# Meta-Stars: Multidimensional Modeling for Social Business Intelligence

Enrico Gallinucci
DISI - University of Bologna
Viale Risorgimento 2
40136 Bologna, Italy
enrico.gallinucci2@unibo.it

Matteo Golfarelli
DISI - University of Bologna
Viale Risorgimento 2
40136 Bologna, Italy
matteo.golfarelli@unibo.it

Stefano Rizzi
DISI - University of Bologna
Viale Risorgimento 2
40136 Bologna, Italy
stefano.rizzi@unibo.it

## ABSTRACT

Social business intelligence is the discipline of combining corporate data with user-generated content (UGC) to let decision-makers improve their business based on the trends perceived from the environment. A key role in the analysis of textual UGC is played by topics, meant as specific concepts of interest within a subject area. To enable aggregations of topics at different levels, a topic hierarchy is to be defined. Some attempts have been made to address some of the peculiarities of topic hierarchies, but no comprehensive solution has been found so far. The approach we propose to model topic hierarchies in ROLAP systems is called meta-stars. Its basic idea is to use meta-modeling coupled with navigation tables and with traditional dimension tables: navigation tables support hierarchy instances with different lengths and with non-leaf facts, and allow different roll-up semantics to be explicitly annotated; meta-modeling enables hierarchy heterogeneity and dynamics to be accommodated; dimension tables are easily integrated with standard business hierarchies. After outlining a reference architecture for social business intelligence and describing the meta-star approach, we discuss its effectiveness and efficiency by showing its querying expressiveness and by presenting some experimental results for query performances.

## Categories and Subject Descriptors

H.2.1 [**Database Management**]: Logical Design; H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision Support*

## Keywords

business intelligence; social media; user-generated content; multidimensional modeling

## 1. INTRODUCTION

The planetary success of social networks and the widespread diffusion of portable devices has contributed, during the last decade, to a significant shift in human communication patterns towards the *voluntary sharing of personal information*. Most of us are able to connect to the Internet anywhere, anytime, and continuously send messages to a virtual community centered around blogs, forums, social networks, and the like. This has resulted in the accumulation of enormous amounts of *user-generated content* (UGC), that include geolocation, preferences, opinions, news, etc. This huge wealth of information about people's tastes, thoughts, and actions is obviously raising an increasing interest from decision makers because it can give them a fresh and timely perception of the market's mood; besides, in many cases the diffusion of UGC is so widespread to directly influence in a decisive way the phenomena of business and society.

Some commercial tools are available for analyzing the UGC from a few predefined points of view (e.g., topic discovery, brand reputation, and topics correlation) and using some ad-hoc KPIs (e.g., topic presence counting and topic sentiment). These tools do not rely on any standard data schema; often they do not even lean on a relational DBMS but rather on in-memory or non-SQL ones. Currently, they are perceived by companies as self-standing applications, so UGC-related analyses are run separately from those strictly related to business, that are carried out based on corporate data using traditional business intelligence platforms. To give decision makers an unprecedently comprehensive picture of the ongoing events and of their motivation, this gap must be bridged.

*Social Business Intelligence* (SBI) is the discipline of effectively and efficiently combining corporate data with UGC to let decision-makers analyze and improve their business based on the trends and moods perceived from the environment. Note that the data to be combined have very different features: while corporate data are structured, reliable, and accurate, UGC is unstructured or poorly structured, possibly fake, often vague and imprecise; however, both types of data are crucial for an effective decision-making process. As in traditional business intelligence, the goal of SBI is to enable powerful and flexible analyses for users with a limited expertise in databases and ICT; this goal is typically achieved by storing information into a data warehouse, in the form of multidimensional cubes to be accessed through OLAP techniques.

In the context of SBI, the category of UGC that most significantly contributes to the decision-making process in the broadest variety of application domains is the one coming in the form of textual *clips*. Clips can either be messages

*DOLAP* '13 October 28, 2013, San Fancisco, CA
Copyright 2013 ACM 978-1-4503-2412-0/13/10
http://dx.doi.org/10.1145/2513190.2513195 ...$15.00.

posted on social media (such as Twitter, Facebook, blogs, and forums) or articles taken from on-line newspapers and magazines. Digging information useful for decision-makers out of textual UGC requires first crawling the web to extract the clips related to a *subject area*, then enriching them in order to let as much information as possible emerge from the raw text. The subject area defines the project scope and extent, and can be for instance related to a brand or a specific market. Enrichment activities range from the simple identification of relevant parts (e.g., author, title, language) if the clip is semi-structured, to the use of either natural language processing or text analysis techniques to interpret each sentence and if possible assign a polarity to it (i.e., *sentiment analysis* or *opinion mining* [5]). Though the issues related to the overall process have been thoroughly investigated in the literature starting from the early 00's and some commercial tools are available to support all or parts of it, the analysis capabilities of the results delivered to end-users are typically very limited: only static or poorly flexible reports are provided, and historical data are not made available. Besides, in standard architectures the flow of textual UGC is separate from the ETL flows carrying business data, which forces an unnatural dividing line within the decision-making process and dramatically reduces its effectiveness.

A key role in the analysis of textual UGC is played by *topics*, meant as specific concepts of interest within the subject area. A first list of relevant topics is normally provided by decision makers and by experts of the subject area, to be then iteratively refined and enriched by analyzing the dynamics of the subject area, possibly using topic discovery algorithms. Users are interested in knowing how much people talk about a topic, which words are related to it, if it has a good or bad reputation, etc. Thus, topics are obvious candidates to become a dimension of the cubes for SBI. Like for any other dimension, users are very interested in grouping topics together in different ways to carry out more general and effective analyses —which requires the definition of a topic hierarchy that specifies inter-topic roll-up (i.e., grouping) relationships so as to enable aggregations of topics at different levels. However, topic hierarchies are different from traditional hierarchies (like the temporal and the geographical one) in several ways:

♯1 Also non-leaf topics can be related to facts (e.g., clips may talk of smartphones as well as of the Galaxy III) [1]. This means that grouping topics at a given level may not determine a total partitioning of facts [10]. Besides, topic hierarchies are unbalanced, i.e., hierarchy instances can have different lengths. Note that, in ROLAP (Relational OLAP) contexts, a hierarchy of this type can be represented by coupling a classical dimension table with a *navigation table* that explicitly represents the transitive closure of the node relationships [3].

♯2 Trendy topics are heterogeneous (e.g., they could include names of famous people, products, places, brands, etc.) and change quickly over time (e.g., if at some time it were announced that using smartphones can cause finger pathologies, a brand new set of hot unpredicted topics would emerge during the following days), so a comprehensive schema for topics cannot be anticipated at design time and must be dynamically defined.

♯3 Some topics (e.g., products) are normally also part of the business hierarchies of the enterprise data warehouse. This suggests to model those topics in such a way as to enable users to establish a direct connection with the cubes storing business data (e.g., on product sales).

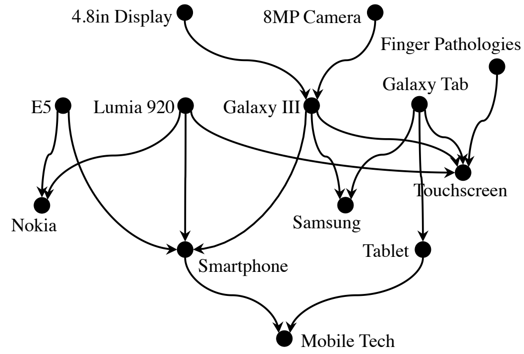

**Figure 1: A topic hierarchy for the mobile technology subject area; arrows represent inter-topic roll-up relationships**

♯4 Roll-up relationships between topics can have different semantics: for instance, the relationship semantics in "Galaxy III has brand Samsung" and "Galaxy III has type smartphone" is quite different. In traditional hierarchies this is indirectly modeled by leaning on the semantics of aggregation levels ("Smartphone" is a member of level Type, "Samsung" is a member of level Brand).

EXAMPLE 1. *In our motivating example, a marketing analyst wants to analyze people's feelings about mobile devices and relate them to the selling trends. A basic cube she will use to this purpose is the one counting, within the textual UGC, the number of occurrences of each topic related to subject area "mobile technologies", distinguishing between those expressing positive/negative sentiment as labeled by an opinion mining algorithm. Figure 1 shows a set of topics for mobile technologies and their roll-up relationships (e.g., when analyzing topic "Samsung", decision makers may wish to also include occurrences of topics "Galaxy III" and "Galaxy Tab"), while Table 1 gives some sample facts (note that the total number of occurrences can be higher than the sum of positive and negative ones, because occurrences may be unbiased). Now, let the decision maker be specifically interested in two types of analysis of the UGS: (i)* brand reputation, *aimed at assessing the people's perception of each brand; (ii)* talking volume, *whose goal is to count the overall occurrences of mobile tech topics; and (iii)* health rumors, *aimed at capturing the customers' concerns about touchscreens and the possible pathologies they may cause. In the first case, the perception of Samsung will be measured by counting the positive and negative occurrences of topics "Samsung", "Galaxy III", and "Galaxy Tab"; in the second case, all occurrences of all topics except "Nokia" and "Samsung" will be counted; in the third case, only the occurrences of "Touchscreen" and "Finger Pathologies" will be considered. The results are shown in Table 2; it appears that, depending on the user's goals, facts can be aggregated in different ways by navigating or not inter-topic relationships with different semantics.*

In light of the above, topic hierarchies in ROLAP contexts must clearly be modeled with more sophisticated solutions than traditional star schemata. Though some attempts have

**Table 1: Sample (fake) facts for topics**

| Topic | positiveOcc | negativeOcc | totalOcc |
|---|---|---|---|
| 4.8in Display | 10 | 2 | 18 |
| 8MP Camera | 0 | 3 | 3 |
| E5 | 30 | 10 | 40 |
| Lumia 920 | 10 | 10 | 20 |
| Galaxy III | 20 | 5 | 25 |
| Galaxy Tab | 22 | 0 | 22 |
| Nokia | 20 | 10 | 35 |
| Samsung | 50 | 10 | 60 |
| Tablet | 5 | 5 | 30 |
| Smartphone | 60 | 20 | 80 |
| Mobile Tech | 10 | 20 | 30 |
| Touchscreen | 60 | 10 | 100 |
| Finger Path. | 0 | 25 | 25 |

**Table 2: Brand reputation, talking volume, and health rumors analyses**

| Topic | positiveOcc | negativeOcc | totalOcc |
|---|---|---|---|
| Nokia | 60 | 30 | |
| Samsung | 92 | 15 | |
| Mobile Tech | | | 268 |
| Touchscreen | | 35 | |

been made in the literature to address some of the mentioned issues (e.g., [17, 1]), no solution to all of them has been found so far. The approach we propose in this paper to deal with topic hierarchies is called *meta-stars*; its basic idea is to use meta-modeling coupled with navigation tables and with traditional dimension tables. On the one hand, navigation tables easily support hierarchy instances with different lengths and with non-leaf facts (requirement ♯1), and allow different roll-up semantics to be explicitly annotated (requirement ♯4); on the other, meta-modeling enables hierarchy heterogeneity and dynamics to be accommodated (requirement ♯2). Finally, dimension tables are easily integrated with standard business hierarchies (requirement ♯3). As discussed in Section 6, an obvious consequence of the adoption of navigation tables is that the total size of the solution increases exponentially with the size of the topic hierarchy. This clearly limits the applicability of the meta-star approach to topic hierarchies of small-medium size; however, we argue that this limitation is not really penalizing because topic hierarchies are normally created and maintained manually by domain experts, which suggests that their size can hardly become too large.

This paper is only focused on topic hierarchies and their effective modeling; the issues related to all the other components of an SBI platform, e.g., how to label a clip with a sentiment and how to discover topics, are out of scope. In the remainder of the paper, after discussing the related literature in Section 2, we sketch an architecture to support SBI in Section 3. Then, in Sections 4 and 5 we present our approach and the types of queries it support. An experimental evaluation is proposed in Section 6, while Section 7 draws the conclusions.

## 2. RELATED WORKS

OLAP techniques are normally applied to multidimensional cubes storing structured business data. Nevertheless, also the problem of storing textual documents in multidimensional form to enable OLAP analyses has been explored in the literature to some extent. For instance, in [2] cubes are exploited to compute multidimensional aggregations on classified documents, using measures such as keyword frequency, document count, document class frequency; the hierarchies used for analyses are based on a given ontology, which limits the approach flexibility. A cube for analyzing term occurrences in documents belonging to a corpus is proposed in [4]; the categorization of terms is obtained from a thesaurus or from a concept hierarchy such as Wordnet. A measureless cube for OLAP analysis of semi-structured documents is presented by [12]; a novel OLAP operation called *focus* is introduced to specify a subject of analysis and aggregate data accordingly. In a related paper [13], a *top keyword* aggregation function is defined to represent a set of documents by their most significant terms using the well-known tf-idf weighing function. Finally, [11] shows how OLAP and information retrieval functionalities can be integrated to access both structured data stored in a data warehouse and unstructured data in form of documents; a global ontology models the business domain and provides the mappings to connect OLAP and information retrieval.

A data warehousing architecture for analyzing large data sets at Facebook, used for instance for friend recommendation, is described in [16]. The paper is mainly focused on flexibility and scalability issues, and no insight on the underlying models is given.

A work sharing some similarities with ours is the one in [14], that presents an architecture to extract tweets from Twitter and load them to a data warehouse. Conceptual models for Twitter streams from both OLTP and OLAP points of view are also proposed. However, both models are focused on the inter-relationships between tweets and between users (the *influencer/followers* mechanism), and little attention is paid to classifying and analyzing tweet topics. An approach for disambiguating and categorizing the entities in the tweets aimed at discovering topics is described in [8]; Wikipedia is used as a knowledge base to this end. The results obtained are used for determining users' topic profiles, and the possibility of analyzing them using OLAP techniques is not considered. The real-time identification of emerging topics in tweets is studied in [7]. Bursty keywords are extracted first, then grouped to identify trends; however, trends are analyzed using a front-end with limited flexibility.

Topic modeling is also the goal of the *Topic Cube* approach [17], that extends traditional cubes to cope with a topic hierarchy and to store probabilistic content measures of text documents learned through a probabilistic topic model. The topic hierarchy is a tree that models parent-child relationships between topics of interest.

In [1] the authors model the topic hiearchy as a DAG of topics where each topic can have several parents. On the one hand, the proposed solution has higher expressivity with respect to traditional hierarchies due to the presence of topic-oriented OLAP operators; on the other hand, it lacks in providing a semantics for the topics in the DAG, that are organized and aggregated only according to their position in the DAG. In other words, with reference to Example 1, the user cannot ask for the average sentiment of each single smartphone since there is no evidence of which instances have type "smartphone".

Apart from the specific social context, advanced modeling of multidimensional hierarchies has been studied by several authors [6, 10]. However, none of the proposed solutions completely match the topic hierarchy requirements.

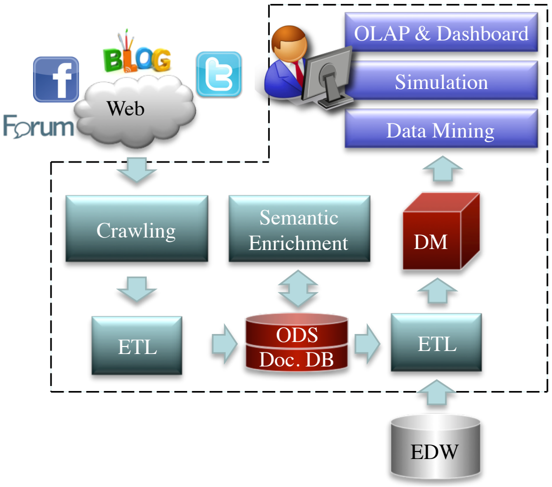

**Figure 2: An architecture for SBI**

To the best of our knowledge, in the commercial world no solution is offered to run OLAP analyses on UGC. The platform whose functionalities are closest to those achieved by our approach is SAS, that exploits its in-memory engine (called SAS In-Memory Analytics Server) to store facts and dimensions in a single, flat in-memory table. The SAS solution supports manual definition of topic hierarchies and their navigation; however, it has inherent limits due to memory availability and does not allow the UGC to be integrated with the information stored in the enterprise data warehouse.

## 3. ARCHITECTURAL OVERVIEW

The architecture we propose to support our approach to SBI is depicted in Figure 2. Its main highlight is the integration between sentiment and business data, which is achieved in a non-invasive way by extracting some business flows from the enterprise data warehouse and integrating them with those carrying textual UGC, in order to provide users with 360° decisional capabilities. In the following we briefly comment on each component.

The *Crawling* component carries out a set of keyword-based queries aimed at retrieving the clips (and the available meta-data) that are in the scope of the subject area. The target of the crawler search could be either the whole web or a set of user-defined web sources (e.g., blogs, forums, web sites, social networks). The semi-structured output of the crawler is turned into a structured form and loaded onto the *Operational Data Store* (ODS), that stores all the relevant data about clips, their authors, and their source channels; to this end, a relational ODS can be coupled with a document-oriented database that can efficiently store and search the text of the clips. The ODS also represents all the topics within the subject area and their relationships. The *Semantic Enrichment* component works on the ODS to extract the semantic information hidden in the clip texts. Depending on the technology adopted (e.g., supervised machine-learning [9] or lexicon-based techniques [15]) such information can include the single sentences in the clip, its topic(s), the syntactic and semantic relationships between words, or the sentiment related to a whole sentence or to each single topic it contains. The *ETL* component periodically extracts data about clips and topics from the ODS, integrates them with the business data extracted from the *Enterprise Data Warehouse* (EDW), and loads them onto the *Data Mart* (DM). The DM stores integrated data in the form of a set of multidimensional cubes to be used for decision making in three complemental ways:

1. *OLAP & Dashboard*: users can explore the UGC from different perspectives and effectively control the overall social feeling. Using OLAP tools for analyzing UGC in a multidimensional fashion pushes the flexibility of our architecture much further than the standard architectures adopted in this context.

2. *Data Mining*: users evaluate the actual relationship between the rumors/opinion circulating on the web and the business events (e.g., to what extent positive opinions circulating about a product will have a positive impact on sales?).

3. *Simulation*: the correlation patterns that connect the UGC with the business events, extracted from past data, are used to forecast business events in the near future given the current UGC.

In our prototypical implementation of this architecture, topics and roll-up relationships are manually defined; we use Brandwatch for keyword-based crawling, Talend for ETL, SyN Semantic Center by SyNTHEMA for semantic enrichment (specifically, for labeling each clip with its sentiment), Oracle for storing the ODS and the DM, and MongoDB for storing the document database. We developed an ad-hoc OLAP & dashboard interface using JavaScript (specifically, the D3, Crossfilter, and Dimensional Charting libraries), while simulation and data mining components are not currently implemented. As already stated, in this work we only focus on the DM component, in particular on how to effectively model topic hierarchies.

## 4. META-STARS

Different multidimensional cubes can be stored in the DM component of Figure 2, focused for instance on the perceived sentiment for the topics in the subject area, on the correlations between topics, on the trending topics, and so on as determined by the semantic enrichment process (Figure 3 shows a simple cube based on Example 1). Typical indicators associated to these cubes are the *topic share* (ratio between the number of occurrences of a topic and the total number of occurrences of all topics in a given time interval), the *topic awareness* (ratio between the number of clips mentioning a topic and the total number of clips), the *market beat* (percentage of positive/negative opinions on a topic), the *average sentiment* (average of biased opinions on a topic). Clearly, topics are first-class citizens for the large majority of relevant analyses that decision-makers find interesting in this field. Thus, expressive and flexible solutions are required to model topics in DM cubes.

It is almost impossible to devise a fixed schema for a subject area at design time and force all newly-discovered topics to fit that schema. However, a large part of topics can be effectively classified into levels, such as Product and Brand in our example, that mostly correspond to aggregation levels in traditional business hierarchies. Like in traditional multidimensional modeling, the relationships between these topics

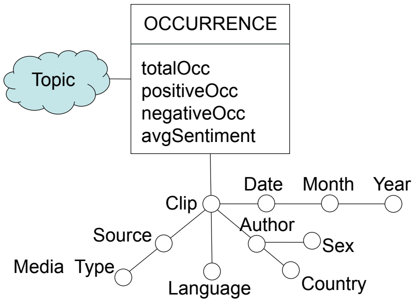

**Figure 3: A conceptual representation of a cube for analyzing textual UGC**

can be captured by a hierarchy schema, to be expressed via roll-up partial orders like shown in Definition 1.

Definition 1. *A* hierarchy schema $\mathcal{S}$ *is a couple of a set* $L$ *of levels and a* roll-up *partial order* $\succ$ *of* $L$. *We will write* $l_k \dot{\succ} l_j$ *to emphasize that* $l_k$ *is an immediate predecessor of* $l_j$ *in* $\succ$.

Example 2. *In our motivating example it is* $L = \{$Product, Type, Category, Brand, Component$\}$ *and* Component$\dot{\succ}$Product $\dot{\succ}$Type$\dot{\succ}$Category, Product$\dot{\succ}$Brand *(see also Figure 4, left-hand side).*

The connection between hierarchy schemata (intension) and topic hierarchies (extension) is captured by Definition 2, that also annotates roll-up relationships with their semantics.

Definition 2. *A* topic hierarchy *conformed to hierarchy schema* $\mathcal{S} = (L, \succ_{\mathcal{S}})$ *is a triple of (i) an acyclic directed graph* $H = (T, R)$, *where* $T$ *is a set of topics and* $R$ *is a set of inter-topic roll-up relationships; (ii) a partial function* $Lev : T \rightarrow L$ *that associates some topics to levels of* $\mathcal{S}$*; and (iii) a partial function* $Sem : R \rightarrow \rho$ *that associates some roll-up relationships to their semantics (with* $\rho$ *being a list of user-defined roll-up semantics). Graph* $H$ *must be such that, for each ordered pair of topics* $(t_1, t_2) \in R$ *such that* $Lev(t_1) = l_1$ *and* $Lev(t_2) = l_2$, *it is* $l_1 \dot{\succ} l_2$ *and* $\forall (t_1, t_3) \in R, Lev(t_3) \neq l_2$.

The intuition behind the constraints on $H$ is that inter-topic relationships must not contradict the roll-up partial order and must have many-to-one multiplicity. For instance, the arc from "Galaxy III" to "Smartphone" is correct because Product$\dot{\succ}$Type, but there could be no other arc from "Galaxy III" to a topic of level Type. In the same way, no arc from a product to a category is allowed; the arc from "Galaxy III" to "Touchscreen" is allowed because the latter does not belong to any level.

Finally, Definition 3 provides a compact representation for the semantics involved in any path of a topic hierarchy.

Definition 3. *Given topic* $t_1$ *such that* $Lev(t_1) = l_1$ *and given level* $l_2$ *such that* $l_1 \succ l_2$, *we denote with* $Anc^{l_2}(t_1)$ *the topic* $t_2$ *such that* $Lev(t_2) = l_2$ *and* $t_2$ *is reached from* $t_1$ *through a directed path* $P$ *in* $H$. *The* roll-up signature *of couple* $(t_1, t_2)$ *is a binary string of* $|\rho|$ *bits, where each bit corresponds to one roll-up semantics and is set to 1 if at least one roll-up relationship with that semantics is part of* $P$, *is set to 0 otherwise. Conventionally, the roll-up signature of* $(t, t)$ *is a string of 0's for each* $t$.

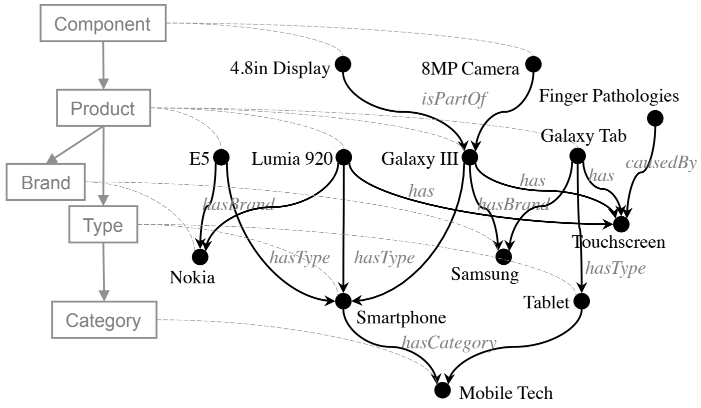

**Figure 4: The annotated topic hierarchy for the mobile technology subject area**

Example 3. *In Figure 4 the topic hierarchy of Figure 1 is reconsidered and annotated with levels and roll-up semantics; for instance, it is* $Anc^{\mathsf{Brand}}$(8MP Camera) = Samsung, $Anc^{\mathsf{Type}}$(8MP Camera) = Smartphone. *Note that topics "Touchscreen" and "Finger Pathologies" do not belong to any level. If* $\rho =$ (isPartOf, hasType, hasBrand, hasCategory, has, causedBy)*, then the roll-up signature of* (8MP Camera, Samsung) *is* 101000 *(because the path from "8MP Camera" to "Samsung" includes roll-up relationships with semantics* isPartOf *and* hasBrand*), that of* (8MP Camera, Smartphone) *is* 110000.

The meta-star approach we propose to model topic hierarchies on ROLAP platforms combines classical dimension tables with recursive navigation tables and extends the result by meta-modeling. Remarkably, the designer can tune the solution by deciding which levels $L^{stat} \subseteq L$ are to be modeled also in a static way, i.e., like in a classical dimension table. Two different tables are used:

1. A *topic table* storing one row for each distinct topic $t \in T$. The schema of this table includes a primary surrogate (i.e., DBMS-generated) key IdT, a Topic column, a Level column, and an additional column for each static level $l \in L^{stat}$. The row associated to topic $t$ has Topic$= t$ and Level$= Lev(t)$. Then, if $Lev(t) \in L^{stat}$, that row has value $t$ in column $Lev(t)$, value $Anc^{l}(t)$ in each column $l$ such that $l \in L^{stat}$ and $Lev(t) \succ l$, and NULL elsewhere.
2. A *roll-up table* storing one row for each topic in $T$ and one for each arc in the transitive closure of $H$. The row corresponding to topic $t$ has two foreign keys, ChildId and FatherId, that reference the topic table and both store the surrogate of topic $t$, and a column RollUpSignature that stores the roll-up signature of $(t, t)$, i.e., a string of 0's. The row corresponding to arc $(t_1, t_2)$ stores in ChildId and FatherId the two surrogates of topics $t_1$ and $t_2$, while column RollUpSignature stores the roll-up signature of $(t_1, t_2)$.

Example 4. *The topic and the roll-up tables for our motivating example when* $L^{stat} = \{$Product, Type, Category$\}$ *are*

TOPIC_T

| IdT | Topic | Level | Product | Type | Category |
|---|---|---|---|---|---|
| 1 | 8MPCamera | Component | – | – | – |
| 2 | GalaxyIII | Product | GalaxyIII | Smartph. | MobTech |
| 3 | GalaxyTab | Product | GalaxyTab | Tablet | MobTech |
| 4 | Smartphone | Type | – | Smartph. | MobTech |
| 5 | Tablet | Type | – | Tablet | MobTech |
| 6 | MobileTech | Category | – | – | MobTech |
| 7 | Samsung | Brand | – | – | – |
| 8 | Finger Path. | – | – | – | – |
| 9 | Touchscreen | – | – | – | – |
| ... | ... | ... | ... | ... | ... |

ROLLUP_T

| ChildId | FatherId | RollUpSignature |
|---|---|---|
| 1 | 1 | 000000 |
| 2 | 2 | 000000 |
| ... | ... | 000000 |
| 1 | 2 | 100000 |
| 2 | 4 | 010000 |
| 2 | 7 | 001000 |
| 4 | 6 | 000100 |
| 8 | 9 | 000001 |
| 2 | 9 | 000010 |
| ... | ... | ... |
| 1 | 4 | 110000 |
| 1 | 7 | 101000 |
| 1 | 9 | 100010 |
| 2 | 6 | 010100 |
| 3 | 6 | 010100 |
| ... | ... | ... |
| 1 | 6 | 110100 |
| ... | ... | ... |

**Figure 5: Meta-star modeling for the mobile technology subject area**

*reported in Figure 5. The eleventh row of the roll-up table states that the roll-up signature of couple* (8MP Camera, Smartphone) *is* 110000*, i.e., that the path from one topic to the other includes semantics* isPartOf *and* hasType.

Choosing which levels are to be modeled in a static way is done at design time, based on a trade-off between efficiency and effectiveness. In particular, as shown in Sections 5 and 6, meta-stars yield higher querying expressiveness, at the cost of a lower time and space efficiency. Meta-stars also better support topic hierarchy dynamics, through the combined use of meta-modeling and of the roll-up table. A whole new set of emerging topics, possibly structured in a hierarchy with different levels, can be accommodated —without changing the schema of meta-stars— by adding new values to the domain of the Level column, adding rows to the topic and the roll-up tables to represent the new topics and their relationships, and extending the roll-up signatures with new bits for the new roll-up semantics. The newly-added levels will immediately become available for querying and aggregation.

# 5. QUERYING META-STARS

A classical OLAP query includes a group-by clause and a selection clause. In this section we show how meta-stars support OLAP queries with increasing expressiveness and complexity, starting from queries using only static levels to end-up with semantics-aware queries. We preliminarily recall that, in this context, facts can also be associated to non-leaf topics. As a consequence, multiple semantics of aggregation are made available to users. For instance, computing the number of occurrences of "Smartphone" may either mean considering only the UGC mentioning the word "Smartphone", or also considering the UGC mentioning products of type smartphones (such as Galaxy III), or also considering the UGC mentioning a component of a product of type smartphone (such as 8MP Camera).

## 5.1 Queries without Topic Aggregation

In this family of queries the topic hierarchy is not navigated, i.e., only occurrences of the very topics of interest are counted. These queries can be always formulated on the topic table by relying on the Level column; for instance, the number of total occurrences for each brand on a given date are obtained as follows:

```
SELECT     TOPIC_T.Topic, SUM(FT.TotalOcc)
FROM       TOPIC_T, DTCLIP, FT
WHERE      FT.IdT = TOPIC_T.IdT AND FT.IdC = DTCLIP.IdC AND
           TOPIC_T.Level = "Brand" AND DTCLIP.Date = "06/22/2013"
GROUP BY   TOPIC_T.Topic;
```

(DTCLIP is a separate dimension table storing clips, see Figure 3).

Clearly, if the required topic level has been modeled as static, like Type, the query can also be equivalently formulated by directly including that level in the group-by clause:

```
SELECT     TOPIC_T.Type, SUM(FT.TotalOcc)
FROM       TOPIC_T, DTCLIP, FT
WHERE      FT.IdT = TOPIC_T.IdT AND FT.IdC = DTCLIP.IdC AND
           TOPIC_T.Level = "Type" AND DTCLIP.Date = "06/22/2013"
GROUP BY   TOPIC_T.Type;
```

## 5.2 Queries with Topic Aggregation

In this family of queries the topic hierarchy is extensively navigated, i.e., each topic of interest is considered together with its descendants when computing the number of occurrences. The portion of topic hierarchy that has been modeled as static is easily navigated using the topic table as if it were a classical dimension table; for instance,

```
SELECT  SUM(FT.TotalOcc)
FROM    TOPIC_T, DTCLIP, FT
WHERE   FT.IdT = TOPIC_T.IdT AND FT.IdC = DTCLIP.IdC AND
        TOPIC_T.Category = "Mobile Tech" AND
        DTCLIP.Date = "06/22/2013";
```

returns the occurrences of "Mobile Tech" counting its types and products (but not its components, because Component $\notin L^{stat}$.

On the other hand, if aggregation is to involve levels that have not been modeled has static, the roll-up table must be used. For instance, this is the case for the talking volume analysis of Example 1, that returns the total number of occurrences for "Mobile Tech" and all its descendants also including components:

```
SELECT  SUM(FT.totalOcc)
FROM    TOPIC_T, ROLLUP_T, DTCLIP, FT
WHERE   FT.IdT = ROLLUP_T.ChildId AND
        ROLLUP_T.FatherId = TOPIC_T.IdT AND
        FT.IdC = DTCLIP.IdC AND
        TOPIC_T.Topic = "Mobile Tech" AND
        DTCLIP.Date = "06/22/2013";
```

In case the desired aggregation includes two or more levels of the topic hierarchy, aliases must be introduced to use different "versions" of the topic and roll-up tables. For instance, the query below computes the average sentiment for each combination of brand and type:

```
SELECT     T1.Topic AS Brand, T2.Topic AS Type, AVG(FT.avgSentiment)
FROM       TOPIC_T T1, ROLLUP_T R1,
           TOPIC_T T2, ROLLUP_T R2, FT
WHERE      FT.IdT = R1.ChildId AND R1.FatherId = T1.IdT AND
           FT.IdT = R2.ChildId AND R2.FatherId = T2.IdT AND
           T1.Level = "Brand" AND T2.Level = "Type"
GROUP BY   T1.Topic, T2.Topic;
```

### 5.3 Queries with Semantics-Aware Topic Aggregation

While the two previous types of queries can also be formulated on a classical star schema extended with a navigation table to model recursion, this type of query uses the user-defined roll-up semantics to filter the way the topic hierarchy is navigated so as to produce custom aggregations. For instance, this is the case with the brand reputation analysis of Example 1, that returns the number of positive and negative occurrences of each brand and of its products:

```
SELECT     TOPIC_T.Topic, SUM(FT.positiveOcc), SUM(FT.negativeOcc)
FROM       TOPIC_T, ROLLUP_T, FT
WHERE      FT.IdT = ROLLUP_T.ChildId AND
           ROLLUP_T.FatherId = TOPIC_T.IdT AND
           TOPIC_T.Level = "Brand" AND
           ROLLUP_T.RollUpSignature = 001000
GROUP BY   TOPIC_T.Topic;
```

Another query of this family is the one for health rumors analysis, that returns the negative occurrences for touchscreens and the related pathologies:

```
SELECT  TOPIC_T.Topic, SUM(FT.negativeOcc)
FROM    TOPIC_T, ROLLUP_T, FT
WHERE   FT.IdT = ROLLUP_T.ChildId AND
        ROLLUP_T.FatherId = TOPIC_T.IdT AND
        TOPIC_T.Topic = "Touchscreen" AND
        ROLLUP_T.RollUpSignature = 000001;
```

## 6. EVALUATION

In this section we evaluate the performance of meta-stars by comparing the efficiency of query execution against star schemata. All tests were conducted using the Oracle 11g RDBMS on a 64-bits AMD Opteron quad-core 2.09GHz virtual machine, with 4GB RAM, running Windows Server 2008 R2 Standard SP1.

To conduct the tests we generated a benchmark of sample cubes with different characteristics but all conformed to the conceptual schema of Figure 3. We created three perfectly height-balanced topic hierarchies with $L^{stat} \equiv L$, in order to create equivalent structures for both the meta-star and the star schema. The parameters used to create the topic hierarchies are the number of levels and the fan-out of each node (i.e., the number of children connected to each father). Table 3 summarizes the characteristics of the topic hierarchies; clearly, the number of topics and the size of the roll-up table increase exponentially with the tree height. In addition, we generated two fact tables, $FT1$ and $FT2$, with 1M and 10M facts respectively, and linked each of them to the previously defined topic tables. For a realistic and fair evaluation, we created B$^{+}$-indexes on all foreign keys, on the Level column, and on all columns corresponding to static levels; no materialized views were created.

To define the workload for evaluation we considered the query family described in Section 5.2 (i.e., the ones based on topic aggregation), that are equally executable on both meta-stars and star schemata and represent the worst case for meta-stars efficiency since they require access to the roll-up table. In particular, we created queries with an increasing number of levels (from 0 to 2) in the group-by clause, in order to evaluate the cost of using one or more roll-up table aliases. The query execution results are shown in Table 4; each execution time displayed is the average time required to run three different queries with the same number of levels in their group-by's and different selection predicates.

**Table 3: Characteristics of meta-stars**

| *Topic hier.* | \|TOPIC_T\| | \|ROLLUP_T\| | *fan-out* | *tree height* |
|---|---|---|---|---|
| $H1$ | 106 | 626 | 4 | 4 |
| $H2$ | 658 | 4,514 | 8 | 4 |
| $H3$ | 27,306 | 334,962 | 4 | 8 |

**Table 4: Execution time of queries (in seconds)**

| *Table* | *\|Group-by\|* | $FT1$ | | $FT2$ | |
|---|---|---|---|---|---|
| | | *Meta-star* | *Star s.* | *Meta-star* | *Star s.* |
| $H1$ | 0 | 13.8 | 12.7 | 140.0 | 137.2 |
| | 1 | 16.0 | 5.8 | 174.6 | 64.3 |
| | 2 | 16.6 | 14.6 | 162.4 | 162.1 |
| $H2$ | 0 | 13.6 | 13.0 | 136.0 | 133.6 |
| | 1 | 16.7 | 5.6 | 179.5 | 179.4 |
| | 2 | 17.0 | 16.2 | 175.8 | 162.2 |
| $H3$ | 0 | 12.2 | 9.0 | 139.1 | 126.6 |
| | 1 | 15.9 | 14.1 | 147.3 | 172.1 |
| | 2 | 35.1 | 16.9 | 187.1 | 144.2 |

Though, as expected, in most cases star schemata outperform meta-stars, the time execution gap is quite limited and perfectly acceptable in terms of *on-line* querying. The gap is significantly smaller, in relative terms, for $FT2$ since the execution time is mostly spent to access the fact table rather than the topic hierarchy. Noticeably, execution times for meta-stars increase smoothly for group-by's with increasing number of levels. The execution time behaves similarly when the cardinality of the topic and roll-up tables increases. In particular, an in-depth analysis of the Oracle execution plans has shown that, although the roll-up table cardinality increases exponentially with the depth of the topic hierarchy (see Table 3), the execution time increases smoothly because indexes allow only the relevant part of that table to be accessed when querying.

In this paper we have chosen to test our approach using the original Oracle plans (no "hints") to get more realistic results. On the other hand, our experiments pointed out that the Oracle Optimizer may choose heterogeneous execution plans even for very similar queries. Although in principle this behavior could be due to slight changes in the estimated costs, it also raises the doubt that Oracle fails in using the best plan thus determining some peaks in the query costs. In the light of this, we argue that all the execution times presented in this section, both for meta-stars and star schemata, could presumably be further improved by fine tuning and forcing smart execution plans.

## 7. FINAL REMARKS

In this paper we have introduced SBI as a relevant area for business and research, and we have proposed an expressive solution to model topic hierarchies based on same specific requirements: heterogeneity and dynamics of topic classifications, integrability with business hierarchies, and semantics-aware aggregation. Noticeably, the choice of the subset of levels to be modeled as static rules the trade-off between the dynamics of topic classification and aggregation and the efficiency of integrating UGC-related facts (accessed via topic hierarchies) with business-related facts (accessed via standard hierarchies).

Remarkably, though in this work we made some limiting assumptions for simplicity, the potentiality of meta-stars goes well beyond. As a first remark, navigation tables also allow for modeling many-to-many relationships (e.g., the 8MP camera could be a component of both the Galaxy III and the Galaxy Tab); of course, as discussed in [10, 6], this requires the summarizability problem to be addressed. Besides, while in this paper we modeled static portions of the topic hierarchies in a redundant fashion (i.e., by modeling inter-topic relationships both in a denormalized form and recursively within navigation tables), to improve performances it will be possible under some circumstances to exclude these portions from navigation tables so as to reduce their size, while preserving full querying expressiveness.

To improve the meta-star approach we are currently working on the following issues:

1. *Historicization*: interesting topics change over time and the system should be capable of considering only those that are valid within a given time range.

2. *Cost model for meta-stars*: defining a cost model will allow the size and the querying efficiency of a topic hierarchy to be evaluated a priori. This will guide the designer in deciding which levels should be static.

3. *Topic hierarchy generation*: the dynamics of topics requires that their values and relationships are continuously maintained. Though the basic topics can be automatically derived from the enterprise business hierarchies, in general they will be manually inserted, possibly by the users. Since feeding the topic and the roll-up tables appears to be a cumbersome task, we are working towards modeling the topic hierarchy through an ontology that can be automatically turned into a meta-star.

4. *Coupling SQL and OWL*: in the same direction, we are also considering the possibility of using the OWL language to directly query the topic hierarchy. This can avoid the storing of the roll-up table that, as already said, could become very large and represents the main limitation when adopting the meta-star approach on large topic hierarchies.

5. *Summarizability for many-to-many relationships*: though many-to-many relationships between topics can be easily handled by meta-stars, it is not clear yet which summarization rationales are valid and can be adopted to produce interesting results for business users.

6. *OLAP front-end*: meta-stars are not supported by traditional OLAP front-ends, so their use requires ad-hoc reporting queries to be written. To solve this issue we will investigate what meta-data are needed and how OLAP front-ends can be extended to effectively support meta-stars.

## 8. REFERENCES

[1] U. Dayal, C. Gupta, M. Castellanos, S. Wang, and M. García-Solaco. Of cubes, DAGs and hierarchical correlations: A novel conceptual model for analyzing social media data. In *Proc. ER*, pages 30–49, Florence, Italy, 2012.

[2] C. Garcia-Alvarado and C. Ordonez. Query processing on cubes mapped from ontologies to dimension hierarchies. In *Proc. DOLAP*, pages 57–64, 2012.

[3] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, 2008.

[4] J. Lee, D. A. Grossman, O. Frieder, and M. C. McCabe. Integrating structured data and text: A multi-dimensional approach. In *Proc. ITCC*, pages 264–271, Las Vegas, USA, 2000.

[5] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer, 2012.

[6] E. Malinowski and E. Zimányi. Hierarchies in a multidimensional model: From conceptual modeling to logical representation. *Data Knowl. Eng.*, 59(2):348–377, 2006.

[7] M. Mathioudakis and N. Koudas. TwitterMonitor: trend detection over the Twitter stream. In *Proc. SIGMOD Conference*, pages 1155–1158, 2010.

[8] M. Michelson and S. A. Macskassy. Discovering users' topics of interest on Twitter: a first look. In *Proc. AND*, pages 73–80, 2010.

[9] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proc. ACL Conf. on Empirical Methods in Natural Language Processing*, volume 10, pages 79–86, Stroudsburg, USA, 2002.

[10] T. B. Pedersen, C. S. Jensen, and C. E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Inf. Syst.*, 26(5):383–423, 2001.

[11] T. Priebe and G. Pernul. Ontology-based integration of OLAP and information retrieval. In *Proc. DEXA Workshops*, pages 610–614, Prague, Czech Republic, 2003.

[12] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh. A conceptual model for multidimensional analysis of documents. In *Proc. ER*, pages 550–565, Auckland, New Zealand, 2007.

[13] F. Ravat, O. Teste, R. Tournier, and G. Zurfluh. Top keyword: An aggregation function for textual document OLAP. In *Proc. DaWaK*, pages 55–64, Turin, Italy, 2008.

[14] N. U. Rehman, S. Mansmann, A. Weiler, and M. H. Scholl. Building a data warehouse for Twitter stream exploration. In *Proc. ASONAM*, pages 1341–1348, 2012.

[15] M. Taboada, J. Brooke, M. Tofiloski, K. D. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307, 2011.

[16] A. Thusoo, Z. Shao, S. Anthony, D. Borthakur, N. Jain, J. S. Sarma, R. Murthy, and H. Liu. Data warehousing and analytics infrastructure at Facebook. In *Proc. SIGMOD Conference*, pages 1013–1020, 2010.

[17] D. Zhang, C. Zhai, and J. Han. Topic Cube: Topic modeling for OLAP on multidimensional text databases. In *Proc. SDM*, pages 1123–1134, 2009.