Information Systems 44 (2014) 120–133

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Discovering OLAP dimensions in semi-structured data

Svetlana Mansmann*, Nafees Ur Rehman, Andreas Weiler, Marc H. Scholl

Box 188, University of Konstanz, 78457 Konstanz, Germany

ARTICLE INFO

Available online 18 October 2013

Keywords:
Data warehousing
OLAP
Multidimensional data model
Semi-structured data

ABSTRACT

OLAP cubes enable aggregation-centric analysis of transactional data by shaping data records into measurable facts with dimensional characteristics. A multidimensional view is obtained from the available data fields and explicit relationships between them. This classical modeling approach is not feasible for scenarios dealing with semi-structured or poorly structured data. We propose to the data warehouse design methodology with a content-driven discovery of measures and dimensions in the original dataset. Our approach is based on introducing a data enrichment layer responsible for detecting new structural elements in the data using data mining and other techniques. Discovered elements can be of type measure, dimension, or hierarchy level and may represent static or even dynamic properties of the data. This paper focuses on the challenge of generating, maintaining, and querying discovered elements in OLAP cubes.

We demonstrate the power of our approach by providing OLAP to the public stream of user-generated content on the Twitter platform. We have been able to enrich the original set with dynamic characteristics, such as user activity, popularity, messaging behavior, as well as to classify messages by topic, impact, origin, method of generation, etc. Knowledge discovery techniques coupled with human expertise enable structural enrichment of the original data beyond the scope of the existing methods for obtaining multidimensional models from relational or semi-structured data.

1. Introduction and motivation

Explosion of social network activity in the recent years has led to generation of massive volumes of user-related data, such as status updates, messaging, blog posts and forum entries, recommendations, connection requests and suggestions and has given birth to novel analysis areas, such as *Social Media Analysis* and *Social Network Analysis*. This phenomenon can be viewed as a part of the "*Big Data*" [1] challenge, which is to cope with the rising flood of digital data from many sources including mobile phones, internet, videos, e-mails, and social network communication. The generated content is heterogeneous and encompasses textual, numeric, and multimedia data. Companies and institutions worldwide anticipate to gain valuable insights from Big Data and hope to improve their marketing, customer services and public relations with the help of the acquired knowledge. Meanwhile, results of Big Data analysis are incorporated into e-commerce sites and social networks themselves in the form of personalized content, such as recommendations, suggestions, advertisement.

The established data warehousing technology with On-Line Analytical Processing (OLAP) and data mining (DM)) functionality is known not only for its universality and high performance, but also for its rigidness and limitations when it comes to semi-structured or complex data. Various solutions have been proposed in theory and practice for warehousing and analyzing heterogeneous data. One class of solutions focuses on extending the capabilities of the predominant technologies, i.e., relational and multidimensional

* Corresponding author. Tel.: +49 176 24088000.
E-mail addresses: svetlana.mansmann@uni-konstanz.de (S. Mansmann), nafees.rehman@uni-konstanz.de (N. Ur Rehman), andreas.weiler@uni-konstanz.de (A. Weiler), marc.scholl@uni-konstanz.de (M.H. Scholl).

0306-4379/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2013.09.002

databases, while others pursue novel paths. A prominent example of the latter class is the NoSQL movement that announces the end of the relational era and proposes a wide range of alternative database approaches [2]. NoSQL databases are non-relational and intended for simple retrieval and appending operations, with the goal being significant performance benefits in terms of latency and throughput. However, they do not necessarily guarantee ACID (Atomicity, Consistency, Isolation, Durability) properties. Our work, however, fits into the “old-school” class since we choose to adapt the mature and established OLAP technology to non-conforming data scenarios. Our approach is based on (1) identifying parts of the dataset that can be transformed to facts and dimensions, (2) enriching the outcome by including external services (e.g., language and location recognition tools) and, finally, (3) extending the obtained structures via content-driven discovery of additional characteristics. The benefit of obtaining a properly structured and consolidated dataset lies in the ability to use the standard stack of tools for data analysis, visualization and mining to perform diverse analytical tasks.

The remainder of the introduction is dedicated to the main components of our solution, namely OLAP, data warehousing and mining as the employed data analysis technology and the social network of Twitter and its APIs as the underlying data source for building a data warehouse.

1.1. *Coupling OLAP and DM*

The necessity to integrate OLAP and DM was postulated in the late 90s [3]. Meanwhile, a powerful data mining toolkit is offered as an integrated component of any mature data warehouse system, such as Microsoft SQL Server, IBM DB2 Data Warehouse Edition, Oracle, and others. DM tools require the input data to be consolidated, consistent and clean. OLAP cubes – where the extracted data undergoes precisely that kind of transformation – appear to be perfect candidates for feeding the DM algorithms. Mining data cubes for dynamic classifications is a popular technique in OLAP applications dealing with customer trending, risk or popularity assessment, etc. However, traditional DM applications return such classifications as the outcome of the analysis, whereas our approach is to feed the obtained classifications back to the data warehouse as elements of the data model (e.g., dimensions or hierarchy levels) in their own right. Converting discovered structures into dimensional characteristics of a cube is an attractive data enrichment opportunity. However, it shakes the very foundations of the multidimensional data model as the latter presumes the non-volatility and static character of dimensional characteristics. The associated research challenges handled later on in this work are maintenance, evolution, temporal validity and aggregation constraints of discovered multidimensional elements.

1.2. *Tweet analysis as motivating example*

Twitter is an outstanding phenomenon in the landscape of social networking. Launched in 2006 as a simple platform for exchanging short messages on the Internet, Twitter rapidly gained worldwide popularity and has evolved into an extremely influential channel of broadcasting news and exchanging information in real-time. It has revolutionized the culture of interacting and communicating on the Internet and has impacted various areas of human activity, such as organization and execution of political actions, crime prevention, disaster management, emergency services. Apart from its attractiveness as a means of communication – with over 140 million active users generating over 340 millions tweets daily as of 2012 [4] – Twitter has also succeeded in drawing the attention of political, commercial, research and other establishments by making its data stream available to the public. Twitter provides the developer community with a set of APIs[1] for retrieving the data about its users and their communication, including the *Streaming API* for data-intensive applications, the *Search API* for querying and filtering the messaging content, and the *REST API* for accessing the core primitives of the Twitter platform.

To understand what type of knowledge can be discovered from this data, it is important to investigate the underlying data model. In a nutshell, it encompasses users, their messages (*tweets*), and the relationships between and within those two classes. Users can be friends or followers of other users, be referenced (i.e., tagged) in tweets, be authors of tweets or retweet other users' messages. The third component is the timeline, which describes the evolution, or the ordering, of user and tweet objects. Using the terminology of the Twitter Developer Documentation [5], the data model consists of the following three object classes:

1. *Status Objects* (tweets) consist of the text, the author and their metadata.
2. *User Objects* capture various user attributes (nickname, avatar, etc.).
3. *Timelines* provide an accumulated view on the user's activity, such as the tweets authored by or mentioning (tagging) a particular user, status updates, follower and friendship relationships, re-tweets, etc.

Even though the above model is not tailored towards OLAP, the offered data perspective is rather suitable for multidimensional aggregation. Essentially, Twitter accumulates various user and message related data over time. With a reasonable effort, this data stream can be transformed into a set of OLAP cubes with a fully automated ETL routine. What makes Twitter a particularly interesting motivating example for introducing the DM feedback loop is the fact that the structure of the original stream contains a rather small number of attributes usable as measures and dimensions of a cube, whereas a wealth of additional parameters, categories and hierarchies can be obtained using data enrichment methods of arbitrary complexity, from simple computations to complex techniques of knowledge discovery. Many of the characteristics (e.g., status, activity, interests, popularity, etc.) are dynamic and, therefore, cannot be captured as OLAP dimensions by definition. However, from the analyst's perspective, such characteristics may represent valuable dimensions of analysis.

[1] https://dev.twitter.com/start

The dataset delivered by the Twitter Streaming API is semi-structured using the JSON (JavaScript Object Notation) as its output format. Each tweet is streamed as an object containing 67 data fields with high degree of heterogeneity. A tweet record encompasses the tweeted message itself along with detailed metadata on the user's profile and geographic location. A straightforward mapping of this set of attributes to a multidimensional perspective results in the identification of cubes *Tweet* and *TweetCounters* for storing the contents and the metadata of the messages and the statistical measurements provided with each record, respectively.

1.3. Related work

The work related to our contribution can be subdivided into three major sections: (1) integrating data warehousing and mining, (2) OLAP for complex data, and (3) social network data analysis.

A pioneering work on integrating OLAP with DM was carried out by Han [3] who proposed a theoretical framework for defining OLAP mining functions. His *mining then cubing* function enables application of OLAP operators on the mining results. An example of implementing such function can be found in the Microsoft SQL Server and is denoted as *data mining dimensions* [6]. These dimensions contain classifications obtained via clustering or other algorithms on the original facts and can be materialized and used (with some limitations) just like ordinary OLAP dimensions. Usman et al. [7] review the research literature on coupling OLAP and DM and propose a conceptual model for combining enhanced OLAP with data mining systems. The urge to enhance the analysis by integrating OLAP and DM was expressed in multiple publications in the past. Significant works in this area include [8–11]. The concept of Online Analytical Mining (OLAM) as the integration of OLAP and DM was introduced by Han et al. [8].

Extending the limitations of the multidimensional data model is another actively researched subject in theory and practice. In 2001 Pedersen et al. [12] formulated 11 requirements of comprehensive data analysis, evaluated 14 state-of-the-art data models for data warehousing against those requirements, and proposed an extended model for handling complex multidimensional data. A similar attempt to classify and evaluate multidimensional models is presented in [13]. However, the authors defined two orthogonal sets of classification criteria, namely, according to the kind of constructs/concepts they provide and according to the design phase at which they are employed. Another assessment of conceptual models is provided in [14], in which the authors propose an exhaustive set of requirements regarding facts, dimensions, measures, operators, etc. A survey of research achievements on providing OLAP to complex data can be found in [15].

A spectacular novel area of data analysis is that of the social media analysis. Rapid expansion and extreme popularity of social networking have confronted the underlying backend architectures with unprecedented volumes of user-generated content. Thusoo et al. from the Facebook developer team describe the challenges of implementing a DW for data-intensive Facebook applications and present a number of contributed open source technologies for warehousing petabytes of data in [16]. Twitter is another leading social network with acute demand for a data warehouse solution. The first quantitative study on Twitter was published in 2010 by Kwak et al. [17] who investigated Twitter's topological characteristics and its power as a new medium of information sharing. The authors obtained the data for their study by crawling the entire Twitter site as no API was available at that time. Twitter API framework launched in 2009 inspired thousands of application development projects including a number of research initiatives. We limit ourselves to overview the related works which focus on discovering valuable information about the contents and the users.

In 2007 Java et al. [18] presented their observations of the microblogging phenomena by studying the topological and geographical properties of Twitter's social network. They came up with a few categories for Twitter usage, such as daily chatter, information and URL sharing or news reporting. Mathioudakis and Koudas [19] proposed a tool called Twitter Monitor for detecting trends from Twitter streams in real-time by identifying emerging topics and bursty keywords. Recommendation systems for Twitter messages are presented by Chen et al. [20] and Phelan et al. [21]. Chen et al. studied content recommendation on Twitter to better direct user attention. Phelan et al. also considered RSS feeds as another source for information extraction to discover Twitter messages best matching the user's needs. Michelson and Macskassy [22] discover main topics of interest of Twitter users from the entities mentioned in their tweets. Hecht et al. [23] analyze unstructured information in the user profile's location field for location-based user categorization.

Recent explosion of Twitter-related research confirms the recognized potential for knowledge discovery from its data. In this work we exploit the advantages of the established OLAP technology coupled with DM to enable aggregation-centric analysis of the meta-data about the Twitter users and their messaging activity.

1.4. Contribution

In this paper, we report our contribution of discovering, modeling and maintaining data warehouse elements from the dynamic and semi-structured data of social networks. In addition, we also demonstrate extraction of DW dimensions from the contents of tweets – which itself is completely unstructured data – by applying various data enrichment methods. Last but not least, the paper also details the process of analyzing current and historic states of the dynamic data.

The rest of the paper is organized as follows. Section 2 describes the process of capturing data from social networks, the transformations data takes at various layers of DW architecture, and acquiring facts and dimensions. Section 3 presents details on the modeling of the discovered elements from semi-structured data using x-DFM modeling approach. Section 4 talks about maintenance strategies of dynamic data and discusses slowly changing dimensions and its methods to respond to various kinds of changes in the data. Section 5 details a demonstration of all the methods presented in this paper using a Twitter dataset relevant to a popular sporting event. We conclude in Section 6.

2. Acquiring facts and dimensions

To exemplify the challenges of transforming semi-structured data into multidimensional cubes, let us recall the relevant concepts of the data warehouse design. Data in a data warehouse is structured according to the aggregation-centric multidimensional data model that uses numeric measures as its analysis objects [24]. A *fact* entry represents the finest level of detail and normally corresponds to a single transaction or event occurrence. A *fact* consists of one or multiple *measures*, such as performance indicators, along with their descriptive properties referred to as *dimensions*. Values in a dimension can be structured into a *hierarchy* of granularity levels to enable drill-down and roll-up operations. Natural representation of a set of facts with their associated dimensions and classification hierarchies is a *multidimensional data cube*. Dimensions in a cube represent orthogonal characteristics of its measure(s). Each dimension is an axis in a multi-dimensional space with its *member* values as coordinates. Finally, each cell contains a value of the measure defined by the respective coordinates.

The terms *fact* and *measure* are often used as synonyms in the DW context. In our work, it appears crucial to distinguish between those terms to account for facts without measures. According to Kimball [25], a fact is given by a many-to-many relationship between a set of attributes. Some scenarios require storing many-to-many mappings in which no attribute qualifies as a measure. Typical cases are event records, where an event is given by a combination of simultaneously occurring dimensional characteristics. Kimball proposed to refer to such scenarios as *factless fact tables* [25]. Mansmann [15] suggests to use a more implementation-independent and less controversial term *non-measurable fact type*.

Another relevant term is that of *Slowly Changing Dimensions* (SCD) introduced by Kimball [25] and formally summarized in [26]. Classically, dimensions in a data cube correspond to non-volatile characteristics of the data. In reality, however, the instance or even the structure of a dimension may be subject to changes. The problem of SCD is well elaborated in the literature, with various strategies proposed for maintaining either the up-to-date or the historical view, or even the entire history of the evolution. Most strategies employ some kind of multi-versioning to preserve various states of the aggregates. Saddat et al. [26] describe a methodology for multi-version querying in the presence of SCD.

2.1. Data warehouse architecture

A DW system is structured into multiple layers to optimize the performance and to minimize the load on the data sources. The architecture comprises of up to five basic layers from *data source* to *frontend* tools of the analysts. Fig. 1 introduces the resulting structure of our Twitter DW implementation. The data source layer is represented by the available Twitter APIs for data streaming and may include additional external sources, such as geographical databases, entity detection, event detection and language recognition systems for enriching the metadata and the contents of the streamed tweet records. The ETL (Extract, Transform Load) layer takes care of capturing the original data stream, bringing it into a format compliant with the target database and feeding the transformed dataset into the DW. The following

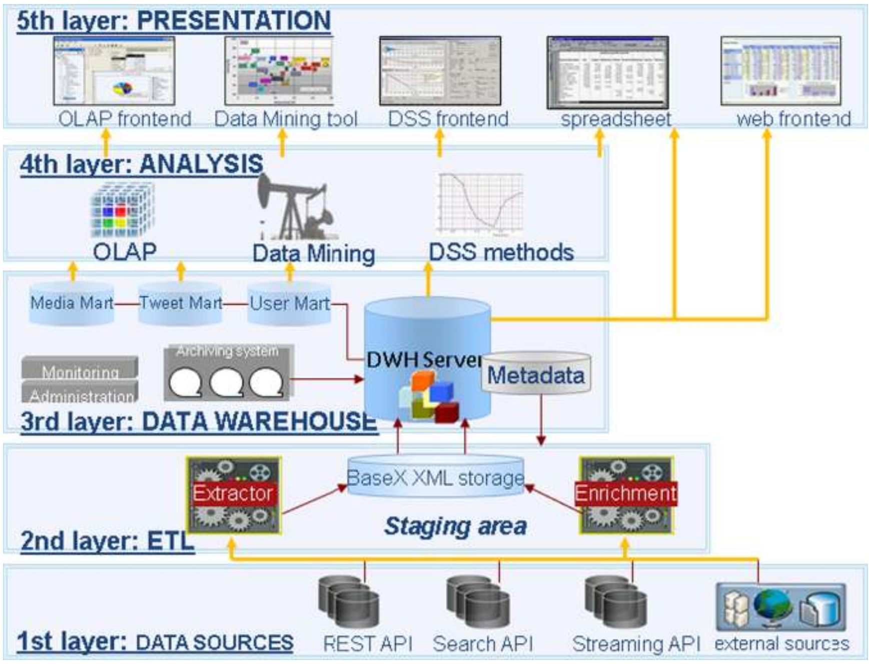

Fig. 1. Data warehouse architecture.

section details the tasks performed at this layer. The core layer of the system is the actual DW. The consolidated dataset in the database provides the basis for denying analysis-specific subtracts of data, denoted *data marts*. For example, the data related to user activity is extracted to *User Mart*, that of the embedded media in the messages can be found in *Media Mart*, etc. Data marts can be defined on demand to meet the requirements of specific areas of analysis. The two upper layers of the architecture comprise the front-end tools for analysis and presentation. The former are the expert tools for OLAP and data mining whereas the latter are the end-user (i.e., decision makers) desktop or web-based interfaces for generating reports, visual exploration of the data, executive dashboards, etc.

2.2. Data transformation

Mapping semi-structured data to multidimensional cubes is generally a challenging task since the original format admits heterogeneity while the target one enforces a rigid structure. In case of the Twitter stream, the degree of heterogeneity is rather low and affects only a few data fields. We investigated the structure of the streamed data by converting JSON objects into an XML and buffering the output into a native XML database BaseX [27] developed within our working group. The following XML snippet gives an example of a converted tweet object:

```
<tweet>
  <text>
    Earthquake with the.scale of 8.9 magnitude
    #PrayForIndonesia #PrayForSumatera
  </text>
  <date>Wed Apr 11 08 : 57 : 02+00002012</date>
  <source>web</source>
  <retweeted>false</retweeted>
  <user>
    <name>Miley ***</name>
    <date>Tue Jun 22 08 : 33 : 12+00002010</date>
    <statuses_count>13101</statuses_count>
    <followers_count>1019</followers_count>
  </user>
</tweet>
```

We use BaseX storage [27] as a staging area for the very fact of transformation required from semi-structured data into structured data. And the fact that tweets stream in into our systems at high rate, i.e., over 2 million semi-structured tweet objects per hour – keeping in mind that

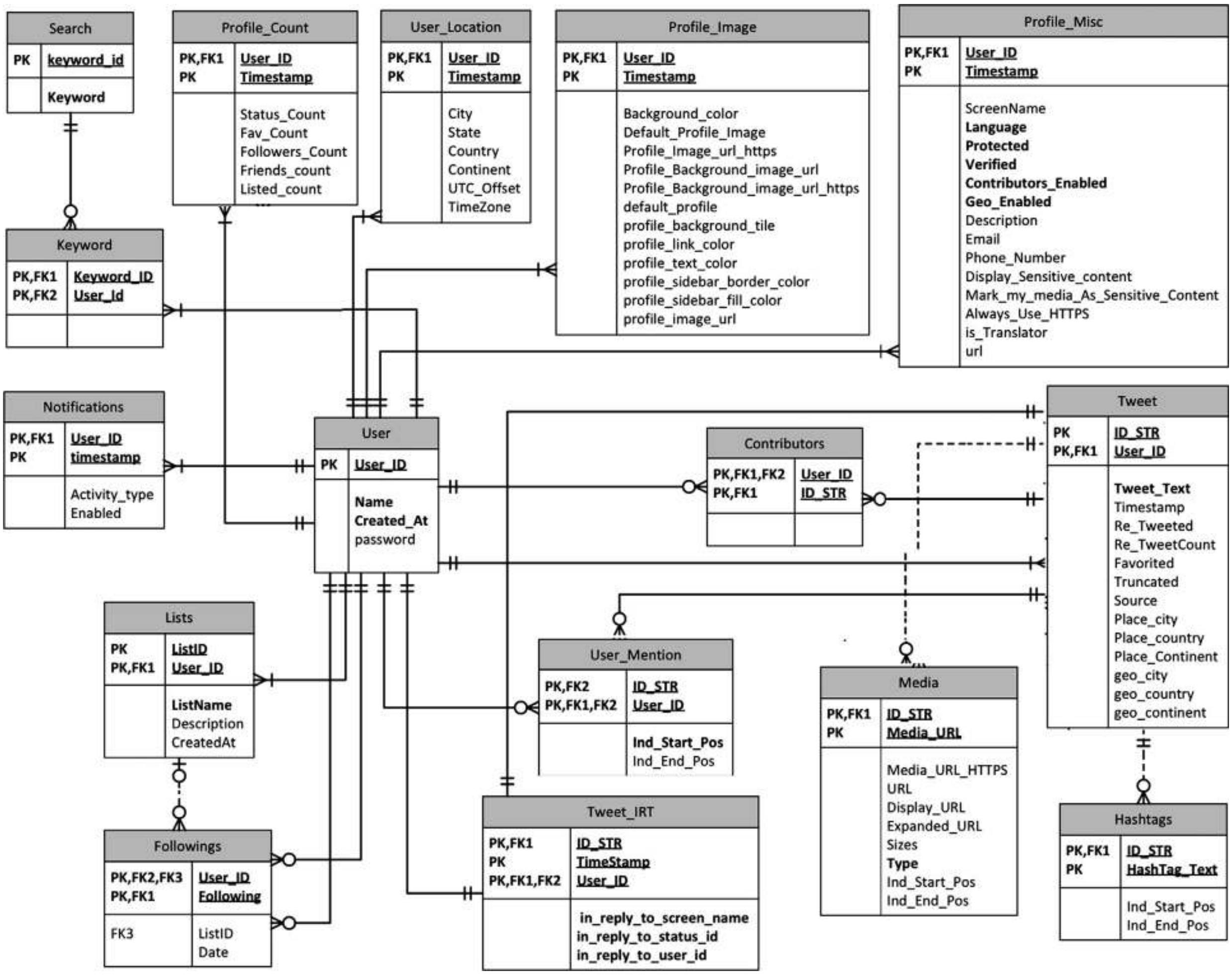

Fig. 2. Relational view of the Twitter stream as a UML class diagram.

we only have access to 10% of the total tweets from Twitter platform. The high performance BaseX is able to cope with such high data arrival rate. These requirements motivate the use of BaseX in our architecture. However, we used it purely as a staging area to transform the semi-structured data and to prepare its loading into the DW. The entire transformation process consists of obtaining the relational view from the XML schema and mapping the latter to the multidimensional model. The intermediate step of obtaining the relational model of the data is helpful in identifying classes of objects, their attributes and relationships, appropriate value domains of the attributes, cardinalities of the relationships, and integrity constraints. Regrettably, Twitter's own documentation of its data model is limited to brief definition of single data fields with no specification of constraints or relationships between those fields. In Fig. 2 we present the results of our attempt of reverse engineering the relational view from the original stream in the UML notation.

Objects describing a Twitter *user* in this relational model are *user*, *Profile_Misc*, *Profile_Image*, *User_Location* and *Profile_Count*. Fields pertaining to the appearance or look-and-feel of the Twitter account are grouped and recorded in *Profile_Image*. Location information and timezone are stored in *User_Location*. *Profile_Misc* lists various fields, e.g, what language the user tweets in, whether the account is verified and or protected. Whether the user allows geo information to be recorded and displayed with the tweets a user makes, and many other relevant fields are given to store user profile settings. Similarly, *Tweet*, *Media*, *User_mention*, *Contributors* and *Tweet_IRT* collectively store a normalized view of tweets a user make. The location information is stored in two sets of fields, i.e., Place and Geo. Place fields reflect values provided by the user about its location while geo fields contain decoded geographic information from latitude and longitude coordinates of a tweet. A tweet object may contain some kind of media, i.e., vine, video, photo or an audio. Such information is stored in the *Media* object. Twitter allows more than one users to contribute to an account and tweets, such information is stored in the *Contributors* object. Information on mentions of users and reply are stored in *User_Mention* and *Tweet_IRT*, respectively. Searching module on Twitter allows to store user's search query. This information is stored in *Search* and *Keyword* objects collectively. A user can follow other users and may group them into lists for better manageability. *Lists* and *Followings* object store such information.

Almost all of the field values in these objects are expected to change during the course of user activity over time excluding only a few, a composite key of User_ID and Timestamp is used to uniquely identify any such change.

The subsequent step of obtaining a multidimensional perspective of the same data is performed in a semi-automated fashion. The manual part is concerned with semantic interpretation of the data and specifying the facts and the measures of interest as well as desired dimensions of the analysis. The automated part is a cardinality-based definition of facts and dimensions as described in [15]. The data model of Twitter contains only a small set of numeric attributes, which qualify as measures. These attributes encompass the counters in the user profile and in the tweet record. Other attributes are of descriptive nature and, therefore, should be mapped to dimensions or hierarchy levels. With the obtained model of the original stream, a *Tweet* event appears to be the fact of the finest grain, with time, location, and user characteristics as its dimensions. All other characteristics are included into the respective dimensions or extracted into other facts.

A dimension is a *one-to-many* characteristic of a fact and can be of arbitrary complexity, from a single data field to a large collection of related attributes, from uniform granularity to a hierarchical structure with multiple alternative and/or parallel hierarchies. At the conceptual modeling stage, a dimension is structured as a graph of hierarchy levels as nodes and the “rolls-up-to” relationships between them as edges. We adopt the graphical notation of the x-DFM (Extended Dimensional Fact Model) [15] which is an extension of the Dimensional Fact Model of Golfarelli et al. [28]. The x-DFM makes provisions for various kinds of behaviors in OLAP dimensions as well for some advanced constructs, such as derived measures and categories, degenerated dimensions and fuzzy hierarchies, relevant for our model. Fig. 3 shows a fragment of modeling a cube for storing various cumulative measures of the user activity in the x-DFM. The structure of the cube is a graph centered at the fact type node (*TweetCount*), which includes all measures (*#friends*, *#followers*, *#status*, *#favorited* and *#listed*) and a degenerated (i.e., consisting of a single data field) dimension (*FactID*). Dimensions are modeled as outgoing aggregation paths. All paths within a dimension converge in an abstract ⊤node, which corresponds to the aggregated value *all*. A level node in a dimension consists of at least one key attribute, but may include further attributes represented as underlined terminal nodes.

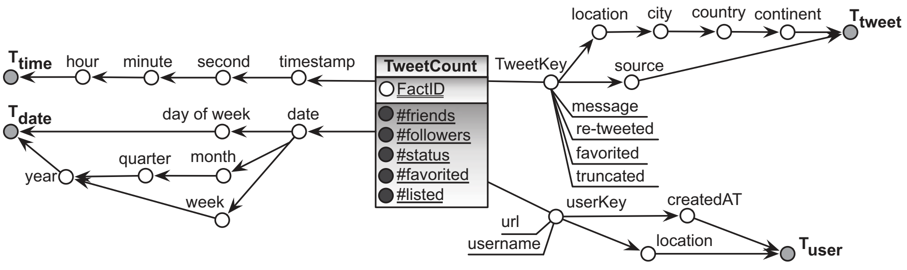

Fig. 3. Tweet fact in the x-DFM.

2.3. Discovering new elements

So far, we only considered the explicitly available characteristics of the original set for constructing the cube. Once those characteristics are mapped, the resulting model can be refined and enriched by adding new elements of type measure, category, dimension or even an entire cube. We were able refine the original dataset and its multidimensional view by applying the following techniques:

1. *Use of additional data sources and APIs*: Inclusion of external sources provides an opportunity to add new dimensional characteristics to a datacube. Here are some prominent examples of detection techniques relevant for enriching the Twitter data:
    - *Language detection* adds the tweet's language as a dimension of the tweet record. Language detection APIs, such as the one offered by Google or JSON, provide such service. Once detected, the language information can be used for enabling cross-lingual analysis and aggregation.
    - *Spam detection* helps identify whether a tweet is spam or contains malicious content. This can be done by employing the APIs of Askimed and Defensio or another similar service. Moreover, early detection of the spam level is beneficial for reducing the relevant dataset prior to its loading into the data warehouse (unless spam preservation is desired for the analysis).
    - *Topic detection* enriches tweet records with topic assignment. Twitter's own Search API can be used to retrieve daily trending topics and identify tweets relevant for a specific topic.
    - *Sentiment detection* assesses the overall emotion of the content (such as positive, negative or neutral). AlchemyAPI and OpenCalais are examples of platforms enabling this type of analysis.
    - *Keyword, Entity, and Event detection* are the methods of structuring the information conveyed by the message. In the original set, the entire content of a tweet is stored as a single text field. Systematic detection of significant keywords, entities (e.g., persons, locations, dates, products, etc.) and events (natural disasters, terror attacks, political elections, sports competitions, etc.) within this field provides its multidimensional perspective and refines the grain of the data from a tweet record down to single terms.

    Used in a combination, the above methods build the foundation for a comprehensive analysis of user-generated content.
2. *Derivation from existing characteristics*: Dimensions of a cube are expected to be orthogonal, i.e., unrelated to one another. In practice, however, it may be beneficial to derive new characteristics from the existing ones and materialize their instances to be able to use them as aggregation paths in OLAP queries. For example, one could add a new tweet dimension *media type* with values “*plain text*”, “*image*”, “*video*”, etc. based on the embedded multimedia content in the tweet message.
3. *Use of knowledge discovery techniques*: DM algorithms are helpful for discovering less obvious or hidden relationships and patterns in the dataset. The underlying dataset can be mined for a variety of descriptive and predictive tasks to build respective classification models. For example, users or tweets in the underlying dataset can be clustered into various groups based on their popularity, tweeting activity, topics discussed, etc., to name a few. These discovered groups aid analytics as they offer new perspectives for multidimensional analysis and can be used as grouping criteria just like statically defined dimension categories.

Note that each of the added characteristics can serve as an input for discovering new characteristics, alone or in combination with other properties. For instance, identifying the language of the content and generating a machine translation of the text into a common default language by using the Google API open up an opportunity to create a multilingual hierarchy of topics, keywords, hashtags, etc. and thus enable a cross-lingual aggregation.

In the next two sections we concentrate on the process of defining and maintaining discovered elements as well as their usage in OLAP queries.

3. Modeling discovered elements

Basically, a cube can be extended by adding new elements of type *measure* or *dimension category*. A measure is a simple atomic field of a fact entry. Therefore, computing a new field of this type does not require additional adjustments to the overall cube structure. However, adding a new dimension or a hierarchy level to an existing dimension imposes a number of challenges with respect to modeling, implementing, querying, and maintaining such added element. We demonstrate the differences in handling static, derived, and discovered dimension categories at the example of the *user* dimension in *TweetCount* cube depicted in Fig. 3, with its bottom-level category *userKey* and its parallel roll-ups by *creation date* and by *location*.

Let us assume an introduction of a derived hierarchy *ranking* → *rating* → *popularity* based on the user's ranking in terms of the number of this/her followers and friends. There exist different methods of computing the ranking of the Twitter user, but most of them agree on the prevailing role of the number of followers. We adopt a simple formula:

$$\textbf{ranking} = 0.8 * \#\text{followers} + 0.2 * \#\text{friends}$$

where *#friends* and *#followers* are the user's most recent counters from the cube *TweetCount*. With the proposed computation, the ranking values may range from 0 to about 25 Mln. Therefore, it appears feasible to introduce additional groupings for this property. We adopt a percentage-based rollup into *rating*, where the users are evenly distributed into 100 groupings according to their ranking. Thereby, rating 1 is assigned to 1% of the total number of users with the highest ranking. To further consolidate the groupings, the next hierarchy level called

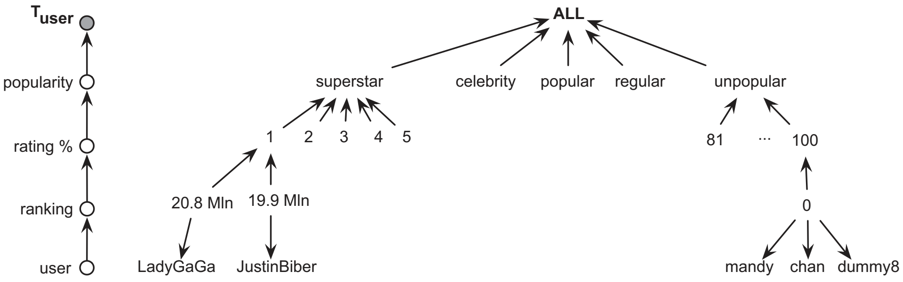

Fig. 4. User rating dimension.

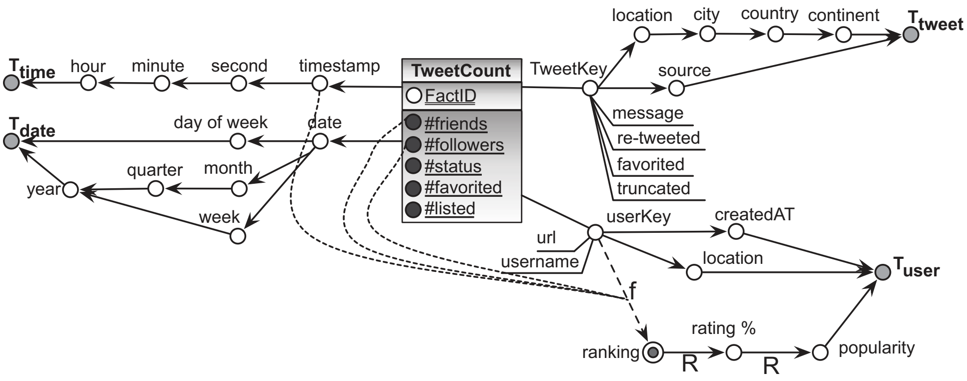

Fig. 5. Adding ranking to the user dimension.

popularity is introduced, offering just five instances, such as *superstar*, *celebrity*, *popular*, *regular*, and *unpopular*, with percentage based assignment (e.g., the bottom 20% are considered *unpopular*, the top 5% are *superstar*, etc.). Fig. 4 shows the schema of the proposed dimension (left) as well as a fragment of its hierarchy instance (right). Obviously, the instance of such a computed hierarchy reflects the state of the data valid at the moment of its computation. The validity of this assignment becomes obsolete once the underlying fields *#friends* and *#followers* get modified.

Another example of an interesting dynamic classification in the user dimension is a taxonomy of user intentions introduced in [18]. The instances of *user intension* comprise *Daily Chatter*, *Conversations*, *Sharing Information*, and *Reporting News*. Assignment of a user to one of the instances in this classification is based on the analysis of multiple criteria including the frequency of twitting, writing direct responses to other users, linking to other sources, focusing on specific topics. Since many of the users may display multiple intension patterns and the intension of a user can evolve over time, the primary current intension can be determined with the help of data mining methods, such as clustering.

Both classifications, i.e., *ranking* and *user intension*, introduced here have a common property distinguishing them from standard OLAP dimensions, namely their sensitivity to the evolution of the underlying dataset. In the extreme case of requiring full consistency with the current state that implies the necessity to update the dynamic elements after each loading of new data into the cube. This observation leads to a more general problem of coping with changes in dimensions to be discussed in the next chapter.

Back to the task of the conceptual modeling of dynamic elements, it is apparent that the formal and the graphical notation of the multidimensional data model needs to be extended to support such elements. The x-DFM notation provides graphical elements for specifying derived categories. We adjust this notation to specify how a dynamic element is computed. Fig. 5 shows the results of adding the user ranking hierarchy to the original cube. The derived category ranking is added as a parent level of userKey and is linked to the elements it is computed from by dotted lines. Since the ranking is computed from the most recent number of followers and friends, the linked input fields are the measures *#followers* and *#friends* as well as the timestamp category of the time dimension. The label “f” attached to the roll-up edge specifies that the category is computed based on a formula. Roll-up from ranking to rating % and popularity is a rule-based one (label “R”) and does not involve any extra input fields. In a similar fashion, roll-up edge notation can be extended to specify characteristics extracted with the help of external services, APIs, etc.

Whenever DM algorithms are used for creating a discovered classification, such as the *user intension* in our example, the available derivation notation may be insufficient. In our previous work [29] we presented an approach to model mined dimensions based on symmetric treatment of measures and dimensions for obtaining a homogeneous graph of all data fields and hierarchical relationships between them.

4. Maintaining dynamic elements

Discovered elements of type dimension category may be of a static nature (e.g., language, sentiment, topic) or evolve over time along with the evolution of the dataset. The former type can be treated just as a full-fledged dimension category since no additional constraints on maintaining the data are imposed in that case. The latter type, however, behaves similar to a *changing dimension* – a term introduced by Kimball in [30]. Kimball distinguishes between slowly and rapidly changing dimensions and identifies various patterns of change occurrence. Several strategies of handling changes in OLAP dimensions have been proposed in the literature and implemented in leading data warehouse systems. Even though none of the previously identified evolution patterns and implementation alternatives deals with the dynamics of discovered categories proposed in this work, we wish to investigate to which extent the former can be adopted for such scenarios.

We wish Kimball had given descriptive names to these responses like "overwrite" instead of "Type 1" for better readability and understanding. However these have become part of the community's language and are frequently used now. Let us recall various types of responding to change according to Kimball and apply them to our examples.

Type 0 response is a passive approach in which no action is taken to reflect the changes in the dimension. A single instance of the dimension exists, in which all attributes preserve their original values. This option of preserving only the historical view may appear satisfactory for some scenarios, but inadmissible in the general case. With dynamic categories, it is obviously necessary to keep the track of the changes in such categories for an up-to-date assignment.

Type 1 response to SCD is to simply overwrite old values with new ones. With this option, a single instance of the dimension is being maintained, in which all values correspond to the most recent assignment. Applying this option to store the user's rating and intension values for Twitter analysis would mean inevitable loss of all previously computed values of these characteristics. Consequently, there will be no possibility to analyze the evolution of those characteristics or to perform historically correct aggregation. Fig. 6 illustrates the effects of storing the user rating according to Type 1.

Type 2 response aims at correct preservation of the prior history by adding a dimension row for each change. Since a single instance in a dimension is stored using multiple rows (one for each change), an extra surrogate key has to be introduced to uniquely identify each row and to be used as a foreign key from within the fact table. Fig. 7 illustrates the effects of storing the user rating according to Type 2. A common extension of Type 2 storage is to add extra columns to the dimension table for storing the start and the end timestamp for each version. Even though this solution provides an accurate change tracking and ensures historically correct aggregation, it has a huge disadvantage of having to replace natural keys by the surrogate ones in the fact table. Especially with dynamic categories, whose values are computed from the fact entries, this approach would imply modification of the existing fact entries.

Type 3 method enables limited change tracking by using a separate column for each version of the changed attribute. This method is not an option for dynamic categories where we expect repeated and unlimited refreshment of the computed values in the dimensional table.

Type 4 response appears much more promising for managing multiple versions of the dimension's instance. This approach keeps the current data in the dimension

TWEETCOUNT-FACT

FactID	time	date	tweetkey	userkey	#friends	#followers	#status	#favorited	#listed
198776	22:53:19	2012-01-30	43611234	1308331	15	143	29	2	0
...	...	...	...	...	...	...	...	...	...
203980	09:12:38	2012-02-02	72693115	1308331	24	208	130	14	1

USER-DIM

userkey	url	name	createdAt	location	ranking	ranking%	popularity	...
~~1308331~~	~~-~~	~~wp-guru~~	~~2010-06-21~~	~~London, GB~~	~~117.4~~	~~83~~	~~unpopular~~	~~...~~
1308331	-	wp-guru	2010-06-21	London, GB	171.2	79	regular	...

Fig. 6. Type 1 SCD strategy for storing user ranking with no history preservation.

TWEETCOUNT-FACT

FactID	time	date	tweetkey	userkey	#friends	#followers	#status	#favorited	#listed
198776	22:53:19	2012-01-30	43611234	1308331a	15	143	29	2	0
...	...	...	...	...	...	...	...	...	...
203980	09:12:38	2012-02-02	72693115	1308331b	24	208	130	14	1

USER-DIM

origkey	userkey	url	name	createdAt	location	ranking	ranking%	popularity	...
1308331	1308331a	-	wp-guru	2010-06-21	London, GB	117.4	83	unpopular	...
1308331	1308331b	-	wp-guru	2010-06-21	London, GB	171.2	79	regular	...

Fig. 7. Type 2 SCD strategy for storing user ranking with history preservation.

USER-DIM

userkey	url	name	createdAt	location	ranking	ranking%	popularity	...
1308331	-	wp-guru	2010-06-21	London, GB	171.2	79	regular	...

USER-HiSTORY

userkey	url	name	createdAt	location	ranking	ranking%	popularity	...	created
1308331	-	wp-guru	2010-06-21	London, GB	117.4	83	unpopular	...	2012-02-01

Fig. 8. Type 4 SCD strategy for storing user ranking with current and previous states.

table and extracts older versions into one or several "*history tables*". Fig. 8 shows the storage of user dimension with only the current state in the dimension table and the previous states in the historical table.

Finally, *Type* 6 is proposed as the hybrid of Types 1, 2, and 3. Just like Type 2, this solution also imposes the use of surrogate keys in the fact table implementation.

Dynamic categories generated from the fact data through DM or other computations can be considered a special case of SCD, in which the changes occur with a certain regularity. The state of the dynamic category is guaranteed to be fully up-to-date, if it was computed from the most recent state of the underlying set of facts. However, it may be unaffordable to recompute the assignment each time new facts get inserted into the cube. Instead, interval-based or on-demand refreshment can be employed depending on the recency requirements and the prevailing change pattern. Back to our examples, user ranking is a rapidly evolving characteristic since the underlying counters of friends and followers change frequently at least for active users. As for user intension, this assignment is expected to be more stable as it is based on the prevailing usage patterns and clustering of similar behaviors.

Whatever refreshment strategy is used in a dimension with dynamic categories, Type 4 response to SCD has proven to offer an adequate solution for managing both the current version and all previous states of the dimension instance. No surrogate keys are necessary and no adjustments in the fact table implementation. The dimension's instance turns into a multi-versioned one, where a particular version can be retrieved by querying the timestamps of the instances.

Last but not least, it appears crucial to normalize the dimension table according to the snowflake schema. In the existence of several dynamic categories or change patterns within a single dimension, storing all attributes and their assignments in the same dimensional table would lead to extreme redundancy and confusion. Decomposition into separated tables for each hierarchy level or at least each hierarchy path makes it possible to handle changes in that particular path using a dedicated history table.

4.1. OLAP queries with multi-versioning

Adopting the Type 4 strategy to handle changes in the dimension generates a multi-versioned instance of any changing dimension. Availability of the current state as well as of each previously valid state makes it possible to perform historically correct aggregation by joining the fact entries with the matching versions of the dimension records. Besides, one can aggregate recent facts along a historical version of a dynamic characteristic or aggregate historical data along the current state of the changing category. Examples of queries containing a deliberate version mismatch are "retrieve the messages twitted in 2009 by the users who are popular now (and not in 2009!)", or "retrieve recent tweets containing the hashtags which were in top 20 in 2008".

If pre-aggregation is used for materializing the aggregates at different levels of grain, co-existence of multiple versions in a dimension does not cause problems because each fact entry has exactly one matching version of the dimension's record. Thereby, pre-aggregation produces historically correct values.

5. Demonstration

Twitter has become a reflection of all real-world events. Let it be the Arab uprising, any natural disaster, political elections, movie/music launch or sport events, it gets reciprocated into a huge social activity on Twitter. Data analysts expect valuable insights from event-oriented analysis of the Twitter stream that delivers user-generated content. Our usage scenario is concerned with the prominent sporting event of the 2012 UEFA European Football Championship,[2] commonly referred to as Euro 2012. Apart from setting a new record on Twitter, Euro 2012 has set a record for both the highest aggregate attendance (1,440,896) and the highest average attendance per game (46,481) under the 16-team format (since 1996).[3]

5.1. Dataset

We consider the dataset obtained for the 2012 European Football Championship final played between Spain and Italy on July 1, 2012 at 17:45 GMT. This game set a new sports-related record on Twitter where 15,000 tweets per second (TPS) were sent across Twitter platform and a total of 16.5 million tweets were sent during the course of the game,[4] We were able to retrieve about half a million tweets encompassing 3 h starting from the beginning of the game. To reduce the load on the data warehouse, we pre-filtered the input within the BaseX system to obtain the relevant set to be uploaded into the data warehouse.

[2] http://www.uefa.com/uefaeuro/index.html

[3] http://en.wikipedia.org/wiki/UEFAEuro2012

[4] http://www.euro2012.twitter.com

We took advantage of the Twitter's own mechanism of trending topics to identify relevant tweets.

5.2. *Semantic enrichment*

In the original dataset, only two fields, namely *User Description* and *Tweet*, are of type arbitrary text where users can fill in any textual information. *User Description* has a maximum length of 180 characters. However, some users either do not fill in anything or hardly update this field. A *Tweet* field must contain some content with a maximum length of 140 characters. It can also include user names and URLs of external websites, photos and videos. These two lengthy fields are fundamental for the semantic analysis as they deliver valuable information about users and their opinions. These fields can be semantically analyzed along multiple perspectives such as *Sentiment Analysis*, *Entity Extraction*, *Keyword Extraction*, *Event Detection*, and *Topic Selection*. A variety of techniques are available for performing such analysis, such as the ones mentioned in [31–34].

We utilized the services of popular text mining platforms AlchemyAPI [35] & OpenCalais [36]. Both of them offer APIs through which the submitted text can be semantically analyzed according to the specified task with results returned in JSON or ATOM format. Unfortunately, both APIs enforce a daily request rate limit. By employing both services we were able to maximize the throughput. The contents of *User Description* and *Tweet* fields were submitted for semantic enrichment. *User Description* was analyzed only once for any user since its value does not change frequently. This allowed us to save time and get maximum utilization within the request limit. As for the tweets, we distinguish between *new* and *re-tweeted* messages (a Twitter synonym for forwarding content). While new tweets are submitted for semantic analysis, the re-tweets are registered by incrementing the *Re-tweet Count* field. Table 1 shows the distribution of results for the Sentiment Analysis performed on a dataset of 428,735 tweets relevant to the event under consideration.

Table 1
Sentiment analysis statistics.

Sentiment	TweetCount
Negative	27,858
Neutral	74,247
No Sentiment	324,725
Positive	64,731

5.3. *Entity detection*

Entity detection performed on the input dataset is helpful in gaining insights into the content shared by sports lovers who engaged in social interaction during the course of the game. The Entity Detection Model [36] that we used identified as many as 36 entity types despite the fact that the message length is limited to 140 characters. Fig. 9 plots the top 10 detected entities of type *Person* and *Country* while Fig. 10(a) shows all detected entity types. Each tweet was scanned to associate it with a *Topic* from a set of supported topics [36] to provide aggregation and enable insightful analysis. Fig. 10(b) plots the list of all *topics* derived from the dataset and shows the distribution of each topic discussed.

The occurrence of macro- and micro-events also gets reciprocated on social networks and potentially contains important information. Analysts can largely benefit from the set of semantic enrichment methods and can leverage the information extracted using *Entity & Event Detection* to offer more – and potentially useful – insights to the users' views. One such example is to investigate how Twitter users reacted to the event of scoring a goal. We put together sentiment analysis and entity detection to see the reaction of Twitter users on the players involved in the micro-event of scoring a goal. Fig. 11(a) shows sentiments for the top mentioned players right after the first goal was scored. Fig. 11(b) depicts sentiments across the top mentioned players right after the second goal.

5.4. *Semantic enrichment across social engagement*

Social engagement represents the user's activity directly triggered by a social action of another user. The Twitter

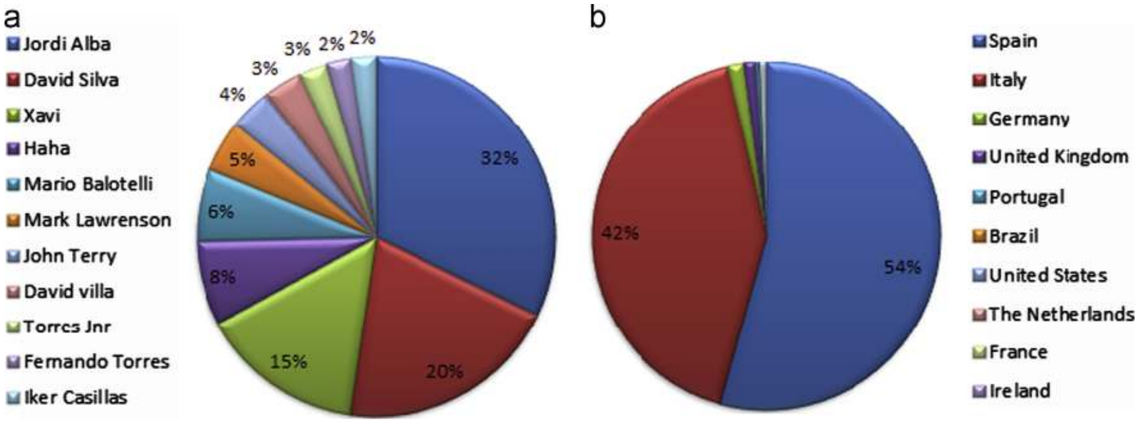

Fig. 9. Entity detection: top 10 entities. (a) Person and (b) country.

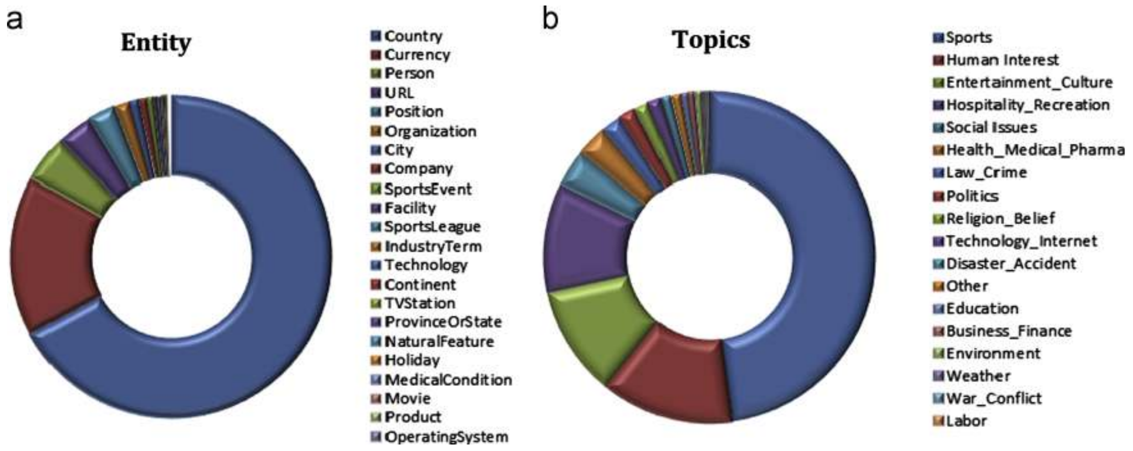

Fig. 10. Distribution of (a) entities and (b) topics.

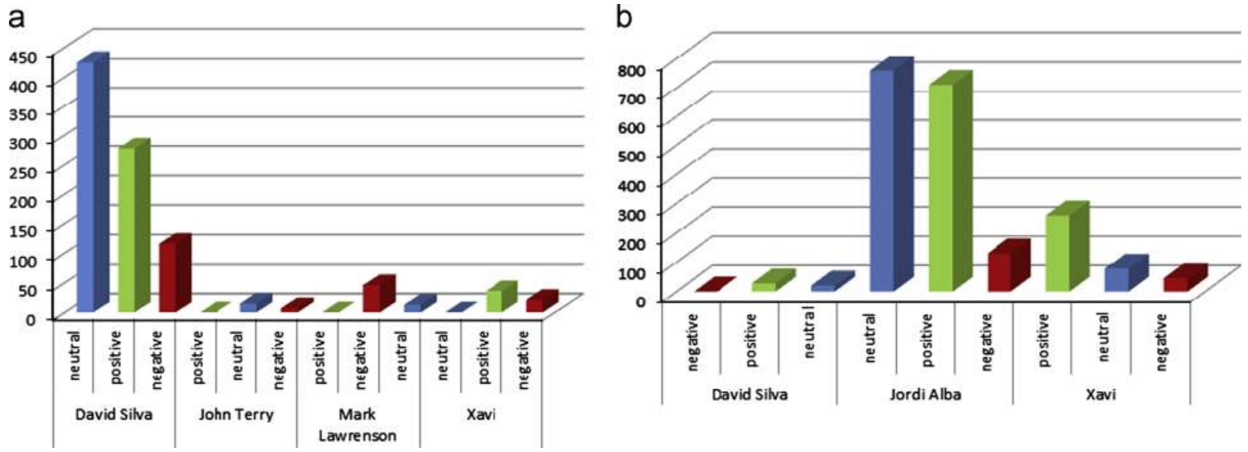

Fig. 11. Sentiment distribution for top players tweeted after (a) first and (b) second goal.

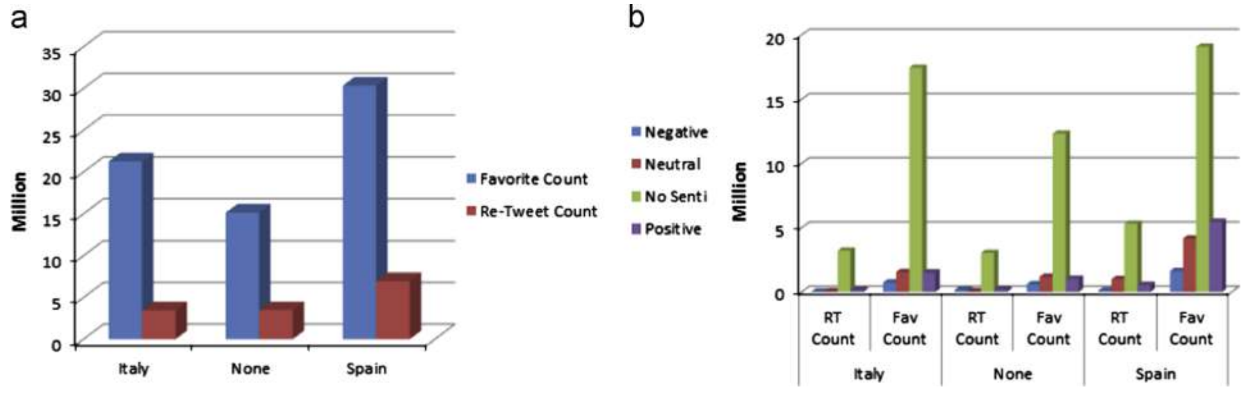

Fig. 12. (a) Distribution of Tweets by sentiment and (b) sentiment distribution across teams.

terminology for social engagements includes *Favorite*, *Retweet* and *Reply-To*. A tweet may trigger none or any combination of these engagements. Fig. 12(a) plots the sum of social engagement for Favorite-Count and Retweet-Count across team orientation of Twitter users. Fig. 12(b) plots similar statistics with the addition of sentiments across each team. This chart shows tweets which received such social actions from Twitter users across the sentiment. A Retweeted message is shared directly with all followers of the given user and, therefore, contributes to trending or popularity of the same message and its content. We employed Favorite-Count and Retweet-Count as measures in our OLAP cube along with other derived measures, whereas topic, entities, events, etc., were modeled as dimensions of those measures. The obtained perspective enables discovery of local and global popular topics, personalities, subjects, events, etc. by exploring the cube along the respective dimensions. Fig. 12(a) is a small reflection of such an exploration depicting popularity of the teams. Tweets for which team support could not be derived are also represented in this chart. Fig. 12(b) plots similar statistics along sentiments and enables analysts to see whether sentiments of the tweet contributed to popularity. The above scenarios of applying sentiment analysis and event detection on the raw textual data demonstrate the opportunities of discovering a multidimensional structure in an unstructured or poorly structured set, thus, making the data analyzable with the established OLAP technology.

6. Conclusions and future work

In this work we proposed to extract multidimensional data cubes for OLAP from semi-structured datasets and to extend the resulting model by including dynamic categories and hierarchies discovered from the data through DM methods and other computations. The discovered classifications reflect “hidden” relationships in the dataset and thus represent new axes for exploring the measures in a cube.

As a non-conventional application for OLAP, we used the publicly available stream of the user-generated data provided by the Twitter platform. Tweeted messages streamed as semi-structured records with over 60 fields can be enriched with additionally extracted characteristics relevant for the analysis. We considered various sources of enriching the original set, from external services and APIs, to derivation from existing characteristics and application of knowledge discovery techniques.

We handled the process of adding discovered categories at the conceptual and logical level and investigated which approaches to implement slowly changing dimensions that are suitable for our scenario. The method of storing only the current state in the dimension table and extracting the previous versions into a history table proved to be the appropriate solution that ensures historically correct aggregation but also enables deliberate historically incorrect aggregation useful for investigating the data evolution itself.

Our approach was tested on the dataset of the Twitter's public stream with a focus on getting more insight into the content. We presented examples of adding a sentiment dimension coupled with topic, entity and event detection. When a usage scenario is limited to a particular event, entity detection can be topped up by introduction of ad hoc hierarchies, such as grouping players by team and country or grouping politicians by party.

Our future work aims at designing a more generic framework for obtaining an enhanced multidimensional perspective of semi-structured data. Staying within the Twitter scenario, we are interested in further investigation of discovering dimensions with entity and event detection methods, and, more specifically, on enabling ad-hoc aggregation hierarchies for such discovered dimensions. The use of knowledge discovery techniques for detecting structural elements in the raw input data appears to be a promising direction for adaptive and comprehensive multidimensional analysis of heterogeneous data volumes.

References

[1] K. Roebuck, Big Data: High-Impact Strategies—What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors, Emereo Pty Limited, 2011.

[2] C. Strauch, Nosql Databases. Lecture Selected Topics on Software-Technology Ultra-Large Scale Sites, Manuscript, Lecture Notes, Stuttgart Media University, 2011.

[3] J. Han, Olap mining: an integration of olap with data mining, in: Proceedings of the 7th IFIP 2.6 Working Conference on Database Semantics (DS-7).

[4] Twitter Team, Twitter Turns Six, 2012.

[5] R. Krikorian, Developing for @twitterapi (techcrunch disrupt hackathon), 2012.

[6] J. MacLennan, Z. Tang, B. Crivat, Mining OLAP Cubes, Wiley Publishing, 2008, pp. 429–431.

[7] M. Usman, S. Asghar, S. Fong, A conceptual model for combining enhanced olap and data mining systems, in: 5th International Joint Conference on INC, IMS and IDC, 2009 (NCM'09), IEEE, pp. 1958–1963.

[8] J. Han, S. Chee, J. Chiang, Issues for on-line analytical mining of data warehouses, in: Proceedings of the Workshop on Research Issues on Data Mining and Knowledge Discovery, Seattle, Washington, pp. 2:1–2:5.

[9] H. Zhu, On-line analytical mining of association rules (Ph.D. thesis), Simon Fraser University, 1998.

[10] S. Dzeroski, D. Hristovski, B. Peterlin, Using data mining and olap to discover patterns in a database of patients with y-chromosome deletions., in: Proceedings of the AMIA Symposium, American Medical Informatics Association, p. 215.

[11] F. Dehne, T. Eavis, A. Rau-Chaplin, Coarse grained parallel on-line analytical processing (OLAP) for data mining, in: International Conference on Computational Science (ICCS 2001), 2001, pp. 589–598.

[12] T.B. Pedersen, C.S. Jensen, C.E. Dyreson, A foundation for capturing and querying complex multidimensional data, Inf. Syst. 26 (2001) 383–423.

[13] A. Abelló, J. Samos, F. Saltor, A framework for the classification and description of multidimensional data models, in: DEXA 2001: Proceedings of DEXA 2001, Springer-Verlag, 2001, pp. 668–677.

[14] S. Lujn-Mora, J. Trujillo, I.-Y. Song, A UML profile for multidimensional modeling in data warehouses, Data Knowl. Eng. 59 (2006) 725–769.

[15] S. Mansmann, Extending the OLAP technology to handle non-conventional and complex data (Ph.D. thesis), University of Konstanz, Konstanz, Germany, 2008.

[16] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. Sen Sarma, R. Murthy, H. Liu, Data warehousing and analytics infrastructure at facebook, in: Proceedings of the 2010 International Conference on Management of Data (SIGMOD'10), ACM, New York, NY, USA, 2010, pp. 1013–1020.

[17] H. Kwak, C. Lee, H. Park, S. Moon, What is twitter, a social network or a news media?, in: Proceedings of the 19th International Conference on World Wide Web (WWW'10), ACM, New York, NY, USA, 2010, pp. 591–600.

[18] A. Java, X. Song, T. Finin, B. Tseng, Why we twitter: understanding microblogging usage and communities, in: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis, ACM, pp. 56–65.

[19] M. Mathioudakis, N. Koudas, Twittermonitor: trend detection over the twitter stream, in: Proceedings of the 2010 International Conference on Management of Data, ACM, pp. 1155–1158.

[20] J. Chen, R. Nairn, L. Nelson, M.S. Bernstein, E.H. Chi, Short and tweet: experiments on recommending content from information streams., in: Proceedings of CHI, ACM, pp. 1185–1194.

[21] O. Phelan, K. McCarthy, B. Smyth, Using twitter to recommend real-time topical news, in: Proceedings of the 3rd ACM Conference on Recommender systems, ACM, pp. 385–388.

[22] M. Michelson, S.A. Macskassy, Discovering users' topics of interest on twitter: a first look, in: Proceedings of the 4th Workshop on Analytics for Noisy Unstructured Text Data (AND 2010), Toronto, Ontario, Canada, 26th October, 2010 (in conjunction with CIKM 2010), ACM.

[23] B. Hecht, L. Hong, B. Suh, E.H. Chi, Tweets from Justin Bieber's heart: the dynamics of the location field in user profiles, in: Proceedings of CHI, pp. 237–246.

[24] S. Chaudhuri, U. Dayal, V. Ganti, Database technology for decision support systems, Computer 34 (2001) 48–55.

[25] R. Kimball, The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses, John Wiley & Sons, Inc., New York, NY, USA, 1996.

[26] E. Saddad, A. El-Bastawissy, M. Rafea, O. Hegazy, Multiversion queries in multidimensional structures, in: Proceedings of the 6th International Conference on Informatics and Systems (INFOS'08), 2008, pp. DB42–DB50.

[27] A. Holupirek, C. Grün, M.H. Scholl, BaseX & DeepFS joint storage for filesystem and database, in: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology (EDBT'09), ACM, 2009, pp. 1108–1111.

[28] M. Golfarelli, D. Maio, S. Rizzi, The dimensional fact model: a conceptual model for data warehouses, Int. J. Coop. Inf. Syst. 7 (1998). 215–247.

[29] N.U. Rehman, S. Mansmann, A. Weiler, M.H. Scholl, Building a data warehouse for twitter stream exploration, in: Proceedings of 1st IEEE ACM International Workshop on Multi-agent Systems and Social Networks (MASSN 2012), in conjunction with ASONAM 2012.

[30] R. Kimball, M. Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley & Sons, Inc., New York, NY, USA, 2002.

[31] R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, X. Luo, N. Nicolov, S. Roukos, T. Zhang, A statistical model for multilingual entity detection and tracking, in: HLT-NAACL.

[32] S. Petrovic, M. Osborne, V. Lavrenko, Streaming first story detection with application to twitter, in: Proceedings of NAACL, vol. 10, Citeseer.

[33] T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in: Proceedings of the 19th International Conference on World Wide Web, ACM, pp. 851–860.

[34] M. Thelwall, K. Buckley, G. Paltoglou, Sentiment in twitter events, J. Am. Soc. Inf. Sci. Technol. 62 (2011) 406–418.

[35] AlchemyAPI, Alchemyapi: Transforming Text into Knowledge ⟨http://www.alchemyapi.com⟩, 2008 (last checked 05.03.2013).

[36] T. Reuters, Opencalais: A Toolkit for Semantic Enrichment ⟨http://www.opencalais.com⟩, 2008 (last checked 05.03.2013).