Event Detection using Images of Temporal Word Patterns

Yunli Wang
Yunli.Wang@nrc-cnrc.gc.ca

Cyril Goutte
Cyril.Goutte@nrc-cnrc.gc.ca

Multilingual Text Processing
National Research Council Canada
Ottawa ON, Canada

Abstract

Detecting events from social media requires to deal with the noisy sequences of user generated text. Previous work typically focuses either on semantic patterns, using e.g. topic models, or on temporal patterns of word usage, e.g. using wavelet analysis. In our study, we propose a novel method to capture the temporal patterns of word usage on social media, by transforming time series of word occurrence frequency into images, and clustering images using features extracted from the images using the convolutional neural network *ResNet*. These clusters are then ranked by burstiness, identifying the top ranked clusters as detected events. Words in the clusters are also filtered using co-occurrence similarity, in order to identify the most representative words describing the event. We test our approach on one Instagram and one Twitter datasets, and obtain performance of up to 80% precision from the top five detected events on both datasets.

1 Introduction

Social media are a rich source for news data, and often report events in a more timely manner than traditional media. Event detection from social media is quite challenging because of the noisy nature of the data. We adopt the definition of event from Hasan et al. (2018): an event, in the context of social media, is an occurrence of interest in the real world which instigates a discussion of event-associated topics by various users of social media, either soon after the occurrence or, sometimes, in anticipation of it. Approaches to event detection can be classified according to event types: *specified* or *unspecified* (Farzindar and Khreich, 2015). For specified event detection, some information such as time, type or description of target events is known beforehands, for example, detecting earthquakes (Sakaki et al., 2010). We focus on unspecified event detection, for which no prior information is available.

Previous work on unspecified event detection typically uses topic modeling or signal processing approaches. Topic modeling methods are able to discover topics based on semantic similarities between words in an unsupervised way (Pozdnoukhov and Kaiser, 2011; Chae et al., 2012; Zhou and Chen, 2014), but the temporal similarity between words is not captured. Signal processing methods such as wavelet analysis pay more attention to the temporal correlation between words (Weng and Lee, 2011; Li et al., 2012; Schubert et al., 2014), but ignore the semantic similarity. One key challenge of unspecified event detection from social media data is to filter noisy messages unrelated to actual events.

In recent years, deep learning approaches have revolutionized image processing, speech recognition, and most of Natural Language Processing. Convolutional Neural Networks (CNNs) have become the leading architecture for many image processing, classification, and detection tasks. The features extracted by CNNs have been shown to provide impressive baselines for various computer vision tasks (Oquab et al., 2014; Sharif Razavian et al., 2014). CNNs were also used for specified event detection: Lee et al. (2017) used CNNs in unsupervised feature learning and supervised classi-

In: A. Aker, D. Albakour, A. Barrón-Cedeño, S. Dori-Hacohen, M. Martinez, J. Stray, S. Tippmann (eds.): Proceedings of the NewsIR'19 Workshop at SIGIR, Paris, France, 25-July-2019, published at http://ceur-ws.org

fication to detect adverse drug events from tweets; Bischke et al. (2016) used visual features extracted from images by X-ResNet (an extension of *ResNet*, He et al. (2016)) and metadata features to detect flood events from satellite images.

We introduce the novel idea of transforming word usage patterns into images, then use features extracted from those images in order to detect unspecified events from social media. The event detection problem is then addressed as an image clustering task by transforming the time series of word occurrences into images. We adopt the deep learning model *ResNet* to extract features from these images, identify clusters based on those images, and rank them by burstiness. Our experiments show that the performance of our system is robust across different parameter settings.

2 Method

Our method includes four steps: transforming time series of word frequencies into images; clustering those images; ranking clusters by burstiness; filtering the words in each cluster by co-occurrence similarity. We name this proposed approach Image Co-occurrence Event detection (ICE), as it relies on representing temporal word usage by images, and selecting relevant words using co-occurrence similarity. In the first step, we adopt the Gramian Angular Filed (GAF) method (Wang and Oates, 2015) to transform time series of the frequency of each individual words into images. We then use *ResNet* to extract features from images and *k-means* to cluster those images. All clusters are ranked based on a burstiness measure. Finally, words in clusters are filtered to remove non-relevant words.

2.1 Transforming time series into images

Given a dataset of messages with time stamps, we first split the time range into T time intervals $\ell = 1 \dots T$ and merge all messages in time interval ℓ into one document. For each of the N unique words in the dataset, we build a temporal profile by counting the frequency of each word in each interval. This produces N time series of size T (Fig. 1), resulting in a $N \times T$ matrix of temporal profiles. Each row of the matrix contains a time series $w_1, w_2, ..., w_T$ for each word W.

GAF turns each time series into an image by first rescaling the time series into $[-1, 1]$:

$$w_i' = \frac{(w_i - W_{\max}) + (w_i - W_{\min})}{W_{\max} - W_{\min}} \qquad (1)$$

with $W_{\max}$ and $W_{\min}$ the maximun and minimum of the time series. Then the temporal correlation within time intervals is represented as a $T \times T$ image:

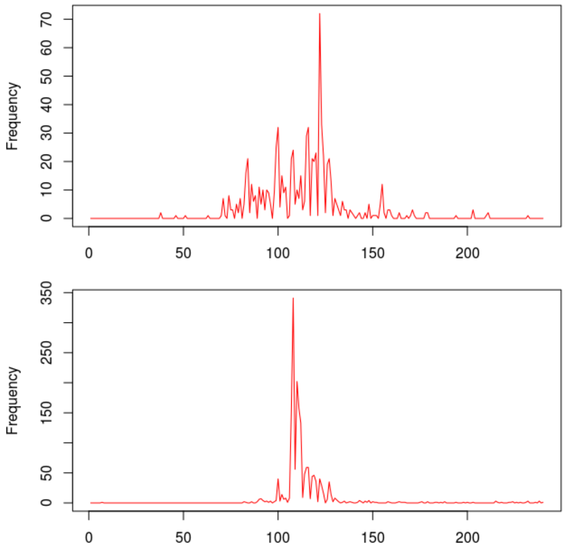

Figure 1: Temporal profiles of ”justiceforfreddie” (top) and ”prayforbaltimore” (bottom). #justiceforfreddie picks up after the death of Freddie Gray on April 19, 2015 ($\ell \approx 75$) with several spikes during the subsequent demonstrations. #prayforbaltimore peaks on the day of Mr Gray’s burial and following days.

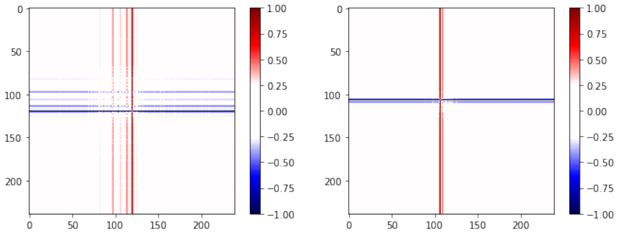

Figure 2: GAF image for ”justiceforfreddie” (left) and ”prayforbaltimore” (right).

$$G_W = \begin{bmatrix} < w_1', w_1' > & \dots & < w_1', w_T' > \\ < w_2', w_1' > & \dots & < w_2', w_T' > \\ \dots & \dots & \dots \\ < w_T', w_1' > & \dots & < w_T', w_T' > \end{bmatrix} \qquad (2)$$

where $\langle w_i', w_j' \rangle = w_j' \sqrt{1 - w_i'^2} - w_i' \sqrt{1 - w_j'^2}$ is a signed dissimilarity, representing the angular dissimilarity of the time series (see Wang and Oates (2015) for details). G_W is a $T \times T$ image representing the time series for a word. For example, GAF images of ”justiceforfreddie” and ”prayforbaltimore” are shown in Figure 2: Although specific to each word, they both show activity in the 100-120 region. GAF preserves the temporal dependency by containing the relative correlation between different time intervals, $G_{i,j}$.

2.2 Clustering images into clusters

After all time series of words are represented as GAF images, we use *ResNet* (He et al., 2016) to extract features from images, and cluster all images into $\mathcal{C}$ clusters using *k-means*. *ResNet* is a deep convolutional neural network. We applied *ResNet-50*, pretrained on ImageNet, to extract features from GAF images. Then, *k-means* was used to generate clusters of words. Words in the same clusters have similar GAF images and, therefore, similar temporal patterns. For instance, GAF images of "justiceforfreddie" and "prayforbaltimore" (Fig. 2) end up into the same cluster .

2.3 Ranking clusters by burstiness

After words are grouped into clusters, we use *burstiness* to rank all clusters. We use DF-IDF score of words to measure *burstiness*. DF-IDF scores of words are significantly higher during a time interval that covers the event than during other time intervals, so we expect the DF-IDF score of a word to peak during the event and be low and stable the rest of the time. The DF-IDF score for cluster $\mathcal{C}$ at interval ℓ is

$$s_{\mathcal{C}}(\ell) = \frac{N_{\mathcal{C}}(\ell)}{N(\ell)} \log \frac{\sum_{i=1}^{T} N(i)}{\sum_{i=1}^{T} N_{\mathcal{C}}(i)} \tag{3}$$

where $N_{\mathcal{C}}(\ell)$ is the number of words from cluster $\mathcal{C}$ that are used in messages from time interval ℓ, summed over all messages and divided by the number of words in $\mathcal{C}$. $N(\ell)$ is the number of messages in time window ℓ. The *burstiness* of cluster $\mathcal{C}$ is given by

$$B(\mathcal{C}) = \frac{\sigma_s(\mathcal{C}) - \mu_s(\mathcal{C})}{\sigma_s(\mathcal{C}) + \mu_s(\mathcal{C})} \tag{4}$$

where μ_s (resp. σ_s) is the average (resp. standard deviation) of $s_{\mathcal{C}}(\ell)$, over ℓ. The *burstiness* index is bounded between -1 and +1, and its magnitude correlates with the signal's burstiness, as bursty signals have a large standard deviation w.r.t. their average (Goh and Barabási, 2008).

2.4 Filtering clusters by word co-occurrence similarity

Each cluster contains words with similar temporal patterns, but these words might discuss different topics. Therefore, we use the co-occurrence similarity to represent the similarity between words at the document level. Words associated with the same event are more likely to be used together. We measure the co-occurrence similarity O_{wm} between words w and m by the cosine similarity of two sparse vectors $\boldsymbol{\delta}_w$ and $\boldsymbol{\delta}_m$, where the kth element of $\boldsymbol{\delta}_w$ is 1 if w appears in message k, and 0 otherwise:

$$O_{wm} = \frac{\sum_k \delta_{wk}\delta_{mk}}{\|\boldsymbol{\delta}_w\|\|\boldsymbol{\delta}_m\|}, \quad \|\boldsymbol{\delta}_{\cdot}\| = \sqrt{\sum_k \delta_{\cdot k}^2}. \tag{5}$$

We remove the noisy words further using hierarchical clustering on the co-occurrence similarity matrix.

- Run hierarchical clustering using co-occurrence similarity matrix $\mathcal{O} = [O_{wm}]$;
- Cut the resulting hierarchy;
- Extract the cluster with maximum number of words as the filtered cluster.

3 Experiments and Results

We tested our method on two social media datasets: the Baltimore dataset, collected from Instagram and the Toronto dataset from Twitter. To detect unspecified events, all messages within the geographical boundary of these two cities were collected during a time period.

3.1 Datasets

The Baltimore dataset contains 385,595 Instagram messages collected in Baltimore, MD, USA from April 1 to May 31, 2015. After removing all non-ASCII characters, URLs, mentions of Instagram users (@username), stop words, and words with certain patterns repeated more than twice (e.g. "booo", "hahahaaa"), there are 358,458 messages and 218,281 unique words left. The Toronto dataset contains 312,836 Twitter messages collected in Toronto from May 17 to May 31, 2018. After removing stop words, there are 231,773 messages and 81,351 unique words left.

3.2 Detected events in the Baltimore dataset

In the Baltimore dataset, we generate the time series for each individual words as their occurrence frequency within six-hours time windows. Since rare words are not likely to be associated with any event, we remove words that appear less than 40 times over all 240 time points, so 8392 unique words are left. We then transform these 8392 time series into images, and generate 100 clusters using *k-means* on features extracted by *ResNet* from these images. The 100 clusters are ranked by burstiness and the top 10 clusters are selected and filtered using hierarchical clustering. Words representing these 10 clusters are listed in Table 1.

In the Baltimore dataset, the major events are related to the 2015 Baltimore protests. They appear in two clusters that correspond to several subsequent events related to the major event. A few local music and culture events are detected as well.

Table 1: Events detected by ICE on Baltimore dataset from 100 clusters.

Words	Event	Date	Burst.
tigolebitties, bigtittyclub, tittietuesday	-	-	0.635
deathmetal, blackmetal, marylanddeathfest,	2015 Maryland deathfest	21-May 24-May	0.267
baltimoretattoo, blackandgreytattoo, baltimoresblackartists	2015 Baltimore Tattoo Arts Convention	10-Apr 12-Apr	0.260
eastersunday, livemusic, jazz, jesus	Easter	01-Apr	0.233
cincodemayo, metgala	-	-	0.219
justiceforblackmeneverywhere, equalrights, bmorestrong, justicefreddie, baltimoreprotests, baltimoreuprising	2015 Baltimore protest	18-Apr 03-May	0.106
prayforbaltimore, mondawmin violence,freddiegrey	Violence students vs. police Mondawmin Mall	27-Apr	0.090
loyolamaryland, graduate afterpartytoallpartys	Loyola University Maryland graduation	19-May	0.077
nationalsiblingsday, baltimoretattooconvention	-	-	0.069
onedirection, imagine, harry, zayn, louis, sauce, fans, london	Onedirection band tour London	12-Apr	0.066

3.3 Detected events in the Toronto dataset

In the Toronto dataset, the time series of word occurrence are obtained using a one-hour time window. After removing words that appear less than 20 times over all time windows, there are 7,095 unique words left and 264 time points. Similarly, we generated 200 clusters from the images of 7,095 words, and the top 10 ranked and filtered clusters are shown in Table 2.

In the Toronto data, several entertainment, sports and political events are detected. The detected events reflect users' interests on social media in these geographical regions.

3.4 Performance in Baltimore and Toronto datasets

Reference events happening in Baltimore (Apr-May 2015) and Toronto (May 2018) are not available, therefore *recall* can not be computed. As a consequence, we use *precision* as the performance measure for event detection, which is consistent with a number of other studies (Farzindar and Khreich, 2015). We used precision on the top ranked detected events to evaluate the performance of ICE. We measure the precision at the top five (P@5) and the top ten (P@10) events in a range of 50 to 1000 clusters on the Baltimore and Toronto datasets (Tables 3, 4). Both of these two datasets achieve a top-5 precision of 80% and top-10 precision of 70%. This indicates that ICE is effective at detecting events from noisy social media messages.

Performance decreases when the number of clusters increases. When there are fewer clusters, each cluster tends to be larger and more likely to contain event-related words. They are also more noisy and may contain mixed events (Tab. 1). On the other hand, with more clusters, clusters are smaller and do not contain mixed events.

4 Discussion

4.1 Transfer learning

Although *ResNet* has been used in transfer learning in many image tasks, transforming time series of words into images and using *ResNet* for event detection is novel as far as we know. Our work differs from Bischke et al. (2016). They used X-ResNet to extract visual features in satellite images for specified event detection. We transform word occurrence frequency into images, and use *ResNet* to extract visual features for unspecified event detection. We adopt the tool for transforming time series to images, and make it possible to use the state-of-art deep neural network architecture for image recognition.

We also tested the use of reduced size Piecewise Aggregation Approximation (PAA) image (Wang and Oates, 2015) as the input for clustering, but the performance on PAA images was very poor.

4.2 Word embedding

In ICE, we use the co-occurrence similarity matrix to filter non-event words in clusters. Co-occurrence similarity represents the semantic similarity between words occurring in the same document. Word embeddings

Table 2: Events detected by ICE on Toronto dataset from 200 clusters.

Words	**Event**	**Date**	**Burst.**
rebuttle, shaw, huxley	-	-	0.327
kindermorgan, pipeline	Kindermorgan pipeline	29-May	0.261
royalwedding, meghanandharry	Royal wedding	19-May	0.256
nobel, pinterest	-	-	0.228
uclfinal, liverpool, championsleaguefinal	UCFL game	26-May	0.220
stanleycupplayoffs, goldenknights, savelucifer	2018 Stanley Cup game Fox's cancellation of "Lucifer" show	28-May	0.210
roseanne's, cancelling	Roseanne show cancelling	29-May	0.173
terryfoxrun	-	-	0.116
fireworks, victoriaday,	Victoria day	21-May	0.106
trudeaumustgo, saveontario	2018 Ontario General Election	26-May	0.055

Table 3: Performance of ICE on the Baltimore dataset.

#clusters	avg #words in cluster	filtered	P@5 (%)	P@10 (%)
50	161.0	134.8	**80**	**70**
100	92.8	74.0	60	**70**
200	40.7	26.7	40	50
1000*	9.6	9.6	40	50

* Clusters with less than 11 words are not filtered

Table 4: Performance of ICE on the Toronto dataset.

#clusters	avg #words in cluster	filtered	P@5 (%)	P@10 (%)
100	74.2	63.5	**80**	**70**
200	34.9	26.0	60	**70**
300*	27.1	20.7	60	60

* Clusters with less than 11 words are not filtered

have been widely used in many NLP applications. We therefore tested the combination of the temporal features extracted from images and semantic features obtained from word embeddings. We first used the 100- and 200-dimension of GloVe word embeddings (Pennington et al., 2014) pre-trained on Twitter data, together with image features. It shows that the performance of using the 100-dimension GloVe embedding with image features is worse than image features alone, and using the 200 dimension GloVe embeddings does not result in any detected event (Table 5). We also trained 200-dimension fastText word embeddings (Mikolov et al., 2017) on the Baltimore dataset, and combined them with image features. The result shows that using fastText word embedding trained on Baltimore data does not hurt or help the overall performance. Overall, the use of word embeddings simultaneously with temporal features does not perform particularly well.

Table 5: Using word embedding for clustering

Features	#clusters	P@5	P@10
Image+100d(GloVe)	100	60	60
Image+200d(GloVe)	100	0	0
Image+200d(fastText)	100	60	70

4.3 Parameter Analysis

The parameters used in ICE include the time window ℓ and the number of clusters $|\mathcal{C}|$. During the cluster filtering step, we use hierarchical clustering, and the largest branch of the clustering tree is chosen to represent events, which does not introduce additional parameters.

As discussed earlier, we keep ℓ as small as possible to gain granularity of clusters. The only tuned parameter in ICE is the number of clusters $|\mathcal{C}|$. As shown in Tables 3–4, increasing $|\mathcal{C}|$ naturally results in a decrease of the number of words in each clusters. We also observed that precision dropped as the number of clusters increases, although this effect was more pronounced with the Baltimore dataset (Table 3). Overall, these results suggest that ICE is relatively robust to mild differences in parameter settings when it comes to detecting relevant events.

5 Conclusions

Event detection from social media is a challenging task due to the noisy nature of user generated text. In this study, we propose a novel method, transforming the time series of word occurrence frequency into images, and using *ResNet* to extract features from images. The images are clustered, clusters are ranked by *burstiness*, and words in each clusters are filtered using the co-occurrence similarity within messages. Converting word occurrence into images allows to capture the dynamic changes in the social media environment. Clustering words with similar temporal patterns using features extracted by advanced convolutional neural network architecture *ResNet* provides a robust method for separating real event from daily chatter on social media. The subsequent steps of ranking and filtering clusters refines the detected events.

Note that our method is not an end-to-end event detection method. End-to-end systems ususally need large amounts of training samples, which are not available for unspecified event detection. For future work, we would like to explore how to combine the temporal patterns and co-occurrence patterns in images and improve the ranking of longer events.

6 Acknowledgments

We would like to thank Yuanjing Cai for writing the code for the burstiness index and co-occurrence matrix used in our method during her co-op term at NRC.

References

B. Bischke, D. Borth, C. Schulze, and A. Dengel. Contextual enrichment of remote-sensed events with social media streams. In *Proc. 24th ACM intl. conf. on Multimedia*, pages 1077–1081, 2016.

J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. S. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *IEEE Conf. on Visual Analytics Science and Technology (VAST)*, pages 143–152, 2012.

A. Farzindar and W. Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.

K.-I. Goh and A.-L. Barabási. Burstiness and memory in complex systems. *Europhysics Letters*, 81(4): 48002, 2008.

M. Hasan, M. A. Orgun, and R. Schwitter. A survey on real-time event detection from the twitter data stream. *Journal of Information Science*, 44(4):443–463, 2018. doi: 10.1177/0165551517698564.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE conf. on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

K. Lee, A. Qadir, S. A. Hasan, V. Datla, A. Prakash, J. Liu, and O. Farri. Adverse drug event detection in tweets with semi-supervised convolutional neural networks. In *Proc. 26th Intl. Conf. on the World Wide Web*, pages 705–714, 2017.

C. Li, A. Sun, and A. Datta. Twevent: segment-based event detection from tweets. In *Proc. 21st ACM intl. conf. on Information and Knowledge Management*, pages 155–164, 2012.

T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin. Advances in pre-training distributed word representations. *arXiv:1712.09405*, 2017.

M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. IEEE conf. on Computer Vision and Pattern Recognition*, pages 1717–1724, 2014.

J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

A. Pozdnoukhov and C. Kaiser. Space-time dynamics of topics in streaming text. In *Proc. 3rd ACM SIGSPATIAL intl. workshop on Location-Based Social Networks*, pages 1–8, 2011.

T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proc. 19th intl. conf. on World Wide Web*, pages 851–860, 2010.

E. Schubert, M. Weiler, and H.-P. Kriegel. Signitrend: scalable detection of emerging topics in textual streams by hashed significance thresholds. In *Proc. 20th ACM SIGKDD intl. conf. on Knowledge Discovery and Data Mining*, pages 871–880, 2014.

A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proc. IEEE conf. on Computer Vision and Pattern Recognition workshops*, pages 806–813, 2014.

Z. Wang and T. Oates. Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Workshops at the 29th AAAI Conf. on Artificial Intelligence*, 2015.

J. Weng and B.-S. Lee. Event detection in twitter. In *Proc. Intl. Conf. on Web and Social Media*, volume 11, pages 401–408, 2011.

X. Zhou and L. Chen. Event detection over twitter social media streams. *The International Journal on Very Large Data Bases*, 23(3):381–400, 2014.