En association avec EGC 2011

25 Janvier 2011 à Brest

Deuxième édition de l'atelier "Sources Ouvertes et Services"

SOS'2011

Table des matières

Comité de programme

Yamine Ait Ameur (LISI/ENSMA, Université de Poitiers)
Gael Blondelle (Obeo)
Sebastien Cantarell (DGA)
Gael de Chalendar (CEA/LIST)
Jean Delahousse (Mondeca)
Valentina Dragos (Onera)
Adil El Ghali (IBM)
Christian Fluhr (Geol. Semantics)
Fabien Gandon (INRIA Sophia Antipolis)
Patrick Giroux (Cassidian)
Bruno Grilheres (Cassidian)
Nicolas Hernandez (Université de Nantes)
Abdel-Illah Mouaddib (Greyc, Université de Caen)
Alexandre Pauchet (LITIS, Rouen)
Yann Pollet (CNAM France)
Haïfa Zargayouna (LIPN, Université de Paris 13)

Comité d'organisation

Président :

Stephan Brunessaux (IPCC - CASSIDIAN)

Membres:

Khaled Khelif (IPCC - CASSIDIAN)
Arnaud Saval (IPCC - CASSIDIAN)

Data Acquisition in Social Networks: Issues and Proposals

Claudia Canali, Michele Colajanni, Riccardo Lancellotti

Department of Information Engineering
University of Modena and Reggio Emilia
{claudia.canali, michele.colajanni, riccardo.lancellotti}@unimore.it

Abstract. The amount of information that is possible to gather from social networks may be useful to different contexts ranging from marketing to intelligence. In this paper, we describe the three main techniques for data acquisition in social networks, the conditions under which they can be applied, and the open problems. We then focus on the main issues that crawlers have to address for getting data from social networks, and we propose a novel solution that exploits the cloud computing paradigm for crawling. The proposed crawler is modular by design and relies on a large number of distributed nodes and on the MapReduce framework to speedup the data collection process from large social networks.

1 Introduction

An unprecedented explosion of user generated content is available on social networks that, thanks to their growing popularity, are gaining top importance as sources of valuable information. Two thirds of the world's Internet population visit a social network site weekly, and the time spent on these sites accounts for more than 10% of all Internet time, with this percentage growing three times faster than the rate of the overall Internet growth [NielsenWire (2009); Canali et al. (2009)].

The growth in terms of registered users motivates the increasing interest of academic and industrial researchers in social networks for different goals including workload characterization, marketing purposes, understanding and forecasting Internet use, identifying main challenges to support future Internet-based services. Unfortunately, collecting data from social networks is a real challenge with respect to other Internet-based sources, such as Web pages, blogs, peer-to-peer systems. Some problems are related to the size and the intrinsic complexity of a social network. Other problems are related to the heterogeneity of the target. Indeed, we have to consider that the term social network identifies a broad category of applications that may differ in many ways, ranging from user communication styles, types of exchanged contents, privacy settings, etc. Hence, the one-size-fits-all approach to data acquisition is unfeasible. The state of the art of academic research includes three main techniques for data acquisition from social networks: network traffic sniffing [Gill et al. (2007); Nazir et al. (2009)]; implementation of specific applications for each social network [Nazir et al. (2008)]; crawling of the user social graph. The last method is the most popular approach for social networks where user data are publicly available, such as MySpace. Traffic sniffing and specific applications are limited to specific contexts where crawling is unfeasible. For example, traffic analysis of dormitory networks was adopted to characterize the access patterns of college students in Facebook [Nazir et al. (2009)]. Throughout this paper we initially present the three main techniques adopted for data acquisition in social networks and we identify the main

open issues of each of them. We then focus on *social network crawling*, and identify the limits of current crawlers through analysis and experimental results. This study induced us to propose an innovative solution that exploits the potential of cloud computing to support data collection in the context of social networks. The novel framework is based on a crawling algorithm that exploits the MapReduce computing paradigm [Dean and Ghemawat (2008)]. This proposal has several advantages: it allows us to speed up the data collection process by distributing the operations on a large number of nodes performing parallel crawling; it reduces the risk of triggering the countermeasures adopted by social network operators against extensive and automatic crawling; it favors the implementation of a modular and easily customizable crawler that is able to acquire data from different social networks.

The remainder of this paper is structured as follows. Section 2 describes the main issues affecting data acquisition from social networks with a special focus on crawling. Section 3 presents the proposed architecture for crawling social networks based on the cloud computing paradigm. Section 4 discusses the related work. Section 5 concludes the paper with some final remarks.

2 Data acquisition in social networks

The social network term includes a broad category of applications. In this paper, we define *social network* any Web-based site that offers the user the possibility to register, to interact with other users, to share contents of any nature through any sort of social links. Similar sites include the most traditional Facebook, LinkedIn, MySpace, and Orkut, but also the sites that have added a lot of social network facilities, such as YouTube, Flickr, Digg, and Twitter.

Acquiring data from a social network requires an exploration of the user population with the goal of collecting different kinds of information, such as the network links among the users, uploaded and downloaded contents, rating, comments [Khrishanmurthy (2009)]. In this paper we are interested to the data collection process only, hence we do not delve into details of which data can be acquired from a social network and which analyses can be carried out on these data. In a similar way, we do not consider typical data anonymization techniques that typically are carried out only after the data collection process.

Here, we outline the three main techniques proposed in literature to acquire data from social networks, and we then discuss in detail some issues we have experienced with crawling. The considered techniques are:

- Network traffic analysis [Gill et al. (2007); Nazir et al. (2009)]
- Ad-hoc applications [Nazir et al. (2008)]
- Crawling the user graph [Mislove et al. (2007); Cha et al. (2008); Lerman (2007); Cha et al. (2009)]

2.1 Network traffic analysis

This is a typical traffic sniffing and analysis technique that captures packet streams from a network link and then analyzes request-response pairs from network traces involving user interactions with a social network. From these request-response pairs it is possible to infer information about social browsing through the network content that is, which users are visiting other users pages. Furthermore, information about the users can be obtained by the analysis of the response payload, that contains the Web pages. While from a theoretical point of view, this approach is always feasible, accurate sniffing of high volume traces presents technical and legal issues [Crovella and Krishnamurthy (2006)].

– All countries impose restrictions on network traffic analysis to protect the privacy of citizens. Collection of network traffic can be carried out only in private contexts and for limited periods of time, such as university campuses or companies, where users are typically notified about the experiments being carried out. As a consequence, the collected data may be not representative of the entire social network user population.
– Traffic analysis in high speed links presents issues as it may overload the packet sniffing mechanism. The resulting loss of data may hinder the detection of request-response links and may reduce the amount of data that is viable for subsequent analyses. To address this issue it is necessary to deploy the packet capture system on a parallel architecture that may be difficult to deploy [Andreolini et al. (2007)].
– In order to extract the payload useful information from the payload of the responses it is necessary to parse the supplied Web pages. As the structure of the Web pages is specific for each social network, it is impossible to implement a general tool to collect data from different social networks, and the support of each social network requires a significant development effort.

2.2 Ad-hoc applications

Ad-hoc application are third-party applications that exploits a set of APIs, such as the Facebook Developer Platform or OpenSocial, to provide services and games to the social network users. In this architecture, a user does not interact directly with the application servers because the social network infrastructure provides an interface layer between user and application. Developing ad-hoc applications to acquire data from social networks allows to collect information about the users in a twofold way. First, the APIs typically allow the application to access information about the profile of users who are registered to the application. Furthermore, the analysis of the log on the application servers allows to extract information about the dynamic user behavior. The analysis of a social network through an ad-hoc application is not affected by the severe legal issues related to traffic sniffing because users must explicitly register to the application and accept the possibility of information disclosure, this technique is not free of issues.

First of all, it can be used only when the social network (e.g., Facebook [Nazir et al. (2008)]) provides third parties with specific APIs for adding new applications. Moreover, the size of the dataset that may be collected depends on the popularity of the applications: if the applications do not attract many subscribers, the available dataset is limited and useless for analysis purposes. We have also to consider that this approach requires the implementation of novel applications that are specifically designed for one targeted social network. This requires great efforts in software investments that do not guarantee returns if the application is not successful.

2.3 Crawling

Crawling is the most popular solution for data acquisition in social networks and consists on querying the social network for publicly available information about users. This approach is viable for most social networks including YouTube, Flickr, Digg, Orkut, MySpace, and Twitter [Mislove et al. (2007); Cha et al. (2008); Lerman (2007); Cha et al. (2009)]. Crawling may take further advantage of the availability of public APIs that some social network operators provide.

Crawling exploits the typical structure of a social network, that can be modeled as a directed graph $G = (U, E)$, where U is the set of nodes (users) and E is the set of edges (social links among users). Each node has *outgoing links*, and *incoming links*. For the goal of collecting information

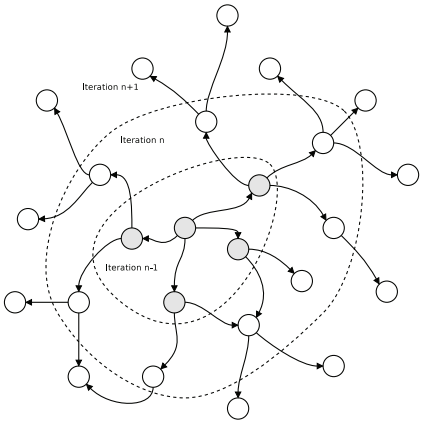


FIG. 1 – *Crawling iterative process*

about the users of a social network, we are not interested in distinguishing between the different types of social links among users (e.g., friends or followed/follower links) but only in devising a way to visit each user in the network.

Crawling the social network graph is an iterative process that starts from a set of initial users and proceeds by discovering new users at every step. The initial setup of the crawler is typically composed by a list of randomly selected users, because starting from multiple random locations is one way to improve the data collection process in terms of duration and data representativeness [Gjoka et al. (2010)].

Different methods can be used to proceed through the crawling graph. They differ in the order through which they visit new users at every step: popular approaches include Breadth-Search-First (BSF), Depth-First-Search (DFS), Forest Fire (FF) and Snowball Sampling (SBS). We adopt the BSF algorithm because it is used extensively in literature for collecting data from social networks [Mislove et al. (2007); Ahn et al. (2007); Wilson et al. (2009)]. Figure 1 illustrates the iterative crawling process following a BSF approach. For the generic step n of the data collection process the crawlers start from the set of previously discovered users (that have been identified at the crawling step $n-1$) and explore the outgoing links to not yet visited users. The set of newly identified users represents the basis for the next crawling step $n+1$. The main reason for the popularity of this method is that an (even incomplete) BSF sample collects a full view (all nodes and edges) of some region in the graph. However, BSF may lead to a bias because it tends to overestimate the node degree by privileging users with high number of social links. However, this bias may be removed by techniques that are able to collect the majority of users in the graph [Wilson et al. (2009)].

To understand the pros and cons of crawling social networks, we implement two crawlers based on the BSF algorithm and we apply them to collect data from YouTube and Digg Canali et al. (2010). We choose YouTube and Digg as examples of popular Web sites that allow users to subscribe, to create social links with other users and to share contents. According to data published by the market research company comScore [comScore comScore (2010)], YouTube is the dominant provider of online video, with a population of over 50 millions of users. In a similar way, Digg is a highly popular bookmark sharing site with more than 15 millions monthly US unique visits in 2008 [Schonfeld (2008)]. Furthermore, YouTube and Digg offer a set of public APIs that allow a

crawler to simplify its access to data of registered users and their social links.

We implemented a crawler for each site because YouTube and Digg offer different API interfaces and adopt different countermeasures to limit extensive crawling. For both crawlers our exploitation of the YouTube and Digg APIs is fully compliant with the terms of use for non-commercial purposes. The efforts for software development and the time required by the execution of the crawlers were significant. Each crawler exceeds 3 thousands lines of code in addition to a DBMS that is used as data storage and for synchronization among different runs of the same crawler. Each crawler was executed on a Pentium IV system with a processor clock of 2.3 GHz and equipped with 2 GBytes of RAM. The YouTube and Digg sites were crawled in the second half of 2009 for a period of 10 days. Table 1 reports the number of users visited and the number of social links as they are the most relevant information for the navigation on the graph $G = (U, E)$ that describes the social network structure. (As anticipated in the Introduction, in this paper we are not interested to the nature and size of collected information.)

TAB. 1 – *Information from social network graph navigation*

Parameter	**YouTube**	**Digg**
Period of crawling	20-30 Aug. 2009	15-25 Nov. 2009
Number of users	1,708,414	349,035
Number of social links	12,935,561	3,212,454

We collected data on nearly 2 million of Youtube users and more than 10 million social links, that correspond nearly to 1% of the network. Moreover, we gathered data for almost 9% of the Digg social network users. Although the data acquisition process lasted for 10 days, the crawler reached a small percentage of the social network structure. Our experience evidenced that crawling the entire user graph in this way may require months of work, thus making impossible to obtain a continuously updated knowledge about the overall network. This problem is primarily caused by the huge size of the user graph to explore. Moreover, the data collection process is further delayed by the countermeasures deployed in the social network to hinder extensive crawling, such as:

– IP banning;
– restrictions on amounts of data results.

IP banning is a typical technique used by social network infrastructures as a protection against Denial of Service (DoS) attacks. A limitation is imposed on the number of requests allowed within a specific time interval (e.g., a few thousands of requests per day) coming from the same IP address. If the amount of requests generated from the same IP address exceeds a security threshold, the servers filter out subsequent requests from that IP for a period of time (e.g., 20 minutes for YouTube). To cope with this problem, we use two network interfaces on the same machine to distribute the crawling traffic among multiple IPs. However, this solution is not sufficient to avoid the risk of IP banning, hence we have to delay subsequent requests to avoid exceeding the limits imposed by the social network operators. In summary, this solution may avoid the risk of IP banning, but it does not resolve the problem of an excessively slow crawling process.

As a further countermeasure, the social network providers limit the maximum number of results that may be returned by the APIs [Youtube Developer's Guide (2010)]. Whenever the APIs are used to download a list of data, such as the list of user outgoing links, the number of returned results is limited to a maximum value. For example, the list of returned outgoing links for YouTube is truncated to 100 entries for each user. This limitation is critical as the node degree distributions within a social network typically determine the presence of *hub* users with several thousands of links. Our crawlers address this issue by integrating pure crawling through APIs with the possibil-

ity of exploiting some information available on the user home page, that contains the complete list of his/her outgoing links. Our crawlers are able to identify the users for whom the API returns an incomplete list of outgoing links, and to integrate this list by parsing the HTML code of the user home pages. This solution is inevitable but it contributes to lengthen the crawling process because it requires separate requests to the user pages to avoid IP banning countermeasures.

To summarize, our experience evidenced that extensive crawling of graphs with millions of users is a challenging task. Traditional crawlers, like the one used in our experiments, are not able to collect a sufficiently large amount of data in reasonable periods of time due to three main issues.

- As the APIs and the structure of users Web pages are specific for every social network, each crawler must be customized and it is not possible to develop a generic crawler common to multiple social network.
- Due to huge network size (in the order of tens of millions of users) it is difficult to reach a large part of the user population in reasonable periods of time.
- The amount of data that can be collected from social networks may be limited. For example, the maximum number of user social links returned by APIs is limited in the order of one hundred, the number of requests allowed within a specific time interval coming from the same IP address is limited to a few thousand requests a day (otherwise *IP banning* occurs). To overcome these limitations, the crawler my parse information available directly on the user Web page. However, this approach is more complex from a programming point of view (to to the complexity of parsing HTML text) and is more time consuming (multiple pages must be accessed to collect the same types of information available through a single API invocation.

These issues motivate our effort to explore innovative approaches for crawling of large social networks.

3 Cloud-based crawling

In this section we describe an innovation solution for crawling, that exploits the cloud computing platform to improve data acquisition of social network data. There are three main goals behind cloud-based crawling:

- to speed up the data collection process by parallelizing the user graph exploration;
- to avoid countermeasures, such as IP banning, introduced by some social network operators to limit extensive crawling, in the respect of terms of use of provided APIs;
- to have a modular software that can guarantee data acquisition from different social networks through the same easily customizable crawler.

The proposed crawling software exploits two main features available in cloud computing platforms: the possibility of distributing the crawling processes over several virtual machines where the exact number can be decided at runtime; a set of software functions for parallel programming and management in the style of Platform as a Service (PaaS).

It is important to have the possibility of specifying at runtime the number and type of necessary virtual machines because the necessary amount of RAM and computational power depends on the size of the social network and on the number of users that are considered at each crawler iteration. The PaaS libraries guarantee a set of software functions for creation/termination of virtual machines, synchronization of parallel jobs, data exchange among jobs. We rely on the *MapReduce* programming paradigm [Dean and Ghemawat (2008)] which is supported by some important cloud computing providers, such as Amazon Elastic Compute Cloud (EC2) [Amazon Elastic Compute Cloud

(2010)]. Our design is compliant with the Hadoop framework [Hadoop MapReduce (2010)] that is the platform for the deployment of our crawler.

The proposed crawler is designed according to a modular architecture with two main parts. The first part contains the engine responsible for the coordination of parallel execution of crawlers. This module is responsible for managing the crawling process, including the choice of the parameters for the number and the characteristics of the virtual machines used for crawling, and the definition of the *map* and *reduce* functions that are used to implement over the cloud platform the crawling algorithm described in Section 3.1. The second part of the crawler software consists of specific code for data acquisition related to each social network. This module contains a standard interface with methods to retrieve information about a user and about his/her social links. However, each social network requires a different implementation of this module because the APIs, when provided, are different, and even the code to parse user home pages and to navigate them cannot be general.

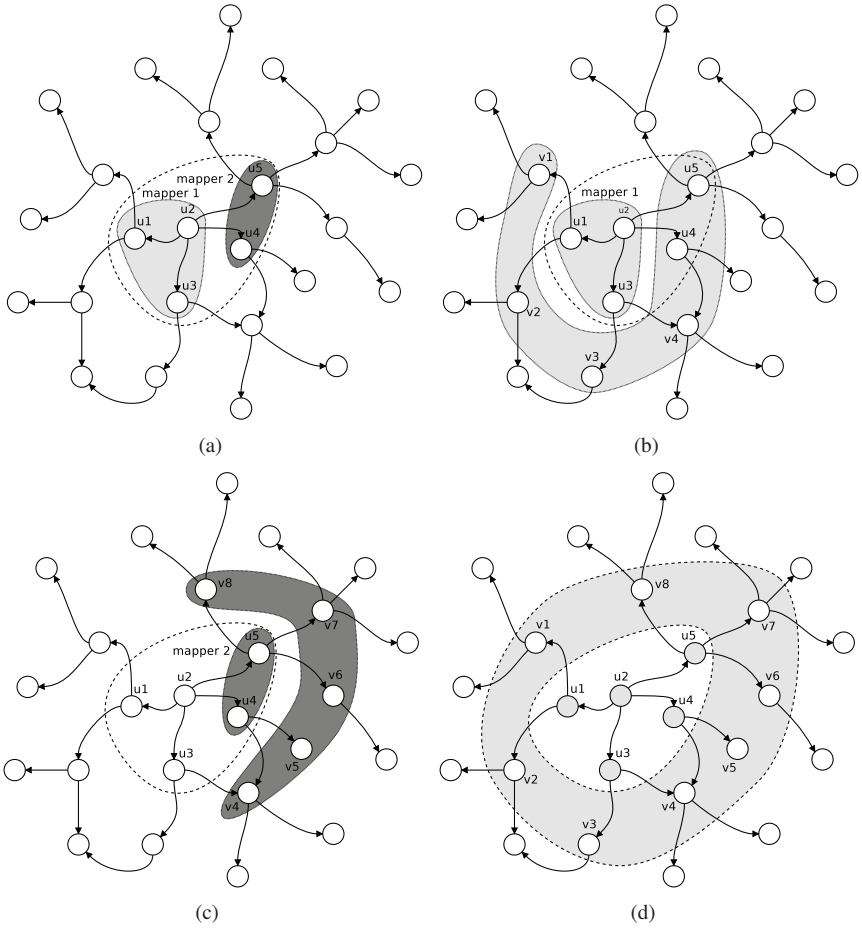
3.1 Crawling engine module

The module responsible for the coordination of parallel crawling tasks exploits the MapReduce programming paradigm, that requires the definition of the *map* and *reduce* phases to model crawling operations. We iterate the map and reduce phases in order to support the Breadth-Search-First approach for data collection within the social network graph, as described in Section 2.

We recall that crawling is an iterative approach where at each step a set of *border users* U is explored for a twofold reason. First, data about the users are retrieved and added to the set of collected information about the social network. Second, all the outgoing social links of the users are scanned to identify the new set of users that will be explored in the next crawling step. Both tasks are implemented using the second module of the crawler that is specific for each social network.

The map phase algorithm starts from the set U that includes the users who are to be explored during the current iteration of the MapReduce algorithm. Figure 2(a) shows an example where the map phase is carried out by two mapper processes: mapper 1 gathers information about the set of users $\{u_1, u_2, u_3\}$, while mapper 2 crawls the users $\{u_4, u_5\}$: the distribution of the users in the original set U is typically carried out by evenly splitting the original input file into chunks. During the map phase, each user u is explored and data are collected about the user and his/her links. A new set V contains the users that are connected to the user u through outgoing social links. In the example, the mapper 1 returns as output the set $\{u_4, u_5, v_1, \ldots v_4\}$ (Figure 2(b)), while the mapper 2 returns the set $\{v_4, \ldots, v_8\}$ (Figure 2(c)). The output of the mapper process is in the format of couples key-value, where the key is the ID of a new user.

The output of the mapper contains duplicate values (in our example v_4) and values that do not belong to the final output. In our example, u_4 and u_5 belong to the initial set of users U and should be filtered out. During the *reduce* phase, duplicated and previously visited users are removed from the list of users that will be explored in the next crawling iteration, as shown in Figure 2(d). The duplicated users are automatically removed by the MapReduce framework as all the key-value pairs with the same key in the output of the map phase are aggregated and sent to the same reducer instance. The task that is carried out by the reduce phase is removed from the output the nodes that were visited in the previous crawling iterations.



3.2 Data collection module

The crawling coordination engine interacts with the data collection module for the acquisition of data from the social network. This latter module contains all the network specific functions of the crawler and provides a standard interface to the mapper. This modular approach guarantees a general purpose crawler that can be easily customized to cope with different social networks.

Figure 2 outlines the data collection module that is activated when the generic i^th mapper invokes the module functions to collect data about a user u. The data collection logic is provided by the *data manager* that can gather information from the social network using two subsystems. First, if the social network provides APIs to third parties, the data manager can use the *API interface* for data collection because this action is much more efficient than Web page parsing. If APIs are not supported or if data obtained through APIs are incomplete, the *Web parser* is activated. It requests the personal Web page(s) of the user u and parses their content. All gathered information is sent to a data storage that in our implementation is a DMBS. Then, the list of social links for the user u is sent back to the coordination engine for the successive phases of crawling.

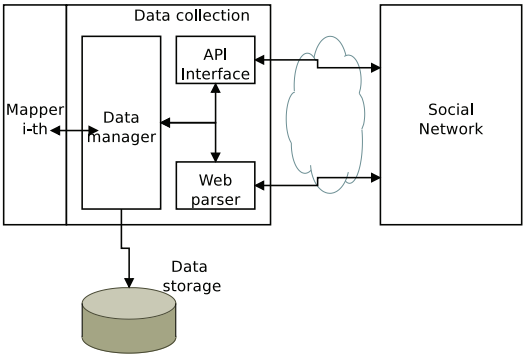


FIG. 2 – *Crawling iterative process*

3.3 Advantages of cloud-based crawling

The proposed cloud-based crawler can address the main issues for large scale data collection in social networks for several reasons.

Using a large number of virtual machines increases data collection speed because the crawling process is distributed over multiple nodes. As data acquisition is a network-bound operation (as it involves only a limited amount of CPU and I/O resources), virtualization provides a further benefit supporting a large number of separated virtual machines, each with its own public IP, on a smaller set of computers. The final effect is a reduction of the crawling time with respect to traditional crawlers, or the crawling of a significantly larger part of the network in the same amount of time. Moreover, since each crawling process runs from a different IP, the risk of IP banning is limited. Furthermore, virtualization allows the fast restart of virtual machine from previously-saved images, and the restart of a virtual machine may be exploited as a way to refresh the pool of IP addresses used for crawling.

We recall that the BSF approach for crawling (described in Section 2.3) may be subject to bias in the collected data when only a small portion of the social network is visited by the crawler. Cloud-based crawling can address this issue because it allows us to collect a large amount of information in a short time period.

It is also important to observe that the crawler modular architecture simplifies the adaptation of the crawler to support data acquisition from multiple social networks because it separates the functions for the coordination of parallel processes (common to any social network) from the interface with the social network specific APIs, including parsing of user home pages, that are the only part of the crawler that needs to be customized for every social network.

4 Related work

Three main techniques have been exploited in literature for data acquisition from social networks: network traffic sniffing and analysis, ad-hoc applications, and crawling the social network graph.

Capturing network data traffic has been exploited by the academic community to collect information about social network user behavior through the analysis of user request-response pat-

terns [Gill et al. (2007); Nazir et al. (2009)]. This technique is typically limited to restricted scenarios such as university campuses and company networks.

Implementing an ad-hoc application for social network analyses has been proposed by Nazir *et al.* [Nazir et al. (2008)] that integrated three novel applications in Facebook to gather the dataset of the subscribed users. This is an interesting and original idea that requires several implementation efforts and that does not guarantee return of investments if the proposed application does not become popular. In any case, the gathered data refer only to the registered users with the high risk of limited analyses and biased results.

In this paper, we focus on crawling that is the most flexible technique to gather data from social networks as testified by many studies [Mislove et al. (2007); Cha et al. (2008); Lerman (2007); Cha et al. (2009)]. We have seen that traditional crawling, although popular, arises many issues, such as the limitations on the number of requests allowed in a specific time period imposed by the social network operators. Many popular social networks offer public APIs that simplify the extraction of data. However, when the data retrieved from APIs are not sufficient to explore the social network structure (for example, because the returned data set is truncated), specific software that gets and parses the user Web pages is required. Due to these and other difficulties that hinder the implementation of a general purpose crawler suitable to multiple social networks, existing studies have analyzed individual or small collections of sites, such as Flickr [Cha et al. (2008); Lerman (2007); Cha et al. (2009)]. The results are limited to detailed views of one popular social network, or provide some comparison between sites that are very similar. A significant exception is the paper [Mislove et al. (2007)], that considers several social networks. However, the study is limited only to the static properties of the network graph, for which data collection is very simple.

We propose to rely on the cloud computing paradigm to build a crawler that allows extensive and complex data gathering and that can be easily customized thanks to its modular structure consisting of a common engine for any social network and a software part specific for each site.

Other tools, such as Nutch [Bialecki (2009)], aiming to exploit parallel data collection and the MapReduce paradigm were proposed initially for Web crawling. We should consider that these tools are designed to navigate through Web pages and hyperlinks and are not designed to explore the graph of social links. Furthermore, these tools cannot take advantage of the social networks APIs to access information about the users, but they are specifically implemented to parse and process Web pages.

A related idea to our proposal has exploited multiple geographically distributed nodes of the Planet-lab network to collect data from a popular social network [Nazir et al. (2009)]. However, the proposed solution as been applied only to one social network and is not designed to cope with different networks with incompatible APIs. Furthermore, the Planet-lab architecture does not provide a flexible support to support the coordination of parallel crawlers, that must be implemented without the facilities provided by the MapReduce approach proposed in our paper.

5 Conclusions

Social networks represent a novel and valuable source of information. In this paper we consider data acquisition techniques and we leave to future work all issues (e.g., legal, privacy) and solutions (e.g., anonymization, sampling) related to the analysis of the sources.

Literature presents three main techniques for data acquisition in social network: network traffic analysis, ad-hoc applications and crawling. We analyze the main features and the limits of each technique with a special focus on crawling that represents the main interest of this paper. In partic-

ular, we evidence that traditional crawling is limited by the large size and complexity of the social network structure that requires excessive times to collect a significant portion of the available data. Moreover, the social network operators tend to use countermeasures to limit excessive crawling from the same IP sources. Finally, each social network provides its own APIs and is characterized by peculiar Web pages and layouts, hence accessing data requires the implementation of a specific crawler for each site.

We propose a novel approach that exploits the MapReduce programming paradigm and the cloud computing platform for a new generation of social network crawlers. Our proposal allows us to design a modular and easily customizable crawler that relies on a large number of distributed nodes to speedup the data collection process and to address the other issues of traditional crawling over social networks.

Acknowledgements

The authors acknowledge the support of FP7–SEC–2009-1 project VIRTUOSO "Versatile InfoRmation Toolkit for end-Users oriented Open-Sources explOitation", Grant Agreement nr. 242352.

References

Ahn, Y.-Y., S. Han, H. Kwak, S. Moon, and H. Jeong (2007). Analysis of topological characteristics of huge online social networking services. In *Proc. of the 16th International Conference on World Wide Web (WWW'07)*, Banff, Alberta, Canada.

Amazon Elastic Compute Cloud (2010). http://aws.amazon.com/ec2.

Andreolini, M., S. Casolari, M. Colajanni, and M. Marchetti (2007). Dynamic load balancing for network intrusion detection systems based on distributed architectures. In *Proc. of 6th IEEE International Symposium on Network Computing and Applications (NCA'07)*, Boston, MA, USA.

Bialecki, A. (2009). Web-scale search engine toolkit. In *Proc. Of Apache Con 2009*.

Canali, C., S. Casolari, and R. Lancellotti (2010). A quantitative methodology to identify relevant users in social networks. In *Proc. of the IEEE International Workshop on Business Applications of Social Network Analysis (BASNA'10)*, Bangalore, India.

Canali, C., M. Colajanni, and R. Lancellotti (2009). Performance Evolution of Mobile Web-Based Services. *IEEE Internet Computing 13*(2), 60 – 68.

Cha, M., A. Mislove, B. Adams, and K. P. Gummadi (2008). Characterizing social cascades in Flickr. In *Proc. of the 1st Workshop on Online Social Networks (WOSP'08)*, Seattle, WA, USA.

Cha, M., A. Mislove, and K. P. Gummadi (2009). A measurement-driven analysis of information propagation in the Flickr social network. In *Proc. of the 18th international conference on World Wide Web (WWW'09)*, Madrid, Spain.

comScore comScore (2010). YouTube Streams All-Time High of 14.6 Billion Videos Viewed. comScore Releases May 2010 U.S. Online Video Rankings.

Crovella, M. and B. Krishnamurthy (2006). *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley and Sons, Inc.

Dean, J. and S. Ghemawat (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM 51*(1), 107–113.

Gill, P., M. Arlitt, Z. Li, and A. Mahanti (2007). YouTube traffic characterization: A view from the edge. In *Proc. of Internet Measurement Conference (IMC'07)*, San Diego, CA.

Gjoka, M., M. Kurant, C. T. Butts, and A. Markopoulou (2010). Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *Proc. of the IEEE International Conference on Computer Communications (INFOCOM'10)*, San Diego, CA.

Hadoop MapReduce (2010). http://hadoop.apache.org/mapreduce/.

Khrishanmurthy, B. (2009). A measure of Online Social Networks. In *Proc. of the 1st International Conference on COMmunication Systems and NETworkS (COMSNETS'09)*, Bangalore, India.

Lerman, K. (2007). Social Information Processing in News Aggregation. *IEEE Internet Computing 11*(6), 16–28.

Mislove, A., M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee (2007). Measurement and analysis of online social networks. In *Proc. of the 7th ACM SIGCOMM conference on Internet measurement (IMC'07)*, San Diego, California, USA.

Nazir, A., S. Raza, and C.-N. Chuah (2008). Unveiling Facebook: a Measurement Study of Social Network Based Applications. In *Proc. of the 8th ACM SIGCOMM Conference on Internet Measurement (IMC'08)*, Vouliagmeni, Greece.

Nazir, A., S. Raza, D. Gupta, C.-N. Chuah, and B. Krishnamurthy (2009). Network level footprints of Facebook applications. In *Proc. of the 9th ACM SIGCOMM conference on Internet measurement conference (IMC'09)*, Chicago, Illinois, USA.

NielsenWire (2009). Social Networking's New Global Footpront. Nielsen Online Report.

Schonfeld, E. (2008). Digg Nearly Triples Registered Users In a Year, Says Sleuth Programmer. Research Report TechCrunch, 2008.

Wilson, C., B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao (2009). User interactions in social networks and their implications. In *Proc. of the 4th ACM European conference on Computer systems (EuroSys'09)*, Nuremberg, Germany.

Youtube Developer's Guide (2010). Data API Protocol – API Query Parameters. http://code.google.com/apis/youtube/2.0/developers_guide_protocol_api_query_parameters.html.

Extraction de connaissances pour le renseignement en sources ouvertes

Laurie Serrano*,**, Bruno Grilheres*, Maroua Bouzid**, Thierry Charnois**

*IPCC, CASSIDIAN
Parc d'Affaires des Portes, 27600 Val de Reuil
**GREYC, Université de Caen
Campus Côte de Nacre, Boulevard du Maréchal Juin, BP 5186 - 14032 Caen

Résumé. Cet article présente un outil d'extraction de l'information pour le renseignement sources ouvertes. Nous détaillons tout d'abord notre modélisation d'une ontologie de domaine destinée à structurer et partager les informations extraites. Puis, nous décrivons notre approche basée sur des techniques linguistiques visant à détecter les entités nommées, évènements et relations d'intérêt. L'implémentation de notre méthode grâce à la plateforme GATE ainsi qu'une évaluation des premiers résultats sont ensuite proposées. Nous concluons cet article en exposant nos perspectives de recherche dans le cadre d'une problématique plus large de capitalisation des connaissances.

1 Introduction

Aujourd'hui, l'abondance des informations accessibles publiquement (dites « sources ouvertes ») a fait émerger le besoin de « fouiller » cette masse de documents afin de repérer, structurer et partager les informations pertinentes et utiles dans un but donné. Les récents efforts en extraction d'information ont donné naissance à des techniques et outils permettant le repérage de ces informations d'intérêt (généralement les entités nommées, relations entre entités et évènements) (Poibeau, 2003). L'élaboration d'un tel système nécessite au préalable de définir la nature de ces informations afin de les partager avec d'autres services de traitement de l'information : l'ontologie de domaine est, à l'heure actuelle, le mode de représentation le plus utilisé dans ce but. Vient ensuite la phase centrale d'analyse textuelle visant à détecter et extraire ces types d'information. Nous pouvons ici distinguer plusieurs approches, parmi lesquelles émergent les techniques linguistiques d'une part et les systèmes statistiques à base d'apprentissage d'autre part. Les informations extraites sont ensuite destinées à peupler l'ontologie de domaine guidant l'extraction.

Nous avons développé un système d'extraction d'information pour l'anglais et le français fondé sur des techniques linguistiques. Nous proposons une extraction d'entités nommées basée sur la construction de grammaires et une approche innovante pour extraire les évènements et relations. Dans un premier temps, nous présentons une synthèse de nos travaux de modélisation et d'extraction de l'information, puis, une évaluation des premiers résultats obtenus. Pour finir, cet article détaille les perspectives de recherche que nous envisageons dans

le domaine de l'extraction d'information et plus généralement de la capitalisation des connaissances.

2 Vers une ontologie du renseignement

La construction de notre système d'extraction a nécessité, au préalable, de définir l'étendue et la nature des informations d'intérêt dans le domaine du renseignement sources ouvertes : c'est-à-dire mettre en place un modèle de connaissances. Nous avons choisi, pour cela, de développer une ontologie de domaine qui servira de guide aux différentes étapes d'extraction. Compte-tenu de la diversité des documents exploités dans le cadre du renseignement sources ouvertes, cette ontologie devra rester assez générale tout en définissant plus amplement les concepts et propriétés relatifs au domaine militaire.

2.1 Travaux existants

Suivant cet objectif, nous avons mené quelques recherches pour faire le point sur les ontologies existantes. En effet, il nous est apparu intéressant de pouvoir éventuellement reprendre tout ou partie d'une modélisation déjà disponible. Nous avons pour cela observé des ontologies générales, dites « de haut niveau » mais également des ontologies du domaine du renseignement.

Nous avons commencé par examiner les ontologies générales les plus connues et utilisées telles que SUMO (Niles et Pease, 2001), PROTON (Terziev et al., 2005), COSMO [1], OpenCyc [2], BFO (Spear, 2006) ou encore DOLCE (Gangemi et al., 2002). Dans un second temps, nous nous sommes intéressés aux ontologies disponibles pour le renseignement. Tout d'abord, nous avons étudié les recommandations de plusieurs standards OTAN. Ceux-ci sont des accords de normalisation ratifiés par les pays de l'alliance définissant des normes pour permettre les interactions entre les différentes armées. Nous avons accordé une attention particulière aux catégories de l'intelligence définies par le STANAG 2433 (NATO, 2005) mais aussi au STANAG 5525 (NATO, 2007). Ces standards restent trop généralistes et techniques mais nous avons pu tout de même nous inspirer des classes principales (« pentagramme du renseignement ») et de certaines propriétés. Enfin, les ontologies « swint-terrorism » (Mannes et Golbeck, 2005), reprenant les concepts principaux nécessaires au domaine du terrorisme, et AKTiveSA (Smart et al., 2007), dédiée à la description des contextes opérationnels militaires autres que la guerre, ont constitué d'autres exemples de modélisation.

Ces différentes modélisations ne correspondant pas exactement au modèle de connaissances défini précédemment nous avons fait le choix de définir notre propre ontologie en nous basant sur nos observations préalables.

2.2 Une proposition de modélisation

Nous avons fait le choix de baser notre ontologie sur 5 concepts correspondant aux éléments centraux du domaine du renseignement : *Units*, *Equipment*, *Places*, *Biographics*, *Events*.

1. http ://micra.com/COSMO/
2. http ://www.opencyc.org/

Une fois cette taxonomie de base définie, il nous a été nécessaire d'affiner notre représentation, guidés par le contexte du renseignement. Celle-ci comporte à l'heure actuelle environ 60 classes et 50 propriétés de classes. Par ailleurs, sa profondeur moyenne est de 3 niveaux et sa profondeur maximale s'élève à 4 niveaux. Nous avons réservé plus de temps à la modélisation de la classe «Event» (43% des classes et 41% des propriétés de notre ontologie) compte tenu de l'importance de ces entités pour le renseignement et la veille.

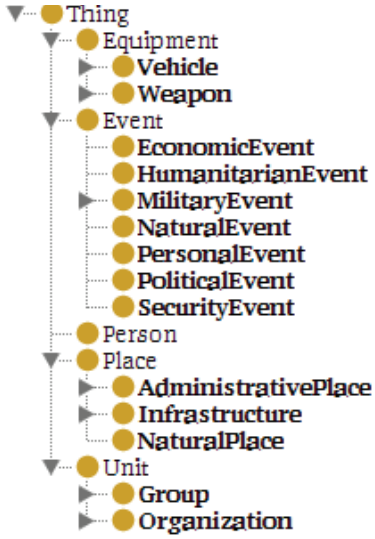


FIG. 1 – *Extrait de notre ontologie du renseignement*

Pour prolonger ces travaux de modélisation, il sera préférable d'interagir avec des opérationnels du renseignement sources ouvertes pour comprendre plus précisément leurs besoins et compléter notre ontologie en conséquence.

3 Extraction d'information pour le renseignement

3.1 Approches existantes

L'on distingue généralement deux types d'approche en extraction d'information : une extraction basée sur des techniques linguistiques d'un côté et des systèmes statistiques à base d'apprentissage de l'autre. Celles-ci se basent, de façon commune, sur des pré-traitements linguistiques «classiques» («tokenization», lemmatisation, analyse morpho-syntaxique).

La première approche exploite les avancées en TAL [3] et repose principalement sur l'utilisation de grammaires formelles construites par la main d'un expert-linguiste. Les pré-traitements cités plus haut servent de base à la construction de règles et patrons linguistiques qui définissent les contextes d'apparition de telle entité ou relation. La seconde approche utilise des techniques

3. Traitement Automatique de la Langue

statistiques pour « apprendre » des régularités sur de larges corpus de textes où les entités-cibles ont été préalablement annotées. Ces méthodes d'apprentissage ou « machine learning » sont plus ou moins supervisées et exploitent des caractéristiques textuelles plus ou moins linguistiques issues des pré-traitements précédemment évoqués.

3.2 Extraction des informations d'intérêt

L'outil que nous présentons ici a été réalisé selon une approche linguistique grâce à la plateforme d'ingénierie textuelle GATE (Cunningham et al., 2002) et vise à extraire les entités nommées, les évènements et les relations. Notre système repose sur le principe des chaines de traitements inhérent à GATE.

3.2.1 Extraction d'entités nommées

Nous avons, tout d'abord, développé une chaine d'analyse de textes anglais et français, permettant le repérage et l'extraction d'entités nommées de type « personne », « organisation », « lieu » et « date ». La plateforme GATE proposant déjà des chaines d'extraction pour l'anglais (ANNIE) et le français, nous avons fait le choix de les réutiliser en y apportant quelques modifications pour améliorer leurs performances et les adapter à notre représentation des connaissances. Pour cela, nous avons fait le choix de privilégier la précision par rapport au rappel. En effet, il nous est apparu plus important dans le contexte de nos travaux d'éviter l'extraction d'informations erronées. Concrètement, cela se traduit par la construction de règles linguistiques dont les résultats sont plus sûrs et la mise à l'écart de règles pouvant entrainer de fausses annotations. De plus, nous avons choisi de réorganiser les différentes phases d'extraction afin de parer à d'éventuelles ambiguïtés entre entités. La réutilisation de la chaine dédiée au français a nécessité de créer nos propres lexiques et règles linguistiques : une simple traduction du système anglais n'aurait pas suffi de par les nombreuses différences syntaxiques et typographiques entre ces deux langues.

3.2.2 Extraction d'évènements

Élément essentiel dans le cadre de la veille stratégique et du renseignement, un évènement peut être défini de façon générale comme une action (un « process ») reliée à un ou plusieurs participants ou circonstants. Avant tout essai d'implémentation, nous avons réfléchi à une nouvelle approche d'extraction d'évènements qui soit la plus générale possible et ceci à différents niveaux. Celle-ci devra, tout d'abord, être applicable à l'analyse de textes en plusieurs langues (français et anglais) et plusieurs domaines, mais aussi à différentes plateformes et environnements de traitement de la langue. De plus, notre méthodologie se voudra adaptée à l'utilisation de plusieurs analyseurs syntaxiques.

Tout d'abord, nous définissons un ensemble de termes considérés comme possibles déclencheurs d'évènement. Ces déclencheurs sont répartis en différentes listes, chacune étant associée à un type d'évènement, autrement dit à une classe d'évènement de notre ontologie. Par la suite, on repère et annote (en précisant la classe de l'ontologie associée) les termes déclencheurs présents dans le texte à analyser. Il nous faut ensuite leur associer les différents participants impliqués dans l'évènement qu'ils déclenchent. Pour cela nous devons repérer les relations entre le déclencheur et d'autres entités de la phrase. Il nous paraît alors judicieux

d'utiliser un analyseur syntaxique donnant les dépendances entre les différents éléments de la phrase. Par observation des sorties de différents analyseurs, nous avons pu établir une méthode générique de détection des participants. La plupart des analyseurs syntaxiques représente une relation de dépendance comme un lien entre la « tête » du syntagme-recteur et la « tête » du syntagme-dépendant. Ainsi, les participants de l'évènement correspondent aux syntagmes dépendants du mot déclencheur. Ceux-ci peuvent être extraits par une analyse en constituants délimitant les différents syntagmes d'une phrase : syntagmes nominaux, verbaux, prépositionnels ou adjectivaux. Une fois ces syntagmes rattachés à l'élément déclencheur par les relations de dépendance, il nous faut leur attribuer un rôle sémantique. Il nous faut, pour cela, réaliser une étude de la structure argumentale des termes déclencheurs : à savoir les rôles sémantiques de leurs différents actants. Pour cela, nous avons choisi de définir des classes argumentales, chacune d'elles correspondant à un type de construction syntaxique. Enfin, pour une meilleure extraction nous devrons prendre en compte d'autres paramètres tels que la voix (passive ou active) du syntagme verbal, la polarité de la phrase (négative ou positive), la modalité mais aussi les phénomènes de valence multiple.

Une fois cette méthodologie établie, nous avons développé une chaine d'extraction d'évènements dans des textes anglais (cf. Fig 2). Notre outil permet, à l'heure actuelle, de repérer les différents types d'évènements définis dans notre ontologie de domaine. Pour chaque événement détecté, nous avons pour objectif d'extraire, s'ils sont présents, les participants / circonstants suivants : la date, le lieu, l'agent et le patient. Nous avons, pour commencer, listé pour chaque type d'évènement un ensemble des termes susceptibles de le réaliser dans un texte. Nous choisissons de nous limiter, pour l'instant, aux déclencheurs verbaux et nominaux et de constituer des listes de lemmes, plus courtes et permettant d'étendre le repérage à toutes les formes fléchies. De plus, pour pouvoir déterminer les rôles sémantiques de chaque argument de l'évènement, nous avons associé à chaque lemme verbal une indication sur sa structure argumentale. Une fois les déclencheurs d'évènements repérés, nous procédons à une analyse syntaxique (grâce au Stanford parser) qui détermine l'ensemble des relations de dépendances. Par la suite, un transducteur JAPE exécute l'ensemble des règles linguistiques que nous avons développées pour extraire les différents arguments de l'évènement et leur attribuer un rôle sémantique.

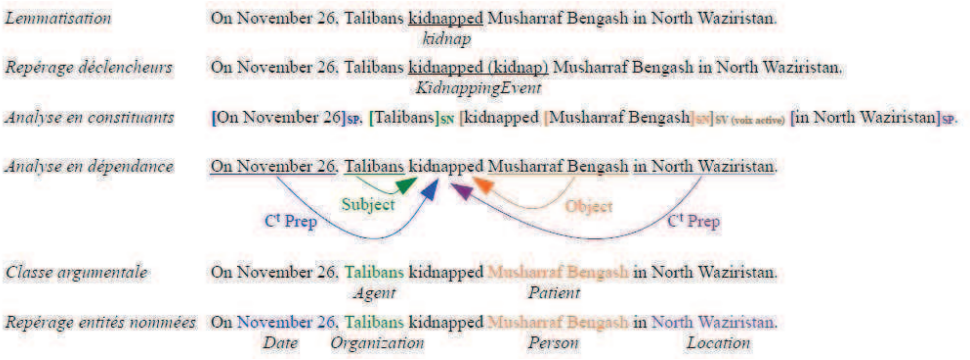


FIG. 2 – *Chaine d'extraction d'évènements pour l'anglais*

A l'heure actuelle, nous obtenons une annotation de type «Event» positionnée sur le déclencheur de l'évènement. Cette annotation présente les différents acteurs de l'évènement ainsi que leur rôle et indique pour chacun d'eux s'il correspond à une entité nommée détectée précédemment.

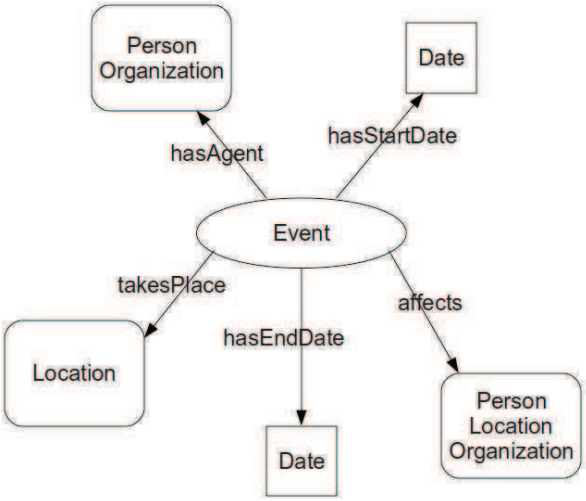


FIG. 3 – *Schéma de l'évènement et ses participants/circonstants*

3.2.3 Extraction de relations

Pour finir, nous nous sommes intéressés à la détection de relations entre entités nommées. Notre méthode consiste à définir, par observation de corpus du domaine visé, un ensemble de mots déclencheurs pour chaque type de relation de l'ontologie. Ceux-ci sont ensuite comparés aux mots du texte à traiter qui, s'il y a correspondance, sont annotés en tant que possibles déclencheurs de relation. Dans un deuxième temps, un ensemble de règles linguistiques teste le contexte de chaque déclencheur pour déterminer la présence d'une relation du type souhaité. Lorsque le contexte concorde, une annotation, dont le type est donné par l'élément déclencheur, est créée sur l'ensemble des éléments de la relation (arguments et lien). Un premier prototype a été réalisé pour l'extraction de relations entre personnes et organisations (appartenance/direction, parenté) dans des textes anglais.

3.3 Premiers résultats

Pour estimer l'efficacité de notre outil, nous avons mené deux types d'évaluation : une évaluation chiffrée de l'extraction d'entités nommées et une évaluation qualitative des deux autres extractions.

Évaluer notre extraction d'entités nommées en anglais et en français a, tout d'abord, nécessité de faire plusieurs choix concernant le protocole à mettre en place. Deux solutions se

sont alors présentées : réutiliser les données d'une campagne d'évaluation existante ou créer notre propre système d'évaluation. Le premier cas impliquait de trouver un corpus du domaine, où les entités nommées ont été préalablement annotées et qui soit accompagné de scripts de « scoring ». Nous avons, pour cela, examiné les données librement diffusées de campagnes d'évaluation telles que MUC (Grishman et Sundheim, 1996), ACE (Doddington et al., 2004) ou ESTER (Gravier et al., 2004) mais n'avons pas trouvé de données répondant à notre besoin. En conséquence, nous avons opté pour la deuxième solution, c'est-à-dire développer notre système d'évaluation. Pour cela, nous avons choisi deux corpus d'évaluation ayant pour thématique le renseignement : le corpus AQUAINT[4] pour l'anglais (une centaine de textes) et un corpus français de même taille composé de dépêches AFP sur le thème de l'Afghanistan. Une fois ces corpus annotés par notre système, les résultats de l'extraction ont été évalués manuellement par quatre spécialistes du « media mining ». Le tableau suivant présente nos résultats.

	Anglais			**Français**		
	Précision	**Rappel**	**F-mesure**	**Précision**	**Rappel**	**F-mesure**
Dates	0,99	0,73	0,84	1,00	0,70	0,83
Lieux	0,98	0,93	0,95	0,99	0,91	0,95
Organisations	0,86	0,73	0,79	0,98	0,78	0,87
Personnes	0,98	0,86	0,92	0,96	0,94	0,95
Entités	0,96	0,86	0,91	0,98	0,85	0,91

TAB. 1 – *Évaluation de l'extraction d'entités nommées*

Tout d'abord, au vu de la F-mesure globale (« Entités »), notre système d'extraction obtient de bons résultats en anglais et en français, la majorité des outils d'extraction d'entités nommées atteignant 90% de F-mesure sur des tâches similaires (Marrero et al., 2009). Nous pouvons dire que notre extraction d'entités nommées atteint l'état de l'art sans oublier pour autant les variations dues aux conditions d'évaluation (corpus, métriques, accord inter-annotateurs, etc.). Par ailleurs, nos résultats s'avèrent globalement équivalents dans les deux langues et, de par notre objectif de départ, notre outil montre une précision supérieure au rappel pour tous les types d'entités. Toutefois, notre système présente encore quelques faiblesses comme en témoignent les scores pour les entités de type « date » et « organisation ».

Notre extraction d'évènements se distingue par une approche originale sur plusieurs points. Tout d'abord, l'utilisation d'un lemmatiseur permet de repérer un évènement ou une relation au travers de ces différentes réalisations linguistiques tout en limitant la taille des lexiques. Par ailleurs, l'analyse syntaxique contribue de façon double à une meilleure extraction des participants : les constituants, d'une part, aident à la délimitation et les dépendances, d'autre part, servent pour le repérage « à distance » et l'attribution des rôles sémantiques. Relevons maintenant les différents points sur lesquels compléter nos extractions d'évènements et de relations. La limite principale concernant les évènements reste l'absence d'implémentation pour le traitement de textes en français. En effet, le manque d'analyseur syntaxique open source et gratuit pour cette langue a constitué un réel obstacle au développement de notre outil. Toutefois, même si un analyseur performant améliore grandement l'extraction d'évènements, notre système peut être amélioré par d'autres moyens. Tout d'abord, il sera nécessaire d'exploiter

4. http ://www-nlpir.nist.gov/projects/aquaint/

toutes les dépendances données par l'analyseur syntaxique pour mieux détecter tous les participants de l'évènement. Il faudra également étudier la construction des déclencheurs nominaux afin de déterminer des classes sur le même principe que pour les verbes. Pour finir, notre outil peut être raffiné en prenant en compte la négation, la modalité et les compléments prépositionnels dans la détection des évènements.

Enfin, notre extraction de relations devra être améliorée sur plusieurs points et implémentée pour les deux langues en question. Nous pourrons pour cela étudier les travaux de Nakamura-Delloye et Villemonte De La Clergerie (2010) proposant de repérer les chemins syntaxiques entre deux entités afin de construire par généralisation un ensemble de patrons de relations syntaxiques spécifique à tel type de relation sémantique.

4 Perspectives : Vers une capitalisation des connaissances orientée utilisateur

Ces premiers travaux nous ont permis de faire le point sur l'efficacité de notre méthode d'extraction et de mettre à jour d'autres problématiques à étudier. Nous souhaitons poursuivre nos réflexions pour tenter de répondre au constat suivant : s'il existe de nombreux systèmes d'extraction d'information, ceux-ci s'avèrent souvent peu fiables et ne prennent pas suffisamment en compte la dimension « connaissance », point essentiel dans ce domaine. Nos recherches viseront donc à définir une approche de capitalisation des connaissances permettant de faciliter et de réduire le travail des opérationnels dans le cadre de la veille économique et stratégique et du renseignement. Pour cela, notre objectif est d'explorer les méthodes d'extraction et de structuration automatique des informations accessibles en sources ouvertes. Les travaux que nous envisageons s'articulent autour de deux axes : « extraction d'information » et « capitalisation des connaissances ».

Au sein de l'axe « extraction d'information », nos recherches viseront à élaborer une approche aussi performante que possible et bien adaptée à nos objectifs. Dans ce but, nous privilégierons une combinaison des meilleures approches actuelles : les observations faites jusqu'à présent nous orientent d'ores et déjà vers le choix d'une approche hybride, combinant des techniques d'extraction à la fois linguistiques et statistiques. Il nous faudra, par ailleurs, tenter de dépasser certaines limites caractéristiques des systèmes d'extraction actuels telles que leur dépendance aux domaine/genre/langue des textes traités ou encore leur coût de développement (construction manuelle de règles/corpus annotés). Un autre objectif sera d'affiner les techniques d'évaluation existantes pour représenter la précision et la confiance à accorder aux informations extraites. Pour cela, nous tenterons de proposer des techniques d'auto-évaluation, prenant en compte l'incertitude ou l'imprécision des règles d'extraction.

L'axe « capitalisation des connaissances » aura pour objectif une structuration automatique des informations, leur capitalisation en base de connaissances mais également leur mise à jour et leur exploitation par des méthodes de raisonnements. Dans un premier temps, il s'agira de structurer les informations extraites en fiches de connaissances ; où une connaissance serait définie non pas comme le contenu d'un document mais bien comme un ensemble organisé d'informations collectées sur plusieurs textes. De plus, afin d'arriver à une réelle structuration et non plus à une simple addition de l'information, nous devrons prendre en compte les

problématiques de continuité de l'information (redondance et contradiction) mais également la temporalité et la modalité discursive exprimée au sein du document.

5 Conclusion

Nous venons de présenter un système d'extraction de l'information pour le renseignement sources ouvertes fondé sur une ontologie de domaine. L'évaluation reportée nous a permis de percevoir plus globalement la qualité de notre extraction mais aussi de mettre en avant certaines limites qui devront être dépassées dans le futur. Notre approche à base de règles contextuelles atteint l'état de l'art pour l'extraction d'entités nommées en anglais et français. Notre méthode d'extraction d'évènements montre le caractère indispensable d'une analyse syntaxique dans le repérage de telles informations. Celle-ci nécessite un analyseur fiable et robuste et nous avons pu constater que de nombreux efforts sont encore à mener dans ce sens en termes de performances et de langues traitées. Par ailleurs, les limites des systèmes linguistiques et statistiques actuels nous orientent vers une future combinaison de ces approches pour une meilleure extraction. Enfin, il s'agira, pour aller plus loin, de donner à l'utilisateur les moyens de tirer partie de ces informations pour acquérir de nouvelles connaissances.

Références

Cunningham, H., D. Maynard, K. Bontcheva, et V. Tablan (2002). Gate : A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA*.

Doddington, G., A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, et R. Weischedel (2004). The Automatic Content Extraction (ACE) program - Tasks, Data, and Evaluation. *Proceedings of LREC 2004*, 837–840.

Gangemi, A., N. Guarino, C. Masolo, A. Oltramari, R. Oltramari, et L. Schneider (2002). Sweetening ontologies with dolce. pp. 166–181. Springer.

Gravier, G., J.-F. Bonastre, E. Geoffrois, S. Galliano, K. M. Tait, et K. Choukri (2004). Ester, une campagne d'évaluation des systèmes d'indexation automatique d'émissions radiophoniques en français.

Grishman, R. et B. Sundheim (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, Morristown, NJ, USA, pp. 466–471. Association for Computational Linguistics.

Mannes, A. et J. Golbeck (2005). Building a terrorism ontology. In *ISWC Workshop on Ontology Patterns for the Semantic Web*.

Marrero, M., S. Sanchez-Cuadrado, J. M. Lara, et G. Andreadakis (2009). Evaluation of named entity extraction systems. *Advances in Computational Linguistics. Research in Computing Science 41*, 47–58.

Nakamura-Delloye, Y. et E. Villemonte De La Clergerie (2010). Exploitation de résultats d'analyse syntaxique pour extraction semi-supervisée des chemins de relations. In *17e*

Conférence sur le Traitement Automatique des Langues Naturelles - TALN 2010, Montréal Canada.

NATO (2005). The military intelligence data exchange standard - aintp-3(b). Technical report.

NATO (2007). Joint c3 information exchange data model - jc3iedm. Technical report.

Niles, I. et A. Pease (2001). Towards a standard upper ontology. pp. 2–9. ACM Press.

Poibeau, T. (2003). *Extraction automatique d'information : Du texte brut au web sémantique*. Lavoisier.

Smart, P., A. Russell, N. Shadbolt, M. Shraefel, et L. Carr (2007). Aktivesa. *Comput. J. 50*, 703–716.

Spear, A. (2006). Ontology for the twenty first century: An introduction with recommendations. Technical report, The Institute for Formal Ontology and Medical Information Science.

Terziev, I., A. Kiryakov, et D. Mano (2005). Base upper-level ontology (bulo) guidance. Technical report deliverable 1.8.1, SEKT project.

Summary

This paper presents an information extraction tool for «open sources» intelligence. We first detail our domain ontology designed for structuring and sharing the extracted informations. Then, we describe an approach based on linguistic techniques to recognize named entities, events and relationships of interest. A first implementation of our method through the GATE architecture and our early results evaluation are then proposed. Finally, we introduce our planned researches within the knowledge capitalization broader field.

Une chaîne UIMA pour l'analyse de documents de réglementation

Samir Derdek*,**, Adil El Ghali*

*IBM CAS France
9 rue de Verdun, BP 85, 94253 Gentilly, France
adil.elghali@fr.ibm.com

**Institut Galilée - Université Paris 13
99 avenue J.-B. Clément, 93439 Villetaneuse, France
samir.derdek@gmail.com

Résumé. Cet article [1] présente une chaîne de traitement UIMA pour l'analyse des documents de réglementation. Cette chaîne incorpore un analyseur syntaxique et un résolveur d'anaphores, qui produisent des annotations exploitables pour détecter des règles et pour faciliter le processus de modélisation de règles. Cette chaîne est destinée à être utilisée par des experts métier pour améliorer la modélisation d'un système de règles à partir de textes.

1 Introduction

Les documents de réglementation contiennent des règles auxquelles doivent se conformer les entreprises dans les processus de fabrication et de commercialisation de leurs produits. Il est donc vital pour celles-ci d'en tenir compte, voire de les intégrer à leurs processus. Ces processus sont, chez un nombre croissant d'industriels, formalisés et implantés dans des systèmes de gestion de règles métier (BRMS), il apparaît donc d'une grande utilité de pouvoir transformer les règles et les contraintes contenues dans les documents de réglementation en des règles formelles et exécutables dans un système de règles.

L'approche que nous avons retenue consiste à analyser ces documents à l'aide d'outils d'analyse linguistique profonde afin de produire des annotations linguistiques pour les éléments composant le texte. Ces annotations sont ensuite exportées au format RDF permettant, d'une part, leur utilisation pour identifier des règles via des requêtes SPARQL traduisant les principaux schémas que respectent les règles dans les documents ; et d'autre part, la possibilité pour les analystes et les experts métier de naviguer dans les bases de documents réglementaires d'une manière efficace pour leur processus de modélisation et de formalisation de règles.

La chaîne de traitement prend en entrée les sources réglementaires qui sont disponibles publiquement, et utilise d'autres sources ouvertes (vocabulaires, bases de données linguistiques, ...), elle peut par ailleurs tirer profit de toute ontologie du domaine existante. Le cadre dans lequel nous avons testé notre chaîne de traitement est celui du projet ONTORULE [2], en parti-

1. Ce travail a été partiellement financé par le projet FP7 EU-IST Integrated Project 2009-231875 ONTORULE.
2. http://www.ontorule-project.eu

culier pour le scénario Audi où il fallait analyser des documents de réglementation européennes pour le test des composants d'un véhicule, et intégrer les règles provenant de ces documents dans une application métier à base de règles permettant de choisir la meilleure manière de tester des composants tout en respectant la réglementation.

Cet article est organisé comme suit : dans la première section, nos décrivons la chaîne de traitement UIMA et ses différents composants, ensuite nous présentons les structures de données manipulées et les annotations RDF exportées. Enfin, nous donnons quelques exemples d'utilisation de ces annotations..

2 Chaîne de traitement des textes de réglementations

Le but de notre expérimentation est de produire à partir des textes en entrée une représentation qui puisse être utilisée par un module de raisonnement permettant (i) de transformer les informations linguistiques dans le texte en des données utilisables dans des modèles (ontologie ou modèle métier) décrivant le domaine métier, et (ii) d'identifier les passages du texte qui correspondent à des règles métier,

Pour ce faire, notre approche consiste à effectuer une analyse linguistique profonde. Puis à réaliser un module de raisonnement effectuant l'identification et la formalisation des règles. Notre chaîne de traitement est composée de deux modules principaux d'analyse : un analyseur syntaxique en dépendance, et un résolveur d'anaphores. Elle permet de regrouper le résultat de ces analyses dans des annotations homogènes qui seront exploitables par le module d'extraction de règles ou par le module d'aide à la modélisation.

De plus, notre approche se décline en deux modes de traitements. Le premier consiste à sélectionner le meilleur résultat de l'analyse linguistique et à le soumettre à son exploitation par les modules d'aide à la modélisation et d'extraction de règles. Le second consiste à conserver tous les résultats de l'analyse linguistique et à les soumettre aux modules, afin de laisser un choix de règles plus important à l'expert métier. Quelque soit le mode choisi, la validation de la pertinence des règles est faite par un expert métier.

2.1 Architecture

Nous avons choisi *Unstructured Information Management Application* [3] (UIMA) Ferrucci et Lally (2004); Hernandez et al. (2010, 2009) pour encoder notre chaîne de traitement car ce framework permet de définir facilement une chaîne de traitement alliant flexibilité (UIMA s'appuie sur une architecture pilotée par les données où les modules de traitements peuvent être facilement remplacés par d'autres) et qualité (les chaînes produites permettent d'analyser un gros volume de données). De plus, il offre un environnement de développement puissant, incluant entre autres des interfaces de visualisation des résultats à chaque étape de l'analyse.

L'architecture de la chaîne de traitement est présentée en figure 1. Les différents modules (analyseur syntaxique et résolveur d'anaphores) produisent des informations linguistiques (morphologiques, syntaxiques, ...) qui sont traduites en annotations UIMA. Ces annotations peuvent par la suite être exportées en RDF.

3. http://uima.apache.org/index.html

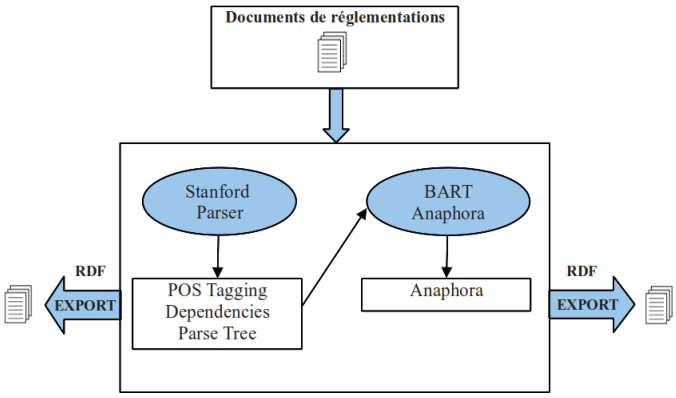


FIG. 1 – *Architecture de la chaîne UIMA.*

Dans l'implantation actuelle les modules utilisés sont le *Stanford Parser* et *BART Anaphora*, mais dans cette chaîne de traitement ceux-ci peuvent aisément être remplacés par d'autres modules fournissant des sorties équivalentes.

2.2 Les composants de la chaîne de traitements

2.2.1 *Stanford Parser*

L'analyseur *Stanford Parser* Klein et Manning (2003a,b) est un analyseur statistique en dépendance qui offre un bon compromis entre rapidité et pertinence Cer et al. (2010), il permet d'extraire trois types d'informations :

– les **parties du discours** pour les mots de la phrase, utilisant le jeu d'annotation du *Penn Treebank* Marcus et al. (1993), l'exemple ci-dessous donne un aperçu de l'étiquetage morpho-syntaxique de l'analyseur :

```
A/DT complete/JJ safety-belt/JJ assembly/NN shall/MD be/VB positioned/VBN ...
```

– l'**arbre syntaxique** en constituants permettant de faire les regroupements grammaticaux, ci-après un exemple :

```
(ROOT
  (S
    (NP (DT A) (JJ complete) (JJ safety-belt) (NN assembly))
    (VP (MD shall)
      (VP (VB be)
        (VP (VBN positioned)
          (PP (IN in)
            (NP (DT a) (NN test) (NN chamber)))
          (SBAR (IN as)
            (S
              (VP (VBN prescribed)
                (PP (IN in)
                  (NP (NNP Annex) (CD 12)))))))))))
```

– les **dépendances** *i.e.* des relations syntactico-sémantiques entre les mots d'une phrase, il utilise le jeu de dépendances typées de Marneffe et Manning (2008), permettant de

regrouper des dépendances élémentaires, comme dans l'exemple ci-dessous :

```
prep(prescribed-13, in-14)
                              →  prep_in(prescribed-13, Annex-15)
pobj(in-14, Annex-15)
```

2.2.2 *BART Anaphora*

Le résolveur d'anaphores *BART Anaphora* Versley et al. (2008) utilise des méthodes statistiques, après une phase d'analyse syntaxique, pour la résolution d'anaphores. L'entraînement s'effectue sur des corpus annotés existants au format MMAX2 [4] Müller et Strube (2006). Les performances du système peuvent être améliorées en l'entraînant sur des corpus de textes de réglementations.

BART Anaphora produit en sortie des phrases annotées par des relations de co-références, permettant de lier des mots ou des groupes de mots. Comme le montre l'exemple ci-dessous :

```
<coref set-id="set_2">
 <w pos="dt">an</w>
 <w pos="nn">assembly</w>
</coref>
...
<coref set-id="set_2">
 <w pos="dt">the</w>
 <w pos="nn">assembly</w>
</coref>
```

2.3 Intégrations des composants dans UIMA

Dans UIMA, le texte, une fois analysé, est chargé dans un *Common Analysis System* (CAS). Le CAS permet l'échange de données entre les différentes annotations que l'on a défini pour représenter les données linguistiques.

Pour le module encapsulant le *Stanford Parser*, on a défini trois types correspondants aux métadonnées les plus pertinentes pour l'analyse des textes de règles :

- le type **Word** représente chaque mot de la phrase et a pour attribut sa partie du discours – ce sont les métadonnées de la phase l'étiquetage morpho-syntaxique – ;
- le type **Constituant** représente les groupements grammaticaux dans la phrase et ayant pour attribut les mots ou les constituants lui appartenant – ce sont les métadonnées issues de l'analyse syntaxique – ;
- le type **Dependent** et **Governor** qui désignent les deux éléments qui composent une relation de dépendance, ils ont pour attributs le Governor (resp. Dependent) associé, ainsi que la relation de dépendance – ce sont les métadonnées pour l'analyse en dépendance –.

Pour le module de résolution d'anaphore, les deux types nécessaires représentent respectivement la notion référencée et ses références : le type **notion** désigne la notion référencée, le type **reference** représente les références trouvées. Notons qu'une classe d'anaphore est une notion propre à *BART* qui permet de regrouper une liste de termes qui ont le même sens.

On a défini également trois *Analysis Engine* (AE) afin de créer une chaîne de traitement permettant l'exécution des composants :

4. http ://mmax2.sourceforge.net

StanfordAE il permet de définir le module qui encapsule le Stanford parser. Il se base sur les résultats du « SentenceAE » (qui traite le texte phrase par phrase) et produit les annotations de dépendances et de mots.

BartAE définit le module qui encapsule *BART*. Il produit les annotations de notions et de références. Cet annotateur prend en entrée le texte dans son intégralité (une référence pouvant référer à une anaphore distante de plusieurs phrases).

ParsingAE définit la chaîne d'exécution composée des AE définis précédemment. C'est un AE d'agrégation (cf Figure 2).

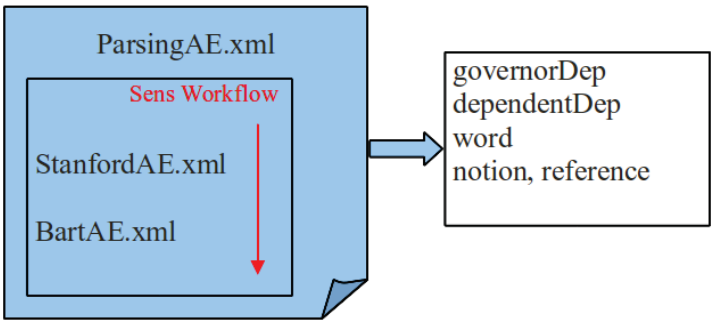


FIG. 2 – *Schéma de la chaîne d'exécution UIMA*

3 Structures de données et annotations

3.1 Les structures de données

Les structures de données utilisées permettent de formater les informations linguistiques extraites du texte avec le *Stanford parser* et *BART* pour qu'elles soient réutilisables de manière optimale par le module d'extraction de règles.

Les marqueurs grammaticaux sont stockés dans une `HashMap` qui pour chaque mot (ayant un identifiant unique) fait correspondre un objet `POS` (objet ayant 4 attributs : le mot lui-même, le numéro de phrase à laquelle il appartient, le numéro de sa position dans la phrase, son marqueur grammatical).

Les dépendances sont sous la forme de trois structures différentes adaptées à divers besoins. La première structure est une `HashMap` qui à tout type de relation trouvée dans le texte associe une liste de couple de mots qui interviennent dans le type de relation, comme le montre l'exemple ci-dessous, pour la relation **aux** (auxiliary), qui lie trois couple de mots :

```
Rel: aux
  [Dep: (necessary, 8), Gov: (may, 6)]
  [Dep: (check, 14), Gov: (to, 13)]
  [Dep: (proceed, 25), Gov: (shall, 24)]
```

La deuxième structure est une `HashMap` qui associe à chaque gouverneur une liste de couples (relation, dépendant), permettant d'accéder à tous les dépendants d'un mot du texte, comme le montre l'exemple ci-après (pour le mot **necessary**) :

```
Gov: (necessary, 8)
 [Reln : (nsubj);Dep : (that, 5)]
 [Reln : (aux);Dep : (may, 6)]
 [Reln : (cop);Dep : (be, 7)]
 [Reln : (dep);Dep : (for, 10)]
 [Reln : (xcomp);Dep : (check, 14)]
```

La troisième structure est un graphe direct acyclique (DAG) qui permet d'organiser les dépendances d'une phrase donnée de façon optimale pour l'identification des règles. La figure 3 montre la représentation en DAG de la section 7.2.1 de la réglementation européenne sur les ceintures de sécurité (cf. section 4). Chaque arc représente une relation et a pour source le gouverneur de la relation et pour cible le dépendant.

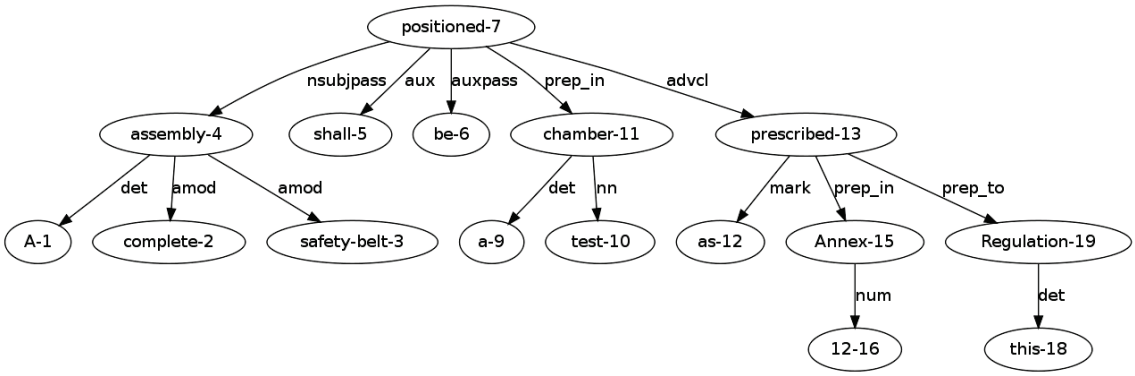


FIG. 3 – *Graphe direct acyclique des dépendances de la section 7.2.1.*

3.2 Annotations RDF

Les informations linguistiques sont exportées sous format RDF, elle peuvent ainsi être exploitées dans le module d'aide à la modélisation via des requêtes SPARQL, mais aussi être réutilisées par les membres du consortium ONTORULE.

Le schéma selon lequel sont définies les annotations RDF étend (cf. figure 4) le vocabulaire d'annotations [5] El Ghali et al. (2010), une classe `brem:LinguisticAnnotation` étend la classe `annot:Annotation`. Elle a pour sous-classes les classes correspondantes aux annotations UIMA. Les triplets produits sont une représentation des annotations UIMA en RDF. En voici quelques exemples [6] :

– Les mots (annotations UIMA de type **Word**) s'exportent dans des instances RDF de la classe `brem:Word` comme le montre l'exemple suivant :

5. http ://vocamp.org/wiki/HypiosVoCampParisMay2010#Annotations_Ontology
6. Dans les exemples nous utilisons la notation N3.

```
@prefix brem: <http://ontorule-project.org/ontologies/IBM/brem>.

brem:W398764   brem:hasvalue              "necessary";
               brem:hasindex               8;
               brem:hassentence            3;
               brem:hasID                  398764;
               brem:hasPOS                 brem:JJ .
brem:W308194   brem:hasvalue              "may";
               brem:hasindex               6;
               brem:hassentence            3;
               brem:hasID                  308194;
               brem:hasPOS                 brem:MD .
```

– Les relations de dépendances sont, elles, exportées comme des instances de la classe `brem:Dependency`, l'exemple ci-dessous montre les triplets RDF pour une instance de la relation " aux " :

```
@prefix brem: <http://ontorule-project.org/ontologies/IBM/brem>.

brem:D245749   brem:hasID                  245749;
               brem:hasdependent           brem:W398764.
               brem:hasgovernor            brem:W308194.
               brem:hastype                brem:Aux.
```

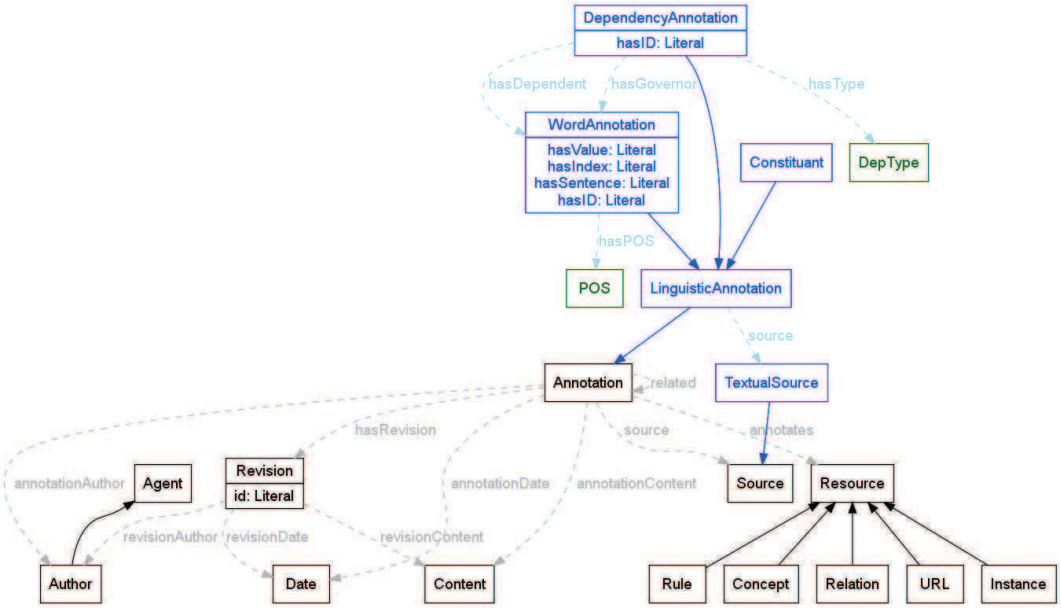


FIG. 4 – *Ontologie d'annotations*

4 Exemples de résultats

Pour effectuer nos tests, nous nous sommes basés sur des données fournies par Audi dans le cadre du projet ONTORULE. Ces textes traitent de réglementation sur les ceintures de sécurité dans les voitures. En voici un extrait :

> *7.2. Corrosion test*
> *7.2.1. A complete safety-belt assembly shall be positioned in a test chamber as prescribed in Annex 12 to this Regulation. In the case of an assembly incorporating a retractor, the strap shall be unwound to full length less 300 ± 3mm. Except for short interruptions that may be necessary, for example, to check and replenish the salt solution, the exposure test shall proceed continuously for a period of 50 hours.*
> *7.2.2. On completion of the exposure test the assembly shall be gently washed, or dipped in clean running water with a temperature not higher than 38°C to remove any salt deposit that may have formed and then allowed to dry at room temperature for 24 hours before inspection in accordance with paragraph 6.2.1.2. above.*

Pour la partie Anaphore on peut observer dans la figure 5 les notions et références trouvées par *BART*, telles qu'elles peuvent être visualisées dans l'interface UIMA.

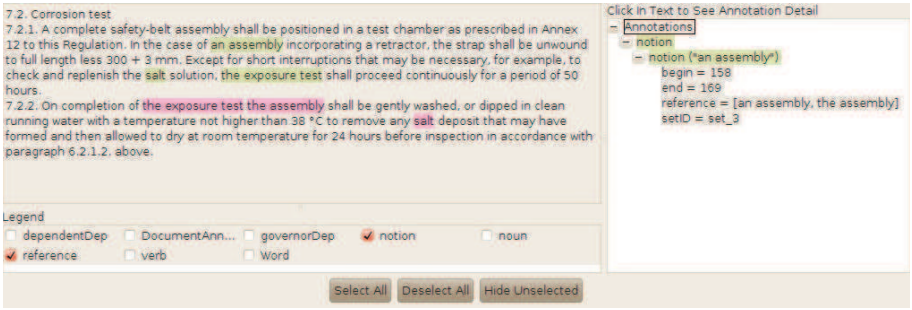


FIG. 5 – *Notions visualisées dans l'interface UIMA*

La figure 6 montre un exemple d'annotations issues de l'analyse en dépendance. Le mot « assembly » (partie 7.2.2 dans le texte) est déterminé par l'article « the » qui est le septième mot de la phrase (le titre "7.2.2." compte pour un mot).

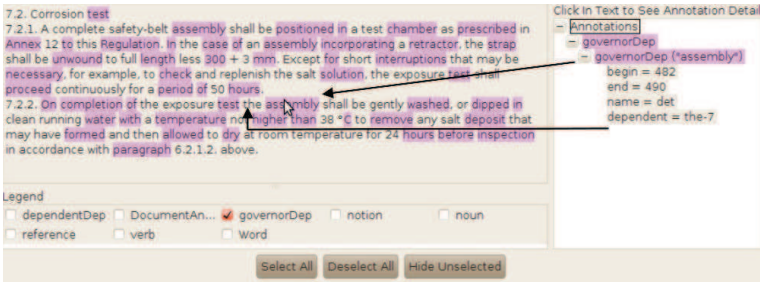


FIG. 6 – *Gouverneur d'une relation visualisée dans l'interface UIMA*

D'autre part, les annotations RDF peuvent être utilisées dans le module d'aide à la modélisation pour assister les analystes dans l'écriture de modèles et de règles. Pour ce faire, nous leur fournissons une batterie de requêtes SPARQL paramétrables, qui leur permettent d'interroger la base d'annotations, par exemple, pour retrouver toutes les relations potentielles dans lesquelles un Concept "Car" dans le modèle joue le rôle de sujet (les relations de dépendances sont représentées dans l'ontologie par les sous-classes de DepType, elles sont organisées dans une hiérarchie, par exemple, la classe DepSubj est la super-classe de NSubj, CSubj, XSubj), ils utiliseront la requête suivante pour pouvoir recupérer l'ensemble des relations possibles associées à la classe "Car" pour l'écriture des règles (voir ci-après).

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX annot: <http://vocamp.org/hyvoc/annotations#>
PREFIX brem: <http://ontorule-project.org/ontologies/IBM/brem#>
SELECT DISTINCT ?r
WHERE {
   ?w a brem:Word .
   ?z a brem:Word .
   ?d brem:hasGovernor ?z .
   ?d brem:hasDependent ?w .
   ?d brem:hasValue ?r .
   ?d brem:hasType brem:DepSubj .
   ?w brem:hasValue ?r "Car"
}
```

5 Conclusion

Dans cet article, nous présentons une chaîne de traitement UIMA pour l'analyse de textes de réglementations. Cette chaîne comporte deux composants linguistiques principaux, d'une part, un analyseur syntaxique le *Stanford Parser* qui fournit des métadonnées morphologiques, syntaxiques et de dépendances, d'autre part, un résolveur d'anaphores *BART Anaphora* qui permet d'identifier les co-références dans le texte. Ces modules encapsulés dans le framework UIMA, produisent des métadonnées qui sont unifiées dans une structure homogène puis exportées en RDF pour être utilisées dans le module de détection de règles et d'aide à la modélisation.

Références

Cer, D., M.-C. de Marneffe, D. Jurafsky, et C. D. Manning (2010). Parsing to stanford dependencies : Trade-offs between speed and accuracy. In *7th International Conference on Language Resources and Evaluation (LREC 2010)*.

de Marneffe, M.-C. et C. D. Manning (2008). The stanford typed dependencies representation. In *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*.

El Ghali, A., D. Eynard, A. Guissé, et F.-P. Servant (2010). A vocabulary for annotations. http://vocamp.org/hyvoc/annotations#. HyVoc'2010, Paris, France.

Ferrucci, D. et A. Lally (2004). UIMA : An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering 10*(3/4), 327–348.

Hernandez, N., F. Poulard, S. Afantenos, M. Vernier, et J. Rocheteau (2009). Apache UIMA pour le Traitement Automatique des Langues. 16ème conférence sur le Traitement Automatique des Langues Naturelles (TALN'09) - Session Démonstration.

Hernandez, N., F. Poulard, M. Vernier, et J. Rocheteau (2010). Building a French-speaking community around UIMA, gathering research, education and industrial partners, mainly in Natural Language Processing and Speech Recognizing domains. In *Workshop Abstracts LREC 2010 Workshop New Challenges for NLP Frameworks*, La Valleta Malta.

Klein, D. et C. D. Manning (2003a). Accurate unlexicalized parsing. In *Proceeding in the 41 st Meeting of the Association for Computational Linguistics*, pp. 423–430.

Klein, D. et C. D. Manning (2003b). Fast extract inference with a factored model for natural language parsing. In *Advances in Neural Information Processing System 15 (NIPS 2002)*, Cambridge, MA, pp. 3–10. MIT Press.

Marcus, M. P., B. Santorini, et M. A. Marcinkiewicz (1993). Building a large annotated corpus of english: the penn treebank. *Computational Linguistics - Special issue on using large corpora: II 19*(2), 679–696.

Müller, C. et M. Strube (2006). Multi-level annotation of linguistic data with MMAX2. In S. Braun, K. Kohn, et J. Mukherjee (Eds.), *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pp. 197–214. Frankfurt a.M., Germany: Peter Lang.

Versley, Y., S. P. Ponzetto, M. Poesio, V. Eidelman, A. Jern, J. Smith, X. Yang, et A. Moschitti (2008). Bart: A modular toolkit for coreference resolution. In *Proceedings of the ACL-08: HLT Demo Session*, Columbus, Ohio, pp. 9–12. Association for Computational Linguistics.
om

Summary

This paper[7] introduce a UIMA workflow for analyzing regulation documents. This workflow rely on a dependency parser and an anaphora resolver, producing annotations that can be used to identify business rules in the text, and also facilitate the modeling process from textual documents for business experts.

7. This work has been partially funded by the FP7 EU-IST Integrated Project 2009-231875 ONTORULE.

Quelle sémantique pour la fusion de données textuelles ?

Valentina Dragos, Vincent Nimier

Office National d'Etudes et Recherches Aerospatiales
Chemin de la Hunière
91761 Palaiseau Cedex
valentina.dragos@onera.fr
vincent.nimier@onera.fr

Résumé. Ce papier aborde le problème de la fusion de données textuelles et s'intéresse à la manière dont la sémantique est utilisée pour mettre en œuvre ce processus. Dans un premier temps nous définissons la notion de fusion de données textuelles, en mettant en évidence ses verrous techniques ainsi que les difficultés sous-jacentes au traitement des textes. Ensuite, nous illustrons à travers différents travaux présents dans la littérature les approches sémantiques proposées pour mettre en œuvre ce processus. Ces approches concernent l'utilisation des ontologies ou des règles métier pour guider ou faciliter la fusion de données issues de textes. Le papier offre une image d'ensemble des solutions sémantiques développées en lien avec la fusion de données textuelles.

1 Introduction

Construire un sens, donner une valeur à une information dont on dispose est un besoin récurrent dans tout processus supportant une prise de décision. La multiplication des sources d'informations ainsi que leur diversification influencent fortement la manière dont on acquiert une information : plutôt que de la retrouver entièrement dans une source bien identifiée, nous sommes amenés de plus en plus souvent à «synthétiser » une information à partir de plusieurs informations fournies par des sources distinctes. Dans certains cas, l'information, fournie par une ou plusieurs sources, se construit au fil du temps.

Ce papier aborde la fusion des informations vue comme un processus permettant de synthétiser ou de faire émerger une information nouvelle à partir de plusieurs informations fournies par des sources distinctes, à différents moments. Par source on entend un gisement d'informations, disponibles en différents formats (images, sons, vidéos, articles de journal, rapports de renseignements d'origine humaine, etc.), indépendamment de son support (électronique, papier, etc.) et de sa disponibilité (sources ouvertes, documents à accès restreint, etc.).

Notre cadre applicatif est le domaine militaire, et la fusion des informations peut offrir une aide au renseignement ou encore un support pour la construction d'une tenue de situation. Dans ce domaine, une place importante a été accordée à la fusion de données numériques. Ainsi, de nombreux algorithmes ont été développés pour la reconnaissance et le tracking des objets d'intérêt [Blasch et al., 2000] ou pour le traitement des images fournies par différentes sources [Bendjebbour et al., 2001]. Pourtant, dans beaucoup de domaines, dont le

domaine militaire, les données textuelles sont prépondérantes. Sans compter les renseignements d'origine humaine, l'ensemble des images, des vidéos, voire des écoutes électromagnétiques débouchent sur des rapports textuels élaborés par des opérationnels, qu'il va falloir corréler avec d'autres informations. Dans ce papier, nous nous intéressons plus particulièrement à la prise en compte des informations issues de données textuelles, que nous désignons, par un raccourci de langage, par le syntagme « fusion de données textuelles ». Il s'agit d'une problématique de recherche émergente, imposée par l'augmentation du nombre de sources textuelles, telles que les sources ouvertes sur Internet et par le besoin de compléter les données numériques par des observations ou des interprétations humaines. Ces données étant difficilement traitables, les solutions de fusion proposées sont fondées sur l'utilisation d'un ensemble de connaissances précédemment acquises, modélisées et formalisées, tels que les ontologies ou les règles métier.

Dans ce papier nous nous intéressons à la manière dont ces connaissances, modélisées pour capturer la sémantique du domaine, interviennent dans le processus de fusion. Nous illustrons, à l'aide de travaux présents dans la littérature, les différentes approches sémantiques développées pour la fusion de données textuelles. Cette vision d'ensemble nous permettra de répondre à l'interrogation présente dans le titre du papier et fournira également un support pour notre travail en cours concernant le développement d'une méthode de fusion de messages textuels.

Le papier est structuré en deux parties : la première définit la fusion de données textuelles et présente ses verrous techniques ainsi que les difficultés liées au traitement des textes ; la deuxième détaille les approches sémantiques proposées pour la fusion de données textuelles.

2 Fusion de données textuelles

Ce paragraphe introduit la notion de fusion de données textuelles. Il présente les verrous techniques de ce processus ainsi que les difficultés liées au traitement de ces données.

2.1 Définition

La notion de fusion d'informations est définie de manière générale par [McGuinness, 2003] comme étant « *l'ensemble des techniques, méthodes et outils permettant d'exploiter la synergie des informations issues de sources multiples et distinctes (capteurs, documents, humains, etc.) afin de produire une information synthétique d'une qualité meilleure*». La notion centrale de cette définition est la synergie grâce à laquelle le résultat obtenu est une information plus complète et plus riche.

Cette définition s'avérant trop générique dans le domaine militaire, plusieurs modèles ont été développés pour apporter des précisions sur le processus de fusion. Ainsi, le modèle λJDL [Llinas et al., 2004] affine cette définition en identifiant trois niveaux de fusion d'informations, détaillés ci-dessous :

1. le niveau 1 (ou la fusion des objets) concerne l'identification des objets à partir de leurs propriétés. Il s'agit d'utiliser plusieurs sources d'informations afin de construire, au fil du temps, une représentation des objets dits d'intérêt évoluant dans un environnement.
2. le niveau 2 (ou la tenue de situation) concerne l'identification des relations existant entre différents objets. Plusieurs sources distinctes sont utilisées afin de retrouver les relations

pertinentes existant entre les différents objets évoluant dans un environnement, en s'appuyant sur les résultats issus de la fusion de niveau 1.

3. le niveau 3 (ou la fusion d'impact) concerne l'identification des conséquences (ou des effets) des relations identifiées lors de la fusion de niveau 2. Les résultats issus des niveaux précédents sont exploités afin d'identifier les conséquences d'une tenue de situation dans un environnement particulier.

Les données exploitées aux niveaux 1 sont de nature numérique : issues de différents types de radars, des images ou des vidéos. La multiplication des sources d'informations et la diversification de leurs types, les limites avérées des méthodes numériques permettant d'exploiter les données issues de différents capteurs, le besoin d'enrichir ou de compléter ces données par des descriptions textuelles réalisées par des opérateurs humains ouvrent la voie à des nouvelles recherches visant à exploiter les données textuelles.

La fusion de données textuelles apparaît ainsi comme une nouvelle direction de recherche. Si ces données sont riches en informations, elles s'avèrent difficilement exploitables et leur fusion nécessite dans un premier temps l'identification des particularités sous-jacentes à leur traitement. Ces difficultés sont discutées dans le paragraphe suivant.

2.2 Difficultés liées au traitement des données textuelles

Selon [Auger et Roy, 2007] les difficultés liées au traitement des données textuelles sont dues à la fois à la forme de ces données (non structurées) et à leur contenu, caractérisé par la présence de plusieurs types d'ambiguïtés.

Les textes sont constitués de suites de mots, agencés selon des règles grammaticales. Cependant, dans un texte les mots ont différentes fonctions (auxiliaires, déterminant, etc.) , par conséquent ils disposent de capacités différentes pour spécifier ou 'porter' le sens global d'un texte. Ainsi, certains mots désignent des entités nommées et permettent d'ancrer le texte dans un contexte particulier, alors que d'autres désignent des concepts-clé d'un domaine spécifiant ainsi la sémantique du texte. Ex. un **attentat à la bombe (concept)** a eu lieu à **Istambul (entité nommée)**. Dans la pratique, il est important d'identifier et de catégoriser ces éléments, mais cela demeure une tâche difficile, due au caractère non-structuré d'un texte.

Au-delà des difficultés évoquées ci-dessus, traiter les textes suppose se heurter à la présence de trois types d'ambiguïtés : linguistiques, référentielles et sémantiques. Ainsi, l'interprétation des textes repose sur le triangle sémiotique proposé par [Ogden et Richards, 1923]. Le triangle, voir fig. 1, met en évidence trois éléments : le sens (signifié, notion, abstrait, concept, idée, pensée), se rapportant à un objet (chose, concret, réalité) désigné par un nom (mot, signifiant, symbole, langage).

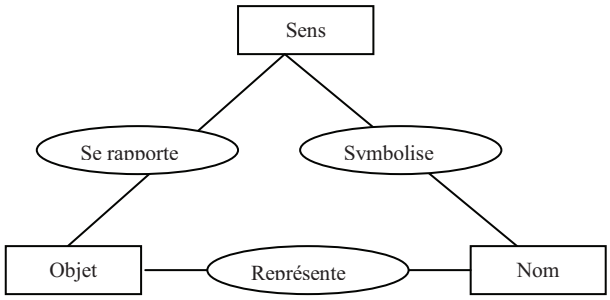


Fig. 1 – Triangle sémiotique, [Ogden et Richards, 1923]

Les ambiguïtés linguistiques sont dues au multiples liens existant entre le signifié et le signifiant (entre la notion et le mot la désignant).La synonymie correspond à l'existence de plusieurs mots faisant référence à la même notion, alors que l'homonymie correspond à un mot unique désignant plusieurs notions, voir fig. 2 et fig. 3

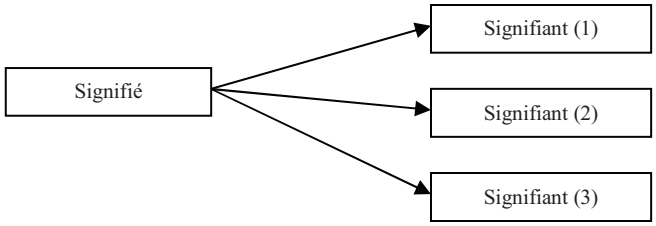


Fig. 2 – Synonymie

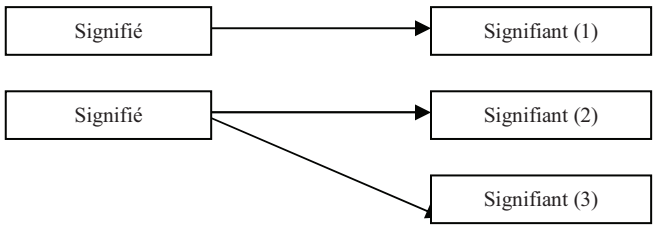


Fig. 3 – Homonymie

Les ambiguïtés référentielles sont dues au décalage entre le contexte de la production de l'acte langagier et le contexte décrit par son contenu. La production du langage se produit *hic et nunc*, et encode des représentations spécifiques du monde telles que perçues par les humains à un moment donné, dans un endroit spécifique. Les artefacts produits par les actes de discours (les données textuelles) sont en revanche interprétées en dehors de leur contexte original, voire dans de nouveaux contextes qui pourraient, à leur tour, avoir un impact sur l'interprétation originale du contenu. Selon un point de vue linguistique, les anapho-

res pronominales sont une source d'ambiguïtés référentielles. Exemple : Michel a vu son frère et **sa** femme. Dans cet exemple, le pronom **sa** peut référencer à la fois la femme de Michel ou la femme de son frère, et des informations contextuelles supplémentaires sont nécessaires pour identifier son rattachement.

Les ambiguïtés sémantiques présentes dans un texte concernent le domaine auquel il fait référence. La levée des ambiguïtés sémantiques permet de circonscrire le texte dans un domaine bien identifié.

Au-delà de ces difficultés inhérentes aux traitements des textes, la fusion de données textuelles fait émerger plusieurs verrous technologiques. Ces verrous, ainsi que la pertinence de solutions fondées sur la sémantique, seront discutées *infra.*

2.3 Verrous technologiques et apport de la sémantique

La fusion des données textuelles a pour but d'obtenir une information de qualité en exploitant des informations fournies par plusieurs sources. Les verrous technologiques de ce processus ont été identifiés par [McGuinness, 2003], et portent sur : la reconnaissance des informations redondantes, l'identification des conflits dans les informations, la reconnaissance de la corrélation des informations, l'enrichissement des informations et la découverte de nouvelles informations.

Les travaux existants dans la littérature s'appuient sur l'utilisation des ontologies et des règles métier pour le développement des solutions capables d'offrir des solutions aux différents verrous technologiques énumérés *supra*, tout en répondant aux spécificités de traitements des textes.

Ayant à la fois un niveau conceptuel, modélisant les entités d'un domaine, et lexical, indiquant les termes désignant ces entités, les ontologies représentent un outil approprié pour mettre en œuvre des solutions aux problèmes engendrés par la fusion de données textuelles. Dans certains cas, les règles métier s'avèrent utiles pour lever des limitations de traitements spécifiques aux ontologies.

Ci-dessous nous détaillons chacun des verrous tout en indiquant les apports de la sémantique dans la levée du verrou.

La reconnaissance des informations redondantes vise à identifier si une information a été fournie par deux sources distinctes et permet d'éliminer l'information dupliquée. Cette tâche peut être réalisée en utilisant le niveau lexical d'une ontologie, qui permet de mettre en évidence l'ensemble de synonymes utilisés pour désigner un concept.

L'identification des conflits dans les informations a pour but d'identifier si des informations contradictoires ont été fournies par des sources distinctes ou par la même source au fil du temps. Les contraintes définies dans une ontologie, telles que les concepts disjoints ou les relations inverses, permettent de mettre en évidence ces conflits.

La reconnaissance de la corrélation des informations vise à mettre en évidence les différents liens existants entre les informations fournies par des sources distinctes. Les relations sémantiques définies dans une ontologie explicitent les liens existant entre différents concepts et s'avèrent capables ainsi de mettre en évidence la corrélation des informations.

L'enrichissement de l'information vise à faire émerger une nouvelle information, plus riche, en combinant les différentes informations fournies par des sources distinctes. Cette tâche représente l'enchaînement logique de la précédente, car elle corrobore les informations complémentaires issues de deux sources. L'utilisation d'une ontologie peuplée permet de rapprocher des informations partageant un contexte commun. Par exemple, il devient possi-

ble de rapprocher des événements similaires ayant eu lieu dans un même endroit, à des moments de temps différents.

La découverte de nouvelles informations permet de retrouver des liens complexes entre les informations fournies par deux sources distinctes. Cette tâche peut être réalisée en utilisant les différents types de liens modélisés dans une ontologie ou encore des règle métier permettant de faire des inférences complexes.

Apres avoir défini la notion de fusion de données textuelles et présenté les caractéristiques de ce processus, nous présentons les différentes approches s'appuyant sur la sémantique développées pour le mettre en œuvre.

2.4 Approches sémantiques pour la fusion de données textuelles

Dans la littérature, les approches sémantiques développées pour la fusion de données textuelles se regroupent selon deux points de vue abordés : un point de vue interne, selon lequel la sémantique intervient dans les différents traitements spécifiques au processus de fusion et un point de vue externe, correspondant à l'utilisation de la sémantique pour la préparation du processus de fusion. Dans le premier cas, le processus de fusion s'appuie sur la sémantique ; dans le deuxième, il est uniquement guidé par la sémantique.

2.4.1 Point de vue interne : définition d'architectures sémantiques

Trois architectures sémantiques sont proposées par [Boury-Brisset, 2003] pour mettre en œuvre la fusion de données textuelles. La première est une architecture construite autour d'une unique ontologie offrant un vocabulaire commun pour décrire les informations fournies par différentes sources, voir fig. 4. L'ontologie sert à interpréter les informations fournies par chaque source selon la même sémantique formelle, ce qui permet de construire des passerelles sémantiques entre les différentes sources.

Cette approche a l'avantage d'une mise en œuvre facile car elle nécessite la construction d'une seule ontologie et s'avère plus facile en termes de gestion de l'évolution du domaine. Cependant, pour des domaines d'application étendus, cette solution engendre une problématique importante : pour assurer une bonne couverture du domaine, l'ontologie pourrait avoir une taille trop importante, ce qui risque de freiner ou de restreindre son exploitation effective.

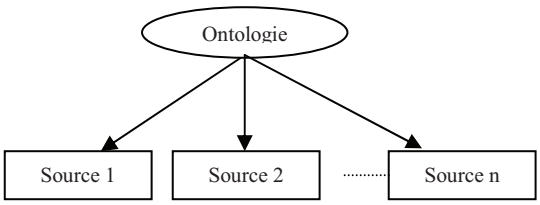


Fig. 4 – Architecture sémantique utilisant une ontologie

La deuxième architecture est fondée sur l'utilisation de plusieurs ontologies, chacune étant assignée à une source d'informations dont elle décrit la sémantique, voir fig.5.

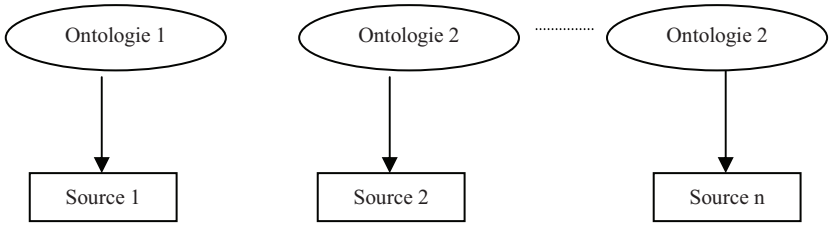


Fig.5 – Architecture sémantique utilisant plusieurs ontologies

En absence d'un vocabulaire commun partagé, la mise en correspondance des ontologies assure une interopérabilité virtuelle entre les différentes sources d'informations. Cette approche a l'avantage d'utiliser plusieurs ontologies décrivant chacune une partie du domaine. Il devient ainsi plus facile de construire chacune de ces ontologies et surtout d'intégrer les évolutions du domaine au niveau de chaque ontologie. Cependant, pour le but final concernant la fusion d'informations, la mise en correspondance des ontologies nécessite un effort assez conséquent. La redéfinition de l'alignement doit être réalisée après chaque mise à jour d'une ontologie, ce qui engendre des coûts d'exploitation assez importants.

La troisième approche est mixte : elle utilise plusieurs ontologies locales pour décrire chacune des sources d'informations, mais ces ontologies sont les spécialisations d'une ontologie globale, voir fig. 6.

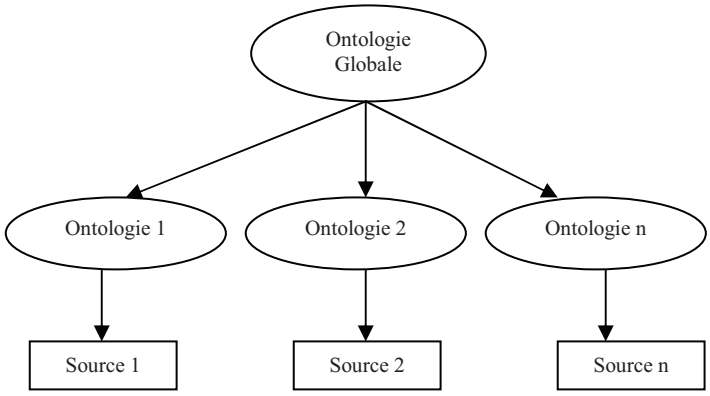


Fig.5 – Architecture sémantique mixte

L'ontologie globale offre le vocabulaire commun permettant de relier les informations fournies par les différentes sources, alors que les ontologies locales modélisent les concepts spécifiques de chaque source d'informations et établissent des correspondances avec les concepts de l'ontologie globale. Par rapport à l'approche précédente, l'utilisation d'une ontologie globale –ontologie pivot raccrochant les ontologies locales- permet d'éviter les problèmes liés à l'alignement entre plusieurs ontologies.

La mise en place d'un alignement d'ontologies orienté de l'ontologie pivot vers chaque ontologie locale réduit le coût de la mise en œuvre de cette approche et facilite également la gestion de l'évolution du domaine. Deux mécanismes permettent de réaliser cette évolution : le premier est un mécanisme top- down : l'intégration de nouveaux concepts dans l'ontologie globale et la spécialisation de ces concepts dans les ontologies locales. Le deuxième mécanisme est bottom–up : de nouveaux concepts sont intégrés dans les ontologies locales et cette intégration engendre des modifications au niveau de l'ontologie globale.

2.4.2 Point de vue externe : utilisation de la sémantique pour faciliter la fusion de données textuelles

Ce point de vue correspond à un positionnement différent concernant l'utilisation de la sémantique : plutôt que de l'exploiter dans le processus de fusion, comme ce fut le cas des travaux mentionnée ci-dessus, on utilise la sémantique pour faciliter la mise en œuvre du processus. Ainsi, selon [Kokar et al., 2004] identifient deux tâches pouvant être guidées par une ontologie : la description du format des données véhiculées par le processus de fusion (ses entrées et ses sortie) et la description des associations entre les différentes sources de données exploitées par la fusion.

Dans ces deux cas, il s'agit d'annoter sémantiquement les données d'entrée/ sortie du système, ou les sources des données afin de faciliter l'identification des corrélations existant entre les données ou les sources, ou encore faciliter la visualisation des résultats issus de la fusion.

2.4.3 Limitation des ontologies et utilisation des règles métier

Si l'apport des ontologies dans la fusion d'informations est mis en évidence par les travaux précédemment cités, leur utilisation dans la pratique est soumise à deux limitations mises en lumière par [Matheus, 2005].

La première limitation est imposée par le formalisme OWL et concerne l'utilisation exclusive des prédicats binaires. Ainsi, les ontologies exprimées en OWL ne permettent pas la modélisation des relations impliquant plus de trois entités.

La deuxième limitation est due à l'impossibilité de prendre en compte les aspects liés à l'évolution temporelle et au degré d'imprécision des données. Ainsi, il n'existe pas de primitive du langage OWL permettant d'exprimer une connaissance tout en indiquant son degré de validité ou de certitude. Pour palier ces problèmes il est possible d'utiliser des règles métier exprimant l'évolution temporelle des données ou leur qualité (degré de précision ou d'incertitude).

2.5 Conclusion

L'exploitation des données textuelles commence à susciter l'intérêt de nombreux chercheurs dans le domaine de la fusion des informations, particulièrement dans un contexte applicatif militaire. Dans ce papier nous avons illustré les principales contributions de cette communauté. Les approches proposées, bien que se trouvant encore en phase d'émergence, tentent de concilier la nature de textes et des schémas de recherche tirant profit de caractéristiques du domaine applicatif. Nous estimons que cette direction de recherche est prometteuse, car son objectif de faire émerger des nouvelles informations à partir de descriptions

textuelles lui permet de se démarquer des piste explorées jusqu'au présent par la fusion numérique des informations. Cette facette de la fusion des informations a toutefois été négligée jusqu'à maintenant, principalement par la difficulté de traitement de ces données.

Comme directions futures, une meilleure exploitation des techniques de traitement de la langue naturelle ainsi que la définition de nouvelles stratégies de prise en compte de la sémantique permettront de faire avancer l'état actuel du domaine et de bien établir ses fondements de recherche.

Références

[Auger et Roy, 2007] Auger, A. et Roy, J. Expression of uncertainty in Linguistic Data In: *In Proceedings International Conference on Weblogs and Social Media*Colorado, USA: , March (2007).

[Bendjebbour et al., 2001] Bendjebbour, A.; Delignon, Y.; Fouque, L.; Samson, V.; Pieczynski, W.; Multisensor image segmentation using Dempster-Shafer fusion in Markov fields context, Geoscience and Remote Sensing, IEEE Transactions on , Volume: 39 , Issue: 8 Digital Object Identifier: 10.1109/36.942557, Page(s): 1789 – 1798, 2001

[Blasch et al., 2000] Blasch, E. et Lang H., Data association through fusion of target track and identification sets, *In Proceedings of the Third International Conference of Information Fusion*, 2000

[Boury-Brisset, 2003] A.-C. Boury-Brisset, Ontology-based approach for information fusion, in: Proceedings of the Sixth International Conference on Information Fusion, pp. 522-529, 2003

[Kokar et al., 2004] Kokar M., Matheus C., Baclawski K., Letkowski J., Hinman M., and Salerno J., Use Cases for Ontologies in Information Fusion, *In Proeedings of Seventh International Conference f Information Fusion*, pages 415-421. (2004)

[Lambert, 2001]Lambert., D. A. Situations for Situation Awareness. In *The Fourth International Conference on Information Fusion,* 2001.

[Llinas et al., 2004] Llinas J, Bowman C., Rogova G., Steinberg A., Waltz E., White F. Revisiting the JDL data fusion model II, in Proceedings of the Seventh International Conference on Information Fusion, 2004.

[Matheus, 2005] Matheus, C., *Using Ontology-based Rules for Situation Awareness and Information Fusion*. Position Paper presented at the W3C Workshop on Rule Languages for Interoperability, April 2005

[McGuinness, 2003] D.L. McGuinness, Ontologies for information fusion, in: Proceedings of the Sixth International Conference on Information Fusion, pp. 650-656, 2003

[Nowak, 2003] Nowak, C., On ontologies for high-level information fusion, Proceedings of the Sixth International Conference of Information Fusion, pp. 657-664, 2003

[Ogden et Richards, 1923] Ogden, C. K. et Richards, I. A., The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, New York, Harcourt, 1923

Quelle sémantique pour la fusion de données textuelles ?

Summary

This paper addresses the problem of textual information fusion and highlights different semantic approaches developed to solve this problem. First, the fusion of textual data is introduced and some of his technical bottlenecks are identified. Forward, semantic approaches implementing this process are illustrated through various solutions proposed by the literature. Those solutions are using ontologies and business rules in order to facilitate the fusion process. The paper provides a comprehensive picture of the semantic solutions developed in connection with textual data fusion.

Toward a Versatile InfoRmation Toolkit for end-Users oriented Open-Sources explOitation : VIRTUOSO

Géraud Canet*, Gael de Chalendar*, Laurent Dubost**, Stéphan Brunessaux***
Gérard Dupont***, Axel Dyevre****, Khaled Khelif***, Bruno Quere**

*CEA-LIST ; Laboratoire Vision et Ingénierie des Contenus ; 18, rue du Panorama ; BP 6
92265 Fontenay aux Roses Cedex ; France
geraud.canet@cea.fr
gael.de-chalendar@cea.fr
**Thales, avenue carnot, 91883 Massy Cedex
laurent.dubost@fr.thalesgroup.com
***Cassidian, IPCC, Parc d'Affaires des Portes - BP 613 - 27106 Val-de-Reuil Cedex
France
gerard.dupont@cassidian.com
khaled.khelif@cassidian.com
****CEIS ; 280, boulevard saint-germain, 75007, Paris
adyevre@ceis.eu

Abstract. This paper describes the FP7-Sec VIRTUOSO project. This project aims at providing technical framework for the integration of tools for collection, processing, analysis and communication of open source information.

1 Introduction : context and motivation

The explosion of on-line information (textual information as well as other unstructured sources such as audio, images, and video) motivates most current work in content analysis and knowledge extraction. Although even if massive volumes of information are available at low cost as free textual, audio or video contents, people cannot read and digest this available information as fast as it is published and they are still looking for tools to mine the acquired content and allowing them to concentrate their attention and effort only on the valuable information. Content mining is the generalisation of text mining that can be seen as "the discovery by computer of new, previously unknown information, from different written resources". The acquisition of knowledge from unstructured data gives us the need to combine a wide variety of technologies. It needs also to be able to fuse information extracted from heterogeneous resources. The exploitation of the acquired knowledge to support the decision-makers is another challenge that will be addressed. The goal of the VIRTUOSO project is to innovate in the following items:

- **End-users' involvement**: integrating the end-users in the conception at all steps of the development/evaluation process.
- **Interoperability**: developing a system for managing and processing open source information, by developing new specific components but also by reusing and integrating available off-the shelf methods and tools. Satisfying the interop-

erability constraint during the design and the realisation of the platform by developing methodology, tools and supporting standardisation to encounter obstacles to interoperability between the technologies is the key success of proposal system.

- **Information Extraction**: developing an increasing approach based on an Open Information Extraction starting from domain specific (supervised information extraction) to domain independent (unsupervised information extraction, e.g. machine-learning)
- **Knowledge building (VKB):** developing a standard acquisition of knowledge from information extraction
- **Decision Support**: developing tools based on the acquired VKB Virtuoso Knowledge Base, to assist the decision-makers in their activities
- **Computing**: studying and evaluating the computation power needed for given application the volume of data

2 Objectives of the VIRTUOSO framework

VIRTUOSO will provide a technical framework for the integration of tools for collection, processing, analysis and communication of open source information. "Plug and play" functionalities that improve the ability of border control, security and law enforcement professionals to use data from across the source / format spectrum in support of the decision making process will be enabled by this middleware framework. As a proof of concept and to highlight the efficiency of this open-source code framework, a prototype will be built and demonstrated using operational scenarios. The project, developed by a consortium of 17 European organisations, is co-funded under the FP7 programme of the EU and will comply with legal considerations, enforcing the principles of privacy and data protection to ensure the interests of citizens within the European Union.

The main functional components of the VIRTUOSO system are:

Information Gathering components: collects the information coming from multiple supports (disks, intranet, the Internet…), multiple format (paper, digital, analogical...), and of multiple natures (textual, audio and visual (image and video), and then converts data into standard format. For instance paper documents will be scanned and translated via OCR API into XML output format, in the same way the speech audio documents will be converted, using speech to text API, into XML format.

Information Extraction and Structuring components: Structuring Open Source Information to Support Intelligence Analysis. It deals with information reduction and structured based on event and information extraction. It comprises information filtering and categorization, Information Retrieval, Entity and event extraction, Information summarization.

Knowledge Acquisition components: These services deal will the storage and the representation of the knowledge built from the extracted information;

Decision support components: The VIRTUOSO Decision-maker support toolkit will be developed to provide a substantial aid to the decision makers by integrating various sources of information and by developing tools to exploit the knowledge stored in the VKB and components for Situation Assessment, Scenario Building and Misinformation detection. It will provide also an intelligent access to or visualize relevant knowledge, and aid the process of structuring the work and the decisions.

3 Overall architecture

The VIRTOSO framework will rely on a "Service Oriented Architecture" (SOA) as the core paradigm for the design and integration of components. Each component that could be integrated in the platform shall implement one or several functionalities that are described by service interfaces. The function workflow needed to provide user applications will be done by putting together services and calling them in the right order. Each component, implementing one or several service interfaces won't have any knowledge of the other services and their capabilities. They will provide to the others one or several processing capabilities (i.e. services) which will be driven by the orchestrator to define business processes. As a consequence, the service definition and conception is a key step in the platform. The granularity of the services should be one of the main concerns during the design and development of a WebLab component.

In order to provide a flexible architecture, the service design should respect the following features:

- Loosely-coupled: It means that services should be as autonomous as possible and that dependencies between components should be avoided;
- Coarse-grained: A service should provide a coherent set of functions and should hide implementation complexity;
- Standardized interfaces: A service should implement one or more standardized interfaces from the service taxonomy and reuse the generic service interfaces. It gives access to the functions provided by the service;
- Integrable: A service can be easily integrated in a global application and be used in a chain with other services.

Every service is provided by a "producer" to a "consumer". The interaction between producer and consumer is carried out by a middleware (the enterprise service bus) that is responsible of the mediation and the communication between the services. As a consequence, a consumer invokes a service to carry out a function of which he is the producer.

A directory of services is provided in the platform to allow the producers to publish their service offers and enable the chain builder to select the right service on each step of a processing orchestration. A service call is done by sending a message through the middleware from the consumer to the publisher either in an asynchronous or a synchronous way.

In the architecture, we will consider:

- Business services that provide business functions (such as video segmentation, text clustering);
- Technical services that are part of the provided baseline (such as security, data access layer, etc.);
- Graphical User Interface (GUI) components that will interact with users on one side and with the service bus on the other side to request process or data.

In Virtuoso, the various components are splitted into four different and mainly autonomous frameworks responsible of different kind of things. They are the Processing framework, the Decision Support framework, the Evaluation framework and the Unified Presentation framework (see Figure 1). Besides the frameworks, there is (possibly several) knowledge bases used to store all the data extracted from components and consolidated by others.

The Virtuoso applications need to physically separate some parts of the system, such as data acquisition and preliminary transformations on the one hand and data processing, storage and exploitation on the other hand. The transfer of data from the first part to the second one has to be specified too.

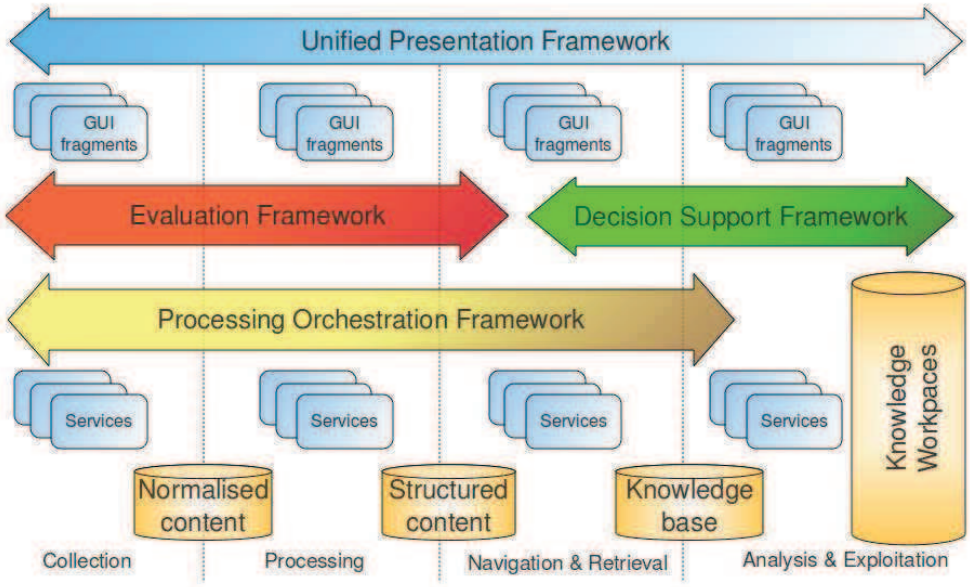


Figure 1 - The Virtuoso Architecture

3.1 The processing framework

All processing components developed in the context of the VIRTUOSO project will be integrated as services using the WebLab platform. The WebLab project is an open source framework developed and maintained by EADS (Giroux et al., 2008). Here, we summarize the WebLab project and its organization. This platform was initiated in the WebContent research project[1].

The WebLab platform provides a set of tools and methods that aim at building information systems for intelligence applications in the economy, strategy and military domains.

[1] http://www.webcontent.fr

Typically, WebLab handles multimedia and unstructured data (text, image, audio and video) through heterogeneous business components providing services. WebLab is made of 3 layers (Figure 2):

- the **WebLab Core** which represents the technical base designed to make heterogeneous components cohabit and work together within a service oriented architecture (SOA),
- the **WebLab Services** which are shown as a set of coherent software services dealing with elementary functionalities as well as GUI components which can be assembled to achieve custom-built applications, and
- the **WebLab Applications** which are the result of the integration of the WebLab Services based upon a common exchange model and common services interface provided by the WebLab Core. Thus, it guarantees a minimal effort for programmers to integrate their tools into a complete application.

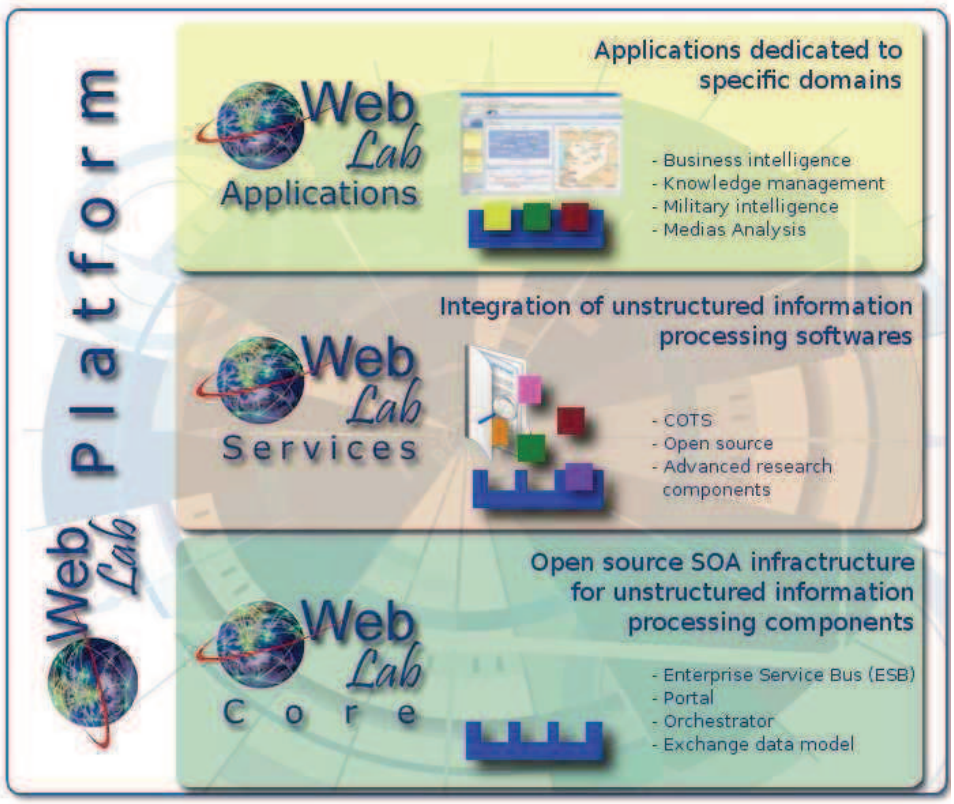


Figure 2 - The WebLab core, services and applications

WebLab relies on many standards, including semantic Web standards. In addition to these standards, WebLab proposes a data exchange model allowing heterogeneous components to communicate with each other (see Figure 3).

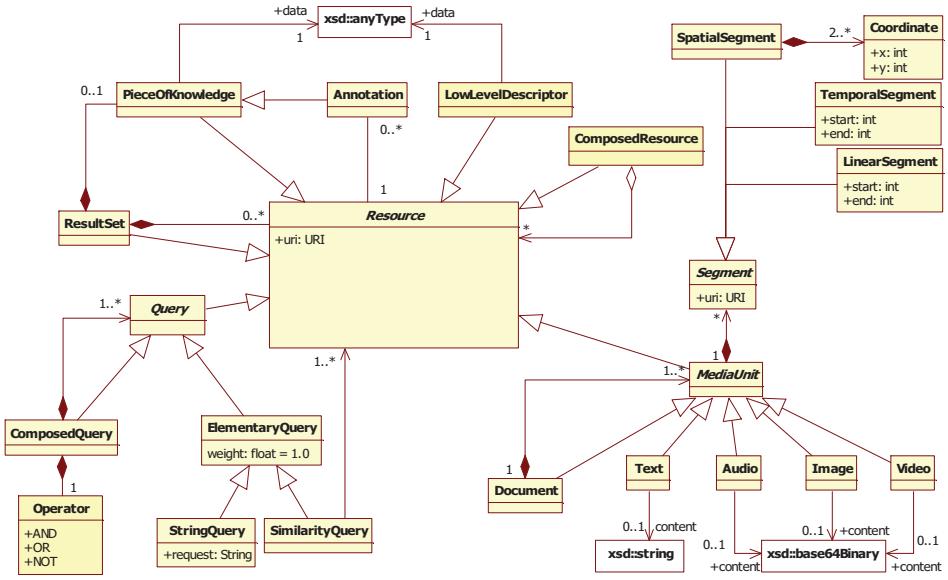


Figure 3- The WebLab exchange model

This exchange model is mainly organized around the *Resource* object which is a generalization of the different types of entities to manage by the platform. The *MediaUnit* object allows describing each information entity (audio, video, image, text) that will have to be managed by the different technical component. The model also enables to annotate the resources with RDF annotations (subject- predicate-object triples).

3.2 The decision support framework

In Virtuoso, Decisions Support and Visualization tools aim at exploiting the information produced by the processing and/or the users and stored in Knowledge Base.

These Decision Support and Visualization modules in VIRTUOSO will be integrated on top of a Decision Support Framework based on the ING Semantic Middleware. ING Semantic Middleware is developed and maintained by THALES. Its objective is to facilitate the development and integration of Decision Support and Visualization applications in the fields of Security, Defence, and Intelligence.

The semantic representation (i.e: by triples such as : Bruno “works for” Thales) may bring great operational benefits. It offers a great expressivity and its flexibility facilitates the interoperability of information systems. Moreover, the semantic representation is very well suited for the representation and computation of social networks which are crucial in security applications. In spite of all these potential operational benefits, semantic representation is rarely used in operational security systems, because operational knowledge is often complex and requires to be handled at a certain a level of abstraction (otherwise developers could easily be “overwhelmed by triples”!). ING solves this problem.

ING provides applications developers with high level standard interfaces to query, update a semantic knowledge base, compare and merge pieces of knowledge and enable adaptation (and hot deployment) of the knowledge base model.

Virtually, these standard interfaces enable application developers to use any database as a semantic repository, provided they use/develop the right connector. In VIRTUOSO, ING uses a connector to the ITM semantic repository developed by MONDECA.

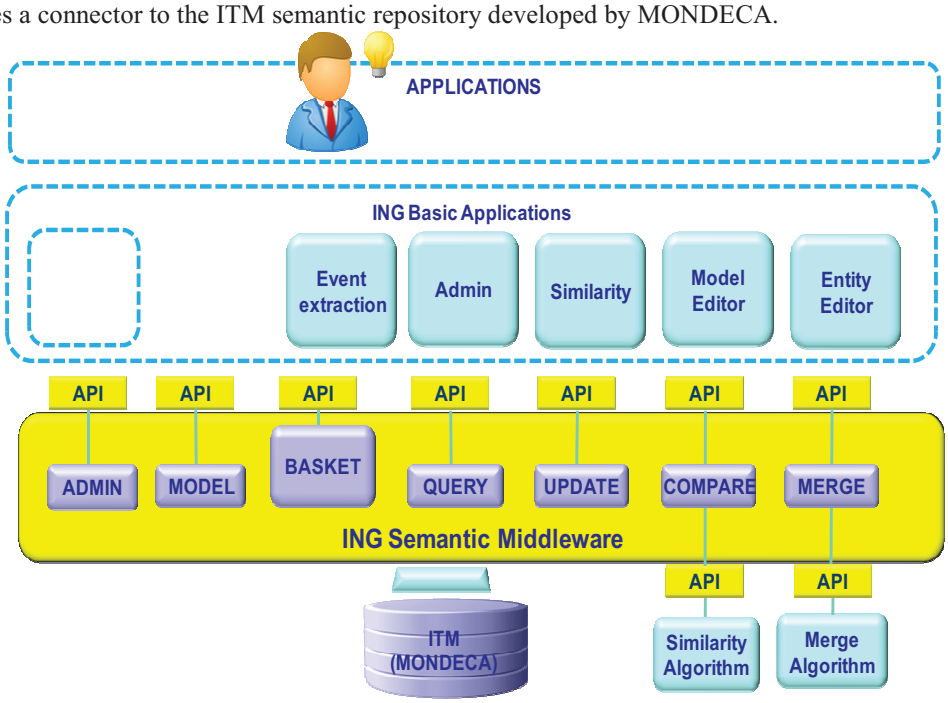


Figure 4 - ING Semantic Middleware APIs & basic applications

All applications developed on ING use the "Basket" service to communicate. Via the basket, the result of a query, can be displayed by another application, such as a graph viewer. Applications read their inputs in the basket and write their outputs in the basket. As a result, the basket provides a simple and practical journalizations of users operations, each operation being a view (i.e: layer) on the knowledge base.

In addition to these middleware features, Thales has developed several basic applications on ING entity edition, model edition, collective data fusion, event extraction… these basic applications are brought to the VIRTUOSO project.

The technical choices of ING Semantic Middleware (J2E, SOA, web client…) make the VIRTUOSO Decision Support Framework fully compatible with Processing Framework based on the WEBLAB project .

The release of the ING middleware as an open source software is currently being considered by Thales within the VIRTUOSO project.

3.3 The evaluation framework

The aim of the evaluation framework is to define a standardized business process to evaluate the scientific quality of the various tools issued from the processing framework. Evaluations will internally use datasets and evaluation software proposed by international evaluation campaigns, such as TREC for information retrieval for example. These tools and datasets will be wrapped inside Web services implementing standard WebLab interfaces. Figure 5 shows the envisioned architecture of an evaluation setup using this evaluation framework.

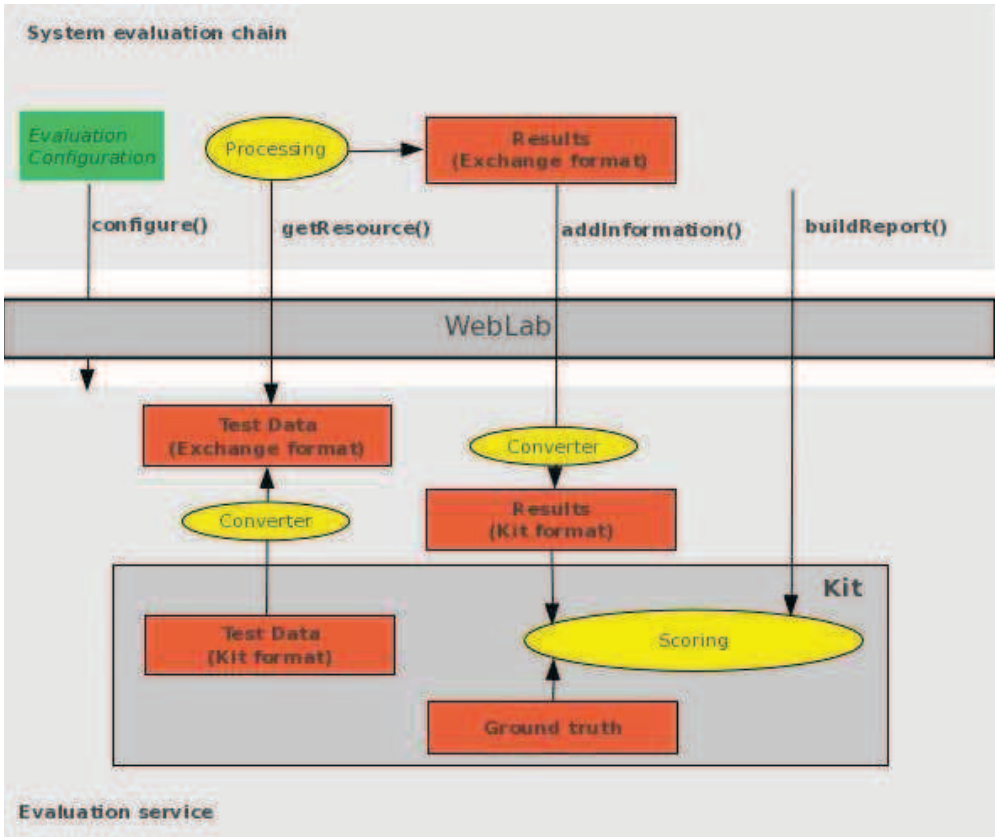


Figure 5 - The evaluation process

An evaluation service must implement three WebLab interfaces:

- *Configurable* to be able to set global information related to an evaluation, such as the run name;
- *SourceReader* to enable the service being evaluated to retrieve test data in WebLab data exchange format;
- *ReportProvider* to allow the service being evaluated to submit its data in WebLab data exchange format and to get the evaluation result as a WebLab document, results being expressed in XML/RDF.

Thus the creation of a Virtuoso evaluation service consists in writing programs able to:

- convert the test data format into the WebLab exchange model format;
- convert the WebLab exchange model format to the submission format used in the evaluation campaign;
- express the evaluation result as XML/RDF and generate a result report in the WebLab exchange model.

For each targeted evaluation campaign, it will be necessary to define an ontology enabling to express the evaluation data such as scores. We will try to use as much as possible a core evaluation ontology for standard elements such as precision and recall and specialized ontologies for the kind of data specific to a campaign.

The work on the evaluation framework has just started. Currently, the list of evaluation kits necessary to evaluate all the kind of processing components that will be developed during the Virtuoso project is not complete. Some of the kits considered are:

- Named Entities Extraction, using ESTER-NE or ACE 2007 data;
- Topics Detection, using TDT 5 data;
- Documents Retrieval, using CLEF 2003 data;
- Objects Classification (in images) using PASCAL VOC data;
- Similar Images Retrieval, using ImageCLEF data;

3.4 The presentation framework

This last layer will ensure the standardisation of the user interfaces of the application upon the platform. This covers three different aspects: graphical aspects, integration methodology and standardization of communication between user interface fragments. As part of the integration in the global architecture the presentation framework will rely on a web interface hosted on a web portal and each pieces of interfaces will be integrated using common technologies.

Graphical aspects

The graphical aspects will enable the use of coherent graphical recommendations to identify functionalities and/or information presented to the user. Since every application can have its own graphical identify which will define the set of colours, shapes and guidelines to represent different part of the interface.

Thus, the standardisation of graphical aspects will recommend using standardized and recognized technologies in order to implement the fragment of interfaces such as the following W3C standards like HTML and CSS.

However to the inherent flexibility of these languages to structure information, extra guidelines will define best practice in the use of these languages in order to attach precise semantic to some meaningful elements of interfaces such as : content extracted from document and/or knowledge base ; generated content ; functionalities that manipulate content... The precise definition of these elements and the way their presentation will be implemented, are currently under specification, including naming conventions to identify the different parts of the interface. The use of supplementary W3C proposal, such as RDFa, will be investigated as part of this specification.

Integration methodology

The graphical user interface will be exposed to the user through a Web portal relying on the Portlet specifications (namely JSR168 and JSR286). These recommendations define the specific life-cycle of user interface as well a communication between interface elements

through the event mechanism. Originally proposed for the standardisation of interface elements in JAVA, the Portlet specifications have been extended with WSRP (Web Service for Remote Portlet) which enable the integration of non-JAVA interfaces that are remotely exposed as a specific Web Service compliant with WSRP.

Standardization of communication

The second aspects will be the use of a specific and standardised communication layer on the user interface level in order to enable exchange of information between the heterogeneous user interface fragments that comes from multiple providers.

The Portlet specification and the event mechanism already provide a technical solution for the production and consummation of messages as well as a way to define multiple types of messages. The framework will define a common typology of possible messages in order to address the multiple levels of information exchanges. The specification of these messages will follow the philosophy used in the processing framework to define generic and standardised services interfaces. Thus each interfaces elements will be organized depending on the data they can consume and/or produce as well as the functionalities they offer. Content of messages will re-use the data exchange model for the exchange of content and semantic standards (such as RDF/XML) for the exchange of high level information coming from the knowledge base.

As for the others frameworks the specifications proposed in at the presentation level will allow to ease integration and composition of multiple components for the design of application.

4 Conclusion

In this paper we described the general approach we designed to tackle the technical needs of the VIRTUOSO project.

Not only technical, the VIRTUOSO project is mainly a professional-oriented project. Over the course of the project, a community of European intelligence specialists and analysts has been gathered in a series of recurring workshops, where inputs are taken from the end-users. Every architecture detail, every innovative tool is confronted to the real-life needs of the end-users community. This community is a premiere in the field addressed by the project.

Another goal of VIRTUOSO is to go beyond existing integration platforms such as UIMA (Ferrucci and Lally, 2004), LinguaStream (Bilhaut, 2003), and Gate (Cunningham et al., 2002) (specially designed for the processing of textual documents) by (i) the use of recognized standards (RDF, XML, BPEL, etc..), (ii) processing multimedia documents, (iii) provide features for technical evaluations and (iv) provide mechanisms for reasoning on extracted information.

References

Giroux, P., Brunessaux, S., Brunessaux, Sy., Doucy, J., Dupont, G., Mombrun, Y., Saval, A. (2008). *Weblab : An integration infrastructure to ease the development of multimedia*

processing applications. In International Conference on Software and System Engineering and their Applications, ICSSEA. Paris

Ferrucci D. and A. Lally, A. UIMA: an architectural approach to unstructured information processing in the corporate research environment, 2004 Natural Language Engineering 10, No. 3-4, 327-348.

Bilhaut F., The LinguaStream Platform 2003 Proceedings of the 19th Spanish Society for Natural Language Processing Conference (SEPLN), Alcalá de Henares, Spain, 339-340.

Cunningham H., D. Maynard, K. Bontcheva, V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. 2002 Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia.

Acknowledgment

Funding by the Virtuoso[2] Project (FP7-SEC-GA-2009- 242352) is kindly acknowledged.

[2] http://www.virtuoso.eu/

Architecture, services et connaissances dans un système d'aide à la veille : Application à la détection de signaux faibles en sources ouvertes

Emilien Bondu*,** Patrick Giroux*
Habib Habdulrab**

*EADS - Cassidian, Parc d'Affaires des Portes - BP 613 - 27106 Val de Reuil Cedex - France
**LITIS, Avenue de l'Université - BP 8 - 76801 Saint-Étienne-du-Rouvray Cedex - France

Résumé. L'activité de veille stratégique et la détection de *signaux faibles* sur des sources ouvertes d'information est aujourd'hui de plus en plus complexe. En effet, avec la croissance du web 2.0 et la multiplication des gisements d'informations, le veilleur se trouve confronté, d'une part, à une masse de données toujours plus importante et d'autre part, à une quantité de formats et structures de données croissante. Dès lors, l'utilisation d'un système d'information adapté à l'activité de veille devient nécessaire pour collecter et filtrer ces informations, mais également pour gérer les différentes connaissances expertes utiles à la détection précoce de menaces. Dans cet article, nous nous intéresserons aux solutions apportées par une plateforme d'intégration open source, dédiée à la fouille de documents multimédia. Nous proposerons alors des solutions pour permettre l'interopérabilité d'outils variés de traitement de l'information ou de gestion de connaissances et également des solutions pour proposer aux veilleurs des applications personnalisables et adaptées à leurs besoins. Nous étudierons alors l'architecture et les solutions techniques de la plateforme ainsi que les standards du Web sémantique pour évaluer leur pertinence en réponse aux problématiques identifiées lors de la mise en place de tels systèmes.

1 Introduction

Avec la croissance d'Internet, notamment par le développement du Web 2.0 et l'émergence de nouvelles sources d'information ouvertes et accessibles à tous, de nouveaux systèmes plus performants sont désormais nécessaires pour accéder et traiter les informations disponibles. De plus, dans le cas de la veille, ces systèmes doivent fournir une aide aux veilleurs pour détecter des *signaux faibles* (Ansoff, 1976), c'est à dire des informations pouvant constituer des signes précoces de changement ou de menace. Pour ce faire, de nombreux outils open source ou COTS [1] sont mis à contribution pour traiter ces informations. Dès lors, le développement de tels systèmes se complexifie et doit s'appuyer sur des architectures logicielles flexibles. En effet, cette flexibilité est aujourd'hui nécessaire pour permettre d'une part l'intégration facile de ces

1. Commercial Off-The-Shelf (composant pris sur étagère)

outils et d'autre part pour autoriser un redimensionnement de l'application pour envisager le traitement de grand volumes d'informations. Nous identifierons dans un premier temps, les principales problématiques rencontrées lors de la mise en place d'un système veille capable de traiter des informations issues de sources ouvertes. Nous nous intéresserons également à l'identification et la gestion de connaissances exploitables pour la détection de signaux faibles. Enfin, nous étudierons la pertinence d'une architecture SOA [2] et d'un ensemble de services dédiés pour répondre à ces problématiques.

2 Problématiques

2.1 Problématiques liées aux sources d'information ouvertes

Le développement des moyens de communication, notamment par le biais du Web 2.0, favorise l'émergence de nouvelles sources d'information de plus en plus complexes à traiter dans le cadre d'une activité de veille. En effet, ces nouvelles sources mettent à disposition des informations possédant des caractéristiques particulières que nous tenterons d'identifier. La prise en compte de ces caractéristiques est primordiale pour obtenir un système adapté, à la fois aux besoins des veilleurs mais également aux informations que le système est amené à traiter. Ainsi, les principales problématiques rencontrées lors du traitement des informations issues de sources ouvertes sont liées à :

La quantité d'information à traiter : la quantité et le volume d'informations mises à disposition aujourd'hui en sources ouvertes, notamment avec la croissance des contenus vidéos en ligne, sont tels qu'il devient impossible de collecter manuellement toutes ces informations. En effet, la collecte de ces grands volumes nécessiterait des connexions réseaux rapides, du temps, ainsi qu'un espace de stockage important, dont on ne dispose pas. De plus, face à cette quantité d'informations disponibles, le veilleur se retrouve noyé (Lesca et Schuler, 1995) et il devient impossible pour lui de traiter efficacement ces nouvelles données, et de discerner clairement les informations clés. Comment traiter ces volumes importants et intégrer dans un système d'aide à la veille, des outils adaptés pour configurer et automatiser la collecte ?

L'hétérogénéité des formats et structures d'informations : les informations disponibles sont proposées dans des formats variés (pages HTML, flux RSS, réseaux sociaux, wiki, blogs, forums, etc) et ne sont pas toujours structurées. Le traitement de ces ressources par les veilleurs passe par l'utilisation d'un ensemble varié d'outils, souvent non interopérables entre eux. Pour utiliser convenablement ces outils le veilleur doit se former à leur utilisation combinée, augmentant encore la complexité de son travail. Comment réduire la complexité du système d'information en rassemblant au sein d'une même application un ensemble varié d'outils, tout en restant ouvert aux formats et structures de données existants, ou en développement ?

2. Service Oriented Architecture

2.2 Problématiques liées à la gestion de connaissances dans un système de veille adapté

Le processus de veille exploite les nombreuses connaissances du veilleur ou de l'expert (Cao, 2006), que ce soit par le biais de son expérience, de sa connaissance des domaines et des thèmes impliqués dans l'activité de veille, ou des connaissances mises en jeu lors du processus de création du sens (Lesca, 2002). De même, sa capacité à raisonner et à synthétiser l'information implique l'usage de processus cognitifs reposant eux-mêmes sur des connaissances variées. Par conséquent, pour mettre en place un système d'aide à la veille adapté et performant, il convient de prévoir la mise en place d'outils dédiés à la gestion de l'ensemble de ces connaissances au sein de ce système. Ces problématiques liées à la gestion de la mémoire de l'entreprise (*Knowledge Managment*) (Dieng-Kuntz et al., 2001) abordent des axes de recherches variés, et font l'objet de nombreuses études et réflexions, menées notamment pour établir et définir les standards du Web Sémantique. Nous résumons ici les principales problématiques, particulièrement impliquées dans le cas de la veille stratégique auxquelles nous tenterons de fournir une solution par la suite :

La construction des connaissances : les experts utilisent des connaissances tacites qu'ils ont souvent du mal à exprimer ou à retranscrire au sein d'un système de gestion des connaissances. De même, la construction de façon collaborative d'une connaissance d'entreprise, partagée et acceptée de tous, reste une tâche longue et complexe du fait, par exemple, des nombreux points de vue, parfois divergents, exprimés par les experts. Comment faciliter la capitalisation des connaissances expertes au sein du système de gestion de connaissances ?

La mise à jour des connaissances : par définition, l'activité de veille s'intéresse aux évolutions d'un environnement changeant, où l'on observe de nouvelles situations et où l'on exploite un ensemble d'informations et de connaissances en rapport avec ces changements. Par conséquent, les connaissances impliquées dans le processus de veille évoluent elles-mêmes en permanence. Un système d'aide à la veille efficace doit donc permettre la mise à jour de ces connaissances. De la même façon que pour leur construction, la mise à jour de façon collaborative des connaissances peut constituer une tâche longue et complexe. Face aux changements rapides de l'environnement, comment rapidement mettre à jour ces connaissances, de façon collaborative ?

L'exploitation des connaissances : pour impliquer l'expert ou le veilleur dans la capitalisation des connaissances, il est nécessaire que le système exploite ces connaissances et présente à l'utilisateur un résultat pertinent issu de cette exploitation, justifiant alors l'effort de capitalisation. Il convient donc de formaliser ces connaissances de façon à les rendre compréhensibles et exploitables par le système. Comment formaliser des connaissances expertes et les rendre exploitables par le système et ainsi justifier l'effort de capitalisation ?

2.3 Problématiques liées à la mise en place d'un système d'aide à la veille

Aujourd'hui, les outils d'aide à la veille tentent de répondre à ces premières problématiques, mais du fait de l'évolution constante des technologies, des formats et des structures

d'information, ces outils ne sont pas toujours cohérents et évolutifs. Exploiter ces outils dans un processus de veille pose également de nombreux problèmes, parmi lesquels :

Le choix des outils et des technologies : l'essor des nouvelles technologies s'accompagne d'une diversité croissante des outils mis à la disposition des veilleurs, que ce soit par les communautés scientifiques ou par les éditeurs spécialisés. L'offre en outils de traitement de l'information est telle qu'il devient difficile pour le veilleur de déterminer quels sont les outils dont il a vraiment besoin. De plus, la prise en main de ces outils peut s'avérer coûteuse en temps pour les veilleurs qui doivent s'approprier l'outil pour en tirer profit. Comment suivre, choisir et profiter des nouvelles technologies et des nouveaux outils en les intégrant facilement au sein d'une application de veille déjà existante ?

L'interopérabilité des outils : nombre de logiciels disponibles sont souvent mal adaptés au veilleur car ils sont soit trop spécialisés, c'est-à-dire qu'ils offrent des fonctionnalités intéressantes mais difficiles à prendre en main, soit ces logiciels fournissent beaucoup de fonctionnalités mais peu d'entre elles sont réellement intéressantes. Dès lors il serait intéressant de combiner au sein d'une même application, les fonctionnalités intéressantes des différents outils existants. Cependant, ces outils n'offrent que peu ou pas de possibilité d'interfaçage entre eux. En effet, leur développement est réalisé indépendamment les uns des autres et chacun possède son propre modèle de données interne. Sans méthodologie dédiée, il est souvent impossible d'intégrer directement les fonctionnalités intéressantes des différents outils au sein d'une même application plus cohérente. De même il est souvent nécessaire d'entreprendre des développements spécifiques et complexes, hors de portée des compétences du veilleur. Comment intégrer facilement, au sein d'une même application, les fonctionnalités intéressantes d'un ensemble d'outils hétérogènes et non interopérables ?

L'ergonomie et l'interface utilisateur : en plus de disposer de fonctionnalités issues de différents outils, il est nécessaire que le veilleur dispose d'un système clair, cohérent et ergonomique. En effet, pour appréhender la complexité du système, le veilleur doit comprendre son fonctionnement et accéder facilement aux fonctionnalités dont il a besoin. De même, de nombreux utilisateurs demandent maintenant que l'interface graphique de l'application qu'ils utilisent soit personnalisable. Comment intégrer de façon ergonomique et personnalisable, au sein d'une même application, des IHM [3] provenant d'outils variés ?

3 Architecture et plateforme d'intégration

3.1 Plateforme d'intégration WebLab©

La plateforme d'intégration open source WebLab, développée par EADS, vise à faciliter l'intégration de composants logiciels et le développement d'applications exploitant des outils et des techniques de fouille de documents multimédia (Giroux et al., 2008). C'est au travers de nombreux projets de recherche comme WebContent ou VITALAS [4] et des projets opérationnels que le département IPCC [5] développe depuis 5 ans cette plateforme et son expertise en

3. Interface Homme Machine
4. Video & image Indexing and reTrievAl in the LArge Scale, http ://vitalas.ercim.org/
5. Information Processing Control and Cognition

traitement des informations non structurées et Web sémantique. L'indexation et l'accès à plusieurs millions d'images et à de milliers d'heures de vidéos dans le cadre du projet VITALAS, ont notamment validé les choix techniques retenus pour le développement de la plateforme. Au sein de ces projets, la plateforme offre la base technique (WebLab Core) en proposant des solutions techniques, notamment l'architecture orientée service (SOA) et un format pivot d'échange et de représentation de document multimédia (basé sur XML [6]). La plateforme s'appuie sur des outils open sources comme un portail Web (basé sur l'outil Liferay [7]) permettant d'accéder à l'application (IHM), ou encore l'ESB [8] Petals [9] pour orchestrer et faire collaborer, au moyen d'interfaces spécialisées, les différents services développés par les acteurs de ces projets (services de collecte de documents Web, d'extraction d'entités nommées, d'indexation, etc). Nous étudierons par la suite comment ces choix techniques retenus facilitent l'intégration et la combinaison de composants logiciels hétérogènes (COTS ou open source), permettant de mettre à disposition des utilisateurs des nombreuses fonctionnalités comme la collecte de site web, l'analyse sémantique de données, la reconnaissance d'entités nommées, la traduction, l'indexation de documents, etc.

3.2 Architecture et interfaces de services dans la plateforme WebLab

Pour ce faire, la plateforme propose une architecture SOA (voir la figure 1) dans laquelle les outils sont intégrés au travers d'un ensemble cohérent de briques logicielles. Chacune d'entre elles est exposée sous la forme de services Web normalisés par des interfaces spécialisées pour le traitement de documents multimédias décrites en WSDL [10] et accessibles par le protocole SOAP [11]. Ces services WebLab donnent accès, de façon indirecte, aux différentes fonctionnalités proposées par les COTS, les composants open source, ou les composants développés par EADS et ses partenaires. Par exemple, un composant de détection de la langue dans le texte comme Ngramj [12] ou un composant d'extraction d'entité nommée dans le texte comme GATE (Cunningham, 2002), se verra intégré dans un service implémentant l'interface WebLab *Analyser*. L'utilisation de ces interfaces rend possible l'ajout dans l'application, à tout moment, d'un nouveau service. Elles donnent également la possibilité d'intervertir directement un composant par un autre remplissant la même fonction (cas par exemple du passage d'un composant open source à un composant commercial plus performant).

Dans le même objectif de composabilité des fonctions, des services autonomes de présentation proposent des fragments d'IHM sous forme de portlets. Un portlet est ici un composant logiciel autonome et interactif de présentation de l'information, qui se charge de produire des fragments de code HTML, intégrables et gérés par un portail, jouant alors le rôle de conteneur et d'agrégateur de contenu. Pour synchroniser les fragments et échanger des informations entre portlets, le portail se charge également de distribuer des événements entre portlets. Ces événements permettent l'interaction des différents services d'affichage et donc la mise en

6. Extensible Markup Language
7. http ://www.liferay.com/
8. Enterprise Service Bus
9. http ://petals.ow2.org/
10. Web Services Language Description
11. Simple Object Acces Protocol
12. http ://ngramj.sourceforge.net/

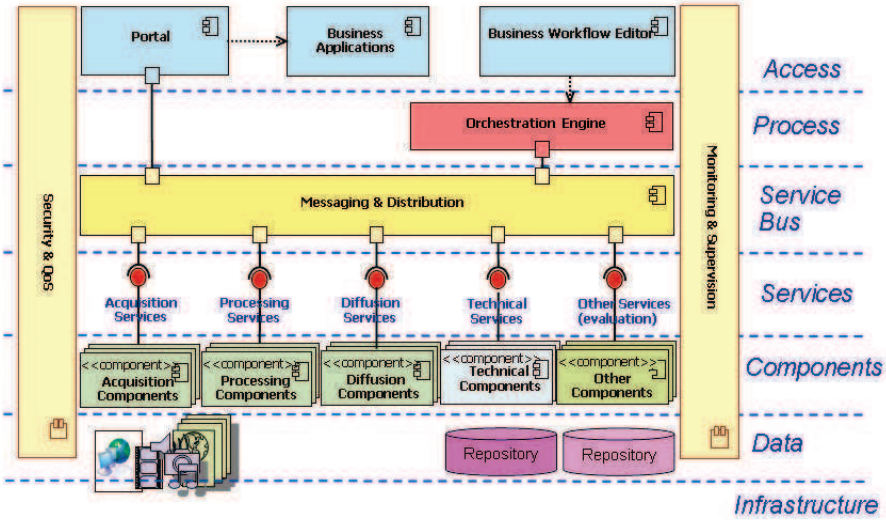


FIG. 1 – *Vue globale de l'architecture de la plateforme WebLab*

place d'interfaces à la fois complètes, cohérentes et flexibles. Cette technologie, développée par Sun, s'intègre aux plateformes d'entreprise J2EE [13] et constitue une solution pertinente pour la composition et la personnalisation d'IHM. Ainsi exploitée dans la plateforme WebLab, la technologie Portlet autorise l'intégration, dans une même application, d'IHM issues de logiciels variés. La personnalisation des pages du portail Web permet alors l'agrégation sur mesure des différents portlets et donc des fonctionnalités qu'ils proposent. Chaque portlet est dedié à la présentation d'une ou plusieurs fonctionnalités et exploite les services web du système au travers de leur interface normalisée. Ainsi, un même service d'affichage, comme par exemple un portlet de recherche de document plein texte est utilisable avec tous les services respectant l'interface WebLab *Searcher*.

3.3 Modèle d'échange

WebLab propose un modèle d'échange de données qui permet de représenter, sous une forme XML, tout type de document multimédia (texte, audio, vidéo, image, etc.) issu d'une source ouverte. Le choix du format XML apporte ici la flexibilité nécessaire à la représentation de tout type de document et suit les recommandations W3C [14]. Ce format d'échange est une solution aux problématiques liées à l'hétérogénéité des formats et des structures de données. En effet, les services de l'application traitent et s'échangent des documents normalisés dans un seul et même format pivot, en faisant abstraction de leurs formats d'origine. La transformation des données accessibles en sources ouvertes vers des documents au format pivot est

13. Java Platform, Enterprise Edition
14. World Wide Web Consortium

alors effectuée au moyen de services de normalisation, basés sur des outils existants comme par exemple Tika [15]. Les connaissances extraites sur ces documents par les outils (titre, auteur, contenu, langue, entités extraites, etc.) sont décrites sémantiquement, conformément à des ontologies de domaine (comme par exemple l'ontologie DublinCore [16] pour les métadonnées) et directement intégrées au document pivot au moyen d'annotations au format RDF/XML. Les documents produits sont ici enrichis sémantiquement au fur et à mesure de leur traitement par les différents services métier. Cette formalisation des connaissances dans un langage du Web sémantique au sein des documents permet d'exploiter de nombreuses fonctionnalités fournies par les outils de gestion de connaissances comme l'exploitation d'ontologies de domaine, l'interrogation des bases de données sémantiques, etc. De fait, la plateforme est tournée vers le Web sémantique où les données sont interprétables sémantiquement par des machines, améliorant leur traitement.

3.4 Gestion des connaissances

Dans le cadre d'une activité de veille assistée, le système d'aide peut s'appuyer sur des connaissances expertes pour améliorer le traitement des données. Dans le cas de la détection de signaux faibles, des connaissances utiles à la reconnaissance de menaces ou de situations redoutées sont exploitées, comme par exemple, les connaissances capitalisées au sein de scénarios ou de gabarits (Delavallade et al., 2007). Il s'agit ici de rassembler dans ces scénarios, des connaissances expertes permettant de décrire des situations redoutées et susceptibles de se réaliser dans l'environnement, en particulier les situations à faible probabilité de se réaliser dans le cas de recherche de signaux faibles. En effet, le veilleur tente dans ce cas de modéliser des situations *a priori* peu probables, ce qui l'amène à se projeter dans le futur et à exploiter ses connaissances tacites (intuition, point de vue) et stimule alors sa réflexion. Nous pensons que la création de tels scénarios au cours de séances de créativité, favorise alors la découverte de signaux faibles car il oblige le veilleur à formaliser ses intuitions, à imaginer des situations inconnues et à explorer de nouvelles connaissances et informations. C'est à l'issu de ces réflexions et en analysant les informations disponibles en rapport avec ces situations que l'expert est amené à repérer les signaux faibles. Dans le cas de la détection précoce de crises, ces connaissances peuvent se révéler complexes à capitaliser du fait de la complexité des phénomènes que l'on cherche à modéliser (comme par exemple des catastrophes naturelles ou des comportements sociaux). De plus, il s'agit également de décrire les corrélations ou les liens entre ces phénomènes qui peuvent conditionner la réalisation de ces crises (combinaison de plusieurs phénomènes ou événements). La construction des scénarios constitue alors une tâche complexe, qui peu se révéler coûteuse en terme de temps. Notons que la découverte de signaux faibles reste, dans tous les cas, une opération complexe et nécessite une activité de veille pérenne dans le temps. C'est pourquoi des outils dédiés, et inexistants à notre connaissance, capables d'aider le veilleur à modéliser et à actualiser facilement des scénarios sont aujourd'hui nécessaires. Cependant de tels outils restent complexes à mettre en place car il doivent facilement permettre la modélisation de situations complexes et variées tout en restant suffisamment flexibles pour autoriser des modifications rapides des scénarios, en réponse aux changements de l'environnement. La mise en place d'un tel outil, notamment par un port-

15. http ://tika.apache.org/
16. http ://www.dublincore.org/

let dédié à la construction graphique de scénarios permettrait de capitaliser ces connaissances expertes et permettrait de valider l'approche retenue. Ces connaissances, une fois intégrées dans le système, seront utilisées par des services chargés de sélectionner les informations les plus pertinentes et de détecter les menaces modélisées (services de reconnaissance d'événements, d'analyse de tendance, de suivi d'indicateurs, etc.). Lors de la construction des connaissances dans l'outil, seront exploitées des *ontologies* de domaine ou des ontologies plus larges comme DBpedia [17] ou Geonames [18] ou les ontologies accessibles via LinkedData [19] . Ces ontologies fournissent ici les référentiels sémantiques dans lesquels sont exprimées les différentes connaissances. Par exemple, pour la représentation d'un événement, les composants de l'événement (acteurs, lieux, type de l'événement, etc.) font référence à des concepts décrits dans différents référentiels sémantiques spécialisés et dans différentes langues. L'ontologie est ici l'objet de formalisation des connaissances, partagé entre les différents services du système, qui assure leur interopérabilité et un traitement cohérent des données. L'utilisation des ontologies conformément aux recommandations du W3C, c'est-à-dire en respectant les standards, permet de répondre à de nombreuses problématiques rencontrées dans la gestion des connaissances, que ce soit pour leur formalisation, leur accès ou leur interrogation. Ainsi, la formalisation des connaissances s'effectue conformément à des ontologies décrites avec le langage OWL [20]. Ce langage repose lui même sur le langage RDF où l'enregistrement d'une connaissance s'effectue au moyen d'un triplet <*Sujet, Prédicat, Objet*>. Les services de gestion des connaissances se chargent alors de stocker ces triplets au sein de base de connaissances (triple store) au moyen d'outils dédiés comme Sésame [21] ou Corèse [22]. Ces outils permettent ensuite d'accéder aux connaissances stockées par le biais du protocole de requêtage SPARQL [23]. Notons que les documents pivots dans la plateforme WebLab sont décrits avec des annotations au format RDF/XML ce qui rend leur traitement et leur stockage possible dans ces outils de gestion de connaissances. Seule la structure d'un document, initialement traduite par sa structure XML, reste à formaliser pour en obtenir une description sémantique complète. Un service dédié, s'appuyant sur les ontologies de la plateforme WebLab, réalise facilement cette transformation. Ainsi, une ressource WebLab entièrement décrite en RDF peut être stockée dans la base de connaissances et l'exécution de requêtes SPARQL complexes, faisant intervenir des règles d'inférence, devient possible. Il est par exemple facile, avec une requête SPARQL, de retrouver les documents contenant du texte dans lequel sont cités des organisations et des lieux décrits dans des scénarios de crise ou de menace. L'analyse manuelle des documents et la validation des extractions permet ensuite de traiter ces indices annonciateurs précoces de menaces pour en déduire un savoir. Le savoir constitué permet alors de consolider et de mettre à jours les connaissances liées à l'évolution de l'environnement dans le système. Ces connaissances sont à nouveau exploitées pour l'analyse. Ce cycle d'exploitation et de validation des connaissances permet de passer de la donnée Web à l'information clé permettant de prendre des décisions en réponse aux changements de l'environnement.

17. http ://dbpedia.org/
18. http ://www.geonames.org/
19. http ://linkeddata.org/
20. Ontology Web Language
21. http ://www.openrdf.org/
22. http ://www-sop.inria.fr/edelweiss/software/corese/
23. SPARQL Protocol and RDF Query Language

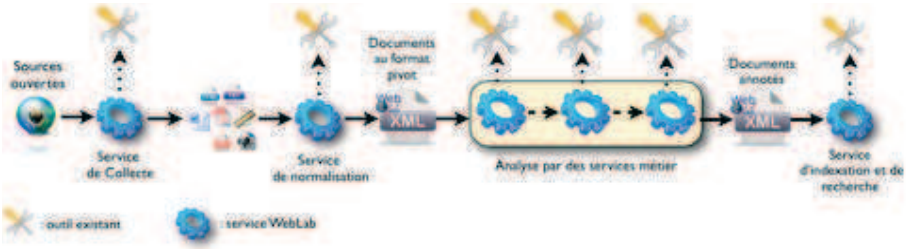


FIG. 2 – *Exemple d'une chaine de traitement dans la plateforme WebLab*

3.5 Orchestration de services

Pour automatiser la collecte et le traitement des ressources Web, la plateforme WebLab met à disposition des chaînes de traitement adaptées, exprimées en BPEL [24] et exécutées par le moteur d'orchestration de l'ESB (qui se charge d'appeler chaque service, de véhiculer les documents pivots de service en service, de répartir la charge entre les services, etc.). L'utilisation du langage BPEL pour la description des chaînes de traitement permet de mettre en place des chaîne de traitement sur mesure et adaptées à des besoins opérationnels précis. La figure 2 illustre le déroulement d'une chaîne de traitement générique au sein d'une application WebLab et détaillée comme suit : (1) un service de collecte (basé par exemple sur l'outil HTTrack [25]) est appelé pour collecter des ressources du Web dans différents formats (pages HTML, PDF, fichiers vidéo, etc.) ; (2) un service de normalisation est appelé pour transformer chacune de ces ressources collectées en un document au format pivot ; (3) chaque document normalisé est ensuite traité et enrichi sémantiquement (ajout d'annotations) par un ensemble de services d'analyse (détection de la langue, traduction, résumé, extraction d'événements, d'indicateurs, etc.) ; (4) enfin, chaque document traité est transmis à un service d'indexation plein texte (basé par exemple sur l'outil Lucene [26]) ou sémantique. Les document traités sont alors accessibles au sein du portail Web au moyen de portlets spécialisés (portlet de recherche et portlet de visualisation de document) offrant des vues dédiées à la présentation des documents et connaissances. Pour la détection des signaux faibles, des chaînes de traitement spécialisées sont mises en place pour focaliser l'extraction sur les entités et les événements modélisés dans les scénarios ainsi que pour collecter incrémentalement et perpétuellement des sources d'information pour suivre dans le temps l'évolution de tendances et des situations. Pour se faire des outils de présentation dédié à la projection temporelle des informations comme les projets SIMILE TimeLine [27] et TimePlot [28] sont intégrés dans des portlets dédiés.

24. Business Process Execution Language
25. http ://www.httrack.com/
26. http ://lucene.apache.org/
27. http ://www.simile-widgets.org/timeline/
28. http ://www.simile-widgets.org/timeplot/

3.6 Apports et limites de la plateforme WebLab pour la mise en place d'une application de veille

Ainsi, la plateforme apporte des solutions pour les problématiques énoncées précédemment :

L'hétérogénéité des formats et des structures de données : Grâce à l'utilisation du format de représentation de documents WebLab, les services de collecte et de normalisation produisent des documents formalisés dans un format pivot structuré, conformément à un modèle d'échange adapté au traitement des documents multimédia. Dès lors, le format ou la structure d'origine des documents collectés ne pose plus de problème pour les services d'analyse qui seront utilisés dans la suite du processus de traitement. Ces derniers traiteront l'ensemble des ressources collectées, indifféremment de leur format ou structure d'origine.

Le choix des outils et des technologies : les fonctionnalités implémentées par les différents outils sont exposées sous la forme de services web, ce qui permet d'abstraire, aussi bien les modèles de données internes des outils, que les technologies qu'ils exploitent. L'ensemble des fonctionnalités de ces outils sont disponibles par l'intermédiaire d'une seule et même technologie normalisée par des standards. Dès lors, l'ajout d'un nouvel outil ou l'exploitation d'une nouvelle technologie nécessite simplement le développement d'un adapteur exposant, sous forme d'un service WebLab, ses fonctionnalités.

L'interopérabilité des outils : la mise en place de chaînes de traitement de l'information spécifiques et adaptées, permettent l'exploitation conjointe des différents services et des fonctionnalités disponibles dans la plateforme WebLab. Ces chaînes de traitement permettent d'automatiser et d'orchestrer les appels aux différents services (services de collecte, services d'analyse, service d'indexation, etc). C'est la combinaison des différents services exploités au sein des processus de traitement, qui assure donc l'interopérabilité des différents outils.

La gestion et l'exploitation des connaissances : les ressources au format WebLab sont entièrement décrites avec les langages et les standards du Web sémantique ce qui assure leur compréhension et leur compatibilité avec les outils basés sur ces nouvelles technologies (gestionnaires de bases de connaissances et d'ontologies ou les moteurs d'inférence et raisonneurs).

L'ergonomie et le couplage d'IHM : La technologie Portlet est une solution aux problématiques de couplage et d'interopérabilité d'IHM. En effet, cette technologie permet de présenter dans une même application, au travers d'un portail Web, des interfaces en provenance des outils intégrés (présentation de l'IHM native de l'outil) ou d'interfaces spécifiques présentant des vues spécialisées (cartographie, projection temporelle, etc). L'ensemble des fonctionnalités de l'application est assuré par ces différents services de présentation qui peuvent être, comme pour les services métier, couplés entre eux pour s'adapter aux besoins de l'utilisateur. La coopération entre les services de présentation est rendue possible grâce à un mécanisme d'évènement proposé par la technologie Portlet. Enfin, les services de présentation exploitent les services métier au travers des interfaces de services proposés par la plateforme WebLab, autorisant l'utilisation d'un même service de présentation avec différents services métiers implémentant la même interface métier.

Le dimensionnement de l'application face aux volumes de données à traiter : des services de collecte d'information WebLab permettent la collecte automatique des informations disponibles, automatisant la tâche de collecte. De plus, la flexibilité apportée par l'architecture SOA, qui autorise l'ajout de nouveaux services permet le redimensionnement de l'application et la répartition géographique des calculs, des espaces de stockage ou des besoins réseau.

Le choix de la plateforme WebLab pour la mise en place d'une application d'aide à la détection de signaux faibles est ici particulièrement pertinent car les nombreux outils intégrés et impliqués dans les différentes phase de l'activité de veille deviennent facilement accessibles aux veilleurs. De plus, le fait que la plateforme s'appuie sur les standards du Web sémantique permet d'exploiter des outils plus intelligents de gestion de connaissances qui assurent un traitement plus efficace des données du Web et des connaissances expertes impliquées dans la détection de signaux faibles, notamment par le biais de scénarios. La plateforme se révèle suffisamment flexible pour envisager l'apport de nouvelles technologies et d'outils qui contribueront à maintenir une activité de veille efficace dans le temps. Cependant, de par la complexité de l'architecture et les nombreuses technologies mises en oeuvre, l'intégration de nouveaux composants dans une application de ce type reste une opération nécessitant de nombreuses compétences, notamment en développement et sur les technologies employées, qui reste aujourd'hui hors de portée des veilleurs. De plus, le choix d'utiliser certaines technologies peut créer une dépendance ou limiter l'intégration de certains composants. C'est, par exemple, le cas de la technologie portlet qui permet l'intégration facile des interfaces Web basées sur HTML mais qui ne se prête pas à l'intégration de clients lourds dans le portail. Enfin, notons que certaines technologies récentes utilisées dans la plateforme sont encore insuffisamment outillées pour en permettre une utilisation simple, comme par exemple pour la construction des chaînes de traitement grâce au langage BPEL.

4 Conclusion

Au cours de cet article, nous avons présenté les problématiques rencontrées lors de la conception et la mise en place des systèmes d'aide à la veille en sources ouvertes, ainsi que les problématiques des veilleurs face à ces applications complexes et les nombreux outils existants. L'exemple de la détection de signaux faibles dans ces sources a mis en évidence le besoin de gérer au sein de l'application, un ensemble de connaissances qu'il convient alors de formaliser de façon à permettre leur exploitation et leur compréhension par le système. Pour répondre à ces problématiques, nous avons présenté une plateforme d'intégration open source, adaptée à la construction d'application de veille. Nous avons également montré la flexibilité de cette plateforme par rapport aux nombreux formats ou structures de données existants. De même, nous avons montré que cette plateforme est particulièrement flexible et adaptée pour l'intégration des outils existants de fouille de données et pour la construction d'IHM riches et personnalisables, tout en soulignant les difficultés rencontrées lors du développement de nouveaux composants et notamment liées au nombre de technologies employées. Enfin, nous avons montré comment exploiter conjointement, les outils du Web sémantique, les documents de la plateforme et des connaissances métier formalisées conformément aux standards et recommandations du W3C.

Références

Ansoff, H. I. (1976). Managing surprise and discontinuity - strategic response to weak signals. *Zeitschrift für Betriebswirtschaft 28*, 129–152.

Cao, T. (2006). *Exploitation du web sémantique pour la veille technologique*. Ph. D. thesis, Université de Nice-Sophia Antipolis.

Cunningham, H. (2002). Gate : A framework and graphical development environment for robust nlp tools and applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Delavallade, T., L. Mouillet, B. Bouchon-Meunier, et E. Collain (2007). Monitoring Event Flows and Modelling Scenarios for Crisis Prediction : Application to Ethnic Conflicts Forecasting. *World 15*, 83–110.

Dieng-Kuntz, R., O. Corby, F. Gandon, A. Giboin, J. Golebiowska, N. Matta, et M. Ribiere (2001). Méthodes et outils pour la gestion des connaissances : une approche pluridisciplinaire du knowledge management.

Giroux, P. et al. (2008). Weblab : An integration infrastructure to ease the development of multimedia processing applications. *ICSSEA*.

Lesca, H. et M. Schuler (1995). Veille Stratégique : Comment ne pas être noyé sous les informations ? In *Actes du Colloque : VSST'95*.

Lesca, N. (2002). *Construction du sens : le cas de la veille stratégique et de l'exploitation collective des signes d'alerte précoce*. Ph. D. thesis, CERAG.

Summary

Business intelligence and *weak signal* detection from information collected on open sources become more and more complex. Due to Web 2.0 and the multiplication of sources of information as blogs or social networks, watchers have to deal with several data in different formats and structures. Consequently, an adapted system, helpfull for business intelligence activities, able to retrieve and process data from the Web, is required. This system also has to manage the knowledge implicated in weak signals detection. In this article, we will focus on solutions given by an open sources integration plateform, dedicated to the multimedia data mining for buisiness intelligence applications. We will put forward solutions to slove problems encountered in interoprability between information processing or knowledge managment tools. We will explain how this plateforme will help us in the making of ergonomic and personnalisable applications. Finally, we will study the importance of the architecture, technical solutions, limitations and semantic Web standards in this kind of application.

Extraction d'information à partir des SMS

Najeh HAJLAOUI

Université de Grenoble, Laboratoire LIG
Najeh.Hajlaoui@imag.fr

Résumé. Dans le cadre de nos travaux sur la multilinguïsation, ou « portage linguistique » des services de gestion de contenu traitant des énoncés spontanés en langue naturelle, souvent bruités, mais contraints par la situation, et constituant un « sous-langage » plus ou moins restreint, nous avons dégagé trois méthodes de portage possibles d'une langue L1 vers une nouvelle langue L2. Nous les avons appliquées sur des cas de systèmes de e-commerce déployés comme CATS "*Classified Ads Through SMS*". C'est une application de e-commerce déployée en Jordanie sur le réseau FastLink. Elle traite des petites annonces envoyées par SMS et concernant l'occasion automobile (Cars), l'immobilier à Amman (RealEstate), l'emploi (Jobs). Le portage par adaptation de l'extracteur de contenu de CATS, est une de ces trois méthodes et a donné de très bonnes performances, et cela avec une légère modification de la partie grammaticale, ce qui prouve que, malgré la grande distance entre l'arabe et le français, ces deux sous-langages sont très proches l'un de l'autres, une nouvelle illustration de l'analyse de R. Kittredge.
Mots clés : énoncés spontanés, SMS, extraction d'information, sous-langage**.**

1 Introduction

Une conséquence majeure résultant de la mondialisation que nous vivons aujourd'hui est l'importance croissante du multilinguisme, accentuée par l'utilisation d'Internet et le multimédia. Face à ce phénomène, les développeurs d'applications informatiques ne peuvent pas encore fournir des applications utilisables dans la langue maternelle de tous les utilisateurs. Les services et les applications sont rarement disponibles dans de nombreuses langues, et peuvent encore moins traiter des textes en langage naturel rédigés spontanément et pouvant contenir des erreurs, des abréviations, différentes formes typographiques, etc.

Dans ce cadre, nous nous intéressons à un problème plus précis, celui de la multilinguïsation des applications de e-commerce traitant des énoncés spontanés en Langue Naturelle (LN). Ces applications extraient le « contenu pertinent » des énoncés en langue naturelle, les représentent dans un langage approprié appelé "Content Representation Language" (CRL), et traitent ensuite les objets obtenus.

2 Systèmes traitant des sous-langages

L'extraction de contenu est rarement fondée sur une analyse complète des énoncés : on utilise le plus souvent des « grammaires locales » et un dictionnaire mettant en relation des termes du domaine et les symboles (concepts, attributs, relations) du CRL. La

multilinguïsation de tels systèmes est en fait un problème difficile, ce qui explique que très peu de services soient multilingualisés.

La multilinguïsation ou le « portage linguistique » dont nous parlons ici n'est pas nécessairement une « localisation » Hajlaoui (2008). Une localisation implique une adaptation à un autre contexte en tenant compte de tous les aspects linguistiques, culturels, géographiques. Par contre, un portage linguistique doit seulement permettre l'accès dans une autre langue à un service de e-commerce, tel qu'il est et où il est (en restant dans le même contexte).

Un service de gestion de contenu utilise une représentation interne spécifique sur laquelle travaille le noyau fonctionnel[1]. Le plus souvent, cette représentation est produite à partir de la langue « native » L1 par un extracteur de contenu. Nous avons dégagé trois approches de portage possibles, et les avons illustrées par le portage en français d'une partie de CATS Daoud (2006), un système de traitement de petites annonces en SMS (en arabe) déployé à Amman, ainsi que sur IMRS "*Impression-based Music-Retrieval System*" Kumamoto (2007), un système de recherche de morceaux de musique dont l'interface native est en japonais et dont seule la représentation de contenu est accessible.

Généralement, les énoncés traités par le genre d'applications auxquelles on s'intéresse constituent un sous-langage plus ou moins restreint. Nous présentons alors une brève étude sur cette notion, une notion sur laquelle plusieurs chercheurs ont travaillé et ont divergé quant à sa définition et son identification. Un sous-langage est-il vraiment un sous-ensemble de la langue standard ?

Kittredge et d'autres ont montré l'importance de la notion de sous-langage dans le traitement des textes d'un langage naturel modifié ou simplifié par l'utilisation de restrictions lexicales, syntaxiques ou sémantiques spécifiques Kittredge and Lehrberger 1982), Grishman and Kittredge (1986), Slocum (1986), Biber (1993), Sekine (1994).

Plusieurs définitions des sous-langages ont été données, la première a été proposée par Harris (1968). Une autre définition a été proposée par Bross et al. (1972) et a été utilisée dans le projet TAUM-METEO (1972-1973) Chandioux (1988), puis pour des manuels de maintenance d'avions dans le cadre du projet TAUM-AVIATION (1974-1981) et du PN-TAO (Projet National de TAO, 1982-87) en France. Grishman *et al.* (1986) et Deville (1989) définissent aussi un sous-langage comme étant une forme spécialisée d'une langue naturelle employée dans un domaine ou un thème particulier.

Nous appellerons « langue standard » l'ensemble des énoncés d'une communauté linguistique formés d'une façon « correcte » par rapport à la grammaire et au vocabulaire usuels. Nous appellerons « langue générale » l'union d'une langue standard et de toutes ses variantes (jargons[2], langues de spécialité, parlers régionaux, langages « techniques », et langages « sécrétés » par des contextes socioprofessionnels).

3 Portage par adaptation interne

Si on a un accès au code de l'application, nous pouvons appliquer une première approche nommée portage « interne » Hajlaoui et Boitet (2007). Comme le montre la FIG. 1, elle consiste à adapter à L2 l'extracteur de contenu de l'application. Cette approche nécessite un corpus et un dictionnaire fonctionnellement équivalents dans la langue d'arrivée (L2).

---

[1] Par exemple le noyau fonctionnel dans CATS est un ensemble de programmes autour d'une base de données.

[2] « jargon » désigne un parler propre aux représentants d'une profession ou d'une activité.

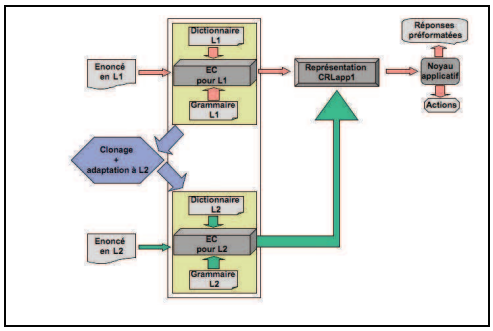


FIG. 1 –*Méthode de portage interne*

3.1 Extracteur de contenu pour les SMS en arabe

L'outil EnCo est un LSPL[3] développé dans le cadre du projet UNL « Universal Networking Language » Uchida et Zhu (2005-2006) pour écrire des « enconvertisseurs » vers le langage pivot UNL. EnCo a été utilisé dans CATS pour produire une représentation syntaxiquement semblable à UNL appellée CRL-CATS.

EnCo attend en entrée :

- Un dictionnaire et une grammaire (linguiciel).
- Un texte découpé en phrases.
- Éventuellement une « base de connaissances »

Le linguiciel est compilé, puis chaque phrase est traitée successivement. Les structures de données manipulées par EnCo sont :

- Une liste de nœuds avec deux têtes de lecture/écriture placées sur deux nœuds successifs (LAW « Left Analysis Windows », RAW « Right Analysis Windows ») et deux têtes de lecture (LCW « Left Conditions Windows », RCW « Right Conditions Windows ») pour les contextes gauche et droit.
- Un graphe de nœuds, initialement vide, pouvant contenir des nœuds de la liste, et dont les arcs portent des « relations » identifiées par des symboles à trois caractères alphabétiques.

Au départ, la liste comporte trois nœuds : la limite gauche, le nœud courant et la limite droite. Le nœud courant contient comme chaîne la phrase à traiter.

De façon générale, un nœud peut contenir quatre éléments : une chaîne, un ensemble d'attributs « de chaîne » (initialisés lors des appels au dictionnaire), une UW "Universal Word" (référence lexicale, venant du dictionnaire ou créée par une règle), et un ensemble d'attributs « de graphe » (préfixés par « @ »). Les attributs sont booléens, et ne sont pas déclarés. Seul « *@entry* » a un rôle spécial.

La FIG. 2 montre la structure d'EnCo. EnCo utilise les fenêtres de condition pour tester si les nœuds voisins des deux côtés des fenêtres d'analyse remplissent les conditions pour l'application d'une règle d'analyse. Les fenêtres d'analyse sont utilisées pour vérifier l'existence de deux nœuds adjacents afin d'appliquer une règle d'analyse. S'il existe une règle applicable aux nœuds courants, EnCo ajoute ou supprime pour ces derniers nœuds les propriétés grammaticales indiquées par la règle appliquée et/ou insère un nouveau nœud dans le graphe, ainsi qu'une relation sémantique, selon le type de la règle.

[3] Langage Spécialisé pour la Programmation Linguistique

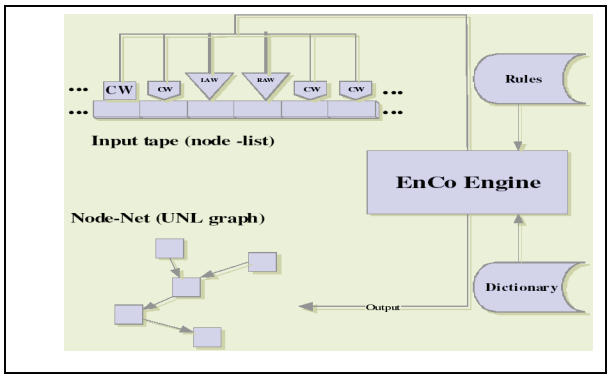


FIG. 2 – *Fonctionnement d'EnCo*

3.1.1 Règles

La syntaxe des règles d'EnCo est la suivante Uchida and Zhu (2005-2006):

```
<TYPE>...(<PRE2>)(<PRE1>){<LNODE>}  {<RNODE>}  (<SUF1>)  (<SUF2>)…
P<PRI>;
avec
<LNODE>:="{“ [<COND1>] ":" [<ACTION1>] ":" [<RELATION1>]":" [<ROLE1>]
"}"
<RNODE>:="{“ [<COND2>] ":" [<ACTION2>] ":" [<RELATION2>]":" [<ROLE2>]
"}"
```

FIG. 3 – *Syntaxe d'une règle en EnCo*

Notons que, pour une règle d'analyse donnée, il est possible d'insérer (respectivement de supprimer) un seul nœud dans (respectivement à partir de) la liste des nœuds. Ici, LNODE fait référence au nœud sous la fenêtre d'analyse gauche (LAW) et RNODE fait référence au nœud sous la fenêtre d'analyse droite (RAW).

L'interprétation générale de la syntaxe d'une règle en EnCo est alors la suivante.

Une règle peut s'appliquer si sous la fenêtre d'analyse gauche (LAW) se trouve un nœud qui satisfait la condition <COND1> et sous la fenêtre d'analyse droite (RAW) se trouve un nœud qui satisfait la condition <COND2>. Quand il y a des nœuds qui remplissent les conditions trouvées dans <PRE1> et <SUF1> du côté gauche (respectivement dans <PRE2> et <SUF2> du côté droit) des fenêtres d'analyse, les propriétés grammaticales dans les fenêtres d'analyse sont réécrites selon les actions <ACTION1> (respectivement <ACTION2>).

- Le champ <TYPE> porte le type de la règle à appliquer, qui indique l'opération qui doit être effectuée dans la liste des nœuds (par exemple une opération d'insertion, de suppression, etc.).
- <COND1> et <COND2> sont des conjonctions de conditions élémentaires testant la présence (ATTR) ou l'absence (^ATTR) de certaines attributs grammaticaux.
- <ACTION1> et <ACTION2> contiennent des attributs grammaticaux à insérer ou à supprimer dans des nœuds sous les fenêtres d'analyse.
- Les champs <RELATION1> et <RELATION2> sont utilisés pour créer des relations UNL entre les nœuds sous les fenêtres d'analyse.
- <ROLE1> et <ROLE2> sont des attributs de la base de connaissances qui sont optionnels et qui ne sont pas utilisés dans CATS.

- <PRI> indique la valeur de priorité de la règle, qui doit être comprise entre 0 et 255. Une règle dont la valeur de priorité n'est pas indiquée est considérée comme une règle de priorité 0.

Les 710 règles utilisées dans le système CATS réalisent l'extraction des informations utiles, et non pas l'analyse linguistique au sens classique. Elles affectent des valeurs à des objets préfinis dans le dictionnaire pour construire des relations semblables au type *Propriété{objet, valeur}* en les représentant dans le graphe construit par *objet --propriété→ valeur*. L'ensemble de ces relations forme la représentation CRL-CATS.

3.1.2 Dictionnaire

EnCo recherche dans le dictionnaire tous les lexèmes candidats à partir du premier caractère, il leur affecte ensuite des priorités basées sur la fréquence d'apparition et la longueur du lexème comme suit :

D'abord, il accorde la priorité la plus élevée à l'entrée qui possède la fréquence d'apparition la plus élevée. Ensuite, en cas de plusieurs candidats (qui ont la même fréquence la plus élevée), il choisit le lexème le plus long.

En cas d'échec ultérieur, ou d'activation directe d'un retour arrière, le système effectue un retour arrière (backtrack) et le lexème suivant dans la liste est alors découpé, puis le traitement reprend sur cette nouvelle « branche » du calcul. L'extracteur de contenu de CATS est construit pour utiliser cette possibilité le plus rarement possible.

Les entrées du dictionnaire ont la syntaxe de la FIG. 4. Pour un blanc, et dans le code ASCII, EnCo crée automatiquement l'entrée suivante dont l'UW est la chaîne vide. `[ ]{} "" (BLK) <., 0, 0>;`

```
[HW]"UW" (ATTR1, ATTR2, …) <FLG, FRE, PRI>;
HW : (Head Word) est un lexème qui peut être un candidat
UW : (Universal Word) est une référence lexicale
ATTR1, ATTR2 sont des attributs
FLG : indique la langue
FRE : indique la fréquence
PRI : indique la priorité
```

FIG. 4 – *Syntaxe d'une entrée dictionnairique*

Le dictionnaire utilisé dans la version arabe et pour l'ensemble des domaines (automobile, immobilier, divers…) a environ 10.000 CW et 30.000 lexèmes, dont 20.000 ont été générés automatiquement en appliquant différentes transformations typographiques.

3.2 Adaptation de l'extracteur de contenu

Pour porter l'extracteur de contenu vers le français, nous avons effectué essentiellement un travail dictionnairique et un travail de modification de règles que nous illustrons par un exemple. L'architecture générale reste la même.

3.2.1 Exemple de traitement

Afin de mieux comprendre le fonctionnement de l'outil EnCo, nous en montrons le détail sur un petit exemple de SMS en français « *recherche voiture* ».

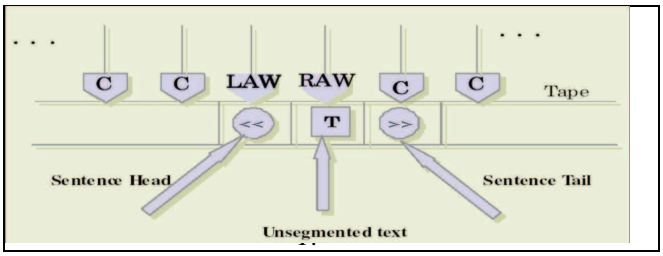


FIG. 5 – *Configuration initiale d'EnCo*

Comme le montre la FIG. 5, initialement la fenêtre d'analyse gauche (LAW) contient le symbole << appelé SHEAD et la fenêtre d'analyse droite (RAW) contient le texte non segmenté, c'est-à-dire « `recherche voiture` ». EnCo consulte alors le dictionnaire pour chercher les préfixes de cette chaîne. Un seul article de dictionnaire est trouvé : `[recherche]{} "wanted" (want) <F,1,1>;`

La FIG. 6 montre le résultat de segmentation de « *recherche voiture* ». RAW pointe sur un nœud contenant la chaîne « *recherche* », l'UW « *wanted* », et l'attribut dictionnairique « *want* », et aucun attribut de graphe. Par suite de la segmentation lexicale, le nœud initial a été divisé en deux nœuds, le deuxième contenant seulement la chaîne restant à analyser `[voiture]`. À ce niveau, EnCo cherche à trouver une règle qui satisfait la configuration actuelle : LAW pointe sur le symbole << et RAW pointe sur le nœud `[recherche]{}"wanted"(want)`.

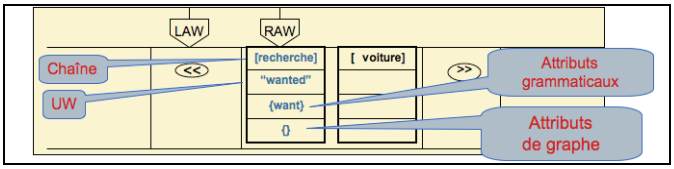


FIG. 6 – *Segmentation de « recherche voiture »*

La première règle qui peut être appliquée est : `R{SHEAD:::}{want:::}P20;` Cette règle fait un "*shift right*" désigné par « R » (TYPE = R). P20 indique la priorité affectée à cette règle. Elle est appliquée sans aucune condition (car COND1 et COND2 sont vides) et sans aucun changement de propriétés grammaticales (car ACTION1 et ACTION2 sont vides). Comme le montre la FIG. 7, après l'application de cette règle, LAW pointe sur « *recherche* » et RAW pointe sur un blanc « BLK ». En effet, le "*shift* " provoque aussi la segmentation d'un préfixe, ici le blanc.

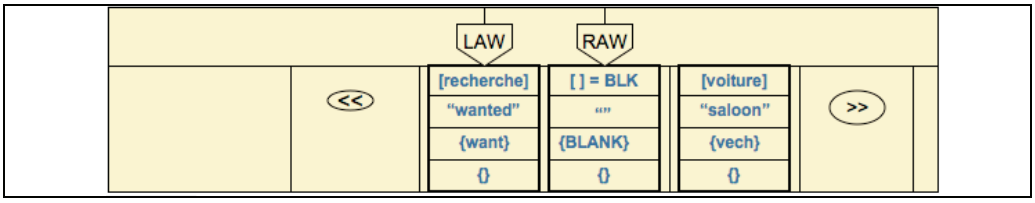


FIG. 7 – *Configuration d'EnCo après un shift right*

A ce moment, la première règle applicable est : `+{:+BLANK::}{BLK:::}P255;`

- Type de l'opération = + qui signifie la combinaison du nœud droit avec le nœud gauche (autrement dit, ajouter l'attribut de chaîne `BLANK` au nœud à gauche).

- `Cond1` = rien
- `Cond2 = BLK` (présence de l'attribut BLK)
- `Action1 = +BLANK` (ajout du symbole `BLANK` à la liste des attributs du nœud à gauche).
- `Action2` = rien
- `P255 = Priorité 255` (élevée)

Suite à l'exécution de cette règle, LAW pointe sur le symbole SHEAD et RAW pointe sur le nœud « recherche ». EnCo fait ensuite un *shift right* en appliquant la règle suivante : `R{:::}{:::}()P1;`

La FIG. 8 montre l'état d'EnCo après l'exécution des deux précédentes règles. LAW pointe sur le nœud contenant la chaîne *[recherche]* et RAW pointe sur le dernier nœud, contenant (après segmentation) les informations du dictionnaire sur la chaîne *[voiture]*. EnCo n'a pas créé de nouveau nœud car la chaîne restante est vide.

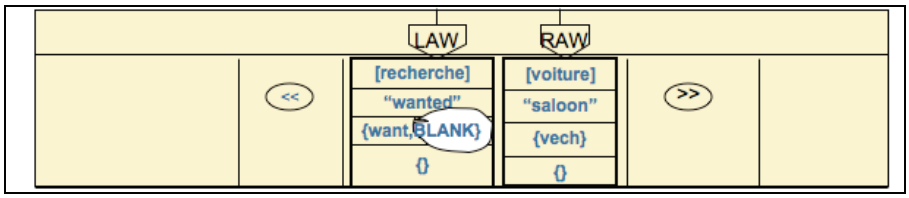


FIG. 8 – *Suppression du BLANK et shift right*

À ce niveau, nous avons besoin d'une règle qui crée, dans le graphe des nœuds, une relation [wan] qui part du nœud contenant `[voiture]` et arrive au nœud contenant `[recherche]`, puisque LAW pointe sur un nœud contenant {`want, BLANK`} et `RAW` pointe sur un `vech`. Cette règle est la suivante :

```
>{want,BLANK:-want:wan:}{vech,^want_add:want_add::}()P70;
```

- Type d'opération = modification à droite désignée par > (le nœud gauche est supprimé de la liste des nœuds, le nœud à droite devient l'origine de l'arc inséré, portant le symbole `wan`, et allant vers l'ancien nœud gauche.
- `Cond1 = want, BLANK` (le nœud sous LAW doit contenir l'attribut `want` et un attribut `BLANK`).
- `Cond2 = vech, ^want_add` (le nœud sous RAW doit contenir l'attribut *vech* et ne doit pas contenir l'attribut `want_add`).
- `Action1 = -want` (supprimer l'attribut `want` au nœud gauche (placé sous LAW)).

`Action2 = want_add` (ajouter l'attribut `want_add` au nœud droit (placé sous RAW)).

- `RELATION1 = wan` (la relation `wan` va du nœud droit au nœud gauche).

La FIG. 9 montre le résultat de l'application de cette dernière règle.

Le résultat final produit par EnCo est :

```
;========== UNL ================
;recherche voiture.
[S]
wan(saloon:0A, wanted:00)
[/S]
   ;================================
```

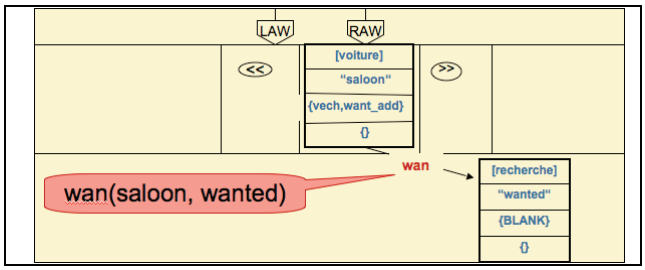


FIG. 9 – *Création d'une relation dans le graphe des nœuds*

3.2.2 Modifications apportées

La bonne surprise de ce travail est que nous n'avons dû modifier que légèrement les règles fabriquées initialement pour la version arabe, et que l'extracteur de contenu obtenu fonctionne bien pour le sous-langage correspondant du français, celui des SMS spontanés pour l'achat et la vente des voitures d'occasion.

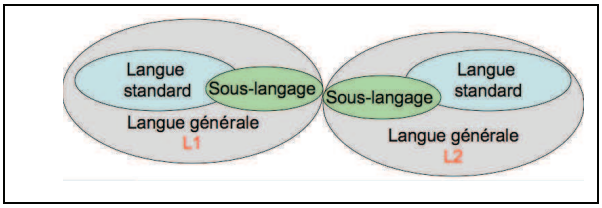


FIG. 10 – *Les sous-langages sont proches*

Comme le montre la FIG. 10, cela confirme les théories linguistiques Kittredge and Lehrberger (1982) selon lesquelles deux sous-langages équivalents dans deux langues différentes sont proches (très proches ici) l'un de l'autre, même si leurs deux langues mères sont éloignées.

Par portage interne, la partie grammaticale a été très faiblement modifiée, ce qui prouve que, malgré la grande distance entre l'arabe et le français, ces deux sous-langages sont très proches l'un de l'autres, une nouvelle illustration de l'analyse de R. Kittredge. TAB. 1 montre la répartition de l'effort pour le portage interne en terme de temps de travail et de pourcentage du code modifié ou ajouté.

Adaptation de EC-CATS	**Dictionnaire**	**Règles**
Temps de travail (H)	100	45
% du code modifié	90	5

TAB. 1 – *Répartition de l'effort pour le portage interne*

3.3 Évaluation de l'extraction d'information

Après avoir porté CATS de l'arabe vers le français, nous avons effectué une évaluation au niveau de la représentation interne. Nous avons utilisé deux mesures : une mesure informationnelle basée sur un corpus de 200 SMS et une deuxième mesure plus fine basée sur l'ensemble des énoncés constituant un corpus plus gros de 1100 SMS.

Nous avons traduit manuellement le corpus d'évaluation utilisé pour l'évaluation de la version arabe (originale) du système. C'est un corpus constitué de 200 SMS réels (100 SMS

d'achat + 100 SMS de vente) envoyés par des utilisateurs réels en Jordanie. Nous avons mis 289 mn pour traduire les 200 SMS arabes (2082 mots, soit environ 10 mots/SMS, ou 8 pages standard[4]) de l'arabe vers une traduction "brute" (littérale), soit 35 mn par page. Nous avons obtenu 200 SMS français jugés fonctionnels (1361 mots, soit 6,8 mots/SMS, environ 5 pages standard). Afin de comparer les résultats obtenus par rapport à une version de référence « CRL-CATS-REF », nous avons corrigé manuellement les 200 CRL-CATS correspondant au corpus d'évaluation. Les erreurs rencontrées concernent essentiellement les propriétés dont les valeurs sont des nombres comme « prix » et « année ».

Pour évaluer les résultats d'extraction, nous avons calculé le rappel R, la précision P et la F-mesure F pour chacune des propriétés les plus importantes (*action de vente ou d'achat, marque, modèle, année, prix*).

	Par rapport à la version originale			**Par rapport à la version de référence**		
Portage	**Minimum**	**Moyenne**	**Maximum**	**Minimum**	**Moyenne**	**Maximum**
Interne	95%	98%	100%	83%	91%	99%

TAB. 2 – *Récapitulatif des résultats d'évaluation informationnelle*

TAB. 2 récapitule les pourcentages de portage, calculés par cette évaluation informationnelle sur le petit corpus « CorpusEvalFr200SMS ». Les pourcentages de portage (rapport des F-mesures) par portage interne (par rapport à la version originale) varient entre 95% et 100%, avec une moyenne de 98 %.

Dans le but d'effectuer une évaluation sur la totalité du corpus de 1100 SMS nommé « CorpusEvalFr1100SMS », nous avons utilisé une deuxième mesure automatique, et plus fine que la précédente. Dans cette mesure, nous allons au-delà de la détection des noms des attributs, et comparons leurs valeurs par rapport à celles d'une version de référence. Nous supposons en pratique que la version originale produite par le système est une version de référence. Ainsi, nous calculons la distance entre la version de référence et la version candidate résultant d'une méthode de portage. Nous avons utilisé l'algorithme de calcul de distance d'édition entre arbres de Selkow (1977) par laquelle nous passons à un calcul des mesures connues (rappel R, précision P, et F-mesure F). Nous avons obtenu ensuite une moyenne des F-mesures trouvées égale à 0,72, ce qui permet toujours de déployer la nouvelle version du système obtenue.

4 Conclusion

Nous avons présenté le détail d'une approche de portage linguistique des applications de traitant des énoncés spontanés d'un sous-langage dans laquelle nous avons modifié que 5% des règles pour passer du sous-langage de l'arabe vers le français. C'est une autre confirmation de l'hypothèse de Kittredge sur les sous-langages selon laquelle deux sous-langages qui se correspondent dans deux langues différentes sont très proches entre eux, et souvent plus proches entre eux qu'ils ne le sont chacun de leur langue-mère respective, ce qui permet de les considérer et de les traiter comme des variantes l'un de l'autre.

[4] Une page standard contient 250 mots.

5 Bibliographie

Biber D. (1993). « Using register-diversified corpora for general language studies ». in *Computational Intelligence*. Vol. 19(2): pp. 219-241.

Bross ID., Shapiro PA., Anderson BB. (1972). « How information is carried in scientific sub-languages ». In *science*. Vol. 176(4041): pp. 1303-1307. juin 1972.

Chandioux, J. (1988). « 10 ans de METEO (MD). Traduction Assistée par Ordinateur ». *Actes du séminaire international sur la TAO et dossiers complémentaires*. Paris Observatoire Francophone des Industries de la Langue (OFIL). A. Abbou. pp. 169-173.

Daoud, D. M. (2006). It is necessary and possible to build (multilingual) NL-based restricted e-commerce systems with mixed sublanguage and contend-oriented methods. Thèse. Université Joseph Fourier. Grenoble, France. September 23, 2006. 296 p.

Deville, G. (1989). Modelization of task-oriented Utterances in a Man-Machine Dialogue System. Thèse. University of Antwerpen. Belgique. 200 p.

Doddington, G. (2002) Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. Proc. HLT 2002. San Diego, California. March 24-27, 2002. vol. 1/1: pp. 128-132 (note book proceedings).

Grishman, R., R. Kittredge.(1986). *Analyzing language in restricted domains*. Hillsdale NJ. Lawrence Erlbaum Associates. 248 p.

Hajlaoui, N. (2008) Multilinguïsation de systèmes de e-commerce traitant des énoncés spontanés en langue naturelle. Thèse. Université Joseph Fourier. Grenoble. 25 septembre 2008. 318 p.

Hajlaoui, N., C. Boitet (2007). « Portage linguistique d'applications de gestion de contenu ». Proc. *TOTh 2007 Conférence sur la Terminologie & Ontologie : Théories et Applications*. Annecy, France. 1 juin 2007. 13 p.

Hajlaoui, N., D. M. Daoud, C. Boitet (2008). « Methods for porting NL-based restricted e-commerce systems into other languages ». Proc. *LREC 2008*. Marrakech, Maroc. june 26-31, 2008. 7 p.

Harris, Z. (1968). « Mathematical structures of language ». in *The Mathematical Gazette*. Vol. 54(388): pp. 173-174. May, 1970.

Kittredge, R., J. Lehrberger.(1982). *Sublanguage - Studies of language in restricted semantic domain*. Walter de Gruyter. Berlin / New York.

Koehn, P. (2004). « Pharaoh: a Beam Search Decoder for Phrase-Based SMT ». Proc. *6th AMTA*. Washington, U.S.A. pp. 115-124.

Kumamoto, T. (2007). « A Natural Language Dialogue System for Impression-based Music-Retrieval ». Proc. *CICLING-07 (Computational Linguistics and Intelligent Text Processing)*. Mexico. February 12-24, 2007. 12 p.

Sekine, S. (1994). « A new direction for sublanguage NLP ». Proc. *International Conference on New Methods in Language Processing*. Manchester, England. 8 p.

Selkow, S. M. (1977). « The tree-to-tree editing problem ». in *Information Processing Letters*. Vol. 6: pp. 184-186. December, 1977.

Slocum, J. (1986). *How one might automatically identify and adapt to a sublanguage*. in Analyzing language in restricted domains. pp. 195-210.

Stolcke, A. (2002). « SRILM - an Extensible Language Modeling Toolkit ». Proc. *ICSLP*. Vol. 2: pp. 901-904. Denver. USA.

Uchida, H., M. Zhu, (2005-2006). *Universal Networking Language* 10 2-8399-0128-5. 218 p.