Clustering Hashtags Using Temporal Patterns

Borui Cai[1], Guangyan Huang[1(✉)], Shuiqiao Yang[2], Yong Xiang[1], and Chi-Hung Chi[3]

[1] School of Information Technology, Deakin University, Melbourne, Australia
{bcai,guangyan.huang,yong.xiang}@deakin.edu.au
[2] School of Computer Science, University of Technology Sydney, Ultimo, Australia
shuiqiao.yang@uts.edu.au
[3] Data61, CSIRO, Sydney, Australia
chihung.chi@csiro.au

Abstract. Twitter hashtags provide a high-level summary of tweets, while cluster hashtags have many applications. Existing text-based methods (relying on explicit words in tweets) are greatly affected by the sparsity of the short tweet texts and the low co-occurrence rates of hashtags in tweets. Meanwhile, semantically related hashtags but using different text-expressions may show similar temporal patterns (i.e., the frequencies of hashtag usages changing with the time), which can help capture events, opinions and synonyms. In this paper, we propose a novel clustering hashtags by their temporal patterns (CHTP) method as a complement to text-based methods. In CHTP, hashtags are represented as hashtag time series that show their temporal patterns, so, hashtag clusters can be discovered by clustering hashtag time series. Density-based clustering algorithms are suitable to discover naturally shaped hashtag clusters but they are not fine enough (use one distance threshold to define density) to differentiate clusters of various density levels. Therefore, we develop a new parameter-free Density-Sensitive Clustering (DSC) algorithm to discover clusters of different density levels and use it in CHTP to group hashtags by temporal patterns. DSC recursively partitions the dataset from coarse-grained to fine-grained (using adaptive distance thresholds) to discover hashtag clusters of different density levels. Experiments conducted on Twitter datasets show that the DSC algorithm finds hashtag clusters of different densities more effectively than counterpart methods, and CHTP (using DSC) can discover meaningful hashtag clusters, 36% of which cannot be found by the text-based approaches.

Keywords: Hashtag · Time series clustering · Cluster density

1 Introduction

Twitter is a popular web microblogging and social networking service, on which users interact and share information with short messages (tweets). A twitter

Z. Huang et al. (Eds.): WISE 2020, LNCS 12342, pp. 183–195, 2020.
https://doi.org/10.1007/978-3-030-62005-9_14

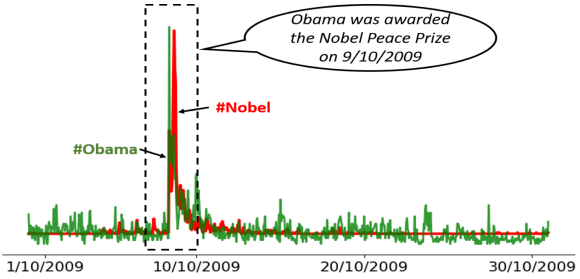


Fig. 1. The similar time series of #Obama and #Nobel can help detect the event that Obama was awarded the Nobel Peace Prize on 9/10/2009.

hashtag is a meta-tag that the user creates for classifying tweets based on their meanings and subjects. For example, tweets tagged with the hashtag #Jobs are related to job opportunities. This high-level summary/label information explicitly provided by users makes hashtag promising in finding dynamic relationships of tweets [2]. Hashtag clustering that finds semantically related hashtags can be used for event discovery [13,16], opinion extraction and synonym detection.

Most existing methods discover hashtag clusters using the tweet texts [13,14]. These text-based methods represent hashtags using explicit words in tweet texts. For example, with the bag-of-words (BoW) representation [13], one hashtag is represented by the multiplicity of the co-occurring words in the tweets. However, the short tweet texts may not reveal the whole hashtag relationships. First, many tweets may report the same event but use totally different words. Second, hashtag co-occurrence is not always reliable since many tweets have no more than two hashtags. Affected by these problems, text-based methods only can extract part of the information revealed by hashtags. The recent method [12] uses the temporal pattern of hashtags related to known events, i.e., time series that record the frequencies of hashtag usages changing with the time, to discover other related hashtags. Inspired by this, we use hashtag time series (which show temporal pattern) to cluster hashtags, because the frequencies of certain hashtag usages are highly correlated with the popularity of the corresponding events/topics. For example, in Fig. 1, the hashtag time series of #Obama and #Nobel can help discover the event that Obama was awarded the Nobel Peace Prize.

In this paper, we propose a novel clustering hashtags by their temporal patterns (CHTP) method. In CHTP, hashtags are represented as hashtag time series, so, clusters can be discovered by clustering hashtag time series. However, clustering hashtag time series faces two challenges. First, it is impractical to manually determine the number of hashtag clusters, because a Twitter dataset includes world-wide events/topics. Second, these hashtag clusters are naturally shaped and follow no specific distributions. Therefore, density-based time series clustering is preferred in CHTP since it can find arbitrarily shaped clusters. Many density-based clustering methods require single distance thresholds to define density of data objects. Unfortunately, specifying a suitable distance threshold is dif-

ficult because there are clusters of different density levels in the complex Twitter datasets. Thus, we propose a novel parameter-free Density-Sensitive Clustering (DSC) algorithm for CHTP to find hashtag clusters of different density levels. DSC is a density-based recursive partition process, and it discovers a hashtag time series group as a hashtag cluster if it cannot be split into multiple subgroups in lower recursions. Comprehensive experiments have been conducted on Twitter datasets to demonstrate the effectiveness of the proposed CHTP method, which uses DSC for hashtag clustering. Therefore, the contributions of this paper are listed as follows:

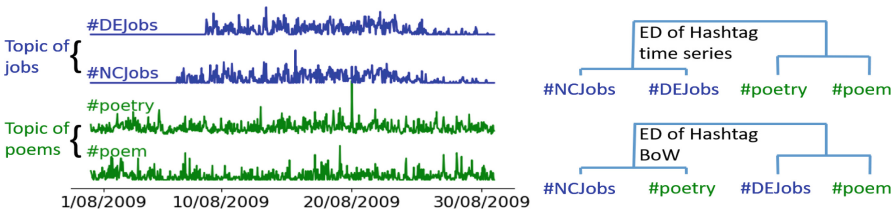
1) We provide a novel CHTP method for hashtag clustering, which groups hashtags using the common temporal patterns, rather than the tweet texts.
2) We develop a new parameter-free Density-Sensitive Clustering (DSC) algorithm for CHTP to discover hashtag clusters of different densities.
3) Comprehensive experiments conducted with four Twitter datasets show that averagely 36% of meaningful hashtag clusters discovered by CHTP (using DSC) cannot be found by text-based approaches; and the proposed DSC algorithm is more effective than the counterpart algorithms to find hashtag clusters.

The rest of this paper is organized as follows. In Sect. 2, we review the related work. We detail the CHTP method in Sect. 3, and then develop the DSC algorithm in Sect. 4. The proposed method is evaluated in Sect. 5. The paper is concluded in Sect. 6.

2 Related Work

Hashtag is widely used in tweets and can represent summary information of tweets to extract useful information. Most hashtag clustering methods discover hashtag clusters by the tweet texts tagged with the hashtags (text-based methods). SMSC [13] uses the bag-of-words (BoW) representation of co-occurred tweet texts with Kmeans to find hashtag clusters. Meanwhile, topic model is used in HGTM [14] to cluster hashtags by analyzing a hashtag graph built with the co-occurrence information of hashtags. The method in [7] further integrates lexical and contextual text information to improve clustering performance. In addition to tweet texts, external knowledge, i.e. the semantics of hashtags obtained from WordNet [8] or Wikipedia [6], is utilized to improve the accuracy of hashtag clustering.

Other than the text-based methods, SAX* [12] uses temporal similarity as the semantic relatedness for hashtags, and it detects hashtag clusters with string patterns deduced from external sources (e.g. Wikipedia) [9]. We extend this idea and discover hashtag clusters from their temporal patterns by clustering hashtag time series. We briefly review the recent time series clustering methods, upon which the proposed method is built developed. Partition-based time series clustering, such as kshape [10] and KSC [15], find cluster representatives and minimize the distances of time series to nearest representatives. However, they adopt



(a) Hashtag time series. (b) Single-linkage clustering.

Fig. 2. Four hashtag time series (as shown in (a)) obtained from a Twitter corpus [15]. (b) shows single-linkage clustering with Euclidean distance on hashtag time series (top) and hashtag BoW representation (bottom), respectively.

spherical-shape clusters that are sensitive to outliers and noise in the complex Twitter datasets. Model-based methods. e.g. Gaussian Mixture Model [4] and Gaussian Inverse Covariance [5], cluster time series by the optimizations of specific models, but a model that well-explain the large Twitter datasets may have impractical complexity. Density-based clustering methods are preferred since they can find natural-shaped clusters. For example, YADING [3] hierarchically adopts DBSCAN to find time series clusters of different densities; however, the inflection points (on the distance-to-nearest-neighbours curve) YADING used to determine densities levels are not significant in the sparse Twitter datasets. TADPole [1] groups time series into clusters using the density and the distance by DPC [11]; but that makes TADPole hard to find clusters of different density levels since a global distance threshold is applied in the entire dataset.

3 Problem Definition

In this section, we present the CHTP method, and it comprises two steps:

1) Represent hashtags as hashtag time series. (see Sect. 3.1)
2) Discover hashtag clusters by clustering hashtag time series. (see Sect. 3.2)

3.1 Hashtag Time Series

CHTP discovers hashtag clusters by hashtag temporal patterns, and each hashtag is represented as a time series denoted as $Z = \{z_1, z_2, ..., z_m\}$, where z_j is the frequency of a hashtag at time bin j (for example, one hour [15]). A hashtag time series dataset DS is a collection of hashtag time series denoted as $DS = \{Z_1, Z_2, ...Z_n\}$. Each hashtag time series is preprocessed to be scale-invariant by applying z-normalization as follows:

$$x_j = \frac{z_j - \mu}{\delta} \ (1 \leq j \leq m), \tag{1}$$

where $\mu = \frac{1}{m}\sum z_j$ and $\delta = \sqrt{\frac{\sum(z_j-\mu)^2}{m-1}}$. The distance of two normalized hashtag time series, X and Y, is measured by Euclidean distance ($ED(X,Y)$).

We show four hashtag time series as an example in Fig. 2 (a). By applying single-linkage clustering with Euclidean distance, two clusters are correctly discovered (the top in Fig. 2 (b)). However, with text-based hashtag BoW representation, unrelated hashtags are grouped together (the bottom in Fig. 2 (b)).

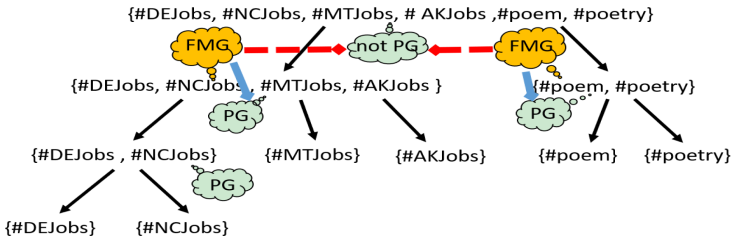


Fig. 3. The depth-first recursive partition of example hashtag time series.

3.2 Clustering by Hashtag Time Series

Specifying the number or the shape of clusters in a Twitter dataset is difficult since tweets posted world-widely cover enormous events/topics, and thus density-based clustering that discovers natural-shaped clusters is favored. Existing density-based time series clustering algorithms, such as TADPole [1], face a major challenge to cluster hashtag time series, i.e. the global distance threshold they used to define density cannot differentiate density levels. That may undermine the clustering accuracy, because a large distance threshold may group irrelevant hashtags into clusters; while a small threshold cannot discover sparse clusters with small densities. Therefore, CHTP demands a density-based clustering algorithm that is adaptable to different densities of hashtag clusters.

4 The DSC Algorithm

In this section, we present the DSC algorithm used in CHTP to cluster hashtag time series. DSC discovers clusters of different density levels by adaptive distance thresholds. In general, DSC partitions the dataset recursively from the coarse-grained to the fine-grained (in a depth-first manner), with adaptive distance thresholds, and clusters of different density levels are discovered as *full pure groups*. A *full pure group* contains highly correlated time series that are less relevant with time series in other *full pure groups*. Before explaining *full pure group*, we first define *pure group*.

Definition 1. *Pure group (PG): a group of time series that cannot be split into multiple subgroups by density, i.e., each partition produces at most one subgroup (contains more than one time series) during the recursive partitioning.*

Examples of PG are shown in Fig. 3, which records the recursive partition process of six hashtags. Apparently, groups with 2 or 3 time series are PGs. Based on the definition, the partitioned subgroups of a PG are also PGs but locates at different hierarchies on the partition tree. PGs on the top hierarchies (as *full pure group*) are regarded as the expected clusters.

Definition 2. *Full pure group (FPG): full pure group is the pure group of the top hierarchy, i.e., full pure group must be partitioned from a non-PG.*

Based on Definition 2, a subgroup of time series (SG) is a FPG only if 1) SG is a PG and 2) S (the group that SG is partitioned from) is not a PG (see the examples in Fig. 3). DSC discovers clusters as FPGs by analyzing the recursive process that partitions the dataset, and now we detail the DSC algorithm.

4.1 Data Partition

In DSC, a recursive *DataPartition* algorithm is used to partition the dataset (DS) by density, and the input dataset for each partition process in the recursion is denoted as S. *DataPartition* has two components, i.e. the group forming function (*FormGroup*) that partitions S by forming subgroups and the cluster detector that indicates whether a subgroup is a FPG. *FormGroup* forms subgroups by density, and one time series, X, finds a set of neighbours (N_X) as follows:

$$N_X = \{Y : ED(X,Y) < d_c, Y \in S\}, \tag{2}$$

where d_c is the adaptive distance threshold (will be discussed later). The density of X, ρ_X, is calculated as follows:

$$\rho_X = \sum \left(1 - \frac{ED(X,Y)^2}{d_c^2}\right), Y \in N_X. \tag{3}$$

We borrow the idea of DPC [11] to group time series by density, that is, X is density connected to a specific neighbour (n_X) as follows:

$$n_X = \underset{Y:\rho_Y > \rho_X, Y \in N_X}{\arg\min} ED(X,Y), \tag{4}$$

X is the centre (C) of a group if $n_X = null$, which means X has the local maximum density among its neighbours. Groups are developed by two steps: 1) each C is assigned a unique group label; 2) starting from centers, the group labels are spread from n_X to X in the decreasing order of density. After all time series acquire group labels from relative centers, S is partitioned into subgroups (SG) for further partitioning.

4.2 Adaptive Distance Threshold

We especially expect the adaptive d_c to satisfy the following two requirements:

1) *FormGroup* can always partition S into subgroups or individuals, which ensures the termination of the recursive partition.
2) Partitions are conducted from the coarse-grained to the fine-grained. This ensures sparse clusters are not omitted.

We use Minimum Spanning Tree (MST) built with S to find the adaptive d_c. Nodes of MST are the time series in S and the edges are the time series distances. Then, d_c of *FormGroup* is assigned as the largest edge on the MST. We show that d_c satisfies the first requirement by proving that at least two time series are split after partitioning. Assume $E_{ij} = ED(X_i, X_j)$ is the longest edge of MST built with S, and we assign $d_c = E_{ij}$ for *FormGroup*.

Algorithm 1. DataPartition.

Input: Hashtag time series dataset/subset S, *Clusters*
MST = $BuildMST(S)$
d_c = largest edge of MST (based on Lemma 1 and Lemma 2)
$Groups = FormGroup(S, d_c)$, $Subgroups = \{\}$, PG_S = False
for each $SG \in Groups$ **do**
  if $|SG| > 1$ **then** $Subgroups = Subgroups \cup \{SG\}$
if $|Subgroups| = 0$ or $|Subgroups| = 1$ **then** PG_S = True
for each $SG \in Subgroups$ **do**
  PG_{SG} = DataPartition$(SG, Clusters)$
  if PG_{SG} = False **then** PG_S = False
  if PG_{SG} = True and PG_S = False **then** $Clusters = Clusters \cup SG$
Output: PG_S

Lemma 1. *X_i and X_j do not stay in the same subgroup after S is partitioned by FormGroup, with $d_c = E_{ij}$.*

Proof. MST is split into $LMST = \{X_{l_1}, ..., X_{l_n}\}$ and $RMST = \{X_{r_1}, ..., X_{r_n}\}$ after removing E_{ij}, and $X_i \in LMST$ and $X_j \in RMST$. Now assume X_i and X_j still stay in the same subgroup. Since $d_c = E_{ij}$, $X_j \notin N_{X_i}$, therefore, X_i and X_j must be connected through $\{X_{\eta_0} = X_i, X_{\eta_1}, ..., X_{\eta_{k-1}}, X_{\eta_k} = X_j\}$, in which $E_{\eta_{t-1},\eta_t} < d_c, \forall t \in \{1, ..., k\}$. Since $X_l \in LMST$ and $X_r \in RMST$, $\exists t \in \{1, ..., k\}$ *s.t.* $X_{\eta_{t-1}} \in LMST$ and $X_{\eta_t} \in RMST$, and thus the spanning tree by connecting $LMST$ with $RMST$ with $E_{\eta_{t-1}\eta_t}$ ($<E_{ij}$) has a smaller weights than MST, which is impossible and the assumption is wrong. It is proved.

Now we show that the second requirement is also satisfied due to Lemma 2.

Lemma 2. *The largest edge on MST of SG (one subgroup of S grouped by FormGroup) is always smaller than the largest edge on MST of S.*

Proof. MST_{SG} (the MST of SG) comprises several subtrees, i.e., $\text{MST}_{SG} = \{ST_1, ..., ST_k\}$ ($ST_i \in \text{MST}_{SG}$ and $ST_i \cap ST_j = \emptyset, \forall i, j \in [1, k]$). In MST_{SG}, $\{ST_1, ..., ST_k\}$ are connected with the $k-1$ edges among them. Assume the largest edge on MST_{SG} (E_{ab}, which is also d_c^{SG}) connects ST_a and ST_b, thus $E_{ab} = \min_{X_l \in ST_a, X_r \in ST_b} ED(X_l, X_r)$. Since SG is grouped by $FormGroup$ on S with d_c^S (the largest edge on MST_S), $\exists ED(X_l, X_r) < d_c^S$ *s.t.* $X_l \in ST_a, X_r \in ST_b$. Therefore, $d_c^{SG} = E_{ab} \leq ED(X_l, X_r) < d_c^S$. It is proved.

The pseudo-code of *DataPartition* is shown in Algorithm 1. d_c is assigned as the longest edge on MST at lines 1–2. S is partitioned at line 3, and the obtained *subgroup*s are further partitioned at line 8. The clusters are discovered as FPGs is shown at lines 8–10.

4.3 Complexity Analysis

The complexity of *DataPartition* for the initial dataset (DS) is $O(n \log n + 2n^2)$, including building MST from DS for $O(n^2)$ and adopting $FormGroup$ for $O(n \log n + n^2)$. Meanwhile, the recursion depth (γ) is usually much smaller than n, because $FormGroup$ can partition the dataset efficiently with a well-designed d_c. Therefore, the complexity of DSC is $O(\gamma n \log n + \gamma n^2)$.

5 Evaluation

In this section, we evaluate the performance of the proposed CHTP method (hashtag time series clustering by DSC) and the DSC clustering algorithm by answering the following two questions:

Table 1. Statistics of the 4 Twitter datasets.

Dataset	Hashtags	Tweets (million)	Date of Tweets
Aug/2009	1875	10.6	1/8/2009–31/8/2009
Sep/2009	1462	7.0	1/9/2009–30/9/2009
Oct/2009	1181	5.4	1/10/2009–31/10/2009
Nov/2009	791	3.4	1/11/2009–30/11/2009

1) Q1: Can DSC find hashtag clusters more effectively than the counterpart TADPole and YADING? (Sect. 5.1)
2) Q2: Is it necessary to use CHTP to cluster hashtag when the text-based approaches exist? (Sect. 5.2)

All algorithms are implemented with python 2.7, and the experiments are run on a Windows 10 platform with 3.4 GHz CPU and 16 GB RAM.

Datasets: We use a Twitter corpus [15] that comprises tweets posted from August to November 2009, for our experiments. These tweets are split into 4 subsets by the months they were posted. In each subset, we select important hashtags (with frequency larger than 1000) along with the tweets they appeared to generate the corresponding dataset. The time bin is specified as one hour following [15]. The statistics of the datasets are shown in Table 1.

Counterparts for DSC: TADPole [1] and YADING [3], two density-based time series algorithms as discussed in Sect. 2, are compared with DSC in experiment 1 (see Sect. 5.1) to answer Q1. To be fair, distance measurements used are unified as Euclidean distance in TADPole, YADING and DSC. **Counterparts for CHTP**: We explicitly choose SMSC [13], which adopts the hashtag BoW (text based) and Kmeans, as the counterpart of CHTP in experiment 2 to answer Q2. We also use the BoW representation with DSC, i.e., TextDSC, to directly compare with CHTP. Besides DSC, TextDSC and YADING, which are parameter-free, the results of TADPole and SMSC are obtained under the optimal parameters that maximize Silhouette Coefficient. Clusters that contain multiple hashtags are used for comparison.

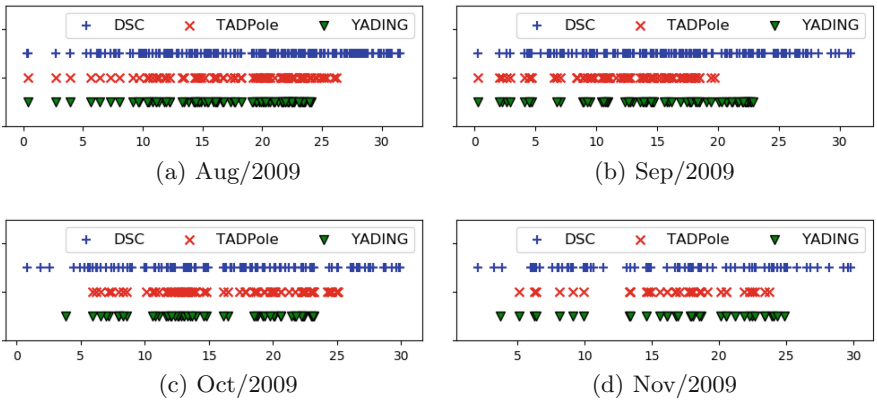
Table 2. Clustering accuracy.

Dataset	TADPole	YADING	DSC
Aug/2009	0.53	0.52	**0.79**
Sep/2009	0.62	0.51	**0.78**
Oct/2009	0.67	0.54	**0.76**
Nov/2009	0.71	0.54	**0.74**
Average	0.63	0.53	**0.77**

5.1 Accuracy and Effectiveness of DSC on Hashtag Time Series

In this experiment, we compare DSC with TADPole and YADING for clustering hashtag time series. We evaluate the clustering accuracy by validating each hashtag cluster by its contained hashtags since manually labeling the hashtags is impractical. A cluster is *valid* only if the contained hashtags are synonyms/abbreviations or represent an event/topic (see Fig. 1) in the search results from Google. The accuracy of hashtag clustering is measured by F_1 score.

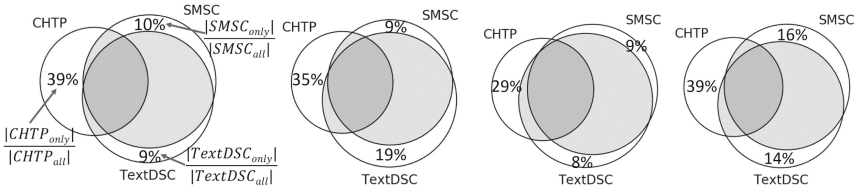
The clustering accuracy results are shown in Table 2. DSC, TADPole and YADING achieve fair clustering results in the datasets and the average accuracy are 0.77, 0.63 and 0.53, respectively. Specifically, DSC out-performs TADPole and YADING in all the 4 datasets, and the average improvements of accuracy to TADPole and YADING are around 22% and 45%, respectively.



(a) Aug/2009 (b) Sep/2009

(c) Oct/2009 (d) Nov/2009

Fig. 4. Density distribution of hashtag clusters discovered by DSC, TADPole and YADING, respectively.



(b) Aug/2009 (c) Sep/2009 (d) Oct/2009 (e) Nov/2009

Fig. 5. Relationships of hashtag clusters discovered by CHTP, SMSC and TextDSC, and shaded area indicates similar clusters shared by multiple methods.

We compare the densities of *valid* hashtag clusters discovered by DSC, TADPole and YADING, with the density of a cluster being the average nearest-neighbour-distance of the contained hashtags. The larger the average nearest-neighbour-distance, the smaller the density, and the results are shown in Fig. 4. Generally, DSC finds more *valid* hashtag clusters than TADPole and YADING in all the 4 Twitter datasets, and YADING discovers the least *valid* hashtag clusters, which results in its low *recall*.

5.2 CHTP vs. Text-Based Approaches

In this experiment, we analyze the significance of hashtag clustering by temporal pattern through showing whether the text-based methods (TextDSC and SMSC) can discover similar *valid* hashtag clusters as CHTP. We regard two *valid* hashtag clusters (discovered by different methods) are similar if they shares more than three fourth hashtags. Then we summarize the relationship of *valid* hashtag clusters discovered by CHTP, TextDSC and SMSC as shown in Fig. 5.

The results show that, in all the 4 Twitter datasets, around 36% of the *valid* hashtag clusters discovered by CHTP cannot be discovered by TextDSC/SMSC,

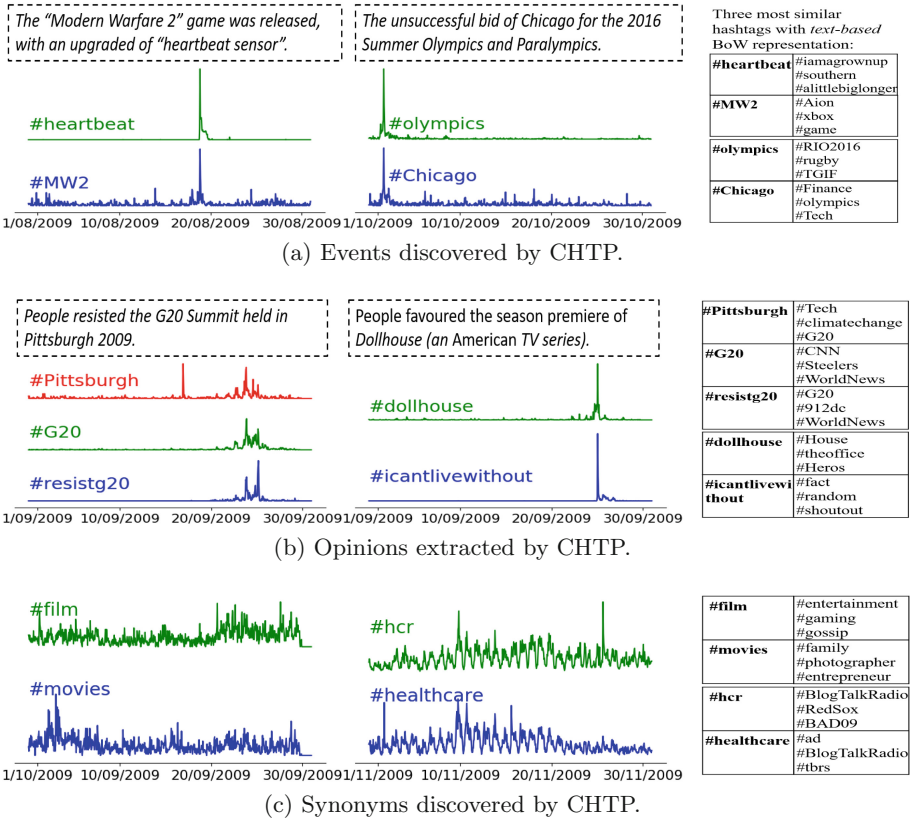


Fig. 6. Examples of events (a), opinions (b) and synonyms (c) discovered by CHTP, but SMSC and TextDSC fail to find. The right column shows the nearest hashtags with hashtag BoW representation.

even when using the same clustering algorithm (DSC). This result suggests that the hashtag temporal pattern partially represent distinctive correlations of hashtags compared with the tweet texts, and CHTP can supplement hashtag clustering that only uses tweet texts. In contrast, TextDSC and SMSC share many hashtag clusters since they use the same clue, i.e., the tweet texts.

We show some examples of hashtag clusters only discovered by CHTP to further understand the hashtag clusters discovered by temporal patterns in Fig. 6. That includes two events discovered only by CHTP (Fig. 6 (a)), two examples of extracted opinion (Fig. 6 (a)) and two examples of discovered synonyms (Fig. 6 (c)). To compare with the text-based BoW representation, we show the three most similar hashtags (with BoW) of the hashtags in the right-side column of Fig. 6. The result shows that hashtags having similar temporal patterns are not quite similar in their BoW representations, and that is the reason that TextDSC/SMSC fails to discover the hashtag clusters.

6 Conclusion

In this paper, we propose a novel CHTP method to discover hashtag clusters by hashtag temporal patterns. CHTP represents hashtags as hashtag time series and uses the proposed DSC algorithm (which can discover clusters of different density levels) to effectively cluster hashtag time series. Experiments conducted on Twitter datasets show that DSC is more effective in discovering hashtag clusters than two counterpart algorithms, and CHTP (uses DSC) can discover 36% hashtag clusters that cannot be discovered by the text-based approaches. Therefore, we conclude that by using the temporal pattern of hashtags, a more complete understanding of the relationship of hashtags can be obtained.

Acknowledgments. This work was partially supported by Australia Research Council (ARC) DECRA Project (DE140100387) and Discovery Project (DP190100587).

References

1. Begum, N., Ulanova, L., Wang, J., Keogh, E.: Accelerating dynamic time warping clustering with a novel admissible pruning strategy. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 49–58. ACM (2015)
2. DeMasi, O., Mason, D., Ma, J.: Understanding communities via hashtag engagement: a clustering based approach. In: Tenth International AAAI Conference on Web and Social Media (2016)
3. Ding, R., Wang, Q., Dang, Y., Fu, Q., Zhang, H., Zhang, D.: YADING: fast clustering of large-scale time series data. Proc. VLDB Endow. **8**, 473–484 (2015)
4. Hallac, D., Nystrup, P., Boyd, S.: Greedy Gaussian segmentation of multivariate time series. Adv. Data Anal. Classif. **13**, 727–751 (2019)
5. Hallac, D., Vare, S., Boyd, S., Leskovec, J.: Toeplitz inverse covariance-based clustering of multivariate time series data. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 215–223. ACM (2017)
6. Javed, A., Lee, B.S.: Sense-level semantic clustering of hashtags. In: Lossio-Ventura, J.A., Alatrista-Salas, H. (eds.) SIMBig 2015–2016. CCIS, vol. 656, pp. 1–16. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55209-5_1
7. Javed, A., Lee, B.S.: Hybrid semantic clustering of hashtags. Online Soc. Netw. Media **5**, 23–36 (2018)
8. Li, T., Wu, Y., Zhang, Y.: Twitter hash tag prediction algorithm. In: Proceedings on the International Conference on Internet Computing (ICOMP), pp. 1–5 (2011)
9. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: a novel symbolic representation of time series. Data Min. Knowl. Discov. **15**, 107–144 (2007)
10. Paparrizos, J., Gravano, L.: Fast and accurate time-series clustering. ACM Trans. Database Syst. (TODS) **42**, 1–49 (2017)
11. Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**, 1492–1496 (2014)
12. Stilo, G., Velardi, P.: Hashtag sense clustering based on temporal similarity. Comput. Linguist. **43**, 181–200 (2017)

13. Tsur, O., Littman, A., Rappoport, A.: Efficient clustering of short messages into general domains. In: Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013, pp. 621–630 (2013)
14. Wang, Y., Liu, J., Qu, J., Huang, Y., Chen, J., Feng, X.: Hashtag graph based topic model for tweet mining. In: 2014 IEEE International Conference on Data Mining (ICDM), pp. 1025–1030. IEEE (2014)
15. Yang, J., Leskovec, J.: Patterns of temporal variation in online media. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 177–186. ACM (2011)
16. Zhong, Z., Zhang, Y., Pang, J.: A graph-based approach to explore relationship between hashtags and images. In: Cheng, R., Mamoulis, N., Sun, Y., Huang, X. (eds.) WISE 2020. LNCS, vol. 11881, pp. 473–488. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34223-4_30