DancingLines: An Analytical Scheme to Depict Cross-Platform Event Popularity

Tianxiang Gao[1], Weiming Bao[1], Jinning Li[1], Xiaofeng Gao[1(✉)], Boyuan Kong[2], Yan Tang[3], Guihai Chen[1], and Xuan Li[4]

[1] Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
{gtx9726,wm_bao,lijinning}@sjtu.edu.cn, {gao-xf,gchen}@cs.sjtu.edu.cn
[2] University of California, Berkeley, CA, USA
boyuan_kong@berkeley.edu
[3] Hohai University, Nanjing, China
tangyan@hhu.edu.cn
[4] Baidu, Inc., Beijing, China
xli@baidu.com

Abstract. Nowadays, events usually burst and are propagated online through multiple modern media like social networks and search engines. There exists various research discussing the event dissemination trends on individual medium, while few studies focus on event popularity analysis from a cross-platform perspective. In this paper, we design DancingLines, an innovative scheme that captures and quantitatively analyzes event popularity between pairwise text media. It contains two models: TF-SW, a semantic-aware popularity quantification model, based on an integrated weight coefficient leveraging Word2Vec and TextRank; and ωDTW-CD, a pairwise *event popularity time series* alignment model matching different event phases adapted from Dynamic Time Warping. Experimental results on eighteen real-world datasets from an influential social network and a popular search engine validate the effectiveness and applicability of our scheme. DancingLines is demonstrated to possess broad application potentials for discovering knowledge related to events and different media.

Keywords: Cross-platform analysis · Data mining
Time series alignment

This work has been supported in part by the Program of International S&T Cooperation (2016YFE0100300), the China 973 project (2014CB340303), the National Natural Science Foundation of China (Grant number 61472252, 61672353), the Shanghai Science and Technology Fund (Grant number 17510740200), CCF-Tencent Open Research Fund (RAGR20170114), and Key Technologies R&D Program of China (2017YFC0405805-04).

S. Hartmann et al. (Eds.): DEXA 2018, LNCS 11029, pp. 283–299, 2018.
https://doi.org/10.1007/978-3-319-98809-2_18

1 Introduction

In recent years, the primary media for information propagation have been shifting to online media, such as social networks, search engines, web portals, etc. A vast number of studies have been conducted to analyze the event disseminations comprehensively on single medium [11,12,23]. In fact, an event is less likely to be captured only by single platform, and popular events are usually disseminated on multiple media.

We model the event dissemination trends as *Event Popularity Time Series* (EPTS) at any given temporal resolution. Inspired by the observation that the diversity of the media and their mutual influences cause the EPTSs to be temporally warped, we seek to identify the alignment between pairwise EPTSs to support deeper analysis.

We propose a novel scheme called DancingLines to depict event popularity from pairwise media and quantitatively analyze the popularity trends. DancingLines facilitates cross-platform event popularity analysis with two innovative models, TF-SW (Term Frequency with Semantic Weight) and ωDTW-CD (ωeighted Dynamic Time Warping with Compound Distance).

TF-SW is a semantic-aware popularity quantification model based on Word2Vec [16] and TextRank [15]. The model first discards the words unrelated to certain events; then utilizes semantic and lexical relations to get similarity between words and highlights the semantically related ones with a *contributive words* selection process. Finally based on similarity, TextRank gives us the importance of each word, then the popularity of a certain event. EPTSs generated by TF-SW are able to capture the popularity trend of a specific event at different temporal resolutions.

ωDTW-CD is a pairwise EPTSs alignment model using an extended Dynamic Time Warping method. It generates sequence of matches between temporally warped EPTSs.

Experimental results on eighteen real-world datasets from Baidu, the most popular search engine in China, and Weibo, Chinese version of Twitter, validate the effectiveness and applicability of our models. We demonstrate that TF-SW is in accordance with real trends and sensitive to burst phases, and that ωDTW-CD successfully aligns EPTSs. The model not only gives an excellent performance, but also shows superior robustness. In all, DancingLines has broad application potentials to reveal knowledge of various aspects of cross-platform events and social media.

The rest of this paper is organized as follows. In Sect. 2, related work is discussed. In Sect. 3, we define the problem. In Sect. 4, we introduce the overview of DancingLines. The two models TF-SW and ωDTW-CD are discussed in details respectively in Sects. 5 and 6. Section 7 verifies DancingLines on real-world datasets from Weibo and Baidu. Finally, we conclude the paper in Sect. 8.

2 Related Work

Event Popularity Analysis. Many researches [1,10,19,22] have focused on event evolution analysis for a single medium. The event popularity was evaluated by hourly page view statistics from Wikipedia in [1]. [10] chose the density-based clustering method to group the posts in social text streams into events and tracked the evolution patterns. Breaking news dissemination is studied via network theory based propagation behaviors in [13]. [22] proposed a TF-IDF based approach to analyze event popularity trends. In all, network-based approaches usually have high computational complexity, while frequency-based methods are usually less accurate on reflecting the event popularity.

Cross-Platform Analysis. From a cross-platform perspective, existing researches focus on topic detection, cross-social media user identification, cross-domain information recommendation, etc. [2] selected Twitter, *New York Times* and Flickr to represent multimedia streams, and provided an emerging topic detection method. An attempt, trying to combine Twitter and Wikipedia to do first story detection, was discussed in [18]. [26] proposed an algorithm based on multiple social networks like Twitter, and Facebook to identify anonymous identical users. The relationship between social trends from social network and web trends from search engine are discussed in [5,9]. Recently, a good prediction of social links between users from aligned networks using sparse and low rank matrix is well discussed in [24]. However, few studies have been conducted for popularity analysis from cross-platform perspective.

Dynamic Time Warping. DTW is a well-established method for similarity search between time series. Originating from speech pattern recognition [20], DTW has been effectively implemented in many domains [5]. Recently, remarkable performance on time series classification and clustering by combining KNN classifiers have been achieved in [4,14]. The well-known Derivative DTW is proposed in [8]. Weighted DTW [7] was designed to penalize high phase differences. In [21], the side effect of endpoints which tends to disturb the alignments dramatically in time series is confirmed and an improvement for eliminating such issue is proposed. We are inspired by these related works when designing our own DTW based model for aligning EPTSs.

3 Problem Formulation

3.1 Event Popularity Quantification

We start from dividing the time span T of an event into n periods, which is determined by the time resolution, each stamped with t_i, $T = \langle t_1, \cdots, t_n \rangle$. A record is a set of words preprocessed from datasets, such as a post from social networks or a query from search engines. Then, we use the notation w_k^i to represent, within time interval t_i, the kth word in a record. The notation $R_j^i = \{w_1^i, w_2^i, \cdots, w_{|R_j^i|}^i\}$ is the jth record within time interval t_i. An *event phase*, corresponded to t_i and

denoted as E_i, is a finite set of words, and each word is from a related record R_j^i. As a result $E_i = \bigcup_j R_j^i$.

We can now introduce the prototype of our popularity function $pop(\cdot)$. For a given word $w_k^i \in E_i$, the popularity of the word w_k^i is defined as

$$pop(w_k^i) = fre(w_k^i) \cdot weight(w_k^i), \tag{1}$$

where $fre(w_k^i)$ is the word frequency of w_k^i within t_i. The weight function, $weight(w_k^i)$, for a word within t_i, is the kernel we solve in the TF-SW part and is the key to generate event popularity. In this work, we propose a weight function not only utilizing the lexical but also semantic relationships. Details about how to define the weight function is discussed in Sect. 5.

Once we get popularity of word w_k^i within t_i, the popularity of an event phase E_i, $pop(E_i)$, can be generated by summing up all words' popularity,

$$pop(E_i) = \sum_{w_i^k \in E_i} pop(w_k^i). \tag{2}$$

We regard the pair $(t_i, pop(E_i))$ as a point on X-Y plane and get a series of points, formalizing a curve on the plane to reflect the dissemination trend of an event $\mathscr{E}$.

To compare the curves from different media, a further normalization is employed,

$$\overline{pop}(E_i) = \frac{pop(E_i)}{\sum\limits_{1 \le k \le n} pop(E_k)}. \tag{3}$$

After the normalization, the popularity trend of an event on a single medium is represented by a sequence, denoted as $\mathscr{E} = \langle \overline{pop}(E_1), \cdots, \overline{pop}(E_n) \rangle$, which is defined as *Event Popularity Time Series*.

3.2 Time Series Alignment

Two EPTSs generated from two platforms of an event $\mathscr{E}$ are now comparable and can be visualized in a same X-Y plane as Fig. 1, which shows normalized EPTSs of Event *Sinking of a Cruise Ship* generated from Baidu and Weibo.

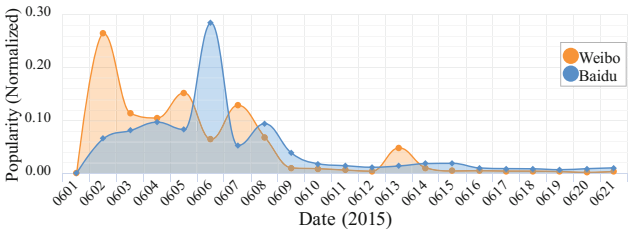


Fig. 1. Normalized EPTSs, *Sinking of a Cruise Ship* (Color figure online)

A Chinese cruise ship called Dongfang Zhi Xing sank into Yangtze River on the night of June 2, 2015 and the following process lasted for about 20 days. X-axis in Fig. 1 represents time and Y-axis indicates the event popularity. If we shifted the orange EPTS, generated from Weibo, to the right for about 4 units, we would notice the blue one approximately overlaps the orange one. This phenomenon indicates a *temporal warp*, which means the trend features are similar, but there exists time differences between EPTSs.

According to Fig. 1, EPTSs are temporally warped. For example, entertainment news tends to be disseminated on social networks and can easily draw extensive attention, but its dissemination on serious media like *Wall Street Journal* is very limited. Another interesting feature is the time differences between EPTSs, the degree of temporal warp, which reveals events' preferences to media. Alignments of EPTSs are quite suitable to reveal such interesting features.

Two temporally-warped EPTSs of an event $\mathscr{E}$ from two media A and B, are denoted as $\mathscr{E}^* = \langle \overline{pop}(E_1^*) \cdots, \overline{pop}(E_n^*) \rangle$, where $\mathscr{E}^*$ represents either $\mathscr{E}^A$ or $\mathscr{E}^B$.

A *match* m_k between E_i^A and E_j^B is defined as $m_k = (i, j)$. Distance between two matched data points is denoted as $dist(m_k)$ or $dist(i, j)$.

There is one problem, *twist*, existing when there are two matches $m_{k_1} = (i_1, j_1)$, $m_{k_2} = (i_2, j_2)$ with $i_1 < i_2$, but $j_1 > j_2$. The reason why there cannot be *twist* is that time sequence and the evolution of events cannot be reversed.

EPTS alignment aims to find a series of twist-free matches $M = \{m_1, \cdots, m_{|M|}\}$ for two $\mathscr{E}^A$ and $\mathscr{E}^B$ that every data point from an EPTS has at least one counterpoint from the other one, and the cumulative distance is the minimum. An intuitive thinking about an optimal alignment is that it should be a feature-to-feature one and differences between aligned EPTSs should be as small as possible. The minimum cumulative distance satisfy these two requirements. The key of alignments is to define a specific, precise, and meaningful distance function $dist(\cdot)$ for our task, which will be fully discussed in Sect. 6.3.

4 Scheme Overview of DancingLines

The overview of DancingLines is illustrated in Fig. 2. We first preprocess the data, then implement the TF-SW and ωDTW-CD models, and finally apply our scheme to real event datasets.

Data Preprocessing is applied on the raw data and has three steps. First of all, in Data-Formatting step, we filter out all irrelevant characters, such as punctuation, hyper links, etc. Secondly, Stopword-Removal step cleans frequently used conjunctions, pronouns and prepositions. Finally, we split every record into words through Word-Segmentation step.

TF-SW is a semantic-aware popularity quantification model based on Word2Vec and TextRank to generate EPTSs at certain temporal resolutions. This model is established by three steps. First of all, a cut-off mechanism is proposed to filter the unrelated words. Secondly, we construct TextRank graph to calculate the relative importance for the remaining words. Finally, a synthesized similarity calculation is defined for the edge weights in TextRank graph. We find

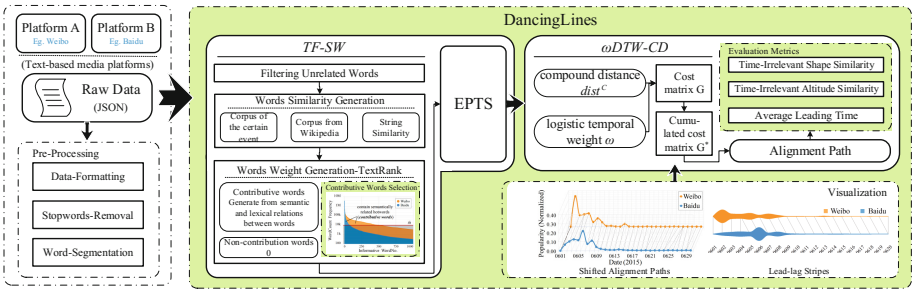


Fig. 2. The overview of DancingLines Scheme

that only the words with both high semantic and lexical relations with other ones truly determine the event popularity. For that, a conception *contributive words* is defined and will be discussed in Sect. 5.

ωDTW-CD is a pairwise EPTSs alignment model derived from DTW. In this model, we innovatively define three distance function for DTW, event phase distance $dist^{\mathscr{E}}(\cdot)$, derivative distance $dist^{D}(\cdot)$, and Euclidean vertical line distance $dist^{L}(\cdot)$. Based on these three distance function, a compound distance is generated. A temporal weight coefficient is also introduced into the model for improving the alignment results. We further introduce these in detail in Sect. 6.

5 Semantic-Aware Popularity Quantification Model (TF-SW)

5.1 Filtering Unrelated Words

Since the number of distinct words for an event can be thousands of hundreds and there are tons of them actually not related to the event at all, it is too expensive to take them all into account. We propose a cut-off threshold mechanism to eliminate these unrelated noisy words and significantly reduce the complexity of whole scheme.

In fact, natural language corpus approximately obey the power law distribution and Zipf's Law [17]. Denoting r as the frequency rank of a word in a corpus and f as the corresponded word's frequency, then

$$f = H \cdot r^{-\alpha}, \tag{4}$$

where α and H are feature parameters for a specific corpus.

Since the words with high frequency is the necessary but not sufficient condition for those words to really reflect the actual event trends, an interesting question that where the majority of distribution of r lies is raised. For any power law with exponent $\alpha > 1$, the median is well defined [17]. That is, there is a point $r_{1/2}$ that divides the distribution in half so that half the measured

values of r lie above $r_{1/2}$ and half lie below. In our case, r as rank, its minimum is 1, and the point is given by

$$\int_{r_{1/2}}^{\infty} f \, dr = \frac{1}{2} \int_{r_{min}}^{\infty} f \, dr \Rightarrow r_{1/2} = 2^{1/(\alpha-1)} r_{min} = 2^{1/(\alpha-1)}. \tag{5}$$

Emphasis should be placed on the words that rank ahead of $r_{1/2}$, and the words within the long tail which are occupied by noise should be discarded. Thus cut-off threshold can now be defined as

$$th = H \cdot r_{1/2}^{-\alpha} = \frac{1}{2} \cdot H \cdot 2^{1/(1-\alpha)} \tag{6}$$

Through this filter, we dramatically reduce the whole complexity of the scheme. For Event *AlphaGO*, the words we need to consider for Baidu reduce from thousands to around 40 and the ones for Weibo reduce to about 350, so the complexity has been reduced by at least 3 orders of magnitude.

5.2 Construction of TextRank Graph

After filtered through threshold, the remaining words are regarded as the representative words that do matter in quantifying the event popularity. However, for the remaining words, the importances are still obscure. They cannot just be naively presented by words' frequency, as a result we introduce TextRank [15] into our scheme.

For our task here, vertex in TextRank algorithm stands for a word that has survived the frequency filter in Sect. 5.1 and we use undirected edges in TextRank instead of directed edges in PageRank, since the relationships between words are bidirectional.

Inspired by the idea of TextRank, we further need to define the weights of edges in the graph described above. We introduce a conception *similarity* between words w_i and w_j, denoted as $sim\,(w_i, w_j)$ for the edges' weights.

However, we notice that there exist some words which passed the first filter but having negative similarity with all the other remaining words, which means these words are semantically far away from the topic of events. This phenomenon, in fact, indicates the existence of paid posters who post a large number of unrelated messages especially on social networks. To address this problem, we focus on the really related words and define a conception *contributive words*, denoted as

$$C_i = \{w_j^i \in E_i \mid \exists w_k^i \in E_i,\ sim(w_k^i, w_k^j) > 0\} \tag{7}$$

and $\mathscr{C} = \bigcup C_i$. It is worth pointing out that this another filter-like process does not increase any computational complexity and we just do not establish edges when their weights are less than zero, then the non-contributive words will be discarded.

We construct a graph for each event phase E_i, where vertices represent the words and edges refer to their similarity $sim(w_i, w_j)$. We run the TextRank

algorithm on the graphs and then get the real importance of each contributive word, $TR(w_i)$. The formula for TextRank is defined as

$$TR(w_i) = \frac{1-\theta}{|\mathscr{C}|} + \theta \cdot \sum_{j \to i} \frac{sim(w_i, w_j)}{\sum\limits_{k \to j} sim(w_k, w_j)} \cdot TR(w_j), \tag{8}$$

where the factor θ, ranging from 0 to 1, is the probability to continue to random surf follow the edges, since the graph cannot be a perfect graph and face potential dead-ends and spider-straps problem in practice. According to [15], θ is usually set to be 0.85. $|\mathscr{C}|$ represents the number of all contributive words, and $j \to i$ refer to words that is adjacent to word w_i.

5.3 Similarity Between Words

In our view, similarity between words are contributed by their semantical and lexical relationships and these two parts will be discussed in this subsection.

First of all, to quantify words' semantic relationships, we adopt Word2Vec [16] to map word w_k to vector $\mathbf{w}_k$. To comprehensively reflect the event characteristics, we integrate two corpora, an event corpus $\mathbb{R}$ from our datasets and a supplementary corpus extracted from Wikipedia with a broad coverage of events (denoted as *Wikipedia Dump*, or $\mathbb{D}$ for short), to train our Word2Vec models. For a word w_k, the corresponding word vectors are $\mathbf{w}_k^{\mathbb{R}}$ and $\mathbf{w}_k^{\mathbb{D}}$ respectively. Both event-specific and general semantic relations between words w_i and w_j are extracted and composed by

$$sem(w_i, w_j) = \beta \cdot \frac{\mathbf{w}_i^{\mathbb{R}} \cdot \mathbf{w}_j^{\mathbb{R}}}{\|\mathbf{w}_i^{\mathbb{R}}\| \cdot \|\mathbf{w}_j^{\mathbb{R}}\|} + (1-\beta) \cdot \frac{\mathbf{w}_i^{\mathbb{D}} \cdot \mathbf{w}_j^{\mathbb{D}}}{\|\mathbf{w}_i^{\mathbb{D}}\| \cdot \|\mathbf{w}_j^{\mathbb{D}}\|}, \tag{9}$$

where β is related to the two corpora and determines which one and to what extent we would like to emphasize.

Secondly, we consider the lexical information and integrate the string similarity so that we can combine the

$$sim(w_i, w_j) = \gamma \cdot sem(w_i, w_j) + (1-\gamma) \cdot str(w_i, w_j), \tag{10}$$

where we introduce a parameter γ to make our model general to different languages. For example, words that look similar are likely to be related in English, while this likelihood is fairly limited for languages like Chinese. We adopt the efficient cosine string similarity as

$$str(w_i, w_j) = \frac{\sum\limits_{c_l \in w_i \cap w_j} num(c_l, w_i) \cdot num(c_l, w_j)}{\sqrt{\sum\limits_{c_l \in w_i} num(c_l, w_i)^2} \cdot \sqrt{\sum\limits_{c_l \in w_j} num(c_l, w_j)^2}}, \tag{11}$$

where $num(c_l, w_i)$ means counts of character c_l in word w_i.

5.4 Definition of Weight Function

Since the sum of vertices' TextRank values for a graph is always 1 regardless of the graph scale, the TextRank value tends to be lower when there are more contributive words within the time interval. Therefore, a compensation factor within each event phase E_i is multiplied to the TextRank values, and the weight function $weight(\cdot)$ for contributive words is finally defined as

$$weight(w_j^i) = \frac{TR(w_j^i)}{|C_i|} \cdot \sum_{w_k^i \in E_i} fre(w_k^i). \tag{12}$$

Recalling that in our scheme, the event popularity $pop(E_i)$ is the sum of popularity of *all* words, for the consistency of Eq. (1), we make the weight function for the non-contributive words identically equal to zero. Then for all words, popularity can be calculated through Eq. (1). For each event phase E_i, according to Eq. (2), we can generate the event popularity within t_i and EPTSs through Eq.(3).

6 Cross-Platform Analysis Model (ωDTW-CD)

6.1 Classic Dynamic Time Warping with Euclidean Distance

We find that, with only the global minimum cost considered, classic DTW with Euclidean distance may provide results suffering from *far-match* and *singularity* problems when aligning pairwise cross-platform EPTSs.

Far-Match Problem. Classic DTW disregards the temporal range, which may lead to *"far-match"* alignments. Since the EPTSs of an event from different platforms keep pace with the event's real-world evolution, alignment of EPTSs' data points that are temporally far away is against the reality. Thus, classic method should be more robust and Euclidean Distance is not ideal enough for EPTS alignment.

Singularity Problem. Classic DTW with Euclidean distance is vulnerable to the *"singularity"* problem elaborated in [8], where a single point in one EPTS is unnecessarily aligned to multiple points in another EPTS. These singular points will generate misleading results for further analysis.

6.2 Event Phase Distance

Recalling Eq. (7) that all the contributive words for an event phase E_i are denoted as C_i and $\mathscr{C}$ is a set of all contributive words for an event $\mathscr{E}$ on single medium, we can utilize the similarity between the contributive word sets C_i to match those event phases. To quantify this similarity, we propose our *event phase distance* measure. Distance between E_i^A and E_j^B is denoted as $dist^{\mathscr{E}}(i, j)$.

Since $\mathscr{C}$ for different platforms are probably not identical, let the general $\mathscr{C}' = \mathscr{C}^A \cup \mathscr{C}^B$. Then, each word list C_i can be intuitively represented as a

one-hot vector $\mathbf{z}_i \in \{0,1\}^{|\mathscr{C}'|}$, where each entry of vectors indicates whether corresponding contributive word exists in word list C_i. However, problem arises when calculating the similarity between these very sparse vectors, especially when the event corpus is of a large scale and there are huge amount of data points in EPTSs. To address this problem, we leverage SimHash [3], adapted from *locality sensitive hashing* (LSH) [6], to hash the very sparse vectors to small signatures while preserving the similarity among the words.

According to [3], s projection vectors $\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_s$ are selected at random from the $|\mathscr{C}'|$-dimensional Gaussian distribution. A projection vector $\mathbf{r}_l$ is actually a hash function that hashes a one-hot vector $\mathbf{z}_i$ generated from C_i to a scalar -1 or 1. s projection vectors hash the original sparse vector $\mathbf{z}_i$ to a small signature $\mathbf{e}_i$, where $\mathbf{e}_i$ is an s-dimensional vectors with entries equal to -1 or 1. Sparse vectors $\mathbf{z}_i^A$ and $\mathbf{z}_j^B$ can be hashed to $\mathbf{e}_i^A$ and $\mathbf{e}_j^B$ and the distance between these two points can be calculated by

$$dist^{\mathscr{E}}(i,j) = 1 - \frac{\mathbf{e}_i^A \cdot \mathbf{e}_j^B}{\|\mathbf{e}_i^A\| \cdot \|\mathbf{e}_j^B\|}. \tag{13}$$

The dimension of short signatures, s, can be used to tune the accuracy we want to remain versus the low complexity. If we want to dig some subtle information in a high temporal resolution, say half an hour, we should increase s to get more accuracy, while if we just want to have a glimpse of the event, a small s is reasonable.

6.3 The ωDTW-CD Model

To more comprehensively measure the distance between data points from two EPTSs, a ωeighted DTW method with Compound Distance (ωDTW-CD) is proposed to balance temporal alignment and shape-matching. ωDTW-CD tries to synthesize trend characters, Euclidean vertical line distance, and event phase distance all together and this overall distance is measured by compound distance $dist^C(i,j)$,

$$dist(i,j) = dist^C(i,j) + \omega_{i,j}. \tag{14}$$

We regard the difference between estimated derivative of EPTS points, $dist^D(i,j)$, as the trend characters distance. According to [8], $dist^D(i,j)$ generated by

$$dist^D(i,j) = \left|D(E_i^A) - D(E_j^B)\right|, \tag{15}$$

where the estimated derivative $D(x)$ is calculated through

$$D(x) = \frac{x_i - x_{i-1} + \frac{x_{i+1} - x_{i-1}}{2}}{2}. \tag{16}$$

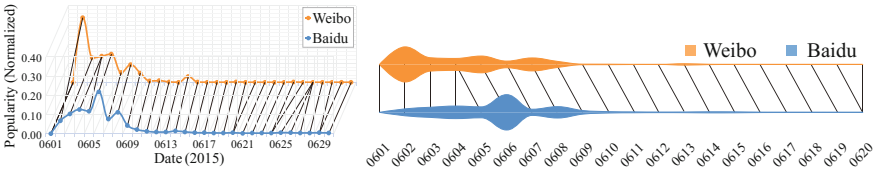
As stated in [8], this estimate is simple but robust to trend characters compared to other estimation methods. The compound distance $dist^C(i,j)$ is generated by

$$dist^C(i,j) = \sqrt[3]{dist^{\mathscr{E}}(i,j) \cdot dist^L(i,j) \cdot dist^D(i,j)}, \tag{17}$$

where $dist^{\mathscr{E}}(i,j)$ is the event phase distance and $dist^{L}(i,j)$ is the Euclidean vertical line distance between data points E_i^A, E_j^B defined as $dist^{L}(i,j) = |E_i^A - E_j^B|$. For the purpose of flexibility [7], we introduce a sigmoid-like temporal weight

$$\omega_{i,j} = \frac{1}{1 + e^{-\eta(|i-j|-\tau)}}. \tag{18}$$

The temporal weight is actually a special cost function for the alignment in our task. It has two parameters, η and τ, to generalize for many other events and languages. Parameter η decides the overall penalty level, which we can tune for different EPTSs. Factor τ is a prior estimated time difference, having the same unit as the temporal resolution we choose, between two platforms based on the natures of different medias.



(a) Aligned EPTSs (b) Lead-Lag stripes for aligned EPTSs

Fig. 3. Visualization of ωDTW-CD, *Sinking of a Cruise Ship*

A visualization is showed in Fig. 3a and it gives a direct way to know how the data points from EPTSs are aligned. The links in the figure represent matches. The lead-lag stripes [25] in Fig. 3b show a more obvious way to know matches. The X-axis represents time and the stripes' vertical width indicates the event popularity in that day. We can find that after the Event *Sinking of a Cruise Ship* happens, the Weibo platform captured and propagated the topic faster than Baidu did in the beginning and then more people started to search on the Baidu for more information so the popularity on Baidu rose.

7 Experiments

7.1 Experiment Setup

Datasets. Our experiments are conducted on eighteen real-world event datasets from Weibo and Baidu, covering nine most popular events that occurred from 2015 to 2016. All the nine events covered in our datasets have provoked intensive discussions and gathered widespread attention. In addition, they are both typical events in distinct categories including disasters, high-tech stories, entertainment news, sports and politics. The detailed information of our datasets is listed in Table 1.

Table 1. Overall information of the datasets

No.	Event name	# of records (k)		Size (MB)	
		Weibo	Baidu	Weibo	Baidu
①	Sinking of a Cruise Ship	308.45	1560.4	320.59	401.48
②	Chinese Stock Market Crash	701.71	420.40	578.77	74.14
③	AlphaGo	838.12	2337.3	654.89	406.83
④	Leonardo DiCaprio, Oscar Best Actor	2569.5	730.82	1788.9	139.52
⑤	Kobe Bryant's Retirement	3655.3	2300.9	2274.8	403.69
⑥	Huo and Lin Went Public with Romance†	1535.2	1615.2	1027.1	289.98
⑦	Brexit Referendum	957.16	2160.4	715.51	392.32
⑧	Pokémon Go	936.38	3652.2	695.90	625.87
⑨	The South China Sea Arbitration	7671.0	7815.3	5918.2	1451.9

Implementation and Parameters. We implement CBOW when doing Word2Vec [16]. The parameters involved in TF-SW are set to be $\beta = 0.7$, with $\gamma = 0.02$ considering the nature of Chinese language, that there are many different characters but almost no meaning changes on words. The factor for TextRank is set to be $\theta = 0.85$ by convention. Without specification, we set each time interval to be 1 day. The corresponding parameters for the sigmoid-like temporal weight are set as $\eta = 10, \tau = 2$.

7.2 Verification of TF-SW

To evaluate the effectiveness of TF-SW, we compare the EPTS generated by our model with the EPTSs by other two baselines, naive frequency and TF-IDF [22]. All the EPTSs generated by Naive Frequency and TF-IDF are normalized in the same way as TF-SW through Eq. (3). Based on the three generated EPTSs, we present a thorough discussion and comparison to validate our TF-SW model.

Accuracy. We pick up the peaks in EPTSs and backtrack what exactly happened in reality. An event is always pushed forward by series of "little" events and we call them sub-events, which are reflected as peaks in EPTS figures.

In the Event *Capsizing of a Cruise Ship*, the real-world event evolution involves four key sub-events. On the night of June 1, 2015, the cruise ship sank in a severe thunderstorm. Such a shocking disaster raised tremendous public attention on June 2. On June 5, the ship was hoisted and set upright. A mourning ceremony was held on June 7, and on June 13, total 442 deaths and only 12 survivors were officially confirmed, which marked the end of the rescue work.

The EPTS generated by TF-SW shows four peaks, which is illustrated in Fig. 4. All these peaks are highly consistent with the four key sub-events in real world, while the end of rescue work on June 13 is missed by approaches based on Naive Frequency and TF-IDF. In conclusion, TF-SW model shows the ability to track the development of events precisely.

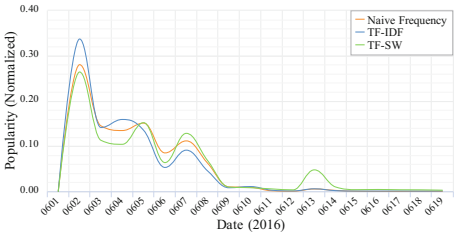


Fig. 4. *Sinking of a Cruise Ship*, Weibo

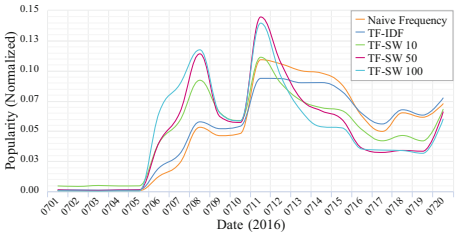


Fig. 5. *Pokémon Go*, Baidu ($th = N$.)

Sensitivity to Burst Phases. Compared with the baselines, our model are more sensitive to the burst phases of an event, as is shown in Fig. 5, especially on data points 07/06, 07/08, and 07/11. The event popularity on these days are larger than those obtained by Naive Frequency and TF-IDF. In another word, the EPTSs generated through TF-SW rises faster, more significant in peaks, and are more sensitive to breaking news which enables the model to capture the burst phases more precisely. From three EPTSs of TF-SW with different th, it is shown that TF-SW is more sensitive to the burst of events with a higher th value, as is shown by the data point 07/06.

An event whose EPTS rises fast at some data points possesses the potential to draw wider attention. It is reasonable for a popularity model not only to depict the current state of event popularity, but also take the potential future trends into consideration. In this way, a quick response to the burst phases of an event is more valuable for real-world applications. This advantage of our model can lead to a powerful technique for first story detection on ongoing events.

Superior Robustness to Noise. To verify whether our model can effectively filter out noisy words, we further implement an experiment on a simulated corpus. We first extract 50K Baidu queries with the highest frequency in the corpus of Event *Kobe's Retirement* and make them as the base data for a 6-day simulated corpus. Then we randomly pick noisy queries from Internet that are not relevant to Event *Kobe's Retirement* at all. The amount of noisy queries is listed in Table 2.

Table 2. Number of noisy records added to each day

Day	1	2	3	4	5	6
# (k)	0.000	1.063	2.235	3.507	4.689	6.026

Since each day's base data are identical, a good model is supposed to filter noisy queries out and generate an EPTS with all identical data points, which form a horizontal line in X-Y plane. EPTSs generated by TF-SW, Naive Frequency and TF-IDF are shown in Fig. 6. It is shown that TF-SW successfully

filters out the noise and generates the EPTS which is a horizontal line and captures the real event popularity, while the other two methods Naive Frequency and TF-IDF are obviously effected by the noisy queries and generate EPTSs that cannot accurately reflect the event popularity.

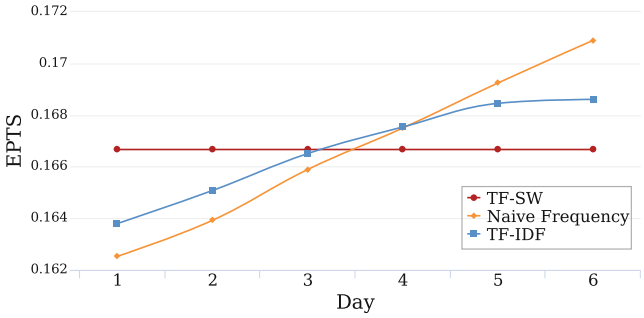


Fig. 6. EPTSs on the simulated corpus

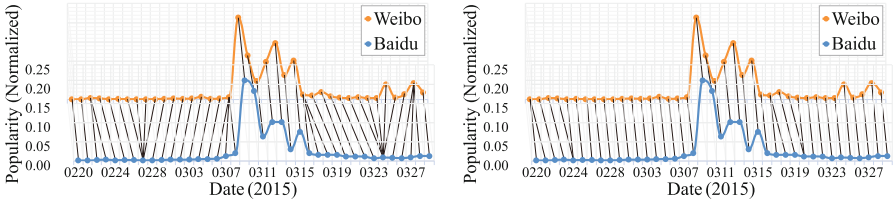
7.3 Verification of ωDTW-CD

To demonstrate the effectiveness of ωDTW-CD, we compare it with seven different DTW extensions listed below.

- *DTW* is the DTW method with Euclidean distance.
- *DDTW* [8] is the Derivative DTW which replaces the Euclidean distance with the difference of estimated derivatives of the data points in EPTSs.
- DTW_{bias} & $DDTW_{bias}$ are the extended DTW and DDTW respectively with a bias towards the diagonal direction.
- *ωDTW & ωDDTW* are the temporally weighted DTW and DDTW, where the sigmoid-like temporal weight defined by Eq. (18) is introduced to the cost matrices.
- *DTW-CD* is a simplification of *wDTW-CD* that implements only $dist^C$ without temporal weight ω.

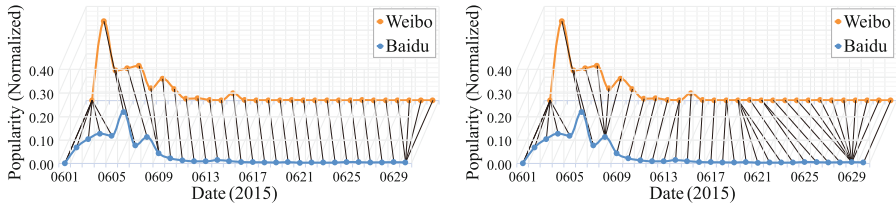
Singularity. Fig. 7 visualizes the results generated by ωDTW and our proposed model. Classic DTW and DTW_{bias} severely suffer the problem of singularity. Compared with ωDTW, ωDTW-CD presents better and more stable performance when aligning the time series with sharp fluctuations. In general, our model is capable of avoiding the singularity problem by involving the derivative differences.

Far-Match. Considering the fact that the time difference between two aligned sub-event can barely exceed two days, far-match exists in the alignment generated by $\text{DDTW}_{\text{bias}}$ and DTW-CD in Fig. 8, but not in our results in Fig. 3a. Thus, the sigmoid-like temporal weight introduced to our model helps avoid the far-match problem.



(a) ωDTW ($\eta = 10, m = 2$) (b) ωDTW-CD ($\eta = 10, m = 2$)

Fig. 7. Alignment results of 2 methods, *AlphaGo*. One data point is categorized as a singular point if it is matched to more than 4 points from the other EPTS.



(a) $\text{DDTW}_{\text{bias}}$ $(1.2, 1, 1.2)$ (b) DTW-CD

Fig. 8. Alignment results of 2 methods, *Sinking of a Cruise Ship*

Overall Performance. All the comparison results on the eighteen real-world datasets are illustrated in Fig. 9, where each color corresponds to a method, each method are ranked respectively for each event, and methods with higher grades are ranked on the top. Results facing *singularity* or *far-match* are marked by red boxes. The performances are graded under the following criteria. The grades are given to show the relative performances among different methods only regarding one event. The method that does not suffer from *singularity* or *far-match* has higher grades than the one that does. The methods giving same alignment results are further graded considering their complexity.

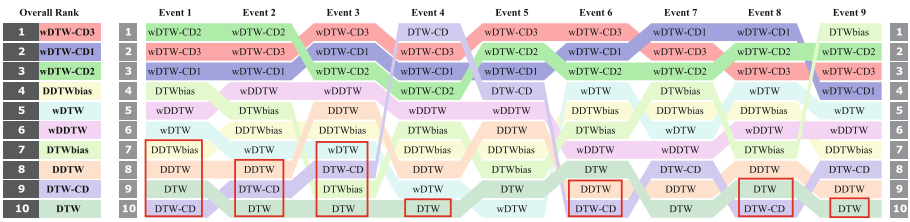


Fig. 9. Ranking visualization of grades for 10 methods on nine real-world events. (Color figure online)

In comparison with existing variants of DTW as well as the reduced version of our method, ωDTW-CD achieves improvements on both performance and robustness on alignment generation and successfully conquers the problem of *singularity* and *far match*. Results shows that the event phase distance, estimated derivative difference, and the sigmoid-like temporal weight simultaneously contribute to the performance enhancement of ωDTW-CD. Moreover, with parameter η and τ, our model is flexible to different temporal resolutions and to events of distinct popularity features. In Fig. 9, ωDTW-CD$_1$ corresponds to $\eta = 5$, $\tau = 3.2$. $\eta = 10$, $\tau = 2$ is for ωDTW-CD$_2$. $\eta = 5$, $\tau = 2.2$ is for ωDTW-CD$_3$. The results show the strong ability of ωDTW-CD to handle specific events.

8 Conclusion

In this paper, we quantify and interpret event popularity between pairwise text media with an innovative scheme, DancingLines. To address the popularity quantification issue, we utilize TextRank and Word2Vec to transform the corpus into a graph and project the words into vectors, which are covered in TF-SW model. To furthermore interpret the temporal warp between two EPTSs, we propose ωDTW-CD to generate alignments of EPTSs. Experimental results on eighteen real-world event datasets from Weibo and Baidu validate the effectiveness and applicability of our scheme.

References

1. Ahn, B., Van Durme, B., Callison-Burch, C.: Wikitopics: what is popular on Wikipedia and why. In: Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages, pp. 33–40 (2011)
2. Bao, B., Xu, C., Min, W., Hossain, M.S.: Cross-platform emerging topic detection and elaboration from multimedia streams. TOMCCAP **11**(4), 54 (2015)
3. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: STOC, pp. 380–388 (2002)
4. Dau, H.A., Begum, N., Keogh, E.: Semi-supervision dramatically improves time series clustering under dynamic time warping. In: CIKM, pp. 999–1008 (2016)
5. Giummolè, F., Orlando, S., Tolomei, G.: A study on microblog and search engine user behaviors: how Twitter trending topics help predict Google hot queries. Human **2**(3), 195 (2013)
6. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: STOC, pp. 604–613 (1998)
7. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern Recogn. **44**(9), 2231–2240 (2011)
8. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: SDM, pp. 1–11 (2001)
9. Kwak, H., Lee, C., Park, H., Moon, S.: What is Twitter, a social network or a news media? In: WWW, pp. 591–600 (2010)
10. Lee, P., Lakshmanan, L.V.S., Milios, E.E.: Keysee: supporting keyword search on evolving events in social streams. In: KDD, pp. 1478–1481 (2013)

11. Li, R., Lei, K.H., Khadiwala, R., Chang, K.: Tedas: a Twitter-based event detection and analysis system. In: ICDE, pp. 1273–1276 (2012)
12. Lin, S., Wang, F., Hu, Q., Yu, P.: Extracting social events for learning better information diffusion models. In: KDD, pp. 365–373 (2013)
13. Liu, N., An, H., Gao, X., Li, H., Hao, X.: Breaking news dissemination in the media via propagation behavior based on complex network theory. Physica A **453**, 44–54 (2016)
14. Maus, V., Câmara, G., Cartaxo, R., Sanchez, A., Ramos, F., Queiroz, G.: A time-weighted dynamic time warping method for land-use and land-cover mapping. J-STARS **9**(8), 3729–3739 (2016)
15. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. In: EMNLP, pp. 404–411 (2004)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS, pp. 3111–3119 (2013)
17. Newman, M.: Power laws, pareto distributions and Zipf's law. Contemp. Phys. **46**(5), 323–351 (2005)
18. Osborne, M., Petrovic, S., McCreadie, R., Macdonald, C., Ounis, I.: Bieber no more: first story detection using Twitter and Wikipedia. In: SIGIR 2012 Workshop on Time-Aware Information Access (2012)
19. Rong, Y., Zhu, Q., Cheng, H.: A model-free approach to infer the diffusion network from event cascade. In: CIKM, pp. 1653–1662 (2016)
20. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. **26**(1), 43–49 (1978)
21. Silva, D.F., Batista, G.E., Keogh, E.: On the effect of endpoints on dynamic time warping. In: SIGKDD Workshop on Mining Data and Learning from Time Series (2016)
22. Tang, Y., Ma, P., Kong, B., Ji, W., Gao, X., Peng, X.: ESAP: a novel approach for cross-platform event dissemination trend analysis between social network and search engine. In: Cellary, W., Mokbel, M.F., Wang, J., Wang, H., Zhou, R., Zhang, Y. (eds.) WISE 2016. LNCS, vol. 10041, pp. 489–504. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48740-3_36
23. Wang, J., et al.: Mining multi-aspect reflection of news events in Twitter: discovery, linking and presentation. In: ICDM, pp. 429–438 (2015)
24. Zhang, J., Chen, J., Zhi, S., Chang, Y., Yu, P.S., Han, J.: Link prediction across aligned networks with sparse and low rank matrix estimation. In: ICDE, pp. 971–982 (2017)
25. Zhong, Y., Liu, S., Wang, X., Xiao, J., Song, Y.: Tracking idea flows between social groups. In: AAAI, pp. 1436–1443 (2016)
26. Zhou, X., Liang, X., Zhang, H., Ma, Y.: Cross-platform identification of anonymous identical users in multiple social media networks. TKDE **28**(2), 411–424 (2016)