Business Opportunity Detection in the Big Data

Lyes Limam, Jean Lecouffe and Stéphane Chau
Altran Research, Région Sud Est, Division IIS, Altran, 1 Place Verrazzano, Lyon, France

Keywords: Big Data, Business Intelligence, Text Mining, Graph Databases.

Abstract: Modern enterprise information systems are characterized by large amounts of data issued from various internal and external business applications, often stored and archived in different supports (databases, documents, etc. The nature of this data (voluminous, unstructured, heterogeneous, inconsistent, etc.) makes them difficult to use for analysis. In fact, it is typically an issue of big data analytics.
The main objective of our research project is to design a solution to detect opportunities (projects, new markets, skills, tenders, etc.) in the continually growing data, while adapting to its constraints. The extracted information should help users to take proactive actions to improve their business (e.g., identify a consultant skill that can be aligned with a given tender).
In this project we are interested in text data. There are two main reasons. The first reason is that text data is the most difficult to analyse by humans, especially when it is voluminous. The second reason is that we are convinced that valuable information is usually textual. Therefore, we define six research axes:

- Intelligent Information Sensing
- Text Mining
- Knowledge Representation (semantics)
- Querying the knowledge
- Results Interpretation
- Self-learning.

1 INTRODUCTION

Big Data mining is a recent and actual research trend (Diebold, 2012). Several approaches were experimented in several domains like: mobile communications (Laurila, et al., 2012), biology (Howe, et al. 2008), economics (World Economic Forum, 2012), (Letouzé, 2012), marketing (Fan, et al., 2015), decision making (Probst, et al., 2013), etc.

In Big Data mining, it is usual to deal with 3V problems: Volume, Variety, and Velocity. Recently, two more Vs were proposed: Variability and Value (Fan and Bifet, 2012). This last V is very important in a business-oriented mining, which has for objective to valuate internal and/or external data through mining.

In our case, the values we want to highlight are:

- Fast answer to customer's requests
- Understand customer's problems and determine new proposals to help solving it
- Find new business opportunities through new projects, new markets, skills, tenders, etc.

Some papers propose to use graph-oriented databases (Lin, 2014), which allow much faster responses than classical relational databases, due to local dependencies of data. A particular model of graph based on RDF (Resource Description Framework) retained our attention. RDF based model consists of "triples" [subject] -> [predicate] -> [object], which define conceptually a labelled graph (Bönström, 2003). That allows representing data dependencies in a clear, simple and efficient way, and allows fast access to data in graph-oriented databases.

One of the main problems is: how to build this graph? I.e. what are the data sources and how is extracted pertinent information. Some documents are relatively structured, like competencies records, but the most are non-structured and thus need to be threated specifically. Research axes to answer this question are crawling and text mining.

Another problem leads in graphs gathering: the objective is to build a global graph by combination of graphs extracted from documents. This problem is not trivial, as simply gathering graphs on their common nodes and links could lead to mistakes, mainly false

Limam, L., Lecouffe, J. and Chau, S.
Business Opportunity Detection in the Big Data.
In *Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS 2016) - Volume 1*, pages 250-254
ISBN: 978-989-758-187-8

positive answers. This is a new research axe: the knowledge representation using graphs.

The interest of such a representation is to efficiently querying it. The issue of query formulation and searching in a global graph is not trivial. The objective is to propose a friendly-user interface for query building, and to give an understandable result: obviously, answer a sub-graph based on triple is not allowed for non-expert users; this becomes a problem of Human Machine Interface. This topic is covered by the axe of querying the knowledge.

After a query session, results have to be interpreted. A graphical representation with nodes and edges of the resulting graph is not adapted for a non-expert user. To allow a more efficient interpretation of the result it is needed to transform the resulting graph to a more user friendly representation. The research axe of result interpretation addresses this topic.

Last but not least, the results may be judged by the requestor more or less suitable. This feedback is necessary to increase efficiency of queries and accuracy of results. The last research axe called self-learning has the subject to consider how self-learning can be implemented in the query engine.

We choose to support this research a test case taken from an Altran's[1] need: how to match a business opportunity with the suitable consultant and in reverse, how to match a given consultant with a business opportunity.

When a customer asks our company for some skills, it is important to be able to find consultants that can answer its wishes in the best time. Internal data are very useful to retrieve consultants in the field of required qualifications: well-structured consultant skills and knowledge records, database of CVs that are also relatively well structured documents. But several external data sources are also very useful in skills search, like viadeo, linkedin, etc.

Matching request with the global graph will give us the possibility to have a look on nodes close to the request in order to retrieve additional information that can help to describe a context like new markets, skills, tenders, etc.

2 SYSTEM ARCHITECTURE

The main objective of Big Data technology is to integrate all data, then to analyze and represent this data in the unified schema. To reach this objective, we propose the following architecture (cf. Figure 1): This architecture is composed of three Parts:

Data Sources

It contains all data provided by Altran's tools. Data are extracted either from the enterprise databases or from flat files which can be structured or not structured.

These data were sorted and collected exhaustively to obtain relevant information that can be used to manage the Big Data System. The idea is to study and find a solution that allows to:

- Identify opportunities (projects, markets, skills, etc.)
- Improve transparency of existing data flow in the internal tools Altran
- Centralize existing information

Big Data Engine

The challenge of big data is to manage a large volume of data with optimal processing time.

We propose a big data engine based on Hadoop. Its HDFS file system allows the processing of very large amounts of data over several discs multiple machines as if it was a single storage volume.

In addition, the use of tools such as Storm can perform calculations and processing on graphs.

Implementing these two packages in Hadoop will reduce the time to treatment and process large volumes of data.

Other tools can also improve performance of process. For example, Kafka tool is used to enable the processing of queued messages.

Representation / Analyse / Compute:

The use of graphs databases (for example: Giraph, Neo4j) to represent the big data provides a unifying representation.

This offers a visual representation, easy to understand by the business. In addition, the use of graphs gives much better performance than relational databases, whether for the graph traversal or to load/import large data volumes.

3 OUR RESEARCH AXES

The objective of this section is to present the global framework of the solution to opportunity detection and its components which are the next research axis (cf. figure 2).

[1] Altran Technologies, SA is a global innovation and engineering consulting firm

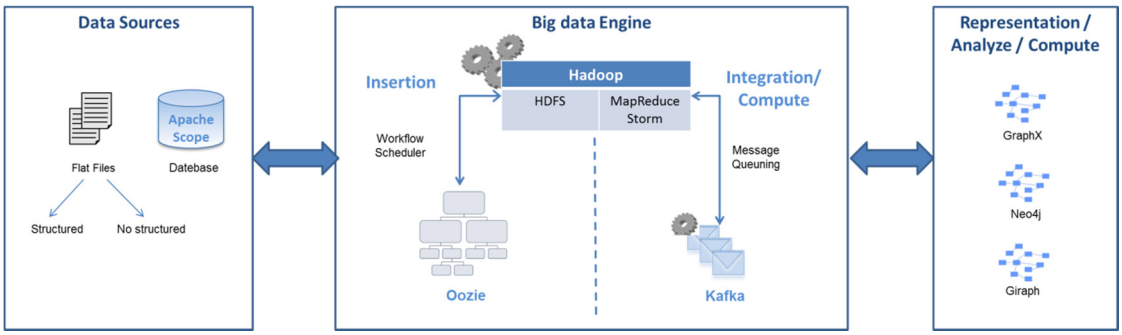

Figure 1: Representation of system architecture.

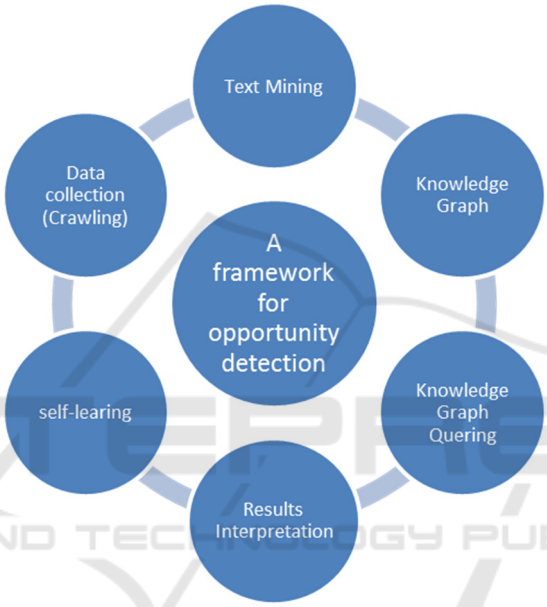

Figure 2: The Components of a Framework for Opportunity Detection.

3.1 Crawling

A crawler is software that explores recursively links found within a web page, from a pivot page, in order to collect and index the resources (web pages, images, videos, documents, etc.).

To enable the crawler to do its job correctly, one must define:

- A selection policy that identify the pages to download;
- A re-visit policy that defines when to check changes in the pages;
- A politeness policy that defines how to avoid overloads pages;
- A Parallelization policy that defines how to coordinate the crawlers in a distributed indexing.

With the introduction of the new Semantic Web research principles have been defined to allow crawlers to operate indexing methods involving more intelligent human-computer combinations as are practiced today.

The goal of this research axis is mainly to catch relevant pages in an intelligent manner. The ideal crawler should allow identifying pertinent data without drilling down the sources.

3.2 Text Mining

The crawling allows extracting from the WEB a set of sources which may have pertinent information. The second step is to inspect these sources to extract the information. It is the purpose of the text mining axis.

There are various types of sources that can be processed, but the large majority of them are textual: it is why we focus on text mining in this work.

Text mining is a complex process which deals with natural languages. The main difficulty is that a natural language is ambiguous, redundant and

implicit. Identifying new keywords and semantic links in a text need to use ontologies, heuristics and other sophisticated algorithms.

We choose to represent textual information as a global graph, where nodes are keywords, and edges are property links between these nodes. Edges are wearing semantic.

The nodes are found by keyword mining in the various structured and non-structured documents available in the company, like CVs, skill's records, etc. Using ontology based on business rules, enables us to categorize the identified keywords into different abstract levels, and to discover the semantic links existing between them. For example, the global graph should contain the consultants and their respective skills. This allows, for instance, retrieving the consultants that match a client request.

New keywords can be identified using a count of words and retaining only those which have a real sense. However for some structured documents like competency files, it is affordable to use the structure of the document to extract the relevant information. Anyway it is important that keyword extraction algorithms be able to adapt to the type of document.

Semantic links needs in many cases a certain degree of understanding of the analyzed document. This involves language treatments, using language tools enabling a more or less detailed analysis of the document content through content analysis techniques.

3.3 Graph based Knowledge Representation

As previously introduced, we choose to represent the extracted knowledge as an oriented and labeled graph where nodes are extracted keywords and links are semantic links between keywords.

Each analyzed document is resulting in a small graph representing the semantics extracted from the document. At this point we use RDF language to represent semantic graphs and to operate on them.

In order to form the global graph, we need to gather the different graphs coming from each analyzed document. As said before, this point is not trivial, because of the need to keep data dependencies from each to other; for example, assume A related to E related to B, and X related to E related to Y: a basic gathering will give A and X related to E related to B and Y, leading potentially to false positive answers A related to E related to Y, and X related to E related to B.

In order to deal with this constraint, we add an instance number in nodes: when such confusion may happen, the node is duplicated and a new instance number is given for each created node. In previous example, the node E is duplicated in two instances E[1] and E[2], thus we have A related to E[1] related to B and X related to E[2] related to Y; even if graphically the nodes are gathered, they are differentiated while analyzing the graph.

Another way may be to consider transitivity between links of different semantics, where A related to E and E related to B involves a potential transitive link A to B with a more precise meaning.

The global graph will first be built using well-structured documents like skills reports and CVs. It will be improved next by client requests and ontologies. Indeed, requests can enlarge application domains, add and refine skills, while ontologies will give abstraction levels that can extend or refine contexts.

3.4 Knowledge Graph Querying

Queries are given as small graphs similar to the global graph. A user interface will help to build queries in an understandable way, proposing refinements or contextual information that can give more precise formulations.

The graph of a query can be used, in active way, to quickly retrieve consultant that can satisfy a client request. It can be used also in a proactive way by augmenting the answer with neighborhood states or taking into account more general points of view, giving an extended view on client's needs, and thus allowing proposing complementary services. In a general active way, the sub-graph can be also used to determine market's trends, and identifying new relevant proposals of collaboration with new customers.

Some nodes of the query shall be "asking nodes": it will match any node of the global graph satisfying its relation links; thus, the entire query can match all the solutions of the need: for example, building a query on competencies with a "consultant asking node" will return all the consultants having these competencies.

An answer is thus a sub-graph deduced from global graph by matching query's nodes and links, with potentially several nodes for each "asking node".

Practically, the query graph construction could be not trivial: asking nodes could cover different information. For example, we could have consultants and enterprises skilled in programming languages: searching with an "empty asking node" will keep back consultants and enterprises. In order to reduce the field of answer, we decided to add a type to nodes; in such an example, consultants could be typed as person and enterprises as society. The query graph could then have a "typed empty asking node" in order

to retrieve only consultants.

Types are very important for query building, as we can propose to the user the list of types in which he can pick to refine its query.

Extracting the answer is not trivial too, as user could skip several intermediary nodes, because of not knowing it or simply to have a simpler query. Thus asking for a consultant skilled in java programming language could return a graph containing a node "object languages" that gather several languages like java, c++, c#, etc. More generally, simple queries could return complex chains of dependencies: the graph matching algorithm has to deal with this.

3.5 Result Interpretation

The query's result is a set of sub-graphs extracted from the global graph and matching the query graph. Representing this result in a human readable manner is not so trivial. It is easy to use a graphical representation where nodes are boxes and links are arrows. This form is acceptable for a human reading as long as the result set is not too big, but becomes unusable if there are hundreds or thousands of nodes.

Addressing this issue is not quite simple. Many studies have been done to try to solve it and many approaches exist (Shengqi et al., 2014), (Bergmann et al., 2014) with different approaches. However there is not any universal good approach. Each need may have own adapted representation. The goal is here to determine a good representation in existing tools, and if needed (and possible), adapt it to closely cover the specific need of our project.

3.6 Self-learning

Automatic learning concerns the design, analysis, development and implementation of methods allowing to a machine to evolve in a systematic process, and so fulfill the tasks difficult or impossible to fill by more conventional algorithmic means.

There are some kinds of self-learning algorithms. In our project we develop a self-learning method based on the user feedback. This method refers to a class of automatic learning problems, where the aim is to learn from experience, to optimize a quantitative reward over time.

The user feedback acts as a reward, and is used to improve the search algorithm, which in turn will be able to provide more accurate results.

4 CONCLUSION

Most of the techniques described above to achieve the goal of opportunity detection are recent research subjects. Some partial answers already exist, but it remains a lot of issues, difficulties and weakness in the big data mining and in the graph based knowledge representation. This research will try, using a test case of business opportunity detection, to address some of them and to propose original solutions to increase the efficiency and accuracy of the knowledge mining and restitution in the big data.

REFERENCES

Bergmann, G., Hegedüs, Á., Gerencsér, G., & Varró, D., 2014, 'Graph Query by Example', in CMSEBA in conjunction with MoDELS, pp. 17-24.

Ching-Yung Lin., 2014. 'Graph Computing and linked big data', Keynote speech at International Conference on Semantic Computing.

Diebold, F. X., 2012, '*A Personal Perspective on the Origin(s) and Development of Big Data: The Phenomenon, the Term, and the Discipline'*, Penn Institute for Economic Research, Department of Economics, University of Pennsylvania.

Fan, S., Lau, R. Y. & Zhao, J. L., 2015, 'Demystifying big data analytics for business intelligence through the lens of marketing mix', *Big Data Research*, vol. 2, no 1, pp. 28-32.

Fan, W. & Bifet, A., 2012, 'Mining big data: Current status, and forecast to the future', ACM SIGKDD Explorations Newsletter, Vol. 14, no 2, pp. 1-5.

Howe, A. D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Rhee, S. Y., 2008, 'Big data: The future of biocuration'. *Nature*, Vol. 455, no 7209, pp. 47–50.

Laurila, J. k. et al., 2012, '*The Mobile Data Challenge: Big Data for Mobile Computing Research'*, Nokia Workshopp, in conjunction with Int. Conf. on Pervasive Computing, no EPFL-CONF-192489

Letouzé, E., 2012. *Big data for development: Challenges and opportunities*, UN Global Pulse.

Probst, L. et al., 2013, '*Big data Analytics and decision making',* Business Innovation Observatory, European Commission.

Shengqi Yang, Yinghui Wu, Huan Sun, and Xifeng Yan, 2014, 'Schemaless and structureless graph querying', *Proc. VLDB Endow.* Vol. 7, no 7 pp. 565-576.

Trelles, O., Prins, P., Snir, M. & C.Jansen, R., 2011, 'Big data, but are we ready?', *Nature*, Vol. 12, no 224.

Valerie Bönström, Annika Hinze, Heinz Schweppe, 2003. 'Storing RDF as a Graph', 1st Latin American Web Congress, pp.27-36

World Economic Forum, 2012. Big Data, Big Impact: New Possibilities for International Development.