Meta-Stars: Dynamic, Schemaless, and Semantically-Rich Topic Hierarchies in Social BI

Enrico Gallinucci
DISI – Univ. of Bologna, Italy
enrico.gallinucci2@unibo.it

Matteo Golfarelli
DISI – Univ. of Bologna, Italy
matteo.golfarelli@unibo.it

Stefano Rizzi
DISI – Univ. of Bologna, Italy
stefano.rizzi@unibo.it

ABSTRACT

A key role in OLAP analyses of textual user-generated content for social business intelligence (SBI) is played by topics, i.e., concepts of interest within a subject area. Topic hierarchies are irregular, heterogeneous, dynamic, and possibly schemaless; besides, unlike in traditional OLAP, different semantics for topic aggregation can be envisioned. In this demonstration we present an architecture for SBI based on meta-stars, a novel approach to topic modeling in ROLAP systems. By coupling meta-modeling with navigation tables, meta-stars can cope with changes in the schema of irregular hierarchies and with schemaless ones; besides, they enable a new class of OLAP queries based on semantically-aware aggregation. The demonstration will focus both on the hierarchy update process and on the querying expressiveness.

1. INTRODUCTION

In the last few years, the success of social networks has led to the accumulation of a huge wealth of user-generated contents (UGCs) about people's tastes, thoughts, and actions —especially, those coming in the form of textual *clips*. This phenomenon is raising an increasing interest from decision makers because it can give them a fresh and timely perception of the market mood [1]. Unfortunately, though some commercial tools are available for analyzing textual clips using a few ad-hoc indicators, they do not support flexible and fully interactive analyses; besides, these tools are essentially built as self-standing applications and are not seen as a permanent part of the company information system.

To bridge this gap, *social business intelligence* (SBI) has emerged as 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 [2]. The goal of SBI is to enable powerful and flexible analyses for users with a limited expertise in databases and ICT; this is typically achieved by storing information into a data warehouse, in the form of multidimensional cubes to be accessed through OLAP techniques.

A key role in the analysis of textual clips is played by *topics*, meant as specific concepts of interest within the subject area [3]. Users are interested in knowing how many 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 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 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. First of all, trendy topics are heterogeneous (e.g., they could include names of people, places, etc.) and change quickly over time (e.g., if at some time a group of politicians were discovered to be corrupt, a new Scandal class of topics would emerge during the following days), so a comprehensive schema for topics cannot be anticipated at design time and must be dynamically defined. For some topics a classification could even be hard due to their fuzzy nature, or unnecessary due to their transitoriness. Even when a schema is present, the expressiveness it requires is often beyond the one of the standard multidimensional model, i.e., topic hierarchies are non-onto, non-covering, and non-strict[1].

While these structural irregularities are already managed in some research models (e.g., [5]), handling hierarchies with dynamic schemata —or even potentially schemaless hierarchies— as required by topic hierarchies still constitutes a big challenge for SBI. Ontologies come to the rescue here, because the wide expressiveness they support enables an effective modeling of topic hierarchies with all their peculiarities; however, the problem of how to move this on a relational platform remains open.

To bridge the gap, in this demonstration we present a complete architecture for SBI based on *meta-stars* [2], a novel approach to topic modeling in relational OLAP systems. By coupling meta-modeled dimension tables with navigation tables, meta-stars can effectively cope with the peculiar requirements of topics hierarchies: on the one hand, meta-modeling enables hierarchy heterogeneity, schema dynamics, and schemaless hierarchies to be accommodated; on the other, navigation tables easily support non-onto, non-

[1]In a *non-onto*, *non-covering*, and *non-strict* hierarchy, instances can have different lengths, non-leaf topics can be related to facts, some hierarchy levels may be missing, and many-to-many relationship between topics may exist [5].

10.5441/002/edbt.2015.50

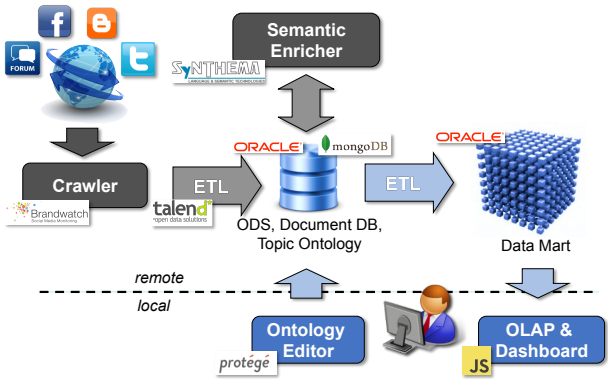

Figure 1: An architecture for SBI

covering, and non-strict hierarchies and also allow different roll-up semantics to be explicitly annotated, which in turn enables a brand new class of OLAP queries based on semantically-aware aggregation.

As already mentioned, topic hierarchies need be continuously updated and refined in both their schemata and instances to keep pace with the quickly-changing social environment. To enable simple and fast editing of hierarchies, we let users manage them in the form of ontologies. In this way, we can take advantage of existing ontology editors, which give good support to the design of irregular hierarchies. Besides, by creating a procedure that automatically loads/updates a meta-star starting from a given ontology, we relieve the user of the task of directly managing meta-stars and enable faster iterations of ontology design and testing.

2. SYSTEM OVERVIEW

The architecture used for this demonstration is depicted in Figure 1 and briefly commented in the following. The components in blue are those actively involved in the demo; the components running locally (on any Internet-enabled PC) and remotely are separated by a dashed line.

The user builds and refines the topic hierarchy by means of an *ontology editor*, then she launches an ETL process to automatically feed the meta-star within the social data mart (see Sections 2.1 and 2.2). The *crawler* component periodically runs a set of keyword-based queries over the web aimed at retrieving the clips (and the related meta-data) that are in the scope of the subject area. The textual content of the clips is then loaded into a document database, while the meta-data are loaded into an *operational data store* (ODS). The *semantic enricher* works on the document database to extract the semantic information hidden in the clip text and writes it in the ODS. A hybrid approach between supervised machine-learning [4] and lexicon-based techniques [6]) is adopted to extract the topics occurring within the single sentences of each clip, understand the syntactic relationships between words, and evaluate the sentiment related to each sentence and topic occurrence. An ETL process periodically extracts data about clips and topic occurrences and co-occurrences from the ODS and loads them into multidimensional cubes within the social data mart. Finally, the user uses OLAP tools and dashboards for flexibly analyzing clips and topics (Section 2.3). The total size of data involved in the demo (ODS + data mart) is about 1TB.

In our prototypical implementation of this architecture we use Brandwatch for keyword-based crawling, Talend for ETL, SyN Semantic Center by SyNTHEMA for semantic enrichment, Protégé for ontology editing, Oracle for storing the ODS, the topic ontology, and the social data mart, and MongoDB as the document database. For OLAP and dashboarding we developed an ad-hoc interface using JavaScript.

Of course, depending on the specific project context, lighter architectures could be sufficient (for instance, semantic enrichment may not be done if users are only interested in analyzing raw data). On the other hand, the architecture in Figure 1 can easily handle the data volumes normally involved in analyses, that in practice are often limited by either the diffusion of the subject area on the web or by the cost for buying clips from third parties.

2.1 Meta-Stars

Topics are first-class citizens for the large majority of relevant analyses that decision-makers find interesting in the field of SBI; thus, expressive and flexible solutions are required to model topics in multidimensional cubes. Meta-stars, introduced in [2], extend star schemata by enabling schemaless hierarchies to seamlessly coexist with hierarchies characterized by an irregular and dynamic schema, while supporting OLAP analyses. The basic idea is that 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, that mostly correspond to aggregation levels in traditional business hierarchies.

A *topic hierarchy* is 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. A topic $t \in T$ can optionally be classified into a *level* $Lev(t)$, and a roll-up relationship $(t_1, t_2) \in R$ can be associated to a semantics $Sem((t_1, t_2)) \in \rho$ (with ρ being a list of user-defined roll-up semantics). The *meta-star* for a topic hierarchy H includes two tables:

1. A *topic table* storing one tuple for each topic in T. The schema of this table includes a primary surrogate key IdT, a Topic column storing the topic name, and a Level column storing the level, if any, in which the topic is classified.

2. A *roll-up table* storing one tuple for each arc in H^+. The tuple for arc (t_1, t_2) has two foreign keys, ChildId and FatherId, storing the surrogates of t_1 and t_2 respectively, and a column RollUpSignature that stores the *roll-up signature* of (t_1, t_2), i.e., a binary string of ρ 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 any directed path from t_1 to t_2, is set to 0 otherwise.

Remarkably, meta-stars defined as above directly support non-onto, non-covering, and non-strict hierarchies (because they pose no constraints on inter-level relationships), allow different roll-up semantics to be explicitly annotated (by storing roll-up signatures), and enable hierarchy heterogeneity and dynamics to be accommodated (by meta-modeling levels in the topic table).

Figure 2 shows an excerpt of the topic hierarchy we will use for the demonstration; the subject area is that of European political elections. Levels are represented by grey

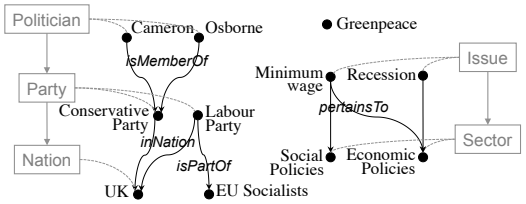

Figure 2: An excerpt of the topic hierarchy for European elections

TOPIC_T

IdT	Topic	Level
1	Cameron	Politician
2	Conservative Party	Party
3	EU Socialists	—
4	Minimum wage	Issue
5	Social Policies	Sector
6	Economic Policies	Sector
7	Greenpeace	—
8	UK	Nation
...	...	...

ROLLUP_T

ChildId	FatherId	RollUpSignature
1	1	0000
2	2	0000
...	...	0000
1	2	1000
1	8	1001
2	8	0001
4	5	0010
4	6	0010
...	...	...

Figure 3: Meta-star for European elections

boxes; topic Greenpeace is unclassified. In this example, the hierarchy is non-onto (also non-leaf topics such as UK can occur in clip sentences) and non-strict (the relationship between issues and sectors is many-to-many). Figure 3 shows a portion of the corresponding topic and roll-up tables where, for instance, the (transitive) relationship between Cameron and UK is expressed by the fourth tuple of the roll-up table, with roll-up signature 1001 (the list of roll-up semantics being $\rho =$ (isMemberOf, isPartOf, pertains, inNation)).

2.2 Feeding Meta-Stars

Clearly, managing topic hierarchies by directly editing topic and roll-up tables would be impractical. For this reason, in our approach topic hierarchies are modeled by users in the form of ontologies. Classes and instances are used to represent levels and topics, respectively, while properties are used to define roll-up relationships between topics. We provided a minimal framework for designing topic hierarchies by defining this set of superclasses and superproperties:

```
<Topic> <rdf:type> <owl:Class> .
<rollsUpTo> <rdfs:range> <Topic> .
<rollsUpTo> <rdf:type> <owl:ObjectProperty> .
<rollsUpTo> <rdfs:domain> <Topic>
```

In this framework, a level is defined as a class that specializes class *Topic* and a topic is defined as an instance of a level (unclassified topics are defined as instances of *Topic*). A roll-up relationship is first defined as a specialization of the *rollsUpTo* superproperty, and its domain and range are properly set considering the levels of its two end topics. Then, the roll-up relationship is implemented as an instance by linking the topics it involves.

The process of automatically generating a meta-star from an ontology takes advantage of the *Spatial and Graph* component in the Oracle DBMS, which allows to store and handle ontologies, as well as to integrate traditional SQL queries with SPARQL queries. Firstly, the ontology is exported from Protégé and loaded into the Oracle database. Then, a stored procedure is launched to read the ontology, determine the hierarchy schema (the levels and their relationships), and generate and execute the DML and the DDL SQL code that updates the data mart.

2.3 Querying Meta-Stars

While the aggregation semantics for OLAP queries is commonly understood and shared, in presence of irregular hierarchies —such as the topic one— some further possibilities arise. In particular, since facts (topic occurrences in clip sentences) can also be associated to non-leaf topics, multiple semantics of aggregation can be made available to users. To deal with these alternative semantics we extend the definition of OLAP query as follows.

Given topic hierarchy $H = (T, R)$, a *schema-aware topic query* is a triple of (i) a *group-by component*, that is a topic level l; (ii) an optional *selection*, that takes the form of a conjunction of Boolean predicates on topic levels; and (iii) a *semantic filter* σ consisting of a subset of allowed roll-up semantics, coded as a binary string of bits. The interpretation of a schema-aware topic query is that of building, for each topic t_i that has level l and satisfies the selection, a group of topics including all topics t such that the roll-up signature of (t, t_i) matches σ. Then, the facts for all topics included in each group are aggregated. Note that, while the group-by component and the selection determine which groups will be built, the semantic filter determines the composition of each group. Queries with $\sigma = 00\ldots0$ are called *queries without topic aggregation*, because the group for topic t_i only includes t_i; queries with $\sigma = 11\ldots1$ are called *queries with full topic aggregation* because all topics t from which t_i can be reached in H are included in the group for t_i. In all the other cases, we will talk of *queries with semantic topic aggregation* as topics are selectively aggregated based on the semantics of the roll-up relationships they are involved in.

Not all topics in T belong to a level, so there is a need for a further class of queries that work independently of the hierarchy schema. In a *schema-free topic query*, the topics of interest are explicitly listed in the group-by component, that takes the form of a set of topics $T' \subseteq T$. A group is built for each topic $t_i \in T'$, the composition of groups is determined like for schema-aware queries, so the same distinction based on topic aggregation can be made.

An example of a schema-aware query is the one asking for the number of occurrences of each Party topic for which Nation=UK, which can be done either without topic aggregation (only clips for the Conservative and Labour Parties are considered) or with topic aggregation (also clips for Cameron and Osborne are counted). An example of a schema-free query with semantic topic aggregation is the one asking for the number of occurrences of topic EU Socialists also considering the UGC mentioning parties of that group such as Labour Party but not the UGC mentioning politicians of those parties (filter on roll-up signature isPartOf).

3. DEMO SCENARIOS

As a case study to demonstrate the power of meta-stars we consider the incoming European elections. In particular, we focused the crawler on social networks, newspapers' websites, and politician's personal blogs from Italian, English, and German sources. In this context, three different scenarios will be demonstrated, yielding an overall demo duration of about 20 minutes.

The first scenario aims at showing how meta-stars can handle irregular and schemaless hierarchies. Non-strictness is shown by creating many-to-many relationships between issues and sectors, like in Figure 2 where Minimum wage per-

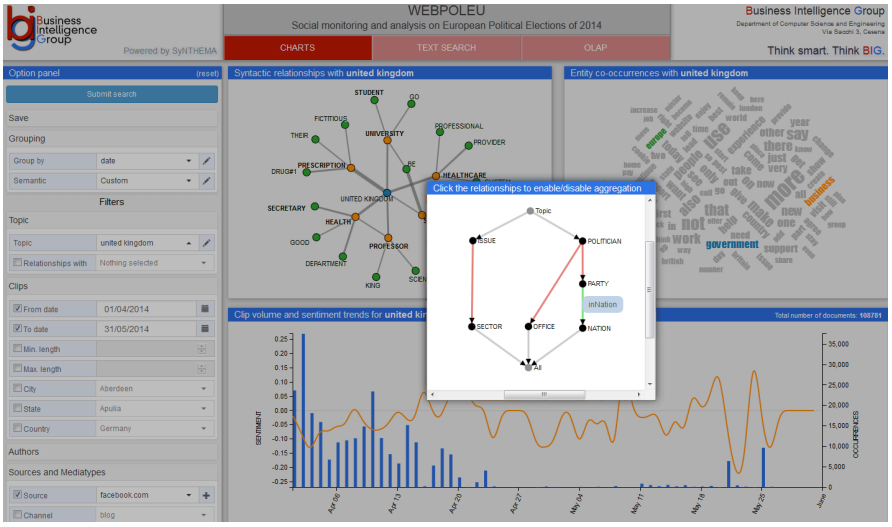

Figure 4: An excerpt of the topic hierarchy for European elections

tains to both Social Policies and Economic Policies. Meta-stars cope with non-strict hierarchies by simply having multiple tuples with the same ChildId in the roll-up table. To deal with non-coverage, a politician who supports the EPP without being member of any national party will be created; since the roll-up table contains every transitive relationship between topics, the problem of missing levels is simply overcame by directly coupling a child to its grandparent. Non-onto hierarchies are transparently accommodated because each topic (even non-leaf ones such as CDU) is represented in the topic table, so it can be directly referred from the fact table. Finally, a schemaless topic hierarchy is created by adding topics (such as Greenpeace in Figure 2) that do not belong to any level, i.e., have attribute Level set to null in the topic table. Note that also unclassified topics can be involved in roll-up relationships with other (classified or not) topics, and that these relationships can be transparently used for topic aggregation.

The second scenario is related to hierarchy dynamics. A recurrent situation in SBI is the discovery of new topics of interest and new topic levels, which requires to start a design iteration that refines the topic hierarchy and updates the meta-star accordingly. During the demo we will use the ontology editor to add new levels and topics, then launch the meta-star feeding procedure to let the new data be immediately available for querying. For instance, with reference to Figure 2, assume that the ontology initially does not include level Sector. Adding Sector and topics Social Policies and Economic Policies leads to update the meta-star as follows: (i) two new tuples are added to the topic table, with attribute Level set to Sector; (ii) the roll-up signature of each existing tuple in the roll-up table is extended with one bit to model the new pertainsTo semantic; and (iii) three tuples are added to the roll-up table to model the roll-up relationships between issues and sector. Note that, while the topic hierarchy has been modified intensionally, i.e., in its schema, the impact of this change on the meta-star level is purely extensional, i.e., it only involves the instances of the tables and not their schemata.

From the analyst point of view, meta-stars significantly increase the expressiveness of OLAP queries. The key element to this end is the roll-up signature, that allows topics to be aggregated by filtering the relationships the user wants to involve. So, the goal of the third scenario is to evaluate meta-stars from the point of view of querying effectiveness and efficiency. For instance, Figure 4 shows an analysis dashboard featuring the results of three different queries. In particular, the lower panel shows the volume and average sentiment for the daily occurrences of topic UK in Facebook clips written in April-May 2014. In the foreground window, the analyst is selecting a semantic filter on inNation to also include the clips mentioning the parties of UK; the SQL code generated for the final query is

```
SELECT    DT_DATE.date, AVG(FT.avgSentiment), COUNT(FT.occurrences)
FROM      TOPIC_T AS T, ROLLUP_T AS R, DT_DATE, DT_CLIP, FT
WHERE     FT.IdT = R.ChildId AND R.FatherId = T.IdT AND T.Topic = 'UK'
AND       BITAND(R.RollUpSignature,0001) = R.RollUpSignature
AND       <star join and selection predicates>
GROUP BY  DT_DATE.Date;
```

4. REFERENCES

[1] M. Castellanos and others. LCI: a social channel analysis platform for live customer intelligence. In *Proc. SIGMOD*, pages 1049–1058, Athens, Greece, 2011.

[2] E. Gallinucci, M. Golfarelli, and S. Rizzi. Meta-stars: multidimensional modeling for social business intelligence. In *Proc. DOLAP*, pages 11–18, S. Francisco, CA, 2013.

[3] L. García-Moya, S. Kudama, M. J. Aramburu, and R. B. Llavori. Storing and analysing voice of the market data in the corporate data warehouse. *Information Systems Frontiers*, 15(3):331–349, 2013.

[4] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *CoRR*, cs.CL/0205070, 2002.

[5] 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.

[6] 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.