrase extraction, i.e., extracting phrases of which occurrence rate in microblog texts posted within a certain period of time (and from a certain location) is much higher than that of the normal state. Extracted bursty phrases are directly used as trends as Twitter[1] officially provides, or sent to higher-order processes such as clustering and event labeling.

Bursty phrases on microblogs are likely to be noun phrases, but sometimes deviate from such standards. The title of a movie, song, game or any creation can be an arbitrary form like a long and/or general phrase (e.g., Spielberg's movie "catch me if you can", the Beatles' song "let it be")[2]. A memorable phrase (e.g., Steve Jobs's phrase "stay hungry, stay foolish") can also be a bursty phrase on microblogs. Even numbers and symbols can be potentially bursty phrases (e.g., "1984" can be a novel, "!!!" can be an artist). Any filtering rule such as stop word removal, part-of-speech (POS) tag restrictions, or length limit can mistakenly filter out bursty phrases.

Extracting irregularly-formed bursty phrases as described in the previous paragraph is difficult since no restriction can be used anymore to filter out incomplete N-grams. However, they are rare and little influence the overall accuracy even if they are correctly extracted. Not only that, tackling such difficult and rare cases easily leads to extracting many incomplete N-grams and deteriorating the accuracy. Almost all existing work has therefore ignored difficult and rare cases, and concentrated on extracting simple phrases (e.g., uni-grams, bi-grams, tri-grams, or noun phrases identified by POS taggers). By sacrificing minorities, most bursty phrases can be extracted with high accuracy. Thus, irregularly-formed phrases have been abandoned in bursty phrase extraction (and alike in many text analysis tasks).

Contributions. In this work, we aim to accurately and exhaustively extract bursty phrases of arbitrary forms from microblogs. The challenge here is: How do we avoid extracting bursty incomplete N-grams without introducing any filtering rule? To solve this challenging problem, we propose a set cover-based method. We found that the

[1] https://twitter.com/.

[2] We used old but popular titles for illustrative purposes.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2358–2367
Copenhagen, Denmark, September 7–11, 2017.

Table 1: Representative trend detection methods based on bursty phrases.

Method	Unit of process	Measure of burst
(Sayyadi et al., 2009)	Noun phrase	TF, DF, IDF
(O'Connor et al., 2010)	Uni-gram, bi-gram, tri-gram	Burstiness
(Mathioudakis and Koudas, 2010)	Uni-gram	Burstiness
(Weng and Lee, 2011)	Uni-gram	DF-IDF, H-measure (wavelet analysis)
(Metzler et al., 2012)	Uni-gram	Burstiness
(Li et al., 2012)	N-gram of any length	Deviation from Gaussian distribution
(Cui et al., 2012)	Hashtag	Deviation from Gaussian distribution
(Benhardus and Kalita, 2013)-1	Uni-gram, bi-gram, tri-gram	Burstiness
(Benhardus and Kalita, 2013)-2	Uni-gram	TF-IDF, entropy
(Aiello et al., 2013)-1	Uni-gram	Burstiness
(Aiello et al., 2013)-2	Uni-gram, bi-gram, tri-gram	DF-IDF
(Abdelhaq et al., 2013)	Uni-gram	Deviation from Gaussian distribution
(Schubert et al., 2014)	Uni-gram (pair)	Deviation from Gaussian distribution
(Feng et al., 2015)	Hashtag	Deviation from Gaussian distribution

burst (i.e., the rapid increase of occurrence) of a phrase causes the burst of overlapping incomplete N-grams. For example, if phrase "let it be" bursts, the occurrence of some overlapping N-grams such as "let it", "let it be is", and "it be is" inevitably increases, possibly generating bursty incomplete N-grams. Given that bursty incomplete N-grams always accompany overlapping bursty phrases, we can avoid extracting bursty incomplete N-grams using the set cover problem (Chvátal, 1979). The proposed set cover-based method finds a minimum set of bursty phrases that cover all bursty N-grams including incomplete ones. Because the set cover problem is NP-complete, the proposed method approximately solves it by iteratively choosing an N-gram that most covers remaining bursty N-grams.

The advantages of the proposed method are as follows. **1) Exhaustive.** The proposed method can extract bursty (contiguous) phrases of arbitrary forms. In our experiment, the coverage has been shown to be larger than that of word-based, noun phrase-based, and segmentation-based methods. **2) Accurate.** With adequate preprocessing of auto-generated texts, the proposed method achieved 99.3% of precision for top 10 bursty phrases and 97.3% for top 50 bursty phrases, which were even higher than the comparative methods. **3) Language-independent.** Because the proposed method processes texts as a sequence of characters (or words), it works in any languages including those without word boundary such as Japanese. **4) Purely data-driven.** The proposed method only requires raw microblog texts and does not need external resources.

2 Related Work

Much work has been focused on trend detection or event detection from microblogs. Majority of representative trend detection methods (summarized in Table 1) start with extracting bursty phrases, often followed by clustering bursty phrases in order to link them to real-world events. Others first build clusters of words (uni-grams) by using word co-occurrence (Pervin et al., 2013) or topic models (Aiello et al., 2013; Diao et al., 2012; Lau et al., 2012) and then apply burst detection for clusters. Also, there are different approaches such as the document-based approach (Aiello et al., 2013), sketch-based model (Xie et al., 2013) and bursty biterm topic model (Yan et al., 2015).

It is noteworthy that most methods in Table 1 only focus on uni-grams, short N-grams (up to tri-grams), or noun phrases (or rely on hashtags). This is because majority of bursty phrases conform to such simple forms. The rest of bursty phrases are rare but their forms are irregular and difficult to stereotype by using rules. Trying to extract such irregularly-formed phrases easily leads to the deterioration of the precision due to incorrect extraction. Also, the recall can hardly be increased since they are only a small portion of all bursty phrases. To balance the precision and recall of bursty phrase extraction, focusing on simple phrases and ignoring rare cases is a reasonable strategy. In the field of trend detection on microblogs, this ignoring-minority strategy has become a de facto standard. However, it always fails to extract irregularly-formed bursty phrases. In

this work, we tackle the challenge of extracting bursty phrases without any restriction of forms.

Among methods in Table 1, Li et al. (Li et al., 2012) have only attempted to extract bursty phrases of arbitrary forms. Their method, Twevent, applies text segmentation (or chunking) before measuring the degree of the burst. Every microblog text is represented as a sequence of word N-grams called segments. N-gram length, Symmetric Conditional Probability (SCP) (da Silva and Lopes, 1999), and semantic resources extracted from Wikipedia are integrated to obtain the best segmentation results. Owing to a good segmentation algorithm, it can potentially detect bursty phrases other than noun phrases, uni-grams, bi-grams, and tri-grams with high accuracy. However, it is still possible to miss irregularly-formed bursty phrases because they are likely to be segmented incorrectly. Our set cover-based method guarantees that all bursty N-grams including irregularly-formed ones must be covered by extracted bursty phrases. Thus, it is unlikely to miss irregularly-formed bursty phrases.

One more thing to note in Table 1 is that how to measure the degree of the burst can be largely classified into a few groups: burstiness, TF-IDF-based measures, and distribution-based methods. The simplest approach is burstiness, which is the ratio of the occurrence rate in target and reference document sets. Reference document set is usually constructed from past microblog texts. TF-IDF-based measures compute term frequency (TF) or document frequency (DF) in the target document set and inverse document frequency (IDF) in the reference document set. Distribution-based methods generally measure how much the observed frequency deviates from the distribution of the normal state using standard scores (z-scores). The Poisson distribution is proper to represent the number of occurrence of phrases, but the Gaussian distribution is often used as its approximation due to computational reasons. We in this work primarily adopt a Gaussian distribution-based approach and use the z-score as the measure of the burst because it reasonably works well for the different magnitude of the number of occurrence.

3 Bursty Phrase Extraction

3.1 Problem Formulation

We formalize the problem of bursty phrase extraction from microblogs. Target document set D_T is a set of microblog messages posted within a certain period of time (e.g., one day, three hours). Reference document set D_R is a set of microblog messages posted before the target time. Both are usually limited to certain locations or languages. Each document is a sequence of characters (or words). The objective here is to extract bursty complete phrases from D_T as much as possible using D_R as the normal state. The output format is a list of bursty N-grams $L = [g_1, g_2, \cdots, g_{|L|}]$ (g_i is an N-gram or a sequence of characters) ranked by the degree of the burst. The accuracy and coverage of top K (K is a user-specified parameter) bursty phrases are important evaluation criteria.

The degree of the burst for N-grams in D_T is defined as the z-score when the Gaussian distribution is estimated from D_R. While most N-grams occur once in a single microblog message, a few N-grams are repeatedly used in it. We thus employ the document frequency-based z-score as the degree of the burst. The z-score of N-gram g is specifically computed as

$$zscore(g) = \frac{df(g) - \mu(g)}{\sigma(g)} \quad (1)$$

where $df(g)$ is the document frequency of g in D_T, and $\mu(g)$ and $\sigma(g)$ are respectively the mean and standard deviation for the document frequency of g estimated from D_R. Given that the Gaussian distribution used here is the approximation of the Poisson distribution, $\sigma(g)$ is approximated by $\sqrt{\mu(g)}$. To smooth $\mu(g)$ when g never occurs in D_R, we add $\delta = 1$ to $\mu(g)$.

3.2 Basic Idea

To exhaustively extract bursty phrases without any restriction of their forms, we have to refrain from using filtering rules such as stop word removal, POS tag restrictions, and length limit. Without filtering rules, there are far more bursty incomplete N-grams than bursty complete phrases. It is challenging to extract bursty phrases including irregularly-formed ones and at the same time to avoid extracting bursty incomplete N-grams. Is there any evident difference between bursty incomplete N-grams and bursty phrases of irregular forms?

We scrutinized Twitter data and found the following fact: Bursty incomplete N-grams always accompany overlapping bursty phrases provided that the definition of the burst is appropriate. We

Algorithm 1: Greedy Set Cover Algorithm for Bursty Phrase Extraction

Input: Target document set D_T, reference document set D_R
Output: Ranked list of bursty phrases L

1 Initialize L;
2 Get a set of valid N-grams $G_V = \{g_i\}$ satisfying $burstiness(g_i) \geq 1 + \epsilon$;
3 Get a set of bursty N-grams $G_B = \{g_i\} \subset G_V$ satisfying $zscore(g_i) \geq \theta$ and $zscore_{tf}(g_i) \geq \theta$;
4 Discard trivial N-grams from G_V;
5 **while** $G_B \neq \emptyset$ **do**
6 Select $g \in G_V$ that most cover the occurrence of bursty N-grams in G_B;
7 Get a set of longer N-grams $G_g \subset G_V$ containing g;
8 Determine a set of containment N-grams $G_C \subset G_g$ for g;
9 **if** $G_C = \emptyset$ **then**
10 Push g into L;
11 Negate the occurrence of all N-grams in G_B and G_V where g overlaps;
12 Delete $g_i \in G_B$ if it does not satisfy $zscore_{tf}(g_i) \geq \theta$;
13 **else**
14 Negate the occurrence of g where containment N-grams $g_i \in G_C$ overlap;
15 **end**
16 **end**
17 Rerank L based on the actual z-score;

explain this phenomenon in due order. When many microblog users intensively use a certain phrase, it becomes a bursty phrase. Here, the increment of the occurrence of the phrase contributes to the increment of the occurrence of overlapping N-grams. Consequently, (incomplete) N-grams that overlap the phrase can also burst. Thus, bursty incomplete N-grams always have their original bursty phrases that overlap each other.

Based on the phenomenon described in the previous paragraph, bursty incomplete N-grams cease bursting if their original bursty phrases disappear from microblog texts. In other words, all bursty N-grams including incomplete ones can be covered (overlapped) by bursty phrases. Given that bursty phrases cause bursty incomplete N-grams but the reverse was not true, we can formalize the extraction of bursty phrases as the set cover problem (Chvátal, 1979) where a minimum number of bursty phrases are selected to cover all bursty N-grams. When all selected bursty phrases are removed from microblog texts, it is guaranteed that there is no bursty N-gram.

3.3 Proposed Algorithm

Algorithm 1 is a pseudo-code of the greedy set cover algorithm for bursty phrase extraction. As formulated in Section 3.1, the input data is target D_T and reference document sets D_R of microblog messages. The output is a ranked list of bursty phrases $L = [g_1, g_2, \cdots, g_{|L|}]$. Basically, Algorithm 1 iteratively selects an N-gram that most covers the occurrence of bursty N-grams (Line 6) until all bursty N-grams are covered. In the following, we describe points of the Algorithm 1.

3.3.1 Bursty N-grams and Valid N-grams

A set of bursty N-grams G_B in Algorithm 1 (Line 3) corresponds to the universe of the set cover problem which should be all covered. Bursty N-grams satisfy z-score threshold θ both in document frequency-based (Eq. (1)) and term frequency-based z-scores (Eq. (2)).

$$zscore_{tf}(g) = \frac{tf(g) - \mu_{tf}(g)}{\sigma_{tf}(g)} \qquad (2)$$

Here, $tf(g)$ is the term frequency of g in D_T, and $\mu_{tf}(g)$ and $\sigma_{tf}(g)$ are respectively the mean and standard deviation for the term frequency of g estimated from D_R. The term frequency-based z-score is used to judge whether a bursty N-gram in G_B still bursts when its occurrences are partly covered. This is required to handle N-grams repeatedly occurring in a single microblog message.

Valid N-grams in G_V (Line 2) are qualified to be bursty phrases to cover G_B. We differentiate bursty N-grams and valid N-grams (specifically, $G_B \subset G_V$) to handle threshold problems.

Namely, it is possible that bursty incomplete N-grams satisfy the z-score threshold but their original bursty phrases do not satisfy the threshold. The criterion of the valid N-gram is defined by using burstiness.

$$burstiness(g) = \frac{df(g)}{df_R(g)} \cdot \frac{|D_R|}{|D_T|} \quad (3)$$

Here, $df_R(g)$ is the document frequency of N-gram g in D_R. To avoid division by zero, smoothing term $\delta = 1$ is added to $df_R(g)$. When $burstiness(g)$ satisfies threshold $1 + \epsilon$ (i.e., the occurrence of g actually increases), g becomes a valid N-gram. To reduce pointless processes, trivial N-grams that can never be phrases (e.g., starting or ending with spaces, occurring only as a part of a sole longer N-gram) are discarded (Line 4).

3.3.2 Occurrence-based Set Covering

Whereas the standard set cover problem assumes that each item is atomic, i.e., the state of an item is either *not covered* or *covered*, the proposed method manages the state of covering by using all occurrences of bursty N-grams. When a valid N-gram is selected (Line 10), the occurrence of all N-grams in G_B and G_V that the valid N-gram overlaps is negated (Line 11). Here, a bursty N-gram in G_B is completely covered if the term frequency-based z-score computed from remaining occurrences of the N-gram does not satisfy the threshold (Line 12).

Occurrence-based set covering can solve the case when a bursty N-gram is covered by multiple bursty phrases. That is, the bursty N-gram is not completely covered even if one of the bursty phrases is selected. For example, given two bursty phrases "let it be" and "let it go" (a song), incomplete N-gram "let it" ceases bursting only when the occurrence of both phrases is negated. Also, it can handle partially overlapping N-grams (e.g., "let it be" and "be is") based on the number of overlaps.

3.3.3 Containment N-grams

N-grams that are contained in multiple phrases should be carefully treated in the set cover problem. Shorter N-grams are likely to be contained in more phrases and chosen in the set cover problem even if they are incomplete. For example, when phrases "let it be" and "let it go" burst, shared incomplete N-gram "let it" is likely to cover the occurrence of bursty N-grams more than the two phrases. To prevent selecting shared incomplete N-grams, we determine containment relations between an N-gram and longer N-grams containing it (Lines 7, 8). We define longer N-grams in containment relations as containment N-grams. Containment relations negate the occurrence of the shorter N-gram where containment N-grams overlap (Line 14). Containment relation is inspired by the idea of rectified frequency used in a segmentation-based quality phrase mining method (Liu et al., 2015), though how to rectify the frequency is different. Note that containment relations do not necessarily mean that shorter N-grams are incomplete because both shorter and longer N-grams can be phrases (e.g., "new york" and "new york times").

How to determine containment relations is designed not to contradict the greedy set cover algorithm. Briefly, containment relations hold when only the containment N-grams among longer N-grams can cover the occurrence of bursty N-grams more than the shorter N-gram owing to the containment relations. That is, we find a stable state of containment relations. To find a stable state, we initially define temporal containment relations and then iteratively find a set of containment N-grams so that containment relations become stable.

Initial containment N-grams are determined using burst context variety, which is an extension of accessor variety (Feng et al., 2004). Accessor variety roughly measures how much an N-gram is likely to be a phrase. It specifically counts the number of distinct characters (or words) before or after the N-gram and employs the minimum one. The drawback of accessor variety is that it handles left and right contexts independently. We thus modify it as context variety in which both left and right contexts are simultaneously counted. Context variety can also be calculated using the set cover problem. In particular, a left or right character (or word) that most covers the occurrence of the N-gram is iteratively selected until all the occurrences are covered. Burst context variety is a further extension of context variety to count the number of contexts only for additional term frequencies given the mean of the term frequency. When the burst context variety of an N-gram is not greater than that of a longer N-gram, we extract the context (i.e., left or right character) and define all longer N-grams having the context as initial containment N-grams.

There are two minute settings for measuring burst context variety. One is that the start and end of every line are all unique and should be counted as distinct contexts. The other is that symbols[3] should be ignored when checking left and right contexts of the N-gram.

3.3.4 Reranking Bursty Phrases

The output L is finally reranked by using actual z-scores (Line 17), which are different from z-scores calculated from raw document frequency. The actual z-score is calculated from the number of occurrences of bursty N-grams that N-gram $g \in L$ actually covered during the set cover process. Since a single valid N-gram usually covers multiple bursty N-grams, the z-score for every covered bursty N-gram is recalculated and the maximum is used as the actual z-score. When the maximum z-score exceeds the original z-score, the original z-score is preserved. Without reranking, incomplete N-grams that covered very few occurrences of bursty N-grams may be overestimated.

4 Evaluation

We evaluated the proposed method using two weeks of Japanese Twitter data.

4.1 Setup

Dataset. We created a dataset using Twitter Streaming API statuses/sample. We chose Japanese as a language because it was one of popular languages used in Twitter and because it has no word boundary and finding phrases is difficult compared to space-delimited languages such as English. We specifically collected 15 days of tweets from September 30 to October 14, 2016[4]. For each day from October 1 to 14, we extracted bursty phrases in reference to the previous day.

To maximally alleviate the influence of auto-generated contents such as tweets posted by bots and spammers, we filtered out them. Detecting bots and spammers (Chu et al., 2010; Subrahmanian et al., 2016) is a nontrivial research task and out of the scope of this paper. In this experiment, we used simple but effective heuristics. First, we only used tweets posted by Twitter official clients[5] because they were mainly used by normal users. Second, we discarded tweets including URLs because most spammers tried to lure users to visit their websites. Third, retweets (actions to propagate someone's tweets) were discarded. The maximum and minimum numbers of remaining tweets per day were 326,002 (Oct. 2) and 253,044 (Oct. 13), respectively. Additionally, we deleted hashtags (starting by #) and mentions (starting by @) from tweets. Note that the degree of the burst for URLs, hashtags, and mentions can be independently measured. We concentrated on extracting bursty phrases from plain texts.

Ground truth. To create the ground truth, we mixed N-grams extracted with all methods and then manually annotated each N-gram by checking its real usage in tweets. While most N-grams were easily judged as complete phrase (i.e., *correct*) or incomplete N-grams (i.e., *incorrect*), a few N-grams were difficult to judge (e.g., a last name that was not frequently used to indicate the person in tweets). We annotated such N-grams as *maybe correct* and regarded as correct in the strict case and incorrect in the loose case when measuring evaluation metrics.

Evaluated methods. In the proposed method (**Proposed**), we processed Japanese texts as sequences of characters. Default threshold parameters θ and ϵ were set to 10 and 0.5, respectively. Comparative methods included the word-based method (**Word**), noun phrase-based method (**NP**), and segmentation-based method (**Segment**) (Li et al., 2012). Because these methods require word breaking, we used MeCab (Kudo et al., 2004) (version 0.996, ipadic as a dictionary), a Japanese morphological analyzer. The word-based method uses all self-sufficient words as candidate phrases. The noun phrase-based method regards concatenated successive nouns as a candidate phrase. In word-based and noun phrase-based methods, the dictionary or vocabulary significantly affects the performance. Thus, we also used neologd[6] (Sato et al., 2017) (version v0.0.5 updated at May 2, 2016), a neologism dictionary extracted from many language resources on the web, as an additional dictionary (**+Dic**). The segmentation-based method detects segments (chunks) from a sequence of words as candidate phrases. The segmentation model was constructed using three

[3] In our experiments, we used isalnum (or iswalnum for wide characters) in C++ standard library to distinguish words and symbols.

[4] We considered that a day changed at 4 am JST.

[5] `https://about.twitter.com/products/list`.

[6] `https://github.com/neologd/mecab-ipadic-neologd`.

Table 2: Average precision of top K bursty phrases (strict case).

Method	Top10	Top20	Top30	Top40	Top50
Word	0.700	0.714	0.731	0.743	0.743
Word+Dic	0.929	0.907	0.895	0.907	0.904
NP	0.936	0.929	0.936	0.927	0.930
NP+Dic	0.971	0.968	0.962	0.955	0.954
Segment	0.921	0.918	0.905	0.918	0.914
Proposed	**0.993**	**0.993**	**0.981**	**0.979**	**0.973**

Table 3: Average precision of top K bursty phrases (loose case).

Method	Top10	Top20	Top30	Top40	Top50
Word	0.764	0.779	0.812	0.821	0.817
Word+Dic	0.950	0.936	0.936	0.945	0.941
NP	0.950	0.943	0.952	0.945	0.946
NP+Dic	0.971	0.979	0.974	0.971	0.973
Segment	0.957	0.954	0.952	0.955	0.953
Proposed	**0.993**	**0.996**	**0.998**	**0.995**	**0.991**

months of tweets (from July 1 to Sept. 30, 2016) and Japanese Wikipedia dump data (as of Oct. 1, 2016).

Evaluation metrics. We employed precision and min-z-score of top K bursty phrases (K was set to 10, 20, 30, 40, or 50) as evaluation metrics. We measured the precision both in strict and loose cases based on the ground truth labels. The min-z-score (the minimum of the z-score, computed from raw document frequency) was introduced to evaluate how much the top K output ranked by z-scores included highly bursty phrases. To increase the min-z-score, all the top K phrases should have high z-scores and hence highly bursty phrases should not be ignored. Thus the min-z-score can evaluate the coverage of extracted bursty phrases using a fixed size of the output. Higher precision and min-z-score indicate that the method can more accurately and exhaustively extract bursty phrases.

4.2 Performance Results: Precision

Tables 2, 3 show the precision of bursty phrase extraction. It was surprisingly that the proposed method achieved higher precision than noun-phrase based methods, which were supposed to be safety by sacrificing irregularly-formed phrases. The precision of the proposed method for top 50 bursty phrases was 97.3% (correct phrases were 681 out of 700) in the strict case and 99.1% (694 out of 700) in the loose case. The precision for top 10 bursty phrases was 99.3% (139 out of 140) even in the strict case. The results demonstrate that the burst information alone can accurately find the boundary of bursty phrases using the set cover problem. Error cases of the proposed method were largely classified into two: base sequences of diversified expressions and phrases with strongly-correlated attached characters.

In comparative methods, the accuracy tended to be high when noun phrases were used and the dictionary was well defined. Especially, the use of the neologism dictionary boosted the precision. The segmentation-based method also marked moderately high precision.

Table 4: Average min-z-score of top K bursty phrases.

Method	Top10	Top20	Top30	Top40	Top50
Word	41.5	25.7	20.4	17.2	15.3
Word+Dic	44.7	29.0	22.5	19.3	16.7
NP	42.6	26.4	20.4	16.7	14.6
NP+Dic	42.7	28.8	21.0	17.6	15.3
Segment	45.0	31.2	24.3	20.5	18.0
Proposed	**50.1**	**33.5**	**24.8**	**21.5**	**19.4**

4.3 Performance Results: Coverage

Table 4 shows the min-z-score of bursty phrase extraction. The proposed method achieved higher min-z-score than the comparative methods. This was because the proposed method extracted bursty phrases regardless of their forms. Noun phrase-based methods tended to miss highly bursty phrases of irregular forms. Therefore the min-z-score of extracted top K bursty phrases became small. Among comparative methods, the segmentation-based method best achieved the min-z-score since it did not restrict the form of phrases. However, it was still possible to miss very irregular phrases due to segmentation mistakes and the min-z-score was less than that of the proposed method. The use of the neologism dictionary increased the min-z-score as well as the precision, indicating that it had no negative effect.

To intuitively assess the coverage of the proposed method, we manually counted the number of bursty phrases that were extracted with the proposed method (when $K = 10$, i.e., 139 correct phrases) but completely missed with the comparative methods. Here, we regarded *completely missed* when neither of the bursty phrase, overlapping N-grams including incomplete ones nor orthographic variants were extracted in top

Table 5: Number of bursty phrases extracted with the proposed method ($K = 10$) but completely missed with comparative methods.

Word	Word+Dic	NP	NP+Dic	Segment
17/139	7/139	10/139	4/139	8/139

Table 6: Average precision of top K bursty phrases with different parameter settings in the proposed method (strict case).

Parameters	Top10	Top20	Top30	Top40	Top50
θ=5, ϵ=0.2	0.993	0.993	0.976	0.977	0.971
θ=5, ϵ=0.5	0.993	0.993	0.976	0.977	0.971
θ=5, ϵ=1.0	0.993	0.993	0.976	0.979	0.971
θ=10, ϵ=0.2	0.993	0.993	0.981	0.979	0.973
θ=10, ϵ=0.5	0.993	0.993	0.981	0.979	0.973
θ=10, ϵ=1.0	0.993	0.993	0.981	0.979	0.973
θ=15, ϵ=0.2	0.993	0.989	0.971	0.971	0.966
θ=15, ϵ=0.5	0.993	0.989	0.971	0.971	0.966
θ=15, ϵ=1.0	0.993	0.989	0.971	0.971	0.966

100 bursty N-grams. Table 5 shows the results. Although the percentages were small, any comparative method completely missed some highly bursty phrases that were extracted with the proposed method. Note that the proposed method did not completely miss top 10 bursty phrases of comparative methods at all since the set cover problem inevitably covered all bursty N-grams.

4.4 Influence of Parameter Settings

We changed threshold parameters θ and ϵ to evaluate their influence on performance. Tables 6, 7 show the performance results with different parameter settings. We confirmed that both parameters, especially ϵ, hardly affected the precision and min-z-score. The results indicate that the threshold parameters can be roughly set based on data.

4.5 Examples

Table 8 shows top 10 bursty phrases (all of them are correct) on Oct. 1. This day contained many irregularly-formed phrases; phrases containing hiragana[7] characters (rank 5, 7, 8, 10), other than noun phrases (rank 5, 7), and containing symbols (rank 3). Especially, 映っちゃった (rank 5) and しやがれ (rank 7) were troublesome since they contain hiragana characters and at the same time they are other than noun phrases. Even with the neologism dictionary, the noun phrase-based method extracted しやがれ but missed 映っちゃった.

[7]In Japanese, hiragana is mainly used for auxiliary words and thus difficult to break into words or phrases.

Table 7: Average min-z-score of top K bursty phrases with different parameter settings in the proposed method.

Parameters	Top10	Top20	Top30	Top40	Top50
θ=5, ϵ=0.2	50.6	33.4	24.9	21.4	19.2
θ=5, ϵ=0.5	50.6	33.4	24.8	21.4	19.2
θ=5, ϵ=1.0	50.6	33.4	24.9	21.4	19.3
θ=10, ϵ=0.2	50.1	33.5	24.8	21.5	19.4
θ=10, ϵ=0.5	50.1	33.5	24.8	21.5	19.4
θ=10, ϵ=1.0	50.2	33.5	24.9	21.5	19.5
θ=15, ϵ=0.2	50.2	33.5	25.1	21.8	19.4
θ=15, ϵ=0.5	50.2	33.5	25.1	21.8	19.5
θ=15, ϵ=1.0	50.2	33.5	25.1	21.8	19.6

Table 8: Example of top 10 bursty phrases in Japanese (Oct. 1, 2016).

1.	三代目 (*Sandaime*, generally meaning "third", short name of a music group)
2.	坂口杏里 (*Sakaguchi Anri*, celebrity)
3.	**HE ★ VENS** (idol group)
4.	プリライ (*Pri-rai*, short name of a live concert)
5.	映っちゃった (*Utsucchatta*, generally meaning "it was reflected", short name of a TV program)
6.	単独 (*Tandoku*, generally meaning "singleness", implying a live concert without supporting acts)
7.	しやがれ (*Shiyagare*, generally a phrase used in an imperative sentence, short name of a TV program)
8.	うたプリ (*Utapri*, short name of an anime)
9.	**WORKING** (short name of an anime)
10.	うたぷり (*Utapri*, short name of an anime)

To demonstrate the language-independent nature of the proposed method, we also applied it to English with character-level processing[8]. Table 9 shows an example of bursty phrases extracted from English tweets. Whereas an incomplete phrase (rank 2) was extracted due to auto-generated contents that could not be eliminated from data, other top bursty phrases were correctly extracted. The proposed method also extracted a very long Internet meme (rank 5), which burst along with its counterpart "anyone that knows me knows i love _____." (rank 17).

5 Conclusions

We proposed a language-independent data-driven method to accurately and exhaustively extract bursty phrases of arbitrary forms from microblogs. We found that bursty incomplete N-grams always

[8]Of course, we can process English in word level.

Table 9: Example of top 10 bursty phrases in English (Oct. 1, 2016, PST). Note that these bursty phrases were generated by processing English tweets in character level.

1.	**LizQuen InSydney Oct30** (maybe an auto-generated phrase used like a tag)
2.	**for The 100 most beautiful faces in 2016** (maybe an auto-generated sequence used to vote a celebrity for the ranking)
3.	**Lamar Jackson** (American football player)
4.	**Clemson** (American football team)
5.	**quote with what you think the answer is and copy this tweet to see what people say about you** (Internet meme)
6.	**Tennessee** (American football team)
7.	**Louisville** (American football team)
8.	**Milner** (American football player)
9.	**FSU** (American football team)
10.	**Louis Walsh** (talent manager)

accompany overlapping bursty phrases by ascertaining the mechanism why incomplete N-grams burst. Based on the findings, the proposed method solves the extraction of bursty phrases as the set cover problem where a minimum set of bursty phrases covers all bursty N-grams including incomplete ones. We confirmed from experimental results that the proposed method outperformed noun phrase-based and segmentation-based methods both in terms of the accuracy and coverage. The source code of the proposed method is publicly available[9].

The future work includes the reduction or estimation of the computation time and memory usage. They increase as the target document set grows or, specifically the number of occurrences of bursty N-grams increases. Also, handling auto-generated contents is an important issue. The proposed method should be used with effective methods of identifying auto-generated contents, bots, and spammers in microblogs.

Acknowledgments

This research is partially supported by the Grant-in-Aid for Scientific Research (A)(2620013) of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and JST, Strategic International Collaborative Research Program, SICORP.

[9] https://github.com/iwnsew/sc-bursty-phrase.

References

Hamed Abdelhaq, Christian Sengstock, and Michael Gertz. 2013. EvenTweet: Online Localized Event Detection from Twitter. *Proceedings of the VLDB Endowment*, 6(12):1326–1329.

Luca Maria Aiello, Georgios Petkos, Carlos Martin, David Corney, Symeon Papadopoulos, Ryan Skraba, Ayse Göker, Ioannis Kompatsiaris, and Alejandro Jaimes. 2013. Sensing Trending Topics in Twitter. *IEEE Transactions on Multimedia*, 15(6):1268–1282.

James Benhardus and Jugal Kalita. 2013. Streaming Trend Detection in Twitter. *International Journal of Web Based Communities*, 9(1):122–139.

Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. 2010. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *Proceedings of Annual Computer Security Applications Conference (ACSAC)*, pages 21–30.

Václav Chvátal. 1979. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235.

Anqi Cui, Min Zhang, Yiqun Liu, Shaoping Ma, and Kuo Zhang. 2012. Discover Breaking Events with Popular Hashtags in Twitter. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 1794–1798.

Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. 2012. Finding Bursty Topics from Microblogs. In *Proceedings of Meeting of the Association for Computational Linguistics (ACL)*, pages 536–544.

Haodi Feng, Kang Chen, Xiaotie Deng, and Weimin Zheng. 2004. Accessor Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30(1):75–93.

Wei Feng, Chao Zhang, Wei Zhang, Jiawei Han, Jianyong Wang, Charu Aggarwal, and Jianbin Huang. 2015. STREAMCUBE: Hierarchical Spatio-temporal Hashtag Clustering for Event Exploration over the Twitter Stream. In *Proceedings of IEEE International Conference on Data Engineering (ICDE)*, pages 1561–1572.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 230–237.

Jey Han Lau, Nigel Collier, and Timothy Baldwin. 2012. On-line Trend Analysis with Topic Models: #twitter trends detection topic model online. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 1519–1534.

Chenliang Li, Aixin Sun, and Anwitaman Datta. 2012. Twevent: Segment-based Event Detection from

Tweets. In *Proceedings of ACM Conference on Information and Knowledge Management (CIKM)*, pages 155–164.

Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining Quality Phrases from Massive Text Corpora. In *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1729–1744.

Michael Mathioudakis and Nick Koudas. 2010. TwitterMonitor: Trend Detection over the Twitter Stream. In *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 1155–1158.

Donald Metzler, Congxing Cai, and Eduard Hovy. 2012. Structured Event Retrieval over Microblog Archives. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 646–655.

Brendan O'Connor, Michel Krieger, and David Ahn. 2010. TweetMotif: Exploratory Search and Topic Summarization for Twitter. In *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 384–385.

Nargis Pervin, Fang Fang, Anindya Datta, Kaushik Dutta, and Debra Vandermeer. 2013. Fast, Scalable, and Context-Sensitive Detection of Trending Topics in Microblog Post Streams. *ACM Transactions on Management Information Systems*, 3(4):19:1–19:24.

Toshinori Sato, Taiichi Hashimoro, and Manabu Okumura. 2017. Implementation of a Word Segmentation Dictionary Called mecab-ipadic-NEologd and Study on How to Use It Effectively for Information Retrieval (in Japanese). In *Proceedings of Meeting of the Association for Natural Language Processing*, pages NLP2017–B6–1.

Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. 2009. Event Detection and Tracking in Social Streams. In *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 311–314.

Erich Schubert, Michael Weiler, and Hans-Peter Kriegel. 2014. SigniTrend: Scalable Detection of Emerging Topics in Textual Streams by Hashed Significance Thresholds. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 871–880.

Joaquim Ferreira da Silva and Jos'e Gabriel Pereira Lopes. 1999. A Local Maxima Method and a Fair Dispersion Normalization for Extracting Multi-word Units from Corpora. In *Proceedings of Meeting on Mathematics of Language (MOL)*, pages 369–381.

V.S. Subrahmanian, Amos Azaria, Skylar Durst, Vadim Kagan, Aram Galstyan, Kristina Lerman, Linhong Zhu, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. 2016. The DARPA Twitter Bot Challenge. *IEEE Computer*, 49(6):38–46.

Jianshu Weng and Bu-Sung Lee. 2011. Event Detection in Twitter. In *Proceedings of International AAAI Conference on Weblogs and Social Media (ICWSM)*, pages 401–408.

Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. 2013. TopicSketch: Real-Time Bursty Topic Detection from Twitter. In *Proceedings of IEEE International Conference on Data Mining (ICDM)*, pages 837–846.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2015. A Probabilistic Model for Bursty Topic Discovery in Microblogs. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, pages 353–359.