The VLDB Journal (2020) 29:177–216
https://doi.org/10.1007/s00778-019-00569-6

SPECIAL ISSUE PAPER

# Microblogs data management: a survey

**Amr Magdy[1] · Laila Abdelhafeez[1] · Yunfan Kang[1] · Eric Ong[1] · Mohamed F. Mokbel[2]**

Received: 3 January 2019 / Revised: 7 April 2019 / Accepted: 29 August 2019 / Published online: 18 September 2019


**Abstract**
Microblogs data is the microlength user-generated data that is posted on the web, e.g., tweets, online reviews, comments on news and social media. It has gained considerable attention in recent years due to its widespread popularity, rich content, and value in several societal applications. Nowadays, microblogs applications span a wide spectrum of interests including targeted advertising, market reports, news delivery, political campaigns, rescue services, and public health. Consequently, major research efforts have been spent to manage, analyze, and visualize microblogs to support different applications. This paper gives a comprehensive review of major research and system work in microblogs data management. The paper reviews core components that enable large-scale querying and indexing for microblogs data. A dedicated part gives particular focus for discussing system-level issues and on-going effort on supporting microblogs through the rising wave of big data systems. In addition, we review the major research topics that exploit these core data management components to provide innovative and effective analysis and visualization for microblogs, such as event detection, recommendations, automatic geotagging, and user queries. Throughout the different parts, we highlight the challenges, innovations, and future opportunities in microblogs data research.

**Keywords** Microblogs · Social media · Twitter · Data management · Systems · Indexing · Query processing · Memory management · Main-memory · Flushing policy · Data analysis · Visual analysis · Event · Event detection · Event analysis · Recommendation · Geotagging · Geo · Spatial · Temporal · Top-$k$ · Textual · Keyword · User · Aggregation · Sampling · Clustering · Classification · Probabilistic models · Statistical · Graph · Summarization · Ranking

## 1 Introduction

Microblogs data, the microlength user-generated data that is posted on the web, such as tweets, online reviews, news comments, social media comments, and user check-ins, has become very popular in recent years. As microlength data, it is easy and quick for users to generate plenty of them every day. In fact, every day, over one billion users post more than four billions microblogs [104,331] on Facebook and Twitter. Such tremendous amounts of user-generated data have rich content, e.g., news, updates on on-going events, reviews, location information, language information, user information, discussions in politics, products, and many others. This richness has motivated researchers and developers worldwide to take advantage of microblogs to support a wide variety of practical applications [227,249], including public health [140,272], disaster response [101,144,145,156,157,161,304], public safety [325], education [354], real-time news delivery [8], geo-targeted advertising [256], and several disciplines of academic research such as social science [330],

Laila Abdelhafeez and Yunfan Kang have equal contributions and are ordered alphabetically.

This work is partially supported by the National Science Foundation, USA, under Grants IIS-1849971, SES-1831615, and CNS-1837577.

✉ Amr Magdy
amr@cs.ucr.edu

Laila Abdelhafeez
labde005@ucr.edu

Yunfan Kang
ykang040@ucr.edu

Eric Ong
eong001@ucr.edu

Mohamed F. Mokbel
mokbel@umn.edu

[1] Department of Computer Science and Engineering, University of California, Riverside, Riverside, CA, USA

[2] Department of Computer Science and Engineering, University of Minnesota, Twin Cities, Minneapolis, MN, USA

information modeling [270], human dynamics [308], engagement in education [328], political sciences [329], behavioral sciences [335], and even medical-related research [135]. The distinguished nature of microblogs data that combines large data sizes, high velocity, and short noisy text, has introduced new challenges, which motivated researchers to develop numerous novel techniques to support microblogs data management, analysis, and visualization at scale.

This paper provides a comprehensive review for existing major techniques and systems for microblogs data management since the inception of Twitter in 2006. The literature on microblogs is rich and includes several major research communities, e.g., data management, natural language processing, and information retrieval. However, this survey paper is addressed to the data management community that provides scalable infrastructures for indexing and querying microblogs and incorporate them in data management systems to enable managing this data at scale. The paper includes three main parts. The first part reviews core indexing and query processing components of microblogs data management, including their query languages and associated main-memory management techniques. The second part focuses on major genres of data management systems that are either designed for microblogs data or equipped with infrastructures to manage fast and large data, which are distinguishing characteristics for microblogs. The third part highlights major research topics that exploit data management infrastructures to build applications and analysis modules on top of microblogs, such as visual analysis, user analysis queries, and event detection. This part does not include other major research directions, e.g., natural language processing and information retrieval, as they are orthogonal to the data management research and out of the scope of this paper. In fact, dedicated survey papers review parts of their literature [80,117].

Figure 1 depicts a summary of different parts and the research topics that will be covered in this survey paper in a timeline format. The horizontal axis in Fig. 1 represents the year of publication or system release for each technique/system, while the vertical axis represents the research topic. The techniques are then classified into three categories: (1) techniques that deal with real-time data, i.e., very recent data, depicted by a filled black circle, (2) techniques that deal with historical data, depicted by a blank circle, and (3) techniques that deal with both real-time and historical data, depicted by a blank triangle. As the vertical axis of Fig. 1 depicts, the paper is organized around three main parts: *indexing and querying*, *systems*, and *data analysis*, each part is outlined below:

(1) *Data indexing and querying*: this part covers existing work for indexing and querying microblogs data that is depicted in the first to third rows of Fig. 1 and includes the following three topics:

- *Query languages*: this work provides generic query languages that support SQL-like queries on top of microblogs. This facilitates basic operators and advanced functions to express a variety of queries on microblogs.
- *Indexing and query processing*: this work includes various indexing and their associated query processing techniques that have been proposed to index incoming microblogs either in main-memory [50,51,211,223,229,305,360] or in disk [60,223]. This includes keyword search based on temporal ranking [51,60], single-attribute search based on generic ranking functions [211], spatial-aware search that exploits location information in microblogs [229], personalized social-aware search that exploits the social graph and produces user-specific search results [205], and aggregate queries [50,225,305] that find trending keywords and correlated location-topic pairs instead of individual microblog items.
- *Main-memory management*: this work includes techniques that optimize for main-memory consumption and utilization. Most microblogs indexing techniques depend on main-memory to manage microblogs in real time. Thus, some techniques are equipped for main-memory management such that memory resources are efficiently utilized, either for aggregate queries [225] or basic search queries that retrieve individual data items [224,229].

(2) *Data management systems:* this part highlights the current state and the challenges of managing microblogs data through major types of big data systems [18,21,24,26,51,223,245,315], depicted in the fourth to eighths rows of Fig. 1. In specific, we give a briefing on system challenges and motivational case studies to provide system-level data management for microblogs. Then, we highlight the data management features that are related to managing microblogs in the following system genres:

- *Specialized systems*: such as Twitter EARLYBIRD [51,245], TAGHREED [223], and KITE [228] that are designed considering the distinguishing characteristics of microblogs data and queries.
- *Big semi-structured data management systems*: such as ASTERIXDB [18] that is a generic big data management system to support various data sources. Recently, ASTERIXDB has extended its components to support fast data [121], e.g., microblogs, natively in the system. We review the fast data support in ASTERIXDB, which shows the current challenges of persisting fast data.
- *Fast data-optimized database systems*: such as VOLTDB [315] that is mainly optimized for database transactions on fast data, e.g., microblogs. We review the challenges of supporting transactional applications on fast data and solutions at the system level.

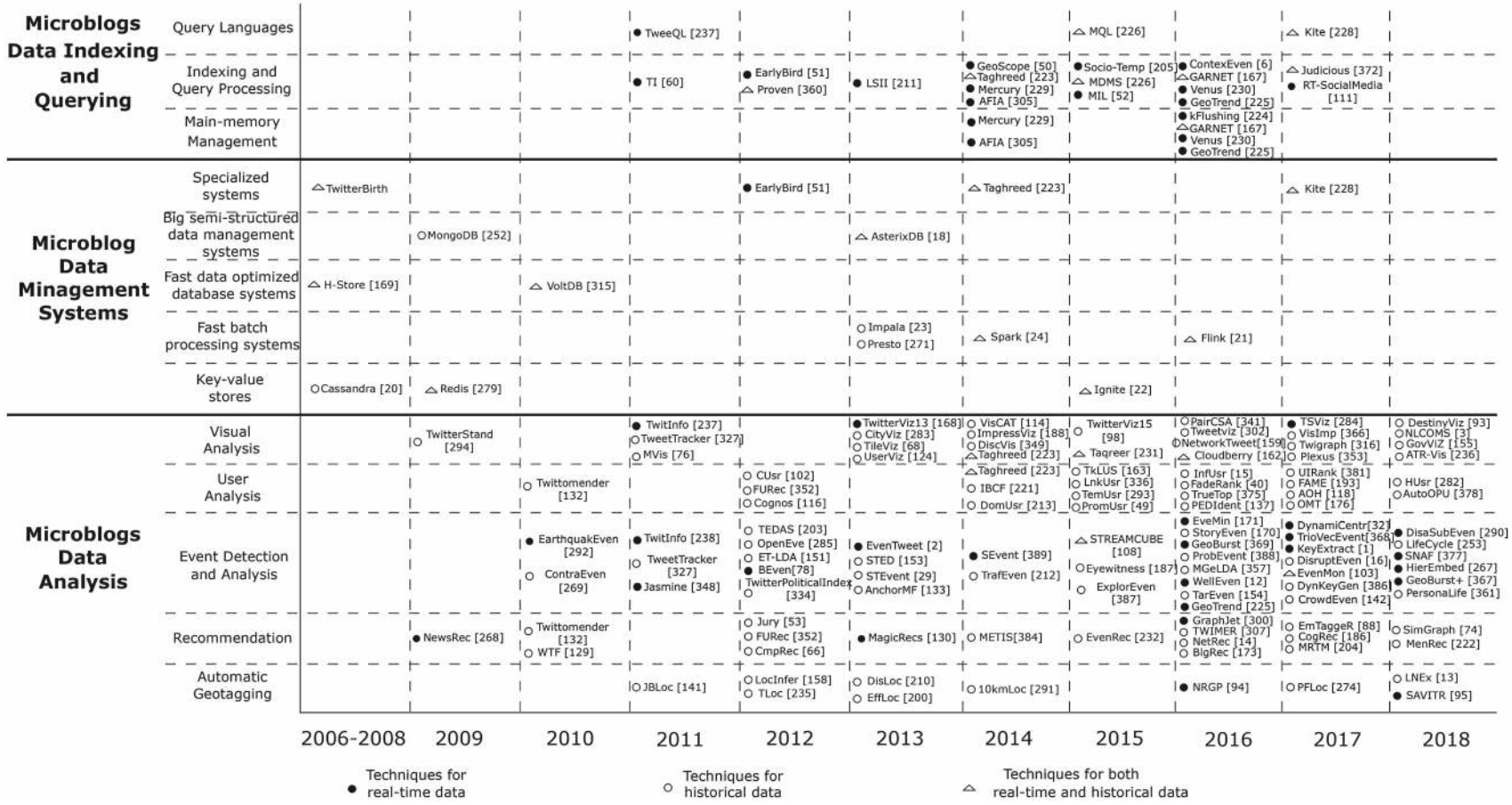



**Fig. 1** Microblogs literature timeline

– *Fast batch processing systems*: such as Apache SPARK [24] and Apache FLINK [21] that are optimized to process high-throughput applications on fast data via batch processing models. We discuss viable use cases as well as challenges and limitations of such systems to support efficient management for different microblogs applications.
– *Key-value stores*: such as Apache CASSANDRA [20] and REDIS [279] that store big datasets in key-value pairs. We discuss the adequacy of such systems to support certain microblogs applications as well as their limitations to support other applications.
– *Hybrid system architectures*: such as gluing stream processing engine, e.g., Apache STORM [26], with a persistent data store, e.g., MONGODB [252]. We discuss the challenges to manage real-time data in such setting showing the need to consider data velocity inherently in different system components.

(3) *Data analysis:* this part covers the *major* types of analysis on microblogs data that are depicted in the ninth to thirteenth rows of Fig. 1. The selected types of analysis are the ones that exploit the data management infrastructure to pose queries of massive number of microblogs and popular in the research community. This does not include either ad-hoc non-research applications, such as web applications that exploit microblogs data, or orthogonal research directions, such as linguistic analysis or information retrieval, which are intractable and dedicated surveys review only portions of them [80,117]. This part includes the following five types of analysis:

– *Visual analysis*: this work covers existing microblogs data visualization techniques that make use of the underlying scalable queries to enable visual analysis for excessive number of microblog records. This work use both aggregate queries, for aggregation-based visualization [93,114,284,316], non-aggregate queries for sampling-based visualization [223,294], or a combination of both [162,236,327].
– *User analysis*: this work is mainly interested in querying user information for different purposes, such as identifying top influential users in certain regions or topics [163,223,336] or discovering users with similar interests [34,132]. Such users, or group of users, can be used in several scenarios, including posting ads and enhancing their social graph.
– *Event detection and analysis*: this work exploits the fact that microblogs users post many updates on on-going events. Such updates are queried, grouped, and analyzed to discover events in real time [2,292] or analyze long-term events [238,327], e.g., revolutions.
– *Recommendation*: this work exploits microblogs user-generated content as means for catching user preferences to support diverse recommendation tasks, such as recommending content to follow [14], real-time news to read [268], authority users to follow [53], products [384], or users who share similar interests [132].
– *Automatic geotagging*: this work tries to attach geo-locations to microblogs data that are not geotagged based on analyzing their different attributes. This is mainly motivated by the small percentage of geotagged microblogs, e.g., less than 4% of tweets, that is faced by the need of many location-aware applications on top of microblogs, e.g., [2,229,256,294].

Other sporadic analysis tasks are addressed on microblogs data in both research community, e.g., news extraction [268, 294], and topic extraction [143,277], and industrial community, e.g., geo-targeted advertising [256] and generic social media analysis [324,382]. However, we outline the major analysis that exploit the data management infrastructure and include a wide variety of research techniques, which is of interest for the data management research community.

The rest of this paper details each of the three parts highlighting existing challenges, innovations, and future opportunities in microblogs data management research. Section 2 gives details of the data indexing and querying part. Section 3 gives details of the data management systems part. Section 4 gives details of the data analysis part. Finally, Sect. 5 concludes the paper and discusses different open problems in microblogs research.

## 2 Microblogs data indexing and querying

This section gives a comprehensive review for data management techniques that support large-scale indexing and querying for microblogs data. We first introduce microblogs query languages, in Sect. 2.1, that enable high-level declarative interfaces for querying microblogs. Then, Sect. 2.2 reviews the core indexing and query processing techniques. Finally, Sect. 2.3 outlines main-memory management techniques that are used in association with in-memory index structures.

### 2.1 Query languages

There are few attempts in the literature to standardize query languages tailored for the needs of microblogs, and inspired by SQL query language: TWEEQL [237] and MQL [226,228], each outlined below.

TWEEQL [237] is a wrapper over Twitter APIs[1] so the user can post SQL-like queries on top of Twitter data and the

[1] https://developer.twitter.com/en/docs/api-reference-index.html.

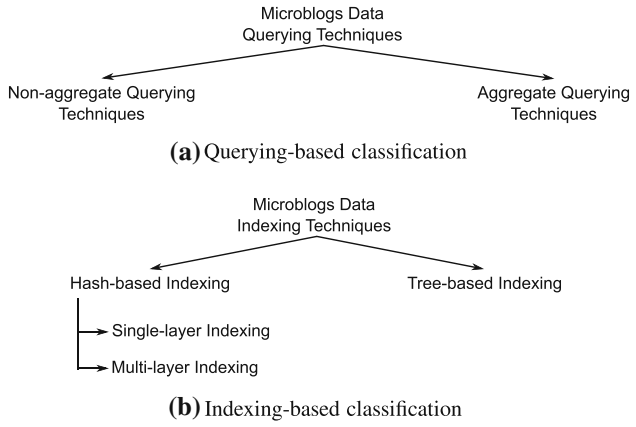



**Fig. 2** An overview of microblogs data management literature

underlying query processing is performed through accessing her Twitter developer account. TweeQL supports *Select-Project-Join-Aggregate* queries, recognizing aggregation as a major part of querying microblogs in several applications, e.g., trend discovery. In addition, TweeQL allows two additional constructs. First, built-in filters for the three major microblog attributes: keywords, spatial, and temporal attributes. Second, user-defined functions that allow higher-level analysis of tweets, such as automatic geotagging and sentiment analysis.

Unlike TweeQL, MQL [226,228], stands for *Microblogs Query Language*, is proposed as an inherent part of data management systems that support microblogs. MQL allows *Select-Project-Join-Count* queries, focusing on *count* as the only useful aggregate measure on microblogs. The major distinction of MQL is promoting *top-k* and *temporal* aspects as mandatory in all queries, arguing that there is no practical microblog query that can avoid these two aspects. Even if the user does not explicitly provide a top-*k* ranking function and temporal horizon for the query, MQL beefs up the query with default values. In addition, MQL supports filtering data based on arbitrary attributes, including spatial boundaries and keywords, and continuous queries similar to traditional data streams [58,125,250,343,356,358,391].

## 2.2 Indexing and query processing

This section reviews indexing and query processing techniques that are proposed to support large-scale querying on microblogs. Figure 2 depicts a high-level overview of this literature classified based on both query type (Fig. 2a) and index type (Fig. 2b). Based on query type, existing techniques are classified into *non-aggregate querying techniques* (detailed in Sect. 2.2.1) and *aggregate querying techniques* (detailed in Sect. 2.2.2). Based on index type, microblogs are indexed using either *tree-based indexing* or *hash-based indexing* that could employ a single or multiple layers of hash-based indexes. Table 1 provides more details summarizing these techniques in terms of the query attribute(s), index structure, index cell content order, and top-*k* ranking function. As the table shows, all existing queries on microblog include both temporal and top-*k* aspects regardless their other details. This is attributed to the nature of microblogs as they come in large numbers around the clock. This large number mandates retrieving the most useful *k* microblogs based on a certain top-*k* ranking function, otherwise, many useless data will be reported. In addition, being a kind of streaming data, the data is real time by nature and many users and applications are interested in recent microblogs. This inspired almost all the existing techniques to embed the time aspect by default in the query signature, unless it is disabled by the user. In fact, without using the time aspect, a query might retrieve data from several years ago, which leads to a significant querying overhead. So, by disabling this default option, users become aware of the implications on querying performance if they consider data of long temporal periods.

A generic query signature that represents all queries in Table 1 is: *"Find top-k microblogs/keywords ranked based on a ranking function F."* In non-aggregate queries that retrieve individual microblogs, the ranking function *F* can be *temporal* [6,51,60], *spatio-temporal* [229,230], *significance-temporal* [211], or *socio-temporal* [108] as shown in Table 1. In aggregate queries [50,167,225,305], the temporal aspect is used as a filter for queried data and the ranking functions mostly depend on keyword counts and their derived measure, e.g., trendline slope, except GeoScope that employs a correlation measure.

Almost all indexing techniques of microblogs are optimized for high digestion rates in a main-memory index for real-time data indexing, and secondary-storage indexing is assumed to have older data to query historical microblogs. The only exception is TI [60] that primarily uses a disk-based index. In addition, the query processing techniques are optimized for top-*k* and temporal queries. The rest of this section briefly outlines each technique that is shown in Table 1, for both non-aggregate and aggregate querying.

### 2.2.1 Non-aggregate indexing and querying

This section reviews non-aggregate querying techniques that *"Find top-k microblogs ranked based on a ranking function F."*, and retrieve individual microblog records.

TI [60] employs a disk-based inverted index structure where microblogs are sorted based on their timestamp. The main idea in TI is to defer indexing unimportant microblogs to reduce the number of microblogs that are indexed immediately and cope up with the large number of incoming data records. So, it keeps in memory a set of *recent and popular queries and topics*. Then, it categorizes each incoming microblog and decides whether it should be indexed

**Table 1** Summary of indexing and top-$k$ querying of microblogs

| | Query attribute(s) | Index structure | Cell content order | Top-$k$ ranking function |
|---|---|---|---|---|
| Top-$k$ non-aggregate queries | | | | |
| TI [60] | Keyword, Temporal | Inverted index | Temporal | Temporal recency |
| EARLYBIRD [51] | Keyword, Temporal | Inverted index | Temporal | Temporal recency |
| CONTEXEVEN [6] | Keyword, Temporal | Inverted index | Temporal | Popularity and temporal recency |
| MIL [52] | Keyword, Temporal | Multi-layer inverted index | Temporal | Keyword similarity |
| MERCURY [229], VENUS [230] | Spatial, Temporal | Partial quad-tree | Temporal | Spatial proximity and temporal recency |
| LSII [211] | Arbitrary, Temporal | Log-structured inverted index | Temporal | Significance, keyword similarity, and temporal recency |
| SOCIO- TEMP [205] | Social, Keyword, Temporal | 3D inverted index | Temporal | Social relevance, keyword similarity, and temporal recency |
| PROVEN [360] | Keyword, Temporal | Inverted index | Temporal | Provenance similarity |
| RT- SOCIALMEDIA [111] | Keyword, Spatial, Temporal, Social | Inverted index | Temporal | BM25, Spatial proximity, Temporal recency, Social distance |
| JUDICIOUS [372] | Keyword | Inverted index | TF-IDF | Keyword similarity |

| | Query attribute(s) | Index structure | Top-k ranking function |
|---|---|---|---|
| Top-$k$ aggregate queries | | | |
| AFIA [305] | Keyword, Spatial, Temporal | Spatiotemporal grid index | Keyword count |
| GEOTREND [225] | Keyword, Spatial, Temporal | Partial quad-tree | Keyword count and trendline slope |
| GARNET [167] | Arbitrary, Keyword, Temporal | Multi-dimensional grid index | Keyword count and trendline slope |
| GEOSCOPE [50] | Keyword, Spatial, Temporal | Hash index | Correlation between locations and topics |

immediately or deferred. The categorization considers the microblog recency, the user's page rank, popularity of the topic, and the textual relevance. The unindexed microblogs are written into log file, and an offline batch indexing is performed periodically to reduce real-time indexing latency. This is the first work to consider temporality in optimizing for microblogs, but it uses disk-based index solution which cannot scale for high microblogs arrival rate. The following techniques use in-memory structures that can digest fast data rates as well as providing low query latency, even though TI achieves higher indexing throughput compared to traditional techniques. Indexing time ranges from 0.1 to 1 s based on index parameter settings, whereas for the traditional index the indexing time is a constant 1.6 s. Query processing time ranges from 30 to 90 ms as the number of involved microblogs increases with growing answer size value. Query accuracy also increases with minimum of 90% for all settings.

EARLYBIRD [51]—*the core retrieval engine that powers Twitter's real-time search service*— is a distributed system where each node manages multiple inverted index segments to index keywords in real time. Incoming data first goes to a partitioner that divides tweets over nodes. In each node, ingested tweets first fill up the active segment before proceeding to the next one. Therefore, there is at most one index segment actively being modified, whereas the remaining segments are read-only. Each index segment is a traditional inverted index; however, postings for each term are maintained in reverse chronological order. It is worth mentioning that EARLYBIRD reduces the concurrency management overhead by adopting a single-writer multiple-readers model to eliminate any contention and race conditions. When a query comes, a blender receives it and determines which nodes should be accessed. Then, the query is posted to these nodes, the partial answers are retrieved and compiled by the blender to return the final answer. The experimental evaluation shows that EARLYBIRD achieves 7000 tweet/s indexing rate at latency of 180 ms.

CONTEXEVEN [6] also supports keyword search queries on real-time microblogs in favor of finding real-time content of the top-$k$ relevant events. It defines the event context with

a set of keywords and organizes incoming data in an inverted index based on these keywords. Each index entry maintains a list of event ids that correspond to a certain keyword ordered by a hybrid score that combines popularity and time recency, while an event is represented with a temporal tree that shows the chronological order of data within the same event [7]. To cope up with high velocity data, each index entry divides its posting list into buckets of exponentially growing sizes to reduce the insertion overhead in real time. In addition, ContexEven adapts a lazy update strategy for the index that defers updating the event id order until it is moved to the $(2 \times k)$th position, scarifying a slight query accuracy with real-time efficiency. The query processor then iterates over all index entries that correspond to the query keywords and aggregate the event final ranking score from all entries to return the final top-$k$ events.

MIL [52] is another event-based real-time search system that employs multi-layer inverted index that organizes event data based on keywords. The index has $m$ layers, each layer maintains a separate inverted index. The index key at the $i^{th}$ layer is a set of $i$ keywords that co-occur in certain events, while the posting list stores a list of event ids that correspond to these keywords. So, layer 1 key has a single keyword, while layer 2 key has a pair of co-occurring keywords and so on. A new microblog is inserted into all layers that correspond to different combinations of its keywords. Incoming queries also access all index layers to perform a nearest neighbor search based on cosine similarity. Experiments show that MIL outperforms variants of its competitor IL in search time, pruning power, and index update time. MIL search time is below 2 ms with different data sizes and query length, where pruning power is constant and it is almost 1. Index update time is less than 0.1 ms for up to 10 millions records.

Spatiotemporal ranking functions can be depicted in Mercury [229] and its successor Venus [230]. Mercury employs a partial quad-tree structure, where each cell contains a list of microblogs that have arrived within the cell boundary in the last $T$ time units, ordered chronologically. As traditional data insertions, expiration, and index structuring are very inefficient for real-time data, Mercury employs bulk data insertion, speculative index cell splitting, piggybacked deletion, and lazy cell merging to significantly reduce the overall indexing overhead and scale for real-time microblogs. The bulk insertion buffers incoming data and insert them every $t$ seconds, where $t$ is 1–2 s, to navigate different index levels once for several thousands microblogs. In addition, deletion and index structuring operations are piggybacked on the insertion navigation. For the index structuring, cell splitting is performed if and only if the cell exceeds maximum capacity and the microblogs in that cell will span at least two quadrants of the quad-tree node. Cell merging is deferred until at least three out of the four quadrant siblings are empty to reduce redundant splits and merges in real time. The query processing in Mercury has two phases, namely the initialization phase and the pruning phase. In the first phase, cells lying within the query range are ordered based on a spatiotemporal proximity score, and microblogs are retrieved from these cells based on their score. The pruning phase tightens the original search boundaries where microblogs outside the new boundaries are early pruned. This significantly reduces the total number of processed microblogs to get the final answer. Experimental results show that Venus supports high arrival rates up to 64,000 microblogs/s and average query latency of 4 ms.

LSII [211] supports top-$k$ queries based on combining three ranking scores for a microblog: its significance, its keyword similarity with the query, and its temporal freshness. A microblog is more significant if it is posted, for example, by an authority user or has high popularity with large number of forwards and replies. High keyword similarity indicates a high relevance to the query and freshness measures the temporal recency of the microblog. LSII consists of a sequence of $m$ inverted indexes where each index $I_i$ is double in size its predecessor index $I_{i-1}$. The first index $I_0$ is a read–write structure to which new data is appended, and the microblog list of each keyword is ordered chronologically. The indexes from $I_1$ to $I_{m-1}$ are read-only indexes, and each keyword has three microblog lists sorted with the three ranking scores. The small size and simple organization of $I_0$ enable high digestion rates of real-time data, while the three sorted lists of subsequent indexes enable efficient query processing. When index $I_{i-1}$ size reaches a certain threshold, a merge operation with index $I_i$ is triggered and index $I_{i-1}$ is flushed. To process an incoming query, LSII first scans $I_0$ to get the initial set of top-$k$ microblogs, then it proceeds in scanning other indexes. If the upper bound of index $I_i$ is no more than the scores of the top-$k$ candidates, index traversal is stopped and proceeds to the next one. Since each index $I_i$ is less recent than index $I_{i-1}$, the search is more likely to get pruned at earlier indexes since they have higher fresh scores. Extensions to LSII include personalized search, when a user is only interested in microblogs from specific users. Performance of LSII is compared to append only approach and Triple-Posting-List approach. The query processing time for LSII is less for both number of microblogs the query asks for and for the number of queries in the mixed stream of queries and updates. The query time is between 1 s and 10 s for varying number of microblogs asked by a query and increases from this range linearly with increasing the number of queries. The total processing time is almost 10 s and does not vary with changing the weights of the ranking function.

Another type of ranking is considering the social relevance as well as the textual relevance along with microblog freshness. A 3-D inverted index structure is proposed in [205] where each index cell is a three-dimensional data cube, a dimension for term frequency, a dimension for social rele-

vance, and a dimension for time freshness. Each dimension is partitioned into intervals; the social graph is partitioned with $k$-way partitioning using minimum cut utility. The time and textual dimensions are sorted at indexing time whereas social dimension is sorted at query time. Data is first partitioned by time; then, cubes in each time interval are indexed with a B+ tree to avoid maintaining many empty cubes. New data records are added to the last time interval. When the size of data in latest time interval exceeds a threshold, it is concluded and a new time interval is introduced. For query processing, cubes are first sorted by their estimated total score. Then, the query processor iterates over neighboring cubes and gets actual scores for microblogs. When the existing top-$k$ records are more relevant than the next unseen cube, the query processing terminates and prunes all remaining data cubes to ensure efficient query latency. The 3-D index outperforms both time pruning and frequency pruning, the two state-of-the-art techniques, with an average of 4–8x speedups for most of the parameter settings.

PROVEN [360] optimizes keyword search on microblogs for a unique similarity measure that depends on data provenance, measured through microblog content such as hashtags, URLs, and keywords. Incoming microblogs are grouped into bundles based on their provenance similarities and ordered based on their temporal evolution. An inverted index organizes bundles that are continuously updated with incoming microblogs. The inverted index has provenance elements, such as hashtags, URLs, and keywords as index keys and bundles as values. Through this index, incoming queries retrieve whole connected bundles of microblogs, which improves the search result relevance.

RT- SOCIALMEDIA [111] proposes a generic index structure for generic query function that can be extended to support temporal, spatial and/or social aspects. It proposes using the inverted index structure with a space-partitioning strategy in which the documents ids are partitioned into intervals, and each interval partitions documents based on keywords into blocks. To facilitate the top-$k$ retrieval, meta-data is stored within each block. The meta-data includes an interval id, a maximum score, and a bit map signature to determine which documents are present in this block. The maximum score is an upper bound for all documents in the block, so if the current top $k$th score exceeds this bound, the block is safely pruned. The signature field also provides a tighter bound to fasten the search process as absent documents are not included in the upper-bound value. Documents are sorted in the inverted index by the document id, so newer documents are appended to the end of the list to naturally support the temporal aspect. To support the spatial aspect, the index is extended with a uniform grid where in each cell we store the interval ids present in this cell, which helps to demote absent documents that do not appear the query cell. To support social aspect, the index meta-data is extended with a friendship bitmap, which helps to determine quickly if a user is a friend of another user. Experiments show that RT- SOCIALMEDIA reports better query latency compared to competitors in keyword search and spatial-keyword search, and better query latency in most cases as compared to LSII [211] in temporal keyword search.

JUDICIOUS [372] is the only microblog querying technique that does not consider temporality in their indexing. It offers a compact inverted index structure that treats rare terms, that are not frequently present in the data, differently from common terms, that are frequently present. For rare terms, a traditional inverted index is used. For common terms, a compact inverted index is proposed that uses block partitioning schemes, where microblogs are hashed into intervals, each interval is stored in a block with maximum score as meta-data for the block to facilitate early pruning in query processing. Thus, whole blocks are pruned if their maximum scores are not within the current query upper bound. Incoming queries has two types, singular queries that ask for one type of terms, either rare terms or common terms, and mixed queries that ask for both rare and common terms. Singular queries are answered from their corresponding index. In mixed queries, the rare item lists are retrieved first and used as fancy lists that tighten the query upper bound score and speed up pruning the search space. Experiments have shown that JUDICIOUS achieves 2–3 times query speedup over the state-of-the-art approaches with much smaller index size. For the same dataset, JUDICIOUS maintains an index of 35GB, whereas competitors BM-OPT and BMW-LB-PB maintain indexes with sizes 49GB and 50GB, respectively. Average response time on TREC queries ranges from 9 to 130 ms with increasing the number of keywords in JUDICIOUS, whereas it ranges from 25 to 290 ms in the other two techniques. With increasing answer size, JUDICIOUS average response time ranges from 21 to 30 ms, whereas the other two techniques range from 70 to 110 ms.

### 2.2.2 Aggregate indexing and querying

This section reviews aggregate querying techniques that *"Find top-k keywords ranked based on a ranking function F."* These techniques retrieve keywords, rather than individual microblog records, ranked based on aggregate information, e.g., frequency or frequency growth over time.

AFIA [305] retrieves top-$k$ frequent keywords that lie within any arbitrary spatial range and temporal interval. To support this at scale, AFIA maintains in main-memory a set of spatial grid indexes at different spatial and temporal granularities. Each grid cell keeps track of a summary of top-$k$ keywords that lie within its spatial and temporal ranges, using a modified version of the SpaceSaving algorithm [243]. At query time, the query range is mapped to the corresponding grid cells; summaries from all cells are merged together to get the top-$k$ keywords for the query spatiotemporal range.

Despite using the SpaceSaving algorithm the consumes small memory footprint, AFIA is consuming significant memory resources when supporting fine spatial and temporal granularities, as shown in [167,225], due to maintaining a huge number of summaries without supporting deletions or data expiration.

Unlike AFIA, GeoTrend [225] limits its search scope to recent microblogs and retrieves top-$k$ *trending* keywords that lie within any arbitrary spatial range within the last $T$ time units. GeoTrend accommodates various trending measures including trendline slope, which gauges the keyword frequency growth over time, and keyword frequency. To support this efficiently, GeoTrend maintains a partial quadtree structure where each cell contains aggregate information about keywords that arrive within its spatial boundaries. A list of top-$k$ keywords is materialized in each cell at indexing time. At query time, GeoTrend first gets local top-$k$ trending keywords within cells that intersect with the query boundaries. Then, to get the global top-$k$ trending keywords, the global trending value of each keyword is aggregated from local values, using Fagin's algorithm [105], and final top-$k$ keywords are returned. Experimental evaluation shows that GeoTrend supports arrival rates up to 50K microblogs/s, average query latency of 3 ms, and 90% query accuracy under limited memory resources.

GARNET [167] generalizes trend discovery to any arbitrary user-defined context instead of being limited to the spatial space. In specific, GARNET finds top-$k$ trending keywords within: (a) a $d$-dimensional context that is defined on arbitrary $d$ microblog attributes, and (b) an arbitrary time interval. For example, it could find trending keywords that are posted by teenagers in Spanish during July 2018. In this example, the context is two-dimensional and defined over age and language attributes. Each of the contextual attribute is divided into a set of discrete values or disjoint intervals, e.g., age attribute can be divided to *child*, *teenager*, and *elder*, while the language attribute can be categorized into *English*, *Spanish*, *French*, and *Others*. Then, a $d$-dimensional grid index is employed to map incoming data to the corresponding context gird cells. An in-memory grid index is maintained for recent data, and in-disk grid index is maintained for historical data. Each in-memory grid cell maintains a list of top-$k$ trending keywords over the last $T$ time units, while each in-disk grid cell maintains a temporal tree that maintains top-$k$ trending keywords for multiple temporal granularities over extended periods. At query time, top-$k$ keywords are aggregated from corresponding grid cells and a final top-$k$ list is compiled in a similar way to [225]. Experimental evaluation has conducted to show index scalability and query performance with different numbers of grid cells. The comparison with AFIA [305] has shown the superiority of Garnet. Garnet in-memory insertion time is below 400 ms for up to 24,000 microblog/s rate and reaches up to 1 s for higher rates. For varying grid cells, query latency ranges from 0.1 to 1 ms for both frequent and trending queries. The naive scanning alternative is not a competitor and increases query latency up to 1 s.

Unlike all other techniques, GeoScope [50] measures localized trending topics based on correlation between topics and a predefined set of locations, e.g., list of cities. The main idea of GeoScope to discover localized trending topics rather than topics that are popular all over the space. For example, a presidential election campaign is trending in many cities all over the country while a city council election campaign is trending only within a specific city. To this end, GeoScope limits the number of monitored locations to the $\theta$-frequent locations, keeps track of topics that is only $\phi$-frequent at least in one location, and then only tracks $\psi$-frequent locations of this topic. GeoScope has two main data structures: *Location-StreamSummary-Table* and *Topic-StreamSummary-Table*. *Location-StreamSummary-Table* maintains top frequent topics for each location while *Topic-StreamSummary-Table* maintains top frequent locations for each topic. At query time, these aggregate information are processed to retrieve topics that correlated only to the query location, distinguishing them from topics that are popular in all locations. Experiments show that GeoScope consumes almost constant amount of memory and reports constant amount of time with increasing window size. Also, it reports perfect recall and near-perfect precision.

### 2.3 Main-memory management

All major indexing techniques of microblogs store data in main-memory to be able to support real-time indexing for fast data and provide low query response time. However, with the rapid increase in number of microblogs, it is infeasible to store all data in main-memory for extended periods. At certain point, the available memory becomes full and part of the memory content has to be moved to a secondary-storage structure to free up memory resources for incoming microblogs. To this end, different indexing techniques use, implicitly or explicitly, flushing policies that decide on which microblogs to flush from main-memory to secondary storage. Although the problem of selecting memory content to evict has been studied before for the buffer management in database systems [97], anti-caching in main-memory databases [85,197,374], and load shedding in data stream management systems [33,112,138], flushing in microblogs data management is different in terms of the optimization goals and the anticipated real-time overhead as detailed in [224]. This section reviews the major flushing policies that are proposed in the literature to manage main-memory for microblogs data management.

Many of the major microblogs indexing techniques implicitly depend on temporal-based flushing [51,60,108,

211,305], where a chunk of the oldest data is flushed to disk to free up memory resources. The main intuition behind this simple policy is that: (a) recent microblogs are more important than old microblogs in several applications, and (b) incoming data, in these techniques, is indexed and ordered based on temporal recency, so flushing the oldest data will encounter very limited overhead in real time. This intuition is correct in a practical sense and gives the major advantage of the temporal flushing, which is its low overhead in real-time environments so its invocation does not limit the system scalability. However, it encounters a major limitation that affects both main-memory utilization and query latency. It underutilizes memory resources and stores $\sim$ 70% of memory data that is never reported to any incoming query, as detailed in [224]. The main reason is that flushing decisions depend solely on data recency without accounting for what is actually needed for incoming queries. Subsequent techniques in the literature have addressed such limitation for different types of queries as outlined below. The main objective of all these techniques is better utilization for main-memory resources, as useless data are evicted and useful data accumulates in main memory. This leads to increasing *memory hit ratio*, so more queries are answered from in-memory content without accessing disk content.

MERCURY [229], and its successor VENUS [230], provide flushing policies that decide on evicting non-aggregate data, i.e., individual microblogs. The flushing policy is optimized for top-$k$ spatiotemporal queries that retrieve microblogs from a spatial boundary $R$, and temporal interval of the last $T$ time units. By default, each index cell stores data from the last $T$ time units. MERCURY flushing policies provide two tighter time bounds, $T_c$ and $T_{c,\beta}$, both of them are no greater than T, where any data record outside $T_c$ or $T_{c,\beta}$ can be flushed to disk. The main observation behind finding such tighter bounds that highly populated areas, e.g., Downtown Chicago, has higher arrival rates than other areas. Then, top-$k$ microblogs can be retrieved from a shorter time than that of areas of less arrival rates. Thus, values of $T_c$ and $T_{c,\beta}$ are derived based on the local arrival rate, ranking function, and query parameters. $T_c$ ensures accurate query answers, which means any data record outside $T_c$ is not reported to any incoming query. On the contrary, $T_{c,\beta}$ employs a load shedding parameter $\beta, 0 \leq \beta \leq 1$, that allows to save up to $100 \times \beta\%$ of the memory with probability $\beta^3$ to miss a needed data record, trading off a slight decrease in query accuracy with a significant saving in memory resources. $\beta$ in this case is an input parameter by the system administrator. Experimental results show that compared with the default case where data from the last $T$ time units are stored, the policy consumes 65% less storage while achieving an accuracy of 98% to 99.5% when $\beta = 0.3$. At $\beta = 0.7$, 75% less memory are consumed and the accuracy is 97.5–99.3%. VENUS [230] extends this to provide an adaptive load shedding technique where the value of $\beta$ is adaptively calculated and automatically adjusted with the distribution changes in incoming queries and data. This leads to different $\beta$ value for each region, based on local data and query distributions, rather than a single global value for all regions. The strategy saves up to 80% of the storage while keeping an accuracy of more than 99% and is considered as significant enhancement over MERCURY.

KFLUSHING [224] is another flushing policy for non-aggregate data, . KFLUSHING accounts for a variety of top-$k$ queries for arbitrary attributes, ranking functions, and index structures. KFLUSHING performs flushing on three phases, a following phase is only invoked when the preceding phase cannot flush B% of memory, where the default value of B is 10. The first phase keeps only $k$ microblogs in each index cell and trims any records beyond k. The following phase removes the infrequent values of indexed attributes, e.g., keywords, with their associated microblogs in ascending order of their *latest arrival time*. If infrequent entries do not clear B% of memory, the last phase removes data in least-recently-used order. The main idea in all three phases is evicting data on the level of index entry rather than the level of individual microblogs. This significantly reduces the real-time overhead and scale in highly dynamic data environments. Comparisons with the *first-in-first-out* and *least-recently-used* policies are made to demonstrate the superiority of KFLUSHING. The results show that KFLUSHING increases memory hit ratio by 26–330% when compared with the existing flushing schemes and saves up to 75% memory resources.

GEOTREND's flushing policy, TRENDMEM [225], depends on aggregate information to evict data from main-memory. GEOTREND queries find top-$k$ trending keywords within an arbitrary spatial region and recent time, where different trend measures depend on keyword count. To effectively utilize memory resources, TRENDMEM evicts keywords that are consistently infrequent during all recent time periods, so they are unlikely to contribute to any top-$k$ trending query answer. Targeting consistent infrequency ensures not to miss a rising keyword. Therefore, TRENDMEM periodically removes $\epsilon$-infrequent keywords every $\frac{1}{\epsilon}$ insertions in each index cell, so dense spatial cells do not affect less populated cells. TRENDMEM achieves significant memory savings while maintaining highly accurate query answers.

GARNET [167] also provides a flushing policy that aims to use the minimal amount of memory rather than utilizing a fixed memory budget. The policy is tailored for its trending queries over arbitrary time periods. Each incoming microblog needs the past $N + 1$ index cells to calculate its trending measure. Thus, only these $N + 1$ cells are kept in memory and any older data is flushed to disk. If less than B% of the memory is flushed, GARNET flushes from the least recently arrived keywords till it reaches B%. Memory usages of TRENDMEM, GARNET, and AFIA are compared. By comparison, TRENDMEM consumes less than 10% of

**Table 2** Summary of systems features for supporting efficient management of microblogs data

| | Indexed attributes | Index/data storage | Supported queries | Flushing policies |
|---|---|---|---|---|
| *Existing systems features* | | | | |
| EARLYBIRD [51] | Keyword, Temporal | Main-memory | Top-$k$, Keyword, Temporal | Temporal flushing |
| ASTERIXDB [18] | Keyword, Spatial, Arbitrary | Disk | Keyword, Spatial, Aggregate, Range | N/A |
| VOLTDB [315] | Arbitrary | Main-memory | Aggregate, Range | N/A |
| TAGHREED [223] | Keyword, Spatial, Temporal | Main-memory and disk | Keyword, Spatial, Temporal | Temporal and top-$k$ flushing |
| KITE [228] | Keyword, Spatial, Temporal, Arbitrary | Main-memory and disk | Top-$k$, Keyword, Spatial, Temporal, Aggregate | Temporal and top-$k$ flushing |
| SPARK [24] | None | Main-memory and disk | Temporal, Aggregate | Least-recently-used flushing |
| FLINK [21] | None | Main-memory and disk | Temporal, Aggregate | Operator-dependant flushing |
| STORM [26] | None | Main-memory | Temporal, Aggregate | N/A |
| MONGODB [252] | Keyword, Spatial, Arbitrary | Disk | Temporal, Aggregate | N/A |
| Key-value stores [20,22,279] | Keyword, Arbitrary | Main-memory or disk | Aggregate | N/A |
| Minimum requirements and ideal system features for microblogs data management | | | | |
| Minimum | Keyword, Spatial, Temporal | Main-memory and disk | Top-$k$, Keyword, Spatial, Temporal, Aggregate | Temporal and top-$k$ flushing |
| Ideal | Keyword, Spatial, Temporal, Arbitrary | Main-memory and disk | Top-$k$, Keyword, Spatial, Temporal, Aggregate, Social, Range, User, Personalized | Temporal, top-$k$, and customized flushing |

AFIA memory, while GARNET consumes around 40% of AFIA memory. It is also shown that GARNET supports the highest arrival rate. The arrival rate supported by TRENDMEM is higher than AFIA and is also an order of magnitude higher than the current Twitter rate.

## 3 Microblogs data management systems

In this section, we highlight the major data management systems that support either microblogs data in particular or similar characteristics so microblogs data can be one of their use cases. Due to the plethora of new systems that are emerging in the data management literature, our review gives representative examples for each major genre of systems. We identified the major genres based on the adequacy of systems features and components to handle microblogs data. In specific, microblogs combine both large volume and high velocity aspects, where major novel techniques on managing microblogs data give particular attention to its fast streaming nature. Managing fast data has been recently got attention in many data management systems, from both academia and industry, which makes some of microblogs queries manageable in different systems genres. This section reviews five genres of systems: specialized systems that are designed and developed for microblogs, semi-structured data management systems, fast-data-optimized database systems, fast batch processing systems, and key-value stores. In addition, we highlight hybrid architectures that combine two different types of systems to manage microblogs, showing the limitations of this approach.

Table 2 summarizes the microblog-related features for systems that are reviewed in this section. It summarizes their capabilities in terms of indexing, supported queries, and flushing policies, highlighting the minimum and ideal requirements for efficient management of microblogs data. The rest of this section outlines different genres of systems, discussing their challenges, solutions, and limitations.

*Specialized systems* The literature has few systems that are specialized for microblogs data. A major example from industry is *Twitter* EARLYBIRD. As introduced in Sect. 2.2, EARLYBIRD system started as a distributed search system that powers real-time keyword search in Twitter [51]. However, Twitter added different functionalities [195,208,209,239,245,246,326] related to real-time data management, large-scale logging, and higher-level data analysis. We focus on one of such functionalities, which is real-time query suggestions, as it was a motivational use case to radically re-design the way Twitter is handling its real-time data and it shows the importance of radically re-thinking batch processing systems to support efficient queries on real-time data as detailed in [245]. When a user poses a keyword query, a query suggestion mod-

ule finds potential related queries to suggest to the user. For example, a user who searches for *football* might receives suggestions such as *soccer*, *FIFA*, or *world cup*. Twitter was supporting query suggestions through a query analyzer that employs Hadoop MapReduce to analyze the query log of EARLYBIRD system and produce the suggestions. However, using Hadoop has led to significant overhead where an hourly data is processed in fifteen minutes. This is much slower than the changes in Twitter queries distribution, which changes every few minutes [209,239]. Thus, fifteen minutes latency to process one-hour data is way behind such fast changes and has led to producing inaccurate query suggestions. To overcome this, Twitter beefed up EARLYBIRD system with in-memory query analyzer modules that directly access user queries through EARLYBIRD blenders (see Sect. 2.2). Each in-memory query analyzer maintains statistics about incoming queries with a ranking module to filter top related query suggestions. Every five minutes, the suggestions are persisted to a distributed file system, that represents a data store from where query suggestions are retrieved to end users. Such addition to EARLYBIRD system was motivational for Twitter to add several latency-sensitive components to their internal systems and radically re-design solutions that depend on batch processing systems such as Hadoop.

Another two examples of specialized systems that come from academia are TAGHREED *and* KITE systems. TAGHREED [223], and its successor KITE [228], were early end-to-end holistic systems that focus on microblogs data management in academic systems groups. In particular, both system designs inherently consider microblogs characteristics of both *data* and *queries* in indexing, query processing, and main-memory management. For data, they support fast and large volume data requirements. To this end, they employ both in-memory and in-disk index structures as core components to store, index, and retrieve recent and historical data. Indexes at different storage tiers are optimized for different objectives. In-memory indexes are equipped with fast data ingestion through batching incoming data and segmenting the index into small segment sizes that is lightly updatable. In addition, in-memory indexes are equipped with flushing policies that are responsible for moving a portion of memory content to disk when the available main-memory budget is full. Flushing policies are optimized to sustain system real-time operations as well as careful selection of victim data to evict to utilize memory resources to store useful data that serve incoming queries. For microblogs queries, they promote temporal, spatial, textual, and top-$k$ queries as first-class citizens through indexing and query processing. So, each of the two systems supports two families of index structures: a spatial index and a keyword index. Each index incorporates the temporal aspect in organizing its data, and in certain settings it incorporates the top-$k$ ranking function. Moreover, index segmentation is based on the time dimension in both memory and disk indexes. Disk indexes are optimized for efficient queries over arbitrarily large temporal periods through a richer segmentation setting. Basically, the data are replicated over different temporal resolutions, e.g., day, week, and month, so that querying data over several months still access limited number of index segments and provide a relatively low query latency. Other than indexing and query processing, both TAGHREED and KITE give a particular attention to main-memory utilization as a core asset to manage hundreds of millions of microblogs. For this, they provide different optimization techniques in their flushing policies so that most useful data accumulates in main-memory and obsolete data is moved earlier to disk.

Although TAGHREED [223] and KITE [228] share many characteristics in both objectives and system internals, TAGHREED is an earlier version of KITE that started to identify core components and requirements to support microblogs data and queries. Thus, TAGHREED focused in a single generic range query that allow to retrieve microblogs data within a spatiotemporal range and relevant to a set of keywords. Then, any further processing, e.g., top-$k$ ranking, is performed on top of TAGHREED query processor. KITE generalized this to allow querying any arbitrary attribute, while still promoting temporal, spatial, and textual as the prime attributes. Also, KITE added support for more advanced queries in the system components, such as top-$k$ queries and aggregate queries. Ideas in these systems are patented [251] and commercialized by a social media analysis startup company.

*Semi-structured data management systems* A major example of such systems is *Apache* ASTERIXDB [18] that is a distributed big data management system that has been developed by academic research groups, and has been recently incubated by Apache Foundation as a top-level Apache project [19]. ASTERIXDB is a general-purpose system that is designed to manage large volume, billion-scale, datasets that are limited to be managed efficiently in other systems. Recently, ASTERIXDB has introduced a core system component, called *data feeds*, to provide scalable ingestion and management for fast data [121], such as microblogs. A data feed digests and preprocesses raw data in main-memory. Then, data is forwarded to primary and secondary index structures. Each index is disk-based; however, it has in-memory components that aggregate data in main-memory before flushing them to disk-resident components. Data is accessible to the query processor when it is resident in the disk components. When data is congested, ASTERIXDB is equipped with different ingestion policies to select a portion of data to ingest promptly, while the rest of data is discarded or deferred. ASTERIXDB has achieved data digestion rates that are comparable to current Twitter peak rates with a cluster of five machines as experimented in [121]. Such performance is higher than what is reported by EARLYBIRD [51] in terms of data digestion per single machine. In terms of

digestion latency (or searchability latency), i.e., average time between a microblog arrives to being indexed and available in search results, AsterixDB data feeds provide low latency that are appropriate for real-time applications with certain ingestion policies, and significantly high latency with other policies. So, it is crucial to configure the system carefully for the underlying application needs. As a general-purpose system that is not designed for microblogs data, AsterixDB provides common utilities that fit for general fast data use cases without focusing on particular microblogs characteristics, such as temporal and top-$k$ query signatures.

*Fast-data-optimized database systems*. Although many of microblogs applications do not require transactional data management, database systems that are optimized for transactions on fast data are strong candidates to be used to handle some of microblogs queries, with optionally turning on or off the transactional features. This is due to their light weight management overhead with streaming data, while sustaining a high throughput of scalable queries. VoltDB is an example for such systems. VoltDB [315] is a distributed in-memory database management system (DBMS) that is designed and optimized to support high-throughput ACID database transactions on fast data. The system has started as an academic project, under the name of H- Store [169], that is commercialized by VoltDB [315,338]. The main additions of VoltDB to traditional disk-based database systems are driven by reducing the overhead of the database transaction manager. Particularly, VoltDB identifies four major sources of overhead in transaction management: (1) multi-threading that is required to manage multiple transactions concurrently, (2) buffer manager that swaps in data pages from disk to a main-memory buffer and evicts pages to disk on full memory buffer, (3) locking that is used to manage data consistency in concurrency control, and (4) logging that is essential in recovery management of completed transactions and rolling back aborted transactions. So, the four main contributions of VoltDB are to tackle such overhead sources to increase the throughout of transactions for fast data management. The multi-threading overhead is totally eliminated by assigning each transaction to a single dedicated CPU core. The buffer management overhead is totally eliminated by eliminating disk storage and storing all data in main-memory, so no buffer is managed in VoltDB. The locking overhead is also eliminated through determining deterministic orders for executing transactions through introducing global and local serializer components. The global serializer is a component that is aware of different data replicas on different machines, while the local serializer has the transactions details on a single local machine. Both components exchange information so the global serializer is able to provide each local replica deterministic orders for transactions, which leads to eliminating the locking overhead. Finally, the logging overhead is significantly reduced through logging data images instead of logging single transaction commands. In particular, VoltDB does not provide recovery management through the traditional write-ahead logging that mandates to write each transaction step to the database log file. Instead, only transaction parameters are written to file proactively. Then, in lazy basis, a full image of current data is written to disk for recovery purposes. This significantly reduces disk access and increases the throughput to 16,000 transaction per core per second, with almost linear scalability when adding more cores. This light management overhead has significantly lifted up managing fast data. Thus, VoltDB indexing and data management infrastructures are suitable to digest fast data efficiently and support important queries in real time, such as keyword queries. However, there are two major concerns for effectively supporting microblogs data end to end. First, VoltDB and similar systems are not optimized for large volume datasets, as stated in their technical documentation, which will lead to limitations in handling historical microblogs, over several months, that are richly exploited in different use cases. Second, it has no support for prime attributes of microblogs, such as the spatial attribute, which makes it inadequate for several important queries even on fast microblogs data.

*Fast batch processing systems*. Recently, a new generation of distributed batch processing systems has been emerged, extending Hadoop-like systems with main-memory data management infrastructures for efficient processing of large and fast datasets. Spark [24] and Flink [21] are prime examples for these systems. Both systems primarily process data in main-memory with options to connect to popular file systems, such as HDFS, or store statuses in persistent data stores, such as RocksDB [286]. As in-memory systems that support fast data through streaming packages, e.g., Spark Streaming [25], some microblogs applications could fit as use cases for these systems. However, unlike all reviewed systems earlier, Spark and Flink do not inherently support data indexing. Instead, they provide an advanced generation of batch processing systems, similar in spirit to Hadoop, that provide efficient parallel scans over all data records using commodity hardware clusters. Batch processing has limitations in several applications that need inherent indexing for either large volume or high velocity data. Newer systems, e.g., Apache AsterixDB, have tackled these limitations and provide different types of indexing for large and fast data. Obviously, many of microblogs applications are among these applications that require data indexing of several types as detailed earlier. For that reason, any system that gives particular attention to microblogs, e.g., Earlybird, Taghreed, or Kite, has provided different types of indexing for microblogs data. Furthermore, batch processing systems, such as Spark and Flink, do not consider query signatures that are popular in microblogs applications, e.g., top-$k$, spatial, and textual queries. This adds more overhead when powering large-scale microblogs applications on batch processing systems. The

pros and cons of SPARK and FLINK apply to other batch processing systems that share similar characteristics and architecture, e.g., Apache IMPALA [23] and PRESTO [271].

*Key-value stores*. A major genre of the emerging big data systems is key-value stores that work as massively distributed hashtables to store data in key-value pairs with various data models, e.g., Apache CASSANDRA [20], REDIS [279], and Apache IGNITE [22]. These systems are suitable for certain microblogs applications that require fast data ingestion with hash-based indexing, e.g., real-time keyword search. In fact, some of microblogs-oriented systems, e.g., EARLYBIRD [51] and KITE [228], are using the key-value store model to support in-memory keyword indexing. However, distributed key-value stores still lack other essential features that are needed in several microblogs applications, such as spatial indexing, temporal awareness, and top-$k$ query processing. Such shortcomings limit them from being an end-to-end solution for managing microblogs, yet they provide a solid foundation to build upon.

*Hybrid architectures*. An alternative way to handle fast and large data is gluing a streaming engine, such as Apache STORM [26], with a persistent data store, such as MONGODB [252]. In fact, MONGODB, a document-oriented database that provides several indexing and querying modules, has got a significant attention as a highly scalable database for persistent data, while Apache STORM has got similar attention for processing streaming data. However, each of them is designed and optimized for one aspect of big data, either large volume or high velocity, but not both. It has been experimented to glue these two systems in [121] to handle fast data that got persisted in large volumes. The comparison with Apache ASTERIXDB has shown up to two orders of magnitudes of higher digestion latency for the glued alternative, assuming that data is queried only when it is persisted to disk. Such significant overhead confirms the need of inherent support of fast data in the system components to provide scalable data indexing and querying. Similar conclusions are also drawn in other studies, e.g., [245], on the adequacy of adapting fast data management in systems that are optimized for large volumes. A major source of overhead is the incompatibility of system optimization goals, which leads to different decisions in different system components. For example, MONGODB is optimized for throughput, write concurrency, and durability, which leads to high wait time per single data write to disk and high ingestion latency in turn. Another source of overhead in such systems is the concurrency and transactions model that assume general-purpose applications with complex scenarios and requirements. This does not allow to use simple and scalable concurrency models, such as single-writer multiple-readers, that is adapted by several microblogs-oriented systems, e.g., [51,211,229].

## 4 Microblogs data analysis

The reviewed data management techniques and systems on microblogs have enabled to power a variety of data analysis tasks at scale. This section highlights the major data analysis research for analysis tasks that exploit the scalable data management infrastructures on microblogs to provide high-level functionality. As microblogs data analysis is a broad literature and include several topics that are not related to the data management community, this section limits its scope only to the analysis tasks that lie in the intersection of two categories. First, they have novel research contributions, which excludes a plethora of development applications that analyze microblogs data without addressing novel problems. Second, they exploit the querying techniques that are developed by the data management community. This excludes major research directions that are orthogonal from the data management research, such as natural language processing and information retrieval. In fact, these research directions have a rich literature where dedicated survey papers review parts of it [80,117]. The goal of this section is not discussing the details of various techniques. Instead, we present a high-level classification for techniques in the literature, and we summarize each topic through a generic framework that is induced from a variety of existing techniques when applicable. Then, we briefly highlight similarities or differences of each major technique in this topic compared with the induced framework. With such contributions, this section represents a road map for various microblogs data analysis that make use of the underlying data management infrastructures. We review major work in five main analysis tasks: *visual analysis* (Sect. 4.1), *user analysis* (Sect. 4.2), *event detection and analysis* (Sect. 4.3), *recommendations using microblogs* (Sect. 4.4), and *automatic geotagging* (Sect. 4.5). Finally, Sect. 4.6 briefly highlights other microblogs analysis tasks.

### 4.1 Visual analysis

Visualizing microblogs data has gained a particular attention due to the importance of end users interactions with microblogs applications, e.g., political and disastrous event analysis, disease outbreaks detection, and user communities analysis. The challenges faced in visualizing microblogs data align with the general challenges in visualizing other types of big data [43,59,99,100,128,178,263]. So, several pieces of the proposed research for big data visualization can be used for microblogs data as one type of big datasets. However, we review visualization work that targets a specific problem in microblogs datasets for different applications. In particular, microblogs have microlength content, which makes them easy to be generated by users all the time, e.g., a user can easily generate a tweet in a few seconds or less. This leads to generating a large number of data records in relatively

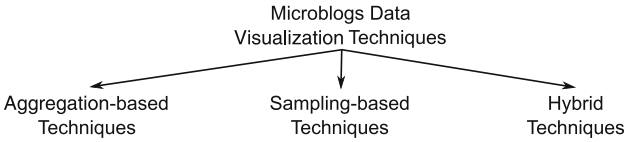
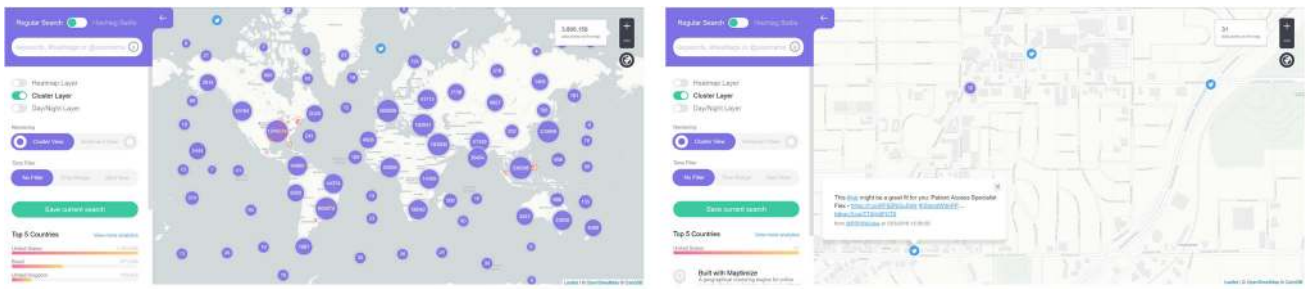



**Fig. 3** An overview of microblogs data visualization literature

short times. Visualizing such large numbers is beyond the capacity of existing frontend technologies, such as mapping technologies, e.g., GoogleMaps. So, visualization techniques that focus on microblogs try to address this problem by either aggregation, sampling, or a combination of both. Figure 3 classifies the visualization literature into three categories of techniques: (1) aggregation-based techniques, (2) sampling-based techniques, or (3) hybrid techniques. The visualization modules in all these categories use underlying querying modules, both aggregate and non-aggregate queries, to retrieve the data to be visualized. Thus, they directly make use of the scalable data management infrastructures that are built for microblogs. The rest of this section outlines each category of techniques.

*Aggregation-based visualization*. Techniques in this category [3,93,114,155,159,236,284,302,316,341,349,353,366] reduce the amount of data to be visualized through visualizing aggregate summaries of microblogs at different levels of aggregation, e.g., different spatial levels or temporal levels, rather than visualizing individual microblogs. Such aggregation is application-dependent and is usually performed either based on major attributes, e.g., temporal aggregation [93,155], spatial aggregation [114,349], or keyword aggregation [93,316], or based on derived attributes, e.g., sentiment [155,284]. Thus, these techniques are lossless and present all available information in a summarized form without ignoring any portion of the data. Aggregation could be based on a single attribute (one-dimensional) or multiple attributes (multi-dimensional). Figure 4 shows an example of aggregation-based visualization based on a single attribute, the spatial attribute [259]. In Fig. 4a, spatial regions that have a large number of data points visualize a variable-size circle that shows the number of points in this region. On the contrary, regions that have sparse data, Arctic Ocean and Norwegian Sea in Fig. 4a, visualize the actual data points. On zooming on the map view, more detailed data is visualized up to the street level that shows detailed data points, as depicted in Fig. 4b that shows street-level data in Riverside, California. Figure 5 shows an example of aggregation-based visualization based on two attributes, the spatial attribute and the language attribute [114]. In this case, number of microblogs is aggregated in each spatial region and the visualized circle categories data based on the language attribute to show percentage of microblogs posted in English, Arabic, Indonesian, Persian, etc.

The literature currently has seventeen visualization modules that employ only data aggregation based on microblogs queries. We next briefly outline each of them, highlighting their aggregation attributes and visualization format. **VisCAT** [114] aggregates data based on categorical attributes, e.g., language, and spatial and temporal ranges. **DisCVis** [349] aggregates tweets based on spatial region, language, and topics. **DestinyViz** [93] aggregates tweets related to certain games based on time, sentiment, and keywords. **NLCOMS** [3] aggregates tweets based on user communities and visualize them in a graph form. **GovViz** [155] aggregates data based on time, country, topic, keywords, sentiment, and content objects, e.g., links, images, and videos. **VisImp** [366] aggregates data based on communities and social interactions. **Twigraph** [316] aggregates data based on keywords and visualizes it in a graph form. **Plexus** [353] aggregates data based on topics and emoji objects in the textual content. **TSViz** [284] aggregates data based on time, sentiment, and hashtags. **PairCSA** [341] aggregates data based on their location stamps or location mentions, to get relation between users locations and the locations they mention. **Tweetviz** [302] aggregates data based on sentiment for business intelligence. **NetworkTweet** [159] aggregates external passenger flow and unusual phenomena based on spatiotemporal attributes, and uses trending keywords from microblogs to understand users' behavior. **TwitterViz13** [168] aggregates data based on tweets' intensity (tweet/second) and tweets sentiment. **CityViz** [283] aggre-



(a) Aggregated microblogs on continent level (b) Individual microblogs on street level

**Fig. 4** Example of aggregation-based visualization based on the spatial dimension

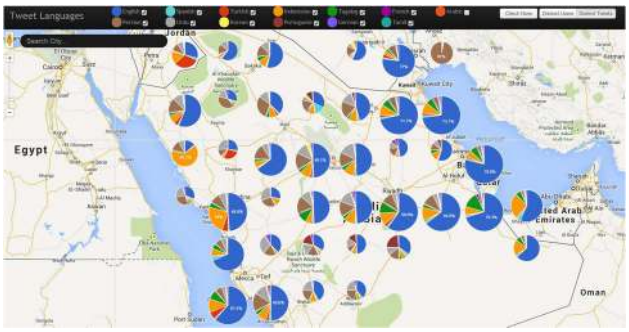



**Fig. 5** Example of aggregation-based visualization based on both spatial and language dimensions


**Fig. 6** Example of sampling-based visualization for tweets with different languages

gates data based on user behavior in cities to visualize periods intense/sparse of user activity. **TileViz** [68] generates summary statistics of the data for each tile for exploring the raw data set. **TwitterViz15** [98] provides two visualization views for Twitter data: (a) spatiotemporal analysis view, and (b) graph analysis view. The spatiotemporal view aggregates data based on spatial regions, sentiment, social bonds combined with spatiotemporal information, temporal evolution, and real-time statistics. The second view aggregates data based on social graph and real-time graph statistics. **ImpressViz** [188] aggregates textual and meta-data information to quantify user impression and visualize data in a six-dimensional impression space.

*Sampling-based visualization*. Techniques in this category [223,294,346] reduce the amount of visualized data through sampling. A sample of data is selected and visualized as a representative for the whole dataset, while the rest of data is not visualized. The sampling technique can be classified based on different dimensions. A sample could be a query-guided sample or an arbitrary sample. An example for a query-guided sample is OmniSci TweetMap[2] (Fig. 6) that samples tweets based their language as the query predicate filters data based on the language attribute. Another example is TwitterStand [294] (Fig. 7) that samples tweets based on textual content that have news stories. For certain

[2] https://www.omnisci.com/demos/tweetmap/.

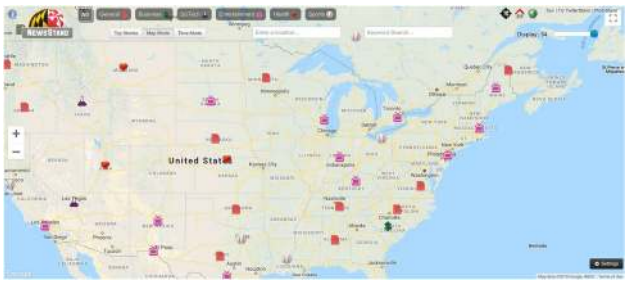


**Fig. 7** Example of sampling-based visualization for news tweets

queries, the query predicate is generating a lot of data that still cannot be visualized efficiently. In this case, applications, e.g., [223], select an arbitrary data sample to reduce the data size. Another classification of the way of sampling is based on the amount of data in the sample. The sample is either fixed or interactive. For example, TwitterStand [294] takes a fixed sample of data that contains new stories. Any interaction for end users with the map view, in Fig. 7, will not change the content of this sample. User interactions only change the subset of this sample that is shown on the map. On the contrary, an interactive sample changes the sample content based on user interactions. At the beginning, an initial sample of 100K, for example, is visualized from all languages including 30K English microblogs. When the user filters out data to show only English microblogs, the visualized English microblogs can be increased to 100K as it is solely visualized. Such interactive technique is exploiting the whole capacity of frontend technologies while increasing the overall amount of data visualized to users. Such technique is not heavily used and has several research challenges to support large-scale data.

Unlike aggregation-based techniques that are lossless, sampling-based techniques might be lossy or lossless depending on the application and the size of query result. If certain application queries are generating a reasonable sample size, then all data points are considered. Otherwise, such as in arbitrary sampling, a subset of data points are ignored and the sampling is lossy.

The literature currently has three visualization modules that employ only data sampling based on microblogs non-aggregate queries. We briefly outline each module highlighting the sampling attributes and stages. **CulTweet** [346] samples data based on language, country, and topic. **Taghreed** [223] performs two-step sampling. First, it samples data guided by query predicates based on spatial, temporal, and keywords. Then, if the sample size is still excessive, it performs an arbitrary sampling. **TwitterStand** [294] samples data based on textual content and spatial extent.

*Hybrid visualization*. Some applications allow to use both aggregation and sampling to reduce the amount of data to be visualized [76,162,236,238,240,327,365]. For example, event analysis applications [238,327] sample microblogs

based on their relevance to specific events. Then, event data need to be aggregated to summarize the event highlights to users, e.g., showing changes over time, space, users, or topics. Such applications usually do not encounter challenges in visualizing their data as the data size is reduced over two different phases, sampling and aggregation, which leads to significant reduction in their size and ease the visualization task. We highlight examples of such applications.

We highlight nine visualization modules that employ both data aggregation and sampling based on microblogs queries. We briefly outline each of them highlighting it different stages. **TweetTracker** [327] samples tweets that are relevant to a set of tracked long-term events; then, it aggregates them based on location, time, and keywords. **TwitInfo** [238] samples event-related data and aggregates them based on sentiment and spatial attributes. **ATR-Vis** [236] samples tweets that are relevant to a set of input debates; then, it aggregates and label tweets based on mentioned hashtags and the corresponding debate. **Cloudberry** [162] samples data based on keywords and aggregates it based on space and time. **TweetDesk** [240] provides a sample of top tweets of an event, along with a summary about the event. **ChineSentiment** [365] visualizes sentiment distribution based on temporal, spatial, and hot events. **EmotionWatch** [174] visualizes sentiment summary of public reactions toward events. It allows visualization of intense emotional reactions (peaks), controversial reactions and emotional anomalies. **UserViz** [124] analyzes users' connections and the frequency of tweets sent by one or a group of users, classifies these tweets, generates a tag cloud, and visualizes the most popular users. **Taqreer** [231] samples microblogs based on user-defined categories, e.g., different car models, defined by a set of keywords; then, data for each category is aggregated based on spatial and temporal ranges and visualized on map and aggregate views.

## 4.2 User analysis

The importance of microblogs in different applications originates from its user-generated nature, where hundreds of millions of users worldwide are posting around the clock. Among the major analysis directions is analyzing the user behavior related to different topics, locations, and communities based on their profiles and content of their microblogs. In fact, such kind of user analysis is highly overlapping in microblogs, i.e., the microlength user-generated data, and social media in general that include both short and long posts and objects, e.g., images and videos. This section limits its scope to analyzing microblogs users where excessive numbers of data records are generated compared to regular social media data due it its microlength.

Figure 8 classifies the literature of user analysis techniques on microblogs into techniques that either (1) find top-$k$ users

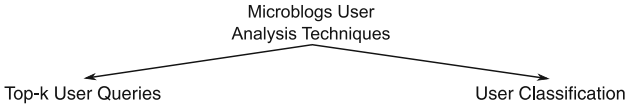



**Fig. 8** An overview of microblogs user analysis literature

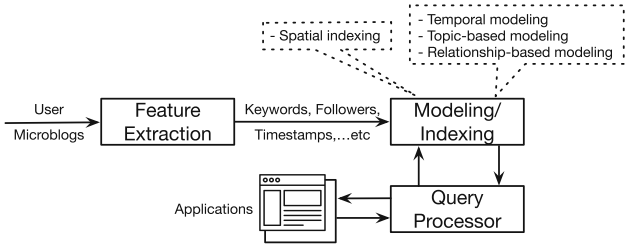



**Fig. 9** A framework for microblogs user analysis

according to a certain ranking criteria, e.g., top-$k$ influential users for a certain topic or top-$k$ active users in a certain location, to provide useful answers for higher-level applications, or (2) classify users based on certain characteristics. Top-$k$ users queries directly benefit from the data indexing and query processing techniques that are introduced in the data management literature to support different types of scalable top-$k$ queries based on various ranking functions. In fact, usernames are used interchangeably with keywords as string keys for the index structures, which makes many of the proposed data management techniques applicable to user queries. Figure 9 depicts a high-level framework for user analysis in microblogs that is induced from the existing literature. The framework consists of three main stages. First, microblogs of each user are fed into a feature extraction module to profile the user behavior through different pieces of information, such as keywords, followers/friends, timestamps, and locations. The actual extracted information is different in different applications. Then, the extracted user information is forwarded to an indexing (or modeling) module that produces a relevant index/model for users based on their information. Finally, a query processor accesses the index/model to answer application-level queries. As the description shows, the last two stages of this user analysis framework significantly make use of the data management techniques, and hence, new advancements in indexing and query processing techniques would positively affect the performance of user analysis tasks. Following the described framework, major techniques in the literature serve different applications with diverse purposes. We outline each category of techniques below.

*Top-k user queries*. Section 4.4 has reviewed several techniques that recommend top-$k$ users as potential friends, which overlaps with top-$k$ user queries. **DomUsr** [213] finds most influential users based on nine features that are aggregated through different models to calculate a final influence score. The used models are both aggregation and SVM

classification models. **TkLUS** [163] finds top-$k$ local users who are most active for a certain topic in certain location. TkLUS uses textual, social, and spatial relevance in hybrid spatial-keyword index to organize and retrieve top-$k$ users efficiently. **PromUsr** [49] finds prominent users for certain event through a probabilistic model that analyzes their temporal and textual information. **LnkUsr** [336] identifies top candidate user entities with limited information on microblogging platforms that can be linked to user entities on other platforms. It extends graph matching techniques with two heuristics to overcome the limited available information. **TwittOmender** [132] finds top users with similar interest to the querying user to expand homogeneous communities of similar interests. It profiles user posts content and use collaborative filtering techniques to find similar users. **TemUsr** [293] models users temporal behavior for different short-term and long-term topics. **CUsr** [102] samples microblogs data records, rather than sampling $k$ users, for efficient user community reconstruction based on strongly connected components. **IBCF** [221] uses dynamic user interactions in different topics to model the dynamics of relationship strength between users and topics over time. Then, the modeled relationships are used in matrix factorization recommendation model to improve social-based recommendation quality. **FURec** [352] predicts the top-$k$ users who will retweet or mention a focal user in the future by formalizing the prediction problem as a link prediction problem over an evolving hybrid network. **InfUsr** [15] finds most influential users in a certain topic. A nodal feature called focus rate is introduced to measure how focal users are on specific topics. Then, they incorporate nodal features into network features and use modified PageRank algorithm to analyze topical influence of users. **FadeRank** [40] evaluates the reputation of Twitter users. It summaries the past history in a bounded number of values and combines them with the raw reputation computed from the most recent behavior to assign a final ranking score. **TrueTop** [375] outputs top-$k$ influential non-sybil users among a given set of Twitter users. The system constructs an interaction graph and then performs iterative credit distribution using weighted eigenvector centrality as the metric to make the influential non-sybil users stand out. **UIRank** [381] identifies influential users whose tweets can cause the readers to change emotion, opinion or behavior. The algorithm is based on random walk and measures the user's tweet influence and information dissemination ability to evaluate the influence of the user. **FAME** [193] finds topical authorities on Twitter for a given topic. The algorithm adopts a focused crawling strategy to collect a high-quality graph and applies a query-dependent personalized PageRank to find important nodes that represent authorities. **Cognos** [116] identifies expert users for a certain topic through mining the meta-data of Twitter user lists that are created by the crowd. Lexical techniques are used to infer user expertise; then, experts in the same topic are ranked based on cover density ranking.

*User classification*. In addition to top-$k$ queries, user analysis is also performed to do user classification. **PEDIDENT** [137] identifies pro-eating disorder (ED) Tumblr posts and Twitter users. They use the associative classification algorithm CMAR to generate classification rules and train a classifier to identify pro-ED posts and users. **AOH** [118] classifies users into automated agents and human users using a random forest classifier. **OMT** [176] identifies the orientation of a user by analyzing tweets which mention more than one orientation using a logistic regression model. **HUsr** [282] identifies hateful users from twitter. They first sample users using a diffusion process based on DeGroot's learning model. Then, a crowd-sourcing service was adopted to manually annotate the samples. **AutoOPU** [378] detects the opioid users through a multi-kernel learning model based on meta-structures over heterogeneous information network.

### 4.3 Event detection and analysis

Event detection and analysis has gained tremendous attention with the rise of microblogging platforms [1,2,12,16,17,32,78,91,103,108,133,151,153,154,170,171,187,203,212,218,253,267,269,285,290,292,348,357,368,369,377,383,387–389]. The reason is the popularity of event-related updates that are posted by users through microblogs around the clock. This includes a wide variety of both short-term and long-term events, such as concerts, crimes, sports matches, accidents, natural disasters, social unrest, festivals, traffic jams, elections, and conflicts. Analyzing the event-related microblogs enabled several applications at different levels of importance, including crucial applications, leisure applications, and in-between applications. An example for crucial applications is rescue services and emergency response that have used microblogs to save hundreds of souls in different natural disasters since 2012 across the world [101,144,145,156,157,161,304]. An example of leisure applications is detecting surrounding entertainment events that are not collected in a single calendar, e.g., concerts, light shows, and special museum exhibitions in Los Angeles area. In-between both types, other types of applications have become popular, such as news extraction based on events [8], event-driven advertising [256], public opinion analysis for political campaigns [333,334], and analyzing protests and social unrest [28,257,332].

The advancements in microblogs data management enable significant performance enhancements in both tasks of events detection and analysis. As noted in the data management section, there are several state-of-the-art indexing and query processing techniques that are tailored for organizing and retrieving event data, such as ContexEven [6] and MIL [52] that are reviewed in Sect. 2.2.1. In a more general context,

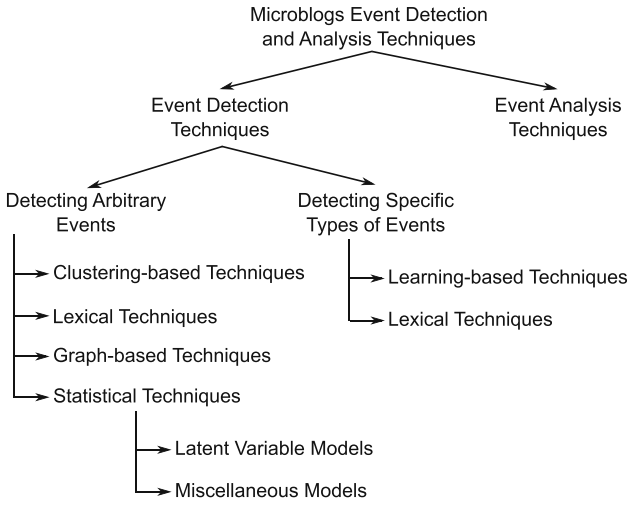



**Fig. 10** An overview of microblogs event detection and analysis literature

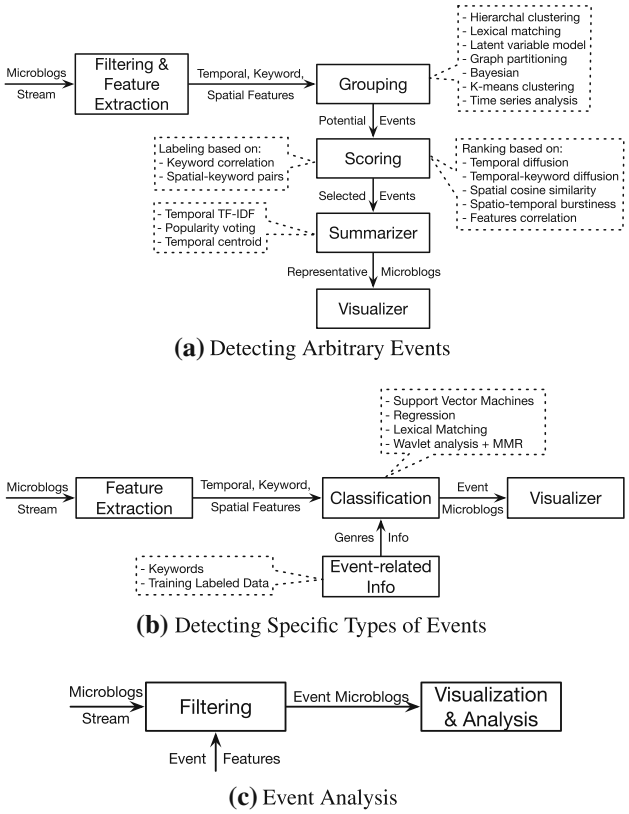



**Fig. 11** Frameworks for microblogs event detection and analysis

event detection makes use of indexing data based on temporal attributes that enables efficient retrieval of recent and temporally compact data, which is a major characteristics for grouping relevant data of a single event. In addition, indexing data based on spatial attributes gives an edge for discovering local events in geographic neighborhoods.

Figure 10 depicts an overview for the literature of event detection and analysis on microblogs. The rich literature is categorized into three main categories: (1) detecting arbitrary events, (2) detecting specific types of events, and (3) analyzing events. We summarize each category with a generic framework that is induced based on major work in the literature. Figure 11 shows three frameworks that correspond to the three categories. In the rest of this section, we review each category describing the different components of its framework and mapping existing literature to this framework highlighting similarities and deviations.

### 4.3.1 Detecting arbitrary events

A major direction of event detection research focuses on detecting arbitrary events that have either no predefined or at most very high-level characteristics. For example, finding coherent discussions on Twitter [32,78] without having a prior idea about what could be such discussions about. Another example is looking for local events in a certain city [2,108] without determining any specific characteristics of such events. These events are arbitrary events as the user does not provide a prior detailed description for the event characteristics. Figure 11a depicts a framework that is followed by most arbitrary event detection techniques. The framework consists of five main stages: (a) *filtering & feature extraction*, (b) *grouping*, (c) *scoring*, (d) *summarization*, and (e) *visualization*. A microblog dataset, either streaming or stored dataset, is processed through the filtering and feature extraction stage to identify potentially relevant microblogs and extract their temporal [2,16,108,187,285,348,368,383,389], textual [16,32,91,348,368,377,389], spatial [2,16,108,171,187,348,368,369,377,383,389], and semantic (part-of-speech (POS) tags/named entities) [218,285,290,357,368,387,388] features. These four types of features are the main drivers for detecting new events. Then, microblogs are forwarded to a grouping stage that assembles microblogs with similar features into groups, each group represents an event candidate. The grouping stage uses different types of techniques, including clustering [2,16,78,108,170,218,348,367–369], lexical matching [290,377], graph partitioning [32,91], and statistical techniques such as Bayesian [368], latent variable models [285,387,388], and regression models [187], as depicted in Fig. 10. The set of candidate events are then forwarded to a scoring module that gives a score (or a label) for each candidate to distinguish actual events from noisy groups. Scoring is performed in different ways, including labeling [171,187,367,368,383,387] or ranking candidates based on diffusion [2,78,285], similarity [16,32,78,170,218,285,377,389], correlation [91,290,348,388], and/or burstiness of different combinations of temporal, keyword, and spatial features [2,108,369,383]. For example, in scoring based on keyword correlation, if the group of microblogs has

scattered keywords that are not related to each other based on statistical co-occurrences of words, then this group is discarded as a noisy group that does not reflect an event. On the contrary, if the set of keywords are cohesive and with high co-occurrence likelihood in real topics, it is assigned a high score as an actual event. Several scoring techniques also consider temporal and spatial similarities besides textual-based measures. Then, top scored candidates are selected as actual events, while the rest of groups are considered noisy events. The events are then fed to an optional summarization module that identifies the most important microblog posts to represent a certain event using different signals of importance, such as popularity of the post, its temporal position, etc. Finally, the events are forwarded to a visualizer that displays representative microblogs along with their labels, content, locations, and temporal details to end users. The visualizer uses microblogs visual analysis techniques that are presented in Sect. 4.1, so the details of this are not duplicated in this section.

The described framework drives the major techniques in the literature. Tables 3 and 4 summarize the different stages of each technique. This literature can be categorized into four categories based on the grouping technique as the major stage that generates event candidates. Figure 10 depicts the four-category classification, namely *clustering-based techniques*, *lexical techniques*, *graph-based techniques*, and *statistical techniques*. Table 3 summarizes clustering-based and lexical techniques and Table 4 summarizes graph-based and statistical techniques. The rest of this section briefly outlines techniques of each category.

*Clustering-based techniques*. **EvenTweet** [2] proposes a framework to detect localized events in real time from a Twitter stream and track their evolution over time by adopting a continuous analysis of the most recent tweets within a time-based sliding window. Event candidate extraction is based on clustering keywords according to their cosine similarity of their spatial signature. Scoring the events is based on keyword burstiness and time diffusion from the cluster. Detected localized events are summarized by the number of related keywords and spatiotemporal characteristics. **STREAMCUBE** [108] system extracts microblogs hashtags along with spatiotemporal attributes. Then, hashtags are grouped through a single-pass hierarchical spatiotemporal clustering to detect event candidates, that are scored and ranked based on burstiness and local features. The system provides ways to explore events with different granularities in both time and location. **EveMin** [171] detects visual events based on photos and locations. Feature extraction calculates area weights and commonness score of words, grouping depends on word bursts using n-gram model and image clustering based on deep convolutional neural network (DCNN), and labeling uses another DCNN. **ReutersTracer** [218] extracts features based on named entities, while grouping uses a novel clustering algorithm that counts for microblogs features. **TrioVecEvent** [368] detects local events through extracting semantic textual, temporal, and spatial features that are used by a multimodal embedding learner to map correlated microblogs to the same latent space. Then, a novel Bayesian mixture clustering model finds geo-topic candidate events. These candidate events are then passed by a classifier that relies on the multimodal embeddings to label whether an event is a local event. **DisruptEven** [16] framework has both classification and clustering. The classification phase is used for filtering event-related posts from noisy posts and based on a naive Bayes model. Then, an online clustering is performed using temporal, spatial and textual set of features. After clustering, the framework offers event summarization using a novel temporal Term Frequency–Inverse Document Frequency (TF-IDF) that generates a summary of top terms without the need of prior knowledge of the entire dataset. **MGeLDA** [357] is a sub-event detection technique that extracts semantic features based on microtopics. The microtopics are identified by a novel mutually generative latent Dirichlet allocation (LDA) model for microblogs hashtags. Then, *k*-means clustering is used to group related topics and discover events. **StoryEven** [170] also introduces a model that summarizes each event as a sequence of sub-events on a timeline based on nonnegative matrix factorization (NMF) clustering.

*Lexical techniques*. **Jasmine** [348] extracts co-occurring words as well as geo-location and timestamp of microblogs. Then, microblogs that are generated within a short time and a small geographic area are grouped to form event candidates. Co-occurring words of each candidate are analyzed to distinguish noisy candidates from local events. **DisaSubEven** [290] extracts sub-events from a bigger event, e.g., a disastrous event has a series of small-scale emergencies such as a bridge collapsing, airport getting shut, and medical aid reaching an area. Feature extraction is based on POS tagging, grouping of sub-events is based on noun-verb pairs, and ranking is based on the frequency of co-occurrence of their constituent nouns and verbs in the corpus. For summarization, DisaSubEven uses an integer linear programming (ILP) technique that considers the maximum occurrence of nouns, verbs, and numerals.

*Graph-based techniques*. **DynamiCentr** [32] combines the first three stages of the framework depicted in Fig. 11a through extracting emergent keywords from incoming data streams based on analyzing the dynamic semantic graphs, where nodes represent the keywords and the edges are the co-occurrence of the keywords. Then, events are summarized based on the minimum weighted set cover applied on the semantic graph of the dynamically highly ranked keywords. **SNAF** [377] detects local events based on spatial and textual features of microblogs. It first filters event-relevant microblogs based on lexical analysis and statistical user pro-

**Table 3** Summary of clustering-based and lexical techniques that detect arbitrary events from microblogs data

| | Features | Grouping | Scoring | Summarization |
|---|---|---|---|---|
| EvenTweet [2] | Textual, Spatial, Temporal | Clustering: spatial cosine similarity | Ranking: Keyword burstiness, temporal diffusion | Number of keywords, spatiotemporal signature |
| STREAMCUBE [108] | Hashtag, Spatial, Temporal | Clustering: hierarchical spatiotemporal clustering | Ranking: Hashtag burstiness , local features | None |
| EveMin [171] | Textual, Spatial, Images | Clustering: word bursts using n-gram model, DCNN for images | Labeling: DCNN | None |
| ReutersTracer [218] | Named entities | Novel clustering technique | Ranking: newsworthiness probabilistic model | Cluster centroid |
| TrioVecEvent [368] | Semantic, Textual, Temporal, Spatial | Clustering: Bayesian mixture model | Labeling: multimodal embeddings classifier | None |
| DisruptEven [16] | Textual, Spatial, Temporal | Online clustering based on cosine similarity | Replaced with naive Bayes filtering | Top keywords ranked on novel temporal TF-IDF |
| MGeLDA [357] | Semantic, LDA microtopics | Clustering: $k$-means | None | Top frequent hashtags |
| StoryEven [170] | Textual, Temporal | Clustering: nonnegative matrix factorization | Ranking: probabilistic model | Sequence of sub-events |
| Jasmine [348] | Textual, Spatial, Temporal | Lexical: based on location and time | Ranking: co-occurring words | None |
| DisaSubEven [290] | POS tags | Lexical: noun-verb pairs | Ranking: noun-verb co-occurrences | Integer linear programming |

**Table 4** Summary of graph-based and statistical techniques that detect arbitrary events from microblogs data

| | Features | Grouping | Scoring | Summarization |
|---|---|---|---|---|
| DynamiCentr [32] | Textual | Graph-based: dynamic semantic graph | Ranking: dynamic eigenvector centrality | Min weighted set cover |
| SNAF [377] | Textual, Spatial | Graph-based: spatial connected components (CC) | Ranking: number of nodes in CC | None |
| GeoBurst [369], GeoBurst+ [367] | Semantic, Textual, Spatial | Graph-based: geo-topics based on spatial and semantic features | Ranking: spatiotemporal bursts [369], Labeling: supervised model [367] | None |
| EvenDetecTwitter [91] | Textual, Temporal | Graph-based: multi-assignment graph partitioning | Ranking: cross-correlation similarity (short-time events), Riemannian distance (long-time events) | None |
| ExplorEven [387] | Semantic, Temporal, Named entities, POS tags | Statistical: latent event and category model (LECM) | Labeling: matching entities with a semantic class | None |
| ProbEvent [388] | Named entities, POS tags | Statistical: unsupervised latent variable model (LEEV) | Ranking: correlation of named entities, dates, locations, and words | None |
| OpenEve [285] | Temporal, Named entities, POS tags | Statistical: latent variable model | Ranking: number of microblogs | Top phrases and entities based on a probabilistic model |
| Eyewitness [187] | Spatial, Temporal, Aggregate | Statistical: regression model | Labeling: anomaly threshold | Top microblogs based on text summarization |
| SpatialEvent [383] | Spatial, Temporal | Statistical: hidden Markov model (HMM) | Labeling: predefined taxonomy | None |
| SEvent [389] | Textual, Spatial, Temporal | Statistical: location-time constrained model | Ranking: similarity join | None |
| BEven [78] | Hashtags, Temporal, User | Statistical: 3D probabilistic model | Ranking: probabilistic model | None |

filing. Then, relevant microblogs are geotagged based on a large gazetteer and distance-based data cleaning algorithms. The cleaned data is then grouped into spatial connected components that represent events. **GeoBurst** [369] uses spatial and keyword features to build a keyword co-occurrence graph that is used to infer semantic features through random walks. Then, geo-topic clusters are formed as candidate events by combining both spatial and semantic features. A set of pivot microblogs are identified for each cluster, and then, they are ranked based on spatiotemporal bursts and top-$k$ are selected. **GeoBurst+** [367] differs from GeoBurst by employing a new supervised framework for selecting the local events, instead of burst ranking. In addition, it performs keyword embedding to capture the subtle semantics of microblogs. **EvenDetecTwitter** [91] framework identifies both short-term and long-term events. It first extracts temporal and textual features that include word frequency, conditional word frequency, inverse trend word frequency, fuzzy representation, and scale time modeling. The features are used to connect data in a graph model. Then, a multi-assignment graph partitioning scheme is employed so that each microblog can belong to multiple events. The similarity measure differs based on event type, for short-term events a cross-correlation similarity measure is used whereas for long-term events Riemannian distance is used.

*Statistical techniques*. This category can be divided into two sub-categories. First, *latent variable models*. **ExploreEven** [387] proposes a pipeline process of event filtering, extraction, and categorization. The filtering is based on lexicon matching and binary classification to opt only event-relevant microblogs. Feature extraction then processes relevant microblogs for time expression resolution, named entity recognition, part-of-speech (POS) tagging and stemming, and the mapping of named entities to semantic concepts. The event candidate extraction and grouping phase is based on an unsupervised latent variable model, called latent event and category model (LECM). For labeling a cluster, the most prominent semantic class obtained based on the event entities is employed as the event type. **ProbEvent** [388] extracts features through POS tagging and named entity recognition, groups microblogs based on a novel unsupervised latent variable model, called LEEV model, which simultaneously extracts events and generates visualizations, and scores candidate events based on the correlation between named entities, dates, locations, and words. **OpenEve** [285] extracts temporal, named entities, and POS tags, that are used to filter irrelevant microblogs through an event tagger module based on conditional random fields (CRF). The microblogs are then grouped based on latent variable model and ranked based on the association with event and time. Second, *miscellaneous models* that use different statistical methods, including regression, Markov models, graphical models, and temporal analysis. **Eyewitness** [187] extracts local events and summarizes them using time series analysis of geotagged tweet volumes from localized regions. The framework identifies features as count of data records based on spatial and temporal localities. Then, for a given region, a regression model is learned to predict volume of data versus data spikes as a function of time. Local event are identified when the actual volume exceeds the prediction by a significant amount. **SpatialEvent** [383] forecasts spatiotemporal events using an enhanced Hidden Markov Model (HMM) that characterizes the transitional process of event development by jointly considering the time-evolving context and space-time burstiness of Twitter streams. To handle the noisy nature of tweet content, words that are exclusive to a single event are identified by a language model that has been optimized by a dynamic programming algorithm to achieve an accurate sequence likelihood calculation. **SEvent** [389] detects related events, e.g., a sinking boat and an on-going flood in same spatial region. It first extracts textual, spatial, and temporal features. Then, a novel graphical model-based framework, called location–time constrained topic (LTT), is used to express each microblog as a probability distribution over a number of topics. To group related microblogs, a KL divergence-based measure is employed to gauge the similarity between two microblogs. Then, another longest common subsequence (LCS)-based measure is used for the link similarity between two sequences of user microblogs. Sequences are grouped based on spatial, temporal, and topical similarities. **BEven** [78] focuses on discovering breaking events and distinguishing real-life events from virtual events that happen only in the online community. Therefore, it categories microblogs based on three features extracted from the hashtags: (1) instability for temporal analysis, (2) meme possibility to distinguish social events from virtual topics or memes, and (3) authorship entropy for mining the most contributed authors. Based on these attributes, an unsupervised technique is used to categorizes microblogs into advertisements, memes, breaking events, or miscellaneous.

The rich literature of event detection on microblogs not only contains holistic frameworks that start with raw data and output events to end users, but also specialized pieces of work that are not proposing holistic frameworks; however, it either focuses on one or more of the stages or studies a problem that is utility for event detection. We outline examples for such work in the rest of this section.

**HierEmbed** [267] focuses on mining topics that are related to events in microblog streams. It presents an unsupervised multi-view hierarchical embedding (UMHE) framework that generates topics with a high accordance to the events from a microblog stream. The framework applies LDA to extract the feed-topic and topic-word distributions. Therefore, for each latent topic, there are two different view features, namely the latent word distribution and the relevant feed collection. Then, it applies a novel multi-view Bayesian

**Table 5** Summary of techniques that detect specific types of events from microblogs data

| | Event Type | Features | Classification | Event-related Info |
|---|---|---|---|---|
| TEDAS [203] | Crimes | Textual, Spatial, Temporal | Lexical matching | Keywords |
| TRAFEVEN [212] | Traffic | Textual | Wavelet analysis | Traffic data |
| DYNKEYGEN [386] | User-defined | Textual | Expectation maximization | Dynamic keywords |
| EARTHQUAKEVEN [292] | Earthquakes | Textual, Statistical, Contextual | SVM | Labeled data |
| CONTRAEVEN [269] | Controversial topics | Statistical, Sentiment, Linguistic | Regression | Labeled data |
| WELLEVEN [12] | Wellness | Textual | Novel supervised model | Labeled data |
| TAREVEN [154] | News-related | Hashtags, Mentions, Replies | Novel semi-supervised model | News articles, historical tweets |
| STED [153] | User-defined | Textual, Named entities | SVM | News articles |
| PERSONALIFE [361] | Personal life | Textual, POS tags | Multi-task LSTM model | Labeled data |
| CROWDEVEN [142] | Bus-related | Sentiment, Named entities | SVM | Labeled data |

rose tree (Mv-BRT) to refactor the latent topics into a hierarchy. A translation-based hierarchical embedding is formulated to encode the topics and relations in low dense vectors to better capture their semantic coherence. **ET-LDA** [151] proposes a joint model based on LDA to extract the topics covered in the event and tweets, and segmenting the event into topically coherent segments. **ANCHORMF** [133] solves the event context identification problem using a matrix factorization technique by leveraging a prevalent feature in social networks, namely the anchor information. A probabilistic model is built to consider users, events, and anchors as latent factors. An anchor selection algorithm is proposed to automatically identify informative anchors for the model. A Gibbs sampler and a maximum a posteriori (MAP) estimator are proposed to estimate the model parameters. **KEYEXTRACT** [1] focuses on extracting real-time local keywords through a time sliding window approach. For each keyword, a probability distribution over co-occurring places is estimated and used to eliminate spatial outliers. The spatial distribution is updated based on inserting new content and removing old content that is expired from the sliding window. **AUTOSUMMARIZE** [17] focuses on automatic summarization of Twitter posts using three methods, namely temporal TF-IDF, retweet voting, and temporal centroid representation. The temporal TF-IDF is based on extracting the highest weighted terms as determined by the TF-IDF weights for two successive time frames. The voting method considers the highest number of retweets a post received in the time window. The temporal centroid method selects posts that correspond to each cluster centroid.

#### 4.3.2 Detecting specific types of events

Another major direction of event detection research focuses on detecting specific types of events that have a set of distinguishing information to characterize the event, e.g., keywords. Examples of such events are crime events, earthquakes, or traffic jams. Crime events can be described by a set of keywords, while earthquakes are characterized by labeled training data, for example. In general, each event type is described by a set of event-related information. Figure 11b shows a framework that utilizes the event-related information along with incoming microblogs data to detect events of a specific type. The framework consists of three main stages: (a) *feature extraction*, (b) *event classification*, and (c) *visualization*. The incoming microblog data is processed to extract temporal [154], textual [154,203,212,386], spatial [154], and sentiment features [361]. Then, the processed data is forwarded to a classification model that uses the event-related information to distinguish relevant data to the event type of interest from irrelevant data. The classification can be performed through two different types techniques, as depicted in Fig. 10: (1) learning-based techniques [12, 142,153,154,269,292,361], such as support vector machines (SVM) [142,153,292] and regression models [269], and (2) lexical techniques [203,212,386]. The type of classification is also coupled with the type of provided event-related information that might be keywords or labeled training data. The classified relevant microblogs are directly fed to a visualizer that displays events to end users. The visualizer still uses one of the visualization techniques that exploit aggregation, sampling, or both as presented in Sect. 4.1. Compared to arbitrary event detection (in Sect. 4.3.1), this framework replaces the clustering and scoring modules with a classification model that exploits the event-related information to directly group and filter relevant data and reduce noisy output.

This framework drives the major existing work on detecting different types of events. Table 5 summarizes the different stages of each technique. The literature includes two categories of techniques based on the classification stage, as depicted in Fig. 10: *learning-based techniques* and *lexical techniques*. We briefly outline techniques of each category.

*Learning-based techniques*. This category includes both supervised and semi-supervised techniques. **EARTHQUAKEVEN** [292] detects earthquake events through Twitter. It uses SVM classifiers and labeled training earthquake data to classify earthquake-related tweets. **CONTRAEVEN** [269] detects controversial events through a regression classification model along with labeled training data on well-known controversial topics, such as Obama Nobel Peace Prize. **WELLEVEN** [12] extracts wellness events from tweets. It extracts features based on a graph-guided multi-task learning model, and classify data based on a novel supervised model that takes task relatedness into account. **TAREVEN** [154] detects social media events that are related to news reports. It extracts features from both tweets and news reports to find out relevant tweets. Then, relevant tweets are splits into positive and negative examples through an EM-based refinement algorithm and final relevance is computed based on textual, spatial, and temporal similarities. The data is then fed to a novel semi-supervised approach for detecting spatiotemporal events from tweets. **STED** [153] proposes a semi-supervised approach that enables automatic detection and visualization of user-defined specific events. The framework first applies transfer learning and label propagation to automatically generate labeled data, then learns an SVM text classifier based on tweet mini-clusters obtained by graph partitioning. Then, it finally applies fast spatial scan statistics to estimate the locations of events. **PERSONALIFE** [361] detects personal life events from users' tweets using multi-task LSTM model with attention. The system detects whether the tweet is an explicit event, implicit event, or not an event and then detects category of the event from predefined life events categories. **CROWDEVEN** [142] treats each bus-related tweet as a microevent which can be further analyzed for event type categorization, entity extraction, and sentiment mining. It uses CRF for entity extraction and one-against-one classification strategy with SVM as the classifier.

*Lexical techniques*. **TEDAS** [203] detects crime events based on crime-related keywords along with lexical matching to classify relevant data. **TRAFEVEN** [212] detects traffic events using related keywords along with wavelet analysis to classify relevant tweets. **DYNKEYGEN** [386] proposes a semi-supervised solution based on expectation maximization mechanism that leverages word information to infer tweet labels. The candidate tweets are selected based on a set of keywords, which are generated and updated dynamically based on word importance score that changes over time.

### 4.3.3 Event analysis

Unlike event detection techniques, where new events are outputs, event analysis techniques take an event as an input and analyze its data in different ways. In specific, event analysis work focuses more on providing exploration tools for known predefined events rather than detecting new events that are not known beforehand. For example, the Syrian revolution is a long-term event that is known beforehand with a set of features such as keywords and locations. So, an event analysis module is interested more in analyzing data of this well-known event rather than discovering a new event that is not known beforehand. Another example is King Tut festival in Hayward, California. This is a short-term event that is known beforehand with a set of keywords, locations, and a time period. Again, an event analysis module focuses more on analyzing data of this event without discovering any new events. Thus, most of existing event analysis work follows a simple framework that is depicted in Fig. 11c. The framework has two stages: (a) *filtering* and (b) *visualization & analysis*. The filtering stage employs simple filters on different attributes, e.g., keywords [29,237,238,327], spatial [29,327], and temporal [29], to extract relevant microblogs to a certain event, e.g., Hurricane Sandy. Then, extracted data is forwarded to a rich visualization module that enables end users to analyze event data based on multiple views, e.g., map view, aggregate views, frequent keywords, influential users, timeline view, sentiment view, or individual microblogs. The features of analysis and visualization views are highly variant and depend on the application and the analysis purpose. The rest of this section presents examples of event analysis applications in the literature.

**TWEETTRACKER** [327] provides an event analysis framework for long-term events, such as Arab Spring uprisings, Occupy Wall Street, and US presidential elections. Users can define new *jobs* to define new events to analyze. Events data are filtered based on keywords, locations, and usernames features. Newly incoming data is tracked based on the event features for a long term. Then, the collected and new data is visualized based on a time series view, geographic map view, trending keywords view, entities view, and individual tweets view. TweetTracker has currently collected 3.2 billion tweets, and it is adding new $\sim$ 700,000 tweet every day. **TWITTERPOLITICALINDEX** [334] is a social media index for US presidential elections co-developed by Twitter and Topsy Labs, a social search and analytics company that owns all Twitter data and is acquired by Apple Inc. [27]. The index visualizes tweets relevant to US elections based on political party, sentiment, locations such as states and counties, and timeline view. **TWITINFO** [237,238] provides a timeline-based event analysis framework that allows users to define events based on relevant keywords. Then, the system collects relevant tweets, categorizes them based on sentiment, and organize them in timeline and map views in both aggregate and individual data records forms. The system addresses scalability problems that are associated with analyzing and visualizing such large number of data records. **STEVENT** [29] analyzes events based on three aspects. First, how topic initiators influence popularity of the topic. Second, the impact of

geography on popularity by partitioning the Twitter network according to regional divisions and studying the behavior of popular and non-popular topics. Third, the effect of topology and the dynamics of topic spread on popularity.

#### 4.3.4 Events and microblogs aggregate queries

Several aggregate querying techniques (Sect. 2.2.2) have been motivated by detecting events from large-scale microblogs data [50,225,305]. This includes detecting highly frequent [305] and highly trending [225] keywords that identify popular topics among users, and detecting highly correlated keywords with different locations [50] that identify localized topics of people interests. Such techniques can be used as scalable infrastructures to detect events from large amount of data. However, the core research methods focus on indexing and query processing on a large scale, which lies in a lower-level of the data analysis stack compared to techniques that are reviewed in this section.

### 4.4 Recommendations using microblogs

Microblogs represent a rich and up-to-date source for user-generated content. Therefore, they are appealing for several recommendation applications to extract up-to-date user preferences, which is essential to recommend relevant items. Although recommendation applications that exploit microblogs data are diverse, being an up-to-date source for user preferences is the common theme that links all of them. From a data management perspective, having such large and highly changing data as a source of preferences introduces significant challenges in updating recommendation models in practice. In fact, this has triggered deep research discussions in the data management community on the ability to support recommendation models efficiently in data management systems [109,181,191,192,295–298,362]. This clearly makes a transformative shift toward a new generation of recommender systems that should be able to recommend relevant items accurately through updating models much more efficient than their ancestor generations of recommender systems. Microblogs data plays a major role as a source of preferences for this new generation of recommender systems and the data management research community is in the heart of addressing their challenges.

Figure 12 depicts a high-level overview about recommendation techniques using microblogs. The literature includes two major recommendation problems, recommending content and recommending friends, in addition to a set of diverse miscellaneous recommendation applications. Such applications are as diverse as recommending news items, products, question answers, events, and scholarly information. The rest of this section highlights each category.

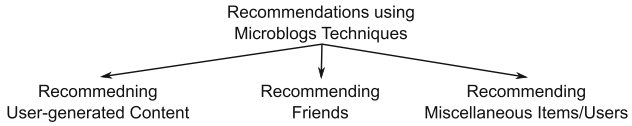



**Fig. 12** An overview of recommendations using microblogs literature

*Recommending user-generated content*. One of the major recommendation problems that is widely studied in the literature is recommending user-generated content, such as recommending other microblogs to read, hashtags to search, and mentions to post. **NETREC** [14] recommends tweets that are not visible to the user, e.g., posted by friends of friends or further, by exploiting the social network, content, and retweet analysis. The importance of invisible tweets is initially estimated by the social distance. Then, both content analysis and user analysis are performed to rank highly relevant users to recommend their tweets. Content analysis is based on textual analysis using bigrams, while user analysis is based on comparing timelines and mutual retweets. **BLGREC** [173] leverages and combines the user's location, social network feeds, and in-app actions to infer the user's interest and develop a personalized recommendation model. A user's feed is then made up of recommended content, including trending news, social network feeds, and social content, on either local or global scales based on the user spatial interests. **TWIMER** [307] performs tweet recommendations based on formulating a query based on the user's interest profile to probabilistic language models. Then, irrelevant and near-duplicate tweets are discarded using threshold-based filtering, locality sensitive hashing, and tweet freshness. **SIMGRAPH** [74] is a scalable recommendation model based on a similarity graph that induces the mutual interest among users by analyzing retweets. The probability of a certain user to like incoming microblogs are estimated based on a propagation model that aggregates top-$k$ tweets and recommends them to the user. **CMPREC** [66] tackles a more fundamental functionality in microblogs recommendation through comparing two approaches to compute similarity among microblogs with brief content: a topic-based approach and WordNet corpus-based approach. The study shows the superiority of WordNet corpus to catch similarity between brief textual content of microblogs.

Hashtag and mention recommendation is another content recommendation task that is popular in the literature, so users can easily search for their topics of interest. **EMTAGGER** [88] is a trained model for learning word embeddings and assigning hashtags with the trained embedding system. **COGREC** [186] proposes two cognitive-inspired hashtag recommendation techniques based on the Base-Level Learning (BLL) equation: $\text{BBL}_{I,S}$ and $\text{BBL}_{I,S,C}$. BLL accounts for the time-dependent decay of item exposure in human memory, once with the current tweet content ($\text{BBL}_{I,S,C}$)

and once without ($BBL_{I,S}$). **MRTM** [204] is a personalized hashtag recommendation model based on collaborative filtering and topic modeling. It integrates user adoption behaviors, user hashtag content, and contextual information into a joint probabilistic latent factor model to recommend hashtags to users. **MENREC** [222] addresses the problem of using both texts and images of microblogs for mention recommendation. A cross-attention memory network is proposed which considers the content of a tweet, interests of the user, and interests of the author to recommend a user to be mentioned for a certain tweet.

*Recommending friends*. Another recommendation problem that is widely studied in the literature by researchers from academia and industry (specifically Twitter) is recommending users to follow to expand and enhance the social graph connected components. **TWITTOMENDER** [132] started exploiting the real-time nature of microblogs by dynamically profiling the users through their recent microblogs. Then, collaborative filtering techniques are used to recommend users with similar interests. **FUREC** [352] tackles the problem from a different angle and recommends top-*k* users who will likely interact with microblog posts of a certain focal user. It uses the existing follower network and creates a new network based on retweets and mentions, then a single hybrid network is composed to recommend the new users. The problem is also studied and realized by Twitter Inc. [129,130,300], where substantial contributions in enriching connections between Twitter users are made. The *Who to Follow* (**WTF**) project [129,300] started to recommend users to follow and enrich Twitter social graph. The core of the system is the *Cassovary* in-memory graph processing engine and a novel technique for performing user recommendation, called *Stochastic Approach for Link-Structure Analysis* (SALSA). SALSA constructs a bipartite graph that include the user's circle of trust on the left side, while the right side includes users who are followed by the users in the left side. Then, this bipartite graph is traversed and ranking scores are assigned, on which users are recommended accordingly. Approximation algorithms are also provided in the second generation of WTF to reduce the complexity of processing hundreds of millions of users. To exploit the time aspect of Twitter data, they added MagicRecs [130] that recommends users who are followed by friends within certain temporal constraints. To expand Twitter's recommendation services, they added content recommendation through GraphJet [300] that is based on a bipartite graph similar to the one maintained in WTF system, except the right side models actual user tweets. A random walk on this graph with a fixed probability of reset outputs a ranked list of vertices that represents the tweets to be recommended to the user.

*Miscellaneous recommendation applications*. A significant portion of the literature is recommending miscellaneous items/users, where the common theme is using microblogs as an up-to-date source for user preferences. **NEWSREC** [268] recommends news items re-ranking based on user preferences extracted from tweets. The user tweets and RSS news feeds are both processed by a preference extraction module that finds out common keywords in both. Then, these keywords are used to promote relevant news in the news feeds timeline, so important news appear early to users. **METIS** [384] recommends products based on detecting purchase intent from microblogs data in near real-time fashion, combining their model with the offline traditional models that are similar to e-commerce website recommendations, e.g., Amazon. Such exploitation of real-time user-generated data has enhanced the effectiveness of product recommendation models. Another recommendation model that handles cold-start problem for product recommendation exploiting user-generated microblogs is **CSPR** [385]. CSPR uses data from microblogging users with no historical purchase records to map users' attributes extracted from microblogs into feature representations learned from e-commerce websites. Thus, given a microblogging user, a personalized ranking of recommended products can be generated to overcome the cold-start problem. **EVENREC** [232] exploits geotagged microblogs to recommend events from Eventbrite, a popular event organization website. The extracted events depend on microblogs locations that are fed to item-user models. This work is orthogonal from event recommendation in event-based social networks [84,122,347], e.g., Meetup.com, which has different nature compared to microblogging platforms, and thus, it is beyond the scope of this paper. **CRAQ** [312] recommends potential answers to a posted question through selecting a group of potential authority users who are selected based on their topically relevant microblogs. Then, the candidate group is iteratively filtered by discarding non-informative users, and top-*k* relevant microblogs are determined as potential answers. **JURY** [53] recommends potential authority users who are able to answer a given question. It adapts a probabilistic model that selects a set of users so that the probability of having wrong answer is minimized. **SCHREC** [364] recommends scholarly information through microblogs posted by researchers who post about their latest findings or research resources. Two neural embedding methods are proposed to learn the vector representations for both users and microblogs. Recommendation is made by measuring the cosine distance of a given microblog and user.

### 4.5 Automatic geotagging

Geo-locations are heavily exploited in several microblogs applications, such as localized event detection [2], geo-targeted advertising [256], local news extraction [294], user interest inference [126], and finding local active users [163]. With all such importance of geo-location data in microblogs applications, still the majority of microblogs are not asso-

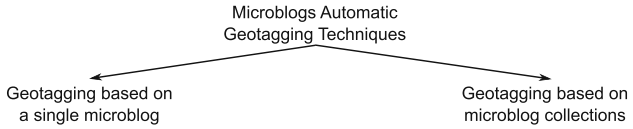



**Fig. 13** An overview of microblogs automatic geotagging literature

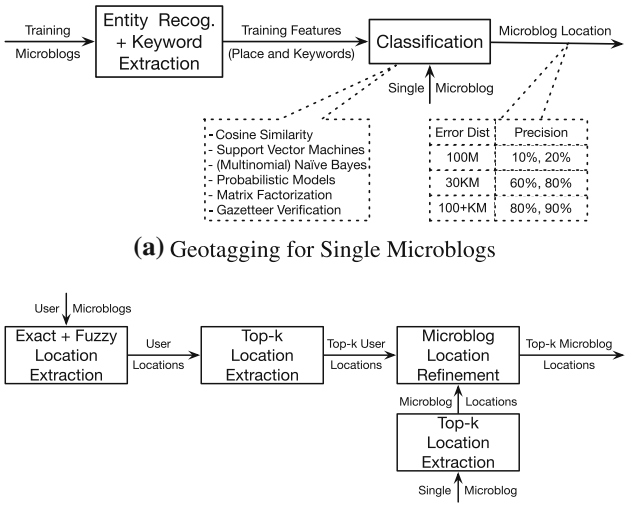



**Fig. 14** Frameworks for microblogs automatic geotagging

ciated with precise location information. In fact, a small percentage ($< 4\%$) of popular microblogging data, e.g., Twitter, is associated with locations sourced from user devices. This triggered a need to associate location information with more microblogs data automatically to exploit as much microblogs as possible in location-aware applications. However, traditional geotagging techniques are limited for enriching microblogs location data due to the brevity of microblogs textual content. Such brief text contains a lot of abbreviations and noisy words that make it hard for named entity recognizers to extract accurate places and locations. In this section, we give an overview about new techniques in the literature that are designed to extract locations from microblogs data. Although traditional geotagging techniques purely depend on linguistic analysis to extract locations, recent geotagging techniques on microblogs go beyond this to identify top-$k$ locations for both users and data records, as elaborated later in this section. This recent paradigm overlaps and makes use of certain indexing and query processing techniques from the data management literature. Thus, automatic geotagging on microblogs is leaning toward making more use of data management infrastructures in addition to the linguistic techniques.

Figure 13 classifies the literature at high-level into techniques that use a single microblog record at a time for geotagging and techniques that use collections of microblogs. Figure 14 shows frameworks for the two types of techniques. In fact, most of microblogs geotagging techniques in the literature depend on classification models to assign location(s) to one microblog at a time. Figure 14a shows a geotagging framework that is induced based on existing work on microblogs. The framework consists of two stages. The first stage is a feature extraction stage that extracts keywords and named entities places from the brief textual content of training microblogs. The extracted keywords and places are used to train the classification model. For each incoming microblog, the classifier assigns a location based on its textual content features. The location classification is performed through different models, such as probabilistic models [199,274,291], multinomial naive Bayes [141], lexical matching [158], ensemble of statistical and heuristic classifiers [235], pure place entity recognition [210], gazetteer verification [13,95], and matrix factorization [94].

A common problem in these techniques is the trade-off between error distance and classification precision. The precision is significantly dropped down for the practical margins of error distance, which represents the distance between actual location and predicted location. For example, with error within 100 m, the precision ranges from 10–20% for different techniques. On increasing the error distance to 30 KM, the precision is raised to 60–80%. With 100+ KM error distance, the precision reaches 80–90%. Therefore, accurate location prediction provides very low precision where over 90% of data is mistakenly geo-located. On the other hand, the significant increase of error distance makes predicted locations not useful for practical applications.

To overcome this problem, a state-of-the-art technique [199] proposed to process microblogs as collections instead of individual records as depicted in Fig. 14b. The technique is collecting all microblogs of each user as one collection and perform exact and fuzzy location extraction on them to identify all possible locations for this user. Then, top-$k$ locations for each user are predicted and identified as the most likely locations where the user is posting microblogs. When a new microblog arrives, a set of top-$k$ locations are extracted from microblogs content and metadata. Then, the $k$ microblog locations and the $k$ user locations are fed into a location refinement module that predicts the final top-$k$ microblog locations. This technique has shown tremendous enhancement in prediction precision and recall (95+%) within 100 m error distance, which is the threshold for accurate location prediction.

## 4.6 Other analysis tasks

The reviewed analysis tasks in previous sections represent the major high-level analysis tasks on microblogs that are of interest to the community of data management and analysis

researchers. However, the microblogs literature and applications are so rich to enumerate all possible analysis types or techniques. In fact, other analysis are sporadically addressed on microblogs in both (1) academic community, such as news extraction [268,294], topic extraction [143,201,277], summarization [47,96,119], situational awareness [289,303], and resource needs matching [41,42], and (2) industrial community, such as geo-targeted advertising [256] and generic social media analysis [324,382]. Yet, the reviewed literature represents the main performed high-level analysis tasks that span a wide variety of interests, applications, and novel research challenges as well as future research opportunities.

## 5 Conclusions and future directions

This paper has provided a comprehensive review for major research work and systems for microblogs data management and the corresponding analysis tasks. The paper categorized the literature into three parts: data indexing and querying, data management systems, and data analysis, where each part is further divided into sub-topics. The data indexing and querying part has reviewed microblogs query languages, individual indexing and query processing techniques, and main-memory management techniques. The systems part has reviewed characteristics of different genres of big data systems, e.g., batch processing systems, big data indexing systems, and key-value stores, in terms of their adequacy to handle microblogs query workloads. It has also discussed challenges and solutions that are provided through these systems for fast data, highlighting their potential limitations to handle certain microblogs applications. The data analysis part provided a detailed roadmap for the major analysis tasks that are directly or indirectly make use of the data management literature: visual analysis, user analysis, event detection and analysis, recommendations, and automatic geotagging. For each task, we presented a generic framework, when applicable, that is induced from major techniques in the literature and drives main research innovations for this task. In addition, we classified the literature based on the major component of this framework to provide better understanding for different techniques and highlight existing challenges and future opportunities in this research direction.

The rich literature of research on microblogs data faces several big challenges and is still rich with opportunities on different fronts. In terms of data management, there are several research opportunities in real-time indexing, query optimization, and system-level integration. For real-time indexing, the microblogs literature does not provide a comprehensive study for supporting spatial-keyword queries on real-time data. This has not been studied before either in existing spatial-keyword querying techniques [54–56,63–65,69,72,73,127,196,200,206,219,223,233,234,343,371,373] that focus on traditional static datasets, e.g., restaurants, or in existing microblog indexing that considers the spatial-keyword combination only in aggregate queries that retrieve frequent or trending keywords [50,225,305]. Existing specialized systems for microblogs supports two separate indexes, a keyword index and a spatial index, as a generic option that allows supporting various queries with few system assets. However, it is not clear how much performance is lost compared to hybrid indexing strategies. Quantifying such performance losses will enable better understanding for parameters that control querying performance on different indexes, which in turn will allow optimizing each index. Such understanding contributes to developing query optimization models for real-time data management as elaborated below. In addition to spatial-keyword queries, social information is still underutilized in supporting scalable personalized queries on real-time microblogs data. Although there exist few techniques that exploit this information [205,211], these queries still suffer from inherent scalability limitations due to the overhead of supporting hundreds of millions of users while sustaining efficient data digestion, indexing, and querying in real time.

Despite the richness of exploring real-time indexing on microblogs, there is almost no work on studying the implications of these novel indexing techniques on query optimization models. For example, the traditional selectivity estimation models assume relatively stable index content that is dominated by read operations and encounter much less write operations. This assumption does not hold on microblogs real-time indexes that have highly dynamic content. In addition, microblogs indexes are segmented based on temporal and spatial ranges, which gives a room for estimation model compression to serve such excessive amount of data with limited storage requirements. In general, the implications of new real-time indexing techniques on traditional query optimization models need to be revisited on microblogs.

Integrating all existing and future techniques of microblogs data management in end-to-end systems is a must to widen the impact of existing data management technology in microblogs applications. Recently, extensive efforts started to develop end-to-end systems to support microblogs data as elaborated in Sect. 3. However, there is still a gap between the available research techniques and their applicability for system-level integration. For example, existing aggregate queries techniques face challenges to be integrated with microblogs systems as they cannot be supported efficiently using existing indexes and require separate indexes. This is not favorable from system point of view to maintain additional indexes. So, new ways need to be innovated to integrate aggregate data structures within index cells of non-aggregate queries at a system level. Another example is flushing policies that are way developed in separate indexes

than the ones supported at system level. This due to a lack of integration techniques that allow flexible flushing policies while maintaining the real-time performance.

In terms of data analysis, there are several untackled challenges on two levels: enhancing the analysis modules and integrating them with microblogs systems to extend their functionality for enriching and facilitating microblogs applications. There are many examples that can be induced from the reviewed literature. We will highlight few of them in different analysis areas. First, developing a unified event detection framework that allows users to express different types of event-based queries. Such framework will exploit the rich literature of event detection and analysis on microblogs to provide common utilities that allow effective and efficient event queries. Second, real-time geotagging of microblogs data. Although recent work started to tackle this problem [94,95], there are still challenges in reducing the geotagging time due to the high computational cost of this task. Achieving the goal of attaching locations to microblogs as they come will widely impact a plethora of location-aware applications that are built on top of microblogs. Third, integrating the rich literature of user analysis techniques with the scalable data management infrastructures, e.g., indexes and query processors, in microblogs systems. Such integration will allow a variety of user-centric applications to be supported at scale. Fourth, developing a unified recommendation framework that exploits microblogs data and allow users to express a variety of recommendation queries flexibly. Such framework will serve a diverse set of applications that are reviewed in Sect. 4.4. The envisioned unified framework could exploit existing work on supporting generic recommendation queries in data management systems [198,296–298].

In addition to enhancing different analysis modules, there is a dire need to integrate such rich literature of analysis techniques with microblogs data systems to widen the impact of microblogs research in a practical sense. Such integration will have tremendous impact of a plethora of applications that benefit the society, the research community, and business applications, including public health, disaster response, public safety, and education. The feasible way to achieve such goal is abstracting different analysis tasks on microblogs into basic building blocks that can be supported in microblogs systems, inspired by SELECT-PROJECT-JOIN building blocks in SQL database management systems. Such task is huge and shall be started with developing generic frameworks for different analysis tasks, as discussed earlier for event detection and recommendations as well as provided throughout Sect. 4.

# Appendix

## A Orthogonal research directions

This appendix gives an overview about sentiment and semantic analysis in microblogs as an example of an orthogonal research direction, from the natural language processing literature, that does not exploit much of the data management infrastructures. The appendix highlights the differences of new techniques on microlength data with the corresponding techniques on traditional long data. For detailed surveys about these topics, the reader can refer to [80,117].

### A.1 Sentiment analysis

Sentiment analysis automatically discovers the polarity of feelings expressed in a chunk of text, e.g., a citizen posts positive or negative opinions about certain election candidate. Traditional sentiment analysis techniques make use of the microblogs brevity to enhance the classification accuracy of user sentiment. As reported in [46], using a traditional sentiment classification technique on microblogs boosts the accuracy up to 10% higher for binary sentiment. This boost is an absolute advantage of the content brevity that makes it less confusing and more decisive to catch positive and negative feelings in user-generated content. However, microblogs brevity introduces both challenges and differences compared with traditional data. For example, feature extraction is more challenging due to lots of abbreviations and noise, e.g., extracting meaningful keywords is harder. In addition, compared with traditional data where sentiment is analyzed on three different level, document level, sentence level, and entity level, microblogs short content mostly limits the sentiment scope to a single sentence or a single entity that represents the whole microdocument. Moreover, microblogs come with additional advantageous features that were not available in traditional data, such as links, user information, and their interactions with different topics. Thus, the sentiment analysis research on microblogs has addressed a wider variety of challenges compared with traditional sentiment analysis. In this section, we give an overview about this rich literature.

Figure 15 depicts an overview of the microblogs sentiment analysis literature. The major techniques can be categorized into four main categories, namely, *machine learning techniques*, *lexical techniques*, *hybrid techniques*, and *miscellaneous techniques*. The machine learning techniques represent the majority of techniques in the literature. It could be further categorized into four sub-categories as depicted in Fig. 15, namely, *supervised*, *classifier ensemble*, *deep learning*, and *semi-supervised*. The first subcategory of techniques use supervised machine learning, i.e., traditional classifiers [4,36,67,87,202,220,254,255,276,

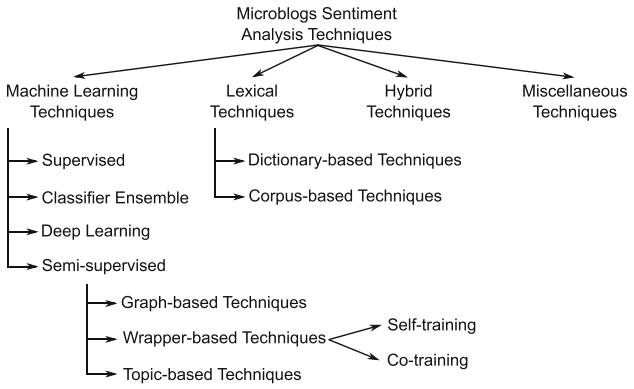



**Fig. 15** An overview of microblogs sentiment analysis literature

281,318,351,393]. The differences among these techniques are the classifier type, stages, and features used to distinguish sentiment. The major used classifiers are support vector machines (SVM) [4,5,31,35,39,83,87,120,131,160,164,172,180,248,262,275,306], (multinomial) naive Bayes (MNB and NB) [31,35,120,131,247,262], *k*-nearest neighbor (kNN) [31,82], MaxEnt [92,120,180], random forest (RF) [89,319], logistic regression [36,202,393], and AdaBoost [185]. The used features include different types of language-based features such as unigrams [5,35,120,164,247,262], bigrams [35,120,262], trigrams [262], n-grams [82,180,185,248], and POS tags [5,39,120,180,185,247,248,262], microblog-specific features [185,248] such as retweets [39], hashtags [35,39,164], emotions [35,39,164,180], links [35,39], and other features such as punctuation-based [5,82,180,248], pattern-based [5,35,82,164,180,248] and semantic-based [180,247].

To enhance the classification accuracy, techniques of the second sub-category ensemble multiple classifiers [61,70,75,79,86,136,172,185,194,208,244,266,317,363]. The set of used features is almost identical to the single classifier techniques, while the used classification algorithms are overlapping but not identical. In specific, SVM [79,136,266], NB [70,136], MNB [79], logistic regression [79,136,208], and AdaBoost [185] are still used, while new classifiers are also introduced such as neural models [86,136,363] and Bayes network [136]. A third sub-category is deep learning, which is an emerging field in machine learning. In the past few years, deep learning is getting increasing popularity and many learning problems migrated to deep learning frameworks. Deep learning offers a black box of neural networks that are trained with huge amounts of data that offer better accuracy over traditional classifiers. In the case of microblog platforms, huge amount of data is generated daily, which has motivated the use of deep learning techniques for sentiment analysis on microblogs. Existing deep learning techniques is exploited in short textual contexts in two-step fashion [37,62,71,86,90,134,147,148,165,265,280,288,321,322,337,342,359]. It first learns word embeddings, and then, it applies them to produce representations for the text sentiment.

The main limitation of all supervised techniques, either with single classifier, multiple classifiers, or deep learning models, is the sensitivity to dataset size. For increasing their performance, there is a high reliance on the manually annotated labels which is extremely expensive. To alleviate this problem, distance supervision has been employed where the labels are generated based on the emoticons and hashtags [258,310]. However, this approach did not perform well. This encouraged a fourth sub-category of semi-supervised techniques to rise. The semi-supervised techniques rely on both a small set of manually annotated data as well as unlabeled data to train the model. They can be further divided into three main types as depicted in Fig. 15: *graph-based*, *wrapper-based*, and *topic-based* techniques. The graph-based techniques [77,313,320,344] use label propagation to label the unlabeled training data based on the similarity metric between two nodes in the graph. Then, a classifier is trained and used as previous techniques. The wrapper-based techniques [44,45,214,215,380] rely either on self-training [44,45,380] or co-training [214,215]. In both types, the classification process is an iterative process, starting with the initial labeled data, classify the other unlabeled data, and use the high confident ones in the next iteration of the classification till all data is labeled or it hits the maximum number of iterations. The difference between the self-training and the co-training is that in self-training only one classifier is used, whereas in the co-training two classifiers with different feature sets are used to provide two different views for the data. The more confident classification within the two classifiers is chosen to be within the labeled data in the next iteration. The last semi-supervised types are topic-based techniques [11,123,139,166,241,301,355], where topic information is extracted with sentiment analysis simultaneously under the observation that the context of the content affects the sentiment.

The second major category in Fig. 15 is lexical techniques [30,146,152,207,260,278,299,323,340], where a predefined list of positive and negative words is employed to classify the sentiment of the new microblog. There are two main sub-categories in lexical techniques, namely *dictionary-based* and *corpus-based*. The dictionary-based techniques [30,146,152,207,299,340] use dictionaries as lexical resources and approximate lexical matching techniques are used to account for microblogs noise and abbreviations. The corpus-based technique [278] uses statistical or semantic methods to match incoming data with existing lexical resources. The third major category in Fig. 15 is hybrid techniques [107,115,175,177,182,189,376] that combine both machine learning and lexical methods to detect microblogs sentiment. These techniques use lexical terms either to train

a machine learning model or to filter data in a first stage that is fed to a classifier for further processing on a second stage.

Other miscellaneous techniques are proposed for microblogs sentiment analysis. **ConSent** [183] uses concept analysis to determine sentiment based on associated topic. **AppSent** [184] uses appraisal terms to outperform supervised techniques. **SocioSent** [150] uses sociological information in the supervised learning process to improve the performance. **ChineSentiment** [365] proposes a rule-based model for analyzing sentiment features of different linguistic components, and a corresponding methodology for calculating sentiment using emoticon elements as auxiliary affective factors.

## A.2 Semantic analysis

Semantic analysis is a popular analysis task that is widely used in microblogs literature for different applications, such as topic modeling [339], knowledge extraction [309], community detection [190,311], stance detection [390], sentiment analysis [182], event analysis [48,261], effective microblog retrieval and ranking [379], and user recommendations [106]. This task automates discovering the meanings of a chunk of text by discovering semantic relationships that relate to real-world entities, such as places, persons, and organizations. For example, a text like *Trump to campaign for Cindy Hyde-Smith in Mississippi* can be related to two persons, *Trump* and *Cindy Hyde-Smith*, and one place, *Mississippi*. This relatedness connects the input text to a predefined set of semantic concepts or categories that are commonly extracted from human-contributed content, such as Wikipedia, or professionally maintained ontologies, such as FOAF and DBpedia ontologies. Such type of analysis used to be performed on long chunks of text, e.g., news articles, blog posts, or web documents. However, in microblogs, the textual content is very short and contains a plenty of abbreviations, informality, and noisy terms. Such brevity hurts the performance of traditional semantic analysis techniques, as shown in [242], that depend on lexical matching and search-based retrieval in, for example, Wikipedia concepts.

To overcome the brevity problem, a general theme of semantic analysis research on microblogs is exploring different ways to enrich the microblogs short textual content to enable accurate semantic relations discovery. Existing techniques in the literature can be categorized into four categories, as depicted in Fig. 16, based on the source of enrichment through: *external documents-based techniques*, *machine learning-based techniques*, *hashtag-based techniques*, and *lexical techniques*. Techniques of the first category [57,113,339,350] depend on linking the microblog short document to external long documents, e.g., news articles or web documents, which allow traditional semantic analysis techniques to be applied with high precision. **ToSem** [339] performs semantic enrichment based on explicit web links that are included in the microblog to associate the linked web document. Then, it extracts both named entities and top-$k$ terms from the web document to be appended to the microblog as auxiliary terms. **NwSem** [57] identifies online news articles that are related to the microblog post in order to extract named entities and include them in the user profile as semantic tags. **UsrSem** [350] explores semantics of user interactions, specifically retweets and links that are embedded in tweets, and their role in inferring notions such as quality of user relationships, trust, and other attributes of user relationships. This could be applied to re-ranking microblogs based on importance, user interest, quality, etc. **DiSem** [113] maps microblog posts to Wikipedia articles, then use the Wikipedia ontology for semantic categorization.

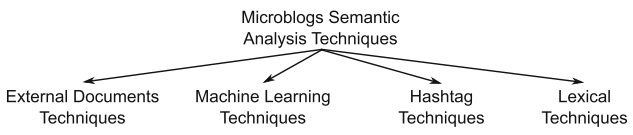



**Fig. 16** An overview of microblogs semantic analysis literature

The second category is machine learning-based techniques [110,149,216,217,242,287,311,370] that use either: (1) clustering to group different related microblogs and use their collective content to semantically label the whole cluster, or (2) classification that exploits annotated training data as an external source of information to learn different semantic classes of new microblogs. **TrSem** [370] introduces a novel transfer learning approach, namely transfer latent semantic learning, that utilizes a large number of tagged documents with rich information from domain-specific sources to discover latent semantics of the abbreviated text. **AccSem** [149] clusters related microblogs and use the collective content of each cluster to automatically assign semantically meaningful labels. The semantic labels are solicited from external knowledge sources, such as Wikipedia and WordNet, based on informative fragments parsed from microblogs contents. **AdSem** [242] uses SVM and naive Bayes classifiers to enhance the precision of mapping tweets to Wikipedia-based concepts. For this, it obtains an initial ranked list of candidate concepts through lexical matching, language modeling, and traditional techniques. Then, annotated training data is used to train classifiers that further classify microblogs based on different feature vectors to the correct semantic category, which significantly boosts both precision and recall. **NomSem** [216] uses SVM classifiers to identify nominal predicates in tweets. Then, a factor graph for each nominal predicate is constructed and joined with graphs of other predicates so their semantic arguments are jointly resolved. **ComSem** [311] clusters related microblogs to detect user groups within sub-communities. Then, a probabilistic model is employed to measure the

semantic, or topical, coherence of the user group and filter out non-coherent groups. **GeoSem** [314] clusters microblogs based on spatial, temporal, and semantic features, including LDA topics, to evaluate the performance of combining different features in retrieving insights from microblogs data. **ST-SRL** [217] proposes a semi-supervised self-training approach that utilizes a small training dataset to label unlabeled tweets in an iterative way to increase the training dataset size. Labeled data records with highest confidence from two different labelers are used to enhance the classification accuracy in the following iterations. **VecSem** [110] has performed a unique study that explores the effect of changing microblog-specific semantic representation features on the performance of semantic prediction. It studies a set of 13 microblog-specific prediction tasks to understand both textual and social aspects of different representations.

The third category is hashtag-based techniques [38,264,345]. Hashtags are user-defined tags included in microblog posts, which indicate the discussed topics and enable posts related to the same topics to be searched easily. These hashtags are used in different ways to discover latent semantic content in microblogs. **SMOB** [264] uses hashtags as seeds to generate potential related links to web documents and ontology entries from both FOAF and DBpedia ontologies. Then, relevant semantic relations to the discovered entities are appended to the microblog. **EntSem** [38] enriches semantics through retrieving a ranked list of the top-*k* hashtags that are relevant to a user's query and segments them into relevant individual words. Then, it retrieves a set of Wikipedia articles that are related to tweet text, hashtags, and segmented hashtags. **HGTM** [345] introduces a new topic model through using hashtags to determine semantic relatedness to each other through a graph structure. A graph of hashtag relatedness is constructed using probabilistic models; then, related hashtags are grouped in coherent topics.

The fourth category is lexical techniques [9,10,48,81,106,179,261,273,309,392] that improve traditional techniques that are used for long text to be effective for short textual microblog content. **InducSem** [273] induces semantic entities using lexical pattern-based approach that match microblog text with seed keywords of each semantic category, e.g., food, sports, or vehicles. **KnoSem** [309] uses lexical resources that include corpus and POS-tagged terms to label tweets with semantic frames for knowledge extraction purposes. **EveSem** [261] analyzes word co-occurrences to discover relationships among word pairs. Then, such features are used to calculate the pairwise similarity of tweets for event detection purposes. **HIVSem** [10] uses lexical matching techniques to analysis the presence of an HIV prevention drug on Twitter. **PlcSem** [179] extracts place semantics through LDA topic modeling from a collection of microblogs to abstract their content through probabilistic models into a set of coherent topics. Then, the extracted place semantics is analyzed for temporal changes, e.g., a sports arena could evolve over time to be a place for concerts and exhibitions. **MonSem** [48] uses lexical matching to match microblog content with semantic knowledge bases to monitor unexpected events on social media. **LikSem** [9] uses semantic user attributes to enhance link prediction among social media users. **RetSem** [392] uses lexical semantic features to enhance microblogs retrieval performance. **TriSem** [81] uses semantic relevance to filter tweets based on Wikipedia concepts and trigrams. **RecSem** [106] uses semantic relatedness to recommend users to follow. It links users to Wikipedia through lexical and disambiguation algorithms; then, similar users are recommended.

## References


1. Abdelhaq, H., Gertz, M., Armiti, A.: Efficient online extraction of keywords for localized events in Twitter. GeoInformatica **21**(2), 365–388 (2017)
2. Abdelhaq, H., Sengstock, C., Gertz, M.: EvenTweet: online localized event detection from Twitter. In: VLDB (2013)
3. Abdelsadek, Y., Chelghoum, K., Herrmann, F., Kacem, I.: Community extraction and visualization in social networks applied to Twitter. Inf. Sci. **424**, 204–223 (2018)
4. Abreu, J., Castro, I., Martínez, C., Oliva, S., Gutiérrez, Y.: UCSC-NLP at SemEval-2017 Task 4: sense n-grams for sentiment analysis in Twitter. In: SemEval-2017 (2017)
5. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R.: Sentiment analysis of Twitter data. In: LSM@ACL (2011)
6. Agarwal, M.K, Bansal, D., Garg, M., Ramamritham, K.: Keyword search on microblog data streams: finding contextual messages in real time. In: EDBT (2016)
7. Agarwal, M.K., Ramamritham, K., Bhide, M.: Real time discovery of dense clusters in highly dynamic graphs: identifying real world events in highly dynamic environments. PVLDB **5**(10), 980–991 (2012)
8. After Boston Explosions, People Rush to Twitter for Breaking News. http://www.latimes.com/business/technology/la-fi-tn-after-boston-explosions-people-rush-to-twitter-for-breaking-news-20130415,0,3729783.story (2013)
9. Ahmed, C., ElKorany, A.: Enhancing link prediction in Twitter using semantic user attributes. In: *ASONAM*, (2015)
10. Ahn, Z., McLaughlin, M., Hou, J., Nam, Y., Hu, C.W., Park, M., Meng, J.: Social network representation and dissemination of pre-exposure prophylaxis (PrEP): a semantic network analysis of HIV prevention drug on Twitter. In: Springer SCSM (2014)
11. Ahuja, A., Wei, W., Carley, K.M.: Microblog sentiment topic model. In: ICDM Workshops (2016)
12. Akbari, M., Xia, H., Nie, L., Chua, T.S: From tweets to wellness: wellness event detection from Twitter streams. In: AAAIz (2016)
13. Al-Olimat, H., Thirunarayan, K., Shalin, V.L., Sheth, A.P.: Location name extraction from targeted text streams using Gazetteer-based statistical language models. In: COLING (2018)
14. Alawad, N.A., Aris, A., Stefano, L., Ida, M., Fabrizio, S.: Network-aware recommendations of novel tweets. In: SIGIR (2016)
15. Alp, Z.Z., Ögüdücü, S.: Influential user detection on Twitter: analyzing effect of focus rate. In: ASONAM (2016)
16. Alsaedi, N., Burnap, P., Rana, O.: Can we predict a riot? Disruptive event detection using Twitter. ACM TOIT **17**(2), 18 (2017)

17. Alsaedi, N., Burnap, P., Rana, O.F.: Automatic summarization of real world events using Twitter. In: ICWSM (2016)
18. Alsubaiee, S., Altowim, Y., Altwaijry, H., Behm, A., Borkar, V.R., Bu, Y., Carey, M.J., Cetindil, I., Cheelangi, M., Faraaz, K., Gabrielova, E., Grover, R., Heilbron, Z., Kim, Y.S., Li, C., Ok, J.M., Onose, N., Pirzadeh, P., Tsotras, V., Vernica, R., Wen, J., Westmann, T.: AsterixDB: a scalable, open source BDMS. PVLDB **7**(14), 1905–1916 (2014)
19. Apache AsterixDB. http://asterixdb.apache.org/ (2018)
20. Apache Cassandra. http://cassandra.apache.org/ (2018)
21. Apache Flink. https://flink.apache.org/ (2018)
22. Apache Ignite. https://ignite.apache.org/ (2018)
23. Apache Impala. https://impala.apache.org/ (2018)
24. Apache Spark. https://spark.apache.org/ (2014)
25. Apache Spark Streaming. https://spark.apache.org/streaming/ (2018)
26. Apache Storm. https://storm.apache.org/ (2014)
27. Apple buys social media analytics firm Topsy Labs. www.bbc.co.uk/news/business-25195534 (2013)
28. A Nobel Peace Prize for Twitter? www.csmonitor.com/Commentary/Opinion/2009/0706/p09s02-coop.html (2009)
29. Ardon, S., Bagchi, A., Mahanti, A., Ruhela, A., Seth, A., Tripathy, R.M., Triukose, S.: Spatio-temporal and events based analysis of topic popularity in Twitter. In: CIKM (2013)
30. Arslan, Y., Birturk, A., Djumabaev, B., Küçük, D.: Real-time Lexicon-based sentiment analysis experiments on Twitter with a mild (more information, less data) approach. In: IEEE Big Data (2017)
31. Asiaee, A., Tepper, M., Banerjee, A., Sapiro, G.: If you are happy and you know it... Tweet. In: CIKM (2012)
32. Avudaiappan, N., Herzog, A., Kadam, S., Du, Y., Thatche, J., Safro, I.: Detecting and summarizing emergent events in microblogs and social media streams by dynamic centralities. In: IEEE Big Data (2017)
33. Babcock, B., Datar, M., Motwani, R.: Load shedding for aggregation queries over data streams. In: ICDE (2004)
34. Bai, S., Hao, B., Li, A., Yuan, S., Gao, R., Zhu, T.: Predicting big five personality traits of microblog users. In: WI (2013)
35. Bakliwal, A., Arora, P., Madhappan, S., Kapre, N., Singh, M., Varma, V.: Mining sentiments from tweets. In: WASSA@ACL (2012)
36. Balikas, G.: TwiSe at SemEval-2017 Task 4: five-point Twitter sentiment classification and quantification. In: SemEval-2017 (2017)
37. Balikas, G., Moura, S., Amini, M.R.: Multitask learning for fine-grained Twitter sentiment analysis. In: SIGIR (2017)
38. Bansal, P., Jain, S., Varma, V.: Towards semantic retrieval of hashtags in microblogs. In: WWW Companion (2015)
39. Barbosa, L., Feng, J.: Robust sentiment detection on Twitter from biased and noisy data. In: COLING (2010)
40. Bartoletti, M., Lande, S., Massa, A.: Faderank: an incremental algorithm for ranking Twitter users. In: WISE (2016)
41. Basu, M., Ghosh, K., Das, S., Dey, R., Bandyopadhyay, S., Ghosh, S.: Identifying post-disaster resource needs and availabilities from microblogs. In: ASONAM (2017)
42. Basu, M., Shandilya, A., Ghosh, K., Ghosh, S.: Automatic matching of resource needs and availabilities in microblogs for post-disaster relief. In: WWW Companion (2018)
43. Battle, L., Chang, R., Stonebraker, M.: Dynamic prefetching of data tiles for interactive visualization. In: SIGMOD (2016)
44. Baugh, W.: Bwbaugh: hierarchical sentiment analysis with partial self-training. In: SemEval, vol. 2 (2013)
45. Becker, L., Erhart, G., Skiba, D., Matula, V.: Avaya: sentiment analysis on twitter with self-training and polarity lexicon expansion. In: SemEval, vol. 2 (2013)
46. Bermingham, A., Smeaton, A.F.: Classifying sentiment in microblogs: Is brevity an advantage? In: CIKM (2010)
47. Bian, J., Yang, Y., Chua, T.S.: Multimedia summarization for trending topics in microblogs. In: CIKM (2013)
48. Bisio, F., Meda, C., Zunino, R., Surlinelli, R., Scillia, E., Ottaviano, A.: Real-time monitoring of Twitter traffic by using semantic networks. In: ASONAM (2015)
49. Bizid, I., Nayef, N., Boursier, N., Faïz, S., Doucet, A.: Identification of microblogs prominent users during events by learning temporal sequences of features. In: CIKM (2015)
50. Budak, C., Georgiou, T., Agrawal, D., Abbadi, A.E.: GeoScope: online detection of geo-correlated information trends in social networks. In: VLDB (2014)
51. Busch, M., Gade, K., Larson, B., Lok, P., Luckenbill, S., Lin, J.: Earlybird: real-time search at Twitter. In: ICDE (2012)
52. Cai, H., Huang, Z., Srivastava, D., Zhang, Q.: Indexing evolving events from tweet streams. TKDE **27**(11), 3001–3015 (2015)
53. Cao, C.C., She, J., Tong, Y., Chen, L.: Whom to ask? Jury selection for decision making tasks on micro-blog services. PVLDB **5**(11), 1495–1506 (2012)
54. Cao, X., Cong, G., Guo, T., Jensen, C.S., Ooi, B.C.: Efficient processing of spatial group keyword queries. TODS **40**(2), 13 (2015)
55. Cao, X., Cong, G., Jensen, C.S., Ooi, B.C.: Collective spatial keyword querying. In: SIGMOD (2011)
56. Cary, A., Wolfson, O., Rishe, N.: Efficient and scalable method for processing top-k spatial boolean queries. In: SSDBM (2010)
57. Celik, I., Abel, F., Houben, G.J.: Learning semantic relationships between entities in Twitter. In: ICWE (2011)
58. Chandrasekaran, S., Cooper, S., Deshpande, A., Franklin, M.J., Hellerstein, J.M., Hong, J.M., Krishnamurthy, S., Madden, S., Reiss, F., Shah, M.A.: TelegraphCQ: continuous dataflow processing. In: SIGMOD (2003)
59. Chavan, H., Mokbel, M.F.: Scout: a GPU-aware system for interactive spatio-temporal data visualization. In: SIGMOD (2017)
60. Chen, C., Li, F., Ooi, B.C., Wu, S.: TI: an efficient indexing mechanism for real-time search on tweets. In: SIGMOD (2011)
61. Chen, C.C., Huang, H.H., Chen, H.H.: NLG301 at SemEval-2017 Task 5: fine-grained sentiment analysis on financial microblogs and news. In: SemEval (2017)
62. Chen, F., Ji, R., Jinsong, S., Cao, D., Gao, Y.: Predicting microblog sentiments via weakly supervised multimodal deep learning. IEEE Trans. Multimed. **20**(4), 997–1007 (2018)
63. Chen, L., Cong, G., Cao, X.: An efficient query indexing mechanism for filtering geo-textual data. In: SIGMOD (2013)
64. Chen, L., Cong, G., Jensen, C.S., Wu, D.: Spatial keyword query processing: an experimental evaluation. In: VLDB (2013)
65. Chen, L., Cui, Y., Cong, G., Cao, X.: SOPS: a system for efficient processing of spatial-keyword publish/subscribe. PVLDB **7**(13), 1601–1604 (2014)
66. Chen, X., Li, L., Guandong, X., Yang, Z., Kitsuregawa, M.: Recommending related microblogs: a comparison between topic and WordNet based approaches. In: AAAI (2012)
67. Chen, X., Sykora, M.D., Jackson, T.W., Elayan, S.: What about mood swings: identifying depression on Twitter with temporal measures of emotions. In: WWW Companion (2018)
68. Cheng, D., Schretlen, P., Kronenfeld, N., Bozowsky, N., Wright, W.: Tile based visual analytics for Twitter big data exploratory analysis. In: IEEE Big Data (2013)
69. Christoforaki, M., He, J., Dimopoulos, C., Markowetz, A., Suel, T.: Text versus space: efficient geo-search query processing. In: CIKM (2011)
70. Clark, S., Wicentwoski, R.: SwatCS: combining simple classifiers with estimated accuracy. In: SemEval@NAACL-HLT (2013)
71. Cliche, M.: BB_twtr at SemEval-2017 Task 4: Twitter sentiment analysis with CNNs and LSTMs. arXiv:1704.06125 (2017)

72. Cong, G., Jensen, C.S.: Querying geo-textual data: spatial keyword queries and beyond. In: SIGMOD (2016)
73. Cong, G., Jensen, C.S., Dingming, W.: Efficient retrieval of the top-k most relevant spatial web objects. PVLDB **2**(1), 337–348 (2009)
74. Constantin, C., Grossetti, Q., Mouza, Cé., Travers, N.: An homophily-based approach for fast post recommendation in microblogging systems. In: EDBT (2018)
75. Corrêa Jr. E.A., Marinho, V.Q., dos Santos, L.B.: Nilc-usp at SemEval-2017 Task 4: a multi-view ensemble for twitter sentiment analysis. arXiv:1704.02263 (2017)
76. Counts, S., Fisher, K.: Taking it all in?. Visual attention in microblog consumption. In: ICWSM (2011)
77. Cui, A., Zhang, M., Liu, Y., Ma, S.: Emotion tokens: bridging the gap among multilingual Twitter sentiment analysis. In: Asia Information Retrieval Symposium (2011)
78. Cui, A., Zhang, M., Liu, Y., Ma, S., Zhang, K.: Discover breaking events with popular Hashtags in Twitter. In: CIKM (2012)
79. da Silva, N.F.F., Hruschka, E.R., Hruschka Jr., E.R.: Tweet sentiment analysis with classifier ensembles. DSS J. **66**, 170–179 (2014)
80. da Silva, N.F.F., Coletta, L.F.S., Hruschka, E.R.: A survey and comparative study of tweet sentiment analysis via semi-supervised learning. ACM Comput. Surv. **49**(1), 15:1–15:26 (2016)
81. Dang, A., Makki, R., Moh'd, A., Islam, A., Keselj, V., Milios, E.E.: Real time filtering of tweets using Wikipedia concepts and google tri-gram semantic relatedness. In: TREC (2015)
82. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using Twitter Hashtags and Smileys. In: COLING (2010)
83. de França Costa, D., da Silva, N.F.F.: INF-UFG at FiQA 2018 Task 1: predicting sentiments and aspects on financial tweets and news headlines. In: WWW Companion (2018)
84. de Macedo, A.Q., Marinho, L.B., Santos, R.L.T.: Context-aware event recommendation in event-based social networks In: RecSys (2015)
85. DeBrabant, J., Pavlo, A., Tu, S., Stonebraker, M., Zdonik, S.B.: Anti-caching: a new approach to database management system architecture. In: VLDB (2013)
86. Deshmane, A.A., Friedrichs, J.: TSA-INF at SemEval-2017 Task 4: an ensemble of deep learning architectures including lexicon features for Twitter sentiment analysis. In: SemEval-2017 (2017)
87. Dey, K., Shrivastava, R., Kaushik, S.: Twitter stance detection—a subjectivity and sentiment polarity inspired two-phase approach. In: ICDM Workshops (2017)
88. Dey, K., Shrivastava, R., Kaushik, S., Subramaniam, L.V.: EmTaggeR: a word embedding based novel method for hashtag recommendation on Twitter. In: ICDM Workshops (2017)
89. Ding, J., Dong, Y., Gao, T., Zhang, Z., Liu, Y.: Sentiment analysis of chinese micro-blog based on classification and rich features. In: Web Information Systems and Applications Conference (2016)
90. Dong, L., Wei, F., Tan, C., Tang, D., Zhou, M., Xu, K.: Adaptive recursive neural network for target-dependent Twitter sentiment classification. In: ACL (2014)
91. Doulamis, N.D., Doulamis, A.D., Kokkinos, P.C., Varvarigos, E.M.: Event detection in Twitter microblogging. IEEE Trans. Cybern. **46**(12), 2810–2824 (2016)
92. Dovdon, E., Saias, J.: ej-sa-2017 at SemEval-2017 Task 4: experiments for target oriented sentiment analysis in Twitter. In: SemEval@ACL (2017)
93. Drescher, C., Wallner, G., Kriglstein, S., Sifa, R., Drachen, A., Pohl, M.: What moves players? Visual data exploration of Twitter and Gameplay data. In: CHI (2018)
94. Duong-Trung, N., Schilling, N., Schmidt-Thieme, L.: Near real-time geolocation prediction in Twitter streams via matrix factorization based regression. In: CIKM (2016)
95. Dutt, R., Hiware, K., Ghosh, A., Bhaskaran, R.: SAVITR: a system for real-time location extraction from microblogs during emergencies. In: CoRR. arXiv:1801.07757 (2018)
96. Dutta, S., Chandra, V., Mehra, K., Das, A.K., Chakraborty, T., Ghosh, S.: Ensemble algorithms for microblog summarization. IEEE Intell. Syst. **33**(3), 4–14 (2018)
97. Effelsberg, W., Härder, T.: Principles of database buffer management. TODS **9**(4), 560–595 (1984)
98. Efstathiades, C., Antoniou, H., Skoutas, D., Vassiliou, Y.: TwitterViz: visualizing and exploring the Twitter sphere. In: SSTD (2015)
99. Ehsan, H., Sharaf, M.A., Chrysanthis, P.K.: MuVE: efficient multi-objective view recommendation for visual data exploration. In: ICDE (2016)
100. Eldawy, A., Mokbel, M.F., Jonathan, C.: HadoopViz: a MapReduce framework for extensible visualization of big spatial data. In: ICDE (2016)
101. Embrace of Social Media Aids Flood Victims in Kashmir. https://www.nytimes.com/2014/09/13/world/asia/embrace-of-social-media-aids-flood-victims-in-kashmir.html (2014)
102. Enoki, M., Ikawa, Y., Raymond, R.: User community reconstruction using sampled microblogging data. In: WWW Companion (2012)
103. Erdoğan, A.E., Yilmaz, T., Sert, O.C., Akyüz, M., Özyer, T., Alhajj, R.: From social media analysis to ubiquitous event monitoring: the case of Turkish tweets. In: ASONAM (2017)
104. Facebook Statistics. http://newsroom.fb.com/company-info/ (2018)
105. Fagin, R., Lotem, A., Naor, M.: Optimal aggregation algorithms for middleware. In: PODS (2001)
106. Faralli, S., Tommaso, G. Di Velardi, P.: Semantic enabled recommender system for micro-blog users. In: ICDM (2016)
107. Feng, S., Song, K., Wang, D., Ge, Y.: A word-emoticon mutual reinforcement ranking model for building sentiment lexicon from massive collection of microblogs. WWW J. **18**(4), 949–967 (2015)
108. Feng, W., Zhang, C., Zhang, W., Han, J., Wang, J., Aggarwal, C., Huang, J.: STREAMCUBE: hierarchical spatio-temporal hashtag clustering for event exploration over the Twitter stream. In: ICDE (2015)
109. Forsati, R., Mahdavi, M., Shamsfard, M., Sarwat, M.: Matrix factorization with explicit trust and distrust side information for improved social recommendation. ACM Trans. Inf. Syst. **32**(4), 17:1–17:38 (2014)
110. Ganesh, J., Gupta, M., Varma, V.: Interpretation of semantic tweet representations. In: ASONAM (2017)
111. Gao, L., Wang, Y., Li, D., Shao, J., Song, J.: Real-time social media retrieval with spatial, temporal and social constraints. Neurocomputing **253**, 77–88 (2017)
112. Gedik, B., Wu, K.L., Yu, P.S., Liu, L.: A load shedding framework and optimizations for M-way windowed stream joins. In: ICDE (2007)
113. Genc, Y., Sakamoto, Y., Nickerson, J.V.: Discovering context: classifying tweets through a semantic transform based on Wikipedia. In: Springer FAC (2011)
114. Ghanem, T., Magdy, A., Musleh, M., Ghani, S., Mokbel, M.: VisCAT: spatio-temporal visualization and aggregation of categorical attributes in Twitter data. In: SIGSPATIAL (2014)
115. Ghiassi, M., Skinner, J., Zimbra, D.: Twitter brand sentiment analysis: a hybrid system using n-gram analysis and dynamic artificial neural network. Expert Syst. Appl. **40**(16), 6266–6282 (2013)
116. Ghosh, S., Sharma, N.K., Benevenuto, F., Ganguly, N., Gummadi, P.K.: Cognos: Crowdsourcing search for topic experts in microblogs. In: SIGIR (2012)

117. Giachanou, A., Crestani, F.: Like it or not: a survey of Twitter sentiment analysis methods. ACM Comput. Surv. **49**(2), 28:1–28:41 (2016)
118. Gilani, Z., Kochmar, E., Crowcroft, J.: Classification of Twitter accounts into automated agents and human users. In: ASONAM (2017)
119. Gillani, M., Ilyas, M.U., Saleh, S., Alowibdi, J.S., Aljohani, N.R., Alotaibi, F.S.: Post summarization of microblogs of sporting events. In: WWW Companion (2017)
120. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. Technical report, Stanford University (2009)
121. Grover, R., Carey, M.: Data ingestion in AsterixDB. In: EDBT (2015)
122. Gu, Y., Song, J., Liu, W., Zou, L., Yao, Y.: Context aware matrix factorization for event recommendation in event-based social networks. In: WI (2016)
123. Guha, S., Chakraborty, T., Datta, S., Kumar, M., Varma, V.: TweetGrep: weakly supervised joint retrieval and sentiment analysis of topical tweets. In: ICWSM (2016)
124. Guilherme, C.R., de Lemos, V.S., Lammel, F., Manssour, I.H., Silveira, M.S., Pase, A.F.: Visualization techniques for the analysis of Twitter users' behavior. In: ICWSM (2013)
125. Guo, L., Zhang, D., Li, G., Tan, K.L., Bao, Z.: Location-aware pub/sub system: when continuous moving queries meet dynamic event streams. In: SIGMOD (2015)
126. Guo, L., Zhang, D., Wang, Y., Huayu, W., Cui, B., Tan, K.-L.: Co2: Inferring personal interests from raw footprints by connecting the offline world with the online world. ACM Trans. Inf. Syst. (TOIS) **36**(3), 31 (2018)
127. Guo, T., Cao, X., Cong, G.: Efficient algorithms for answering the M-closest keywords query. In: SIGMOD (2015)
128. Guo, T., Feng, K., Cong, G., Bao, Z.: Efficient selection of geospatial data on maps for interactive and visualized exploration. In: SIGMOD (2018)
129. Gupta, P., Goel, A., Lin, J.J., Sharma, A., Wang, D., Zadeh, R.: WTF: the who to follow service at Twitter. In: WWW (2013)
130. Gupta, P., Satuluri, V., Grewal, A., Gurumurthy, S., Zhabiuk, V., Li, Q., Lin, J.J.: Real-time Twitter recommendation: online Motif detection in large dynamic graphs. PVLDB **7**(13), 1379–1380 (2014)
131. Hamdan, H., Béchet, F., Bellot, P.: Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In: SemEval@NAACL-HLT (2013)
132. Hannon, J., Bennett, M., Smyth, B.: Recommending Twitter users to follow using content and collaborative filtering approaches. In: RecSys (2010)
133. Hansu, G., Gartrell, M., Zhang, L., Lv, Q., Grunwald, D.: AnchorMF: towards effective event context identification. In: CIKM (2013)
134. Hao, Y., Lan, Y., Li, Y., Li, C.: XJSA at SemEval-2017 Task 4: a deep system for sentiment classification in Twitter. In: SemEval-2017 (2017)
135. Harvard Medical School Researchers Awarded Twitter Data Grant. https://hms.harvard.edu/news/harvard-medical-school-researchers-awarded-twitter-data-grant (2014)
136. Hassan, A., Abbasi, A., Zeng, D.: Twitter sentiment analysis: a bootstrap ensemble framework. In: SocialCom (2013)
137. He, L., Luo, J.: What makes a pro eating disorder Hashtag: using Hashtags to identify pro eating disorder Tumblr posts and Twitter users. In: IEEE Big Data (2016)
138. He, Y., Barman, S., Naughton, J.F.: On load shedding in complex event processing. In: ICDT (2014)
139. He, Y., Lin, C., Gao, W., Wong, K.F.: Tracking sentiment and topic dynamics from social media. In: ICWSM (2012)
140. Health Department Use of Social Media to Identify Foodborne Illness—Chicago, Illinois, 2013–2014. https://www.cdc.gov/mmwr/preview/mmwrhtml/mm6332a1.htm (2014)
141. Hecht, B.J., Hong, L., Suh, B., Chi, E.H.: Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles. In: CHI (2011)
142. Hoang, T., Cher, P.H., Prasetyo, P.K., Lim, E.P.: Big data: crowdsensing and analyzing micro-event tweets for public transportation insights. In: IEEE (2016)
143. Hong, L., Ahmed, A., Gurumurthy, S., Smola, A.J., Tsioutsiouliklis, K.: Discovering geographical topics in the Twitter stream. In: WWW (2012)
144. How Facebook Is Transforming Disaster Response. https://www.wired.com/2016/11/facebook-disaster-response/ (2016)
145. How Twitter, Facebook, WhatsApp And Other Social Networks Are Saving Lives During Disasters. http://www.huffingtonpost.in/2017/01/31/how-twitter-facebook-whatsapp-and-other-social-networks-are-sa_a_21703026/ (2017)
146. Htait, A., Fournier, S., Bellot, P.: LSIS at SemEval-2017 Task 4: using adapted sentiment similarity seed words for English and Arabic tweet polarity classification. In: SemEval (2017)
147. Hu, G., Bhargava, P., Fuhrmann, S., Ellinger, S., Spasojevic, N.: Analyzing users' sentiment towards popular consumer industries and brands on Twitter. arXiv:1709.07434 (2017)
148. Hu, Q., Pei, Y., Chen, Q., He, L.: SG++: Word representation with sentiment and negation for Twitter sentiment classification. In: SIGIR (2016)
149. Hu, X., Tang, L., Liu, H.: Enhancing accessibility of microblogging messages using semantic knowledge. In: CIKM (2011)
150. Hu, X., Tang, L., Tang, J., Liu, H.: Exploiting social relations for sentiment analysis in microblogging. In: WSDM (2013)
151. Hu, Y., John, A., Wang, F., Kambhampati, S.: ET-LDA: joint topic modeling for aligning events and their Twitter feedback. In: AAAI, vol. 12 (2012)
152. Hu, Y., Nian, T., Chen, C.: Mood congruence or mood consistency? examining aggregated Twitter sentiment towards Ads in 2016 super bowl. In: ICWSM (2017)
153. Hua, T., Chen, F., Zhao, L., Chang-Tien, L., Ramakrishnan, N.: STED: semi-supervised targeted-interest event detectionin in Twitter. In: SIGKDD (2013)
154. Hua, T., Chen, F., Zhao, L., Lu, C.-T., Ramakrishnan, N.: Automatic targeted-domain spatio-temporal event detection in Twitter. GeoInformatica **20**(4), 765–795 (2016)
155. Hubert, R.B., Estevez, E., Maguitman, A.G., Janowski, T.: Examining government-citizen interactions on Twitter using visual and sentiment analysis. In: DG.O (2018)
156. Hurricane Harvey Victims Turn to Twitter and Facebook. http://time.com/4921961/hurricane-harvey-twitter-facebook-social-media/ (2017)
157. In Irma, Emergency Responders' New Tools: Twitter and Facebook. https://www.wsj.com/articles/for-hurricane-irma-information-officials-post-on-social-media-1505149661 (2017)
158. Ikawa, Y., Enoki, M., Tatsubori, M.: Location inference using microblog messages. In: WWW (2012)
159. Itoh, M., Yokoyama, D., Toyoda, M., Tomita, Y., Kawamura, S., Kitsuregawa, M.: Visual exploration of changes in passenger flows and tweets on mega-city metro network. IEEE Trans. Big Data **2**(1), 85–99 (2016)
160. Jabreel, M., Moreno, A.: SiTAKA at SemEval-2017 Task 4: sentiment analysis in twitter based on a rich set of features. In: SemEval (2017)
161. Japan earthquake: how Twitter and Facebook helped. http://www.telegraph.co.uk/technology/twitter/8379101/Japan-earthquake-how-Twitter-and-Facebook-helped.html (2011)

162. Jia, J., Li, C., Zhang, X., Li, C., Carey, M.J., Su, S.: Towards interactive analytics and visualization on one billion tweets. In: SIGSPATIAL (2016)
163. Jiang, J., Lu, H., Yang, B., Cui, B.: Finding top-k local users in geo-tagged social media data. In: ICDE (2015)
164. Jiang, L., Yu, M., Zhou, M., Liu, X., Zhao, T.: Target-dependent Twitter sentiment classification. In: ACL (2011)
165. Jianqiang, Z., Xiaolin, G., Xuejun, Z.: Deep convolution neural networks for Twitter sentiment analysis. IEEE Access **6**, 23253–23260 (2018)
166. Jo, Y., Oh, A.H: Aspect and sentiment unification model for online review analysis. In: WSDM (2011)
167. Jonathan, C., Magdy, A., Mokbel, M.F., Jonathan, A.: GARNET: a holistic system approach for trending queries in microblogs. In: ICDE (2016)
168. Jones, A.J., Carlson, E.: TwitterViz: a robotics system for remote data visualization. In: ICWSM (2013)
169. Kallman, R., Kimura, H., Natkins, J., Pavlo, A., Rasin, A., Zdonik, S.B., Jones, E.P.C., Madden, S., Stonebraker, M., Zhang, Y., Hugg, J., Abadi, D.J.: H-store: a high-performance, distributed main memory transaction processing system. PVLDB **1**(2), 1496–1499 (2008)
170. Kalyanam, J., Velupillai, S., Conway, M., Lanckriet, G.: From event detection to storytelling on microblogs. In: ASONAM (2016)
171. Kaneko, T., Yanai, K.: Visual event mining from the Twitter stream. In: WWW Companion (2016)
172. Karanasou, M., Ampla, A., Doulkeridis, C., Halkidi, M.: Scalable and real-time sentiment analysis of Twitter data. In: ICDM Workshops (2016)
173. Kazai, G., Iskander, Y., Daoud, C.: Personalised news and blog recommendations based on user location, Facebook and Twitter user profiling. In: SIGIR (2016)
174. Kempter, R., Sintsova, V., Musat, C.C., Pu, P.: EmotionWatch: visualizing fine-grained emotions in event-related tweets. In: ICWSM (2014)
175. Khan, F.H., Bashir, S., Qamar, U.: TOM: Twitter opinion mining framework using hybrid classification scheme. DSS J. **57**, 245–257 (2014)
176. Khatua, A., Khatua, A.: Cricket World Cup 2015: predicting user's orientation through mix tweets on twitter platform. In: ASONAM (2017)
177. Khuc, V.N., Shivade, C., Ramnath, R., Ramanathan, J.: SAC: towards building large-scale distributed systems for Twitter sentiment analysis. In: ACM (2012)
178. Kim, A., Blais, E., Parameswaran, A.G., Indyk, P., Madden, S., Rubinfeld, R.: Rapid sampling for visualizations with ordering guarantees. PVLDB **8**(5), 521–532 (2015)
179. Kim, E., Ihm, H., Myaeng, S.H.: Topic-based place semantics discovered from microblogging text messages. In: WWW Companion (2014)
180. Kiritchenko, S., Zhu, X., Mohammad, S.M.: Sentiment analysis of short informal texts. JAIR **50**, 723–762 (2014)
181. Kitazawa, T., Yui, M.: Query-based simple and scalable recommender systems with Apache Hivemall. In: RecSys (2018)
182. Kolovou, A., Kokkinos, F., Fergadis, A., Papalampidi, P., Iosif, E., Malandrakis, N., Palogiannidi, E., Papageorgiou, H., Narayanan, S., Potamianos, A.: Tweester at SemEval-2017 Task 4: fusion of semantic-affective and pairwise classification models for sentiment analysis in Twitter. In: SemEval@ACL (2017)
183. Kontopoulos, E., Berberidis, C., Dergiades, T., Bassiliades, N.: Ontology-based sentiment analysis of Twitter posts. Expert Syst. Appl. **40**(10), 4065–4074 (2013)
184. Korenek, P., Simko, M.: Sentiment analysis on microblog utilizing appraisal theory. WWW J. **17**(4), 847–867 (2014)
185. Kouloumpis, E., Wilson, T., Moore, J.D.: Twitter sentiment analysis: the good the bad and the OMG! In: ICWSM (2011)
186. Kowald, D., Pujari, S.C., Lex, E.: Temporal effects on hashtag reuse in twitter: a cognitive-inspired hashtag recommendation approach. In: WWW (2017)
187. Krumm, J., Horvitz, E.: Eyewitness: identifying local events via space-time signals in Twitter feeds. In: SIGSPATIAL (2015)
188. Kumamoto, T., Suzuki, T., Wada, H.: Visualizing impression-based preferences of Twitter users. In: SCSM-HCI (2014)
189. Kumar, A., Sebastian, T.M.: Sentiment analysis on Twitter. IJCSI **9**(4), 372 (2012)
190. Kuramochi, T., Okada, N., Tanikawa, K., Hijikata, Y., Nishida, S.: Applying to Twitter networks of a community extraction method using intersection graph and semantic analysis. In: Springer HCI (2013)
191. Lacic, E.: Real-time recommendations in a multi-domain environment. In: ACM HT (2016)
192. Lacic, E., Kowald, D., Parra, D., Kahr, M., Trattner, C.: Towards a scalable social recommender engine for online marketplaces: the case of apache solr. In: WWW Companion (2014)
193. Lahoti, P., De Francisci Morales, G., Gionis, A.: Finding topical experts in twitter via query-dependent personalized PageRank. In: ASONAM (2017)
194. Laskari, N.K., Sanampudi, S.K.: TWINA at SemEval-2017 Task 4: Twitter sentiment analysis with ensemble gradient boost tree classifier. In: SemEval-2017 (2017)
195. Lee, G., Lin, J., Liu, C., Lorek, A., Ryaboy, D.V.: The unified logging infrastructure for data analytics at Twitter. PVLDB **5**(12), 1771–1780 (2012)
196. Lee, T., Park, J.W., Lee, S., Hwang, S.W., Elnikety, S., He, Y.: Processing and optimizing main memory spatial-keyword queries. PVLDB **9**(3), 132–143 (2015)
197. Levandoski, J., Larson, P., Stoica, R.: Identifying hot and cold data in main-memory databases. In: ICDE (2013)
198. Levandoski, J.J., Sarwat, M., Mokbel, M.F., Ekstrand, M.D.: RecStore: an extensible and adaptive framework for online recommender queries inside the database engine. In: EDBT (2012)
199. Li, G., Hu, J., Feng, J., Tan, K.L.: Effective location identification from microblogs. In: ICDE (2014)
200. Li, G., Wang, Y., Wang, T., Feng, J.: Location-aware publish/subscribe. In: KDD (2013)
201. Li, J., Liao, M., Gao, W., He, Y., Wong, K.F.: Topic extraction from microblog posts using conversation structures. In: ACL (2016)
202. Li, Q., Shah, S., Nourbakhsh, A., Fang, R., Liu, X.: funSentiment at SemEval-2017 Task 5: fine-grained sentiment analysis on financial microblogs using word vectors built from StockTwits and Twitter. In: SemEval (2017)
203. Li, R., Lei, K.H., Khadiwala, R., Chang, K.C.C.: TEDAS: a Twitter-based event detection and analysis system. In: ICDE (2012)
204. Li, Y., Jiang, J., Liu, T., Qiu, M., Sun, X.: Personalized microtopic recommendation on microblogs. ACM TIST **8**(6), 77 (2017)
205. Li, Y., Bao, Z., Li, G., Tan, K.L.: Real time personalized search on social networks. In: ICDE (2015)
206. Li, Z., Lee, K.C.K., Zheng, B., Lee, W.-C., Lee, D.L., Wang, X.: IR-Tree: an efficient index for geographic document search. TKDE **23**(4), 585–599 (2011)
207. Lim, K.H., Lee, K.E., Kendal, D., Rashidi, L., Naghizade, E., Winter, S., Vasardani, M.: The grass is greener on the other side: understanding the effects of green spaces on Twitter user sentiments. In: WWW Companion (2018)
208. Lin, J., Kolcz, A.: Large-scale machine learning at Twitter. In: SIGMOD (2012)
209. Lin, J., Mishne, G.: A study of "Churn" in tweets and real-time search queries. In: ICWSM (2012)

210. Lingad, J., Karimi, S., Yin, J.: Location extraction from disaster-related microblogs. In: WWW (2013)
211. Lingkun, W., Lin, W., Xiao, X., Xu, Y.: LSII: An indexing structure for exact real-time search on microblogs. In: ICDE (2013)
212. Liu, M., Fu, K., Lu, C.T., Chen, G., Wang, H.: A search and summary application for traffic events detection based on Twitter data. In: SIGSPATIAL (2014)
213. Liu, N., Li, L., Guandong, X., Yang, Z.: Identifying domain-dependent influential microblog users: a post-feature based approach. In: AAAI (2014)
214. Liu, S., Li, F., Li, F., Cheng, X., Shen, H.: Adaptive co-training SVM for sentiment classification on tweets. In: CIKM (2013)
215. Liu, S., Zhu, W., Xu, N., Li, F., Cheng, X.Q., Liu, Y., Wang, Y.: Co-training and visualizing sentiment evolvement for tweet events. In: WWW (2013)
216. Liu, X., Fu, Z., Wei, F., Zhou, M.: Collective nominal semantic role labeling for tweets. In: AAAI (2012)
217. Liu, X., Li, K., Zhou, M., Xiong, Z.: Enhancing semantic role labeling for tweets using self-training. In: AAAI (2011)
218. Liu, X., Li, Q., Nourbakhsh, A., Fang, R., Thomas, M., Anderson, K., Kociuba, R., Vedder, M., Pomerville, S., Wudali, R., et al.: Reuters tracer: a large scale system of detecting & verifying real-time news events from Twitter. In: CIKM (2016)
219. Long, C., Wong, R.C.W., Wang, K., Fu, A.W.C.: Collective spatial keyword queries: a distance owner-driven approach. In: SIGMOD (2013)
220. Lozić, D., Šarić, D., Tokić, I., Medić, Z., Šnajder, J.: TakeLab at SemEval-2017 Task 4: recent deaths and the power of nostalgia in sentiment analysis in Twitter. In: SemEval-2017 (2017)
221. Lu, X., Li, P., Ma, H., Wang, S., Xu, A., Wang, B.: Computing and applying topic-level user interactions in microblog recommendation. In: SIGIR (2014)
222. Ma, R., Zhang, Q., Wang, J., Cui, L., Huang, X.: Mention recommendation for multimodal microblog with cross-attention memory network. In: SIGIR (2018)
223. Magdy, A., Alarabi, L., Al-Harthi, S., Musleh, M., Ghanem, T., Ghani, S., Mokbel, M.: Taghreed: a system for querying, analyzing, and visualizing geotagged microblogs. In: SIGSPATIAL (2014)
224. Magdy, A., Alghamdi, R., Mokbel, M.F.: On main-memory flushing in microblogs data management systems. In: ICDE (2016)
225. Magdy, A., Aly, A.M., Mokbel, M.F., Elnikety, S., He, Y., Nath, S., Aref, W.G.: GeoTrend: spatial trending queries on real-time microblogs. In: SIGSPATIAL (2016)
226. Magdy, A., Mokbel, M.: Towards a microblogs data management system. In: MDM (2015)
227. Magdy, A., Mokbel, M.: Microblogs data management and analysis (tutorial). In: ICDE (2016)
228. Magdy, A., Mokbel, M.: Demonstration of kite: a scalable system for microblogs data management. In: ICDE (2017)
229. Magdy, A., Mokbel, M.F., Elnikety, S., Nath, S., He, Y.: Mercury: a memory-constrained spatio-temporal real-time search on microblogs. In: ICDE (2014)
230. Magdy, A., Mokbel, M.F., Elnikety, S., Nath, S., He, Y.: Venus: scalable real-time spatial queries on microblogs with adaptive load shedding. TKDE **28**(2), 356–370 (2016)
231. Magdy, A., Musleh, M., Tarek, K., Alarabi, L., Al-Harthi, S., Elmongui, H.G., Ghanem, T.M., Ghani, S., Mokbel, M.F.: Taqreer: a system for spatio-temporal analysis on microblogs. IEEE Data Eng. Bull. **38**(2), 68–76 (2015)
232. Magnuson, A., Dialani, V., Mallela, D.: Event recommendation using Twitter activity. In: RecSys (2015)
233. Mahmood, A.R., Aref, W.G., Aly, A.M.: FAST: frequency-aware indexing for spatio-textual data streams. In: ICDE (2018)
234. Mahmood, A.R., Aref, W.G., Aly, A.M., Tang, M.: Atlas: on the expression of spatial-keyword group queries using extended relational constructs. In: SIGSPATIAL (2016)
235. Mahmud, J., Nichols, J., Drews, C.: Where is this tweet from? Inferring home locations of Twitter users. In: ICWSM(2012)
236. Makki, R., de Carvalho, E.J., Soto, A.J., Brooks, S., de Oliveira, M.C.F., Milios, E.E., Minghim, R.: ATR-Vis: visual and interactive information retrieval for parliamentary discussions in Twitter. TKDD **12**(1), 31–333 (2018)
237. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Tweets as data: demonstration of TweeQL and TwitInfo. In: SIGMOD (2011)
238. Marcus, A., Bernstein, M.S., Badar, O., Karger, D.R., Madden, S., Miller, R.C.: Twitinfo: aggregating and visualizing microblogs for event exploration. In: CHI (2011)
239. McCullough, D., Lin, J., Macdonald, C., Ounis, I., McCreadie, R.M.C.: Evaluating real-time search over tweets. In: ICWSM (2012)
240. McMinn, A.J., Tsvetkov, D., Yordanov, T., Patterson, A., Szk, R., Rodriguez Perez, J.A., Jose, J.M.: An interactive interface for visualizing events on Twitter. In: SIGIR (2014)
241. Mei, Q., Xu, L., Wondra, M., Su, H., Zhai, C.: Topic sentiment mixture: modeling facets and opinions in weblogs. In: WWW (2007)
242. Meij, E., Weerkamp, W., de Rijke, M.: Adding semantics to microblog posts. In: WSDM (2012)
243. Metwally, A., Agrawal, D., Abbadi, A.E.: Efficient computation of frequent and top-k elements in data streams. In: ICDT (2005)
244. Miranda-Jiménez, S., Graff, M., Tellez, E.S., Moctezuma, D.: INGEOTEC at SemEval 2017 Task 4: A B4MSA ensemble based on genetic programming for Twitter sentiment analysis. In: SemEval (2017)
245. Mishne, G., Dalton, J., Li, Z., Sharma, A., Lin, J.: Fast data in the era of big data: Twitter's real-time related query suggestion architecture. In: SIGMOD (2013)
246. Mishne, G., Lin, J.: Twanchor text: a preliminary study of the value of tweets as anchor text. In: SIGIR (2012)
247. Mohammad, S.: #Emotional tweets. In: *SEM@NAACL-HLT (2012)
248. Mohammad, S., Kiritchenko, S., Zhu, X.: NRC-Canada: building the state-of-the-art in sentiment analysis of tweets. In: SemEval@NAACL-HLT (2013)
249. Mokbel, M., Magdy, A.: Microblogs data management systems: querying, analysis, and visualization (tutorial). In: SIGMOD (2016)
250. Mokbel, M.F., Aref, W.G.: SOLE: scalable on-line execution of continuous queries on spatio-temporal data streams. VLDB J. **17**(5), 971–995 (2008)
251. Mokbel, M.F.H., Ahmed, A.M.M.: System and method for microblogs data management, provisionally filed in U.S. Patent and Trademark Office on August 31, 2015, Application number: 14/841299. http://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&s1=20160070754.PGNR
252. MongoDB. https://www.mongodb.com/ (2018)
253. Mu, L., Jin, P., Zheng, L., Chen, E.H., Yue, L.: Lifecycle-based event detection from microblogs. In: WWW Companion (2018)
254. Mulki, H., Haddad, H., Gridach, M., Babaoğlu, I.: Tw-StAR at SemEval-2017 Task 4: sentiment classification of Arabic tweets. In: SemEval-2017 (2017)
255. Nasim, Z.: IBA-Sys at SemEval-2017 Task 5: fine-grained sentiment analysis on financial microblogs and news. In: SemEval (2017)
256. New Enhanced Geo-targeting for Marketers. https://blog.twitter.com/2012/new-enhanced-geo-targeting-for-marketers (2012)

257. New Study Quantifies Use of Social Media in Arab Spring. www.washington.edu/news/2011/09/12/new-study-quantifies-use-of-social-media-in-arab-spring/ (2011)
258. Nodarakis, N., Sioutas, S., Athanasios K.T., Giannis, T.: Large scale sentiment analysis on Twitter with spark. In: EDBT Workshops (2016)
259. One Million Tweet Map. http://onemilliontweetmap.com/ (2016)
260. Ortega, R., Fonseca, A., Montoyo, A.: SSA-UO: unsupervised Twitter sentiment analysis. In: Joint Conference on Lexical and Computational Semantics (* SEM), vol. 2 (2013)
261. Ozdikis, O., Senkul, P., Oguztüzün, H.: Semantic expansion of tweet contents for enhanced event detection in Twitter. In: ASONAM (2012)
262. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREC (2010)
263. Park, Y., Cafarella, M.J., Mozafari, B.: Visualization-aware sampling for very large databases. In: ICDE (2016)
264. Passant, A., Bojars, U., Breslin, J.G., Hastrup, T., Stankovic, M., Laublet, P.: An overview of SMOB 2: open, semantic and distributed microblogging. In: ICWSM (2010)
265. Paul, D., Li, F., Teja, M.K., Yu, X., Frost, R.: Compass: spatio temporal sentiment analysis of US election what Twitter says! In: SIGKDD (2017)
266. Penagos, C.R., Batalla, J.A., Codina-Filbà, J., Narbona, D.G., Grivolla, J., Lambert, P., Saurí, R.: FBM: combining lexicon-based ML and heuristics for social media polarities. In: SemEval@NAACL-HLT (2013)
267. Peng, M., Zhu, J., Wang, H., Li, X., Zhang, Y., Zhang, X., Tian, G.: Mining event-oriented topics in microblog stream with unsupervised multi-view hierarchical embedding. TKDD **12**(3), 38 (2018)
268. Phelan, O., McCarthy, K., Smyth, B.: Using Twitter to recommend real-time topical news. In: RecSys (2009)
269. Popescu, A.M., Pennacchiotti, M.: Detecting controversial events from Twitter. In: CIKM (2010)
270. Prediction, Optimization and Control for Information Propagation on Networks: A Differential Equation and Mass Transportation Based Approach. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1620342 (2017)
271. Presto. http://prestodb.io/ (2018)
272. Public Health Emergency, Department of Health and Human Services. http://nowtrending.hhs.gov/ (2015)
273. Qadir, A., Mendes, P.N., Gruhl, D., Lewis, N.: Semantic lexicon induction from Twitter with pattern relatedness and flexible term length. In: AAAI (2015)
274. Qian, Y., Tang, J., Yang, Z., Huang, B., Wei, W., Carley, K.M.: A probabilistic framework for location inference from social media. In: CoRR. arXiv:1702.07281 (2017)
275. Qiu, L., Lei, Q., Zhang, Z.: Advanced sentiment classification of Tibetan microblogs on smart campuses based on multi-feature fusion. IEEE Access **6**, 17896–17904 (2018)
276. Rajendram, S.M., Mirnalinee, T.T., et al.: SSN_MLRG1 at SemEval-2017 Task 4: sentiment analysis in Twitter using multi-kernel gaussian process classifier. In: SemEval (2017)
277. Ramage, D., Dumais, S.T., Liebling, D.J.: Characterizing microblogs with topic models. In: ICWSM (2010)
278. Ranganathan, J., Irudayaraj, A.S., Tzacheva, A.A.: Action rules for sentiment analysis on Twitter data using spark. In: ICDM Workshops (2017)
279. Redis. https://redis.io/ (2018)
280. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Context-sensitive Twitter sentiment classification using neural network. In: AAAI (2016)
281. Ren, Y., Zhang, Y., Zhang, M., Ji, D.: Improving Twitter sentiment classification using topic-enriched multi-prototype word embeddings. In: AAAI (2016)
282. Ribeiro, M.H., Calais, P.H., Santos, Y.A., Almeida, V.A.F., Meira, W. Jr.: Characterizing and detecting hateful users on Twitter. In: CoRR. arXiv:1803.08977 (2018)
283. Rios, M., Lin, J.J.: Visualizing the "Pulse" of world cities on Twitter. In: ICWSM Citeseer (2013)
284. Rios, R.A., Pagliosa, P.A., Ishii, R.P., de Mello, R.F.: TSViz: a data stream architecture to online collect, analyze, and visualize tweets. In: SAC (2017)
285. Ritter, A., Etzioni, O., Clark, S., et al.: Open domain event extraction from Twitter. In: SIGKDD (2012)
286. RocksDB. https://rocksdb.org/ (2018)
287. Romero, S., Becker, K.: A framework for event classification in tweets based on hybrid semantic enrichment. Expert Syst. Appl. **118**, 522–538 (2019)
288. Rozental, A., Fleischer, D.: Amobee at SemEval-2017 Task 4: deep learning system for sentiment detection on Twitter. arXiv:1705.01306 (2017)
289. Rudra, K., Ghosh, S., Ganguly, N., Goyal, P., Ghosh, S.: Extracting situational information from microblogs during disaster events: a classification-summarization approach. In: *CIKM* (2015)
290. Rudra, K., Goyal, P., Ganguly, N., Mitra, P., Imran, M.: Identifying sub-events and summarizing disaster-related information from microblogs. In: *SIGIR* (2018)
291. Ryoo, K., Moon, S.: Inferring Twitter user locations with 10 km accuracy. In: WWW Companion (2014)
292. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors. In: WWW (2010)
293. Sang, J., Lu, D., Xu, C.: A probabilistic framework for temporal user modeling on microblogs. In: CIKM (2015)
294. Sankaranarayanan, J., Samet, H., Teitler, B.E., Lieberman, M.D., Sperling, J.: TwitterStand: news in tweets. In: SIGSPATIAL (2009)
295. Sarwat, M.: Recdb: towards DBMS support for online recommender systems. In: Proceedings of the ACM SIGMOD/PODS PhD Symposium 2012, Scottsdale, AZ, USA, May 20, 2012, pp. 33–38 (2012)
296. Sarwat, M., Avery, J.L., Mokbel, M.F.: A RecDB in action: recommendation made easy in relational databases. PVLDB **6**(12), 1242–1245 (2013)
297. Sarwat, M., Avery, J.L., Mokbel, M.F.: RECATHON: a middleware for context-aware recommendation in database systems. In: MDM (2015)
298. Sarwat, M., Moraffah, R., Mokbel, M.F., Avery, J.L.: Database system support for personalized recommendation applications. In: ICDE (2017)
299. Satapathy, R., Guerreiro, C., Chaturvedi, I., Cambria, E.: Phonetic-based microtext normalization for Twitter sentiment analysis. In: ICDM Workshops (2017)
300. Sharma, A., Jerry, J., Praveen, B., Brian, L., Jimmy, L.: GraphJet: real-time content recommendations at Twitter. In: VLDB, pp. 1281–1292 (2016)
301. Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., Deng, X.: Exploiting topic based Twitter sentiment for stock prediction. In: ACL, vol. 2 (2013)
302. Sijtsma, B., Qvarfordt, P., Chen, F.: Tweetviz: visualizing tweets for business intelligence. In: SIGIR (2016)
303. Singh, V.K., Gao, J.R.: Situation detection and control using spatio-temporal analysis of microblogs. In: WWW (2010)
304. Sina Weibo, China Twitter, comes to rescue amid flooding in Beijing. http://thenextweb.com/asia/2012/07/23/sina-weibo-chinas-twitter-comes-to-rescue-amid-flooding-in-beijing/ (2012)
305. Skovsgaard, A., Sidlauskas, D., Jensen, C.S.: Scalable top-k spatio-temporal term querying. In: ICDE (2014)

306. Smith, K.S., McCreadie, R., Macdonald, C., Ounis, I.: Analyzing disproportionate reaction via comparative multilingual targeted sentiment in Twitter. In: ASONAM (2017)
307. Soto, A.J., Brooks, S., Raheleh, M., Milios, E.E.: Twitter message recommendation based on user interest profiles. In: ASONAM (2016)
308. Sparsity Models for Forecasting Spatio-Temporal Human Dynamics. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1737770 (2017)
309. Søgaard, A., Plank, B., Alonso, H.M.: Using frame semantics for knowledge extraction from Twitter. In: AAAI (2015)
310. Song, K., Chen, L., Gao, W., Feng, S., Wang, D., Zhang, C.: Persentiment: a personalized sentiment classification system for microblog users. In: WWW Companion (2016)
311. Sotiropoulos, D.N., Kounavis, C.D., Giaglis, G.M.: Semantically meaningful group detection within sub-communities of Twitter blogosphere: a topic oriented multi-objective clustering approach. In: ASONAM (2013)
312. Soulier, L., Lynda, T., Gia-Hung, N.: Answering Twitter questions: a model for recommending answerers through social collaboration. In: CIKM (2016)
313. Speriosu, M., Sudan, N., Upadhyay, S., Baldridge, J.: Twitter polarity classification with label propagation over lexical links and the follower graph. In: Workshop on Unsupervised Learning in NLP (2011)
314. Steiger, E., Resch, B., Zipf, A.: Exploration of spatiotemporal and semantic clusters of Twitter data using unsupervised neural networks. IJGIS **30**(9), 1694–1716 (2016)
315. Stonebraker, M., Weisberg, A.: The VoltDB main memory DBMS. IEEE Data Eng. Bull. **36**(2), 21–27 (2013)
316. Sundararaman, D., Srinivasan, S.: Twigraph: discovering and visualizing influential words between Twitter profiles. In: Social Informatics (2017)
317. Symeonidis, S., Effrosynidis, D., Kordonis, J., Arampatzis, A.: DUTH at SemEval-2017 Task 4: a voting classification approach for Twitter sentiment analysis. In: SemEval (2017)
318. Symeonidis, S., Kordonis, J., Effrosynidis, D., Arampatzis, A.: DUTH at SemEval-2017 Task 5: sentiment predictability in financial microblogging and news articles. In: SemEval (2017)
319. Tabari, N., Seyeditabari, A., Zadrozny, W.: SentiHeros at SemEval-2017 Task 5: an application of sentiment analysis on financial tweets. In: SemEval (2017)
320. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: SIGKDD (2011)
321. Tang, D., Wei, F., Qin, B., Liu, T., Zhou, M.: Coooolll: a deep learning system for Twitter sentiment classification. In: SemEval@COLING (2014)
322. Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., Qin, B.: Learning sentiment-specific word embedding for Twitter sentiment classification. In: ACL (2014)
323. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment strength detection for the social web. JASIST **63**(1), 163–173 (2012)
324. Topsy Analytics: Find the insights that matter. www.topsy.com (2014)
325. Turet, J.G., Costa, A.P.C.S.: Big data analytics to improve the decision-making process in public safety: a case study in Northeast Brazil. In: Springer ICDSST (2018)
326. Tweet Complete Index. https://blog.twitter.com/engineering/en_us/a/2014/building-a-complete-tweet-index.html
327. TweetTracker: track, analyze, and understand activity on Twitter. tweettracker.fulton.asu.edu/ (2014)
328. Twitter and Informal Science Learning and Engagement. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1438898 (2017)
329. The Power of Images: A Computational Investigation of Political Mobilization via Social Media. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1727459 (2017)
330. Twitter Data Changing Future of Population Research. http://news.psu.edu/story/474782/2017/07/17/research/twitter-data-changing-future-population-research (2017)
331. Twitter Statistics. https://about.twitter.com/company (2018)
332. The Twitter War: Social Media's Role in Ukraine Unrest. news.nationalgeographic.com/news/2014/05/140510-ukraine-odessa-russia-kiev-twitter-world/ (2014)
333. Twitter a Big Winner in 2012 Presidential Election. https://www.computerworld.com/article/2493332/social-media/twitter-a-big-winner-in-2012-presidential-election.html (2012)
334. Topsy Analytics for Twitter Political Index. https://blog.twitter.com/official/en_us/a/2012/a-new-barometer-for-the-election.html
335. Understanding Social and Geographical Disparities in Disaster Resilience Through the Use of Social Media. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1620451 (2017)
336. Vesdapunt, N., Garcia-Molina, H.: Identifying users in social networks with limited information. In: ICDE (2015)
337. Vo, D.T., Zhang, Y.: Target-dependent Twitter sentiment classification with rich automatic features. In: IJCAI (2015)
338. VoltDB. https://www.voltdb.com/ (2018)
339. Vosecky, J., Jiang, D., Leung, K.W.-T., Xing, K., Ng, W.: Integrating social and auxiliary semantics for multifaceted topic modeling in Twitter. ACM TOIT **14**(4), 271–2724 (2014)
340. Vydiswaran, V.G.V., Romero, D.M., Zhao, X., Yu, D., Gomez-Lopez, I.N., Lu, J.X., Iott, B., Baylin, A., Clarke, P., Berrocal, V.J., et al.: "Bacon Bacon Bacon": food-related tweets and sentiment in metro detroit. In: ICWSM (2018)
341. Wakamiya, S., Jatowt, A., Kawai, Y., Akiyama, T.: Analyzing global and pairwise collective spatial attention for geo-social event detection in microblogs. In: WWW Companion (2016)
342. Wang, M., Chu, B., Liu, Q., Zhou, X.: YNUDLG at SemEval-2017 Task 4: A GRU-SVM model for sentiment classification and quantification in Twitter. In: SemEval-2017 (2017)
343. Wang, X., Zhang, Y., Zhang, W., Lin, X., Wang, W.: AP-Tree: efficiently support continuous spatial-keyword queries over stream. In: ICDE (2015)
344. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: CIKM (2011)
345. Wang, Y., Liu, J., Huang, Y., Feng, X.: Using hashtag graph-based topic model to connect semantically-related words without co-occurrence in microblogs. TKDE **28**(7), 1919–1933 (2016)
346. Wang, Y., Siriaraya, P., Nakaoka, Y., Sakata, H., Kawai, Y., Akiyama, T.: A Twitter-based culture visualization system by analyzing multilingual geo-tagged tweets. In: ICADL (2018)
347. Wang, Z., Zhang, Y., Li, Y., Wang, Q., Xia, F.: Exploiting social influence for context-aware event recommendation in event-based social networks. In: INFOCOM (2017)
348. Watanabe, K., Ochi, M., Okabe, M., Onai, R.: Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. In: CIKM (2011)
349. Weber, I., Garimella, V.R.K.: Visualizing user-defined, discriminative geo-temporal Twitter activity. In: ICWSM (2014)
350. Welch, M.J., Schonfeld, U., He, D., Cho, J.: Topical semantics of Twitter links. In: WSDM (2011)
351. Wu, F., Huang, Y.: Personalized microblog sentiment classification via multi-task learning. In: AAAI (2016)
352. Wu, S., Gong, L., Rand, W., Raschid, L.: Making recommendations in a microblog to improve the impact of a focal user. In: RecSys (2012)

353. Wu, X., Bartram, L., Shaw, C.: Plexus: an interactive visualization tool for analyzing public emotions from Twitter data. In: CoRR. arXiv:1701.06270 (2017)
354. Wu, Y.: Language E-learning based on learning analytics in big data era. In: International Conference on Big Data and Education (2018)
355. Xiang, B., Zhou, L.: Improving Twitter sentiment analysis with topic-based mixture modeling and semi-supervised training. In: ACL, vol. 2 (2014)
356. Xie, Q., Zhang, X., Zhixu, L., Zhou, X.: Optimizing cost of continuous overlapping queries over data streams by filter adaption. TKDE **28**(5), 1258–1271 (2016)
357. Xing, C., Wang, Y., Liu, J., Huang, Y., Ma, W.Y.: Hashtag-based sub-event discovery using mutually generative LDA in Twitter. In: AAAI, pp. 2666–2672 (2016)
358. Xiong, X., Mokbel, M.F., Aref, W.G.: SEA-CNN: scalable processing of continuous K-nearest neighbor queries in spatio-temporal databases. In: ICDE (2005)
359. Yang, T.H., Tseng, T.H., Chen, C.P.: deepSA at SemEval-2017 Task 4: interpolated deep neural networks for sentiment analysis in Twitter. In: SemEval (2017)
360. Yao, J., Cui, B., Xue, Z., Liu, Q.: Provenance-based indexing support in micro-blog platforms. In: ICDE (2012)
361. Yen, A.Z., Huang, H.H., Chen, H.H.: Detecting personal life events from Twitter by multi-task LSTM. In: WWW Companion (2018)
362. Yin, H., Cui, B., Chen, L., Hu, Z., Zhang, C.: Modeling location-based user rating profiles for personalized recommendation. TKDD **9**(3), 191–1941 (2015)
363. Yin, Y., Song, Y., Zhang, M.: NNEMBs at SemEval-2017 Task 4: neural Twitter sentiment classification: a simple ensemble method with different embeddings. In: SemEval (2017)
364. Yang, X.W., Yu, Z.: Xinjie: user embedding for scholarly microblog recommendation. In: ACL, vol. 2 (2016)
365. Zhiwen, Y., Wang, Z., Chen, L., Guo, B., Li, W.: Featuring, detecting, and visualizing human sentiment in Chinese micro-blog. TKDD **10**(4), 48 (2016)
366. Zayer, M.A., Gunes, M.H.: Analyzing the use of Twitter to disseminate visual impairments awareness information. In: ASONAM (2017)
367. Zhang, C., Lei, D., Yuan, Q., Zhuang, H., Kaplan, L., Wang, S., Han, J.: GeoBurst+: effective and real-time local event detection in geo-tagged tweet streams. ACM TIST **9**(3), 34 (2018)
368. Zhang, C., Liu, L., Lei, D., Yuan, Q., Zhuang, H., Hanratty, T., Han, J.: Triovecevent: embedding-based online local event detection in geo-tagged tweet streams. In: SIGKDD (2017)
369. Zhang, C., Zhou, G., Yuan, Q., Honglei Z., Yu., Z., Lance K., Wang, S., Han, J.: Geoburst: real-time local event detection in geo-tagged tweet streams. In: SIGIR (2016)
370. Zhang, D., Liu, Y., Lawrence, R.D., Chenthamarakshan, V.: Transfer latent semantic learning: microblog mining with less supervision. In: AAAI (2011)
371. Zhang, D., Chan, C.Y., Tan, K.L.: Processing spatial keyword query as a top-k aggregation query. In: SIGIR (2014)
372. Zhang, D., Nie, L., Luan, H., Tan, K.-L., Chua, T.-S., Shen, H.T.: Compact indexing and judicious searching for billion-scale microblog retrieval. ACM TOIS **35**(3), 27 (2017)
373. Zhang, D., Tan, K.L., Tung, A.K.H.: Scalable top-k spatial keyword search. In: EDBT (2013)
374. Zhang, H., Chen, G., Ooi, B.C., Wong, W.F., Wu, S., Xia, Y.: "Anti-caching"-based elastic memory management for big data. In: ICDE (2015)
375. Zhang, J., Zhang, R., Sun, J., Zhang, Y., Zhang, C.: TrueTop: a sybil-resilient system for user influence measurement on Twitter. IEEE/ACM TON **24**(5), 2834–2846 (2016)
376. Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., Liu, B.: Combining lexicon-based and learning-based methods for Twitter sentiment analysis. HP Laboratories, Technical Report HPL-2011, p. 89 (2011)
377. Zhang, Y., Szabo, C., Sheng, Q.Z., Fang, X.S.: SNAF: observation filtering and location inference for event monitoring on Twitter. WWW J. **21**(2), 311–343 (2018)
378. Zhang, Y., Fan, Y., Ye, Y., Li, X., Winstanley, E.: Utilizing social media to combat opioid addiction epidemic: automatic detection of opioid users from Twitter. In: AAAI Workshops (2018)
379. Zhang, Z., Lan, M.: Estimating semantic similarity between expanded query and tweet content for microblog retrieval. In: TREC (2014)
380. Zhao, J., Lan, M., Zhu, T.: ECNU: expression-and message-level sentiment orientation classification in Twitter using multiple effective features. In: SemEval (2014)
381. Zhao, J., Gui, X., Tian, F.: A new method of identifying influential users in the micro-blog networks. IEEE Access **5**, 3008–3015 (2017)
382. Zhao, J., Lui, J.C.S., Towsley, D., Wang, P., Guan, X.: Sampling design on hybrid social-affiliation networks. In: ICDE (2015)
383. Zhao, L., Chen, F., Chang-Tien, L., Ramakrishnan, N.: Online spatial event forecasting in microblogs. ACM TSAS **2**(4), 15 (2016)
384. Zhao, W.X., Guo, Y., He, Y., Jiang, H., Wu, Y., Li, X.: We know what you want to buy: a demographic-based system for product recommendation on microblogs. In: KDD (2014)
385. Zhao, W.X., Sui, L., Yulan, H., Chang, E.Y., Ji-Rong, W., Li, X.: Connecting social media to e-commerce: cold-start product recommendation using microblogging information. TKDE **28**(5), 1147–1159 (2016)
386. Zheng, X., Sun, A., Wang, S., Han, J.: Semi-supervised event-related tweet identification with dynamic keyword generation. In: CIKM (2017)
387. Zhou, D., Chen, L., He, Y.: An unsupervised framework of exploring events on Twitter: filtering, extraction and categorization. In: AAAI (2015)
388. Zhou, D., Gao, T., He, Y.: Jointly event extraction and visualization on Twitter via probabilistic modelling. In: ACL, vol. 1 (2016)
389. Zhou, X., Chen, L.: Event detection over Twitter social media streams. PVLDB **23**(3), 381–400 (2014)
390. Zhou, Y., Cristea, A.I., Shi, L.: Connecting targets to tweets: semantic attention-based model for target-specific stance detection. In: WISE (2017)
391. Zhu, R., Wang, B., Yang, X., Zheng, B., Wang, G.: SAP: improving continuous top-K queries over streaming data. In: ICDE (2018)
392. Zhu, X., Huang, J., Zhu, S., Chen, M., Zhang, C., Li, Z., Dongchuan, H., Chengliang, Z., Li, A., Jia, Y.: NUDTSNA at TREC 2015 microblog track: a live retrieval system framework for social network based on semantic expansion and quality model. In: TREC (2015)
393. Zini, T., Becker, K., Dias, M.: INF-UFRGS at SemEval-2017 Task 5: a supervised identification of sentiment score in tweets and headlines. In: SemEval (2017)