Discovering Generalized Association Rules from Twitter

Discovering generalized association rules from Twitter

Luca Cagliero* and Alessandro Fiori

Abstract

The increasing availability of user-generated content coming from online communities allows the analysis of common user behaviors and trends in social network usage. This paper presents the TweM (Tweet Miner) framework that entails the discovery of hidden and high level correlations, in the form of generalized association rules, among the content and the contextual features of posts published on Twitter (i.e., the tweets). To effectively support knowledge discovery from tweets, the TweM framework performs two main steps: (i) taxonomy generation over tweet keywords and context data and (ii) generalized association rule mining, driven by the generated taxonomy, from a sequence of tweet collections. Unlike traditional mining approaches, the generalized rule mining session performed on the current tweet collection also considers the evolution of the extracted patterns across the sequence of the previous mining sessions to prevent the discarding of rare knowledge that frequently occurs in a number of past extractions. Experiments, performed on both real Twitter posts and synthetic datasets, show the effectiveness and the efficiency of the proposed TweM framework in supporting knowledge discovery from Twitter user-generated content.

Keywords: Social network analysis, User-generated content, Generalized association rule mining, Taxonomy inference

*Dipartimento di Automatica e Informatica, Politecnico di Torino Corresponding author's email address: Luca Cagliero, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino (Italy), email: luca.cagliero@polito.it.

1 Introduction

Social networks are becoming one of the most commonly used resources to communicate news or share documents, photos, and videos with a large community. Social networks sites, such as Facebook or Twitter, are accessed by millions of people every day. The huge amount of data generated by social network users represents a powerful source of knowledge that is worth considering in the analysis of online social communities and their user activities. Indeed, the application of well-founded data mining techniques to the online community user-generated content (UGC) is definitely an appealing and challenging research topic. Recently, many research efforts have been devoted to defining user profiles and behaviors, mining opinions about products, and generating models to represent the shared knowledge [14]. For instance, knowledge provided by online communities has been profitably exploited in social behavior analysis [26, 28], Web object categorization [39], and service recommendation [25, 32, 38]. In the last years, Twitter has become one of the most popular micro-blogging and social network Web sites. Thus, the analysis of Twitter UGC has captured the interest of the research community. For instance, TwitterMonitor [28] extracted contextual knowledge from Twitter streams to detect most common topic trends, while, in [15], topic trends are discovered, by using information retrieval techniques, to support analyst decision-making.

This paper presents the TweM (Tweet Miner) framework that addresses the extraction of hidden and high level recurrences, in the form of generalized association rules, from messages posted by Twitter users. The analysis of Twitter posts is focused on two different but related features: their textual content and their submission context (e.g., the place and the submission timestamp). TweM is based on three-step process: (i) taxonomy generation, (ii) generalized rule mining from tweet collections, and (iii) rule querying, based on the characteristics of the discovered rules. A taxonomy, i.e., a is-a hierarchy generated over keywords and contextual data, is used to drive the extraction of generalized association rules. A novel generalized association rule miner, namely the EGP MINER (Evolving Generalized Pattern MIner), exploits the historical evolution of patterns in the sequence of tweet collections.

The taxonomy generation task is accomplished by discovering and selecting strong associations over the analyzed data items (e.g., keywords, times, places) suitable for driving the generalization process. Relationships holding among contextual tweet features (e.g., the time and the geographical coordinates) are derived by means of Extraction, Transformation and Load (ETL) procedures, while an association-based approach is proposed to infer reliable implications among tweet content keywords. In particular, high-quality (i.e., frequent and high-confidence) associations involving couples of tweet keywords are selected and organized in a hierarchical fashion. The proposed selection strategy prefers high-quality aggregations well spread across different abstraction levels and, thus, combines knowledge aggregations so that they provide meaningful and not too general concepts. For instance, consider the following high-quality (i.e., frequent and high-confidence) rules $Obama \Rightarrow President$, $President \Rightarrow Person$, and $Obama \Rightarrow Person$ stating that the corresponding relationships among keywords $Obama$, $President$, and $Person$ hold. By generalizing keywords based on the discovered associations the most specialized keywords (e.g., $Obama$) are gen-

eralized into concepts at a higher abstraction level (e.g., $President$). To aggregate the low level concept $Obama$ at a higher level of abstraction, the two-step rule chain $Obama \Rightarrow President \Rightarrow Person$ is preferred to the single aggregation rule $Obama \Rightarrow Person$.

Taxonomies are exploited to discover correlations among tweets in the form of generalized association rules [33]. The mining process is performed on a sequence of tweet collections. Differently from traditional approaches, to guarantee the selection of the most relevant and persistent knowledge, the history of the already extracted patterns is exploited to drive the generalization process. More specifically, patterns that are frequent in a number of previously analyzed tweet sets (e.g., in a number of previous days) are expected to be of interest even in the current one (e.g., in the current day). Thus, as soon as they become infrequent, the discarding of their covered knowledge is, possibly, prevented by triggering their corresponding generalization. The TweM framework allows the investigation of recurrent trends and spatial correlations in the evolution of most relevant tweet topics. For instance, a correlation between a newsworthy topic and a geographical location, e.g., $(Keyword, Obama) \Rightarrow (Location, New\ York\ City)$, may be pointed out as frequent in a couple of days, but infrequent in the next day. The EGP MINER algorithm triggers its generalization when it performs generalized rule mining from tweets posted in the last day, e.g., by aggregating the term $Obama$ in $President$ and the city in the corresponding state. Thus, the generalized rule $(Keyword, President) \Rightarrow (Location, New\ York\ State)$ is extracted.

This paper is organized as follows. Section 2 overviews most relevant related works concerning data mining from user-generated content. Section 3 presents the architecture of the proposed framework and describes its main blocks. Section 4 assesses the effectiveness of TweM in extracting hidden information from tweets as well as describes examples of real-life use-cases. Finally, Section 5 draws conclusions and presents future developments.

2 Related work

Since the birth of social network sites on, many research efforts have been devoted to investigating the structure of online communities and identifying patterns relevant for characterizing the dynamics behind community user/group behavior. For instance, authors in [27] investigated the evolution of online communities by means of the maximum a posteriori (MAP) estimation, while, in [8] click-stream data is analyzed to identify most common Web user activities, such as universal searches, message sending, and community creation. Differently, in [19] the characteristics of the lifetime of the user-generated content (UGC) are investigated.

The application of data mining techniques to discover relevant social knowledge from the UGC has shown a steady growth in recent years [14]. Several data mining approaches, based on UGC analysis, are focused on (i) developing new recommendation systems to enhance the quality of product promotions [38], (ii) improving the understanding of online resources [6, 26, 39], and (iii) building query engines that take advantage of the emerging semantics in social networks [7, 22]. The UGC analysis can be also useful for identifying most notable topic trends. For instance, in [15] trend

patterns extracted from Twitter are exploited to support analyst decision-making. More specifically, they focused on discovering Twitter users who contribute towards the discussions on specific trends. A number of previous approaches addressed the discovery of association rules from user-generated content. For example, in [10] the authors compare several data mining techniques, among which association rule mining, to discover user patterns from Facebook data. Differently, the classification and link prediction method presented in [13] exploits association rules to discover correlations among data features related to the major user interests. However, to the best of our knowledge, the discovery of generalized rules in the presence of taxonomies constructed over both the tweet content and context of publication has never been investigated so far.

This paper address the usage of a well-founded data mining technique, i.e., generalized association rule mining, to perform knowledge discovery from Twitter posts. The problem of generalized association rule mining has been first introduced in [33] in the context of market basket analysis as an extension of the traditional association rule mining task [1]. Several data mining approaches are focused on proposing more efficient generalized itemset mining algorithms (e.g., [3, 20, 29, 40]). They all try to avoid exhaustive taxonomy evaluation by preliminary pruning uninteresting patterns. Among them, authors in [3] exploit a support-driven approach to itemset generalization, i.e., they generalize an itemset only if has at least an infrequent descendant according to the given taxonomy. A similar approach has been also exploited to support context-aware user and service profiling [4]. However, all the aforementioned approaches do not consider the evolution of patterns extracted in different mining sessions to drive the generalization process. Active data mining [2] was the first attempt to represent and query the history of the discovered association rule quality indexes by incrementally updating a common rule base. More recently, other approaches focused on detecting changes in itemset and rule quality indexes (e.g., support and confidence), based on either objective and subjective measures [5, 11, 12, 36]. However, these methods either do not address rule generalization or drive the generalization process based on the characteristics of the current time period solely. Differently, the approach proposed in this paper discriminates and generalizes patterns based on their frequency of occurrence in both the current and the previous time periods. More specifically, patterns that are frequent in a number of past mining sessions are expected to be of interest even in the current one. Thus, as soon as they become infrequent, rare knowledge discarding is possibly prevented by triggering their corresponding generalizations.

The generalized association rule mining process is typically driven by analyst-provided taxonomies. Differently, TweM integrates a taxonomy generation step to ease the domain expert's task. A number of approaches have been devoted to building or automatically inferring taxonomies from data by exploiting well-known data mining techniques. Most of them propose to exploit hierarchical clustering algorithms to organize concepts [16, 18, 23]. However, taxonomies extracted by means of clustering approaches may provide weakly informative results [21]. In the last years, a few attempts to support taxonomy inference from the user-generated content by means of association discovery has been done. For instance, the analysis of folksonomies [30] and Web resource tags [17, 31] has been recently conducted by using association discovery techniques. In [17], the authors analyzed the correlations among context-aware tags to perform taxonomy generation. Association rules, extracted by means of Apriori algo-

rithm [1], are modeled in a graph-based representation and exploited to represent semantic relationships among concepts. TweM adopts, similarly to [17], an association-based approach to discover meaningful aggregations over the tweet keywords. Unlike previous approaches, it entails (ii) discriminating among potentially relevant aggregations based on their suitability to drive the generalized rule mining process, and (ii) coping with heterogeneous data features (i.e., contextual data and content keywords).

3 The TweM Framework

The TweM (Tweet Miner) framework is a data mining environment that focuses on supporting domain experts in the discovery of relevant recurrences from the user-generated Twitter posts. To address this issue, it analyzes the evolution of the most significant patterns hidden in a sequence of tweet collections. Figure 1(a) reports the TweM framework architecture. It is composed of the following main blocks:

- **User-generated content representation.** Tweets are modeled as records (i.e., set of items) that describe either their content (i.e., the most relevant keywords) and their submission contextual features (e.g., the geographical location, the time stamp). Tweets are partitioned and stored in a sequence of collections based on the values of a selection of the considered features.
- **Taxonomy generation.** This block addresses the generation of taxonomies built over the tweet content and contextual features. Taxonomies include a set of aggregation hierarchies that provides a high level abstraction of the mined knowledge.
- **Evolving generalized pattern miner.** A novel generalized association rule mining algorithm is exploited to discover high level correlations among the tweet collections, according to the generated taxonomy (see Figure 1(b)). Generalized association rules are discovered from each tweet collection by adopting the usual two-step process: (i) frequent generalized itemset extraction, and (ii) rule generation, starting from the extracted frequent generalized itemsets. Itemset generalization, driven by the taxonomy, is lazily triggered based on the analysis of its observed frequency in the previously analyzed collections belonging to the same sequence. The extracted rules are ranked, based on their support and confidence values, and queried, according to either their content or schema, to allow analysts to retrieve the information of interest efficiently.

A more detailed description of the main TweM blocks is presented in the following sections.

3.1 User-generated content representation

A suitable user-generated content representation is needed to successfully accomplish the mining task. Twitter (`http://twitter.com`) posts can be accessed by means of the Search Application Programming Interfaces (APIs). Data returned by the Twitter

APIs is stored in the JSON (Java Script Object Notation) format, which is an XML-based standard for client-server data exchange. A simplified example of two tweet messages in the JSON format is reported in Figure 2. Tweets are characterized by short textual messages enriched by several contextual information (e.g., publication place, city, date, hour). Some of the available contextual features are peculiar characteristics of the context in which tweets are posted (e.g., the GPS coordinates), while others are just high level aggregations of the previous ones (e.g., the city).

Consider the textual message and the low level contextual features first. Couples $(attribute, value)$, where $attribute$ is the textual message or the description of the contextual feature (e.g., the date) and $value$ is the collected information (e.g., "This is a message by Obama", 2010-10-10), are denoted as items in the following. A tweet could be represented as a set of items, called record, in which each attribute occurs at most once. Each record is characterized by a level l that identifies the collection, in the sequence of tweet sets, to which the record (tweet) belongs to. A set of records (tweets), all characterized by a common level l, is denoted as relational tweet set of level l.

Definition 1 Relational tweet set of level l. *Let $\mathcal{T}=\{t_1, t_2, \ldots, t_n\}$ be a set of attributes, which describes the main data features and $\Omega=\{\Omega_1, \Omega_2, \ldots, \Omega_n\}$ the corresponding domains. Let r be a set of pairs $(t_i, value_i)$, called record, where $value_i \in \Omega_i$ and each t_i appears at most once in r. A relational tweet set D of level l is a collection of records, where each record r is characterized by level l.*

Since Twitter posts do not comply with the relational tweet set format, a preprocessing phase is needed. A data cleaning procedure is exploited to discard useless or redundant information and correctly manage missing values. Furthermore, the cleaned data is modeled as a relational tweet set (Cf. Definition 1), in which the records represent (i) the most representative keywords belonging to the tweet (see Section 3.2.2), and (ii) the tweet contextual features. For each tweet, the following pieces of information are considered:

- **Location**: GPS coordinates
- **Time**: publication date and time stamp
- **Content**: keywords

For instance, suppose to partition the retrieved tweets in 2-hour time intervals and sort them in order of increasing time interval. Tweets reported in Figure 2 belong to level 7 as their time stamps are in the range $[12p.m., 14p.m.)$. Their relational tweet schema is reported in Figure 3, where the relation primary key (i.e., the Tweet ID attribute) is printed in bold.

3.2 Taxonomy generation

This block aims at generating taxonomies, tailored to the analyzed data, that are suitable for effectively driving the generalized association rule mining process. A taxonomy is a hierarchical representation of the main concepts within a domain and the is-a

relationships holding among them. It is composed of aggregation hierarchies, namely the aggregation trees, built over the domains of the source data attributes.

Definition 2 Aggregation tree. *Let t_i be an attribute and Ω_i its domain. An aggregation tree AT_i is a tree representing a predefined set of aggregations over values in Ω_i. AT_i leaves are all the values in Ω_i. Each non-leaf node in AT_i is an aggregation of all its children. Node $\perp$ aggregates all values for attribute t_i.*

Let $\mathcal{T}$=$\{t_1, t_2, \ldots, t_n\}$ be a set of attributes and ρ=$\{AT_1, \ldots, AT_m\}$ a set of aggregation trees defined on $\mathcal{T}$. We define a taxonomy $\Gamma \subseteq \rho$ as a set of aggregation trees. For the sake of simplicity, in the following we will consider taxonomies that contain at most one aggregation tree $AT_i \in \rho$ for each attribute $t_i \in \mathcal{T}$. A portion of an example aggregation tree built over the $date$ attribute items is reported in Figure 4.

TweM adopts different taxonomy generation strategies to construct aggregation trees over the tweet content and its contextual features. In the following sections, they will be discussed separately.

3.2.1 Taxonomy generation over context data

Taxonomies over contextual data features (e.g, the spatial and the temporal information) can be derived by means of aggregation functions based on a hierarchical model. The hierarchical model represents the relationships between different levels of aggregation. Similarly to what usually done in data warehousing [24], this information is extracted by means of Extraction, Transformation and Load (ETL) processes, called here aggregation functions. For example, in the relational tweet representation, aggregation functions may define either aggregations among different contextual attributes (e.g., $City \Rightarrow State$) or aggregations over a singular contextual attributes (e.g., $Date \Rightarrow Semester$) which could be derived by simply parsing the corresponding attribute domain values. By applying aggregation functions, the analyst may generate taxonomies over the contextual data without a prior knowledge about the analyzed data distribution.

Given a set of aggregation functions built over the tweet contextual features, we associate with each item the corresponding set of generalizations, organized in a hierarchical fashion. For instance, consider a temporal contextual feature (e.g., $Month$) that represents a high level abstraction of another one (e.g., $Date$). A conceptual hierarchy of aggregations may be devised by mapping the two attribute domains by means of the corresponding aggregation function (e.g., $Date \Rightarrow Month$). Consider again the $Date$ attribute and its high level aggregation $Semester$. Although the corresponding higher level attribute does not exist yet, the mapping may be simply derived by parsing the lower level $Date$ domain values (e.g., $2010-10-10$) and, thus, generating the upper level concepts (e.g., $2nd\ Semester\ 2010$) according to the corresponding aggregation function (i.e., $Date \Rightarrow Semester$). In Table 1 the aggregation functions exploited in the experiments (see Section 4) for the generation of the taxonomies over temporal and spatial contextual data features are resumed. However, the TweM framework allows the usage of different and more complex aggregation functions as well.

Algorithm 1 Keyword taxonomy generation

Input: $\mathcal{T}$ /*collection of tweet content stems */, min_sup /* minimum support threshold */, min_conf /* minimum confidence threshold */
Output: KT /* taxonomy over tweet content stems */
1: max_len = 2 /* maximum rule length */
2: // Apriori-based extraction of association rules satisfying both min_sup, min_conf, and max_len
3: $R := Apriori(\mathcal{T}, min\_sup, min\_conf, max\_len)$
4: $ARG := G(V, E)$
5: // rule pruning
6: **for all** r in R **do**
7: **if** $sup(r_{body}) < sup(r_{tail})$ **then**
8: $V := V \cup r_{body} \cup r_{tail}$
9: $E := E \cup r$
10: **end if**
11: **end for**
12: // assign aggregation level to each node
13: $L = \{v | v \in ARG \wedge in\_degree(v) = 0\}$
14: **for all** v in ARG/L **do**
15: $al_v = max_{l_i \in L} \{h(l_i, v)\}$
16: **end for**
17: $KT :=$ pruneGraph(ARG) /* prune ARG edges and generate the taxonomy */
18: **return** KT

3.2.2 Taxonomy generation over tweet keywords

Taxonomies over the tweet keywords are generated by following a two-step procedure:

- **Association rule graph extraction.** The bag-of-word (BOW) representation of the tweet content is processed by a traditional Apriori-based association rule mining algorithm [1] to extract strong correlations among couples of frequent terms. The most reliable rules, i.e., the ones that frequently occur in the analyzed data and hold in most cases, are represented in a graph-based model, namely the association rule graph.

- **Graph partitioning and pruning.** The association rule graph is visited and pruned to generate taxonomies suitable for driving the generalized rule mining task. A set of high-quality rules well spread across the taxonomy aggregation levels is preferred to a combination of less specialized (i.e., too general) aggregations.

The pseudo-code of the keyword taxonomy generation procedure is reported in Algorithm 1. In the following, each algorithm step is described in detail.

Association rule graph extraction. This first step focuses on building an association rule graph [17] that captures and represents strong (i.e., frequent and high-confidence) correlations among couples tweet content keywords. Each tweet textual content is modeled by means the BOW representation [35]. A stemming algorithm is exploited to remove stop-words, numbers, and website URLs to avoid noisy information and retrieve the stems of the processed terms. Since the goal is to identify the stems that represent concepts rather than just term root forms, we selected, among the ones available in literature, a stemming algorithm based on WordNet [9].

The BOW vector associated with each tweet could be represented as a transaction (i.e., a set of items), in which each WordNet stem represents an item. The Apriori algorithm [1] is exploited to discover hidden correlations among the transactional tweet

content representation (see line 3 in Algorithm 1). Association rule mining is constrained by both (i) minimum support and confidence thresholds to discover strong rules (i.e., frequent association rules that hold in most cases), and (ii) a maximum rule length, which stops the iterative Apriori itemset mining loop when itemsets of length greater than two are considered. Extracted rules are then combined in a graph-based representation, namely the association rule graph (lines 6-11). Association rule graph vertexes are the tweet content stems, while edges reflect the implications stated by the extracted rules. A more formal definition of association rule graph follows.

Definition 3 Association rule graph. *Let $\mathcal{T}$= $\{T_1, T_2, \ldots, T_n\}$ be a tweet content collection in which each tweet content T_i= $\{t_1, t_2, \ldots, t_k\}$ is represented as a collection of stems $t_j \in T_i$. Let min_sup and min_conf be, respectively, the minimum support and confidence thresholds. Let $\mathcal{R}$ be the set of association rules $t_i \Rightarrow t_j$ such that $\forall\ r \in \mathcal{R}$: (i) sup(r) $\geq min\_sup$, (ii) $conf(r) \geq min\_conf$, and (iii) $sup(t_i) < sup(t_j)$. The association rule graph ARG is an oriented graph whose vertexes are all distinct stems $t_i \in T_j \in \mathcal{T}$. An oriented edge e_{ij} belongs to ARG and connects vertex t_i to vertex t_j if and only if exists $r : t_i \Rightarrow t_j$ such that $r \in R$.*

Suppose that, for instance, the following six association rules, satisfying all mining constraints, have been extracted:

- $Obama \Rightarrow President$
- $Clinton \Rightarrow President$
- $Obama \Rightarrow Democratic$
- $Clinton \Rightarrow Secretary$
- $Democratic \Rightarrow President$
- $Barack \Rightarrow Democratic$

According to Definition 3, the corresponding association rule graph is reported in Figure 5(a).

Graph partitioning and pruning. This step aims at reducing the association rule graph to a tree-based hierarchy (i.e., a taxonomy) by pruning less relevant edges. The problem of selecting a subset of graph edges such that (i) reduces ARG to a taxonomy and (ii) maximizes the significance of the reduced graph according to a given cost function is known to be NP-hard. Indeed, we propose an heuristics that partitions vertexes in aggregation levels and keeps adjacent levels connected so that keywords are maximally spread across the aggregation levels. Notice that the quality of the selected relationships is preliminary guaranteed by the mining constraints enforced during the association rule mining step. By maximizing the spread of the selected edges among the identified aggregation levels more specialized aggregations are preferred and used to drive the generalization process to reduce bias at higher abstraction levels. Consider, for instance, the following concepts: *Football*, *Sport*, and *Activity*. The generalization of keyword *Football* in either the higher level concept *Sport* or directly in *Activity*

are both sound and, thus, acceptable aggregations. However, knowledge discovery driven by the generalization process may become more effective when a chain of more specialized aggregations is provided, e.g., if *Football* is generalized in *Sport* that, in turn, is generalized in *Activity*. To achieve this goal, the following steps are sequentially performed: (i) vertex labeling and (ii) edge pruning.

(i) Vertex labeling. To each vertex v belonging to ARG, an aggregation level al_v, which represents the level of abstraction of the corresponding stem (keyword) in the taxonomy, is assigned (lines 13-16). Let L be the set of vertexes belonging to ARG characterized by no incoming edges (i.e., in-degree equal to 0). Let v and x be two arbitrary graph vertexes and let $h(x, v)$ be the maximum number of hops on ARG between vertexes x and v. Without any loss of generality, we set $h(x, v)$=0 if the two nodes are disconnected. The aggregation level al_v of an arbitrary vertex v belonging to ARG is defined by:

$$al_v = max_{l_i \in L} \{h(l_i, v)\} \quad (1)$$

Thus, all vertexes belonging to L have aggregation level equal to 0 and could be selected as leaf nodes of the resulting keyword taxonomy. In Figure 5(a) the aggregation level associated with each vertex is put in brackets. Vertexes $Obama$, $Clinton$, and $Barack$ are taxonomy leaves (i.e., aggregation level 0), while $Secretary$ and $Democratic$ are aggregations of level 1, and $President$ has aggregation level 2.

(ii) Edge pruning. This step prunes the set of available edges so that (i) all vertexes keep connected and (ii) all vertexes, except for the root nodes (i.e., the nodes having no outgoing edges), have out-degree equal to one. To this aim, for each not-leaf node a sub-graph including all its descendants is built. Each subgraph is characterized by the aggregation level of its root node. Starting from the subgraphs with lowest aggregation level, a top-down depth-first visit is performed. To avoid vertex isolation, the procedure visits subgraphs/nodes in order of ascending in-degree, and, on equal terms, in lexicographical order. For each subgraph, once a node is visited, all its outgoing edges belonging to ARG that are not connected to its ancestor in the corresponding subgraph are pruned.

Consider, for instance, the labeled graph reported in Figure 5(a). We identify three different subgraphs whose root nodes are, respectively, $Secretary$, $Democratic$, and $President$. Among the nodes with the same aggregation level, the one with minimum in-degree is considered first (e.g., $Secretary$ at aggregation level 1). Once a descendant is visited (e.g., $Clinton$), its outgoing edges connected with the ancestors not belonging to the subgraph (e.g, the edge from $Clinton$ to $President$) are pruned from ARG. The resulting keyword taxonomy is reported in Figure 5(b). Note that $Secretary$ becomes a root node and its aggregation level corresponds to the generated aggregation tree height.

Since the vertexes belonging to the keyword taxonomy represent the most frequent stems in the tweet collection, keywords included in each tweet record are an ordered selection of the most representative content stems. For instance, consider the tweet set reported in Figure 2. According to the keyword taxonomy reported in Figure 5(b), records belonging to the corresponding relational tweet set (see Figure 3) include,

respectively, the items $(Keyword, Obama)$ and $(Keyword, Clinton)$, whose item values are nodes of the corresponding keyword taxonomy. Finally, the generated taxonomy is validated by the domain expert, who is in charge of assessing its semantic soundness.

3.3 Evolving Generalized Pattern MIner

This block focuses on discovering strong correlations, in the form of generalized association rules, from a sequence of tweet collections. The generalization process exploits the previously inferred taxonomies to discover correlations at higher abstraction levels. The rule mining process is typically addressed by means a two-step process [33]: (i) generalized itemset mining and (ii) generalized association rule generation. This section is organized as follows. Section 3.3.1 provides preliminary definitions and the related notation. Section 3.3.2 thoroughly describes the EGP MINER algorithm itemset mining step. Finally, Section 3.3.3 describes the generalized rule generation step.

3.3.1 Preliminary definitions

The first step of the generalized association rule mining process entails the discovery of generalized and not generalized itemsets from the analyzed data.

Definition 4 (Generalized) itemset. *Let T be a set of attributes, Ω the corresponding domain, and Γ a taxonomy defined on values $value_i \in \Omega_i$. A not generalized itemset is a set of items $(t_i, value_i)$ in which each attribute t_i may occur at most once. A generalized itemset is an itemset that includes at least a generalized item $(t_i, value_i)$ such that $value_i \in \Gamma$.*

For instance, according to the aggregation functions reported in Table 1, $\{(Place, New\ York)$,$(date, October\ 2010)\}$ is a generalized itemset of length 2 (i.e., a generalized 2-itemset). A (generalized) itemset covers a given record (tweet) r of level l, i.e., $r \in D_l$, if all its (possibly generalized) items $x \in X$ are either included in r, or ancestors of items $i \in r$ (i.e., $\exists\ i \in leaves(x) \mid i \in r$). The support of a (generalized) itemset X in a relational tweet set D_l of level l is given by the number of tweets $r \in D_l$ covering X divided by the cardinality of D_l. Consider the example relational tweet set D_7 of level 7 reported in Figure 3. The generalized itemset $\{(Place, New\ York), (date, October\ 2010)\}$ has support 50% in D_7 as it covers half of the records belonging to the tweet set. A descendant is associated with similar itemsets at different aggregation levels. We denote a (generalized) itemset X as a descendant of a generalized itemset Y if (i) X and Y have the same length and (ii) for each item $y \in Y$ there exists at least an item $x \in X$ that is a descendant of y. Consider the itemset $\{(Place, New\ York), (date, 2010-10-10)\}$. According to the aggregation tree generated from the aggregation functions reported in Figure 4, it is an example of descendant of the generalized itemset $\{(Place, New\ York)$, $(date, October\ 2010)\}$.

The second mining step focuses on generating generalized association rules from the set of extracted (generalized) itemsets. A generalized association rule is an implication $X \Rightarrow Y$, where X and Y are disjoint generalized or not generalized itemsets, as stated by the following definition.

Definition 5 Generalized association rule. *Let A and B be two (generalized) itemsets such that attr(A) $\cap$ attr(B) $= \emptyset$, where $attr(X)$ is the set of attributes belonging to itemset X. A generalized association rule is represented in the form $A \Rightarrow B$, where A and B are, respectively, the body and the head of the rule.*

Generalized association rules are usually characterized by support and confidence quality indexes. The rule support sup is the support of the (generalized) itemset $A \cup B$, while the rule confidence $conf$ is given by $\frac{sup(A \cup B)}{sup(A)}$ and represents the rule strength.

3.3.2 The EGP MINER itemset mining

The EGP MINER (Evolving Generalized Pattern MIner) itemset mining step extracts, from each relational tweet sets (Cf. Definition 1) of level l, all frequent not generalized itemsets and the set of frequent generalized itemsets having at least an infrequent descendant at level l that is frequent in the previous $history\_size$ levels. We formalize the problem addressed by EGP MINER as follows.

Definition 6 EGP MINER **itemset mining problem statement.** *Given a set of relational tweet sets $D = \{D_1, D_2, \ldots, D_n\}$ of increasing levels 1, 2, ..., n, a taxonomy Γ built over D, a minimum support and confidence thresholds min_sup and min_conf, and a maximum history size $history\_size$, the* EGP MINER *itemset mining step extracts from each relational tweet set D_l, $1 \leq l \leq n$*

A) the set of all not generalized itemsets satisfying the minimum support threshold min_sup in D_l, and

B) the set of all generalized itemsets having at least a descendant (with respect to Γ) such that (i) do not satisfy min_sup in D_l, and (ii) satisfy min_sup in all D_j such that $j \geq 1$ and $l - history\_size \leq j \leq l - 1$.

If $history\_size = 0$ then condition (B) could be ignored.

Algorithm 2 reports the pseudo-code of the itemset mining step of the EGP MINER algorithm. It iteratively performs generalized itemset mining sessions from tweet collections D_i of increasing level i by adopting an Apriori-like level-wise approach [1]. At an arbitrary step k, EGP MINER accomplishes the following tasks: (i) k-itemset generation from dataset D_i (line 20), (ii) support counting and generalization of (generalized) k-itemsets that are infrequent in D_i but frequent in all D_j such that $j \geq 1$, $i - history\_size \leq j \leq i$ (lines 6-17), (iii) infrequent candidate pruning, and (iv) generation of k-candidate (generalized) itemsets of length $k + 1$ by joining k-itemsets (line 20). The relational data format (Cf. Definition 1) allows preventing the generation of candidates including two items belonging to the same attribute. The generalized itemset generation procedure is lazily invoked only when itemsets infrequent in D_i but frequent in all previous D_j are considered. Given a (generalized) itemset c and a taxonomy Γ built over D, the taxonomy evaluation procedure generates a set of generalized itemsets by applying on each item $(t_j, value_j)$ of c the corresponding aggregation tree $AT_j \in \Gamma$ (see Definition 2). All the itemsets obtained by replacing one or more items in c with their generalized versions are generated and included into the Gen set (line 12).

Algorithm 2 Evolving Generalized Pattern MIner itemset mining step

Input: set of relational tweet sets $D = \{D_0, D_1, \ldots, D_n\}$ of levels 0, 1, . . ., n, minimum support threshold *min_sup*, taxonomy Γ, maximum history size $history\_size$
Output: set of generalized and not generalized itemsets I=$\{I_l\}$ $1 \leq l \leq n$

1: $i = 1, k = 1, I = \emptyset$
2: **for all** D_i in D **do**
3: $C_k = \emptyset$ // set of (generalized) k-itemsests from D_i
4: add in C_1 the set of 1-itemsets from D_i
5: **repeat**
6: scan D_i and count the support $sup(c, D_i)$ $\forall$ $c \in C_k$
7: $Gen_l = \emptyset$ // level-l generalized itemset container
8: **for all** c in C_k **do**
9: **if** $sup(c, D_i) < min\_sup$ and $sup(c, D_k) > min\_sup$ $\forall$ k $|k \geq 1$ and $i - history\_size \leq k \leq i - 1$ **then**
10: $gen(c)$ = set of new generalizations of itemset c
11: $gen(c)$ = taxonomy_evaluation(Θ, c)
12: $Gen = Gen \cup gen(c)$
13: **end if**
14: **end for**
15: **if** $Gen \neq \emptyset$ **then**
16: count support in D_i for each itemset $gen(c) \in Gen$
17: **end if**
18: $I_k = \{$ itemsets in $\{C_k \cup Gen\}$ whose support $\geq$ *min_sup* $\}$
19: $k = k + 1$
20: C_{k+1} = candidate_generation(I_k)
21: **until** $C_k = \emptyset$
22: **end for**
23: **return** I

Finally, their support is computed by performing a dataset scan (line 16). The EGP MINER algorithm ends the mining loop on each $D_i \in D$ when the set of candidate itemsets is empty (line 21).

3.3.3 The EGP MINER rule generation

This step generates the generalized association rules satisfying both the minimum support threshold min_sup and the minimum confidence threshold min_conf. Since the confidence of rules generated from the same itemset has the anti-monotone property, candidate rules of length k are generated by merging two $(k-1)$-length rules that share the same prefix in the rule consequent [1]. Discovered rules are sorted based on confidence and support quality indexes to better support in-depth analysis. However, the TweM framework allows easily integrating other quality indexes as well (e.g., lift [34]).

The domain expert may query the ordered rule set based on either their schema or content, i.e., the attributes and/or the items to appear in the rule body or head. An example of query based on the rule schema is: $\{(Keyword, *)\} \rightarrow \{(Place, *)\}$. It selects all 2-length rules that include, respectively, an item characterized by attribute $Keyword$ in the rule body and attribute $Place$ in the rule head. For instance, the generalized rule $\{(Keyword, Sport)\} \rightarrow \{(Place, U.S.)\}$ satisfies the requested schema. Differently, an example of query on the rule content is: $\{*\} \rightarrow \{(Place, U.S.)\}$. It selects all rules that contain the item $(Place, U.S)$ as rule consequent. Rule $\{(Keyword, Sport)\} \rightarrow \{(Place, U.S.)\}$ satisfies also the item constraint.

4 Experimental results

We evaluated the TweM framework by means of a large set of experiments by addressing the following issues: (i) the usefulness of the mined generalized rules in different examples of use cases (see Section 4.1), (ii) the performance of the TweM framework (see Section 4.2), and (iii) the characteristics of the aggregation rules (see Section 4.3). The experiments have been performed on 3.0 GHz Pentium IV system with 4 GB RAM, running Ubuntu kernel 2.6.

4.1 Examples of TweM use-cases

In this section, we evaluate the effectiveness of the TweM framework in discovering valuable correlations among tweets in different real-life use-cases. In each scenario the analyst may perform only the selection of an initial set of tweets of major interest (e.g., the top-tweets) and the definition of the constraints to categorize the tweets in different collections. The TweM framework fully supports the following steps: (i) generation of taxonomies over textual and context data, and (ii) generalized association rule mining from an analyst-provided sequence of tweet sets. To test our framework in real application scenarios we exploited a tweet crawler to retrieve and categorize the tweets based on the constraints defined by the analyst. Some examples of real-life use-cases follows.

Use-case 1: Content propagation analysis. This application scenario allows analysts to discover most significant spatial and temporal correlations about knowledge propagation in Twitter. For instance, starting from the top-tweets, which are subsets of tweets of major interest, the analyst may categorize the tweets retrieved by the crawler based on their propagation level in the chain of answers/citations to the initial set. Then, the EGP MINER algorithm is exploited to provide to the analysts a set of potentially meaningful implications, in the form of generalized association rules. Finally, the analyst can query the rules based on either their schema or content (e.g., $(Keyword, *) \rightarrow (Place, *)$).

Use-case 2: Spatial correlation analysis in message posting. This application scenario allows analysts to discover most relevant recurrences hidden in tweets posted from a delimited geographical area. For instance, correlations among tweets collected in faraway places may highlight social, political, or economical linkages. The order in which tweet collections are analyzed by EGP MINER should reflect the expected way of knowledge propagation. For instance, if some topical news are matter of contention in the U.S.A., it may be worth investigating their spatial propagation from the U.S.A. to the other countries.

Use-case 3: Temporal evolution analysis. The third application scenario analyzes the temporal tweet propagation by considering the time stamp at which messages are posted. The analyst may partition tweets in distinct collections using the time stamp information (e.g., 1-day time period). The discovered rule may represent unexpected trends in the evolution of relevant tweet topics. For instance, analysts may wonder how breaking news are matter of contention on Twitter in consecutive days. The achieved results strictly depend on the granularity of the selected time periods.

4.1.1 *Examples of extracted rules*

In this section, we report some examples of mined generalized association rules relative to the second and the third TweM use-cases. To investigate both spatial and temporal knowledge propagation in Twitter posting, we crawled, by means of Twitter APIs, two relational tweet sets (Cf. Definition 1). The first dataset collects tweets posted within a $2,500$ km radius far from New York (i.e., the lands along the Eastern American coastline), while the second one includes messages posted within a $2,500$ km radius far from London (i.e., the North-West of Europe). The tweet submission dates are uniformly distributed in the time period [2011/03/20-2011/03/24]. Consider, for instance, the spatial evolution of the content published in tweets posted in the U.S.A. Extracted rules, among which the ones below have been selected, highlight that both American and English users are interested in the recent conflict in Libya.

Relational tweet set $NewYork$

(i) $(Keyword_1, Obama)$, $(Place, Washington,\ D.C.) \rightarrow \{(Date,\ 2011/03/22)\}$ $(sup = 3.6\%, conf = 100\%)$

(ii) $(Keyword_1, Congress)$, $(Place, Washington,\ D.C.) \rightarrow \{(Date, 2011/03/22)\}$ $(sup = 2.2\%, conf = 97\%)$

Relational tweet set $London$

(iii) $(Keyword_1, Obama) \rightarrow \{(Keyword_2, Libya)\}$ $(sup = 3.6\%, conf = 94\%)$

(iv) $(Keyword_1, Obama) \rightarrow \{(Place, United\ Kingdom)\}$ $(sup = 3.5\%, conf = 82\%)$

Rules have been extracted by enforcing a minimum support threshold equal to 1% and a minimum confidence threshold equal to 80%. A particular attention is paid to the foreign policy undertaken by president Obama and the American Congress, which has been under discussion in the past meeting held in the United States Capitol Washington, D.C. (USA) on March, 22^{nd} 2011. To delve into the impact of breaking news coming from the United States Capitol in the very next days, we reorganized and sorted crawled tweets in order of submission date. By setting $history\_size$=1, the EGP MINER algorithm is exploited to perform generalized rule mining from tweet sets posted in consecutive days. Pattern generalization prevents the discarding of knowledge that is expected to be of analyst's interest, like the one reported the following example.

Relational tweet set March, 22^{nd}

(v) $\{(Keyword_1, Obama)$,$(Keyword2, Libya)\} \rightarrow \{(Place, Washington,\ D.C.)\}$ $(sup = 1.3\%, conf = 100\%)$

Relational tweet set March, 23^{rd}

(vi) $\{(Keyword_1, Obama)$,$(Keyword_2, Libya)\} \rightarrow \{(Place, U.S.A.)\}$ $(sup = 2.5\%, conf = 100\%)$

Keywords $Obama$ and $Libya$, which have been frequently posted on March, 22^{nd} in Washington, D.C. due to the Congress meeting, become infrequent, in the same place, the day after. However, the lazy generalization adopted by EGP MINER allows figuring out that the same topic is still of interest in the U.S. country.

4.2 TweM performance evaluation

We evaluated the performance of the TweM framework by addressing the following issues: (i) the number of generalized and not generalized itemsets extracted by the EGP MINER algorithm (Section 4.2.1), and (ii) the scalability, in terms of the extraction time, of both the taxonomy generation procedure and the generalized association rule mining steps (Section 4.2.2).

4.2.1 EGP MINER algorithm performance

We analyzed the performance of the EGP MINER algorithm, in terms of the number of extracted itemsets, by comparing it with our implementation of the two following generalized frequent itemset miners: (i) Cumulate [33] and (ii) GENIO[3]. Unlike EGP MINER, they do not consider the history of the previously extracted patterns to drive the generalization process. While Cumulate performs an exhaustive taxonomy evaluation by generating all the possible frequent combinations of generalized and not generalized itemsets, GENIO generates a higher level itemset only if it has at least an infrequent descendant in the current mining session. The proposed EGP MINER algorithm generalizes the GENIO algorithm in a dynamic context by considering eligible for generalization exclusively the itemsets that are infrequent in a given time period k but frequent all the previous $history\_size$ ones $[k - history\_size, \ldots, k - 1]$.

We evaluated the pruning selectivity of the EGP MINER algorithm in terms of the number of generated itemsets on synthetic datasets generated by means of the TPC-H generator [37]. The TPC-H data generator consists of a suite of business-oriented ad-hoc queries. The queries and the data populating the database have been chosen to have broad industry-wide relevance. By varying the scale factor parameter, files with different sizes could be generated. We generated a dataset starting from the line item table by setting a scale factor equal to 0.075 (i.e., around 450,000 records). To partition the whole dataset in three distinct time-related data collections, we queried the source data by enforcing different constraints on the shipping date value (attribute $ShipDate$). More specifically, we partitioned line items shipped in the three following time periods: $[1992-01-01, 1994-02-31]$, $[1994-03-01, 1996-05-31]$ $[1996-06-01, 1998-12-01]$. For the sake of brevity, we will denote the corresponding datasets as data-1, data-2, and data-3 in the rest of this section.

Since the minimum support threshold enforced during the itemset mining step significantly affects the number of extracted itemsets and rules, we performed different mining sessions, by varying the minimum support threshold, for all combinations of algorithms and datasets. In Figures 6, 7, and 8 we plotted the number itemsets mined, respectively, from data-1, data-2, and data-3. To better highlight the pruning selectivity on generalized itemsets, we differentiated generalized from not generalized itemsets.

To test the EGP MINER algorithm, we considered the generated datasets in increasing order of shipment date. Since data-1 represents the first collection of the sequence, results obtained by the EGP MINER algorithm and GENIO are the same (see Figure 6(b)). For all the tested algorithms and datasets and for most of the minimum support values, the percentage of extracted frequent itemsets including at least one generalized item is significant (i.e., at least 65%). For both Cumulate and GENIO, the number of mined (generalized) itemsets significantly increases for lower minimum support values (e.g., 1%). Hence, it may become difficult to look into the extracted patterns. Moreover, most of the discovered patterns neither represent relevant knowledge nor highlight a significant trend in the sequence of tweet collections. In GENIO, infrequent items are aggregated during the extraction process regardless of the past mining results thus the percentage of generalized itemsets increases when higher support threshold are enforced. When high support thresholds (e.g., 7%) are enforced, most of the extracted itemsets are generalized itemsets whose dataset coverage is too wide to provide interesting knowledge. Oppositely, when lower support thresholds are enforced (e.g., 2%), the mining algorithm generates a larger amount of patterns, possibly including a subset of itemsets of interest.

The EGP MINER algorithm generalizes the infrequent itemsets that have been frequent in the previous $history\_size$ mining steps. Figures 7(c) and 8(c) show that the proposed approach to itemset generalization significantly reduces the number of generated generalized itemsets (e.g., 6% reduction with respect to GENIO at $minsup$=1% on data-3). Figures 9(a) and 9(b) show the selectivity of the minimum support threshold on the set of not generalized itemsets extracted by EGP MINER, respectively, from data-1 and data-2. In particular, Figures 9(a) and 9(b) report the number of not generalized itemsets mined, respectively, from data-0 and data-1 that became infrequent in the next mining steps at different minimum support values. The above itemsets are the ones over which the EGP MINER itemset lazily triggers the generalization process. By setting $history\_size$ = 2, only those infrequent itemsets in data-3 that are frequent in both data-1 and data-2 are generalized. Thus, the pruning effectiveness is maximal and the amount of generated generalization becomes significant only when lower support thresholds are enforced (e.g., 1%). Nevertheless, by setting $history\_size$ = 1 a significant amount of generalized itemsets have been extracted even at medium support thresholds. In summary, the lower are the values of the history set size $history\_size$ and the minimum support threshold min_sup the more significant is the impact of the generalization process.

4.2.2 TweM extraction time

We also analyzed, on synthetic datasets, the time spent by TweM in each mining phase, i.e., taxonomy generation, frequent generalized itemset mining, and generalized association rule generation. To perform the analysis, we exploited the same TPC-H lineitem tables data-1, data-2, and data-3 presented in the previous section.

In Figure 10, we compared the time spent by the EGP MINER algorithm in generalized itemset mining with the one spent by Cumulate and GENIO, by varying the minimum support threshold value. The extraction time is mainly affected by the generalization process in all the three algorithms. The impact of generalization in EGP

MINER significantly decreases with respect to Cumulate and GENIO when lower minimum support values are enforced (e.g., 1%). This effect is partially counteracted by the higher time spent in checking whether itemsets are eligible or not for generalization (i.e., the history set look back). This makes the extraction time comparable to GENIO when higher support threshold values are enforced.

We also analyzed the impact of the TweM taxonomy and rule generation procedures separately. As expected, most of of the TweM extraction time (from 70% to 90%) is spent in the generalized frequent itemset mining phase. For all datasets and for any combination of support and confidence thresholds, the rule generation step never takes more than 7% of the whole extraction time. Time spent in keyword taxonomy generation ranges from 5% to 20% of the whole extraction time and its impact is mostly due to the stemming algorithm processing time. For instance, by considering the TPC-H dataset composed of around $450,000$ records, the stemming algorithm takes around 267 seconds, the association rule graph creation (by setting minimum support threshold 1% and minimum confidence 50%) takes 1.5 seconds, while the graph pruning phase takes around 0.038 seconds.

4.3 Characteristics of the aggregation rules

In this section we analyzed the characteristics of the rules used to aggregate tweet keywords into higher level ones and their suitability to drive the generalization process. To evaluate the quality of the discovered aggregations, we compared the aggregation rules selected by our approach with the ones selected by a recently proposed approach [17] that also adopts association discovery to perform taxonomy generation in a different context (i.e., Web tag analysis).

We retrieved, by means of Twitter APIs, two examples of tweet collections concerning two very famous persons, i.e., the president of the United States of America, Barack Obama, and the theoretical physicist, Albert Einstein. To perform a fair comparison, we exploited the same quality index used in [17]. In particular, we evaluated the local quality index, first defined in [17], that measures the average significance, in terms of the confidence quality index, of the selected rule set. Figures 11(a) and 12(a) plot the values of local quality for both datasets by varying the minimum confidence threshold. The quality of the rules selected by our approach is slightly worser than that achieved by the ones selected in [17] at lower confidence thresholds, while the quality gap disappears when higher confidence thresholds are enforced. Nevertheless, the selected aggregations are better spread across the taxonomy levels. We define the spread as the average number of hops across the taxonomy Γ to move from an arbitrary non-root node $t_i \in V$ to its corresponding root $t_j \in R$:

$$Spread(\Gamma) = \frac{\sum_{t_i \in (V \setminus R)} hops(t_i)}{|V \setminus R|} \tag{2}$$

In Figures 11(b) and 12(b) we plot the local quality in terms of its corresponding spread value by setting any confidence threshold. Our approach yields a significant improvement, in terms of local quality, at higher spread values. Thus, aggregation rules discovered by TweM are almost as accurate as the ones selected in [17] and

better spread across the different abstraction levels. Indeed, they may deemed more suitable for being exploited to drive the rule generalization process.

5 Conclusions and future work

This paper presents the TweM framework that focuses on discovering hidden and high level correlations among Twitter user-generated content. It investigates the use of taxonomies to drive the association rule mining process from a sequence of tweet collections. Experimental results show both the effectiveness and the efficiency of the TweM framework in performing knowledge discovery from Twitter user-generated content. The system is proved to be very effective in supporting the analysts in the discovery of the most notable temporal and spatial topic trends. Monitoring the evolution of the user-generated content and its related context is crucial for enhancing advanced expert analysis and reactively suiting the decision making process to the actual online community expectations.

We focused our framework on the analysis of the information published by the Twitter community. However, our approach can be successfully applied to any application scenario and domain in which textual and contextual pieces of information are available. As future work, we will scale up the system with the integration of different Web sources like news, blogs, and other social network posts. Furthermore, we will address the incremental updating of both the generated taxonomies and the extracted rules. Finally, the study of novel metrics to evaluate the authoritativeness of a user based on both his posted messages and his relationships with the other users and groups is another interesting future research direction.

References

[1] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In *SIGMOD Conference*, pages 207–216, 1993.

[2] R. Agrawal and G. Psaila. Active data mining. In *Proceedings of the 1st international conference on knowledge discovery and data mining*, pages 3–8, 1995.

[3] E. Baralis, L. Cagliero, T. Cerquitelli, V. D'Elia, and P. Garza. Support driven opportunistic aggregation for generalized itemset extraction. In *5th IEEE International Conference of Intelligent Systems*, pages 102–107, 2010.

[4] Elena Baralis, Luca Cagliero, Tania Cerquitelli, Paolo Garza, and Marco Marchetti. Cas-mine: Providing personalized services in context-aware applications by means of generalized rules. *Knowledge Information System*, 2010.

[5] Steffan Baron, Myra Spiliopoulou, and Oliver Gnther. Efficient monitoring of patterns in data mining environments. In *Advances in Databases and Information Systems*, volume 2798 of *Lecture Notes in Computer Science*, pages 253–265. 2003.

[6] P. Basile, D. Gendarmi, F. Lanubile, and G. Semeraro. Recommending smart tags in a social bookmarking system. In *Bridging the Gep between Semantic Web and Web*, volume 2, pages 22–29, 2007.

[7] M. Bender, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J.X. Parreira, R. Schenkel, and G. Weikum. Exploiting social relations for query expansion and result ranking. In *IEEE 24th International Conference on Data Engineering Workshop*, pages 501 –506, 2008.

[8] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 49–62, 2009.

[9] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, 2009.

[10] A.S. Bozkır, S. Güzin Mazman, and E. Akçapınar Sezer. Identification of user patterns in social networks by data mining techniques: Facebook case. *Technological Convergence and Social Networks in Information Management*, pages 145–153, 2010.

[11] Mirko Bttcher, Detlef Nauck, Dymitr Ruta, and Martin Spott. Towards a framework for change detection in data sets. In *Research and Development in Intelligent Systems XXIII*, pages 115–128. 2007.

[12] Luca Cagliero. Discovering temporal change patterns in the presence of taxonomies. *IEEE Trans. Knowl. Data Eng.*, In press.

[13] W.A.V.B.D. Caragea and W.H. Hsu. Ontology-aware classification and association rule mining for interest and link prediction in social networks. 2009.

[14] T. Cerquitelli, A. Fiori, and A. Grand. Community-contributed media collections: Knowledge at our fingertips. *Community-Built Databases: Research and Development*, page 21, 2011.

[15] M. Cheong and V. Lee. Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base. In *Proceeding of the 2nd ACM workshop on Social web search and mining*, pages 1–8, 2009.

[16] C. Clifton, R. Cooley, and J. Rennie. TopCat: data mining for topic identification in a text corpus. *IEEE Transactions on Knowledge and Data Engineering*, 16(8):949–964, 2004.

[17] B. Cui, J. Yao, G. Cong, and Y. Huang. Evolutionary Taxonomy Construction from Dynamic Tag Space. *Web Information Systems Engineering*, pages 105–119, 2010.

[18] S.C. Gates, W. Teiken, and K.F. Cheng. Taxonomies by the numbers: building high-performance taxonomies. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 568–577, 2005.

[19] L. Guo, E. Tan, S. Chen, X. Zhang, and Y.E. Zhao. Analyzing patterns of user content generation in online social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 369–378, 2009.

[20] J. Han and Y. Fu. Mining multiple-level association rules in large databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(5):798–805, 2002.

[21] P. Heymann and H. Garcia-Molina. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. *Stanford InfoLab Technical Report*, (10), 2006.

[22] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, 2008.

[23] D. Ienco and R. Meo. Towards the Automatic Construction of Conceptual Taxonomies. *Data Warehousing and Knowledge Discovery*, pages 327–336, 2008.

[24] R. Kimball, M. Ross, and R. Merz. *The data warehouse toolkit: the complete guide to dimensional modeling*. Wiley, 2002.

[25] Q. Li, J. Wang, Y.P. Chen, and Z. Lin. User comments for news recommendation in forum-based social media. *Information Sciences*, 2010.

[26] X. Li, L. Guo, and Y.E. Zhao. Tag-based social interest discovery. In *Proceeding of the 17th international conference on World Wide Web*, pages 675–684, 2008.

[27] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Transaction on Knowledge Discovery from Data*, 3:8:1–8:31, 2009.

[28] M. Mathioudakis and N. Koudas. TwitterMonitor: trend detection over the twitter stream. In *Proceedings of the 2010 international conference on Management of data*, pages 1155–1158. ACM, 2010.

[29] I. Pramudiono and M. Kitsuregawa. FP-tax: Tree structure based generalized association rule mining. In *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, page 63, 2004.

[30] C. Schmitz, A. Hotho, R. Jaschke, and G. Stumme. Mining association rules in folksonomies. *Data Science and Classification*, pages 261–270, 2006.

[31] E. Schwarzkopf, D. Heckmann, D. Dengler, and A. Kroner. Mining the structure of tag spaces for user modeling. In *Proceedings of Workshop on Data Mining for User Modeling*, pages 63–75, 2007.

[32] A. Shepitsen, J. Gemmell, B. Mobasher, and R. Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *Proceedings of the ACM conference on Recommender systems*, pages 259–266, 2008.

[33] R. Srikant and R. Agrawal. Mining generalized association rules. In *VLDB*, pages 407–419, 1995.

[34] P.N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 41, 2002.

[35] P.N. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*. Pearson Addison Wesley Boston, 2006.

[36] Yingying Tao and M. Tamer Özsu. Mining frequent itemsets in time-varying data streams. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM, pages 1521–1524, 2009.

[37] TPC-H. The TPC benchmark H. Transaction Processing Performance Council, 2009.

[38] Y. Xue, C. Zhang, C. Zhou, X. Lin, and Q. Li. An Effective News Recommendation in Social Media Based on Users' Preference. In *International Workshop on Education Technology and Training*, volume 1, pages 627–631, 2009.

[39] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 957–966, 2009.

[40] M.J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, et al. New algorithms for fast discovery of association rules. In *Proceddings of the 3rd International Conference on Knowledge Discovery and Data Mining*, volume 20, 1997.

6 Tables

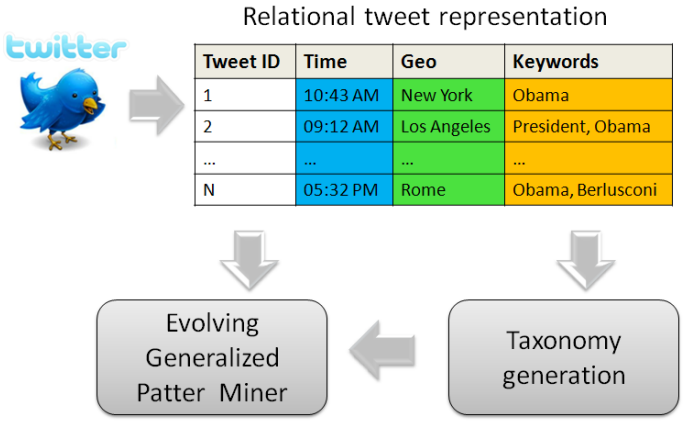
Data feature	**Aggregation function**
Temporal	$Date \Rightarrow WeekDay$ $Date \Rightarrow Month$ $Month \Rightarrow Year$ $Time \Rightarrow Hour$ $Hour \Rightarrow TimeSlot$
Spatial	$GPSCoordinates \Rightarrow Id$ $Id \Rightarrow Place$ $Place \Rightarrow Region$ $Region \Rightarrow State$

Table 1: Aggregation functions used for the taxonomy generation over the temporal and the spatial context features.

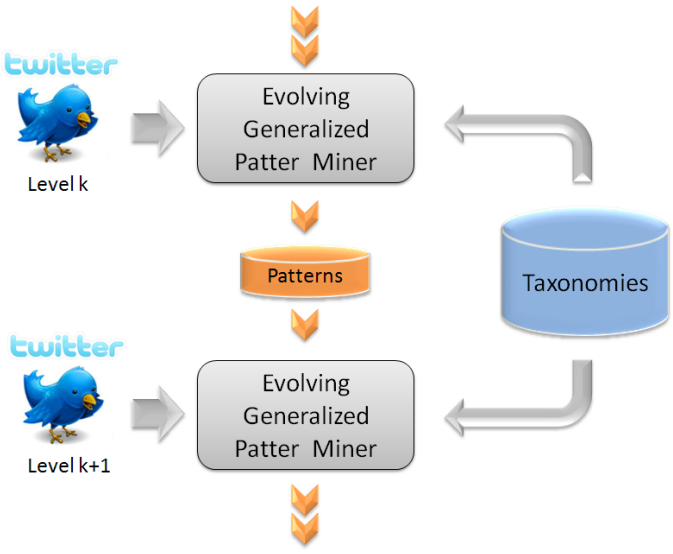
7 Figure captions

- Fig. 1: The TweM framework.
- Fig. 2: A simplified example of tweet set in the JSON data format.
- Fig. 3: A simplified relational tweet set of level 7 generated from two tweets in the JSON data format.
- Fig. 4: A portion of an aggregation tree over the date attribute.
- Fig. 5: An example of association rule graph ARG and the relative taxonomy.
- Fig. 6: Generalized and not generalized itemsets extracted from data-1.
- Fig. 7: Generalized and not generalized itemsets extracted from data-2.
- Fig. 8: Generalized and not generalized itemsets extracted from data-3.
- Fig. 9: History of not generalized itemsets.
- Fig. 10: Extraction time by varying the minimum support threshold.
- Fig. 11: $Einstein$ Dataset. Characteristics of the aggregation rules. Minimum support threshold min_sup=1%.
- Fig. 12: $Obama$ Dataset. Characteristics of the aggregation rules. Minimum support threshold min_sup=1%.

8 Figures



(a) The TweM architecture.



(b) The TweM Evolving Generalized Pattern MIner.

Fig. 1: The TweM framework

```
1 - UserA: [{profile_image_url:..., created_at: Sun, 10 Oct 2010
12:43:31 +0000, from_user:.., metadata: {result_type:recent}, to_user_id: X,
text: This is a message by Obama, id: Y, from_user_id: X, to_user: UserB,
geo:{coordinates:+X -Y id: Z, place: New York City, place_type: city
Country: NY-United States of America}, iso_language_code: en, source..

2 - UserB: [{profile_image_url:..., created_at: Wed, 20 Oct 2010
13:30:12 +0000, from_user:.., metadata:{result_type: recent}, to_user_id: X,
text: This is a post about Clinton, id: X, from_user_id: X, to_user: User2,
geo:{coordinates: +X -Y id: Z, place: Los Angeles, place_type: city
Country: California-United States of America}, iso_language_code:en, source..
```

Fig. 2: A simplified example of tweet set in the JSON data format

TWEETS
((Tweet ID, 1), (Username, UserA), (Place, New York City), (Date, 2010-10-10), (Time, 12:43:31 +000, (Keyword, $Obama$))
((Tweet ID, 2), (Username, UserB), (Place, Los Angeles), (Date, 2010-10-20), (Time, 13:30:12 +000, (Keyword, $Clinton$))

Fig. 3: A simplified relational tweet set of level 7 generated from two tweets in the JSON data format

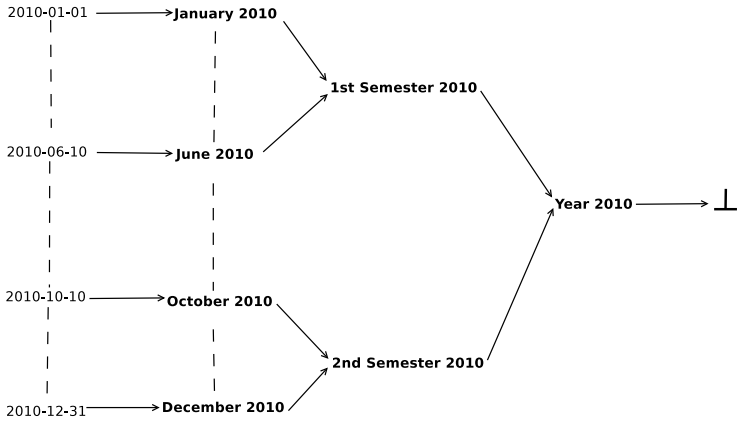
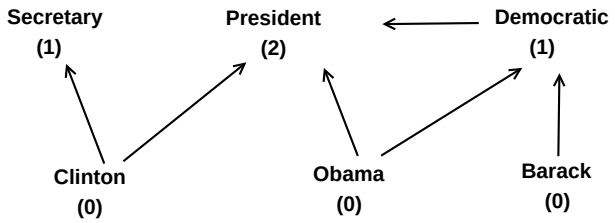
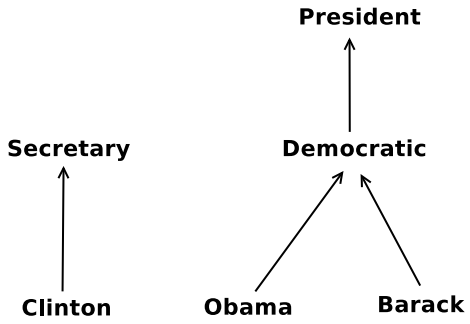


Fig. 4: A portion of an aggregation tree over the date attribute

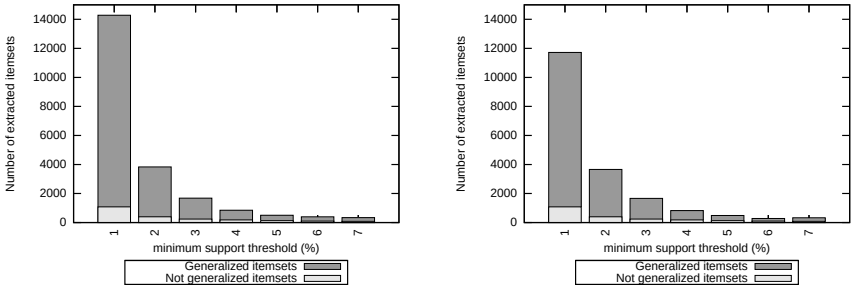


(a) ARG with aggregation levels of each node put in brackets



(b) Generated taxonomy

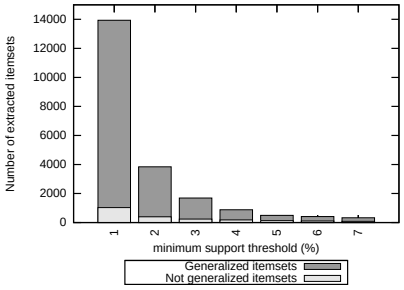
Fig. 5: An example of association rule graph ARG and the relative taxonomy



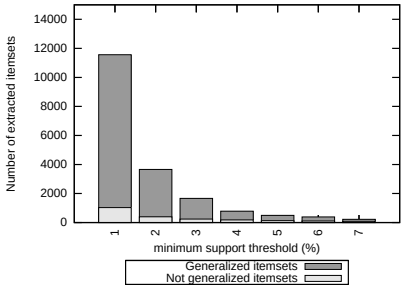
(a) Cumulate

(b) GenIO - EGP MINER

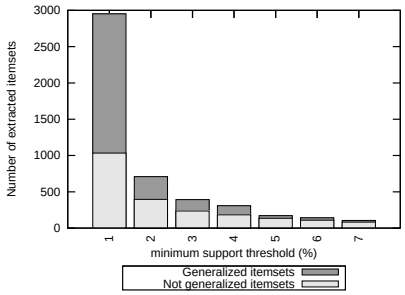
Fig. 6: Generalized and not generalized itemsets extracted from data-1



(a) Cumulate

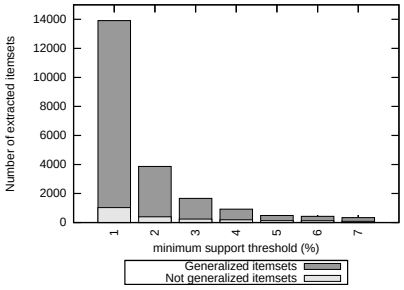


(b) GenIO

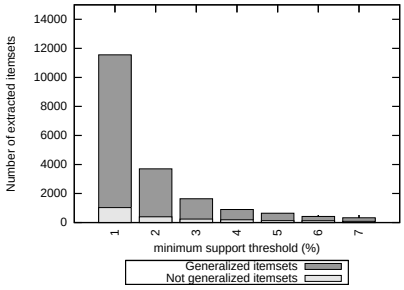


(c) EGP MINER

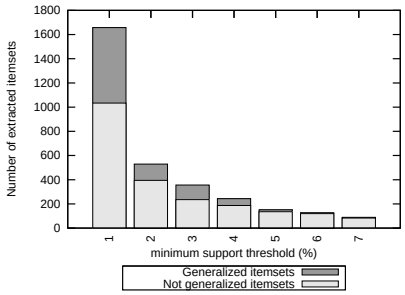
Fig. 7: Generalized and not generalized itemsets extracted from data-2



(a) Cumulate

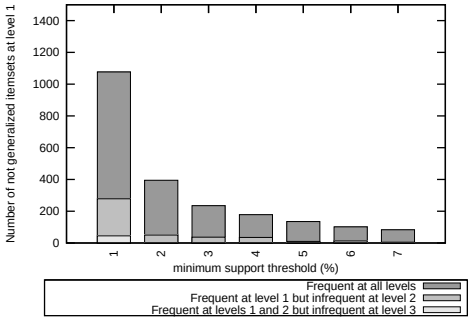


(b) GenIO

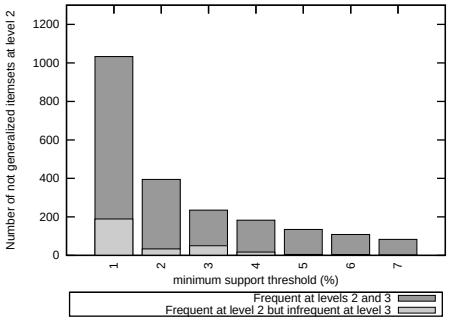


(c) EGP MINER

Fig. 8: Generalized and not generalized itemsets extracted from data-3

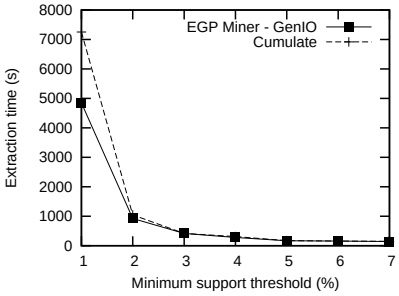


(a) Not generalized itemsets generated from data-1

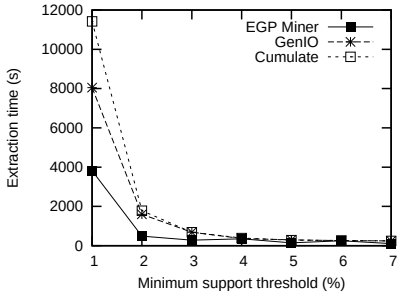


(b) Not generalized itemsets generated from data-2

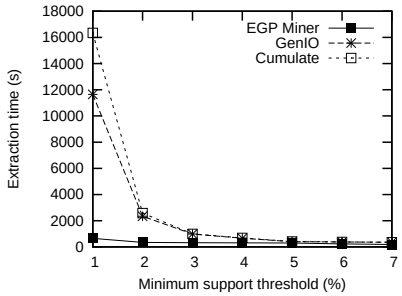
Fig. 9: History of not generalized itemsets



(a) Data-1.

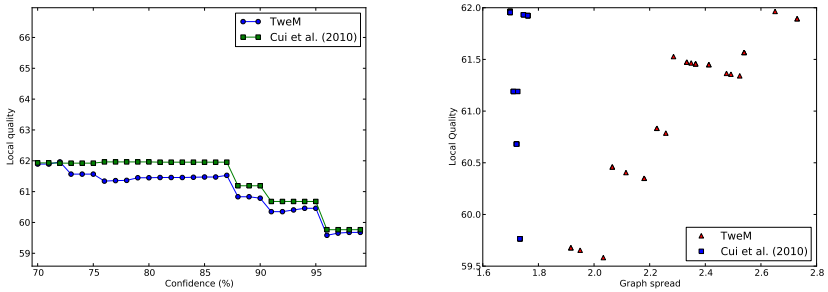


(b) Data-2.



(c) Data-3.

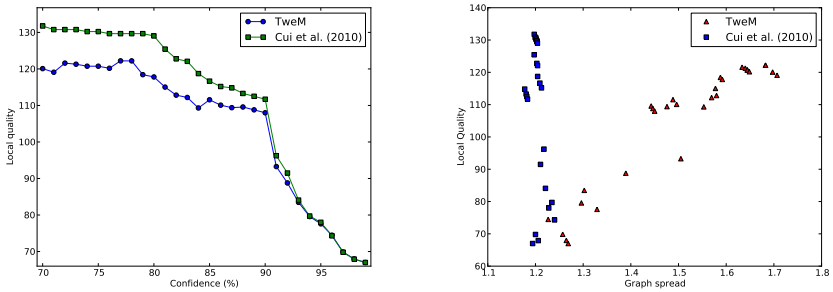
Fig. 10: Extraction time by varying the minimum support threshold.



(a) Local quality.

(b) Local quality vs. spread.

Fig. 11: $Einstein$ Dataset. Characteristics of the aggregation rules. Minimum support threshold min_sup=1%.



(a) Local quality.

(b) Local quality vs. spread.

Fig. 12: *Obama* Dataset. Impact of the minimum confidence threshold on taxonomy quality indexes. Minimum support threshold min_sup=1%.