[Article] Twitter data analysis by means of Strong Flipping Generalized Itemsets

Twitter Data Analysis by means of Strong Flipping Generalized Itemsets

Luca Cagliero*, Tania Cerquitelli, Paolo Garza, Luigi Grimaudo

Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129, Torino, Italy

Abstract

Twitter data has recently been considered to perform a large variety of advanced analysis. Analysis of Twitter data imposes new challenges because the data distribution is intrinsically sparse, due to a large number of messages post every day by using a wide vocabulary. Aimed at addressing this issue, generalized itemsets - sets of items at different abstraction levels - can be effectively mined and used to discover interesting multiple-level correlations among data supplied with taxonomies. Each generalized itemset is characterized by a correlation type (positive, negative, or null) according to the strength of the item correlation.

This paper presents a novel data mining approach to supporting different and interesting targeted analysis - topic trend analysis, context-aware service profiling - by analyzing Twitter posts. We aim at discovering contrasting situations by means of generalized itemsets. Specifically, we focus on comparing itemsets discovered at different abstraction levels and we select large subsets of specific (descendant) itemsets that show correlation type changes with respect to their common ancestor. To this aim, a novel kind of pattern, namely the Strong Flipping Generalized Itemset (SFGI), is extracted from Twitter messages and contextual information supplied with taxonomy hierarchies. Each SFGI consists of a frequent generalized itemset X and the set of its descendants showing a correlation type change with respect to X.

Experiments performed on both real and synthetic datasets demonstrate

*Corresponding author. Tel.: +39 011 090 7084. Fax: +39 011 090 7099.

Email addresses: luca.cagliero@polito.it (Luca Cagliero), tania.cerquitelli@polito.it (Tania Cerquitelli), paolo.garza@polito.it (Paolo Garza), luigi.grimaudo@polito.it (Luigi Grimaudo)

Preprint submitted to

the effectiveness of the proposed approach in discovering interesting and hidden knowledge from Twitter data.

Keywords: Social Network Analysis and Mining, Data Mining and Knowledge Discovery, Generalized Itemset Mining

1. Introduction

In recent years, social networks and online communities have become a powerful source of knowledge. Social network sites, such as Twitter and Facebook, are accessed by millions of people every day and their users usually publish and continuously update multimedia resources, posts, and blogs.

Since actions undertaken by Web users reflect their habits, personal interests, and professional skills, a particular attention has been paid to the analysis of data acquired from Twitter. Although a large body of research addresses social network data mining [11, 12, 14, 26, 31, 32, 34, 45], the potential business impact of mining social data is still largely unexplored. Service providers (e.g., TV channels, radio stations) may explore and analyze Twitter posts to improve service provision according to the knowledge hidden in Twitter data. From a business point of view, it is worth profiling Twitter user trends and message topics to plan targeted promotions or to identify exceptional situations. For example, let us consider a music tour organizer. To plan a singer tour in Italy, it is important to known in which Italian cities and in which time periods people are most likely to attend concerts. The analysis of the Twitter messages posted by Italian users about a given singer may support organizers in planning tours and advertising sessions. For example, they can promote album or songs to specific user segments. Innovative analytics solutions are needed to effectively and efficiently support service profiling and topic trend analysis as well as to discover exceptional situations from large social data collections. However, Twitter data is intrinsically sparse because posts range over many different topics and use a wide vocabulary. To address this issue, a promising research direction is to exploit semantics-based models (e.g., ontologies, taxonomies) to drive the data mining process and to discover interesting correlations among Twitter data at different abstraction levels.

Generalized itemset mining [40] is an exploratory data mining technique that allows us to discover multiple-level correlations among data supplied with an analyst-provided taxonomy. The taxonomy (i.e., a set of is-a hierar-

chies) is used to aggregate low-level data items into higher-level concepts. For example, a city (e.g., Milan) can be generalized as the corresponding country (e.g., Italy). A generalized itemset (e.g., {(Location,Italy),(Day,Working day)}) consists of a set of items that either occur in the source dataset or represent data item generalizations according to the given taxonomy. However, the potentially large set of extracted itemsets could be hardly manageable by domain experts for manual inspection. By comparing itemsets extracted at different abstraction levels we focus the analysis on worthwhile itemset subsets, representing only contrasting or unexpected situations, because in social data analysis targeted actions are often triggered by exceptional or surprising events [20, 33]. Specifically, itemsets can be further evaluated and compared by using established correlation indexes (e.g., Kulc, interest, lift [44]), which indicate the item correlation type (i.e., positive, negative, or null) and strength. If the correlation type of an ancestor (high-level) itemset is in contrast with those of many of its low-level descendant itemsets, then an anomalous situation may come out. For example, if items in {(Location,Italy),(Day,Working day)} are positively correlated with each other, we expect that itemsets such as {(Location,Milan),(Day,Working day)} have the same correlation type. Otherwise, the comparison between the two itemsets can be worth investigating more in detail. Even though itemsets with contrasting correlation have already been studied in [5], to the best of our knowledge an approach to mining *large groups* of itemsets in contrast with a common ancestor in terms of correlation type has never been proposed so far.

This paper presents the TFC Analyzer (Twitter Flipping Correlation Analyzer) system to support different and interesting targeted analysis, i.e., topic trend analysis, context-aware service profiling, outlier detection. It aims at analyzing Twitter posts to discover subsets of frequent generalized itemsets that potentially represent contrasting situations. Given the Twitter posts (i.e., the tweets) enriched with their publication context (i.e., date, time, place), a novel kind of pattern, namely the Strong Flipping Generalized Itemset (SFGI), is mined. SFGIs are patterns in the form $X \sim \Psi$, where X is a frequent generalized itemset having a *large* set Ψ of low-level exceptions, i.e., frequent descendant itemsets of X whose correlation type changes with respect to X. The existence of a large group of contrasting low-level itemsets may indicate the presence of an unexpected situation in Twitter data. To extract all SFGIs whose number of contrasting low-level correlations is equal to or exceeds a given (analyst-provided) threshold, TFC Analyzer exploits

Table 1: Example dataset $\mathcal{D}$.

Location	Day
Milan	Working day
Milan	Working day
Turin	Working day
Trento	Working day
Naples	High day

an efficient LCM-based (Linear Time Closed Itemset Miner-based) itemset mining algorithm combined with an ad-hoc post-pruning phase.

Experiments performed on real-life data coming from Twitter demonstrate the effectiveness of the proposed system in discovering interesting knowledge. Furthermore, the performance and scalability of the adopted mining strategy have been evaluated on real and synthetic datasets.

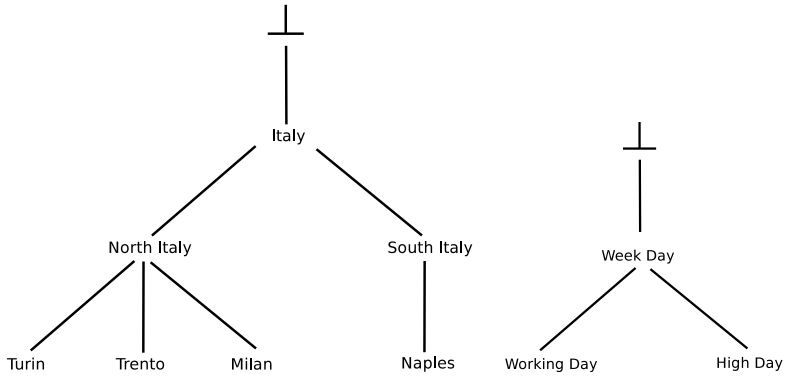
Even though this work focuses on Twitter data analysis, it is worth mentioning that the proposed patterns can be successfully mined and exploited to support knowledge discovery from data coming from different contexts.

This paper is organized as follows. Section 2 presents a motivating example. Section 3 compares our work with related approaches. Section 4 thoroughly describes the characteristics of the TFC Analyzer system, while Section 5 describes the experiments performed. Section 6 discusses the extension of TFC Analyzer in a distributed environment. Finally, Section 7 draws conclusions and discusses future work.

2. Motivating example

We are interesting in analyzing Twitter data to efficiently support business applications based on social network data mining, e.g., context-aware service profiling. Let us consider the example Twitter dataset $\mathcal{D}$ in Table 1. It consists of 5 Twitter posts. For each post the publication weekday and the city of provenance of the author are available. For the sake of simplicity, in this preliminary example we disregard the textual content of the tweet as well as any other contextual information.

Figure 1 shows an example of taxonomy built on the analyzed data. It generalizes cities as the corresponding region and country, whereas publication days (working or high days) are aggregated into *weekday*. Frequent generalized itemsets are sets of items or generalized items that (i) represent



(a) GH_{location} (b) GH_{day}

Figure 1: Taxonomy built over data items in $\mathcal{D}$

high-level correlations among data and (ii) frequently occur in the dataset, i.e., their support value is above a given threshold [40]. Itemset quality measures (e.g., lift, Kulc, interest [44]) have been proposed to evaluate item correlation strength and type (i.e., positive, negative, or null). Since a potentially large number of itemsets can be extracted, a manual inspection of the result set could be a challenging task. To discover unexpected and potentially interesting situations we focus on subsets of item correlations at different abstraction levels that are in contrast in terms of correlation type.

Table 2 reports the frequent generalized itemsets mined from $\mathcal{D}$ by a traditional approach [40] (see Column 2). The mining task was accomplished by exploiting the taxonomy in Figure 1 and by enforcing an absolute minimum support threshold min_sup equal to 1. For each itemset, the support value (Column 3), the generalization level (Column 4), and the value of an established correlation index, i.e., the Kulczynsky index [44] (hereafter denoted as Kulc), are also reported. By setting the maximum negative and minimum positive Kulc thresholds max_neg_cor and min_pos_cor to 0.7 and 0.8, respectively, itemsets with Kulc between 0.7 and 0.8 are uncorrelated, itemsets with Kulc below 0.7 show negative item correlation, whereas itemsets with Kulc above 0.8 indicate a positive item correlation, i.e., their items co-occur more than expected.

From the comparison between each frequent generalized itemset in Table 2 and its frequent descendants it appears that 3 out of 17 frequent generalized

Table 2: SFGIs extracted from $\mathcal{D}$. min_sup = 1. max_neg_cor= 0.7. min_pos_cor= 0.8. min_except = 1

Id	**Generalized itemset X (Kulc correlation)**	**Sup**	**Level**	**Low level descendant set Ψ**	**SFGI $X \sim \Psi$**
					Mined: X Pruned: -
1	{Turin} (1)	1	1	-	-
2	{Naples} (1)	1	1	-	-
3	{Trento} (1)	1	1	-	-
4	{High Day} (1)	1	1	-	-
5	{Milan} (1)	2	1	-	-
6	{Working Day} (1)	4	1	-	-
7	{Naples,High Day} (1)	1	1	-	-
8	{Turin,Working Day} (0.625)	1	1	-	-
9	{Trento,Working Day} (0.625)	1	1	-	-
10	{Milan,Working Day} (0.75)	2	1	-	-
11	{South Italy} (1)	1	2	{Naples} (1)	-
12	{North Italy} (1)	4	2	{Milan} (1) {Turin} (1) {Trento} (1)	-
13	{Week Day} (1)	5	2	{Working Day} (1) {High Day} (1)	-
14	{South Italy,Week Day} (0.6)	1	2	{Naples,High Day} (1)	X
15	{North Italy,Week Day} (0.9)	4	2	{Turin,Working Day} (0.625) {Milan,Working Day} (0.75) {Trento,Working Day} (0.625)	X
16	{Italy} (1)	5	3	{North Italy} (1) {South Italy} (1)	-
17	{Italy,Week Day} (0.2)	5	3	{North Italy,Working Day} (1) {South Italy,High Day} (1)	X

itemsets have at least one exception (i.e., a low-level descendant itemset with different correlation type). The existence of a large number of exceptions may prompt experts to manually explore these pattern subsets, because items unexpectedly change their correlation type while climbing up or down the taxonomy. A SFGI $X \sim \Psi$ combines a frequent generalized itemset X with the corresponding subset Ψ of frequent low-level exceptions. SFGIs mined by setting the minimum number of exceptions min_except to 1 are marked with X at Column 5 in Table 2. Focusing on the generalized itemsets with at least one exception, itemsets (11), (12), and (13) are not yet considered. Let us consider the positively correlated itemset (15) {North Italy, Week Day} and its corresponding negatively correlated descendant itemsets {Turin, Working Day} and {Trento, Working Day}. Their comparison and analysis could be deemed to be relevant by domain experts for service shaping and maintenance. For example, since Twitter posts submitted from Turin and

Trento from Monday to Friday are less than expected, service maintenance activities in these cities are less likely to cause disruption if scheduled over the working days rather than over the weekend.

3. Related work

A significant research effort has been devoted to analyzing the content and structure of online communities and social networks. For example, in [6, 35] the authors analyze click-stream and blog data to identify most common Web user activities, such as universal searches, message sending, and community creation. Similarly, in [20, 33] the evolution of online communities and the lifetime of their User-Generated Content (UGC) are investigated. Sentiment analysis techniques (e.g., [28]) have also been used to detect attitudes and opinions of social network and online community users. In recent years, the application of data mining techniques to discover relevant social knowledge from the UGC has become an appealing research topic [31, 45, 24]. In [19, 43] a particular attention has been paid to microblogging websites, such as Twitter, which represent rapidly evolving communities. The extraction of hidden associations from Twitter UGC has already been investigated in [11, 14]. Specifically, the authors in [14] discover trend patterns from Twitter data to identify the users who contribute towards the discussions on specific trends. The approach presented in [11] addresses topic trend analysis from Twitter data by exploiting generalized association rules. Unlike [11, 14], this paper focuses on discovering unexpected situations from Twitter data by exploiting a new type of generalized patterns, namely the Strong Flipping Generalized Itemsets.

A parallel research issue is frequent itemset and association rule mining driven by item correlation measures [1, 5, 8, 39]. The frequent itemset and association rule mining problems were first introduced in [2] to discover hidden item correlations from potentially large market basket data. The first attempt to evaluate association rule significance by means of the chi square test for correlation has been made in [8]. To avoid generating all candidate frequent itemsets, the upward closure of the chi square measure is exploited to early prune part of the search space. Since negatively correlated itemsets are usually characterized by a low support value [42], their extraction becomes a challenging task when coping with large and complex datasets. To overcomes this issue, in [1, 39] the authors propose two novel itemset quality measures, namely the collective strength and support expectation, which can be used to

indirectly mine negative associations among data. Unfortunately, since chi square, support expectation, and collective strength are not null-invariant, their value depends on the dataset size. In [44] the authors investigate the use of null-invariant correlation measures [25] in frequent itemset mining. The approach proposed in [5] focuses on exploiting the null-invariant Kulczynsky measure to discover flipping correlations among data supplied with taxonomies. Flipping correlations are itemsets whose correlation type flips from positive to negative (or vice versa) when items are generalized to a higher level of abstraction for *every* generalization step. However, in case many sibling itemsets flip their correlation type with respect to a common ancestor, many different patterns are generated and thus the result could become redundant and hardly readable. Furthermore, in real-world data the itemset correlation type (i) rarely changes at every generalization step and (ii) may also change from correlated to uncorrelated (and not only from positive to negative or vice versa). To overcome these issues, we propose a new type of generalized pattern, namely the Strong Flipping Generalized Itemsets (SFGIs). A SFGI consists of a generalized itemset and its corresponding set of frequent descendant itemsets with contrasting correlation type (positive, negative, or null). SFGIs differ from flipping correlations [5] because (i) they do not consider one itemset per abstraction level, but groups of sibling itemsets with contrasting correlation type with respect to their common ancestor, (ii) they also consider uncorrelated descendants and not only positive or negative item correlations. SFGIs that contain many descendant itemsets with contrasting correlation are selected because they may represent unexpected situations in the analyzed data.

A parallel effort has been devoted to efficiently extracting generalized frequent itemsets and association rules. The first generalized frequent itemset mining algorithm has been proposed in [40] in the context of market basket analysis. It generates itemsets by considering for each item all its parents in the hierarchy. To avoid generating all the possible candidates in the taxonomy the authors in [3, 41] propose to push (analyst-provided) constraints into the mining process. Many algorithm optimizations have also been proposed [4, 13, 21, 29]. For example, the approach presented in [21] proposes an optimization strategy based on a top-down hierarchy traversal. It identifies in advance itemsets that cannot be frequent in a transactional dataset by exploiting the Apriori principle [2]. In [29] the authors propose to mine closed and maximal [36] generalized itemsets. More recently, support-driven approaches to mining generalized itemsets and analyzing their changes over

time have been presented [4, 9, 10]. In [4, 9] only the itemsets infrequent with respect to the support threshold are generalized at a higher level of abstraction, whereas in [10] only the descendant itemsets with different correlation contribute to the support counting of a high-level ancestor. Unlike [10, 4, 9], our approach does not rely on a support-driven generalized pattern selection strategy. Conversely, we discover *large* groups of sibling itemsets with contrasting correlation with respect to their common ancestor.

4. The TFC Analyzer framework

TFC Analyzer (Twitter Flipping Correlation Analyzer) is a new data mining environment to analyze Twitter data with the aim at highlighting unexpected and potentially interesting situations. The main components of TFC Analyzer are described below.

Twitter data collection and preprocessing. The textual content of the Twitter posts (i.e., the tweets) and their publication context (i.e., the publication date and time stamp, the geographical location of the user who posted the tweet) are retrieved through the Twitter Stream APIs (Application Programming Interfaces). Twitter data are first preprocessed to make them suitable for the subsequent mining steps and then integrated into a common transactional data repository to perform off-line data analysis. The preprocessed dataset contains a set of transactions, where each transaction corresponds to a different tweet and contains both textual content and contextual feature values.

Expert-driven taxonomy generation. A taxonomy (i.e., a set of is-a hierarchies) is generated over the analyzed Twitter data with the help of domain expert. A semi-automatic procedure based on an established lexical databases (i.e., WordNet [15]) is used to aggregate tweet words into higher-level concepts, whereas analyst-provided aggregation functions (possibly relying on external database queries) are used to generalize spatial and temporal contextual data.

Strong Flipping Generalized Itemset Miner. Given a Twitter dataset and the corresponding taxonomy, a novel kind of pattern, namely the Strong Flipping Generalized Itemsets (SFGI), is extracted. SFGIs are worth being manually explored by domain experts because they represent unexpected and potentially interesting situations in Twitter data.

A more thorough description of the TFC Analyzer components is given in the following sections.

4.1. Twitter data collection and preprocessing

Tweets are short, user-generated, textual messages of at most 140 characters long and publicly visible by default. Beyond the textual content, some additional tweet features describing the context of publication of the tweet (i.e., the GPS coordinates of the user who posted the tweet, the tweet publication date and time stamp) are commonly available.

This component performs the retrieval of data published on the Twitter microblogging website (*http://twitter.com*). To make data suitable for the subsequent (offline) data mining analysis a preprocessing step is applied to raw data. The textual content of each tweet and its corresponding contextual information are retrieved through the Stream Application Programming Interfaces (APIs). Data is gathered, prior to tweet deletion, by establishing and maintaining a continuous connection with the stream endpoint. Twitter data is retrieved in the JSON (JavaScript Object Notation) format, which is a text-based open standard for data exchange. A simplified example of two Twitter messages in the JSON format is reported in Figure 2.

```
TWEET ID 1
UserB: [{profile_image_url:..., created_at: Mon, 01 Oct 2012
13:30:12 +0000, from_user:.., metadata:{result_type: recent}, to_user_id: X,
text: Data mining enables knowledge discovery, 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..

TWEET ID 2
UserA: [{profile_image_url:..., created_at: Tue, 30 Oct 2012
11:43:31 +0000, from_user:.., metadata: {result_type:recent}, to_user_id: X,
text: Data analysis a complex task!, 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..
```

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

To suit the raw textual data to the following mining process, some preliminary data cleaning and processing steps are applied. Specifically, textual messages are preprocessed by eliminating stopwords, numbers, links, non-ascii characters, mentions, and replies (not only the @ sign), and reducing words to their lemmas by using the WordNet lexical database [7].

After preprocessing, textual words and contextual data are both mapped to distinct data items, whose formal definition follows.

Definition 4.1. Item. *Let TW be a tweet set. An item i is a literal expression occurring in $tw_j \in TW$. It is expressed as a pair (feature:value), where $feature$ denotes a tweet feature (e.g., text, location, time, date, day) while value its corresponding value in tw_j.*

Hereafter we will consider the following subset of tweet features, which can be directly read from the JSON tweet representation: (i) **Text**: textual content, (ii) **Location**: GPS coordinates, (iii) **Time**: publication time stamp, (iv) **Date**: publication date, (v) **Day**: publication weekday.

Examples of items occurring in tweet 1 of Figure 2 are *(Time:1.30 p.m.)*, *(Text:Data)*, *(Text:Knowledge)*, and *(Day:Monday)*.

To perform the subsequent data mining analysis we tailor Twitter data to the transactional data format. To this aim, for each tweet we generate the corresponding transaction. Each transaction consists of a set of items related to both textual content and contextual features. To avoid item repetitions within each transaction the stemmed words occurring in the tweet content are mapped to a set of distinct data items (i.e., words occurring many times in the same tweet are represented just once). Note that word repetitions are very unlikely because tweets are relatively short messages. A more formal definition of the transactional data model is reported below.

Definition 4.2. Transactional Twitter dataset. *Let TW be a tweet set. The transactional Twitter dataset $\mathcal{T}$ associated with TW is a set of transactions t_j, one for each tweet $tw_j \in TW$. Each transaction t_j=$\{item_1, item_2, \ldots, item_k\}$ is a set of distinct items related to tw_j.*

For example, Figure 3 shows the tweets of the transactional Twitter dataset generated from the example in Figure 2.

TWEETS
Tweet ID 1
(Text:Data), (Text:Mining), (Text:Enable), (Text:Knowledge), (Text:Discovery), (Location:Los Angeles), (Date:2012-30-01), (Day:Monday), (Time:1.30 p.m.)
Tweet ID 2
(Text:Data), (Text:Analysis), (Text:Complex), (Text:Task), (Location:New York), (Date:2012-10-30), (Day:Tuesday), (Time:11.43 a.m.)

Figure 3: An example of transactional Twitter dataset

4.2. Expert-driven taxonomy generation

Given a transactional Twitter dataset $\mathcal{T}$, this component accomplishes the task of generating a taxonomy Θ over items in $\mathcal{T}$ with the help of a domain expert. A taxonomy Θ is a set of generalization hierarchies. Each generalization hierarchy is a tree-based structure whose leaf nodes are items in $\mathcal{T}$, whereas upper-level nodes, named *generalized items*, aggregate items in $\mathcal{T}$ into higher-level concepts.

For example, recalling the running example in Figure 3, *(Time:1.30 p.m.)* is an example of taxonomy leaf, whereas *(Time:From 1 p.m. to 3 p.m.)* is an example of generalized item which aggregates time stamps into the corresponding time slot.

Each generalization hierarchy refers to a specific Twitter data feature. For example, a generalization hierarchy may aggregate specific textual terms (e.g., *(Text:Mining)*) into higher-level concepts (e.g., *(Text:Process)*), whereas another hierarchy can aggregate specific geographical locations (*(Location: Value)*) into the corresponding city, region, and country. Note that some spatial aggregations are already indicated in the JSON tweet format. For example, for each pair of GPS coordinates in Figure 2 the corresponding city and country are also available (see tags *Place*, *Place_type* and *Country*).

To extend the set of available data aggregations and generate a complete taxonomy over the transactional Twitter dataset, TFC Analyzer exploits a semi-automatic, expert-driven approach. According to the type of data item considered, two different strategies have been adopted:

Domain-specific database querying. The first strategy entails defining aggregation functions which rely on semantics-based models, such as controlled vocabularies, lexical or domain-specific databases. For example, an aggregation function could access a geographical database (e.g., through the Google Maps APIs available at https://developers.google.com/maps/) to derive from a pair of GPS coordinates the corresponding city, region, and country. Similarly, the WordNet lexical database [15] could be queried to retrieve the most relevant semantic relationships between lemmas. More specifically, we derive hyponyms (i.e., is-a-subtype-of relationships) from WordNet to generalize lemmas as higher-level concepts. For example, according to the WordNet relationship <dog> is-a-subtype-of <domestic animal> we generalize item *(Text:Dog)* as *(Text:Domestic animal)*. The generalization process is iterated to find higher-level aggregations, e.g., according to the WordNet relationship <domestic animal> is-a-subtype-of <animal> we further generalize *(Text:Domestic animal)* as *(Text:Animal)*).

Data warehousing-like hierarchy generation. Hierarchy definition is a standard and established step of the data warehouse design process [27]. Specifically, hierarchies are usually defined on temporal data by exploiting predefined hierarchies or aggregation functions which extract *is-a relationships* parsing item values. For example, by aggregating dates into the corresponding month item *(Date:2012-10-30)* can be simply parsed and generalized as *(Date:2012-10)*. Items and generalized items may be also aggregated according to different data facets. For example, months may be further aggregated into the corresponding trimester or 4-month time period. Similar ad-hoc aggregation functions are usually exploited to generate hierarchies on non-temporal data (e.g., products may be generalized as the corresponding category or brand, etc).

In general, a taxonomy may include items that belong to many generalization hierarchies. For the sake of simplicity, in the following we only consider taxonomies in which items belong to at most one generalization hierarchy.

4.3. Strong Flipping Generalized Itemset mining

This component aims at analyzing Twitter data to discover the Strong Flipping Generalized Itemsets (SFGIs).

This section is organized as follows. Section 4.3.1 introduces some preliminary and well-known concepts related to the frequent generalized itemset mining problem. Section 4.3.2 formally states the SFGI mining problem, while Section 4.3.3 thoroughly describes the SFGI mining process adopted in TFC Analyzer.

4.3.1. Preliminary concepts and notation

Given a transactional Twitter dataset $\mathcal{T}$, a taxonomy Θ, and a minimum support threshold min_sup, the frequent generalized itemset mining problem, first introduced in [40], entails discovering all the frequent generalized itemsets from $\mathcal{T}$.

An itemset [2] is defined as a set of dataset items. For example, {*(Location: Los Angeles)*, *(Date:2012-01-30)*} is an example of itemset occurring in the running example (see Figure 3). A taxonomy is used to generalize items at a higher level of abstraction and discover generalized (high-level) itemsets. For example, if *(Location:U.S.A)* and *(Date:2012-01)* are ancestors of *(Location:Los Angeles)* and *(Date:2012-01-30)* with respect to a given taxonomy, then {*(Location:U.S.A)*, *(Date:2012-01)*} is a generalized itemset.

Generalized itemsets are characterized by several remarkable properties, which are briefly outlined below.

Level-sharing itemset. The level of a generalized item i_j with respect to a given taxonomy is defined as the height of the subtree rooted in i_j. Generalized itemsets whose items all have the same level are called *level-sharing itemsets* [21]. The level of a level-sharing itemset with respect to a given taxonomy corresponds to the one of its items. Similar to [21], in the work we focus on this generalized itemset subset.

To mine level-sharing itemsets, analysts have to generate balanced taxonomy hierarchies (i.e., all the hierarchies in the taxonomy must have the same height). Since the semi-automatic taxonomy generation procedure described in Section 4.2 could generated unbalanced hierarchies, we re-balance taxonomy hierarchies before itemset mining by means of the following procedure: given a taxonomy with maximal aggregation tree height H_{max}, for each aggregation tree with height $H < H_{max}$ we perform a depth-first visit. For each tree leaf node with depth less than H_{max} we added several copies of the item with highest level sharing the same branch, say i, as ancestors of i until depth H_{max} is reached. As discussed in [5], the aforementioned procedure is established in level-sharing itemset mining.

Descent relationship. A generalized k-itemset I_1 (i.e., a generalized itemset composed of k distinct items) is a descent of another k-itemset (I_2) if for every item $i_j \in I_1$ there exists an item $i_k \in I_2$ such that either i_j=i_k or i_j is a descendant of i_k with respect to the given taxonomy.

Coverage and support. A generalized itemset is said to *cover* a given dataset transaction $t_i \in \mathcal{T}$ if all its generalized items are either contained in t_i or ancestors of items in t_i. The support of a generalized itemset I in $\mathcal{T}$ is defined as the ratio between the number of transactions in $\mathcal{T}$ covered by I and the total number of transactions in $\mathcal{T}$. A generalized itemset whose support is equal to or above a given threshold min_sup is said to be *frequent*. For example, itemset {*(Location:U.S.A.)*, *(Date:2012-01)*} has support $\frac{1}{2}$ in the dataset in Figure 3, because it covers only the first transaction.

Correlation. Correlation measures are used to evaluate the strength of the correlation between the items contained in a given itemset. The Kulczynsky (Kulc) correlation measure of a generalized k-itemset I is defined as follows:

$$\text{kulc(I)} = \frac{1}{\text{k}} \sum_{j=1}^{\text{k}} \frac{sup(I, \mathcal{T})}{sup(i_j, \mathcal{T})} \tag{1}$$

where $sup(I, \mathcal{T})$ is the support of I in $\mathcal{T}$ and i_j $[1 \leq j \leq k]$ is the j-th item in I.
As follows from Equation 1, Kulc index values range from 0 to 1. Unlike several other existing itemset correlation measures (e.g., lift, interest [42]), Kulc has the null (transaction)-invariant property, which implies that the correlation measure is independent from the dataset size [44].

By properly setting a maximum negative and a minimum positive Kulc thresholds, hereafter called *max_neg_cor* and *min_pos_cor*, generalized itemsets may be classified, according to their correlation value, as negatively correlated, not correlated, or positively correlated itemsets. More specifically, generalized itemsets with Kulc between max_neg_cor and min_pos_cor are uncorrelated, generalized itemsets with Kulc below max_neg_cor show negative item correlation, whereas generalized itemsets with Kulc above min_pos_cor indicate a positive item correlation, i.e., the co-occurrence between its items holds more than expected. For the sake of brevity, we respectively denote the above itemset correlation value ranges as *null*, *negative*, and *positive* correlation types throughout the paper.

4.3.2. Mining task

We extend the traditional generalized itemset mining problem by proposing a novel kind of pattern, namely the Strong Flipping Generalized Itemset (SFGI). SFGIs are patterns in the form $X \sim \Psi$, where X is a frequent generalized itemset having a large set Ψ composed of X's descendants whose correlation type differs from the one of X. A more formal definition follows.

Definition 4.3. Strong Flipping Generalized Itemset (SFGI). *Let $\mathcal{T}$ be a transactional Twitter dataset, Θ a taxonomy, min_sup a minimum support threshold, and FI the set of frequent generalized itemsets mined from $\mathcal{T}$ by enforcing min_sup. Let min_except a minimum number of exceptions. Let $X \in FI$ be a frequent generalized itemset of level l with correlation type t (positive, null, or negative) such that Ψ, which is the subset of X's frequent descendants of level $l-1$ with correlation type different from t, has cardinality greater than or equal to min_except. A SFGI is a pattern in the form $X \sim \Psi$.*

SFGIs $X \sim \Psi$ are worthy of consideration by domain experts, because they provide interesting insights into data correlation changes when generalizing items at different abstraction levels. As shown in Section 5, SFGI extraction from Twitter dataset is deemed to be particularly useful by a domain expert, because the extracted patterns may prompt targeted actions,

such as topic trend analysis, context-aware service profiling, and outlier detection.

Given a Twitter dataset $\mathcal{T}$, a taxonomy Θ, a minimum number of exceptions min_except, a minimum support threshold min_sup, and two maximum negative and minimum positive Kulc correlation thresholds max_neg_cor and min_pos_cor, TFC Analyzer discovers all the SFGIs (Cf. Definition 4.3) satisfying all the above constraints.

4.3.3. SFGI mining

To accomplish the SFGI mining task, the TFC Analyzer system exploits an efficient LCM-based itemset mining algorithm [18] combined with an ad-hoc postpruning phase. A thorough description of the mining process is reported below.

SFGI extraction entails the following steps: (i) Traditional frequent level-sharing itemset mining and (ii) SFGI extraction at the top of the previously extracted itemsets. During the itemset mining process the taxonomy is evaluated in a bottom-up fashion, i.e., the SFGIs are generated in order of increasing generalization level. A pseudo-code of the mining process is reported in Algorithm 1.

Frequent level-sharing itemset mining. For each SFGI $X \sim \Psi$ its X part is a frequent level-sharing itemset, while its Ψ part also contains frequent level-sharing X's descendants. Hence, we use a traditional level-sharing itemset mining algorithm to drive SFGI generation (see line 2).

To accomplish this task efficiently, we adopted a FP-Growth-like itemset miner, i.e., Linear Time Closed Itemset Miner v.2 (LCMv2) algorithm [18]. Similar to FP-Growth [23], LCM relies on a projection-based approach, i.e., it entails: (i) the creation and in-memory storage of an FP-tree-based dataset representation and (ii) the mining of frequent itemsets by recursively visiting conditional FP-tree projections. To suit the standard LCM implementation to generalized itemset mining, we adopted the strategy, first proposed in [40], of extending the dataset transactions by appending to each transaction all the item generalizations in Θ. Furthermore, the recursive generation of the projected FP-tree is tailored to level-sharing itemset mining. More specifically, the conditional FP-tree related to the level-l item i is populated by adding only level-l items. In such a way, the generation of candidate not level-sharing itemsets is prevented. Frequent level-sharing itemsets are stored in $\mathcal{TI}$ (line 2).

Algorithm 1 SFGI mining procedure

Input: a transactional Twitter dataset $\mathcal{T}$, a taxonomy Θ, a minimum support threshold min_sup, a minimum number of exceptions min_except, a maximum negative Kulc threshold max_neg_cor, and a minimum positive Kulc thresholds min_pos_cor
Output: the set of all SFGIs $\mathcal{L}$

1: */* Frequent level-sharing itemset mining. */*
2: $\mathcal{TI}$ = mineTraditionalLevelSharingItemsets($\mathcal{T}$, Θ, min_sup)
3: */* SFGI mining */*
4: $\mathcal{L} = \emptyset$
5: */* Generate SFGIs $X \sim \Psi$ having X with level $l > 1$ */*
6: **for** l=2 to *maxlevel* **do**
7: */* One candidate SFGI of level l is generated for each frequent level-sharing itemset of level l */*
8: */* maxlevel is the height of taxonomy Θ */*
9: **for all** W in $\mathcal{TI}[l]$ **do**
10: */* Create a candidate SFGI of level l with W as its X part. Ψ is initially set to the empty set */*
11: insert the candidate SFGI ($W \sim \emptyset$) into $C[l]$
12: **end for**
13: */* Exploit frequent level-sharing itemsets of level $l-1$ to populate the Ψ part of candidate SFGIs of level l. */*
14: **for all** it in $TI[l-1]$ **do**
15: */* Retrieve the candidate itemset genit of level l that is ancestor of it and update genit.Ψ */*
16: $genit$ = retrieveAncestor($TI[l]$,it,l,Θ);
17: cor_type_genit=Kulc($genit$,$\mathcal{T}$,max_neg_cor,min_pos_cor)
18: cor_type_it=Kulc(it,$\mathcal{T}$,max_neg_cor,min_pos_cor)
19: */* If the level-$(l-1)$ candidate it has a correlation type different from genit then it must be inserted into genit.Ψ */*
20: **if** $cor\_type\_genit \neq cor\_type\_it$ **then**
21: insert it into $genit.\Psi$
22: **end if**
23: **end for**
24: */* Select candidate SFGIs of level l with a large number of low level exceptions */*
25: **for all** c in $C[l]$ **do**
26: **if** $|\Psi| \geq$min_except **then**
27: insert c into $\mathcal{L}[l]$
28: **end if**
29: **end for**
30: **end for**
31: **return** $\mathcal{L}$

SFGI mining: Once all the frequent level-sharing itemsets X are generated, TFC Analyzer first associates with each of them a candidate SFGI $X \sim \Psi$ and populates its corresponding Ψ set. Then, SFGIs are evaluated and filtered according to their number of low-level exceptions (see Definition 4.3).

SFGIs are mined by following a level-wise approach, i.e., by climbing up the taxonomy step-wise from level 2 until the maximum generalization level is reached (lines 6-30). Note that, by Definition 4.3, level-sharing itemsets of level-1 cannot be the X part of any SFGI. Level-wise taxonomy evaluation prevents the need for multiple itemset scans. In fact, SFGIs with level-l X

part are generated from the sets of level-l and level-$(l-1)$ frequent level-sharing itemsets $\mathcal{TI}[l]$ and $\mathcal{TI}[l-1]$. In Algorithm 1 *genit* represents the level-l generalization of an arbitrary level-$(l-1)$ itemset *it* according to the input taxonomy. While level-l itemsets in $\mathcal{TI}[l]$ are used to populate the X part of the level-l SFGIs $X \sim \Psi$ (lines 9-12), level-$(l-1)$ itemsets in contrast with X in terms of correlation type are used to fill their Ψ set (lines 14-23). Hence, when mining level-l SFGIs, all the traditional frequent itemsets with level less than $l-1$ can be discarded.

Finally, the SFGIs of level l having a number of exceptions greater than or equal to min_except are added to the output set (lines 25-29).

Time complexity. The analysis of the complexity of the SFGI mining process can be divided into two steps. The first step concerns the analysis of the time complexity of the frequent level-sharing itemset mining task. Since it has been accomplished by a LCMv2 algorithm, it is linear in the number of extracted itemsets [18]. The second step entails the analysis of the SFGI mining procedure, which is performed at the top of level-sharing itemsets. Since the SFGI extraction requires a level-wise scan of the list of extracted itemsets, its time complexity is again linear in the number of mined itemsets.

5. Experimental results

We performed experiments on both real-life and synthetic datasets to evaluate effectiveness and efficiency of the proposed approach. Specifically, we analyzed (i) the applicability and usefulness of the TFC Analyzer system on three real Twitter datasets with the help of a domain expert (see Section 5.2), (ii) the expressiveness of the mined SFGIs compared to that of the patterns extracted by a state-of-the-art approach [5] from Twitter datasets (see Section 5.3), (iii) the impact of the parameters of the SFGI mining process on the mined SFGIs (see Section 5.4), and (iv) the scalability of the SFGI mining process on synthetic datasets (see Section 5.5).

The TFC Analyzer system was implemented in the C++ programming language. Experiments were performed on a 3.30 GHz Intel(R) Xeon(R) CPU E3-1245 PC with 16 GB main memory running Linux (kernel 3.2.0).

5.1. Datasets

This section briefly describes the characteristics of the evaluated datasets and taxonomies. All the datasets and taxonomies that were used in the experiments are available at [16].

Table 3: Main Twitter dataset characteristics.

Dataset name	Number of transactions	Average transaction length	Taxonomy height
Music	4,464	11.6	4
Sport	39,508	9.7	4
Cinema	3,431	9.2	4

Twitter datasets. We evaluated the usefulness and applicability of the SFGIs generated by the TFC Analyzer system from three real datasets retrieved from Twitter (http://twitter.com). The main characteristics of the Twitter datasets are summarized in Table 3.

TFC Analyzer exploits a crawler to efficiently access the Twitter global stream. To generate the three real Twitter datasets, concerning three different topics (i.e., Sport, Cinema, Music), we monitored the public stream endpoint offered by the Twitter APIs over a 1-month time period and we tracked the tweets containing a selection of approximately 10 significant keywords per topic. The crawler establishes and maintains a continuous connection with the stream endpoint to collect and store Twitter data. As described in Section 4.1, tweets ranging over the same topic are first preprocessed and then integrated into a transactional dataset (Cf. Definition 4.2). We collected the textual content of the tweets as well as their most relevant contextual features. For the latter we considered the tweet publication date, weekday, and time stamp as well as the GPS coordinates of the author location. Taxonomies over Twitter data are generated by exploiting the aggregation functions described in Section 4.2.

Synthetic datasets. We analyzed the scalability of the SFGI extraction process on transactional datasets generated using a synthetic generator [40]. The data generator allows us to automatically produce datasets and taxonomies with different characteristics. Specifically, the number of generated transactions, the average transaction length, and the taxonomy height can be manually configured.

5.2. Expert validation

We validated the usefulness of the SFGIs mined from the three real Twitter datasets reported in Table 3 with the help of a domain expert. The

Table 4: Examples of interesting classes of SFGIs and number of mined SFGIs per class.

Class	Number of SFGIs		
	Music	**Sport**	**Cinema**
Context-aware service profiling	1	5	4
Context-aware topic trend analysis	293	198	66
Outlier detection	5	4	6

experiments were performed by setting min_sup = 1%, max_neg_cor= 0.4, min_pos_cor= 0.5, and min_except = 2 for all the three considered datasets. As discussed in Section 5.4, we suggest using the aforementioned parameter setting as standard configuration, because it produces manageable sets of SFGIs for all the three analyzed Twitter datasets. More specifically, the number of SFIGs mined from the Cinema, Music and Sport datasets by setting the standard configuration is 327, 2059, 851, respectively. Note that the number of "traditional" level-sharing generalized itemsets mined from the same datasets by applying a traditional approach, such as the one proposed by Han et al in [22], is two orders of magnitude higher while enforcing the same minimum support threshold (i.e., 20933 for Cinema, 450383 for Music, 56824 for Sport). Since SFGIs are more easily manageable than traditional patterns, they are in practice more usable to discover interesting knowledge from Twitter data.

The domain expert, who has experience in traditional itemset minign from Twitter data, is in charge of comparing the information provided by SFGI with that provided by traditional itemsets. Specifically, the expert validation task is to pinpoint interesting and previously unknown knowledge from all the three considered datasets and, at the same time, highlight the higher usefulness of SFGIs compared to traditional itemsets. To focus the analysis on a worthwhile pattern subset, the expert first defined a set of SFGI "classes". We denote as a "class" a subset of SFGIs that are all related to the same subset of data features and that can be used to perform a similar in-depth analysis. Once the SFGI classes have been identified, the domain expert selected three classes that are deemed to be most useful in three real-life application contexts: context-aware service profiling, contextualized topic trend detection, and outlier detection. Table 4 reports the number of mined SFGIs per selected classes. All the SFGIs that comply with the aforementioned classes are considered to be interesting by the domain expert for in-depth analysis and thus they manually explored. To gain an insight into the achieved results, Table 5 reports a subset of representative SFGIs

Table 5: Examples of selected SFGIs $X \sim \Psi$ mined from the Twitter datasets.

ID	Dataset	Generalized itemset X (Kulc, support, level)	Exception set Ψ
1	Music	{(Date:Working Day),(Loc:Twickenham Rugby Stadium)} (Kulc=0.36 (Negative), Sup=11.0%, Level=2)	{(Date:2012-09-08),(Loc:51.45542_-0.34165)} (Kulc=0.66 (Positive), Sup=8.3%) {(Date:2012-09-10),(Loc:51.45542_-0.34165)} (Kulc=0.51 (Positive), Sup=1.5%)
2	Cinema	{(Date:Working Day),(Loc:Bayernring)} (Kulc=0.51 (Positive), Sup=7.6%, Level=2)	{(Date:2012-09-07),(Loc:52.48102_13.39100)} (Kulc=0.20 (Negative), Sup=1.7%) {(Date:2012-09-08),(Loc:52.48102_13.39100)} (Kulc=0.39 (Negative), Sup=4.7%)
3	Music	{(Date:Working Day),(Loc:Sumatralaan), (Text:mtv)} (Kulc=0.92 (Positive), Sup=72.4%, Level=2)	{(Date:2012-09-26),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.34 (Negative), Sup=1.6%) {(Date:2012-09-25),(Loc:52.23512_5.17347), (Text:mtv)} (Kulc=0.36 (Negative), Sup=4.1%) {(Date:2012-09-24),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.44 (Null), Sup=13.3%) {(Date:2012-09-22),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.31 (Negative), Sup=5.0%) {(Date:2012-09-21),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.37 (Negative), Sup=5.7%) {(Date:2012-09-20),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.42 (Null), Sup=10.2%) {(Date:2012-09-19),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.44 (Null), Sup=12.9%) {(Date:2012-09-18),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.48 (Null), Sup=16.8%) {(Date:2012-09-17),(Loc:52.23512_5.17347),(Text:mtv)} (Kulc=0.35 (Negative), Sup=2.7%)
4	Sport	{(Date:Working Day),(Loc:Stratford Walk), (Text:olympic)} (Kulc=0.53 (Positive), Sup=5.8%, Level=2)	{(Date:2012-09-08),(Loc:51.53841_-0.01648),(Text:olympic)} (Kulc=0.29 (Negative), Sup=2.8%) {(Date:2012-09-07),(Loc:51.53841_-0.01648),(Text:olympic)} (Kulc=0.30 (Negative), Sup=2.1%)
5	Cinema	{(Time:PM),(Loc:USA)} (Kulc=0.58 (Positive), Sup=15.7%, Level=3)	{(Time:[18-23],(Loc:CA_94607)} (Kulc=0.37 (Negative), Sup=1.9%) {(Time:[18-23]),(Loc:CA_90731)} (Kulc=0.23 (Negative), Sup=1.7%) {(Time:[12-17]),(Loc:CA_90731)} (Kulc=0.16 (Negative), Sup=1.2%) {(Time:[18-23]),(Loc:TX_75212)} (Kulc=0.30 (Negative), Sup=1.1%)
6	Sport	{(Time:AM),(Loc:South Korea)} (Kulc=0.59 (Positive), Sup=6.3%, Level=3)	{(Time:[00-05],(Loc:Seoul)} (Kulc=0.45 (Negative), Sup=3.9%) {(Time:[06-11]),(Loc:Seoul)} (Kulc=0.24 (Negative), Sup=2.3%)

selected, clustered by the corresponding class. For each SFGI in Table 5 we indicate the real dataset from which it was mined, and the Kulc value and the correlation type (positive, negative, or null). The considerations of the domain expert about the results reported in Table 5 are summarized below.

Context-aware service profiling. SFGIs may drive experts in profiling services according to user location and post submission time. Based on the analyzed patterns, experts can plan targeted location- and time-aware promotions. For example, the X part of the SFGI with ID 1 represents a high-level negative item correlation discovered from the Music dataset, which indicates that Twitter users do not often post tweets while they are nearby the Twickenham Rugby Stadium over working days. However, considering only the generalized itemset X by itself could be misleading, because when some important events (e.g., concerts) take place, user interests may unex-

pectedly change. Changes appear to be relevant by analyzing the exception set Ψ. For example, it contains a positive correlation between a specific date (September 8th, 2012) and location (Loc:51.45542_-0.34165) nearby the Twickenham Rugby Stadium. A similar positive correlation holds between September 10th, 2012 and the same location. Since the exception set Ψ of SFGI 1 contains 2 exceptions, the expert deemed its extraction to be relevant for in-depth analysis. Furthermore, analysts located very far away from the tweet submission place are often unaware of the current breaking events related to different countries or places. Hence, for marketing purposes they may would like to analyze the past Twitter user activities to discover most suitable time periods over which specific services (e.g., news services, video promotion, travel offers) should be offered. Surfing the Internet, the expert discovers that Lady Gaga performed two sold-out concerts at the Twickenham Rugby Stadium exactly on the same dates highlighted in the exception set Ψ of SFGI 1. Therefore, low-level correlation type changes are motivated by an extraordinary public event.

Similar to SFGI 1, the SFGI with ID 2 represents a correlation between working days and a location, which can be used to perform service profiling. Unlike the former case, the X part of SFGI 2 represents a positive item correlation, whereas the exception set contains a set of negative item correlations. Note that if we consider only the SFGIs that correspond to combinations of locations and weekdays the mining result becomes easily manageable by domain experts. For example, only 5 SFGIs belonging to this class were extracted from Sport (see Table 4).

Context-aware topic trend analysis. SFGIs may drive experts in profiling trends and topics of interest based on location and time. For example, let us consider SFGI 3. It indicates that many tweets about the MTV channel (Text:mtv) were posted from a specific location of the Netherlands (Sumatralaan) during working days. The X part of this SFGI has a positive item correlation. Hence, MTV can be considered to be of topical interest for people coming from Sumatralaan during working days. On the other hand, the same topic becomes less appealing in some specific days. The set of negatively correlated or uncorrelated exceptions is reported in the Ψ set. Discovering unexpected changes in user interests and classifying them according to user provenance or time period could be useful for planning targeted promotional campaigns for specific products or services. SFGI 4 was extracted from the Sport dataset and it represents a positive correlation between a specific combination of weekday, location and topic (Olympic),

which is worthy for topic trend analysis.

Outlier detection. SFGI analysis can be also targeted to identifying anomalous situations. Exploring generalized itemset exceptions, which may consist of positively correlated, negatively correlated, or even uncorrelated items, may allow experts to figure out unexpected or anomalous behaviors. For example, let us consider SFGI 5. It indicates that, although American people are used to post tweets "after midday" (i.e., from 12 A.M. to 12 P.M.), this recurrence does not hold for some specific locations and time slots (e.g., the correlation between users in located in CA_90731 and time slot [18 p.m., 23 p.m.] is negative). From a marketing viewpoint, this information could be valuable for outlier detection and analysis. For example, analysts may focus their attention on this kind of anomalous situations in Twitter data to discover interesting and underlying user behaviors. SFGI 6, which was mined from the Sport dataset, shows a similar contrasting situation between South Korea and Seoul. Specifically, "before midday" time slot (Time:AM) and South Korea are positively correlated, whereas between the time slot [00:00, 05:00] and the capitol of Korea (Seoul) the item correlation type is negative.

The domain expert confirms the higher manageability and significance of the extracted SFGIs with respect to traditional itemsets (traditional itemset sets are at least one order of magnitude larger than SFGI sets).

5.3. Comparison with a state-of-the-art flipping correlation miner

We also compared, with the help of the domain expert, the expressiveness of the SFGIs mined from the Twitter datasets with that of the flipping patterns extracted by a recently proposed approach, namely Flipper [5]. Flipper discovers flipping correlations, i.e., frequent itemsets whose correlation type flips from positive to negative (or vice versa) when itemsets are generalized to a higher level of abstraction. Note that a flipping correlation is a chain of itemsets, with pairwise ancestor/descendant relationships, and contains exactly one frequent (generalized) itemset per taxonomy level (e.g., 4 itemsets when dealing with 4-level taxonomies). Furthermore, each included generalized itemset (i) must be in contrast, in terms of correlation type, with both its corresponding ancestor and descendant and (ii) cannot be uncorrelated (i.e., the itemsets with correlation type null are disregarded even if their ancestor/descendant items are correlated with each other).

The comparison was performed on the Cinema dataset by enforcing the standard configuration (min_sup = 1%, max_neg_cor= 0.4, min_pos_cor= 0.5,

Table 6: Comparison between a representative SFGI and three flipping correlations [5]. Cinema datasets. max_neg_cor=0.4. min_pos_cor=0.5.

SFGI **min_sup = 1%, min_except = 2, max_neg_cor=0.4, min_pos_cor=0.5.**		
itemset	**kulc**	**support (%)**
{(Date:Working Day),(Time:from 7 a.m. to 12 a.m.)}	0.51 (Positive)	22.3
Exceptions:		
{(Date:Working Day),(Time:11 a.m.)}	0.38 (Negative)	3.07
{(Date:Working Day),(Time:10 a.m.)}	0.38 (Negative)	2.44
{(Date:Working Day),(Time:9 a.m.)}	0.42 (Null)	2.44
{(Date:Working Day),(Time:8 a.m.}	0.19 (Negative)	3.87
{(Date:Working Day),(Time:7 a.m.}	0.46 (Null)	0.16
{(Date:Working Day),(Time:6 a.m.)}	0.39 (Negative)	3.87
Flipping correlations **min_sup=0.08%, max_neg_cor=0.4, min_pos_cor=0.5.**		
itemset	**kulc**	**support (%)**
{(Date:Working Day),(Time:from 7 a.m. to 12 a.m.)}	0.51 (Positive)	22.3
{(Date:Working Day),(Time:6 a.m.)}	0.39 (Negative)	3.87
{(Date:Working Day),(Time:14-06)}	0.50 (Positive)	0.29
{(Date:2012-09-22),(Time:06.14.19)}	0.39 (Negative)	0.17
{(Date:Working Day),(Time:from 7 a.m. to 12 a.m.)}	0.51 (Positive)	22.3
{(Date:Working Day),(Time:6 a.m.)}	0.39 (Negative)	3.87
{(Date:Working Day),(Time:14-06)}	0.50 (Positive)	0.29
{(Date:2012-09-21),(Time:06.14.19)}	0.13 (Negative)	0.08
{(Date:Working Day),(Time:from 7 a.m. to 12 a.m.)}	0.51 (Positive)	22.3
{(Date:Working Day),(Time:10 a.m.)}	0.38 (Negative)	2.44
{(Date:Working Day),(Time:27-10)}	0.5 (Positive)	0.16
{(Date:2012-09-21),(Time:10.27.14)}	0.34 (Negative)	0.08

and min_except = 2). However, similar situations are present also in the patterns mined from the other datasets. The upper part of Table 6 reports a representative SFGI mined from the Cinema dataset. It consists of a positively correlated generalized itemset (i.e., {(Date:Working Day), (Time:From 7 a.m. to 12 a.m.)}) and its corresponding set of low-level exceptions. We compared the above SFGI with the patterns extracted by Flipper [5]. To achieve this goal, we performed several runs with the competitor algorithm by varying the min_sup threshold and by using the same correlation thresholds max_neg_cor= 0.4 and min_pos_cor= 0.5. To discover similar knowledge with Flipper, we had to enforce a significantly lower support threshold value (0.08%), because, since real-world data is relatively sparse, low-level itemsets occur rarely in the analyzed data. The mined flipping correlations are reported at the bottom of Table 6. Note that they represent only a subset of the SFGI's exceptions, because some of them are either uncorrelated or infrequent, at the least abstraction level, with respect to the minimum support

threshold. Hence, a set of flipping correlations is needed just to partially represent the knowledge covered by a single SFGI. In conclusion, the domain expert acknowledges that SFGIs provide a more complete and compact view of contrasting item correlations, which is commonly more usable than flipping correlations when coping with real-world data.

5.4. Impact of the algorithm parameters

We separately analyzed the impact of each TFC Analyzer parameter on the number of SFGIs mined from the three Twitter datasets. To this purpose, we run approximately 200 experiments. The results are summarized below. When not otherwise specified, the standard configuration was considered.

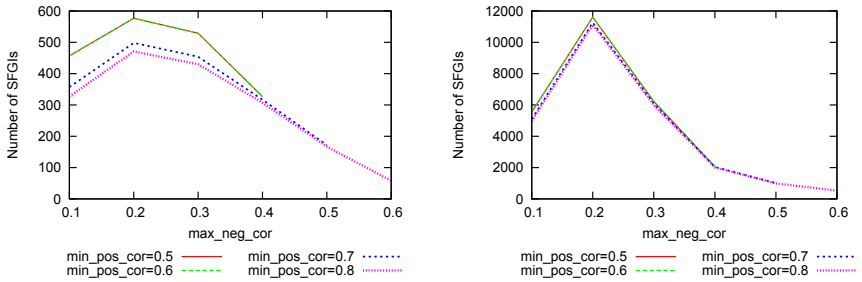
5.4.1. Effect of the correlation thresholds

Setting different values of maximum negative and minimum positive correlation thresholds may change the result of the SFGI mining process. To separately analyze the impact of the positive and negative correlation thresholds on the number of SFGIs mined from the Twitter datasets, in Figure 4 we plotted the number of SFGIs mined by varying max_neg_cor in the range [0.2-0.9] and by setting four representative min_pos_cor values[1], whereas in Figure 5 we analyzed the opposite situation, i.e., we varied min_pos_cor in the range [0.1-0.6] by setting four representative values for max_neg_cor. Since max_neg_cor $<$ min_pos_cor, then some curve points are missing. We performed all the mining sessions by setting standard values for the other parameters (i.e., min_sup = 1% and min_except = 2).

From the results reported in Figure 4, it appears that max_neg_cor has a significant impact on the number of mined SFGIs for all the considered Twitter datasets. For the analyzed datasets the highest number of SFGIs is extracted setting max_neg_cor in the range [0.2, 0.3]. To explain the reasons behind this trend, we thoroughly analyzed the distribution of the itemset correlation in the analyzed data. Itemsets are unevenly distributed in terms of correlation value for every Twitter dataset. Many itemsets have correlation between 0.2 and 0.4. Hence, the maximum number of SFGIs is extracted when max_neg_cor and min_pos_cor fall in this value range, because the generalization process is more likely to flip itemset correlation types.

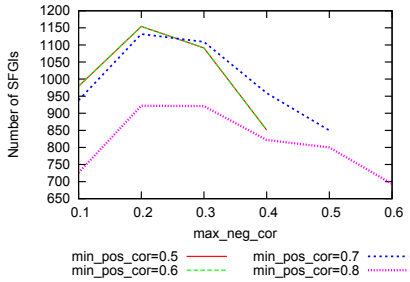
The impact of the min_pos_cor parameter appears to weakly affect SFGI mining performance. According to the results reported in Figure 5, the

[1]Curves related to min_pos_cor=0.5 and min_pos_cor=0.6 are overlapped in Figure 4.



(a) Cinema

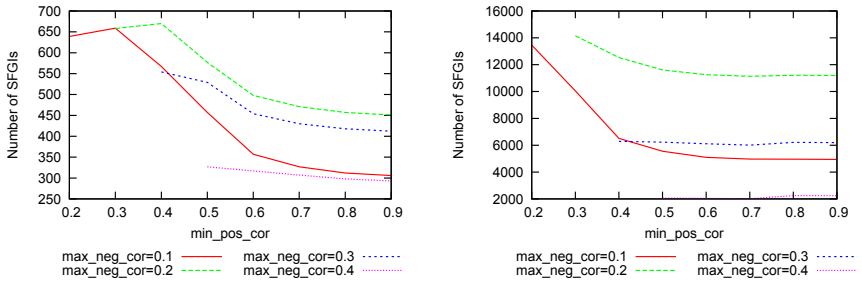
(b) Music



(c) Sport

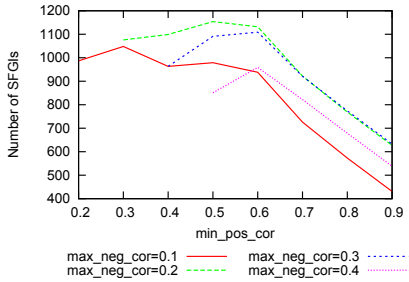
Figure 4: Effect of the maximum negative correlation max_neg_cor on the number of extracted SFGIs. min_sup=1%. min_except=2.

number of SFGIs mined from the Cinema and Music datasets is slightly affected by min_pos_cor variations while enforcing max_neg_cor values equal to or greater than 0.5 (especially when max_neg_cor $\geq$ 0.3). Despite the Sport dataset shows a relatively different data distribution with respect to Cinema and Music, and thus the corresponding curves are less stable, the number of SFGIs mined remains acceptable for most configuration settings. As expected, minimum and maximum correlation thresholds are strongly correlated with each other. Specifically, increasing the positive correlation threshold (or decreasing the negative one) yields a reduction in the number of extracted SFGIs, because correlation type changes at different abstraction levels are less likely to occur. Furthermore, setting relatively close correlation threshold values on average yields a higher number of extracted SFGIs,



(a) Cinema

(b) Music



(c) Sport

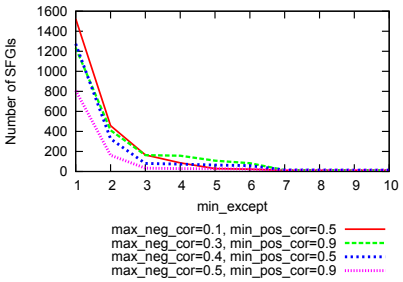
Figure 5: Effect of the minimum positive correlation min_pos_cor on the number of extracted SFGIs. min_sup=1%. min_except=2.

because correlation flipping occurrences occur, on average, more frequently. To generate a set of SFGIs easily manageable by domain experts and potentially useful for advanced analysis, we suggest experts set max_neg_cor to 0.4 and min_pos_cor to 0.5. As shown in Section 5.2, this configuration allows us to achieve a good trade-off between SFGI significance and cardinality on real-life Twitter datasets.

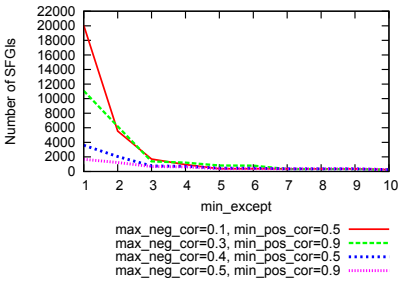
5.4.2. Effect of the minimum number of exceptions

The minimum number of exceptions is a threshold value that is used to drive SFGI mining. Specifically, only the frequent generalized itemsets having a large number of flipping low-level descendants are deemed to be worthy for manual inspection, because this patterns are more likely to represent contrasting and potentially interesting situations. Figure 6 reports,

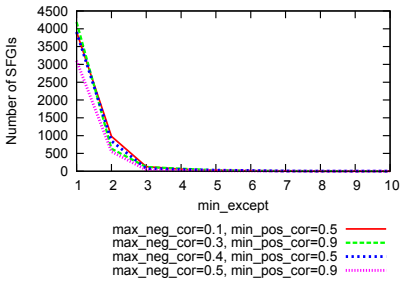
for each Twitter dataset, the number of SFGIs extracted by varying the minimum number of exceptions in the range [1,10] and by setting four representative pairs of values for the positive and negative correlation thresholds min_pos_cor and max_neg_cor. The minimum support threshold min_sup was set to its standard value (1%).



(a) Cinema



(b) Music



(c) Sport

Figure 6: Effect of the minimum number of exceptions min_except on the number of extracted SFGIs. min_sup=1%.

As expected, while increasing the minimum number of exceptions the cardinality of the SFGIs extracted decreases significantly. However, extremely large and unmanageable sets of SFGIs were mined only setting min_except to 1. The decreasing trend in the number of mined SFGIs is smoothed when setting correlation threshold values not fairly close with each other, because the total number of exceptions averagely reduces.

To discard uninteresting and potentially misleading SFGIs and to gener-

ate a manageable number of patterns, we suggest experts to set the minimum number of exceptions min_except to 2.

5.4.3. Effect of the minimum support threshold

The minimum support threshold may significantly affect the result of traditional itemset mining algorithms (e.g., Apriori [2], FP-Growth [23]). Hence, its impact on SFGI mining performance is worth considering as well.

Similar to traditional itemsets, the number of mined SFGIs increases enforcing lower min_sup values. In fact, when the minimum support threshold decreases the number of traditional generalized and non-generalized itemsets increases. Hence, even the number of candidate SFGIs does. Although setting different min_pos_cor and max_neg_cor values may yield different curve slopes, the achieved curves resemble in trend one another. The choice of the minimum support threshold is always a complex task. Setting relatively high support threshold (e.g., 4%) some interesting patterns can be accidentally discarded. Specifically, most of the negatively correlated itemsets are discarded because they are likely to have low frequency in the analyzed data. On the other hand, setting lower support thresholds (e.g., 0.5%) could yield the extraction of a large and potentially redundant set of patterns that is hardly manageable by domain experts. Setting min_sup to 1% is usually a good trade-off between result manageability and relevance.

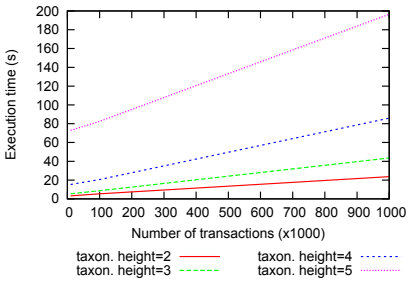
5.5. Scalability

We evaluated the scalability of the SFGI mining process, in terms of execution time, on synthetic datasets with (i) the number of dataset transactions, (ii) the taxonomy height (i.e., the number of generalization levels), and (iii) the average transaction length. Figure 7(a) reports the execution time by varying the number of transactions from 10,000 to 1,000,000 on datasets characterized by an average transaction length equal to 25 and by exploiting taxonomies with height in the range [2,5]. All the experiments were performed by setting the minimum support threshold to its standard value (1%). Similar to traditional generalized itemset mining algorithms (e.g., [3]), the SFGI mining process scales roughly linearly with the dataset cardinality. The curves obtained using taxonomies of different height show a similar trend but a different slope (see Figure 7(a)).

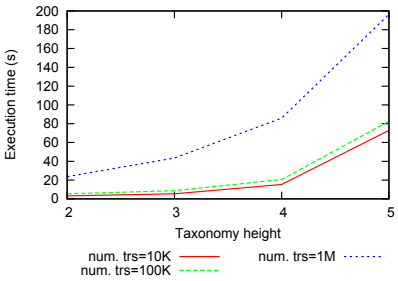
We also analyzed the impact of the taxonomy height on the miner execution time by exploiting taxonomies with height in the range [2,5] and datasets with three representative sizes (10K, 100K, and 1M). The obtained

results are reported in Figure 7(b). Similar to traditional generalized itemset mining algorithms (e.g., [40]) our mining approach scales more than linearly with the taxonomy height due to the combinatorial increase in the number of extracted high-level patterns. However, its execution time remains acceptable (i.e., less than 200 s) even when coping with quite complex datasets and taxonomies (e.g., dataset cardinality=1,000,000, taxonomy height=5).

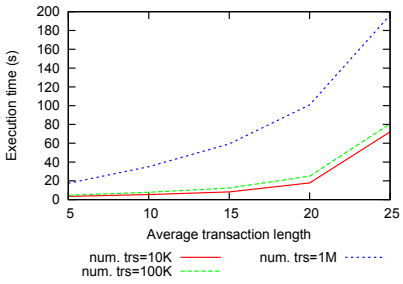
Finally, we also analyzed the scalability of our approach with the average transactions length. The results achieved on datasets with different size are reported in Figure 7(c). They demonstrate that our mining strategy scales non-linearly with the transactions length due to the potentially high number of generated combinations. However, the execution time remains limited (i.e., less than 200 s) for all the tested configurations.



(a) Num. of transactions.



(b) Taxonomy height.



(c) Avg transaction length.

Figure 7: Scalability analysis. min_sup=1%.

6. Perspectives of extension of TFC Analyzer in a distributed environment

To efficiently track and analyze the huge amount of messages posted on Twitter, we are currently investigating new solutions to extend the proposed approach towards Big data issues. Since most operations can be mapped to the MapReduce programming paradigm [17], the TFC Analyzer system may easily lend itself to a distributed-based implementation.

The distributed-based extension of TFC Analyzer consists of a series of distributed jobs running in a distributed system, such as Apache Hadoop [37]. Each job takes as input the result of one or more preceding jobs and performs a specific step needed to mine SFGIs from Twitter datasets. Most jobs are performed by one or more MapReduce tasks running on an Hadoop cluster.

For the extension of TFC Analyzer in a distributed environment we envisage the following steps:

- *Data acquisition*, which exploits a set of crawlers to gather data from Twitter and store new tweets in an HDFS distributed file system.

- *Data preprocessing and transformation*, which performs the following activities: (i) Textual data cleaning (e.g., removal of stopwords, numbers, etc.), (ii) Lemmatization, by using the WordNet lexical database [15], (iii) Data conversion to the transactional format (i.e., mapping of textual words and contextual data to distinct data items), (iv) Extension of the dataset transactions by appending all the item generalizations, according to the input taxonomy. The above tasks can be performed as distributed MapReduce jobs.

- *Taxonomy mapping*, which maps the input taxonomy to the Twitter data items.

- *Traditional level-sharing generalized itemset mining.* This step is accomplished by a combination of MapReduce tasks. We plan to extend the Apache Mahout project [38], which currently provides only a FP-Growth-like top-k closed itemset miner [30], with the twofold aim at (i) extracting *all* frequent itemsets and (ii) mining also high-level (generalized) level-sharing itemsets, according to the input taxonomy.

- *SFGI extraction* at the top of the previously extracted level-sharing itemsets. The key idea is to implement a MapReduce program that

relies on the following operations. (i) A Map operation, which takes as input the set of level-sharing itemsets and generates one key <parent itemset, level-sharing itemset> for each itemset by climbing up the taxonomy of one step. (ii) A Reduce operation, which aggregates pairs <key, value> generated by the Map operation and identifies for each "parent" itemset all of its descendants with the goal of selecting the "parent" itemsets that can produce an SFGI.

7. Conclusions

This paper presents Twitter Flipping Correlation Analyzer(TFC Analyzer), a novel data mining approach to efficiently supporting business applications based on social network data mining, e.g., context-aware service profiling and topic trend analysis. TFC Analyzer aims at discovering contrasting and unexpected situations from Twitter dataset equipped with taxonomies by extracting Strong Flipping Generalized Itemsets (SFGIs), a newly proposed type of patterns. The experiments performed on Twitter datasets demonstrate the applicability of the proposed approach to perform advanced analysis, such as topic trend analysis, context-aware service profiling, and outlier detection. The efficiency and effectiveness of the proposed system can be enhanced in several directions.

The two most promising future research perspectives are:

System scalability towards big data. As discussed in Section 6, since Twitter messages are being posted at an ever increasing rate, new and distributed solutions to extend the proposed approach towards big data issues need to be investigated.

SFGI mining from data coming from other research contexts. Given a taxonomy built over data items, SFGIs can be potentially extracted from any transactional dataset. Despite in this work we focus our analysis on Twitter datasets, because their characteristics are particularly suitable for data generalization and item correlation analysis, we aim at studying also the applicability of SFGIs to data coming from diverse application domains (e.g., network traffic data, sports data, financial data).

References

[1] C.C. Aggarwal, P.S. Yu, A new framework for itemset generation, in: Proceedings of the seventeenth ACM symposium on Principles of

database systems, PODS '98, ACM, New York, NY, USA, 1998, pp. 18–24.

[2] R. Agrawal, T. Imielinski, Swami, Mining association rules between sets of items in large databases, in: ACM SIGMOD 1993, pp. 207–216.

[3] R. Agrawal, R. Srikant, Mining association rules with item constraints, in: KDD 1997, pp. 67–73.

[4] E. Baralis, L. Cagliero, T. Cerquitelli, V. D'Elia, P. Garza, Support driven opportunistic aggregation for generalized itemset extraction, in: IEEE Conf. of Intelligent Systems, pp. 102–107.

[5] M. Barsky, S. Kim, T. Weninger, J. Han, Mining flipping correlations from large datasets with taxonomies, VLDB Endow. 5 (2011) 370–381.

[6] F. Benevenuto, T. Rodrigues, M. Cha, V. Almeida, Characterizing user behavior in online social networks, in: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, pp. 49–62.

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

[8] S. Brin, R. Motwani, C. Silverstein, Beyond market baskets: generalizing association rules to correlations, SIGMOD Rec. 26 (1997) 265–276.

[9] L. Cagliero, Discovering temporal change patterns in the presence of taxonomies, IEEE Trans. Knowl. Data Eng. 25 (2013) 541–555.

[10] L. Cagliero, T. Cerquitelli, P. Garza, L. Grimaudo, Misleading generalized itemset discovery, Expert Syst. Appl. 41 (2014) 1400–1410.

[11] L. Cagliero, A. Fiori, Discovering generalized association rules from twitter, Intell. Data Anal. 17 (2013) 627–648.

[12] L. Cagliero, A. Fiori, L. Grimaudo, Personalized tag recommendation based on generalized rules, ACM TIST 5 (2013) 12.

[13] L. Cagliero, P. Garza, Itemset generalization with cardinality-based constraints, Inf. Sci. 244 (2013) 161–174.

[14] M. Cheong, 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, pp. 1–8.

[15] W.L. Database, Available at http://wordnet.princeton.edu. Last Accessed: 01/10/2012, 2012.

[16] DBDMG, Database and Data Mining Web Site, Politecnico di Torino, available at https://dbdmg.polito.it/wordpress/research/strong-flipping-generalized-itemsets/ last accessed: 28/10/2013, 2012.

[17] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, in: Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6, OSDI'04, pp. 10–10.

[18] T.F. Gharib, An efficient algorithm for mining frequent maximal and closed itemsets, Int. J. Hybrid Intell. Syst. 6 (2009) 147–153.

[19] N.S. Glance, M. Hurst, T. Tomokiyo, Blogpulse: Automated trend discovery for weblogs, in: WWW 2004 Workshop on the weblogging ecosystem: aggregation, analysis and dynamics, ACM, 2004.

[20] L. Guo, E. Tan, S. Chen, X. Zhang, Y. 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, pp. 369–378.

[21] J. Han, Y. Fu, Mining multiple-level association rules in large databases, IEEE Transactions on knowledge and data engireering 11 (1999) 798–805.

[22] J. Han, Y. Fu, Mining multiple-level association rules in large databases, IEEE Trans. Knowl. Data Eng. 11 (1999) 798–804.

[23] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, In SIGMOD'00, Dallas, TX (2000).

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

[25] R.J. Hilderman, H.J. Hamilton, Knowledge Discovery and Measures of Interest, Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[26] G. Kasneci, M. Ramanath, F. Suchanek, G. Weikum, The YAGO-NAGA approach to knowledge discovery, ACM SIGMOD Record 37 (2009) 41–47.

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

[28] J. Kumar Pal, A. Saha, Identifying themes in social media and detecting sentiments, in: Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on, pp. 452–457.

[29] D. Kunkle, D. Zhang, G. Cooperman, Mining frequent generalized itemsets and generalized association rules without redundancy, J. Comput. Sci. Technol. 23 (2008) 77–102.

[30] H. Li, Y. Wang, D. Zhang, M. Zhang, E.Y. Chang, Pfp: parallel fp-growth for query recommendation, in: Proceedings of the 2008 ACM conference on Recommender systems, RecSys '08, ACM, New York, NY, USA, 2008, pp. 107–114.

[31] Q. Li, J. Wang, Y. Chen, Z. Lin, User comments for news recommendation in forum-based social media, Information Sciences (2010).

[32] X. Li, L. Guo, Y. Zhao, Tag-based social interest discovery, in: Proceeding of the 17th international conference on World Wide Web, pp. 675–684.

[33] Y.R. Lin, Y. Chi, S. Zhu, H. Sundaram, B.L. Tseng, Analyzing communities and their evolutions in dynamic social networks, ACM Transaction on Knowledge Discovery from Data 3 (2009) 8:1–8:31.

[34] M. Mathioudakis, N. Koudas, TwitterMonitor: trend detection over the twitter stream, in: Proceedings of the 2010 international conference on Management of data, ACM, 2010, pp. 1155–1158.

[35] D. Pagano, W. Maalej, How do open source communities blog?, Empirical Software Engineering 18 (2013) 1090–1124.

[36] N. Pasquier, Y. Bastide, R. Taouil, L. Lakhal, Discovering frequent closed itemsets for association rules, in: Proceedings of the 7th International Conference on Database Theory, ICDT '99, Springer-Verlag, London, UK, UK, 1999, pp. 398–416.

[37] T.A.H. Project, The Apache Hadoop Project. Available: http://hadoop.apache.org/ Last access on October 2013 (2013).

[38] T.A.M. Project, The Apache Mahout machine learning library. Available: http://mahout.apache.org/ Last access on October 2013 (2013).

[39] A. Savasere, E. Omiecinski, S.B. Navathe, Mining for strong negative associations in a large database of customer transactions, in: Proceedings of the Fourteenth International Conference on Data Engineering, ICDE '98, IEEE Computer Society, Washington, DC, USA, 1998, pp. 494–502.

[40] R. Srikant, R. Agrawal, Mining generalized association rules, in: VLDB 1995, pp. 407–419.

[41] K. Sriphaew, T. Theeramunkong, A new method for finding generalized frequent itemsets in association rule mining, in: Proceeding of the VII Intern. Symposium on Computers and Communications, pp. 1040–1045.

[42] P.N. Tan, V. Kumar, J. Srivastava, Selecting the right interestingness measure for association patterns, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), pp. 32–41.

[43] Y. Tian, P. Achananuparp, I.N. Lubis, D. Lo, E.P. Lim, What does software engineering community microblog about?, in: MSR'12, pp. 247–250.

[44] T. Wu, Y. Chen, J. Han, Re-examination of interestingness measures in pattern mining: a unified framework, Data Min. Knowl. Discov. 21 (2010) 371–397.

[45] Z. Yin, R. Li, Q. Mei, 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, pp. 957–966.