Ersch. in: 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE Computer Society. - 1 (2012). - S. 1341-1348. - ISBN 978-0-7695-4799-2, Istanbul, Turkey, August 2012, IEEE Press.

2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Building a Data Warehouse for Twitter Stream Exploration

Nafees Ur Rehman*, Svetlana Mansmann, Andreas Weiler, Marc H. Scholl
University of Konstanz, Germany
Email: {nafees.rehman, svetlana.mansmann, andreas.weiler, marc.scholl
@uni-konstanz.de}

***Abstract*—In the recent year Twitter has evolved into an extremely popular social network and has revolutionized the ways of interacting and exchanging information on the Internet. By making its public stream available through a set of APIs Twitter has triggered a wave of research initiatives aimed at analysis and knowledge discovery from the data about its users and their messaging activities.**

While most of the projects and tools are tailored towards solving specific tasks, we pursue a goal of providing an application-independent and universal analytical platform for supporting any kind of analysis and knowledge discovery. We employ the well established data warehousing technology with its underlying multidimensional data model, ETL routine for loading and consolidating data from different sources, OLAP functionality for exploring the data and data mining tools for more sophisticated analysis. In this work we describe the process of transforming the original stream into a set of related multidimensional cubes and demonstrate how the resulting data warehouse can be used for solving a variety of analytical tasks. We expect our proposed approach to be applicable for analyzing the data of other social networks as well.

I. Introduction and Motivation

Explosion of social network activity in the recent years has lead to generation of massive volumes of user-related data, such as status updates, messaging, blog and forum entries, etc, which, in its turn, has given birth to a novel area of data analysis, namely *Social Media Analysis*. Companies and institutions worldwide anticipate to gain valuable insights from obtaining access to such data and hope to improve their marketing, customer services and public relations with the help of the acquired knowledge. The results of social media analysis are incorporated in e-commerce sites and social networks themselves in form of personalized content, such as recommendations, suggestions, advertisement, etc.

This work is dedicated to providing a data warehouse (DW) solution for hosting the public data stream of Twitter (http://twitter.com/) messaging for the purpose of its comprehensive analysis. We will demonstrate how the analysis of social media can benefit from this established and mature technology. Most of the Twitter-related analysis tools are developed for solving specific tasks known at design time. These tasks include but are not limited to trend discovery, content enrichment, user profiling, topic-based clustering, sentiment analysis, etc. Our work distinguishes itself from the above kind of projects by pursuing a more generic and application-independent perspective on the data. This perspective is achieved by transforming the data into a set of points in a multidimensional space. The benefit of having such a consolidated and standardized data set is the ability to explore the latter with existing tools for data analysis, visualisation and mining. The remainder of the introduction is dedicated to the main components of our solution, namely Twitter as the underlying data source and the data warehousing as the employed technology.

A. Why Twitter?

Twitter is an outstanding phenomenon in the landscape of social networking. Initially introduced 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 the means of real-time information exchange. It has revolutionized the culture of interacting and exchanging information on the Internet and impacted various areas of human activity, such as organization and execution of political actions, crime prevention, disaster management, emergency services, etc. Apart from its attractiveness as a means of communication – with over 140 million active users as of 2012 generating over 340 millions tweets daily [1] – 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.

B. Data Warehousing and OLAP

Data Warehouses (DW) and OnLine Analytical Processing (OLAP)[2] tools are used in Business Intelligence (BI) applications and beyond to support decision-making processes. This technology originated in the early 90s as a response to the problem of providing all the key people in the enterprise with access to whatever level of information they need for decision making [3]. Data warehousing is a specialization of the database technology for integrating, accumulating and

*This work is partially supported by DFG Research Training Group GK-1042 "Explorative Analysis and Visualization of Large Information Spaces"

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

978-0-7695-4799-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ASONAM.2012.230

analyzing data from various sources. It employs the multidimensional data model, which structures the data into cubes containing measures of interest characterized by descriptive properties drawn from a set of dimensions. OLAP tools provide means to query and to analyze the warehoused information and produce online statistical summaries (reports) at different levels of detail. These summaries are computed using aggregate functions (e.g. `SUM`, `AVG`, `MIN`, `MAX`, `COUNT`, etc.). Users can explore multidimensional cubes by performing OLAP operations (e.g., roll-up, drill-down, pivot, rank, etc.). Data mining functionality has also become an integral part of any mature DW system. The former enables automatic discovery of correlations and causal relationships within the data and thus enriches the original data set with additional characteristics.

Applicability of data warehousing is by no means restricted to business scenarios. Comprehensive data analysis has become indispensable in a variety of real-world applications with data warehouses being deployed in non-business domains, such as government, science, education, research, medicine, to name the prominent ones. In our previous co-authored works we applied data warehousing to the academic field of managing student enrollments [4] and to the medical field of surgical workflow analysis [5].

C. Related Work

Social networks are a rather new phenomenon on the web, but their rapid expansion and extreme populariry have confronted the underlying backend architectures with unprecedented volumes of user-generated content. Data warehousing technology has established itself as the leading solution for large-scale data management and analysis. 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 [6] .

Twitter launched in 2006 is only 6 years old. The first quantitative study on Twitter was published in 2010 by Kwak et al. [7] 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. Industrial applications are mostly marketing oriented, while other Twitter analysis works focus on improving the search and navigation in a huge flow of messages as well as on discovering valuable information about the contents and the users. We are more interested in the latter types of works as we pursue a multi-purpose analysis approach.

In 2007 Java et al. [8] 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 [9] 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. [10] and Phelan et al. [11]. 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 [12] discover main topics of interest of Twitter users from the entities mentioned in their tweets. Hecht et al. [13] analyze unstructured information in the user profile's location field for location-based user categorization.

Explosion of Twitter-related research confirms the recognized potential for knowledge discovery from its data. While other contributions focus on mining or enhancing the contents of tweets, improving the frontend or generating meaningful recommendations, we exploit the advantages of the established OLAP technology coupled with data mining to enable aggregation-centric analysis of the meta-data about the Twitter users and their messaging activity.

II. 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. Figure 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, taxonomies, 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

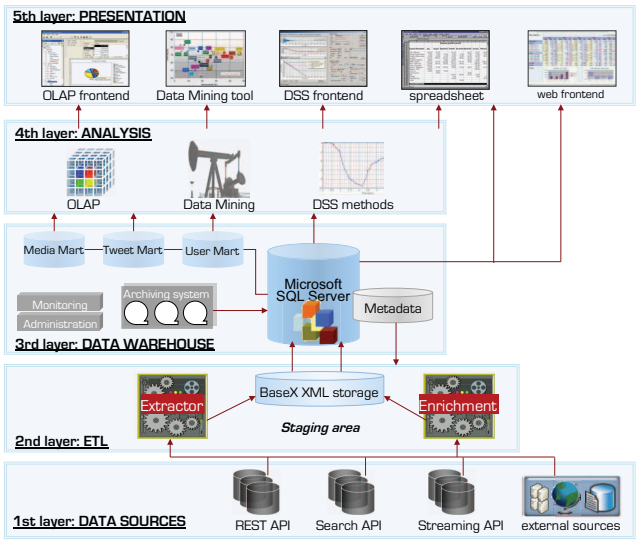

Fig. 1. Multi-layered architecture of the Twitter data warehouse system

dataset into the DW. The dataset delivered by the Twitter Streaming API is semi-structured using the JavaScript Object Notation (JSON) as its output format. Each tweet is streamed as a JSON 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. 10 % of the total public stream provided by the Streaming API covers more than one million tweets per hour, which is a heavy load of data even for a high performance data warehouse system.

Our solution to coping with such a massive data stream is to convert every single streamed object into an XML structure and buffer it in a native XML database BaseX [14] developed within our working group.

The following XML snippet shows an excerpt of a tweet object:

```
<tweet>
  <text>
    Earthquake with the.scale of 8.9 magnitude
    #PrayForIndonesia #PrayForSumatera
  </text>
  <date>Wed Apr 11 08:57:02 +0000 2012</date>
  <source>web</source>
  <retweeted>false</retweeted>
  <user>
    <name>Miley ***</name>
    <date>Tue Jun 22 08:33:12 +0000 2010</date>
    <statuses_count>13101</statuses_count>
    <followers_count>1019</followers_count>
  </user>
</tweet>
```

With a highly efficient BaseX storage [15] we are able to buffer the entire streamed dataset, which would not be possible by buffering directly to the relational database. The usage of BaseX as a data buffer also brings the advantage of selective loading of the new data into the DW by setting up a filter on the input stream and discarding irrelevant parts of the stream without loading them into the target database. For example, for the usage scenario presented later in this work, we filtered the data to retrieve the tweet records from a specific hour only those records matching any of the 25 Trending Topics[2] published by Twitter for that hour.

It is also important at this stage that the buffered output of different APIs can be combined into a single dataset and the data can also be enriched with the information from other sources. For instance, we apply reverse geocoding to enhance the geographic characteristic of each individual tweet with the corresponding city, country, and continent values. In the next step, our ETL routine extracts the XML data into the target star schema model of the DW described in the next section.

The core layer of the system is the actual DW. We employed the Microsoft SQL Server with its powerful set of analysis services including OLAP and data mining. The consolidated dataset in the database provides the basis for defining 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. Due to the standardization of the relational DW technology implemented by the Microsoft SQL Server it is possible to connect front-end tools of various commercial and open-source providers.

The main challenge of implementing a DW for Twitter analysis lies in providing a mapping of the semi-structured original data stream delivered by the Twitter APIs into a rigidly structured multidimensional data set. This mapping should be fully automated to enable continuous insertion of new data into the DW. We proceed by investigating the data model behind Twitter and the format in which it is streamed and describe its transformation into multidimensional cubes.

III. Multidimensional data model for Twitter

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 [16], 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 characteristics (nick-name, 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 can be adapted for multidimensional aggregation. One data record in the stream encompasses a single tweet event stored as the message itself (content and metadata) along with a detailed description of the authoring user's profile in terms of various activity counters. The provided dataset already displays some favorable characteristics for data warehousing, such as being *temporal* (by including the time dimension), *non-volatile* (no modifications of existing entries), and *measure-centric* (maintaining accumulative counters). However, the multidimensional data model and its relational mapping as a star or snowflake schema come with a set of further constraints, such as homogeneity, atomicity, summarizability, avoidance of NULL values, etc., which are not met by the input dataset.

[2]https://dev.twitter.com/docs/api/1/get/trends/daily

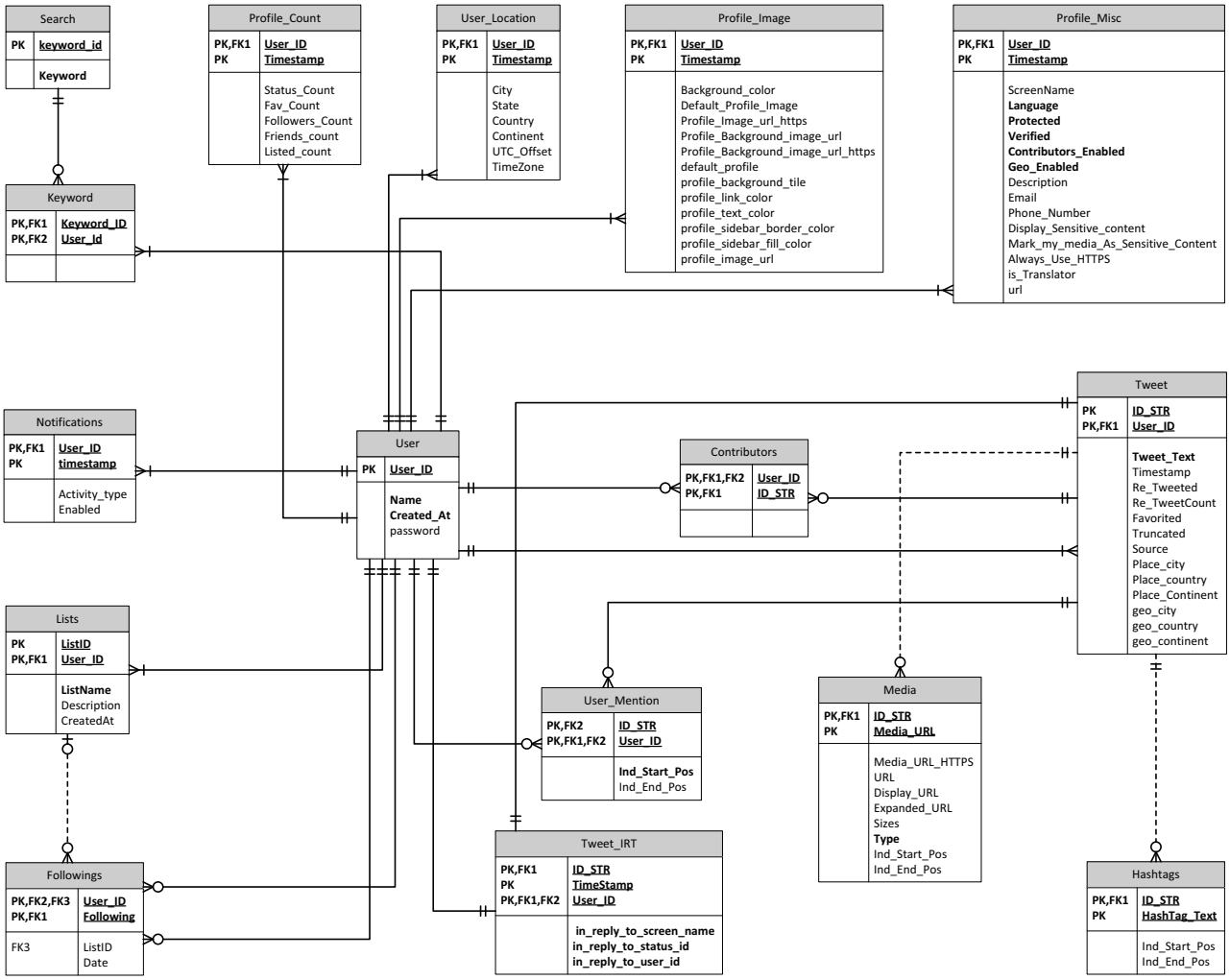

Fig. 2. Conceptual model of the Twitter stream as a UML Diagram

Another observation is that the total of 67 data fields in a tweet entry is a rather small number of attributes for defining a set of measures and and dimensions for a comprehensive analysis. Therefore, we seek to enrich this data structure by additional features, either extracted from external sources or acquired from the available fields by applying various methods and functions from simple computations to complex techniques of knowledge discovery.

We obtain an OLAP-conform multidimensional pespective of the Twitter stream via a series of transformation steps.

A. Relational View of a Tweet Record

The initial step is to get a structured view of the semi-structured record in the original stream. The purpose of this step is to identify the available entitites, their attributes value domains, constraints, and relationships betwen entities. A relationship is specified in terms of the cardinalities for each participating entity. We use the UML notation to represent the identified structural elements. Figure 2 shows the results of the relational mapping as a set of relations linked by foreign key constraints.

The main classes are obviously the *user* and the *tweet*, whereas all other elements are related to either or both of them. User-related characteristics encompass the profile information i.e., the image, the location, the searches performed, the notifications received, and the statistics about the user's interaction. Statistics about the user interation is run as accomulative counters on the followers and following others, status updates, and friendships. Users are related to one another through following (i.e., receiving the other user's tweets) either directly or via a user-defined list, also known as channel.

Tweet-related characteristics include the location and the source of tweeting, the hashtags used, other users mentioned, media embedded, as well as statistics on re-tweeting and favoriting the tweet. The relationships between users and tweets can be that of authoring/retweeting the message, contributing

to it[3], or being mentioned in the message.

B. Multidimensional View of a Tweet Record

Data in a DW is structured according to the aggregation-centric multidimensional data model, which uses numeric measures as its analysis objects [17]. 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 rollup operations. A 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 multidimensional 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 data warehouse context. In our work, however,it is imperative to distinguish between those terms to enable facts without measures. According to Kimball [18], 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 represented by a combination of simultaneously occurring characteristics. Kimball proposed to refer to such scenarios as *factless fact tables*[18]. Mansmann [19] suggests to use a more implementation-independent and less controversial term *non-measurable fact type*.

Back to Twitter, its data model contains only a small set of numeric attributes, which can be treated as measures. These attributes encompass the counters in the user profile and the tweet record. Other attributes are of descriptive nature and, therefore, should be mapped to dimensions or dimension hierarchies. Our approach is to treat a tweet event as a 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 parallel hierarchies. At the stage of the conceptual modeling, 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 extended Dimensional Fact Model (x-DFM) [19], which makes provisions for various kinds of behaviours in OLAP dimensionsm which is an extension of the Dimensional Fact Model (DFM) of Golfarelli et al. [20]. The x-DFM provides some advanced constructs, such as derived measures and categories, degenerated dimensions and fuzzy hierrarchies, relevant for our model. Figure 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 of a dimension converge in an abstract $\top$ 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.

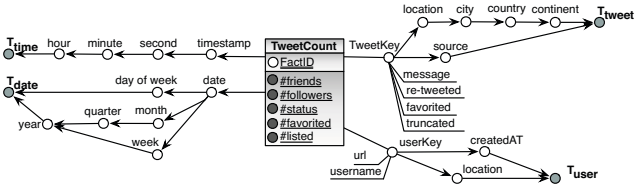

Fig. 3. Fragment of the tweet record in the x-DFM

C. Extending the Original Dataset

In general, a datacube can be extended by adding new elements of type a) *measure*, b) *dimension*, or c) *hierarchy level*. Besides, new datacubes can be defined for accomodating additional data. New elements can be retrieved from the existing ones, from applying some functions or services or can be added by including other data sources.

External functions and data source provide an opprtunity to add completely new dimensions to a datacube. Here are some prominent examples. A useful property of the tweet's language can be added by using a language detection API, such as the one offered by Google or JSON. Another important property to detect is whether a tweet is spam or has malicious content. This can be done by employing the APIs of Askimed and Defensio or another similar service. We are currently working on integrating the above language and spam detection features to enable comprehensive text analysis of user-generated content.

Adding a new element by computing its values from the existing fields can be rather trivial if the computation is based on the values within the same fact record. For example, we could add a category *author_type* with values *vip* and *standard*, computed from the measures *#friends* and *#followers*. A tweet fact is assigned the value *vip* whenever the followers counter significantly outnumbers the friends counter and *standard* otherwise. Even those such attributes can be computed at query time, it is a common OLAP practice to materialize such field to make them explicitely available as aggregation paths or measures of the analysis.

As for discovering less obvious relationships in the dataset, data mining algorithms provide the necessary functionality. These algorighms proceed by analyzing the whole set in order to build clusters or associations or to discover rules. In data warehousing, data mining tools are typically employed at the front-end layer to gain new insights into the data and to use

[3]The *contributor* feature is currently unavailable.

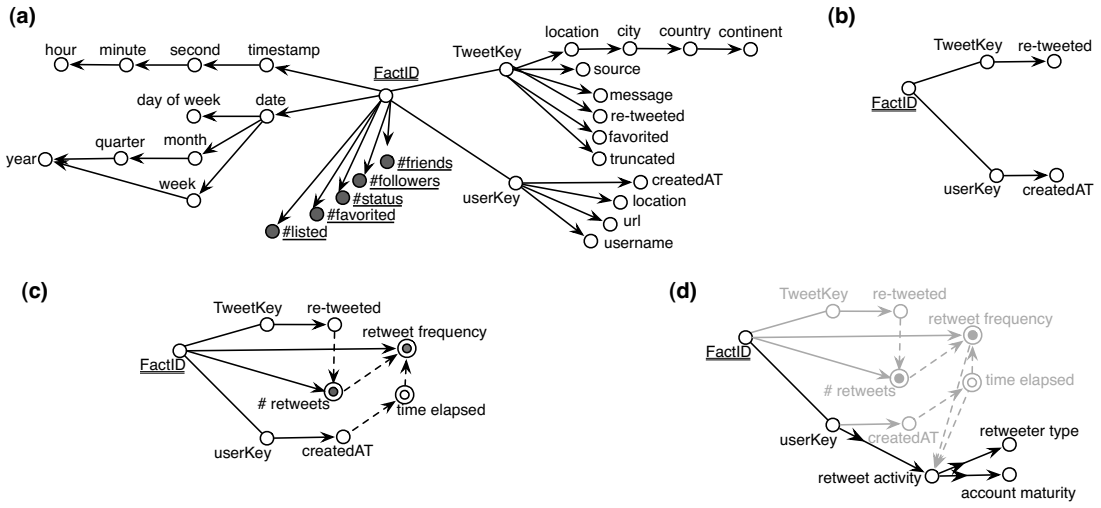

Fig. 4. Stages of acquiring new hierarchy levels

their output for reporting or decision making. In our approach, however, data mining algorithms are applied at the backend in order to discover clusters or rules useful for extending the input data sets and the models of the available cubes. For this purpose, the input set has to be transformed into a more generic representation than the one offered by the multidimensional model. The goal is to treat all elements symmetrically as potential input fields for discovering new categories. To achieve this, we transform the graph model of the cube as to get rid of different types of nodes and edges based on the observation that all edges are of type "many-to-one" or even "one-to-one", i.e. can be represented by "rolls-up-to" edges, and that all nodes are of type attribute. Figure 4 (a) shows the transformed graph from Figure 3. This generic view is suitable for generating the input set for data mining algorithms by selecting a subgraph with relevant characteristics and retrieving its data into a pre-joined view.

Let us consider an example of adding a new complex category *re-tweet activity* to the *user* dimension reflecting the frequency of re-tweeting relative to the period ellapsed since the creation of the user's account. This category should assign each user into one of four clusters: *mature and active*, *new and active*, *mature and passive*, and *new and passive* respectively for those users who registered long ago or recently and who re-tweet more or less frequently, respectively. Neither the time elapsed since the user registration nor the frequency of re-tweeting are explicit in the dataset, but both of them can be computed from other data fields.

Subfigures (b), (c), and (d) in Figure 4 demonstrate the process of computing the necessary attributes for adding the cluster-based category *re-tweet activity*. The subgraph relevant for performing this task is given in Figure 4 (b). The graph in Figure 4 (c) shows the derivation of categories required for clustering, namely, *time elapsed* as the difference between the current and the account creation date, cumulative measure *# retweets* as the number of messages with the *re-tweeted* value set to true, and, finally, *retweet frequency* as *# retweets* divided *time elapsed*. Finally, Figure 4 (d) shows the the result of adding *re-tweet activity* as a hierarchy level to the *user* dimension. Note that in the conceptual model, a discovered or derived category can be treated just as a normal one. For instance, we added parallel hierarchy levels *retweeter type* with member values *active* and *passive* and *account maturity* with member values *new* and *mature* on top of *re-tweet activity*. Once a discovered element has been added to the model and populated with values, it can be used in OLAP queries in the same manner the static elements of the same type are used. Due to limited space, we skip further details of maintaining discovered elements of OLAP cubes.

IV. Experiments and Evaluation

In this section we demostrate the power of applying OLAP when solving specific Twitter-related analysis tasks. For the experiments presented in this work we extracted the data spanning three hours with, at least, a mention of a topic from Table I pertaining to the earthquake in Indonesia on April 11, 2012. The task at hand was to learn about the role of Twitter as a communication medium in case of such an emergency.

Social media in general and Twitter in particular have changed the way people socialize and share content on the Internet. Twitter continue to grow at a record pace, with more than 465 million accounts, 140 million active users and more than 340 million Tweets each day[1]. While US leads the way with over 107.7 million users, Twitter is massively popular in Indonesia too. Indonesia's love affair with social media has never been a secret. Indonesia have become the 5th largest home to Twitter with 19.5M users[21].

Indonesia, being on the edges of the Pacific, Eurasian and Australian tectonic plates, makes it the site of numerous volcanoes and frequent earthquakes[22]. Given the social media usage statistics and the fact that it is on the geo fault-lines, it is of interest to analyze how users on Twitter

TABLE I
WORLD-WIDE TRENDING TOPICS - APRIL 11, 2012

No.	Trending Topic
1	#PrayForSumatera
2	#sumatra
3	#tsunami
4	#10favouritebands
5	Earthquake in Indonesia
6	Magnitude 8.9
7	Sumatra

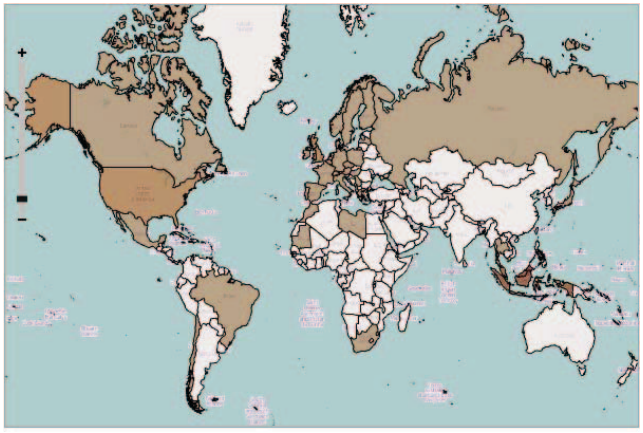

Fig. 5. Tweet activity across the world about the earthquake

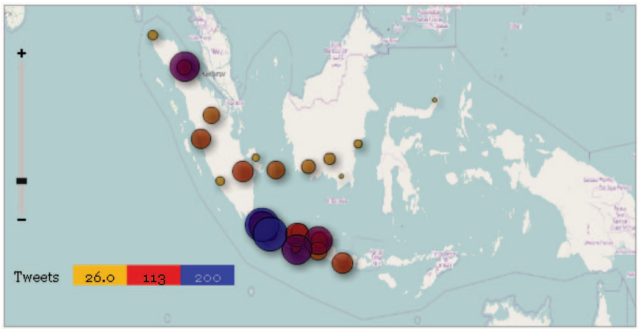

Fig. 6. City-wise tweet activity in Indonesia

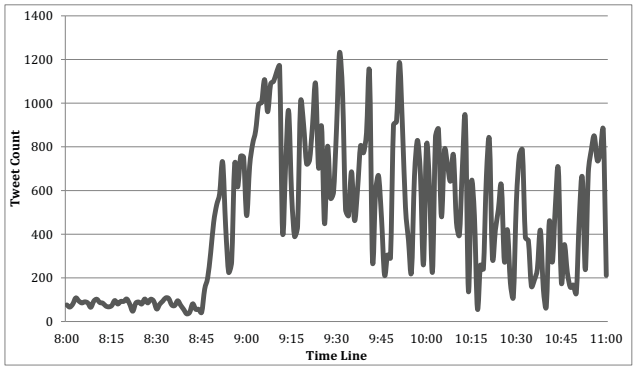

Fig. 7. Tweet activity across the Time-Line during the earthquake

spread the news in case of an earthquake or tsunami. For this purpose, we consider the most recent case of an earthquake in Indonesia. An earthquake with a magnitude of 8.6 struck south west of Banda Aceh Sumatra on Wednesday, April 11, 2012 at 08:38:37 UTC[23]. Following the earthquake, tsunami warnings were issued in Indonesia and across the world, however, the latter were later taken back. For the analysis, we used the streaming API in combination with search API to extract tweets relevant to the trending topics on Twitter, as listed in Table I. The tweets recorded are from 08:00:00 AM UTC to 11:00:00 AM UTC.

There are about 86,000 tweets in the dataset. Based on this set, we performed a series of analysis tasks. The first task relates to the number of tweets originated across the world with a mention of Indonesia's earthquake. The top 4 countries from where most tweets originated include Indonesia, Malaysia, UK and USA with 4283, 1060, 475 and 442 tweets, respectively. Over all, about 73 countries tweeted about this earthquake in the first two hours. Figure 5 plots this information. Figure 6 depicts the top 25 tweeting cities of Indonesia. Jakarta is on the top with 1011 tweets.

Figure 7 plots the Tweet count of the obtained dataset onto a Time Series chart where Time-Line on the x-axis plots the time with an interval of 15 minutes and the Tweet count is mapped to the y-axis. Tweet count is mostly below 100 from 08:00 till 8:45 AM with a sharp increase from 8:45 onwards. The earthquake struck at 08:38:37 AM and it took about 5-7 minutes to become a world-wide trending topic. The Indonesian Meteorological, Climatological and Geophysical Agency or BMKG[24] tweeted 15 times about this earthquake and tsunami and the collective re-tweet count for all of its tweets is 22086, while interestingly, a single tweet from a Canadian celebrity about the same topic got retweets of more than 20889 times. This surely contributed to turning the topics in Table I into a world-wide trend. A pie chart in Figure 8 shows the distribution of tweets across the Source device for top 7 countries. A source can be web (Twitter.com), Mobile Application or Twitter Clients. Indonesia-Mobile and Indonesia-Web make 46% and 20%, respectively, of all tweets. This is quite opposite to the statistics presented in [21] where only 16% of users access Twitter using Mobile Application. One explanation, particular to this case, could be that many would have vacated buildings/homes soon after the earthquake and used their cell phones to tweet about it. Since there was an after shock, as strong as the earthquake, it might have also made people to stay outside, while many were on the run to safer places, as reported in the news.

All the numbers and the figures presented in this section were obtained using the Analysis Services Toolkit of the Microsoft SQL Server. It offers a user-friendly interface for interactive visual exploration the data and generation of visual representations. The analyst navigates through the elements (measures and dimensions) of the cubes and drags the elements of interest to the data presentation area. No knowledge of the underlying query language or physical characteristics of the data is necessary to use such an interface. Any other analysis

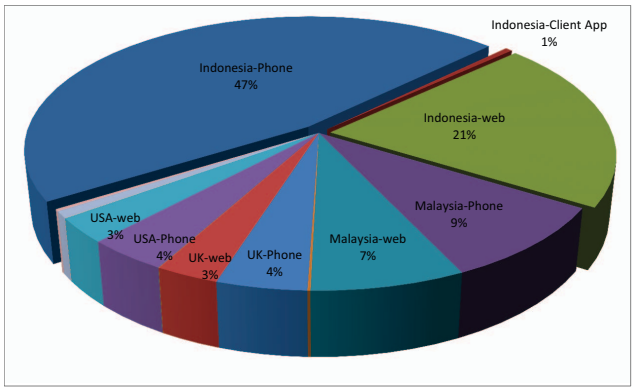

Fig. 8. Tweets by Source Device: Top 5 countries

task based on the data provided through the Twitter API can be solved in a similar fashion.

V. Conclusions and Future Work

We applied the data warehousing technology to enable comprehensive analysis of massive data volumes generated by the social network Twitter. Traditionally, DW store historical data in an aggregation-centric fashion, where the source data undergoes a series of transformations to be consolidated and pre-aggregated to a particular level of detail. However, in case of Twitter, used primarily as a means of spreading news, most analytical tasks are concerned with accessing the most recent data in a near real-time. This imposes additional requirements on the data model as well as on the continuous loading of new data streamed by the Twitter API.

We presented the multi-layered system architecture of our implementation focusing on the critical stage of transforming the original stream into a structured multidimensional dataset consisting of measures and dimensions. We also elaborated on various options of enriching the dataset and its structure by means of derivation, data mining, linking to addional sources or using external APIs for detecting new features. Finally, we demonstrated the power of our approach by solving a series of tasks related to the analysis of Twitter usage patterns during the recent earthquake in Indonesia. Since the DW accumulates the entire data streamed by Twitter, the former can be used in a similar fashion for solving any tasks based on aggregating or mining that data.

Our project on building a data warehouse for Twitter is rather new and the directions for future work are manifold. One promising direction is to enable contents analysis of tweets by building corresponding keyword indices, enabling language detection and translation as well as spam filtering. Another work in progress is event and entity detection by importing the corresponding services of Yahoo, Wikipedia and others. In terms of performance, our goal is to provide a near real-time analysis by optimizing the loading of the new data.

References

[1] Twitter Team, "Twitter turns six," 2012. [Online]. Available: http://blog.twitter.com/2012/03/twitter-turns-six.html

[2] E. F. Codd, S. B. Codd, and C. T. Salley, "Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate," E.F.Codd & Associates, Tech. Rep., 1993.

[3] K. Orr, "Data warehousing technology," The Ken Orr Institute, 1996, White Paper.

[4] S. Mansmann and M. H. Scholl, "Decision Support System for Managing Educational Capacity Utilization," *IEEE Transactions on Education*, vol. 50, no. 2, pp. 143–150, 2007.

[5] T. Neumuth, S. Mansmann, M. H. Scholl, and O. Burgert, "Data Warehousing Technology for Surgical Workflow Analysis," in *CBMS 2008: Proceedings of the 21st IEEE International Symposium on Computer-Based Medical Systems*. Jyvskyl, Finland: IEEE Computer Society, 2008, pp. 230–235.

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

[7] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 591–600.

[8] A. Java, X. Song, T. Finin, and 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, 2007, pp. 56–65.

[9] M. Mathioudakis and 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.

[10] J. Chen, R. Nairn, L. Nelson, M. S. Bernstein, and E. H. Chi, "Short and tweet: experiments on recommending content from information streams." in *Proc. CHI*. ACM, 2010, pp. 1185–1194.

[11] O. Phelan, K. McCarthy, and B. Smyth, "Using twitter to recommend real-time topical news," in *Proceedings of the third ACM conference on Recommender systems*. ACM, 2009, pp. 385–388.

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

[13] B. Hecht, L. Hong, B. Suh, and E. H. Chi, "Tweets from justin bieber's heart: the dynamics of the location field in user profiles." in *Proc. CHI*. ACM, 2011, pp. 237–246.

[14] A. Holupirek, C. Grün, and 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*, ser. EDBT '09. ACM, 2009, pp. 1108–1111.

[15] C. Grün, A. Holupirek, M. Kramis, M. H. Scholl, and M. Waldvogel, "Pushing xpath accelerator to its limits," in *ExpDB*, P. Bonnet and I. Manolescu, Eds. ACM, 2006.

[16] R. Krikorian, "Developing for @twitterapi (techcrunch disrupt hackathon)." [Online]. Available: https://dev.twitter.com/docs/intro-twitterapi

[17] S. Chaudhuri, U. Dayal, and V. Ganti, "Database technology for decision support systems," *Computer*, vol. 34, no. 12, pp. 48–55, 2001.

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

[19] S. Mansmann, "Extending the olap technology to handle non-conventional and complex data," Ph.D. dissertation, Konstanz, Germany, 2008.

[20] M. Golfarelli, D. Maio, and S. Rizzi, "The Dimensional Fact Model: A conceptual model for data warehouses," *International Journal of Cooperative Information Systems*, vol. 7, no. 2-3, pp. 215–2–47, 1998.

[21] Inforgraphic and Benenett, Shea, "Just how big is twitter in 2012?" 2012. [Online]. Available: http://www.mediabistro.com/alltwitter/twitter-statistics-2012_b18914

[22] Wikipedia, "Indonesia," 2012. [Online]. Available: http://en.wikipedia.org/wiki/Indonesia

[23] US Geological Survey, "Earthquake off the west coast of sumatra," 2012. [Online]. Available: http://earthquake.usgs.gov/earthquakes/recenteqsww/Quakes/usc000905e.php

[24] BMKG, "Indonesian meteorological climatological and geophysical agency," 2012. [Online]. Available: http://www.bmkg.go.id