A Methodology for Social BI

Matteo Francia
CIRI-ICT – Univ. of Bologna
Bologna, Italy
matteo.francia3@unibo.it

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

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

ABSTRACT

Social BI (SBI) is the emerging discipline that aims at combining corporate data with textual user-generated content (UGC) to let decision-makers analyze their business based on the trends perceived from the environment. Despite the increasing diffusion of SBI applications, no specific and organic design methodology is available yet. In this paper we propose an iterative methodology for designing and maintaining SBI applications that reorganizes the activities and tasks normally carried out by practitioners. Effective support to quick maintenance iterations is a key feature in this context due to the huge dynamism of the UGC and to the pressing need of immediately perceiving and timely reacting to changes in the environment. The paper is completed by two case studies of real SBI projects, related to Italian politics and to the consumer goods area respectively, aimed at proving that the adoption of a structured methodology positively impacts on the project success.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision Support*; D.2.10 [**Software Engineering**]: Design—*methodologies*

Keywords

Business Intelligence, Sentiment Analysis, Design methodologies, User-Generated Content

1. INTRODUCTION AND MOTIVATION

Social networks and portable devices enabled simplified and ubiquitous forms of communication which significantly contributed, during the last decade, to a boost in the voluntary sharing of personal information. Most of us can connect to the Internet anywhere, anytime, and continuously send messages to a virtual community centered around blogs, forums, social networks, and the like. As a result, an enormous amount of *user-generated content* (UGC) related to people's tastes, thoughts, and actions has been made available in the form of preferences, opinions, geolocation, etc. This huge wealth of information is raising an increasing interest from decision makers because it can give them a timely perception of the market mood and help them explain the phenomena of business and society.

Social Business Intelligence (SBI) is the emerging discipline that aims at 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 [6]. As in traditional business intelligence, the goal of SBI is to enable powerful and flexible analyses for decision makers (simply called *users* from now on) with a limited expertise in databases and ICT. In the context of SBI, the most widely used category of UGC is the one coming in the form of textual *clips*. Clips can either be messages posted on social media (such as Twitter, Facebook, blogs, and forums) or articles taken from on-line newspapers and magazines. Digging information useful for users 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 may simply identify the structured parts of a clip, such as its author, or even use *sentiment analysis* techniques [13] to interpret each sentence and if possible assign a *sentiment* (also called *polarity*, i.e., positive, negative, or neutral) to it. We will call *SBI process* the one whose phases range from web crawling to users' analyses of the results.

SBI has emerged as an application and research field in the last few years. Though a wide literature is available about the different phases of the SBI process, no methodology is available yet to organize the different design activities. Indeed, in real SBI projects, practitioners typically carry out a wide set of task but they lack an organic and structured view of the design process. In particular, a distinctive and nonnegotiable feature of these projects is that they call for an effective and efficient support to maintenance iterations, because of the huge dynamism of the UGC and of the pressing need of immediately perceiving and timely reacting to changes in the environment. In the direction of achieving the required responsiveness, in this paper we propose an iterative methodology that reorganizes the activities and tasks for developing and maintaining SBI processes. To evaluate the impact of an engineered approach on the project success (in terms of both correctness and productivity) we present

IDEAS '14, July 07 - 09 2014, Porto, Portugal
Copyright © 2014 ACM 978-1-4503-2627-8/14/07 ...$15.00.
http://dx.doi.org/10.1145/2628194.2628250

and discuss two case studies of real SBI projects, related to Italian politics and to the consumer goods area respectively. While the first one was fully supported by our methodology, the second one was mainly guided by the previous experience of the design team.

The paper is structured as follows. After discussing the related literature in Section 2, in Section 3 we describe an architecture for SBI and in Section 4 we introduce our methodology and its activities. Then, in Section 5 we discuss two case studies, while in Section 6 we draw the conclusions.

2. RELATED WORK

SBI is at the crossroads between several research areas that differently contribute to make the resulting analyses effective and helpful to users. As shown in Figure 1, the SBI process requires first of all to capture and store large set of unstructured or semi-structured data available on the web, social networks, and other textual repositories. Web crawling is a central issue in information retrieval, in whose context powerful languages to automatically and precisely capture the relevant data to be extracted were studied [4, 20, 2, 5]. Storing the crawled data for post-analysis obviously poses a *big data* problem due to the cardinality of the clips and to the heterogeneity of the related metadata [27].

Semantic enrichment of raw clips and text understanding have been studied in several areas of computer science. 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 (NLP) or text analysis techniques to interpret each sentence and if possible assign a sentiment to it (i.e., *sentiment analysis* or *opinion mining* [13]). While NLP approaches try to obtain a full text understanding [29], text mining approaches rely on different techniques (e.g., n-grams) either to find in the text interesting patterns (e.g., named entities [21], relationships between topics [23], or clip sentiment [19]) or to classify/cluster them [28]. The effectiveness of the different approaches largely depends on the quality of the raw text to be analyzed; in general, NLP is effective on syntactically-correct texts (such as on-line newspapers and blogs) while it falls short on ill-formed sentences or when Internet dialects are used (e.g., on social networks). Also hybrid approaches between classical NLP and statistical techniques have been tried, either mainly user-guided, like in [12], or automated and unsupervised, like in [7].

In the area of BI, most efforts for the social content field have been focused in identifying data representations that enable powerful and flexible analyses of data. For example, the *topic cube* approach [30] extends traditional cubes to cope with a topic hierarchy and to store probabilistic content measures of text documents learned through a probabilistic topic model. In [3], the authors model the topic hierarchy as a directed acyclic graph of topics where each topic can have several parents. In [6] we proposed *meta-stars*, whose basic idea is to use meta-modeling coupled with navigation tables and with traditional dimension tables to cope with the dynamism of topic hierarchies; to the best of our knowledge, this is currently the only proposal that enables full OLAP analyses on social data. Finally, in [7] a multidimensional data model is proposed to integrate sentiment data extracted from opinion posts in a corporate data warehouse.

As to the methods for designing classical BI applications, the available literature mainly focuses on traditional, linear approaches such as waterfall with specific reference to data warehouse design. A waterfall approach was first proposed in [8]; a distinguishing feature was the inclusion of a conceptual design phase aimed at better formalizing the data schema. Later on, the same authors proposed Four-Wheel-Drive [9], an agile methodology that specializes recent findings in software engineering to the peculiarities of BI projects. Similarly, the work in [11] breaks with strictly sequential approaches by applying two agile development techniques, namely *scrum* and *eXtreme Programming*. A different approach to tackle data warehouse design complexity is the MDA methodology proposed in [14] to better separate the system functionality from its implementation; in practice, strictly applying this methodology may be hard due to the poor aptitude of users to reading formal models and investing resources in low-values activities. A pragmatic comparison between data warehouse design methodologies is offered in [25], where 15 different solutions proposed by BI vendors are examined. The authors emphasize the lack of software-independent approaches, and point out that all the proposed solutions hardly can deal with changes and market evolution, which creates a robustness problem. This is the first reason why methods for designing classical BI applications cannot be directly applied to the SBI domain. One further reason is that in a BI project most of the attention is dedicated to static (multidimensional) modeling, that largely determines the overall effectiveness, while in SBI a satisfying level of effectiveness can only be achieved only through a coordinated design of crawling and semantic enrichment.

3. AN ARCHITECTURE FOR THE SBI PROCESS

In [6] we proposed an architecture for the SBI process where the information resulting from clip analysis is stored into a data mart in the form of multidimensional cubes to be accessed through OLAP techniques. This allows for overcoming the limitations of traditional approaches to the analysis of textual UGC, where only static or poorly flexible reports are provided and historical data are not made available. The core of our architecture is shown in Figure 1, and it features:

- An *ODS (Operational Data Store)* that stores all the relevant data about clips, their topics, their authors, and their source channels; to this end, a relational database is coupled with a document-oriented database that can efficiently store and search the text of the clips and with a triple store to represent the topic ontology.
- A *data mart* that stores clip and topic information in the form of a set of multidimensional cubes to be used for decision making.
- A *crawling* component that runs a set of keyword-based queries to retrieve the clips (and the related meta-data) that lie within the subject area.
- An *ETL (Extraction, Transformation, and Loading)* component that turns the semi-structured output of the crawler into a structured form and loads it into the ODS, and then periodically extracts data about clips and topics from the ODS to load them into the data mart.

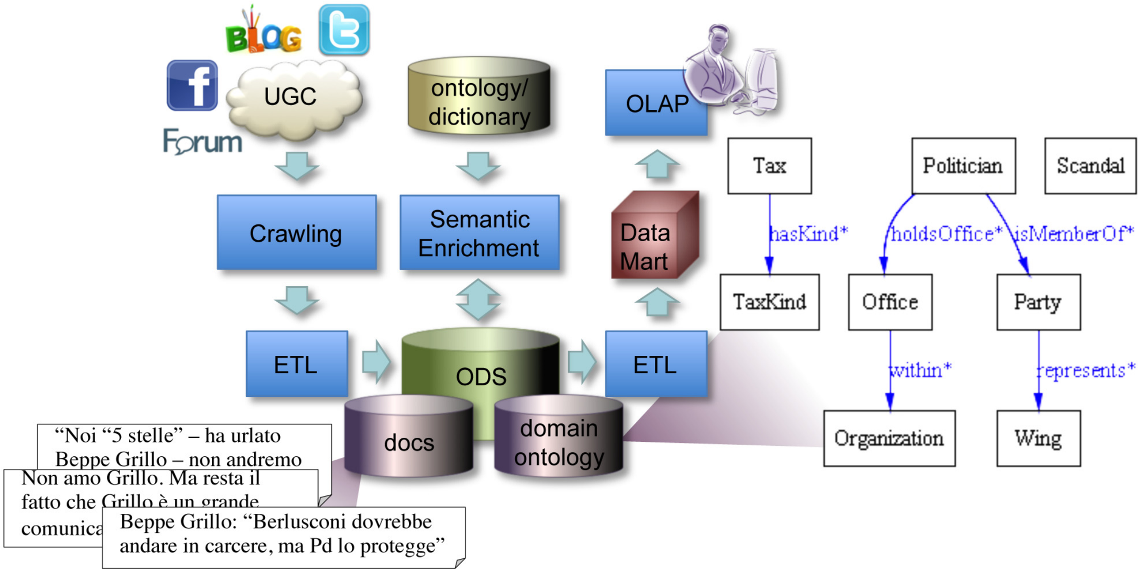

Figure 1: An architecture for SBI (three sample clips in Italian on the left, an excerpt from a domain ontology on the right)

- A *semantic enrichment* component that works on the ODS to extract the semantic information hidden in the clips, such as the topic(s) related to the clip, the syntactic and semantic relationships between words, and the sentiment related to a whole sentence or to each single topic it contains.
- An *OLAP* front-end to enable interactive and flexible analysis sessions of the multidimensional cubes.

In the implementation adopted for both case studies discussed in Section 5 we used Brandwatch (a well-known media monitoring commercial tool, www.brandwatch.com) for keyword-based crawling, Talend (www.talend.com) for ETL, SyN Semantic Center (www.synthema.it) for semantic enrichment, Oracle for storing the ODS, the domain ontology, and the data mart, and MongoDB (www.mongodb.org) as the document database; for OLAP analyses we developed an ad-hoc interface using JavaScript.

The components mentioned above are normally present, though with different levels of sophistication, in most current commercial solutions for SBI. However, as we will show in Table 1, the roles in charge of designing, tuning, and maintaining each component may vary from project to project. In regards to this, SBI projects can be classified as follows:

- *Level 1: Best-of-Breed.* In this type of projects, a best-of-breed policy is followed to acquire tools specialized in one of the steps necessary to transform raw clips in semantically-rich information. This approach is often followed by those who run a medium to long-term project to get full control of the SBI process by finely tuning all its critical parameters, typically aimed at implementing ad-hoc reports and dashboards to enable sophisticated analyses of the UGC. For example, SAS provides a set of modular components to support the different process phases (e.g., crawling and text mining) that can be separately tuned and used in combination with components provided by other vendors.
- *Level 2: End-to-End.* Here, an end-to-end software/service is acquired and tuned. Customers only need to carry out a limited set of tuning activities that are typically related to the subject area, while a service provider or a system integrator ensures the effectiveness of the technical (and domain-independent) phases of the SBI process. Examples of tools in this category are Brandwatch and Tracx (www.tracx.com), both offered in a software-as-a-service fashion and able to manage most phases of the SBI process.
- *Level 3: Off-the-Shelf.* This type of projects consists in adopting, typically in a *as-a-service* manner, an off-the-shelf solution supporting a set of reports and dashboards that can satisfy the most frequent user needs in the SBI area (e.g., average sentiment, top topics, trending topics, and their breakdown by source/author /sex). With this approach the customer has a very limited view of the single activities that constitute the SBI process, so she has little or no chance of positively impacting on activities that are not directly related to the analysis of the final results. The service provider, for instance Lexalytics or Verint, is in charge of ensuring the effectiveness of the process.

Moving from level 1 to 3, projects require less technical capabilities from customers and ensure a shorter set-up time, but they also allow less control of the overall effectiveness and less flexibility in analyzing the results. Noticeably, our architecture fits projects of all three levels —though only in a best-of-breed project customers would have a direct and complete view of all the components.

4. METHODOLOGICAL FRAMEWORK

The iterative methodology we propose is aimed at letting harmoniously coexist all the activities involved in an SBI project. These activities are to be carried out in tight connection one to each other, always keeping in mind that each

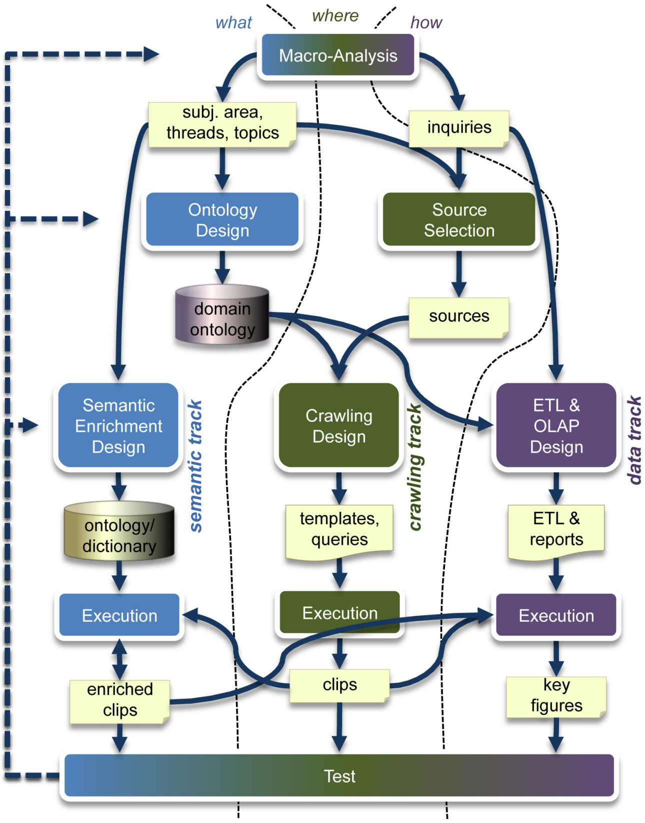

Figure 2: Functional view of our methodology for SBI design

of them heavily affects the overall system performance and that a single problem can easily neutralize all other optimization efforts.

The activities that make up our methodology are shown in Figure 2; they were conceived on the one hand to support and speed up the initial design of an SBI process, on the other to maximize the effectiveness of the user analyses by continuously optimizing and refining all its phases. These maintenance activities are necessary in SBI projects because of the continuous —and often quite fast— environment variability mainly related to volatile nature of web data sources, which asks for high responsiveness. This variability impacts every single activity, from crawling design to semantic enrichment design, and leads to constantly having to cope with changes in requirements. Note that three tracks are depicted in Figure 2: a *crawling track* centered on the crawling component, a *semantic track* centered on semantic enrichment, and a *data track* centered on ETL and OLAP. These tracks, whose activities require different technical skills and may be executed by different team members, are partially concurrent in our methodology but they still require a high coordination.

Table 1 shows which activities and single tasks are carried out for each track by the customer depending on the project level as defined in Section 1. The remaining activities are either in charge of the service provider/system integrator or they are not carried out at all, thus reducing the effectiveness of the SBI process. On the other hand, Table 2 shows, with reference to a level-1 project, the team roles mainly involved in each activity and task, distinguishing between designer, programmer, and end-user. The underlying idea is that a programmer, besides showing database and BI skills, should be competent in information retrieval, text mining, and NLP; the designer is a 360° SBI expert and must be able to guide the customer through all the crucial decisions required by the project, ranging from properly picking the crawling keywords to correctly organizing the topic ontology.

4.1 Macro-Analysis

During this activity, users are interviewed to define the project scope and the set of inquiries the system will answer to. An *inquiry* captures an informative need of a user; from a conceptual point of view it is specified by three components: *what*, i.e., one or more topic on which the inquiry is focused (e.g., the Prime Minister); *how*, i.e., the type of analysis the user is interested in (e.g., top related topics); *where*, i.e., the data sources to be searched (e.g., the Wall Street Journal website).

Inquiries drive the definition of subject area, themes, and topics. The *subject area* of a project is the domain of interest for the users (e.g., Italian national politics), meant as the set of themes about which information is to be collected. A *theme* (e.g., education) includes a set of specific *topics* (e.g., school reform). Laying down themes and topics at this early stage is useful as a foundation for designing a core taxonomy of topics during the first iteration of ontology design; themes can also be used to enforce an incremental decomposition of the project. In practice, this activity should also produce a first assessment of which sources cannot be excluded from the source selection activity since they are considered as extremely relevant (e.g., the corporate website and Facebook pages).

Two examples of inquiries in the subject area of Italian politics are:

- *What are the reactions to the Job Act proposed by the Prime Minister on newspapers belonging to different political areas?*
- *To what extent European Election themes influence the Prime Minister behavior?*

These inquires determine two different themes, namely *labor policy* and *European policy*, and several topics such as *welfare*, *minimum wage*, *Maastricht Treaty*, and *Eurosceptic*.

4.2 Ontology Design

During this activity, customers work on themes and topics to build and refine the domain ontology that models the subject area (see [17] for a survey of techniques for ontology design). Noticeably, the domain ontology is not just a list of keywords; indeed, it can also model relationships (e.g., hasKind, isMemberOf) between topics. An excerpt from the domain ontology for the Italian politics subject area, designed using Protégé, is shown in Figure 1. Once designed, this ontology becomes a key input for almost all process phases: semantic enrichment relies on the domain ontology to better understand UGC meaning; crawling design benefits from topics in the ontology to develop better crawling queries and establish the content relevance; ETL and OLAP design heavily uses the ontology to develop more expressive, comprehensive, and intuitive dashboards. With

Table 1: Activities (in italics) and tasks in charge of the customer for different project types, grouped by track; tasks executed in projects of higher levels are carried out in projects of lower levels too

	Crawling Track	Semantic Track	Data Track
Level 3	"where" macro-analysis	"what" macro-analysis	"how" macro-analysis
Level 2	*source selection*, query design, content relevance analysis	*ontology design*, polarization, correctness analysis	KPI & dashboard design
Level 1	template design	dictionary enrichment, inter-word relation definition	ETL design & implementation

Table 2: Activities/tasks and their assignment to team roles in a level-1 project

	Crawling Track	Semantic Track	Data Track
Designer	"where" macro-analysis, *source selection*, query design	"what" macro-analysis, *ontology design*, dictionary enrichment, inter-word relation definition	"how" macro-analysis, ETL design & implementation, KPI & dashboard design
Programmer	template design, content relevance analysis	ontology coverage analysis, dictionary enrichment, inter-word relation definition, correctness analysis	ETL design & implementation, KPI & dashboard design
End-User	"where" macro-analysis, *source selection*, query design, content relevance analysis	"what" macro-analysis, *ontology design*, dictionary enrichment, inter-word relation definition, correctness analysis	"how" macro-analysis, KPI & dashboard design

reference to Figure 1, organizing the politicians in a hierarchy allows to roll-up from a politician to its party and to its wing, which means for instance that the opinions about a wing can be obtained as an average of the opinions about all the politicians belonging to the parties of that wing.

The complexity of ontology design, maintenance, and evolution may vary according to the adopted tool and techniques [18, 26, 10]. In practice, the main task of this activity consists in detecting as many domain-relevant topics and themes as possible and organizing them into a classification hierarchy. In most cases this entails distinguishing all the existing relationships between topics and expressing them into a categorization framework with a fixed number of predefined levels that supports the types of analyses users are expected to carry out. This fixed-depth limitation can actually be overcome; for instance, in the architecture proposed in [6], a *meta-star* solution enables topics to be arranged in a dynamic and recursive hierarchy so as to support more sophisticated OLAP queries.

An effective way to measure the ontology maturity level is to use as an indicator the *coverage* that the ontology achieves of the retrieved clips (i.e., the percentage of clips that include at least one ontology topic). Obviously, the goal is to achieve a 100% coverage, meaning that all the clips retrieved are relevant to the subject area. This gives rise to an important task of ontology design, which we call *ontology coverage analysis*. Unfortunately, the coverage tends to decrease in time due to the dynamism of the UGC. Indeed, new potentially relevant keywords are continuously brought to the users' attention by an analysis of the retrieved clips; if these keywords are confirmed to be relevant (so-called *emerging topics*), they must be timely included in the crawling queries so as to avoid that some interesting UGC is missed and some critical trend or phenomenon is not detected. This leads to enlarging the scope of retrieved clips, and inevitably to reduce the coverage. Once it is confirmed that the emerging topic is really pertinent to the project scope, it must be added to the domain ontology and related to the existing topics so as to increase the coverage again. Note that assessing the ontology coverage is made harder by off-topic clips (see Section 4.4) that negatively impact on the coverage; this induces a strong connection between ontology design and crawling design.

4.3 Source Selection

Source selection is aimed at identifying as many web domains as possible for crawling. The set of potentially relevant sources can be split in two families: *primary sources* and *minor sources*. The first set includes all the sources mentioned during the first macro-analysis iteration, namely:

1. All the corporate communication channels (the corporate website, Facebook page, Twitter account, and any other official brand profile on any platform). Every interaction recorded on these sources and every opinion expressed on these media could be critical and has to be brought to the company attention as soon as possible.

2. So-called *generalist* sources, such as the online version of the major publications. Though these sources publish information dealing with several areas, not only with the project one, they must be monitored because of their wide user-base and of their quality and credibility.

The user-base of minor sources is smaller but not less relevant to the project scope. Minor sources include lots of small platforms which produce valuable information with high informative value because of their major focus on themes related to the subject area: in short, a small group of users who generate a high rate of pertinent clips.

There are several ways for identifying the set of potentially relevant sources: (i) Conducting interviews with domain experts, who usually are marketing operators; (ii) Analyzing back-links and third-party references to the corporate communication channels; (iii) Searching the web using themes and topics as keywords, which can be done through search engines ranging from generalist ones such as Google to domain-specific and platform-specific ones such as Openpolis and Social Mention; (iv) Considering all the local editions of major newspapers.

Once a set of candidate sources has been established, deciding which of them are to be actually crawled is the result of a trade-off between achieving a satisfying coverage of the subject area on the one hand, and optimizing the effort for analyzing the retrieved clips. To evaluate this trade-off, tools such as web directories (e.g., Alexa) can be used to estimate the number of accesses and traffic level of candidate web sites. Of course, even the set of selected sources must be maintained, so the web must be periodically monitored to evaluate and dynamically include new relevant sources.

4.4 Crawling Design

A relevant source that produces *in-topic* clips, normally also generates lots of valueless content (*off-topic* clips) that lies outside the project scope and slows down the whole process while possibly hiding relevant content. Crawling design, maybe one of the most complex and time-consuming activities, aims at retrieving in-topic clips by filtering off-topic clips out. Starting from the topics in the domain ontology and from the additional keywords possibly discovered during source selection, a set of queries are created to search for relevant clips across the selected sources. Three subsequent tasks are involved in this activity:

1. *Template design* consists in an analysis of the code structure of the source website to enable the crawler to detect and extract only the informative UGC (e.g., by excluding external links, advertising, multimedia, and so on).

2. Based on the templates designed, *query design* develops a set of queries to extract the relevant clips. Normally, these are complex Boolean queries that explicitly mention both relevant keywords to extract on-topic clips and irrelevant keywords to exclude off-topic clips.

3. *Content relevance analysis* aims at evaluating the effectiveness of crawling by measuring the percentage of in-topic clips. At this stage, the analysis must be carried out by manually labeling a sample of the retrieved clips. Besides distinguishing between in-topic and off-topic clips, users should also try to classify the causes of errors to speed up the following iterations. Identifying clips that have been retrieved due to an incorrect query template and keywords that led to extracting off-topic clips is typically very useful, because it enables the team to trigger a new iteration where crawling queries are refined to more effectively cut off-topic clips out.

Note that filtering off-topic clips at crawling time could be difficult due to the limitations of the crawling language, and also risky because the in-topic perimeter could change during the analysis process. For these reasons, the team can choose to release some constraints aimed at letting a wider set of clips "slip through the net", and only filter them at a later stage using the search features of the underlying document DBMS (e.g., MongoDB). This choice must be carefully evaluated: on the one hand, it implies that more data are retrieved and stored, on the other hand, it enables the team to delay (and possibly to change) its decisions about the in-topic perimeter. Even if the team chooses to let more clips enter the document repository, the inherent nature of tasks 2 and 3 does not change; however, these tasks are partially delayed to a further filtering step to be carried out before semantic enrichment.

With reference to Italian politics, a simple crawling query that retrieves clips related to the opinions expressed in Italy about the debate between the Italian and German Prime Ministers about the Italian debit, could be `((Renzi AND Merkel) NEAR/20 (deficit OR raw:3%)) AND country:it`[1].

4.5 Semantic Enrichment Design

This activity involves several tasks whose purpose is to increase the accuracy of text analytics so as to maximize the process effectiveness in terms of extracted *entities* and sentiment assigned to clips; entities are concepts that emerge from semantic enrichment but are not part of the domain ontology yet (for instance, they could be emerging topics). The specific tasks to be performed depend on the semantic engine adopted and on how semantic enrichment is carried out. For instance, SyN Semantic Center (used for both case studies presented in this paper) executes a two-steps process [16]: first, relevant knowledge is identified from the clips through lexical analysis, i.e., by detecting semantic relations and facts based on the *slot grammar method* [15] and adopting morphological, syntactic, semantic, semiometric, and statistical criteria; then, clips are classified according to their topics using both supervised and unsupervised clustering criteria.

In general, two main tasks that enrich and improve its linguistic resources can be distinguished:

- *Dictionary enrichment*, that requires including new entities missing from the dictionary and changing the sentiment of entities (*polarization*) according to the specific subject area (e.g., in "I always eat fried cutlet", the word "fried" has a positive sentiment, but in the food market area a sentence like "These cutlets taste like fried" should be tagged with a negative sentiment because fried food is not considered to be healthy). Here, a typical error is related to failing to connect an entity to its different synonyms or aliases, which dramatically distorts all the figures based on counting topic occurrences. To avoid this problem, a layer of aliases can be added between topics and entities. Aliases are useful to associate to a single topic entities that can differ from the given topic due to typos or due to the use of synonyms. For example, in the Italian politics domain, "PD" and "PD – L" are both synonyms of "Partito Democratico". Such knowledge can be hosted either in the ontology (see [6]) or within the semantic engine.

- *Inter-word relation definition*, that establishes or modifies the existing semantic, and sometimes also syn-

[1] The query syntax is the one used by Brandwatch.

tactic, relations between words. Relations are linguistically relevant because they can deeply modify the meaning of a word or even the sentiment of an entire sentence determining the difference between right and wrong interpretation (e.g., "a Pyrrhic victory" has negative sentiment though "victory" is positive).

Modifications in the linguistic resources may produce undesired side effects; so, after completing these tasks, a *correctness analysis* should be executed aimed at measuring the actual improvements introduced and the overall ability of the process in understanding a text and assigning the right sentiment to it. This is normally done, using regressive test techniques, by manually tagging an incrementally-built sample set of clips with a sentiment; it is always recommended to ask different users for tagging clips, and then use a voting system to determine a majority group that will be considered as an oracle. The overall correctness of semantic enrichment strongly depends on the selected sources and on how specific the subject area is. Reaching a correctness level of 70% can be seen as a very good result considering that, according to the literature, a realistic upper bound to the *inter-tagger agreement* among three or more users when manually tagging clips is around 70% [1].

4.6 ETL & OLAP Design

The main tasks in this activity are:

- *ETL design and implementation*, that strongly depends on features of the semantic engine, on the richness of the meta-data retrieved by the crawler (e.g., URLs, author, source type, platform type), and on the possible presence of specific data acquisition channels like CRM, enterprise databases, etc.

- *KPI design*; different kinds of KPIs can be designed and calculated depending on which kinds of meta-data the crawler fetches.

- *Dashboard design*, during which a set of reports is built that captures the user needs expressed by inquiries during macro-analysis. Note that, in some cases, the specific tool adopted for reporting may be unable to satisfactorily meet the users requirements; in this case a totally custom interface should be implemented from scratch.

4.7 Execution and Test

This activity has a basic role in the methodology, as it triggers a new iteration in the design process. Crawling queries are executed, the resulting clips are processed, and the reports are launched over the enriched clips. The specific tests related to each single activity, described in the preceding subsections, can be executed separately though they are obviously inter-related. The first test executed is normally the one of crawling; even after a first round, the semantic enrichment tests can be run on the resulting clips. Similarly, when the first enriched clips are available, the test of ETL and OLAP can be triggered.

The test results are inspected with users, which may easily lead to:

1. go back to crawling design to better tune the crawling queries or templates to improve precision and recall;

2. go back to semantic enrichment design to solve problems related for instance to misunderstandings of sentences or wrong polarization of clips.

3. go back to ETL & OLAP design to fix ETL errors and improve reports and dashboards;

4. go back to ontology design to further enrich and extend the domain ontology with new relevant entities that emerged from an analysis of the clips retrieved;

5. go back to source selection to include new sources or exclude some sources that are no longer relevant;

6. go back to macro-analysis to enlarge the subject area or refine inquiries.

5. CASE STUDIES

In this section we will describe our experience with two real SBI projects, which helped us in tailoring our methodology and demonstrate that an engineered approach positively impacts on the project success, meant in terms of both correctness and productivity. In particular we will analyze two projects: a level-1 project in the subject area of Italian politics (PR-Pol) and a level-2 project in the subject area of a large consumer goods company (PR-CG). Both projects adopted an iterative approach and the tasks carried out are approximately the same, but while in PR-Pol our methodology was enforced, in PR-CG the team was mainly guided by its previous experience. As shown later, this will lead to some inefficiencies in PR-CG.

The PR-CG working group was led by a system integrator with significant skills in SBI, featuring one project manager, one chief of consulting services, and six developers. The team was completed by an external scientific supervisor and by the innovation chief of the customer company. Though PR-CG was a level-2 project, we had a chance to monitor the activities of both the customer and the system integrator. The PR-Pol working group was quite smaller: it only included one project manager, one scientific supervisor, two developers, and the customer (the mayor of a large Italian city in this case). Overall, though the two projects are not fully comparable in terms of size and working group composition, they cover most of the critical issues related to SBI projects so they provide a good support for discussing the features of our methodology.

According to the classification proposed by [24], our case studies can be described as *explanatory/exploratory* (they aim at confirming the effectiveness of our methodology in real contexts, but also at finding new insights and at better tuning the approach), *positivist* (they use effort and correctness measurements), *quantitative* and *qualitative* (they quantitatively assess the validity of the approach, but they also collect qualitative judgments by the team), and *flexible* (due to the inherent dynamics of an SBI project, the requirements continuously change during the case studies). A more complete description can be given by answering the basic questions proposed by [22]:

- *Objective—What to achieve?*: the case studies aim at proving that the adoption of our methodology has a positive impact on the productivity and correctness of SBI projects.

- *The case—What is studied?*: we study two real projects with different characteristics and in different areas; both projects were carried out by skilled teams but with different compositions and size.

- *Theory—Frame of reference*: the theoretical framework we adopted is the one defined by the activities and tasks our methodology builds upon.

- *Research questions—What to know?*: we study how the two projects differ in terms of required effort and delivered utility.

- *Methods—How to collect data?*: for PR-CG, the effort for the different activities and tasks was derived a posteriori from an analysis of the time-sheets recorded by the system integrator, while for PR-Pol it has been measured at project time; as to correctness, it has been estimated by asking some domain experts to manually tag a set of clips and comparing the resulting tags with those automatically obtained by semantic enrichment.

- *Selection strategy—Where to seek data?*: we selected two projects of different levels to achieve a wide coverage of the aspects involved in SBI design. PR-Pol was a level-1 project on a very wide and dynamic domain, led by a small team; PR-CG was a level-2 project on a more narrow domain, led by a system integrator.

In Table 3 we show the time spent on each task distinguishing the first iterations from the maintenance ones; missing items in the maintenance column denote activities made on demand, i.e., only at some iterations. Some comments on the values reported are necessary:

- Even if macro-analysis poses no particular problems, it usually requires a large amount of time because it is carried out during non-technical meetings that involve several different corporate departments.

- Maintaining the domain ontology requires more time in PR-Pol than in PR-CG. The reason is that the Italian politics subject area is quite wider than the consumer goods one, which implies a larger amount of dynamic contents to be analyzed in order to verify which new topics are to be added to the ontology.

- The time saving in semantic enrichment design for PR-Pol is mainly due to the adoption of a structured set of tests that has led the team to easily obtain the desired level of performances. This time saving is not apparent in maintenance iterations due to the higher complexity of the politics subject area.

- In query design and content relevance analysis, the amount of time needed to test how the developed queries work largely depends on the project level. In a level-2 project, the customer usually delegates crawling to an external service provider, who normally is capable of estimating the volume of clips retrieved by each specified query. Conversely, in a level-1 project, crawling has to be managed in every aspect, so that the effectiveness of a query can only be assessed after a whole clip acquisition session, that usually lasts 24 hours; as a result, the execution of this activity can be significantly longer.

- The customer's effort is clearly reduced in a level-2 project. In particular, if no external provider is used for crawling, template design may end up for being very time consuming, which results in the largest time overhead.

As to semantic enrichment design, we will focus on sentiment analysis, one of the more complex and important phases of the SBI process, that consists in determining the sentiment associated to a specific clip. Though the correctness of this analysis is obviously related to the capabilities of the semantic enrichment engine, a fine tuning can lead to dramatic improvements. Both our projects share the same engine: *SyN Semantic Center*, a well-known commercial suite that enables a linguistic and semantic analysis of any piece of textual information based on its morphology, syntax, and semantics using logical-functional rules. So we investigated how the correctness of sentiment analysis was affected by the adoption of our methodology by asking five domain experts to manually tag a large set of clips (about $1,500$) with their sentiment and then submitting them to the tuned/non-tuned engine. Tuning had a similar duration in the two projects (about two months) and led to a similar number of changes in the engine (about 330). Table 4 shows the results: clips are classified according to three criteria (media type, difficulty of a human expert in defining the sentiment, sentiment); the correct sentiment is assumed to be the one chosen by the majority of the domain experts. The semantic engine initially performed worse for Pr-Pol than for Pr-CG because the politics subject area uses a wider terminology and is probably more complex than the consumer goods one. However, the improvements obtained for Pr-Pol are clearly larger than those for Pr-CG. An in-depth analysis of the approach adopted by the Pr-CG team evidenced a lack of attention to the side effects of word polarization, that often introduced as many errors as those that were solved. Conversely, a more structured approach (see Section 4.5) and a continuous and iterative check of the side effects made the PR-Pol team's effort more effective.

Our case studies confirmed that ontology design and crawling design are the two most strictly-coupled activities and that their synchronization is a key factor to increase the overall performance. On the one hand, within crawling design, the query design and content relevance analysis tasks are based on the topics determined by ontology design; on the other, the coverage achieved for the domain ontology mostly depends on how effectively crawling is able to exclude off-topic clips. In PR-Pol, at each iteration of ontology design, coverage analysis of the available clips is always made twice: once before adding new topics and once afterwords. The clips that remain uncovered are then handed on, together with the updated ontology, to crawling design and signaled as off-topic clips (i.e., crawling queries must be updated to discard these clips). This simple but effective protocol is applied every two days; in about 8 solar weeks the topics in the ontology increased from 139 to 225, and its coverage from 93% to 98%.

6. DISCUSSION AND FINAL REMARKS

Responsiveness in an SBI project is not a choice but rather a necessity, since the frequency of changes requires a tight involvement of domain experts to detect these changes and rapid iterations to keep the process well-tuned. Such a fran-

Table 3: Time spent on tasks, expressed in man-days for first iterations and in man-days per week in maintenance iterations (n.a. stands for not available because the task has been outsourced)

Activity/Task	PR-CG		PR-Pol	
	1st Iter.	Maint. Iter.	1st Iter.	Maint. Iter.
Macro-Analysis	10	—	9	—
Ontology Design	4	0.6	7	1.5
Topics Definition	2	0.5	2	1
Inter-Topic Relation Definition	2	0.1	5	0.5
Source Selection	3	1	5	1
Semantic Enrichment Design	7	0.75	5	1
Crawling Design	10	1	29	1.5
Template Design	n.a.	n.a.	15	—
Query Design & Content Relevance Analysis	10	1	14	1.5
ETL & OLAP Design	15	—	24	—
ETL Design & Implementation	5	—	10	—
KPI Design	5	—	7	—
Dashboard design	5	—	7	—
Execution & Test	3	—	5	—
Total	52	3.35	84	5
In charge to the customer	15	0.85	84	5

Table 4: Correctness of sentiment analysis

	PR-CG			PR-Pol		
	Non-tuned	Tuned	Improvement	Non-tuned	Tuned	Improvement
Total	54.0%	57.4%	3.4%	51.8%	60.3%	8.5%
Social	52.5%	55.9%	3.4%	46.1%	58.1%	12.0%
Qualified	55.0%	58.3%	3.3%	54.6%	61.4%	6.8%
Hard	34.3%	37.2%	2.9%	35.0%	47.0%	12.0%
Standard	67.3%	71.1%	3.8%	61.4%	68.1%	6.7%
Negative	46.6%	46.6%	0.0%	50.5%	59.7%	9.2%
Neutral	45.6%	49.1%	3.5%	62.0%	71.3%	9.3%
Positive	69.5%	76.3%	6.8%	47.8%	52.4%	4.6%

tic setting imposes a radical change in the project management approach with reference to traditional BI projects and a huge effort to both end users and developers (about one full-time person in both our projects). To reduce such effort, customers often try to outsource the activities yielding the worst trade-off between effort and added value for the SBI process. Besides the different technical skills required, this is the main motivation for conducting a level-2 project rather than a level-1 one.

During a project review session we analyzed, together with some members of the PR-CG team, the main problems they perceived, that turned out to be a lack of synchronization between the activities, that reduced their effectiveness, and an insufficient control on the effects of changes. With our methodology we tried to solve such problems through:

- A clear organization of goals and tasks for each activity.
- A protocol and a set of templates (not discussed in this paper for brevity) to record and share information between activities.
- A set of tests to be applied. The definition of each test includes the testing method and the indicators that measure the test results, for instance in terms of correctness of a process phase, as well as how these results have improved over the previous iteration.

7. REFERENCES

[1] T. Chklovski and R. Mihalcea. Exploiting agreement and disagreement of human annotators for word sense disambiguation. In *Proc. RANLP*, Borovets, Bulgaria, 2003.

[2] V. Crescenzi, G. Mecca, P. Merialdo, and P. Missier. An automatic data grabber for large web sites. In *Proc. VLDB*, pages 1321–1324, 2004.

[3] 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.

[4] M. Diligenti, F. Coetzee, S. Lawrence, L. Giles, and M. Gori. Focused crawling using context graphs. In *Proc. VLDB*, pages 527–534, Cairo, Egypt, 2000.

[5] T. Furche, G. Gottlob, G. Grasso, C. Schallhart, and A. J. Sellers. OXPath: A language for scalable data extraction, automation, and crawling on the deep web. *VLDB J.*, 22(1):47–72, 2013.

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

[7] 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.

[8] M. Golfarelli and S. Rizzi. *Data Warehouse design: Modern principles and methodologies*. McGraw-Hill, 2009.

[9] M. Golfarelli, S. Rizzi, and E. Turricchia. Modern software engineering methodologies meet data warehouse design: 4WD. In *Proc. DaWaK*, pages 66–79, Toulouse, France, 2011.

[10] M. Hepp, P. D. Leenheer, A. de Moor, and Y. Sure, editors. *Ontology Management*, volume 7 of *Semantic Web And Beyond Computing for Human Experience*. Springer, 2008.

[11] R. Hughes. *Agile Data Warehousing: Delivering world-class business intelligence systems using Scrum and XP*. IUniverse, 2008.

[12] J. Kahan and M.-R. Koivunen. Annotea: an open RDF infrastructure for shared web annotations. In *Proc. WWW*, pages 623–632, Hong Kong, China, 2001.

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

[14] J.-N. Mazón and J. Trujillo. An MDA approach for the development of data warehouses. In *Proc. JISBD*, pages 208–208, 2009.

[15] M. McCord. Slot grammar: A system for simpler construction of practical natural language grammars. In R. Studer, editor, *Natural Language and Logic*, volume 459 of *Lecture Notes in Computer Science*, pages 118–145. Springer, 1989.

[16] F. Neri, C. Aliprandi, and F. Camillo. Mining the web to monitor the political consensus. In U. K. Wiil, editor, *Counterterrorism and Open Source Intelligence*, volume 2 of *Lecture Notes in Social Networks*, pages 391–412. Springer, 2011.

[17] N. F. Noy and C. Hafner. The state of the art in ontology design: A survey and comparative review. *AI Magazine*, 18(3):53–74, 1997.

[18] N. F. Noy and M. C. A. Klein. Ontology evolution: Not the same as schema evolution. *Knowl. Inf. Syst.*, 6(4):428–440, 2004.

[19] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proc. EMNLP*, pages 79–86, 2002.

[20] S. Raghavan and H. Garcia-Molina. Crawling the hidden web. In *Proc. VLDB*, pages 129–138, Rome, Italy, 2001.

[21] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proc. EMNLP*, pages 1524–1534, Edinburgh, Scotland, 2011.

[22] C. Robson. *Real World Research*. Blackwell, 2002.

[23] B. Rosenfeld and R. Feldman. Clustering for unsupervised relation identification. In *Proc. CIKM*, pages 411–418, Lisbon, Portugal, 2007.

[24] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.

[25] A. Sen and A. P. Sinha. A comparison of data warehousing methodologies. *Commun. ACM*, 48(3):79–84, 2005.

[26] L. Stojanovic. *Methods and tools for ontology evolution*. PhD thesis, Forschungszentrum Informatik, Karlsruhe, 2004.

[27] 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.

[28] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proc. ICDM*, pages 697–702, Washington, DC, USA, 2007.

[29] J. Yi, T. Nasukawa, R. C. Bunescu, and W. Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proc. ICDM*, pages 427–434, Melbourne, Florida, 2003.

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