Diversifying Trending Topic Discovery via Semidefinite Programming

Hui Wu[1,2], Yi Fang[2], Huming Wu[3], and Shenhong Zhu[3]

[1]Google, 1600 Amphitheatre Parkway, Mountain View, California 94043, USA
[2]Santa Clara University, 500 El Camino Real, Santa Clara, California 95053, USA
[3]Yahoo!, 701 1st Ave, Sunnyvale, California 94089, USA

***Abstract*—Discovering trending topics from the Web has attracted much attention in recent years, due to users' increasing need for time-sensitive information. A large body of the existing research focuses on detecting trends by examining search traffic fluctuations, and the queries with large traffic increments would be detected as trends. The weakness of this approach is that the trends are dominated by popular fields such as celebrities, as those related queries have large search traffic. Consequently, other topics such as travel and shopping are rarely regarded as trends.**

In this paper, we present a scalable diversified trending topic discovery system with a MapReduce implementation. The trending topics are discovered based on three criteria: diversity, representativeness, and popularity. We explicitly model these three factors in our objective function and propose an efficient Semidefinite Programming algorithm to solve the corresponding optimization problem. To the best of our knowledge, no prior work in the literature tackles trending topic diversification. We conduct a comprehensive set of experiments with case studies to demonstrate the effectiveness of our approach. The proposed system has been successfully tested in the real-world operational environment, yielding significant improvement in traffic over the existing production system.

***Keywords*-Trending Topic Discovery; Diversification; Semidefinite Programming**

I. Introduction

Trending topic discovery addresses an age-old question "what are people talking about?". As people increasingly turn to the search engines for news and information, it is tempting to view search activity at any moment in time as a snapshot of the collective consciousness. Consequently, major search engines begin to provide services about what people are currently searching, such as Yahoo!'s Trending Now[1], Google Trends[2] and Bing's Popular Now[3]. These services reflect the instantaneous intentions of the global population and can be utilized to enhance user interaction with search engines. For example, the *Trending Now* module, shown in Figure 1, is a trending topic discovery system deployed on Yahoo!'s homepage. The trending topics displayed were extracted from search query log, mainly globally trendy and of interest to a wide user base. Every click on the trending query topics takes the user to a search result page where the topic is entered automatically as a query. Through the search result page, users can then find detailed information about the trending topic.

Fig. 1: The *Trending Now* module on Yahoo! homepage at 5:17pm on August 8, 2016

The existing trending topic discovery systems aim to find individual trending topics without looking at those topics as a whole. The resulting topics largely lack of diversity and specifically Yahoo!'s *Trending Now* is often dominated by the presence of celebrities as demonstrated in Figure 1. The lack of diversity may frustrate users often with diverse information needs and would significantly harm user engagement. In this paper, we address the task of diversifying trending topics. We use the setup of Yahoo! *Trending Now* to demonstrate our methodology. In this scenario, k topics (e.g., $k = 10$) is selected from a pool of candidate topics of various categories. The selection criterion are based on diversity, representativeness and popularity of the topics. First of all, the selected trending topics should be as diverse as possible to reduce redundant information and satisfy users' various information needs. Secondly, the selected topics should be a good representative of all the candidates in the pool. Since the user interfaces such as *Trending Now* can only show a limited number of topics, we need to select the topics to convey as much information as possible. Last but not the least, the selected topics should be popular and hopefully lead to increased traffic by better engaging users. The major challenge of considering these three factors all together lie in the fact that they often conflict with each other. In this paper, we explicitly model the three factors in a unified framework. Our

This research was conducted when the first author worked at Yahoo! while a part-time PhD student at Santa Clara University.

[1]http://www.yahoo.com/
[2]http://www.google.com/trends
[3]http://www.bing.com/

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

contributions in this paper can be summarized as follows:

1) To the best of our knowledge, no prior work exists on diversifying trending topic discovery.
2) We model diversity, representativeness and popularity in a unified objective function and propose an efficient Semidefinite programming (SDP) algorithm to optimize the function.
3) We present a scalable system with a MapReduce implementation. The system has been extensively tested in the real-world operational environment, yielding significant improvement in traffic over the existing production system.
4) A comprehensive set of experiments with case studies demonstrate the effectiveness of our approach and the efficiency of our system.

II. Related Work

Detecting trending topics from web search queries has been an active area in both industry and academia. Several studies have formulated it as an anomaly detection task which aims at finding irregularities of queries such as a large divergence from the mean number of occurrences. The underlying assumption is that when a topic is trending, it tends to diverge from its regular temporal behavior. Dong et al. [1] defines the “buzz" score of a query based on the difference between the language models of the query at two different time slots. An extension of [1] is done by [2] to provide a location-aware search trend detection algorithm using geographical properties of query entries. Fang et al. [3] addresses the data sparsity issue in the localized query trend detection task. Vlachos et al. [4] investigate periodicities and bursts in web search queries using Fourier analysis. Kleinberg [5] proposes to model the “bursts of activity" by state transitions in an infinite-state automaton. Parikh and Sundaresan [6] find interesting bursty patterns from large-scale E-commerce query logs on an incremental or real-time basis. They further analyze classification of such temporal bursts based on both cause and effect periods. Golbandi et al. [7] et al. develop a linear auto-regression model to predict query counts in order to expedite search trend detection.

Numerous works have investigated query dynamics, or how query volume changes over time. Three general types of temporal query profiles are identified in [8]. Kulkarni et al. [9] extend [8]’s work to build a rich picture of changes to query popularity. Others examine the relationship between query dynamics and news events. Ginsberg et al. [10] show how query dynamics reflect real-world events such as emerging flu outbreaks. Adar et al. [11] identify when changes in query frequency are correlated to mentions in traditional media and blogs. Goel et al. [12] show that consumer behavior is reflected by search volume.

Besides web queries, other types of information may also indicate trends of topics. Brendan et al. [13] identifies and visualizes interesting peaks in news coverage using extracted keywords and iconic images. Vaca et al. [14] proposes a matrix co-factorization approach to discovering and tracking emerging topics in news. Biessmann et al. [15] presents a method that captures canonical trends in a pool of influential news websites. Mukherjee et al. [16] describes an approach that detects trending topics by looking at the hourly Wikipedia page visitation statistics. Some topic models are proposed to examine topics and their changes across time on academic publications [17]. Much work investigates social media such as Twitter to detect trends. Mathioudakis et al. [18] introduce *TwitterMonitor*, a framework for online trend detection in Twitter. Weng et al. [19] use a queuing method for bursty keyword detection and wavelet techniques for trend detection on Twitter. Based on an anomaly detection approach, [20] propose an algorithm for online bursty keyword detection in the Twitter stream. Naaman et al. [21] develop a taxonomy of trends based on Twitter messages and identify important features to categorize trends. Yang and Lesvkovec [22] examine the patterns of temporal behavior for hashtags in Twitter by presenting a time-series clustering algorithm. Asur et al. [23] conduct a analysis of trending topics on Twitter for the formation, persistence, and decay of trends. *TrendMinder* [24] is proposed as a context-sensitive method of detecting trending topics in a microblog post stream. Cataldi et al. [25] have developed metrics to individually identify each word that might indicate a trending topic on Twitter. Recently, Vicient and Moreno [26] propose unsupervised topic discovery in microblogging social networks. Dang et al. [27] build a dynamic bayesian network to represent the temporal evolution of keyword. A formal concept analysis is presented for topic detection on Twitter [28].

Another closely related field is search result diversification which has been extensively studied in the information retrieval community. Some of the works combine both novelty and relevance of search results [29]. Zhai et al. [30] propose a risk minimization framework that allows users to define an arbitrary loss function for a given set of results. Agrawal et al. [31] makes use of a taxonomy for classifying queries and documents and create a diverse set of results according to this taxonomy. Clarke et al. [32] focus on developing a framework of evaluation that takes into account both novelty and diversity. Carterette et al. [33] propose a probabilistic approach to maximize the coverage of the retrieved documents with respect to the aspects of a query. Recently, Kato and Tanaka [34] propose a method of optimizing search result presentation for queries with diverse intents. Ozdemiray and Altingovde [35] re-rank the candidate documents for each query aspect and then merge these rankings by adapting the score and rank aggregation methods. A comprehensive survey on search result diversification is presented in [36]. However, to the best of our knowledge, no prior work has been done on diversifying trending topic discovery.

III. Background

Our proposed system is based on *Time Sense* [2] which is an internal trending detection property at Yahoo!. It detects trends from web search logs and also mines the query’s search intent category. Various works in the literature [2], [1], [7] describe

Time Sense from different angles. Figure 2 shows its latest architecture called Time Sense Trend Detection (TSTD).

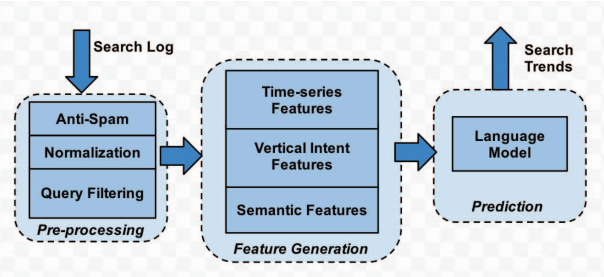


Fig. 2: The architecture of Time Sense Trend Detection (TSTD)

In Figure 2, the "Search Log" input comes from Yahoo!'s Web search query logs which record two types of event information: *view event* and *click event*. Specifically, *View event* stores all the elements in each search result page, and *click event* includes all the information about every click on search results. The query log data comes in streams with a very short delay. As shown in the figure, the entire process of the query log based trend detection system has three stages: "Pre-processing", "Feature Generation", and "Prediction".

"Pre-processing" includes three major components: *Anti-Spam*, *Normalization*, and *Query Filtering*. *Anti-Spam* removes spam view events and click events from search logs, which prevents Yahoo! *Trending Now* from being abused by advertisement publishers or malicious parties. Various ways are used to detect spams, such as IP address based rule filtering, session based filtering, and query semantic based filtering. *Normalization* reduces a query's variants to a stem representation, and it mainly handles punctuations, case folding, and coding issues. *Query Filtering* removes adult or offensive queries that are deemed inappropriate.

"Feature Generation" generates a set of features for the machine learning approach used at the later "Prediction" stage. The module mainly generates three types of features: *Time-Series Feature*, *Vertical Intent Feature*, and *Semantic Feature*. *Time-Series Feature* indicates the changes in search traffic of a given query during a period of time. The traffic fluctuations are an important feature to determine whether this query is trending or not. Usually a common query has small traffic fluctuations while a trending query has big ones. *Vertical Intent Feature* identifies a query's vertical intents. In Yahoo! Search, the returned results not only come from general Web search, but also from vertical segments. By checking which verticals' search results are shown for a given query, the system could infer this query's vertical intents (e.g., "iPhone 6" for Shopping and "Sochi Olympics" for Sports). Furthermore, each query topic is tagged with a category which is determined by which vertical channel that query was searched in. Thus, one query topic could belong to multiple categories. In the system, the candidate query topics in the pool come from six categories of Yahoo!'s vertical channels including News, Multimedia, Entertainment, Travel, Shopping and Sports. *Semantic Feature* is generated through the search results of queries, and represents each query with a feature vector. The similarities between queries are computed based on their feature vectors.

With the features generated in *Feature Generation*, the trending topics are finally discovered by *Language Model* in the "Prediction" module. *Language Model* produces a list of trending topics by assigning a trending score to each candidate query to show its importance (a higher score indicates the query topic is a better candidate to show in Yahoo! *Trending Now*). Table I shows an exemplar set of trending topics generated by TSTD with their corresponding traffic defined in Section V-A.

Trend	$\rho_t(S)$	NE	SH	SP	MU	EN	TR
jennifer connelly	43	11.8	11.8	0	11.8	0	0
malaysia airlines plane	26	11.7	11.7	0	11.7	0	0
hayden panettiere	20	11.4	0	0	11.4	0	0
holy grail of guitars	9	11.4	11.4	0	11.4	0	0
anna fenninger	15	11.4	0	0	11.4	0	0
lacey holsworth	5	11.4	0	0	11.4	0	0
carmen berra	14	11.3	0	0	11.3	0	0
shannon szabados	19	11.1	0	0	11.1	0	0
reese witherspoon	9	10.7	0	0	10.7	0	0
khloe kardashian	54	10.4	0	0	10.4	0	0
OVERLAP	N/A	10	3	0	10	0	0
TOTAL TRAFFIC	214						

TABLE I: The top 10 trending topics generated by TSTD with their corresponding traffic ($\rho_t(S)$). The six categories are as follows. NE: News, SH: Shopping, SP: Sports, MU : Multimedia, EN: Entertainment, TR: Travel. The time generating these topics is 18:00 on March 8, 2014 (UTC).

Except for the first row, all the queries in the first column are the predicted trending topics. Their associated traffic in the next period of time is given in the second column. All the other columns show the trending scores in each category. For example, "jennifer connelly" has three categories and its trending score is 11.8, and it does not have intent in Shopping, Entertainment and Travel as the trending scores are 0. "OVERLAP" shows the number of trending topics in each category, and "TOTAL TRAFFIC" gives the total search traffic of the 10 trending topics.

Based on the statistics in Table I, we have two noticeable observations. First of all, the trending topics are dominated by only two categories: News and Multimedia, and quite a few categories including Shopping, Entertainment and Travel do not offer any trending topics. Secondly, the total search traffic was quite low given the time span and the majority of the topics were not searched much, which indicates that the selected topics may not be very interesting to many users. To address the above issues, we design a diversified trending topic discovery system with the details given in Section IV and its performance evaluation described in Section VI.

IV. Diversified TSTD

Figure 3 shows the architecture of our proposed system called Diversified Time Sense Trend Detection (DTSTD). As shown in this architecture, DTSTD includes three major components: Candidate Re-selection, Trend Representation, and Trend Selection. The function of "Candidate Re-selection" is to balance trending candidates among different categories

of topics. “Trend Representation" enriches the contextual information of each candidate by using search log, and generates a weighted graph to represent all trending candidates and their semantic relations. “Trend Selection" employs a Semidefinite Programming Optimization algorithm to select k topics to form the final set of trends. The details of these three components are given in the following subsections.

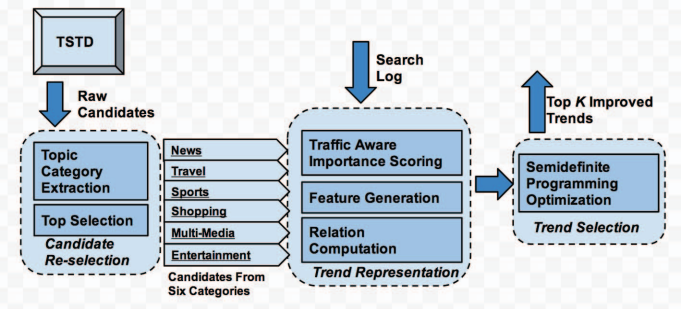


Fig. 3: The architecture of Diversified Time Sense Trend Detection (DTSTD)

A. Candidate Re-selection

As shown in Table I, we have seen that TSTD's results lack of diversity. To tackle this issue, DTSTD limits the number of candidate topics from each category. Specifically, the module “Topic Category Extraction" classifies each original topic candidate into the six categories. “Top Selection" selects the top h trending candidates from each category, and the union of all the candidate topics forms the final candidate set. In the experiments, we chose $h = 80$, i.e., each category includes 80 candidate topics. Formally, let E represent the entire candidate set, and the trending score of q where $q \in E$ is denoted as q_{buzz}.

B. Trend Representation

To select the representative for each topic, DTSTD utilizes the relations among the candidates, “Topic Representation" generates a weighted graph for representing all trending topics and their semantic relations, where each vertex represents one trending topic and each edge with a weight indicates the relation between its two topics. We re-calculate the trending score of each topic by considering both its original trending score and its search traffic during the past period of time. The “Traffic Aware Importance Scoring" module computes the new trending score according to Equation 1.

$$\phi(q) = q_{buzz} \times q_{traffic}, \tag{1}$$

where $q_{traffic}$ is the search traffic of q. In DTSTD as shown in Equation 1, the final trending score of q is determined by both query buzz score q_{buzz} and query popularity $q_{traffic}$, while TSTD only considers the buzz score. The “Relation Computation" module then measures the relation between q_i and q_j as follows:

$$w(q_i, q_j) = \frac{\gamma(q_i, q_j) \times \phi(q_i)}{\sum_{q \in E \setminus \{q_i\}} \gamma(q, q_i)} + \frac{\gamma(q_i, q_j) \times \phi(q_j)}{\sum_{q \in E \setminus \{q_j\}} \gamma(q, q_j)} \tag{2}$$

and

$$\gamma(q_i, q_j) = \text{Cosine}(F_i, F_j), \tag{3}$$

where F_i and F_j denote the semantic features of q_i and q_j respectively. $\text{Cosine}(F_i, F_j)$ is the Cosine similarity between F_i and F_j. “Feature Generation" generates F_i by checking the search results of q_i. For example, $F_1 = \{f_{url_1}, f_{url_2}, \cdots, f_{url_N}\}$ where f_{url_i} represents the number of times url_i is shown in the search results of q_1. If $w(q_i, q_j)$ is larger than zero, q_i and q_j are regarded as the related topics with each other.

There are several advantages of representing trending topics and their relations by Equation 1 and 2. First of all, this representation takes the latest search traffic into account, and the search traffic of a query topic reflects the degree of of user interest in this topic. In other words, if one topic has a high search traffic, it means many people are interested in it. According to Equation 1, the topics with high search traffic will be assigned high trending scores. Secondly, this representation considers semantic relations among topics, which can help deduplication. Moreover, based on the semantic relations, we can measure representativeness of a given topic. If one topic has high semantic relations with others, it could be a good representative of the others.

C. Trend Selection

The weighed graph introduced in Section IV-B characterizes the candidate trending topics and their relations. Based on the weighted graph, the “Trend Selection" module will select a small number of trending topics to achieve the goal of diversity, representativeness, and popularity. We treat the selection as a partition task. Formally, the final set of trending topics is denoted as $S \subset E$ where $|S| = k$ ($k \in R^+$ is a given number) and the set of the topics not selected is $T = E \setminus S$. We will solve the following optimization problem to identify S:

$$\begin{aligned} \arg\max_{\substack{S \subset E \\ |S|=k \\ T=E \setminus S}} \quad & \frac{1}{|S| \times |T|} \sum_{\substack{q_i \in S \\ q_j \in T}} w(q_i, q_j) \\ & - \frac{1}{|S| \times |S|} \sum_{\substack{q_i \in S \\ q_j \in S}} w(q_i, q_j) \end{aligned} \tag{4}$$

Equation 4 has two goals. One is to maximize the relations between S and T, which means the selected topics have strong relations with the set of the candidates not being selected. Thus, these selected topics are good at representing the whole pool of the candidate topics. Another goal is to minimize the internal relations among S. In other words, the selected topics have weak semantic relations within themselves and thus they are semantically diversified.

Mathematically, Equation 4 resembles the Maximum Cut problem [37], but there exists a key difference. The known Maximum Cut variants typically only maximize the relation between S and T. Thus, the existing solutions to Maximum Cut cannot be easily applied to Equation 4. Inspired by the approach given in [38], we propose to use Semidefinite Programming (SDP) optimization. Formally, let $\mathbf{x}$ be the vector

indicating the assignments of the partition of S/T, where

$$x_i = \begin{cases} 1 & q_i \text{ is assigned to } S \\ -1 & q_i \text{ is assigned to } T, \end{cases} \quad (5)$$

and $|E| = n$ which means there are n trending topics in E. The Semidefinite Programming optimization objective function for Equation 4 is as follows (the detailed derivations are skipped in this paper, but can be found at the link[4]):

$$\begin{aligned} & \arg\min \mathbf{tr}(C^T X) \\ s.t. \quad & \forall 1 \le i \le n,\ \mathbf{tr}(A_i^T X) = (1 + \tfrac{2\times k}{n-k})^2 - 1 \\ & \forall n+1 \le i \le 2n,\ \mathbf{tr}(A_i^T X) = -1 \\ & \mathbf{tr}(A_{2n+1}^T X) = (2 + \tfrac{2\times k}{n-k})^2 * k + (\tfrac{-2\times k}{n-k})^2 * (n-k) \\ & \mathbf{tr}(A_{2n+2}^T X) = (2 + \tfrac{2\times k}{n-k}) * k + (\tfrac{-2\times k}{n-k}) * (n-k) \\ & X \succeq 0 \end{aligned} \quad (6)$$

where

$$C = \begin{pmatrix} \mathbf{W}_{n\times n} & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n} \end{pmatrix}$$

where $\mathbf{W}$'s elements are defined in Equation 2.

$$X = \mathbf{z}\mathbf{z}^T$$

$$\mathbf{z} = (y_1, y_2, \cdots, y_n, 1_{n+1}, 1_{n+2}, \cdots, 1_{2n})^T$$

$$y_i = \tfrac{n+k}{n-k} x_i + 1$$

$\forall 1 \le i \le 2n-1, A_i \in R^{2n\times 2n}$, where

$$A_{i,a,b} = \begin{cases} 1 & a = b = i; \\ -1 & a = i, b = 2n; \\ -1 & a = 2n, b = i; \\ 0 & \text{others.} \end{cases}$$

$A_{2n} \in R^{2n\times 2n}$, where

$$A_{2n,a,b} = \begin{cases} 1 & a = b = 2n; \\ -1 & a = 2n-1, b = 2n; \\ -1 & a = 2n, b = 2n-1; \\ 0 & \text{others.} \end{cases}$$

$A_{2n+1} \in R^{2n\times 2n}$, where

$$A_{2n+1,a,b} = \begin{cases} 1 & a = b, 1 \le a, b \le n; \\ 0 & \text{others.} \end{cases}$$

$A_{2n+2} \in R^{2n\times 2n}$, where

$$A_{2n+2,a,b} = \begin{cases} 0.5 & 1 \le a \le n, b = 2n; \\ 0.5 & a = 2n, 1 \le b \le n; \\ 0 & \text{others.} \end{cases}$$

After solving Equation 6, we obtain the approximate solution to Equation 4. The whole process is described in Algorithm 1. Despite the approximateness of the solution, the experiments demonstrate its effectiveness.

[4]http://www.cse.scu.edu/~yfang/SDP.pdf

Data: $A_1, \cdots, A_{2n+2}$, C, n and k to Equation 6
Result: S to Equation 4
1. Input $A_1, \cdots, A_{2n+2}$, C, n and k to Equation 6, and obtain X by CSDP[5]; 2. Get the largest eigenvector of X: $\vec{\mathbf{v}}$;
3. Set $S = \emptyset$, $E = \{e_1, e_2, \cdots, e_n\}$;
while $|S| < k$ **do**
  4.1.
  $$\begin{aligned} q = \quad & \arg\max_{e_i \in E \setminus P} |v_i| \\ s.t. \quad & \forall e_j \in S, C_{i,j} < \varepsilon \end{aligned}$$
  where v_i is the i-th value in $\vec{\mathbf{v}}$;
  4.2. $S = S \cup \{q\}$;
end
5. Output S.
Algorithm 1: The SDP based algorithm to obtain the solutions to Equation 4

D. MapReduce Implementation

The real-world trending topic discovery systems have to analyze query log in (near) real time and quickly update the trending topics. Thus, we propose a MapReduce implementation based on Apache Hadoop[6]. The Mapper and Reducer workflow is illustrated in Figure 4. As we can see, there are six Mapper-Reducer pairs which are used for various tasks: to form the categorical diversified candidates from TSTD's original candidates, to aggregate the search traffic of each query in the last one hour, to collect the search results of each query and their impressions in the last one hour, to recalculate the trending score of each query by considering its original trending scores and its latest search traffic, to compute semantic relation between every two queries by using their URL features, and to run Algorithm 1 for generating the final trending topics.

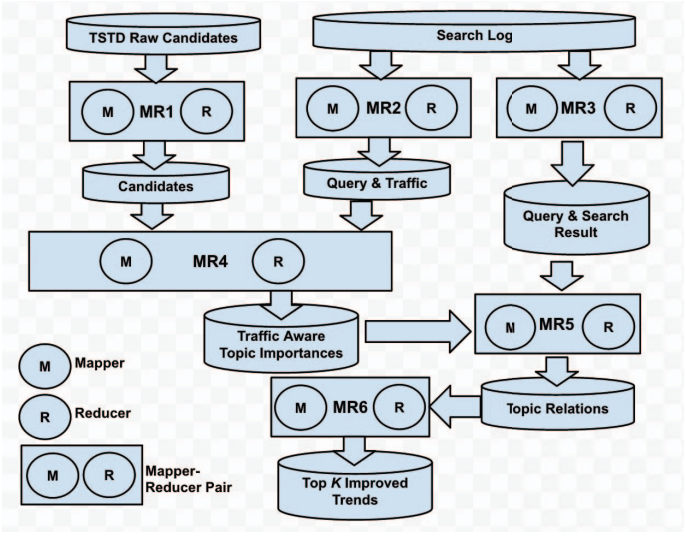


Fig. 4: The MapReduce implementation of DTSTD

[5]https://projects.coin-or.org/Csdp/
[6]http://hadoop.apache.org/

V. Case Study

In this section, we conduct a case study to compare DTSTD with TSTD. We first introduce our evaluation methodology with the evaluation metric, and then analyze the trending topics discovered by our system and compare them with those found by the baseline.

A. Evaluation Methodology

We use Time Sense Trend Detection (TSTD) described in Section III as the baseline for comparison with our system. TSTD is the existing production systems at Yahoo! and they have yielded good empirical performance for various applications.

We use search traffic of a given trending topic within a time period as the evaluation metric. In general, search traffic is not a good metric for measuring trending topics, as many popular queries always have very high search traffic (e.g., “Facebook", “Yahoo", “CNN", etc.). However, it is an appropriate metric for our system because those popular queries are removed by the "Query Filtering" component in TSTD and they are not considered by TSTD and DTSTD. All the candidate topics of interest are assumed to be reasonable trending topics. Therefore, search traffic can measure user engagement with the detected trending topics. In other words, good trending topics are expected to lead to increased traffic. Formally, the search traffic is defined as follows

$$\rho_t(S) = \sum_{i=1}^{K} \rho_t(s_i) \tag{7}$$

where $S = \{s_1, s_2, \cdots, s_K\}$ is the set of selected k query topics, $\rho_t(s_i)$ is the traffic of s_i between time interval $t-1$ and t. $\rho_t(S)$ is the total traffic of S from $t-1$ to t. Because the task of trend detection is highly time sensitive, the evaluation metric should take time information into account. In TSTD and DTSTD, the time interval is one hour, i.e., every hour the systems predict trends for the next hour and the traffic of the next hour will be used to evaluate the discovered trending topics.

B. TSTD vs DTSTD

We use DTSTD to generate the top 10 trending topics at 18:00 on March 8, 2014, the same time with that in Table I in Section III. Table II shows the results. Comparing the trending topics in the two tables, it is noticeable that DTSTD works better than TSTD in both popularity and diversity. The trending topics found by DTSTD have the total traffic of 1214, while it is only 214 for TSTD. Moreover, the trends of DTSTD are more evenly distributed in five categories, while TSTD includes only three categories. Furthermore, there are no dominating categories in DTSTD, while News and Multimedia seem over-represented in TSTD’s results. In the following subsections, we will investigate the two methods in more details and reveal how DTSTD achieves the improvement.

Trends	$\rho_t(S)$	NE	SH	SP	MU	EN	TR
mega millions winning numbers	635	6.8	0	0	0	0	0
malaysia missing plane	299	6.6	6.6	0	6.6	0	0
miki howard	31	6.6	6.6	0	6.6	0	0
shannon szabados	19	11.1	0	0	11.1	0	0
mr peabody and sherman	49	0	0	0	0	2.9	0
paul bettany	2	6.5	6.5	0	6.5	0	0
daylight savings time	35	2.6	0	0	2.6	0	0
jennifer connelly pictures	2	4.1	4.1	0	4.1	0	0
frozen niagara falls	78	0	0	0	0	1.2	1.2
son of god movie 2014	64	0	0	0	0	2.1	0
OVERLAP		7	4	0	6	3	1
TOTAL Traffic	1214						

TABLE II: The top 10 trending topics generated by DTSTD with their corresponding traffic ($\rho_t(S)$). The six categories are as follows. NE: News, SH: Shopping, SP: Sports, MU : Multimedia, EN: Entertainment, TR: Travel. The time generating these topics is 18:00 on March 8, 2014 (UTC).

C. Representativeness

One of the main goals of DTSTD is to identify a representative for each topic. A good representative should have two properties. First of all, it should have strong correlations with other candidates in this topic. If two topics have a strong correlation, one topic can cover most of the information of another and they are semantic alternatives to each other. For example, “malaysia missing plane" and “malaysia airlines missing plane" have a high correlation, and either of them can be replaced by another without losing much information. Secondly, the representative should be popular among users. As previously described, the same query topic could be searched by users through all kinds of variants because of their language preferences. The representative topic should be a popular one so that the topic can be found interesting by most users.

Let us take the trending topic of “malaysia airlines misses mh370" as an example to see how TSTD and DTSTD choose their respective representatives. Table III contains the details of the topic in TSTD and DTSTD. As shown in the first row of the table, the representative in DTSTD is “malaysia missing plane" and the one in TSTD is “malaysia airlines plane". The columns of 1, 3, 5, 7 show other candidates in this topic, and 2, 4, 6, 8 show the relations R (Equation 2) between the candidates and the representatives. For example, the relation R between “malaysia missing plane" and “malaysia airlines plane picture" is 49.

As we can see, the representative in DTSTD is better than that in TSTD in terms of both relation and popularity. The total relation between “malaysia missing plane" and all the other candidates is 3378, which is much large than 515 of “malaysia airlines plan". It indicates that “malaysia missing plane" better represents all the other candidates. Furthermore, according to Table I and II, “malaysia missing plane" has the traffic of 299, which is also much large than “malaysia airlines plan"’s traffic of 26. It shows that the representative in DTSTD is much more popular than that in TSTD. The advantages of DTSTD over TSTD may come from two factors. DTSTD recalculates the trending score of each candidate by considering its recent popularity, which boosts the importance of the popular topics

and promotes those topics to be chosen as representatives. Secondly, the objective function of DTSTD explicitly favors the queries with strong relations with others.

D. Detection of Emerging Topics

Besides detecting an already trendy topic, one interesting but challenging question is whether a trend discovery system can detect trends at their early stages. In other words, it is desirable the system can predict whether a topic will become trendy. Among the trends given in Table II, “mega millions winning numbers" is at its early stage when it was coming up. Table IV provides the details of this topic. The first row of the table is the representative of the topic, the first column shows all the queries related to this topic, the second column shows the original trending score given by TSTD, and the last column shows the relation between each candidate and its representative. For example, the original trending score of “mega lottery" is 1.4 and its relation to the representative of “mega millions winning numbers" is 566. The last row shows the avenge trending score and the average relation score.

As we can see, none of the queries related to “mega millions winning numbers" has a high trending score. The highest score is 6.8 from “mega millions winning numbers", but it is still even lower than the lowest trending score in Table I. On the other hand, Table II shows that “mega millions winning numbers" is very popular among users, which has the highest search traffic among the 10 trending topics. This example demonstrates that DTSTD’s capability to detect trending topics at their early stages.

Representative: **mega millions winning numbers (6.8)**		
Candidate	**Ori. Imp.**	**R.**
mega lottery	1.4	566
lottery numbers mega millions	1.5	583
lottery numbers	2.1	4
mega millions winning numbers california	1.6	418
mega millions jackpot analysis	1.4	137
mega millions numbers	2.7	636
mega millions lottery	3.0	737
winning mega millions numbers	2.2	618
mega ball lottery results	1.4	465
mega millions winner	1.8	573
mega millions	4.9	854
past mega millions winning numbers	1.5	121
mega million winning number	1.3	458
AVG	2.4	474.6

TABLE IV: Emerging topics detected by DTSTD

E. Removing Stale Topics

To keep the list of trending topics updated, it is necessary to detect stale topics and remove them. Table V shows the stale topics found by DTSTD. The first column shows the trending topics, and the second column includes their original trending scores given by TSTD, the third column gives their related queries and the last column shows the relation between the trends and their related queries. As we can see, these topics have very low traffic and they are not interesting to people any more, which means they are stale topics.

Compared with the results in Table IV, the common attribute of these stale topics is that they have very few related candidate topics. As shown in Table V, only “holy grail of guitars” has one related candidate, and the other three are completely isolated. When one topic becomes trending, people are likely to look for its related information with variants. Their search traffic fluctuations become positive, and consequently they are detected as trends. On the other hand, if one topic is not trending any more, less and less people search for it and its search traffic fluctuations become negative. Consequently this topic will have fewer and fewer related queries as illustrated in Table V. Therefore, by judging the number of related queries to the given topic, we can also tell the trendiness of this topic. In fact, this factor is considered by the objective function of DTSTD. According to Equation 4, if one topic has many variants, the correlations between its representative and the other variants will be large, and then this representative will strongly help maximize the objective function, which leads to this representative being selected as one trending topic.

Trend	**Score**	**Related Topics**	**R**
holy grail of guitars	11.4889	the holy grail of guitars	244
anna fenninger	11.4404	None	None
lacey holsworth	11.4076	None	None
reese witherspoon	10.7813	None	None

TABLE V: Stale topics detected by DTSTD

In summary, a typical trending topic usually has many variants which are strongly related with each others. On the other hand, one stale topic has very few related queries. This case study shows the advantage of considering candidate correlations while detecting trending topics. Furthermore, this case study demonstrates the importance of representative selection. If the representative not only convey the information of the topic but also is popular among users, this topic will engage many users. All of these factors are taken into account by DTSTD as shown in this case study.

VI. Experiments

We test the proposed system in Yahoo!’s operational environment during the time span from 07:00 on March 07 to 16:00 on March 11, 2014 in UTC Time.

A. Popularity

In this section, we investigate the popularity of trending topics. Figure 5 shows the traffic results of DTSTD, TSTD and all individual categories over the time span. Table VI contains the mean and variance of traffic for each method. As we can see, DTSTD generated much more traffic than TSTD and the individual categories did during most of the time. TSTD did not yield a comparable performance, even worse than the methods only use two individual categories: News and Multimedia. The advantages of DTSTD may come from the fact that it takes more factors into account. First of all, DTSTD considers the recent search traffic of each trend and thus the trending score not only reflects its trendiness but also indicates its popularity. Secondly, each trend is the representative of its topic and users are very likely to use it to search for this topic. Both factors would lead to increased traffic. Take an example in Table III. “malaysia missing plane"

DTSTD selected representative: **malaysia missing plane**				TSTD selected representative: **malaysia airlines plane**			
Candidates	R	Candidates	R	Candidates	R	Candidates	R
malaysia airlines plane picture	49	plane missing	69	malaysia airlines plane picture	8	malaysian plane	14
malaysia airlines flight	179	malaysian plane	119	malaysia airlines flight	10	malaysia missing airlines	179
malaysia airlines plane missing	52	malaysian plane crash	41	malaysia airlines plane missing	8	malaysian plane crash	5
malaysia airlines missing plane	145	kuala lumpur malaysia	11	malaysia airlines missing plane	16	kuala lumpur malaysia	2
missing malaysian plane	23	missing malaysia airlines	88	missing malaysian plane	4	missing malaysia airlines	12
mh370 malaysia airlines	60	malaysia airlines boeing 777	50	mh370 malaysia airlines	14	malaysia airlines boeing 777	6
malaysian plane missing	39	malaysia airlines missing	137	malaysian plane missing	6	malaysia airlines missing	17
malaysia airlines plane	179	malaysia plane crash	51	malaysia airlines crash	3	malaysia plane crash	8
malaysia airlines crash	14	plane crash	13	missing malaysian airline flight	9	plane crash	5
missing malaysian airline flight	80	missing airline	87	missing plane malaysia	8	missing airline	10
missing plane malaysia	60	malaysia crash	9	malaysia airlines plane crash	45	malaysia crash	1
malaysia airlines plane crash	66	malaysia airlines plane photo	57	missing plane	11	malaysia airlines plane photo	12
missing plane	46	malaysia plane	130	missing malaysia plane	8	malaysia plane	15
missing malaysia plane	58	malaysia airlines plane passports	26	malaysia plane missing	11	malaysia airlines plane passports	3
malaysia plane missing	83	missing plane found	22	malaysia flight	13	missing plane found	8
malaysia flight	156	malaysia airlines plane image	74	malaysia plane	11	malaysia airlines plane image	7
malaysia airlines	956	malaysian airlines crash	33	malaysia airlines news	11	malaysian airlines crash	4
malaysia airlines news	116	TOTAL	3378	plane missing	11	TOTAL	515

TABLE III: Different representatives selected by DTSTD and TSTD for the same topic where R is the relation calculated by Equation 2.

better described the topic that Malaysia Airlines is missing the plane of MH370, and it was searched more than “malaysia airlines plane" on this topic. One interesting fact is that if this event did not happen, “malaysia airlines plane" would have been searched much more often.

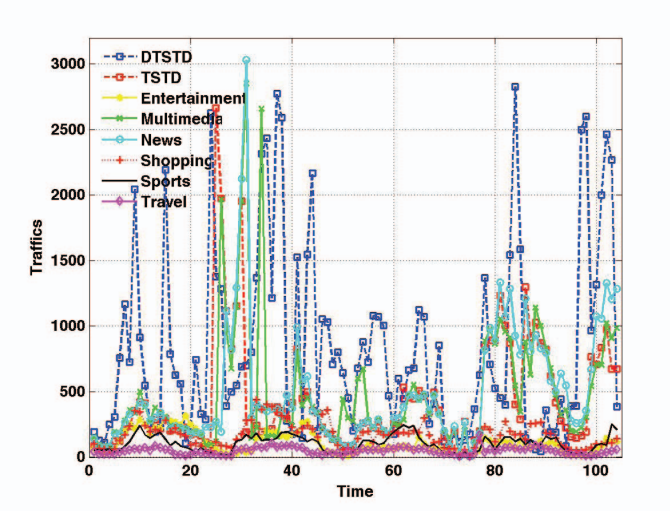


Fig. 5: Comparison of various methods in traffic

Method	Mean	STD
DTSTD	831	743
News	491	456
Multi-media	478	478
TSTD	438	432
Shopping	168	94
Entertainment	108	78
Sports	102	62
Travel	46	26

TABLE VI: Mean and standard deviation of various methods (STD) in traffic

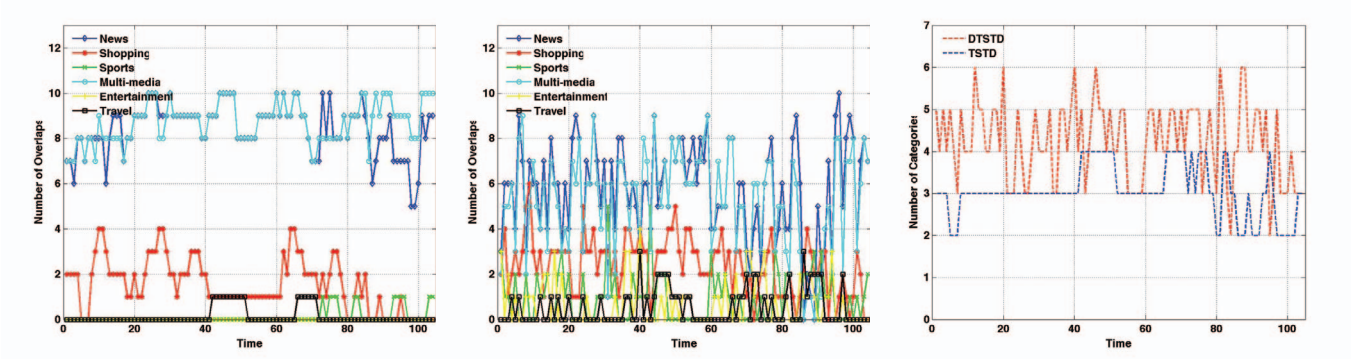
B. Diversity

In this section, we study the diversity effect of the proposed approach. Figure 6a and Figure 6b show the overlapping topics detected by individual categories and TSTD, and by individual categories and DTSTD, respectively. In these two figures, the horizontal axis shows the time points and the vertical axis shows the number of overlapping topics. For example, at the time point of 1 in Figure 6a, among the 10 trending topics in TSTD, 7 of them are in News, 7 in Multimedia, 2 in Shopping, and 0 in others. Except News, Multimedia and Shopping, all the other categories have no contributions. On the other hand, 6b has the trending topics overlapped with a wide variety of categories: i.e., 3 in News, 2 in Shopping, 3 in Sports, 2 in Multimedia, and 3 in Entertainment. By comparing these two figures, we can clearly see that DTSTD yields more diverse trending topics than TSTD did.

Figure 6c illustrates the difference in diversity between TSTD and DTSTD where its vertical axis is the number of involved categories for the two systems. For example, at time 1, the trending topics in TSTD come from three categories, while those in DTSTD are from five categories. As we can see, the topics in DTSTD distribute in more categories, yielding more diversified results.

C. Parameters

In this section, we investigate the parameters that may affect the performance of DTSTD. As shown in Section IV-C, DTSTD seeks to detect k popular topics and selects one good representative for each topic to form a set of trends. Formally, in Equation 4, the goal is to select k items from E, and these k items have strong relations with the rest of items in the set, while they have weak relations with each other. In this weighted graph, the objective function depends on two parameters. One is the size of the graph, i.e. $|E|$, and another is the relations among the nodes, which is measured by Graph Density [39]. The intuition is that if there are many candidate topics, the chance to have a better performance will be larger. It is difficult to say whether Graph Density should be larger since there exist two extreme cases. One is when all the nodes are isolated, and the results will be random as any



(a) Number of overlapping topics between TSTD and individual categories

(b) Number of overlapping topics between DTSTD and individual categories

(c) Number of distinct categories in TSTD and DTSTD

Fig. 6: Diversity of the trending topics

combination could make the objective function equal to zero. Another scenario is when all the nodes are strongly connected to each other, which is also difficult to partition the graph or select nodes since each one has good alternatives and it does not matter which nodes are chosen. To better understand the impact of these two parameters on our system, we show Figure 7a. The numbers of candidates, relation density and traffic are normalized between 0 and 1 in the figure. It is hard to draw a clear conclusion on the relation between the candidate number and the traffic, or between the relation density and the traffic, but there are several interesting observations. First of all, a high relation density and a small number of candidates do not lead to good performance. For example, at the time point 27 when relation density is large and the number of candidates is small, the corresponding traffic is small. Secondly, a small relation density and large candidate number do not yield good performance either. For instance, at the time point 92 when the candidate number is very large and the relation density is small, the corresponding traffic is not high. On the other hand, if the gap between candidate number and relation density is small, the performance would be good. For example, at the time point 31, those two numbers are very close and the corresponding traffic is relatively high.

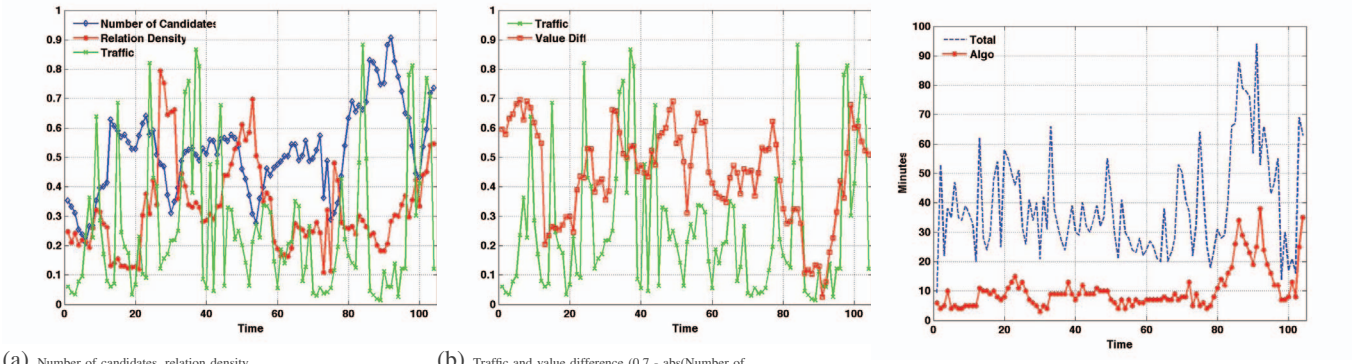
Based on these observations, we further show Figure 7b. In this figure, “Value Diff" is equal to “0.7 - abs(Number of Candidates - Relation Density)". The figure demonstrates that “Value Diff" is correlated with the traffic. The Pearson correlation between them is 0.27, which is much higher than the Pearson correlation of 0.0096 between “Number of Candidates" and “Traffic", and the Pearson correlation of 0.1592 between “Relation Density" and “Traffic". This may come from the fact that both “Number of Candidates" and “Relation Density" are global information of a graph, which can only describe the general information about a graph. On the other hand, DTSTD more focuses on identifying a topic by utilizing local information. As shown in the case study in Section V, if all the candidates related to a given topic are strongly connected, it would be a good trending topic.

D. *Efficiency*

In this section, we investigate the efficiency of DTSTD. We are particularly interested in Algorithm 1 since it is the core component in our system to diversify trending topics. Figure 7c shows the running time of Algorithm 1 and also the total running time of the system. As we can see in Figure 7c, Algorithm 1 took less than 10 minutes to complete in most of the cases. Moreover, the percentage of the running time of DTSTD over the whole system is small with the average percentage of 27.65%, which indicates that DTSTD is not the bottleneck of the entire trend detection process. In fact, we found that the data preprocessing took most of the time. In future work, we plan to optimize the system in two directions. Firstly, we will conduct data filtering at the very beginning of the process. In this way, both Hadoop I/O operation and computation time would be reduced. Secondly, we will minimize the Mapper-Reducer pairs by combing several steps, which will shorten the time of I/O operation on data transmission.

VII. Conclusions and Future Work

In this paper, we study the task of diversifying trending topic discovery. We propose a Semidefinite Programming (SDP) based approach by considering diversity, representativeness, and popularity of topics in a unified objective function. We also present a scalable system with a MapReduce implementation supporting the whole trend detection process. The proposed approach has been tested in the real-world operational environment and has demonstrated its advantages over the existing production system. In future work, we plan to utilize more data sources to diversify trending topics. For example, we can incorporate Twitter streams since they contain much realtime information especially when there is mass involvement in a news event such as a disaster. We will also conduct a theoretical analysis on the approximateness of the proposed SDP method. Furthermore, the proposed approach can be applied to many other applications that demand diversity of results. The examples include top news selection and related search recommendation.



(a) Number of candidates, relation density, and traffic in DTSTD

(b) Traffic and value difference (0.7 - abs(Number of Candidates - Relation Density)) in DTSTD

(c) Running time of Algorithm 1

Fig. 7: Effect of parameters and efficiecy in DTSTD

REFERENCES

[1] A. Dong, Y. Chang, Z. Zheng, G. Mishne, J. Bai, R. Zhang, K. Buchner, C. Liao, and F. Diaz, "Towards recency ranking in web search." WSDM, 2010.

[2] Z. A. Bawab, G. H. Mills, and J.-F. Crespo, "Finding trending local topics in search queries for personalization of a recommendation system." SIGKDD, 2012.

[3] Y. Fang, Z. A. Bawab, and J.-F. Crespo, "Collaborative language models for localized query prediction," *ACM Transactions on Interactive Intelligent Systems*, 2014.

[4] M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos, "Identifying similarities, periodicities and bursts for online search queries," in *SIGMOD*. ACM, 2004, pp. 131–142.

[5] J. Kleinberg, "Bursty and hierarchical structure in streams," *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373–397, 2003.

[6] N. Parikh and N. Sundaresan, "Scalable and near real-time burst detection from ecommerce queries," in *SIGKDD*. ACM, 2008, pp. 972–980.

[7] Z. Golbandi, L. Katzir, Y. Koren, and R. Lempel, "Expediting search trend detection via prediction of query counts." WSDM, 2013.

[8] R. Jones and F. Diaz, "Temporal profiles of queries," *ACM Transactions on Information Systems*, vol. 25, no. 3, p. 14, 2007.

[9] A. Kulkarni, J. Teevan, K. M. Svore, and S. T. Dumais, "Understanding temporal query dynamics," in *WSDM*. ACM, 2011, pp. 167–176.

[10] E. Aramaki, S. Maskawa, and M. Morita, "Twitter catches the flu: detecting influenza epidemics using twitter," in *EMNLP*. ACL, 2011, pp. 1568–1576.

[11] E. Adar, D. S. Weld, B. N. Bershad, and S. S. Gribble, "Why we search: visualizing and predicting user behavior," in *WWW*. ACM, 2007, pp. 161–170.

[12] S. Goel, J. M. Hofman, S. Lahaie, D. M. Pennock, and D. J. Watts, "Predicting consumer behavior with web search," *PNAS*, vol. 107, no. 41, pp. 17 486–17 490, 2010.

[13] B. Jou, H. Li, J. G. Ellis, D. Morozoff-Abegauz, and S.-F. Chang, "Structured exploration of who, what, when, and where in heterogeneous multimedia news sources," in *ACM Multimedia Conference*. ACM, 2013, pp. 357–360.

[14] C. K. Vaca, A. Mantrach, A. Jaimes, and M. Saerens, "A time-based collective factorization for topic discovery and monitoring in news," in *WWW*, 2014, pp. 527–538.

[15] F. Biessmann, J.-M. Papaioannou, M. Braun, and A. Harth, "Canonical trends: Detecting trend setters in web data," *ICML*, 2012.

[16] S. Mukherjee, R. Sujithan, and P. Subasic, "Detecting trending topics using page visitation statistics," in *WWW*, 2014, pp. 347–348.

[17] N. Kawamae, "Trend analysis model: trend consists of temporal words, topics, and timestamps," in *WSDM*. ACM, 2011, pp. 317–326.

[18] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *SIGMOD*. ACM, 2010, pp. 1155–1158.

[19] J. Weng and B.-S. Lee, "Event detection in twitter." in *ICWSM*, 2011.

[20] J. Guzman and B. Poblete, "On-line relevant anomaly detection in the twitter stream: an efficient bursty keyword detection model," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*. ACM, 2013, pp. 31–39.

[21] M. Naaman, H. Becker, and L. Gravano, "Hip and trendy: Characterizing emerging trends on twitter," *JASIST*, vol. 62, no. 5, pp. 902–918, 2011.

[22] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *WSDM*. ACM, 2011, pp. 177–186.

[23] S. Asur, B. A. Huberman, G. Szabo, and C. Wang, "Trends in social media: Persistence and decay." in *ICWSM*, 2011.

[24] N. Pervin, F. Fang, A. Datta, K. Dutta, and D. Vandermeer, "Fast, scalable, and context-sensitive detection of trending topics in microblog post streams," *ACM Transactions on Management Information Systems*, vol. 3, no. 4, p. 19, 2013.

[25] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proceedings of the Tenth International Workshop on Multimedia Data Mining*. ACM, 2010, p. 4.

[26] C. Vicient and A. Moreno, "Unsupervised topic discovery in micro-blogging networks," *Expert Systems with Applications*, vol. 42, no. 17, pp. 6472–6485, 2015.

[27] Q. Dang, F. Gao, and Y. Zhou, "Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks," *Expert Systems with Applications*, vol. 57, pp. 285–295, 2016.

[28] J. Cigarrán, Á. Castellanos, and A. García-Serrano, "A step forward for topic detection in twitter: An fca-based approach," *Expert Systems with Applications*, vol. 57, pp. 21–36, 2016.

[29] S. Gollapudi and A. Sharma, "An axiomatic approach for result diversification," in *WWW*. ACM, 2009, pp. 381–390.

[30] C. Zhai, W. W. Cohen, and J. Lafferty, "Beyond independent relevance: methods and evaluation metrics for subtopic retrieval," in *SIGIR*. ACM, 2003, pp. 10–17.

[31] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong, "Diversifying search results," in *WSDM*. ACM, 2009, pp. 5–14.

[32] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon, "Novelty and diversity in information retrieval evaluation," in *SIGIR*. ACM, 2008, pp. 659–666.

[33] B. Carterette and P. Chandar, "Probabilistic models of ranking novel documents for faceted topic retrieval," in *CIKM*. ACM, 2009, pp. 1287–1296.

[34] M. P. Kato and K. Tanaka, "To suggest, or not to suggest for queries with diverse intents: Optimizing search result presentation," in *WSDM*. ACM, 2016, pp. 133–142.

[35] A. M. Ozdemiray and I. S. Altingovde, "Explicit search result diversification using score and rank aggregation methods," *JASIST*, vol. 66, no. 6, pp. 1212–1228, 2015.

[36] I. Ounis, C. Macdonald, and R. L. Santos, "Search result diversification," *Foundations and Trends in Information Retrieval*, vol. 9, no. 1, pp. 1–90, 2015.

[37] F. Hadlock, "Finding a maximum cut of a planar graph in polynomial time," *SIAM Journal on Computing*, vol. 4, no. 3, pp. 221–225, 1975.

[38] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.

[39] R. Diestel, *Graph Theory*. Springer-Verlag, 2005.