Recherche d'Information SEmantique, RISE 2014

J-P. Chevallet, Catherine Roussey, H. Zargayouna

Sixième Atelier Recherche d'Information SEmantique RISE, Nancy 18 mars 2014

Associé à SDNRI 2014

ACTES DE L'ATELIER RECHERCHE D'INFORMATION SEMANTIQUE RISE 2014

Édité par
Jean-Pierre CHEVALLET, LIG, Grenoble (France)
Catherine ROUSSEY, IRSTEA, Clermont Ferrand (France)
Haïfa ZARGAYOUNA, LIPN, Paris (France)

Sixième édition de l'atelier Recherche d'Information SEmantique

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE,
SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Atelier Recherche d'Information SEmantique RISE, Nancy 18 mars 2014

Associé à SDNRI 2014

1. Introduction

Nous avons le plaisir d'organiser à Nancy la sixième édition de l'atelier Recherche d'Information SEmantique, RISE 2014, associé à la Semaine du Document Numérique et de la Recherche d'Information et avec le soutien de l'ARIA (Association francophone de Recherche d'Information et Applications).

Le but de l'atelier est de proposer un espace d'échange autour de la synergie entre acquisition et gestion de ressources sémantiques (ontologies, terminologies, thesaurii, ...) et la Recherche d'Information. Ces thématiques sont à la croisée du Web Sémantique, de l'Ingénierie des Connaissances, du Traitement Automatique des Langues et de la Recherche d'Information.

Les thèmes couverts par les contributions acceptées à RISE sont les suivant :

- Reformulation de requêtes et structures d'index
- Enrichissement sémantique et peuplement d'ontologies

La conférence invitée de cette année a pour titre «What can you do with a Semantically Annotated Web in your Pocket?». Elle met l'accent sur les nouvelles applications de la recherche d'information sémantique, les types d'annotations utiles, les profiles des utilisateurs et les cas d'usage. Ci-dessous une brève présentation de Jaap KAMPS et de son intervention. Tous les supports sont en ligne sur le site de l'atelier.

Jaap Kamps is an associate professor of information retrieval at the University of Amsterdam's iSchool, PI of a stream of large research projects on information access funded by NWO and the EU, member of the ACM SIG-IR executive committee, organizer of evaluation efforts at TREC and CLEF, and a prolific organizer of conferences and workshops. His research interests span all facets of information storage and retrieval -- from user-centric to system-centric, and from basic research to applied research. A common element is the combination of textual information with additional structure, such as document structure, Web-link structure, and/or contextual information, such as meta-data, anchors, tags, clicks, or profiles. See:http://staff.science.uva.nl/~kamps/.

What can you do with a Semantically Annotated Web in your Pocket?

The early years of research on semantics annotations focused almost exclusively on the ``supply" end: how to turn unstructured information into machine readable "semantically" enriched data by relying on modern Web languages, user tagging and annotation, emerging robust NLP tools, and an ever growing volume of linked data. Recent years have seen an increasing focus on the "demand" end: what types of annotation and structure are actually useful, and for what sort of users, applications and use cases? This talk will give an overview of some of the current developments and directions, and some of the challenges and barriers to success ahead.

Nous remercions vivement Jaap KAMPS ainsi qu'aux auteurs et participants de contribuer activement à la bonne tenue de l'atelier RISE.

2. Comité de programme

2. Table des matières

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

An Attempt to Use Ontologies for Document Image Analysis

Bart Lamiroy[1] and Shabai Zheng[2]

[1] Université de Lorraine – LORIA (UMR 7503)
Campus Scientifique – BP 239, 54506 Vandoeuvre-lès-Nancy CEDEX, France
Bart.Lamiroy@loria.fr
[2] Université de Lorraine – École Nationale Supérieure des Mines de Nancy
Campus ARTEM – CS 14 234, 92 Rue Sergent Blandan, 54042 Nancy, France

Abstract. This paper presents exploratory work on the use of semantics in Document Image Analysis. It is different than existing semantics-aware approaches in the sense that it approaches the problem from a a very domain specific angle, and tries to incorporate an open model based on a reduced ontology. As presented here, it consists of enhancing an existing platform for Document Image Analysis benchmarking using off-the-shelf tools. The platform on which it is based hosts a wide variety of image interpretation algorithms as well as a wide range of benchmarking data. These data are stored in a relational database, as well as their type definition, the association between data and algorithms, *etc.* This work tries to provide an experimental indication whether ontologies and automated reasoning can provide new or alternative ways to extract relations among different stored facts, or infer dependencies between various user-defined types, based on their interactions with algorithms and other types of data.

1 Introduction

The aim of this paper is to report preliminary experimental setups related to introducing open semantics in Document Image Analysis (DIA). The topics described relate to a benchmarking platform, specifically conceived for DIA research, and stem from the need of intelligently handling a specific kind of semantics, as will be made clear later. The DAE Platform[1], on which this work is based, provides an experimental environment in which users can upload algorithms, store relevant data, conduct experiments and compare their results. This system is conceived around a flexible and open data model that links together document images, algorithms, user-defined interpretations and user information. All is stored in a relational database. Among these, the most important is a wide range of data related to document analysis algorithms that can generally be divided into user-defined data types as inputs of algorithms and automatic interpretations as results. However, the flexibility of this model in terms of interpretations and data types is one of its main weaknesses. One may define a type without considering that it may have other equivalent types in the system defined by others. Moreover, sometimes existing

types may form super types or subtypes of the one the user intended to create. The goal of this research consists in developing a semantic model of the knowledge stored in the database, which will allow automatic reasoning engines to infer dependencies between various user-defined types, based on their interactions with algorithms and other data types. One important step in this direction is to represent the underlying data model as an ontology.

The next section will explain in further details the goals of the benchmarking platform, the specific need for semantics it has, and how it differs from other existing platforms. In section 3 we explain the techniques used to integrate ontologies into the model. Section 4 provides some experimental results.

2 Context

2.1 DAE-platform

The DAE-Platform's [1] primary goal is to serve as a reference repository for reproducible experimental research in Document Image Analysis, and as such offers access to reference data, state-of-the-art algorithms, and benchmark annotations. It enables registered users to upload new reference material, to define, extend or correct the interpretation of stored data and aims to support experimental research through an open and evolving framework that can be extended by community driven contributions.

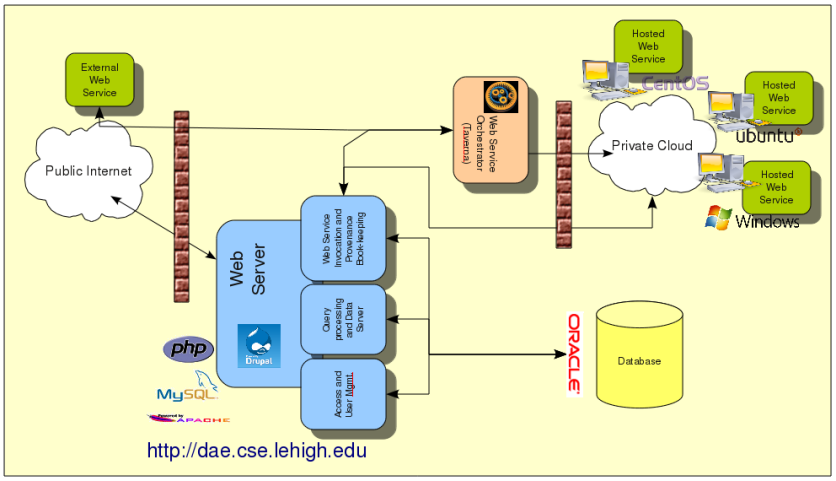

Fig. 1. Architecture of the DAE Platform

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Previous accounts of the features and the potential uses of the platform [2,3] focused on its architecture, implementation choices, and its potential to impact experimental research, especially with respect to reproducibility and accountability. The platform has been implemented using a classical LAMP architecture [4] but also integrates a full WSDL interface [5], allowing it to be integrated in a fully distributed service-oriented architecture. The DIA reference data and annotations themselves are stored in an Oracle relational database back-end. An independent cluster of servers is used to execute stored algorithms. Fig 1 shows its structure.

2.2 Data model

Besides the technical architecture of the server, as depicted in Fig 1, the DAE Platform is an implementation of a data model [6] .The data model is based on the following claims: all data is typed; users can define new types;

- Data can be attached to specific parts of a document image (not obliged)
- Both data and algorithms are modelled; algorithms transform data from one type into data of another type;
- Full provenance of data history is recorded.

The result is that the platform can be used as a benchmarking repository. Since all provenance is recorded, one can leverage it to compare results from algorithms on identical data, and compare annotations by different users.
A simplified representation of the data model is represented in Fig 2. It consists of three key elements: `algorithms, data_items` and `algorithm_runs`.

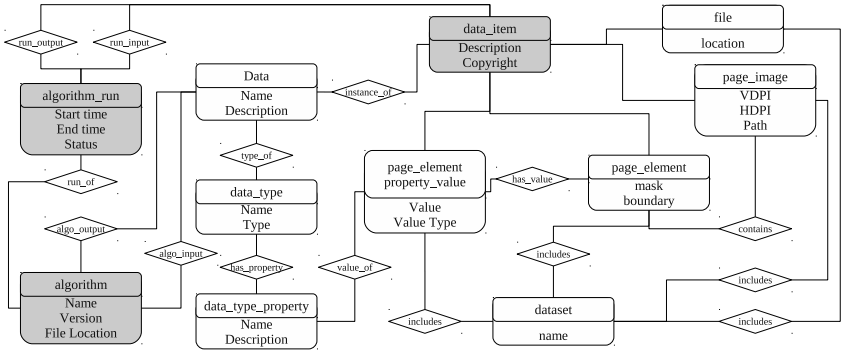

Fig. 2. The DAE Data Model (simplified)

The underlying reasoning is that data is transformed by algorithms. `data_items` are instances of data and are related to algorithms by explicit `algorithm_runs`. New `data_items` thus produced are stored in the

database with the exact information of how they were obtained. There are also pre-defined types of `data_items: page_images, page_elements, page_element_properties, files` and `datasets`. In this document we focus on the first three.

- `page_image` is an image file representing a physical page at a given resolution and with a given image quality. It is perfectly possible to have multiple `page_images` representing the same physical page, as shown in Fig 2;
- `page_element` is an area of a `page_image`, defined in as unconstrained a way as possible, either by a bounding box or a pixel map
- `page_element_property` are any kind of user-defined property or interpretation attached to a `page_element`.

3 Semantics and Ontologies

3.1 A Specific Kind of Semantics

As already mentioned, the flexibility of DAE platform causes polysemy. Unlike other image annotation initiatives, our main goal is not to try and relate visual representations to higher level known or fixed ontologies [7]. Although we will eventually benefit from connecting our ontology with it, it is currently not within the scope of this paper.
Our main concern comes from the fact that different users very likely operate in different interpretation contexts, which inevitably may result in several types for the same visual element. We have already shown that this is a normal, and necessary "feature" in machine perception and interpretation related topics [8]. The problem is, the system cannot automatically deduce or indicate the relation among these interpretations. Our goal is to try and investigate if an appropriate ontology, capturing the semantics of our benchmarking data can help resolve this.

3.2 Example

In order to illustrate our point, let us develop the example depicted in Fig. 3. It represents the platform with various data objects and algorithms, created by different users (researchers).
We assume that one of the stored algorithms was contributed by researcher A, who defined its input as being of type X and its output of type Y. Another researcher B successfully execute this algorithm with, as input, a specific data item i whose type was previously defined as being Z.
It is straightforward to infer that:

$$\begin{cases} \mathrm{Type}(Z) \cap \mathrm{Type}(X) \neq \emptyset \\ i \in \mathrm{Type}(Z) \cap \mathrm{Type}(X) \end{cases}$$

The fact that types Z and X are considered being different may come from either having different users using different name labels for the same semantics (*e.g.* "text area" and "text zone"), or it may result

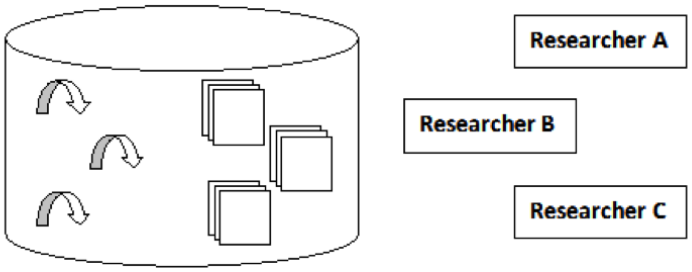

Fig. 3. Various Data Used by Different Users

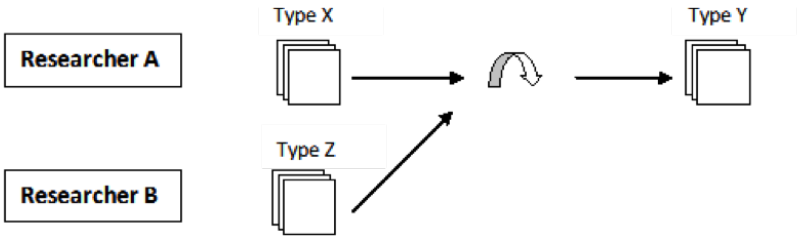

Fig. 4. Different Data Used by the Same Algorithm

from having both users using slightly different interpretation contexts with overlapping, yet not fully identical semantics. This situation is neither far-fetched, nor unique. Since the researches in domain of document analysis can be very specific (e.g. concentrate in translating a file from a particular form to another) and the domain itself is becoming more and more complex, some researchers may work on quite specialised subproblems, that may eventually prove to be instances of a more generic problem. In order to detect this, we need to extract the dependency between equivalent user-defined types.

3.3 Semantics of the data model

In the DAE Data Model, algorithms are declared having precise `data_type`s for their input and output values (called `data`). However, an algorithm can have any number of executions (`algorithm_run`s), associating specific `data_item`s. In other words, specific `data` (algorithm inputs/outputs) may possess various instances at run-time. The problem discussed above lies in the fact that, a given `data_item` can be sometimes applied in executions of different algorithms, which means it serves as `data` of potentially different types. Conversely, this means that once we find all the `data_type`s of a `data_item`, we can easily trace to every algorithms where this `data_item` is involved. Our study is based on this point. Therefore, one important issue is to establish the link from specific `data` to each of its instance `data_item`s.

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Given the architecture of the DAE platform, we could achieve this using SQL queries. However, this solution is difficult to generalise: every time when we want to check a particular `data_item`, we need to modify the query statement (e.g. `data_item`'s id). Since the number of `data_items` is enormous, using SQL complicated, tedious and error prone. Introducing semantic querying provides a more readable presentation for complicate hierarchical relationships between information. Furthermore, DL reasoners will allow us to infer a transitive chain of a series of relations between `data_type` and `data_item`.

3.4 D2R

To achieve this, we will be using D2R [9]. D2R Server is a tool for publishing the content of relational databases on the Semantic Web. Database content is mapped to RDF by a declarative mapping which specifies how resources are identified and how property values are generated from database content. D2R Server uses a specific language[3] to express mappings between application-specific database schemas and RDFS schemas or OWL. They specify how resources are identified and how property values are generated from database content. The central object in D2RQ is the *ClassMap*. A *ClassMap* represents a mapping from a set of entities described within the database, to a class or a group of similar classes of resources. Each *ClassMap* has a set of *PropertyBridges*, which specify how resource descriptions are created. Property values can be created directly from database values or by employing patterns or translation tables. D2RQ supports conditional mappings on *ClassMap* and *PropertyBridge* levels, mapping of multiple columns to the same property and the handling of highly normalised table structures where instance data is spread over multiple tables. It also includes a tool that automatically generates a D2RQ mapping from the database table structure, generating the appropriate RDF for each database, using table names as class names and column names as property names.

4 Experiments and Analysis

4.1 Matching `data` and `data_items`

As mentioned above, given a specific algorithm execution allows to relate a `data_item` to `data` properties. We know that an algorithm is defined as: ($output_1$, $ouput_2$, ... , $output_n$) = algorithm ($input_1$, $input_2$, ... , $input_m$) Where $input_i$ (i = 1 ... m) is defined as $\texttt{data}_i$, who possess a type $\texttt{data\_type}_i$. In the same way, $output_j$(j = 1 ... n) is defined as $\texttt{data}_j$ of $\texttt{data\_type}_j$. These data are associated directly with the algorithm in a strict order. This position information is available in our provenance data from the DAE platform. Fig. 5 we see the logical order of the inference. We have implemented this inference chain in Protégé[4]. Fig. 7 shows that we are capable of correctly inferring new types on data items.

[3] D2RQ Mapping Language: http://d2rq.org/d2rq-language
[4] http://protege.stanford.edu/

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

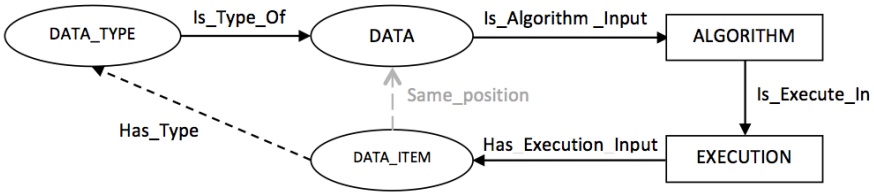

Fig. 5. Inference of Data Types from Algorithm Executions

Up to this point, we can conclude that if all relevant properties are properly defined(extracted), the desired deduction set by inference chain will be established automatically on the whole model after running the DL reasoning engine. However, after the implementation and tests, several issues left to be resolved:

- Complexity: a chain of 4 properties slows down the reasoning;
- Limited by the fact that the enforcement of argument positioning requires reification of the relations (our example is limited to two inputs/outputs only);
- We failed to realize the extraction by ordering: Because the D2RQ mapping language cannot yet function normally on conditional mapping.

Besides its inconveniences and technical limitations, this semantic model functions as expected, so the applied methods proved to be correct. Moreover, it proposes an important usage of the ontology modeling: verification of complex relations in relational database.

4.2 Simplification

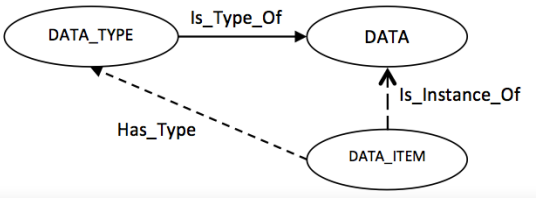

Fig. 6. Simplified Inference of Data Types from Algorithm Executions

In order to avoid all the problem occurred in the first approach and to provide a functional model, we decided to replace the ordering information by storing a direct link between `data_item` and `data_type` in our provenance database. Every time when an algorithm runs, we record each of the `data_items` used as its inputs/outputs, along with the information

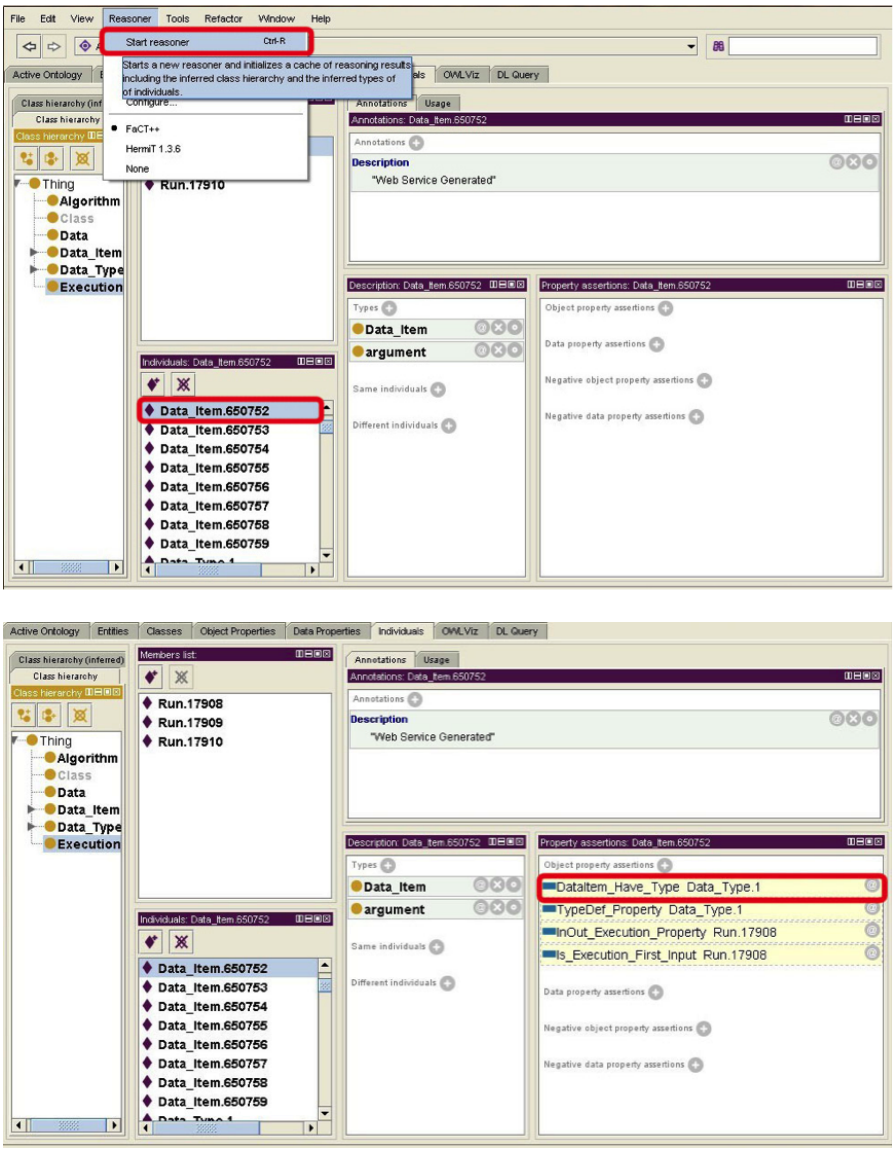

Fig. 7. Results of Type Inference from Algorithm Executions using Protégé

of this algorithm in the database. Besides, the corresponding data are also recorded. The resulting semantics is shown in Fig. 6.
This model provides the solution in a most simplified way with only three classes and is more efficient than the one in the first approach.

4.3 About Performance Analyisis

Knowing that using the ontology model for queries can provide an alternative to relational database, we would like to examine its performance comparing to traditional SQL in a given case, which is described below. The approach described here is fairly naive and would require a more thorough experimental investigation. For one, using a reasoner is unfair, and definitely not the optimal approach. Further work will necessarily need to include benchmarking with triple-store databases [10,11].
However, to show the importance of correctly choosing tools and approaches, we will develop the comparative experimentation we have conducted, even though it is open to criticism. As mentioned before [6], a `page_image` consists of several `page_element`s (*cf.* Fig. 2). A `page_element` itself is a `data_item`, but it usually contains "sub" `data_item`s. In fact, it is defined as "a physical sub-part of an image" and its basic description is only its geometry. Furthermore, attached to them are `page_element_property_value`s to record "interpretations" (*e.g.* OCR results, layout labels...). Meanwhile, these interpretation also have a `data_type`, *etc.* ending up by composing a complex network of semantic relations.
Let us now consider the following example: suppose there is a `page_element` whose `data_type` is "textline" (an area containing a line of text), and it have several `page_element_property_value`s such as "number of the next line" and "transcription" (*i.e.* its literal content). In that case, the `data_type` of "number of the next line" would be "page_element id" or simply "integer", for instance; the `data_type` of "transcription" would be "String"; "transcription" would also be considered as a specific property.
So what if we need to know all annotations and their types attached to a given `page_image`? Seen from a part of the data model shown in Fig. 8, this logic is not directly reflected in the current database and requires a series of SQL join operations to extract the list of all `data_type`s attached to a given `page_image`.

```
SELECT pi.ID AS PAGE_IMAGE_ID,dt.ID AS Data_Type_ID
FROM PAGE_IMAGE pi
LEFT JOIN CONTAINS_PAGE_ELEMENT cpe ON
           pi.ID = cpe.PAGE_ELEMENT_ID
LEFT JOIN PAGE_ELEMENT_UNDERLYING pe ON
           cpe.PAGE_ELEMENT_ID = pe. ID
LEFT JOIN HAS_VALUE hv ON pe.ID = hv.PAGE_ELEMENT_ID
LEFT JOIN PAGE_ELEMENT_PROPERTY_VALUE pe_pv ON
           hv.PAGE_ELEMENT_PROPERTY_VALUE_ID  = pe_pv.ID
LEFT JOIN VALUE_OF vo ON
```

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

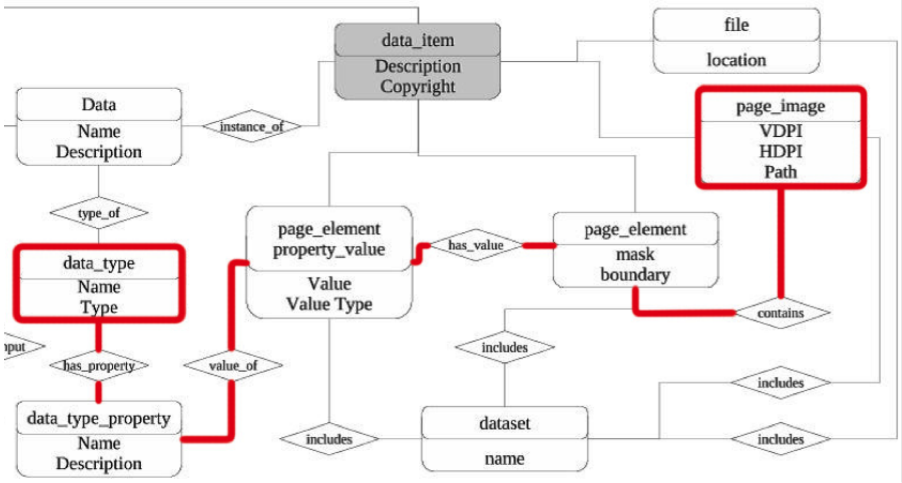

Fig. 8. Representation of required join operations to extract all annotations (and their types) of a `page_image`

```
          pe_pv.ID = vo.PAGE_ELEMENT_PROPERTY_VALUE_ID
LEFT JOIN DATATYPE_PROPERTY dp ON
          vo.DATA_TYPE_PROPERTY_ID = dp.ID
LEFT JOIN HAS_PROPERTY hp ON dp.ID = hp.DATA_PROPERTY_ID
LEFT JOIN DATATYPE dt ON hp.DATA_TYPE_ID = dt.ID
```

We have tested this query with eight join operation between nine tables. With about 20,000 `page_image`s, the wall clock execution time (inter-continental network traffic included) of this query is around 5 seconds.

We can also express the join using a property chain in Protégé and use reasoners to infer the information.

DataType_Have_Property o DataType_Property_Have_Value o Value_Of_Element o Element_Of_Image → DataType_Appears_In_Image

It is notable that the running speed of launching the reasoner largely depends on the size of the extracted initial dataset. For example, in a predigest version of the database with 50 `page_image`s on a personal computer, the running time is around 1.4 seconds; but if the number of `page_image`s increase to 2000, the reasoning will take more than 20 minutes. Also, the choice of reasoner in this experiment can influence the result. During the experiment, we could obtain the expected test result with FaCT++ [12] and Hermit (1.3.6) [13]. However, with RacerPro [14], the property chain failed to work.

Although we are aware of the limited scope of the above experiment, it does highlight some interesting points. The SQL query is complicated, requires in depth knowledge of the data model and not easily reusable. However, it is very efficient. The reasoner approach is much less efficient and dependant on the underlying reasoner implementation or technique. However, it reformulates the intricate table structures into a concise graph. Moreover, the query expression of model is highly reusable as it is very simple to define new properties based on existing ones.

5 Conclusion

In this work, we have demonstrated that the use of semantics opens new perspectives of looking at document image annotation and benchmarking repositories, since it allow to infer and extract new kinds of relations between data. This can be used as a basis of trying to uncover whether different algorithms or different experimental setups share similar interpretation contexts, and thus discover new relations between various interpretations of the same data. Our approach offers advantages compared to the previously existing tool. Relations can be more easily created, modified and integrated using the appropriate an ontologies. In the domain of document analysis research, the ontology modelling is helpful in inferring the dependency between existing algorithms developed by individual researchers. Moreover, it can be considered as an efficient way to verify existing relationships between tables in the database.

Acknowledgements

Part of this work was based on previous, unpublished work on configuring a D2RQ server, by students A. Hombourger, P. Lauffenburger and J. Pruvost. Without their technical help, this work would never have been possible.

References

1. Lamiroy, B., Lopresti, D., Korth, H., Heflin, J.: How Carefully Designed Open Resource Sharing Can Help and Expand Document Analysis Research. In Gady Agam, C.V.G., ed.: Document Recognition and Retrieval XVIII - DRR 2011. Volume 7874., San Francisco, United States, SPIE, SPIE (January 2011)
2. Lamiroy, B., Lopresti, D.: An Open Architecture for End-to-End Document Analysis Benchmarking. In: 11th International Conference on Document Analysis and Recognition - ICDAR 2011, Beijing, China, International Association for Pattern Recognition, IEEE Computer Society (September 2011) 42–47
3. Lamiroy, B., Lopresti, D.: A Platform for Storing, Visualizing, and Interpreting Collections of Noisy Documents. In: Fourth Workshop on Analytics for Noisy Unstructured Text Data - AND'10. ACM International Conference Proceeding Series, Toronto, Canada, IAPR, ACM (October 2010)

4. Wikipedia: Lamp (software bundle) — Wikipedia, the free encyclopedia (2008) [Online; accessed 01-Sep-2008].
5. Chinnici, R., Moreau, J.J., Ryman, A., Weerawarana, S.: Web services description language (wsdl) version 2.0 part 1: Core language. World Wide Web Consortium, Recommendation REC-wsdl20-20070626 (June 2007)
6. Korth, H.F., Song, D., Heflin, J.: Metadata for structured document datasets. In: Proceedings of the 8th IAPR International Workshop on Document Analysis Systems. 547–550
7. Halaschek-Wiener, C., Golbeck, J., Schain, A., Grove, M., Parsia, B., Hendler, J.: Photostuff - an image annotation tool for the semantic web. In: 4th International Semantic Web Conference - Poster Paper. (2005)
8. Lamiroy, B.: Sur les limites de la perception artificielle et de l'interprétation. Hdr, Université de Lorraine (December 2013)
9. Bizer, C., Cyganiak, R.: D2r server-publishing relational databases on the semantic web. 5th international Semantic Web conference (2006) 26
10. Bizer, C., Schultz, A.: Benchmarking the performance of storage systems that expose sparql endpoints. World Wide Web Internet And Web Information Systems (2008)
11. Voigt, M., Mitschick, A., Schulz, J.: Yet another triple store benchmark? practical experiences with real-world data. In: SDA. (2012) 85–94
12. Tsarkov, D., Horrocks, I.: Fact++ description logic reasoner: System description. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006). Volume 4130 of Lecture Notes in Artificial Intelligence., Springer (2006) 292–297
13. Motik, B., Shearer, R., Horrocks, I.: Hypertableau Reasoning for Description Logics. Journal of Artificial Intelligence Research **36** (2009) 165–228
14. Haarslev, V., Hidde, K., Möller, R., Wessel, M.: The racerpro knowledge representation and reasoning system. Semantic Web **3**(3) (2012) 267–277

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Service contextuel d'aide à la recherche d'information par couplage requête / moteur

Aurélien Saint Requier[1] , Youssouf Saidali[1], Sébastion Adam[1], Yves Lecourtier[1],

[1] Université de Rouen, LITIS, 76801 Saint Etienne du Rouvray, France

aurelien.saint-requier@etu.univ-rouen.fr
{youssouf.saidali, sebastien.adam, yves.leourtier}@univ-rouen.fr

Résumé. Nous présentons ici l'état de nos travaux sur l'élaboration d'un système d'aide à la recherche d'information par une sélection automatique de services web en fonction du besoin et du contexte utilisateur. Dans une première section, nous décrivons les démarches de modélisation et sélection de Services de Recherche d'Information (SRI). Nous présentons ensuite notre approche sur la sélection de SRI basées sur un profil long et court terme de l'utilisateur. Puis, nous détaillons la réalisation d'un système expérimental sur la fusion et l'exploitation de ces profils pour proposer un couple moteur/requête adapté.

Mots-clés: Recherche d'Information, Modélisation de l'utilisateur, Sélection de Services de Recherche d'Information

1 Introduction

Dans cet article, nous nous sommes intéressés à l'exploitation de profils utilisateurs pour la personnalisation de la RI. Nous avons constaté qu'en RI personnalisée, la dimension centrale d'un profil utilisateur était le domaine d'intérêt qui regroupe les informations ciblées par l'utilisateur et son niveau d'expertise sur un domaine particulier. Différentes représentations des centres d'intérêt sont couverts par la littérature: ensembliste [1][2], connexionniste et conceptuelle[3]. La représentation ensembliste apporte l'avantage de la simplicité de mise en oeuvre mais elle manque de structuration et de relations de corrélations entre les divers centres d'intérêts de l'utilisateur. La représentation conceptuelle comble le manque de sémantique de la représentation connexionniste, mais est souvent difficile à mettre en oeuvre dans un processus de personnalisation du fait que la majorité des services de recherche d'information se base sur une représentation ensembliste du couple requête/documents.
Dans le cadre de notre approche, nous nous intéressons plus particulièrement à la personnalisation de la requête de l'utilisateur, notre but étant de sélectionner divers services en fonction du besoin de l'utilisateur. Il nous est donc nécessaire de

transformer la requête de l'utilisateur dans une forme permettant à notre système de comprendre le besoin attendu par l'utilisateur. Par ailleurs, les systèmes de recherche d'information web ne donnant que très rarement accès à l'algorithme du calcul de pertinence et aux résultats retournés, nous ne pouvons pas agir sur ces étapes du processus de recherche d'information. Nous nous appuierons sur un profil utilisateur afin de capitaliser sur les informations extraites des activités de l'utilisateur sur le service de recherche d'information. Afin d'avoir un profil utilisateur sémantique, nous représenterons les centres d'intérêts de l'utilisateur par un ensemble de concepts issues d'une ontologie générale. De plus, nous différencierons le profil long terme de l'utilisateur, qui représentent sa connaissance, du profil court-terme, qui représente ses intérêts courants. Notre processus de personnalisation consistera donc à transformer le besoin exprimé sous forme de mots-clés en un besoin conceptuel en fonction du profil utilisateur afin de sélectionner le service de recherche d'information web adapté à ce besoin.
Si on regarde dans la littérature, les travaux sur la sélection de services de recherche d'information [4][5] sont basés sur la compréhension de la tâche de recherche d'information que l'utilisateur réalise pour identifier son besoin d'information. Nos travaux proposent d'exploiter les caractéristiques de cohérence thématique entre un couple de requêtes afin d'identifier si elles appartiennent à un même objectif pour déduire quels services de recherche sont adaptés au besoin de l'utilisateur.

2 Construction et fusion de profils utilisateur

Pour mener avec succès une tâche de recherche d'information sur le Web, l'utilisateur doit adopter une stratégie en plusieurs étapes. Le modèle standard de la Recherche d'Information définit par Sutcliffe et Ennis [6] présente un cycle de 4 activités :

1. Identification du problème
2. Définir le besoin d'information
3. Formuler la requête
4. Evaluer les résultats

L'utilisateur débute une tâche de recherche en définissant son besoin d'information, exprimé dans un premier temps dans une forme verbale. Ensuite l'utilisateur formule ce besoin d'information sous forme d'une requête. La requête est ensuite exploitée par le moteur de recherche sélectionné par l'utilisateur qui lui fait correspondre des documents. L'utilisateur évalue l'ensemble des documents retournés par le moteur de recherche et raffine sa requête si besoin. Le cycle est répété jusqu'à ce que le besoin d'information de l'utilisateur soit comblé.
L'approche que nous proposons a pour objectif d'aider l'utilisateur dans la mise en place de cette stratégie de recherche, en reformulant le besoin exprimé par l'utilisateur et en sélectionnant un service de recherche adapté à ce besoin

2.1 Présentation de notre approche

Pour assister l'utilisateur dans son processus de recherche d'information nous nous basons sur le schéma de la figure 1.

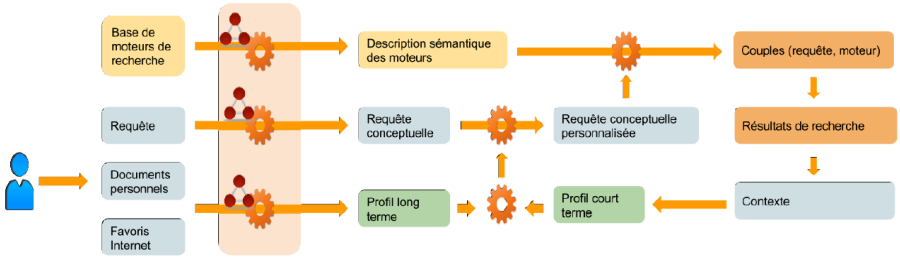

Fig. 1. Modélisation du processus de recherche de l'approche proposée

Le résultat final proposé à l'utilisateur est une suggestion de couples formés d'une requête conceptuelle personnalisée et d'un service de recherche adapté afin de guider l'utilisateur vers un service de recherche adapté à son besoin d'information avec une requête reflétant le plus fidèlement ce besoin. Le modèle global du système proposé peut-être décomposé en trois processus : (i) la modélisation des intérêts de l'utilisateur par un profil long et court terme, (ii) la personnalisation de processus de recherche d'information par une transformation de la requêtes mots-clés en une requête conceptuelle personnalisée et (iii) le processus de sélection du service de recherche par le couplage d'une requête conceptuelle avec un service de recherche.

2.2 Modélisation des intérêts de l'utilisateur

L'approche de construction du profil utilisateur que nous proposons repose une représentation des centres d'intérêt de l'utilisateur par des concepts de l'ontologie DBPedia. La base de connaissance DBpedia fournit un ensemble de données structurées provenant de Wikipedia qui peut être interrogé et lié à d'autres ensembles de données. La base de connaissance a de nombreux avantages sur les bases de connaissance existantes [7] : elle couvre un large éventail de domaines, elle represente un réel accord de la communauté, elle évolue automatiquement avec les mises à jour de Wikipedia, elle est multilangues et est accessible sur le Web. L'ontologie est organisé en 320 classes qui forment une hiérarchie de subsomption et sont décrites par 1650 propriétés différentes. Actuellement, la base de données décrit plus 3,4 millions d'entités dont 1,5 millions sont classées dans une ontologie co- hérente, dont 764 000 personnes, 573 000 lieux, 333 000 oeuvres de création, 192 000 organisations, 202 000 espèces et 5 500 maladies. De plus, les entités sont catégorisées dans plus de 800 000 catégories Wikipedia. Les catégories sont utilisées dans le but de lier les articles sous un thème commun. L'ensemble des catégories forment une hiérarchie, bien que les sous-catégories peuvent appartenir à plus d'une catégorie. Elles sont décrites dans les langages formels SKOS (Simple Knowledge Organization System) et DCMI

Terms. Le langage SKOS permet une représentation standard des thésaurus, classifications ou tout autre type de vocabulaire contrôlé et structuré. Un concept SKOS est ainsi défini comme une ressource RDF, qui contient des propriétés RDF comme un label, des synonymes, des définitions et des relations. Le Dublin Core [16] est un schéma de métadonnées générique qui permet de décrire des ressources numériques ou physiques et d'établir des relations avec d'autres ressources. Les catégories Wikipedia sont décrites par la métadonnée « subject » et contiennent comme valeur un concept SKOS.

Ainsi, pour nous, un profil utilisateur est défini par:

$$P_{cat} = \langle (cat_1, \omega_1), (cat_2, \omega_2), \ldots (cat_i, \omega_i) \rangle$$

où cat_i est un concept SKOS décrivant une catégorie Wikipedia et ω_i le poids du concept correspondant. La déduction des catégories Wikipedia cat_i appartenant au profil est réalisée à partir des concepts extraits des documents de l'utilisateur (ses documents personnel pour le profil long terme et les pages web visitées durant la session de recherche pour le profil court terme). Soit $E = \langle (c_1, p_1), (c_2, p_2), \ldots (c_i, p_i) \rangle$ l'ensemble des concepts extraits où c_i est un concept DBpedia et p_i le poids associé correspondant à la proportion de documents du corpus qui contiennent c_i, alors :

$$\forall\, i,j \in |E| \text{ et } \forall\, k \in C(c_i),\ cat_k(c_i) \begin{cases} \in P_{cat} & \text{si } \exists\, l \in C(c_j) \text{ tel que } cat_k(c_i) \equiv cat_l(c_j) \\ \notin P_{cat} & \text{sinon} \end{cases}$$

avec $C(c_i) = (cat_1, cat_2, \ldots, cat_k)$ l'ensemble des catégories Wikipedia du concept c_i. Le poids ω_i d'une catégorie Wikipedia cat_i d'un profil utilisateur P_{cat} est calculé comme suit :

$$w_i = \begin{cases} \sum_{i,j} p(c_i) + p(c_j) & \text{si } \exists\, l \in C(c_j) \text{ tel que } cat_k(c_i) \equiv cat_l(c_j) \\ 0 & \text{sinon} \end{cases}$$

Le poids d'un concept c_i est calculé différemment selon que l'on construise le profil long-terme ou le profil court-terme. Dans le cadre du profil long-terme, le poids correspond à la proportion de documents du corpus qui contiennent le concept c_i normalisé entre 0 et 1. Dans le cadre du profil court-terme, nous avons utilisé une pondération temporelle en nous basant sur le postulat que les interactions de l'utilisateur ont plus d'importance moins elles sont éloignées dans le temps. Nous avons adapté la fonction de décroissance présentée dans [7] à notre approche. Soit $p(p_v)$ le nombre de pages web visitées par l'utilisateur durant une session de recherche alors, $p(p_v) = 1$ est la page la plus récemment consultée par l'utilisateur. Nous définissons $c^{p(p_v)-1}$ comme fonction de décroissance, ou c est le facteur de décroissance. Nous fixons $c = 0.95$ qui est un compromis entre la forte accentuation des extrêmes et l'uniformité de toutes les actions [7].

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

La figure 2 montre un exemple de déduction du profil utilisateur P_{cat} à partir d'un ensemble de concepts.

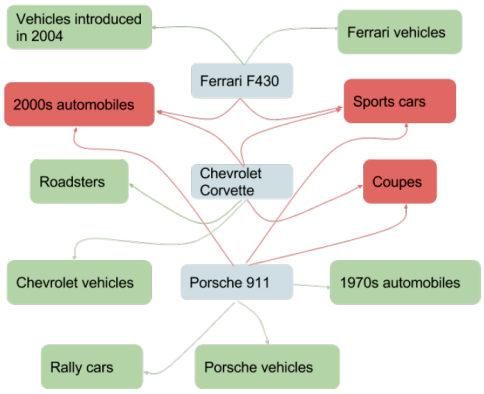

Fig. 2. Déduction d'un profil thématique à partir des concepts DBpedia.

Nous considérons que les trois concepts (en bleu) Ferrari F430, Chevrolet Corvette et Porsche 911 ont été extraits des documents de l'utilisateur. Pour chaque concept, nous récupérons l'ensemble de leurs catégories Wikipedia (en vert) correspondant à la propriété *dcterms :subject*. Les catégories Wikipédia possédant au moins une relation avec l'ensemble des concepts extraits sont ajoutées au profil utilisateur (en rouge). Notre profil utilisateur P_{cat} donné en exemple sera donc constitué des catégories Wikipedia illustrées en rouge : 2000s automobile, Coupes et Sports cars.

Ensuite, pour d'exploiter l'information contenu dans ces deux types de profils dans le processus de recherche d'information, nous proposons un algorithme de fusion du profil long-terme avec le profil court-terme.

2.3 Fusion des profils

Afin d'exploiter les deux types de profils dans le processus de recherche, nous présentons une fonction de fusion Φ du profil court-terme $P_c = \langle (cat_1, \omega_1), (cat_2, \omega_2), \ldots (cat_i, \omega_i) \rangle$ et du profil long-terme $P_l = \langle (cat_1, \omega_1), (cat_2, \omega_2), \ldots (cat_j, \omega_j) \rangle$ telle que:

$$P_{l \cup c} = \Phi(P_c, P_l) = P_c \cup P_l = \langle (cat_1, w_1), \ldots, (cat_k, w_k) \rangle$$

où:

$$w_k = \begin{cases} mean(P_c) + mean(P_l) + (w_i + w_j) & \text{si } cat_k \in P_c \cap P_l \\ mean(P_l) + w_i & \text{si } cat_k \in P_c \\ mean(P_c) + w_j & \text{si } cat_k \in P_l \end{cases}$$

avec:

$$mean(P) = \frac{1}{|P|} * \sum_{i=0}^{i<|P|} w_i$$

La fonction de fusion permet ainsi de renforcer le poids des catégories qui sont présentes à la fois dans le profil court-terme et dans le profil long terme. Il faut noter ici que si $P_c = \varnothing$ alors $P_{l \cup c} = P_l$. Cette propriété est importante car elle permet de résoudre le problème de démarrage à froid. En effet, au début d'une session de recherche, le profil court-terme peut être vide d'où l'intérêt de tenir compte du profil long terme pour la personnalisation. Cependant, l'utilisateur peut avoir consulté des pages web qui peuvent être pertinentes pour personnaliser le processus de recherche avant même d'avoir débuté une session de recherche [8]. La personnalisation du processus de recherche d'information se basera donc uniquement sur les informations présentes dans le profil long-terme de l'utilisateur si le profil court-terme est vide. La figure 3 montre le résultat de la fusion (courbe bleue) entre un profil court-terme (courbe orange) et un profil long-terme (courbe jaune). Nous constatons que les intérêts du profil long-terme avec un poids important (par exemple *Information Retrieval*) ont toujours un poids important dans le profil fusionné. De plus, les intérêts présents dans les profils court et long terme comme *Javascript* ou *Scripting languages* prennent de l'importance dans le profil fusionné.

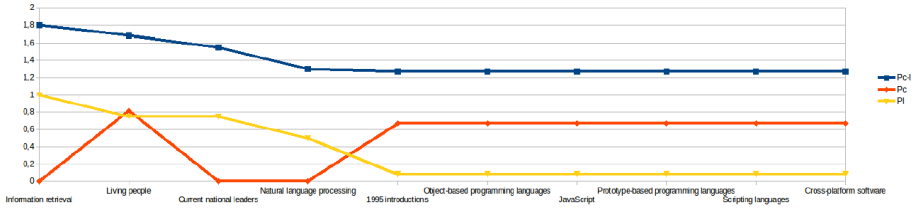

Fig. 3. Exemple de fusion de profil

C'est donc ce profil résultant de la fonction de fusion qui sera exploité dans le processus de personnalisation.

2.4 Reformulation du besoin

Le processus de recherche d'information peut être personnalisé à plusieurs niveaux : la personnalisation du besoin, de l'algorithme de pertinence ou des résultats de recherche. Nous nous consacrons à la personnalisation du besoin d'information.

La technique que nous proposons consiste à ordonner l'ensemble des concepts L_c récupérés à partir des mots clés de la requête de l'utilisateur en fonction du profil utilisateur. Pour chaque concept récupéré, on calcul un indice de similarité en fonction des concepts des catégories Wikipedia présents dans le profil utilisateur. La similarité sémantique entre chaque concept c de la liste L_c avec le profil conceptuel

$$Sim(c/P_{cat}) = \sum_{i=0}^{|P_{cat}|} w_i \times \sum_{j}^{|L_c|} \sum_{k}^{|cat(c)|} Sim\left(cat_k(c_j), cat_i\right)$$

de l'utilisateur P_{cat} se base sur l'approche de Milne et Witten [9]. Ce calcul de similarité sémantique est donné par la relation suivante :

où: $$Sim\left(cat_k(c_j), cat_i\right) = \begin{cases} 1 & \text{si } cat_k(c_j) \equiv cat_i \\ 0 & \text{sinon} \end{cases}$$

avec ω_i le poids de la catégorie cat_i et $cat(c_j)$ les catégories Wikipedia du concept c_j correspondantes. Une requête mots-clés pouvant être décrite par plusieurs concepts, c'est la combinaison de l'ensemble des concepts obtenant un score de similarité sémantique le plus élevé qui traduira au mieux la requête mots clés de l'utilisateur.

3 Suggestion de couples requête conceptuelle et SRI adaptée

Notre approche consiste à créer une base et une description des différents outils de recherche d'information existants sur le Web et de guider l'utilisateur sur le service de recherche correspondant au besoin exprimé par sa requête.
La description est constitué de l'élément principal *SearchEngine*. Cet élément principal contient les éléments de desciption suivants : *Id*, *Name*, *URL*, *Description*, *ShortDescription* , *Specialized*, *Popularity* et *Searchable* qui indiquent respectivement l'identifiant, le nom, l'url, une description, une description courte, si le service est un service spécialisé ou non, un indice de popularité du service de recherche web et si celui si est interrogeable via son url. Les éléments *ContentType* et *Thematic* permettent de décrire sémantiquement un service de recherche. L'élément *ContentType* décrit un ou plusieurs types de contenu et l'élément *Thematic* représente une ou plusieurs thématiques. Pour garder une cohérence avec la représentation du profil utilisateur et du besoin que nous avons choisi, la valeur du sous-élement *Subject* de l'élément *Thematic* correspond à une catégorie *Wikipédia* de l'ontologie DBpedia et le sous-élement *Type* de l'élément *ContentType* à un concept de l'ontologie DBpedia décrivant un type de média.
A partir de ce schéma de description sémantique d'un service de recherche d'information, nous avons construit une base sémantique et annoté manuellement plus d'une centaine de services de recherche d'information web. Le référencement des services de recherche s'est basé sur la liste des services de recherche proposée par le site web Pandia[1] et sur une veille d'information sur l'actualité du web. Les SRI web généralistes identifiés sont les trois grands leaders de la RI sur le web, Google Search, Yahoo!Search et Microsoft Bing. Les SRI web verticaux référencés sont spécialisés sur un ou plusieurs types de contenu (image, vidéo, blog, etc.) ou sur une ou plusieurs thématiques (juridique, économique, médicale, gouvernemental, scientifique, etc.) .

[1] http ://www.pandia.com/powersearch/index.html

Nous proposons donc la description suivante pour un service de recherche SE:

$$SE = (S(SE), T(SE)) = ((s_1, s_2, \ldots, s_i), (t_1, t_2, \ldots, t_j))$$

où $(s_1, s_2, \ldots, s_i)$ est l'ensemble des concepts décrivant les thématiques et $(t_1, t_2, \ldots, t_i)$ est l'ensemble des concepts décrivant les types de média associés au service de recherche.

Ensuite nous proposons une approche de suggestion de couples constitués d'une requête thématique et d'un service de recherche adapté. La construction de ce couple repose sur la définition d'une fonction d'appariement entre une requête conceptuelle et un service de recherche sémantique. Nous avons implémenté dans notre prototype une fonction de similarité proche de la mesure proposée par Resnik [10] qui permet de déduire la valeur informelle entre deux concepts. Ainsi pour une requête $R_c = (c_1, c_2, \ldots, c_i)$ et un service de recherche *SE*, la fonction d'appariement est définie par :

$$Sim(R_c, SE) = \frac{1}{|R_c|} \sum_{i=0}^{|R_c|} sim(c_i, SE)$$

Le score de pertinence entre la requête conceptuelle et un service de recherche est compris entre 0 et 1. Plus le score sera proche de 1, plus le service de recherche sera pertinent par rapport à la requête conceptuelle. Une liste *L** de couples formés d'un appariement requête conceptuelle-service de recherche sera suggérée à l'utilisateur comme aide à la recherche d'information.

4 Système expérimental

Le but de notre expérimentation est de valider les deux hypothèses de recherche présentées dans les sections 2 et 3 :

- la pertinence de la représentation thématique de profil utilisateur et son apport dans le processus de recherche ;
- la pertinence de notre approche de suggestion de couples (requête conceptuelle, moteur de recherche).

La solution retenue pour la récupération des informations de l'utilisateur est l'utilisation du framework Java Aperture[2], qui permet d'extraire le texte et les méta-données contenus dans des fichiers quelque soit le format (plain text, HTML, XHTML, XML, PDF, Microsoft Office, OpenOffice, ou StarOffice). Le texte récupéré à partir des différentes sources d'information (documents fournis par l'utilisateur, pages web marquées par l'utilisateur, pages web visités par l'utilisateur, favoris) analysées par des outils d'extraction adaptés aux formats des sources est ensuite exploité par un service d'extraction de concepts afin de construire un profil utilisateur sémantique.

[2] http ://aperture.sourceforge.net/

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Nous avons choisi l'outil Zemanta[3] comme extracteur de concepts. Cet outil utilise des techniques statistiques et sémantiques de traitement automatique de la langue pour désambiguiser les entités extraites comme une comparaison statistique par rapport aux bases de connaissance ou une mesure de cohérence thématique entre les entités. Pour notre approche, nous nous sommes restreints aux liens Wikipedia suggérés par l'outil Zemanta, qui correspondent à des concepts ou des entités de Wikipedia afin de pouvoir les aligner sur l'ontologie DBpedia structurant les données de Wikipedia.
Le profil utilisateur est stocké dans un fichier APML[4] (Attention Profiling Mark-up Language). Un concept est décrit en APML par les propriétés suivantes *: key, value, from et updated*. Pour notre représentation du profil utilisateur, la propriété *key* correspond au label d'une catégorie Wikipedia cat_i, la propriété *value* est le poids ω_i correspondant, la propriété *from* est l'URI DBpedia de la catégorie Wikipedia cat_i et la propriété *updated* est la date de mise à jour du concept dans le profil utilisateur.

Le système développé pour l'expérimentation est basé sur la plateforme open source WebLab[5] de développement d'applications dédiées au traitement de documents multimédias. Il met en oeuvre des composants sous forme de Web Services pour le traitement et des portlets intégrées dans le portail permettent la composition de l'interface utilisateur. Cette interface contient différentes pages accessibles sous forme d'onglets dont les plus importantes sont consacrées (1) à la gestion et la visualisation du profil long terme de l'utilisateur et (2) à la suggestion/recherche d'information.

4.1 Architecture

L'interface utilisateur du système est composée de portlets standards WebLab déployées sur le portail open source Liferay[6]. Elle comprend différentes pages accessibles sous forme d'onglets (Fig.4) dont les deux plus importantes sont consacrées à la construction du profil long terme de l'utilisateur et à la recherche d'information.
La figure 1 présente la page dédiée à la construction, la gestion et la visualisation du profil long terme de l'utilisateur qui est composée d'une Portlet. Celle-ci permet à l'utilisateur de lancer la construction des différents modes de construction du profil et de les visualiser. De plus, une case à cocher va permettre à l'utilisateur de sélectionner les concepts, les mots-clés ou les thématiques qu'il considère comme pertinent pour nous permettre d'évaluer la pertinence des différents modes de représentation du profil long terme. Enfin, une section affiche la liste des documents utilisés pour la construction du profil long terme et une case à cocher permet la

[3] http ://www.zemanta.com/
[4] http ://apml.pbworks.com/w/page/10312542/FrontPage
[5] http://weblab-project.org
[6] http://www.liferay.com/

suppression d'un ou des documents sélectionnés et met à jour le profil. L'utilisateur peut fournir des documents au système pour la construction du profil long-terme à partir d'une page dédiée ou en utilisant l'extension Firefox de suivi des activités de l'utilisateur.

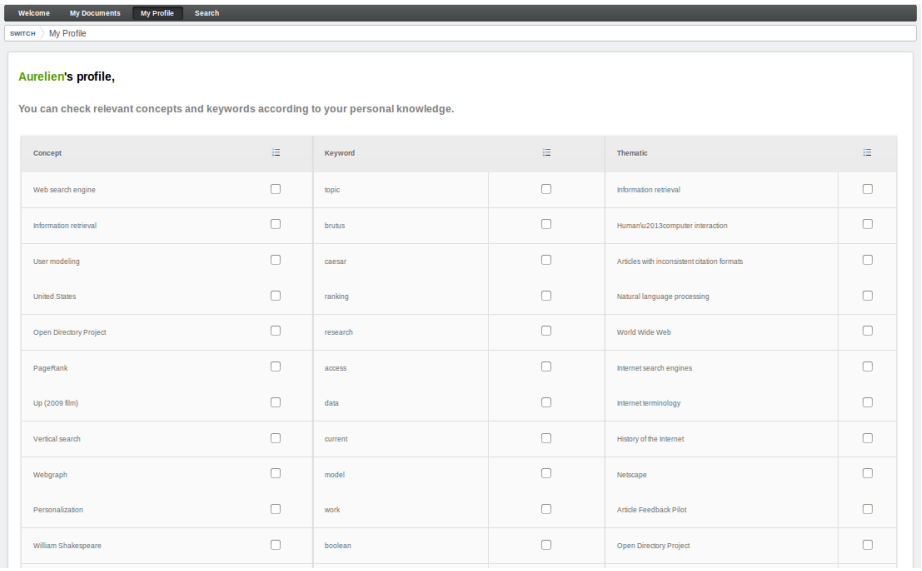

Fig. 4. Aperçu de l'interface de construction, de gestion et de visualisation du profil long terme.

La figure 5 quant à elle, illustre la page consacrée à la recherche. Celle-ci est composée de deux Portlets :

- Recherche : Cette Portlet permet à l'utilisateur d'écrire et de soumettre une requête plein texte. Des suggestions lui sont proposées par les deux outils de suggestion au fur et à mesure de sa frappe. L'utilisateur a aussi la possibilité de construire une requête en glissant/déposant des concepts et un moteur de recherche à partir des suggestions proposées par l'outil de suggestion sémantique. Enfin, un historique des suggestions ou des requêtes soumises est proposé à l'utilisateur.
- Profil court-terme : Cette Portlet permet à l'utilisateur de visualiser les thématiques du profil court-terme déduites à partir des pages web consultées pendant la session de recherche. Trois actions sont disponible à l'utilisateur : « play », « pause » et « stop ». L'utilisateur peut à tout moment mettre en pause ou reprendre l'analyse du contexte de la recherche via les actions « play » et « pause ». Il peut aussi stopper l'analyse du contexte par le bouton « stop », qui réinitialisera le profil court-terme.

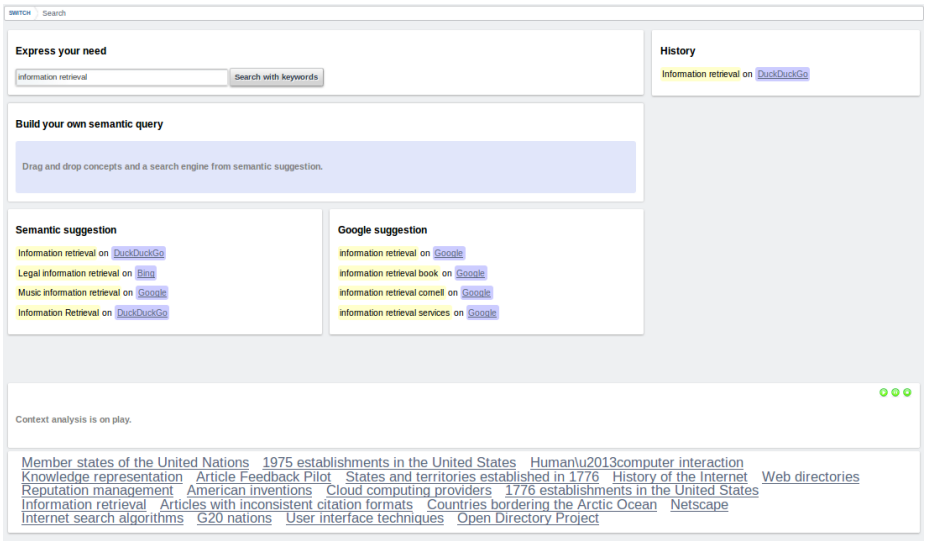

Fig. 5. Aperçu de l'interface utilisateur de recherche du système expérimental

L'interface est relativement simple et cherche à se rapprocher du standard des interfaces classiques d'un moteur de recherche afin de ne pas perturber les utilisateurs. Des capteurs spécifiques (fonctions Javascript) ont été ajoutés afin de pouvoir stocker en temps réel les traces utilisateur. Une trace sera constituée : d'un timestamp, de la requête originale, d'une requête suggérée (si sélectionnée), d'un service de recherche, d'un mode de requête (KEYWORDS, BUILD, SEMANTIC, GOOGLE) et du rang de la suggestion (pour les modes SEMANTIC et GOOGLE).
Les modes de requête correspondent à :

- KEYWORDS : requête mots-clés envoyés soumise au moteur ;
- BUILD : requête construite explicitement par glissé/déposé ;
- SEMANTIC : requête issue de l'outil de suggestion sémantique ;
- GOOGLE : requête issue de l'outil de suggestion Google.

Ces traces nous permettront d'évaluer les différentes hypothèses de recherche posées dans cet article.

4.2 Fonctionnalités

Dans notre système expérimental nous avons implémentés trois modules de construction du profil long-terme proposant chacun un mode de construction différent. Les modes de construction sont les suivants : *mots-clés*, *conceptuel* et *thématique*.
Le modèle *mots-clés* est basé sur le calcul de la mesure statistique TF-IDF de chaque terme des documents fournis par l'utilisateur (après lemmatisation et stemmatisation). La représentation du modèle mots-clés peut se formaliser par :

$$P_{mots-clés} = \langle (\omega_1, tf_1), (\omega_2, tf_2), \ldots (\omega_n, tf_n) \rangle$$

avec ω un terme du corpus de documents fournit par l'utilisateur et tf le poids (TF-IDF) associé.
La modélisation *conceptuelle* du profil long terme de l'utilisateur consiste à représenter ce profil par les concepts DBPedia extraits des documents fournis par l'utilisateur. nous l'avons formalisé par:

$$P_{conceptuel} = \langle (c_1, df_1), (c_2, df_2), \ldots (c_n, df_n) \rangle$$

avec c un concept extrait du corpus de documents et df le poids correspondant à la proportion de documents du corpus qui contiennent le concept.
La modélisation *thématique* du profil long-terme est formalisée par:

$$P_{thematique} = \langle (cat_1, \omega_1), (cat_2, \omega_2), \ldots (cat_i, \omega_i) \rangle$$

où cat_i est un concept SKOS décrivant une catégorie Wikipedia et ω_i le poids du concept correspondant.

4.3 Moteur de suggestion

En plus de ces différentes fonctionnalités, nous avons intégré au système 2 modes de suggestions de couples composés d'une requête et d'un moteur de recherche. Ces composants ont été nommés *semantic* et *google*.
Le mode *semantic* met en oeuvre la nouvelle approche de suggestion de couples (requête sémantique, moteur de recherche) décrite précédemment et qui se base sur une personnalisation du besoin de l'utilisateur avec un appariement sémantique du besoin à un moteur de recherche. Pour notre expérimentation, 110 moteurs de recherche ont été décrits sémantiquement de façon manuelle et couvrent un large ensemble de thématiques (sciences, médecine, musique, etc.) et de types de média (vidéo, image, etc.). Une liste *L** de couples formés d'un appariement requête conceptuelle/service de recherche sera suggérée à l'utilisateur comme aide à la recherche d'information. Pour notre étude, nous avons fixé la taille de la liste à 4 couples suggérés.
Le mode *google* repose comme son nom l'indique sur les suggestions de requêtes fournies par le service de recherche Google. Les suggestions renvoyées reflètent les activités de recherche de l'ensemble des internautes et le contenu des pages Web indexées par Google. Afin d'avoir un outil de suggestion comparable à l'outil proposant notre approche, nous avons associé à chaque requête suggérée le service de recherche Google pour former des couples (requête, service de recherche).

5 Conclusion

Dans cet article, nous avons présenté une approche d'aide à la recherche d'information originale: la suggestion de couples formés d'une requête conceptuelle et d'un service de recherche. Dans le but de réaliser cette suggestion, nous avons développé deux contributions : (1)la personnalisation du besoin utilisateur par une traduction d'une requête mots-clés en une requête conceptuelle personnalisée et (2)

une approche de suggestion d'un couple composée d'une requête conceptuelle et un service de recherche. Dans un premier temps, nous avons donc proposé une modélisation des centres d'intérêts de l'utilisateur par un profil utilisateur. Ce profil utilisateur est décomposé en deux sous-profils qui distinguent les centres d'intérêts long-terme (les connaissances de l'utilisateur) et les centres d'intérêts court-terme (le contexte de la recherche courante). Nous avons choisi de représenter chaque sous profil utilisateur par un vecteur sémantique ou les dimensions du vecteurs correspondent aux catégories thématiques de l'ontologie DBPedia. La pondération de chaque dimension correspond à une probabilité d'apparition dans les sources d'information ayant permis la construction du profil. Dans le cas du profil long-terme, nous avons utilisé des documents jugés représentatifs des centres d'intérêt par l'utilisateur lui-même et les pages web marquées comme favorites. Afin d'utiliser un profil tenant compte des intérêts court et long terme, nous avons défini une fonction de fusion pour obtenir un profil regroupant les informations provenant des deux types de profil.

Nous avons proposé une approche de personnalisation du besoin utilisateur en exploitant son profil sémantique. La personnalisation se traduit par la transformation de la requête exprimée sous forme de mots-clés en plusieurs requêtes conceptuelles, où les concepts sont ordonnés par une mesure de similarité sémantique en fonction du profil utilisateur.

A partir du besoin de l'utilisateur transformé en une requête sémantique, nous proposons une approche de suggestion de couple (requête, service de recherche) basée sur une fonction d'appariement d'une requête conceptuelle avec un service de recherche. Pour cela, une description sémantique d'un service de recherche a été définie et une modélisation mathématique d'un service de recherche sémantique a été réalisée. Enfin nous avons défini une fonction d'appariement basée sur une mesure de similarité sémantique qui permet de former des couples composés d'une requête conceptuelle et d'un service de recherche afin des les suggérer à l'utilisateur pour l'aider dans son processus de recherche d'information.

References

1. G. Salton and C. Yang. On the specification of term values in automatic indexing. 1973.
2. A. Sieg, B. Mobasher, S. Lytinen, and R. Burke. Using concept hierarchies to enhance user queries in web-based information retrieval. In in Proceedings of the International Conference on Artificial Intelligence and Applications, IASTED 2004, 2004.
3. A. Pretschner and S. Gauch. Ontology based personalized search. In ICTAI '99 : Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, page 391, Washington, DC, USA, 1999. IEEE Computer Society.
4. X. Li, Y.-Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In SIGIR '08 : Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 339–346, New York, NY, USA, 2008. ACM.
5. J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, SIGIR '09, pages 315–322, New York, NY, USA, 2009. ACM.

6. A. Sutcliffe and M. Ennis. Towards a cognitive theory of information retrieval. Interacting with Computers, 10(3) :321 – 351, 1998. HCI and Information Retrieval.
7. P. N. Bennett, R. W. White, W. Chu, S. T. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of short- and long-term behavior on search personalization. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12, pages 185–194, New York, NY, USA, 2012. ACM.
8. D. J. Liebling, P. N. Bennett, and R. W. White. Anticipatory search : using context to initiate search. In SIGIR, pages 1035–1036, 2012.
9. D.Milne, I. H. Witte, Learning to link with wikipedia, In Proceeding of the 17th ACM conference on Information and knowledge management, CIKM, Oct 2008.
10. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers.

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

L'impact de l'enrichissement sémantique sur la classification de textes: Application au domaine médical

Shereen Albitar – Sébastien Fournier – Bernard Espinasse

Aix-Marseille Université, CNRS, LSIS UMR 7296, 13397, Marseille, France
{prénom.nom@lsis.org}

Résumé. L'utilisation de la sémantique dans la classification supervisée de texte peut améliorer son efficacité, en particulier, dans des domaines spécifiques. La plupart des travaux utilisent les concepts comme une alternative aux mots et transforment le classique sac de mots (BOW) en un sac de concepts (BOC). Cette transformation se fait à travers la tâche de conceptualisation. De plus, le BOC peut être enrichi par l'utilisation de concepts connexes par la prise en compte de ressources sémantiques pouvant ainsi améliorer l'efficacité de la classification. Cet article se focalise sur l'étude de l'impact, pour la classification supervisée de texte, de l'application d'une stratégie d'enrichissement sémantique à une représentation de texte déjà conceptualisée. Cette stratégie est basée sur une méthode d'enrichissement mutuel des vecteurs. Nous présentons une étude expérimentale pour évaluer cette stratégie d'enrichissement sémantique en utilisant la méthode de classification supervisée Rocchio dans le domaine médical, en utilisant l'ontologie UMLS (Unified Medical Language System) et le corpus Ohsumed. Grâ4ce à l'enrichissement sémantique, les résultats démontrent des améliorations significatives sur la classification de textes dans l'espace des concepts.

Mots-clés: classification supervisée de texte, sémantique, conceptualisation, enrichissement sémantique, mesures de similarité sémantique, domaine médical, UMLS, Rocchio

1 Introduction

La classification supervisée de textes est actuellement un sujet à la pointe de la recherche, en particulier dans des domaines tels que la recherche d'information, de la recommandation, de la personnalisation, des profils d'utilisateurs, etc. Parmi les méthodes les plus populaires pour la classification de texte, nous citons notamment la méthode Bayésienne (NB), les Machines à vecteurs de support (SVM) et Rocchio ou bien la classification basée sur les centroïds. Malgré leur popularité et les résultats corrects qu'elles affichent, ces méthodes, utilisant les sacs de mots (BOW) pour la représentation de texte, souffrent d'un manque de sémantique au niveau de la représentation de texte et ignorent tout aspect sémantique présent au sein du texte. Elles souffrent aussi d'un manque de sémantique au niveau du processus de classification lui-même. En outre, comme le montre [1], ces méthodes ont aussi des problèmes pour

gérer les classes larges (c'est-à-dire dont le spectre sémantique est étendu) et les classes peu peuplées (ayant peu d'exemples d'apprentissage). Ces méthodes ont aussi plus de difficultés à effectuer la tâche de classification lorsqu'elle est réalisée dans un domaine spécifique. Afin de résoudre ces différents types de problème, nous pensons que l'emploi de la sémantique semble être le plus approprié. De plus, de nombreux travaux montrent que l'utilisation de la sémantique dans la classification de texte peut améliorer son efficacité en particulier dans des domaines spécifiques [2, 3]. Dans le but d'utiliser la sémantique pour la classification supervisée de texte, plusieurs options s'offrent à nous. Il est possible d'utiliser la sémantique avant l'indexation, avant et après l'apprentissage et au moment de la prédiction de la classe. Toutefois, même si l'emploi de la sémantique semble prometteur, il nous semble important de mieux cerner dans quel cadre il est intéressant de l'employer, c'est-à-dire dans quel cas le gain est significatif par rapport aux méthodes classiques.

Dans ce travail, au travers un certain nombre d'expérimentations, nous essayons d'estimer l'impact d'une méthode d'enrichissement sémantique de la représentation sur la classification supervisée de texte. Il s'agit de la méthode *« Enriching Vectors »*. Dans ces travaux, nous avons choisi d'utiliser Rocchio [4], même s'il est moins performant que SVM pour certaines tâches, pour sa relative efficacité et sa simplicité en plus de son extensibilité par rapport à l'utilisation des ressources sémantiques dans le modèle d'apprentissage. En effet, Rocchio est capable d'utiliser la sémantique aussi bien dans sa représentation de texte au travers l'utilisation des sacs de concepts (BOC) qu'avant ou après l'apprentissage jusqu'à la phase de prédiction. Il est donc capable d'utiliser tout le spectre possible de l'implication de la sémantique dans la tâche de classification. Les expériences que nous comptons réaliser afin de tester cette méthode que nous présentons sont effectuées dans un domaine spécifique : le domaine médical. Pour cela, nous utilisons le corpus Ohsumed et la base de connaissances UMLS.

Dans la section 2, nous présentons un bref état des lieux des méthodes de classification de texte utilisant la sémantique. Ensuite, dans la section 3, nous présentons un cadre conceptuel général pour l'intégration de la sémantique dans le processus de la classification supervisée de texte en utilisant une stratégie d'enrichissement sémantique à partir d'une représentation BOC. Dans la section 4, nous présentons la stratégie d'enrichissement, basé sur l'enrichissement par la méthode des vecteurs enrichis « *Enriching Vectors* ». Dans la section 5, nous présentons brièvement Rocchio, les ressources sémantiques, le corpus Ohsumed, et les outils utilisés dans cette recherche. La section 6 présente notre processus d'expérimentation. Ensuite dans la section 7, nous présentons les résultats obtenus. Enfin, nous terminons par une évaluation de notre travail, suivi par différentes perspectives de recherche.

2 Classification supervisée de texte par usage de la sémantique

Selon la littérature, de nombreux travaux proposent des approches impliquant la sémantique dans la classification de texte à différents niveaux, par exemple, en faisant valoir l'utilité de la sémantique dans la représentation de texte [2, 5]. La plupart de ces

travaux ont transformé le classique sac de mots (BOW) représentant le texte dans l'espace vectoriel en sac de concepts (BOC) en choisissant les concepts comme une caractéristique alternative aux mots [3, 6]. Ils sont alors appliqués lors de l'indexation. D'autres travaux utilisent la similarité sémantique entre concepts ainsi que l'enrichissement de la représentation par les concepts. Ils sont généralement appliqués après l'indexation mais avant la prédiction. Trois grandes approches se distinguent pour l'enrichissement de la représentation du texte : (i) les noyaux sémantiques - généralement employés par les classifieurs SVM [2, 5, 7] , (ii) la généralisation [3], et (iii) l'enrichissement de vecteurs [6]. Cependant, les auteurs de [3] concluent que l'application de la généralisation pour les tâches de classification appliquées à un domaine spécifique provoque une détérioration de la performance. Enfin, il est possible d'impliquer la sémantique au niveau de la prise de décision en utilisant par exemple des mesures de similarité sémantique entre textes [8]. Dans cet article, notre travail se focalise plus particulièrement sur l'enrichissement de vecteurs.

3 Un cadre conceptuel pour la classification par enrichissement sémantique

La **Fig. 1** propose un cadre conceptuel résumant l'approche présentée dans cet article, impliquant la sémantique dans le processus de classification supervisée de textes. Dans cette approche, la sémantique est impliquée dans les différentes étapes du processus de classification : En premier, elle est impliquée lors de l'indexation au travers de la conceptualisation, puis en appliquant l'utilisation de la sémantique après l'apprentissage. La conceptualisation est le processus de recherche et de correspondance d'un concept pertinent provenant d'une ressource sémantique et qui traduit le sens d'un ou plusieurs mots d'un texte. Les concepts couvrant un document texte composent alors le vecteur sémantique qui représente le document en tant que BOC. L'utilisation de la sémantique après l'apprentissage se fait grâce à un enrichissement sémantique par l'usage de mesures de similarité sémantique.

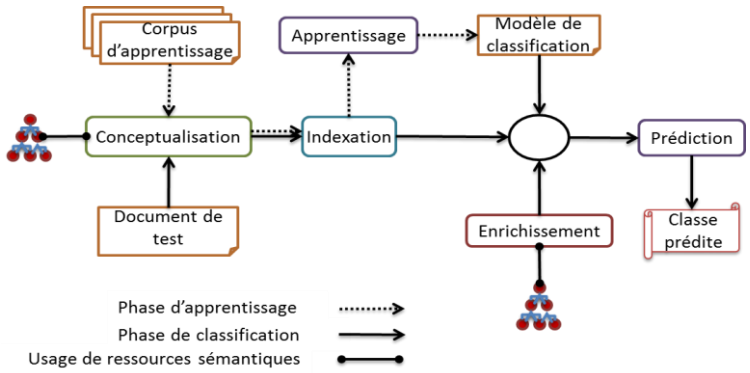

Fig. 1. Un framework conceptuel pour l'intégration de la sémantique dans la classification supervisée de textes

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Dans ce travail, nous avons l'intention d'investiguer cette stratégie d'enrichissement qu'est « *Enriching Vectors* » en l'appliquant au domaine médical afin d'évaluer leur influence sur la classification supervisée de texte. L'étape de conceptualisation du texte est réalisée grâce à des travaux présentés dans [1, 9]. Ainsi, nous impliquons les connaissances sémantiques au niveau de l'indexation par l'utilisation de concepts au niveau de la représentation même du texte.

4 La méthode « Enriching Vectors »

Les auteurs dans [6] ont proposé cette méthode et l'ont appliquée pour de la catégorisation en utilisant K-Means et pour de la classification en utilisant kNN. Afin de comparer deux documents, les auteurs appliquent cette méthode sur les vecteurs représentant ces documents et ensuite applique une mesure de similarité classique comme Cosinus pour prédire la classe. Selon les auteurs, cette méthode a montré une meilleure corrélation avec un jugement humain, par rapport à l'application de la mesure de similarité classique sur les vecteurs d'origine sans enrichissement.

Les mesures de similarité classiques habituellement déployées pour comparer des documents de texte représentés dans l'espace vectoriel comme Cosinus dépendent d'une correspondance lexicale. En fait, ces mesures tiennent principalement compte des caractéristiques communes entre les vecteurs négligeant d'autres similitudes telles que la similarité sémantique entre des caractéristiques non partagées. En d'autres termes, si deux textes ne partagent pas les mêmes mots mais utilisent des synonymes, ils sont présumés dissemblables. Cet inconvénient a, entre autres, été souligné par [1].

Pour aller au-delà de la correspondance lexicale, nous avons l'intention d'appliquer *« Enriching Vectors »* à chaque paire de vecteurs avant la comparaison : chacun des vecteurs enrichit l'autre vecteur en utilisant ses caractéristiques exclusives. Étant donné deux documents A, B représentés à l'aide d'un vocabulaire de plusieurs concepts. Nous notons qu'une caractéristique est exclusive pour B, si elle est en correspondance avec un ou plusieurs mots du document B uniquement (la caractéristique n'est pas présente dans le document A) et réciproquement. Comme le montre la **Fig. 2**, l'objectif principal de cette approche est d'introduire les caractéristiques exclusives de B (C_2) dans A et de leur attribuer des poids appropriés en tenant compte des caractéristiques de A et vice versa. Ces pondérations sont estimées en utilisant les pondérations des autres caractéristiques du document traité et en utilisant la similarité sémantique entre ces caractéristiques et la caractéristique manquante. Pour ce faire, nous utilisons une matrice de proximité sémantique composée des similarités sémantiques entre les concepts du vocabulaire pair-à-pair.

Les nouveaux poids des concepts dans les vecteurs enrichis sont calculés comme suit :

$$w(c,A) = w\big(SC(c,A)\big) * sim\big(c, SC(c,A)\big) * CC(c,A)$$

Où $w\big(SC(c,\mathrm{A})\big)$ est le poids de la plus forte connexion du concept c (Strongest Connection) dans A ce qui correspond au poids du concept le plus similaire à c.

$sim\big(c, SC(c,\mathrm{A})\big)$ est la mesure de similarité entre c et le concept ayant la plus forte connexion dans A.

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

$CC(c, \mathrm{A})$ est la centralité contextuelle (CC) du concept c dans le document A qui est donné par la formule suivante :

$$CC(c, A) = \frac{\sum_{c_i \in A} sim(c, c_i) * w(c_i, A)}{\sum_{c_i \in A} w(c_i, A)}$$

Où :
$sim(c, c_i)$ est la similarité sémantique entre le concept c et c_i du document A.
$w(c_i, \mathrm{A})$ est le poids du concept c_i dans le document A.

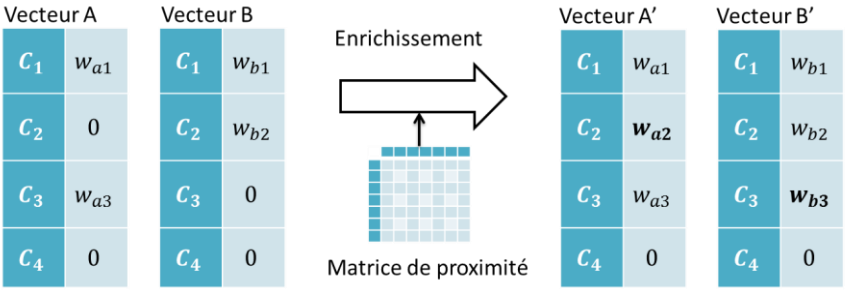

Fig. 2. Exemple d'enrichissement de vecteurs par la méthode *« Enriching Vectors »*

5 Les ressources et outils utilisés

Dans ces travaux, nous avons besoin des ressources sémantiques et de plusieurs outils. Ces derniers sont présentés dans la suite de cette section.

Nous utilisons dans nos travaux ***Rocchio*** pour la classification supervisée de textes. Dans Rocchio [4], chaque classe est représentée par un vecteur centroïd. Les centroïds obtenus lors de l'apprentissage représentent un modèle de classification qui résume les caractéristiques des documents de chaque classe. Au cours de la phase de classification, chaque document de test est comparé aux centroïds en utilisant des mesures de similarité non sémantique afin de lui attribuer la classe dont le centroïd est le plus similaire. Il existe un grand nombre de ces mesures de similarité, dans nos travaux nous en utilisons cinq : Cosinus, Jaccard, Kullback-Leibler, Levenshtein et Pearson [10]. L'utilisation de cinq mesures de similarité, au lieu d'utiliser seulement la plus connue : cosinus, nous permet d'estimer la différence d'impact de l'utilisation de la sémantique en fonction de la mesure de similarité et si cette différence est significative ou pas. La similarité, entre deux vecteurs A et B, est calculée selon les formules suivantes :

$$\mathrm{Sim}_{Cosinus}(\mathrm{A}, \mathrm{B}) = \cos(\mathrm{t}) = \frac{\sum_i a_i * b_i}{\sqrt{\sum_i a_i^2} * \sqrt{\sum_i b_i^2}}$$

$$\mathrm{Sim}_{Jaccard}(\mathrm{A}, \mathrm{B}) = \frac{\sum_i a_i * b_i}{\sqrt{\sum_i a_i^2} + \sqrt{\sum_i b_i^2} - \sum_i a_i * b_i}$$

$$Sim_{Levenshtein}(A, B) = 1 - (\sum |a_i - b_i| \,/\, \sum Max(a_i, b_i))$$

$$\text{Sim}_{Pearson} = \frac{n\sum a_i b_i - \sum a_i \sum b_i}{\sqrt{\left[n\sum a_i^2 - (\sum a_i)^2\right]\left[n\sum b_i^2 - (\sum b_i)^2\right]}}$$

$$\text{Sim}_{KullbackLeibler}(A,B) = \sum_i \left(\pi_1 * D(a_i||w_i) + \pi_2 * D(b_i||w_i)\right)$$

Où :

$\pi_1 = \frac{a_i}{a_i + b_i}$	$\pi_2 = \frac{b_i}{a_i + b_i}$
$w_i = \pi_1 * a_i + \pi_2 * b_i$	$D(a_i\|\|w_i) = a_i * \log(\frac{a_i}{w_i})$

Le corpus ***Ohsumed*** [11], utilisé pour l'apprentissage et les tests, est composé de résumés d'articles biomédicaux de l'année 1991 extraits de la base de données MEDLINE et indexés à l'aide de MeSH (Medical Subject Headings). Les premiers 20000 documents de cette base de données ont été sélectionnés et classés en utilisant 23 sous-concepts du concept « Disease ». Le corpus est alors divisé en deux parties : une pour l'apprentissage et l'autre pour les jeux d'essai. Dans ce travail, les centroïds des classes sont calculés par Rocchio pour chacune des cinq classes les plus fréquentes d'Ohsumed énumérées dans la **Table 1**.

Category	Description	Training	Test
C04	Neoplasms	972	1251
C06	Digestive System Diseases	588	632
C14	Cardiovascular Diseases	1192	1256
C20	Immune System Diseases	502	664
C23	Pathological Conditions, Signs and Symptoms	976	1181
Total		**4230**	**4984**

Table 1. Le Corpus Ohsumed

Unified Medical Language System (UMLS ®) [12] a été développé afin de modéliser le langage biomédical et celui de la santé. UMLS organise les concepts de différentes sources de vocabulaires (comme MeSH, SNOMED-CT, etc.) selon leurs sens en regroupant des concepts communs. Nous avons choisi, notamment pour des raisons de performance, d'effectuer la conceptualisation de textes en utilisant les concepts de SNOMED-CT uniquement.

En complément des ressources sémantiques comme UMLS, de nombreux outils ont été conçus afin de faciliter l'utilisation de ressources sémantiques pour le développement de systèmes médicaux. Dans ce travail, nous utilisons ***MetaMap*** [13] qui permet de faire la correspondance entre le texte et les concepts présents dans UMLS (et donc aussi dans SNOMED-CT).

L'outil ***UMLS-Similarity*** est un module Perl qui évalue la similarité sémantique entre les concepts d'UMLS. Nous utilisons dans ce travail cinq mesures de similarités sémantique issues de la version UMLS-similarity 1.33. Ces cinq mesures sont basées sur la structure de l'ontologie. Leur simplicité est à l'origine de leur efficacité qui a été

démontrée dans de nombreux domaines d'application dans lesquels les mesures de similarité sémantique sont utilisées [14]. Il s'agit de :

- *cdist* : cette mesure compte le nombre d'arêtes entre les concepts [15]. Son domaine de définition est compris entre zéro et deux fois la profondeur de l'ontologie. L'équation est la suivante :

  $$Sim_{Rada}(c_1, c_2) = min_{i\in[1,N]} |path_i(c_1, c_2)|$$

  Où :
  $path_i$ est le nombre de nœuds entre c_1 et c_2
  i est dans le domaine [1,N], N est le nombre de chemins possibles entre ces concepts dans l'ontologie.

- *wup* : cette mesure est calculée par deux fois la profondeur de la généralisation commune la plus spécifique des concepts (msca), divisé par la somme des profondeurs des concepts [16]. Son domaine de définition se situe entre zéro et un.

  $$Sim_{w\&P}(c_1, c_2) = \frac{2*depth(c)}{depth(c_1)+depth(c_2)} = \frac{2H}{N1+N2+2H}$$

  Où
  $N1$ et $N2$ correspondent aux nombres de connexions IS-A entre le concept commun le plus spécifique et c_1 et c_2 respectivement.
  H est le nombre de liens IS-A entre c et la racine de l'ontologie.

- *lch* : Cette mesure est le logarithme négatif du plus court chemin entre deux concepts divisé par deux fois la profondeur totale de l'ontologie [17]. Son domaine de définition va de 0 à la profondeur de l'ontologie.

  $$Sim_{lch}(c_1, c_2) = \text{Max}\left[-\log\left(\frac{Sim_{Rad}}{2D}\right)\right]$$

  Où : D est la profondeur maximum de l'ontologie et Sim_{Rad} est la similarité *cdist*.

- *zhong* : Cette mesure est la somme de la différence entre la « *milestone* » du msca et celle de chacun des concepts [18]. La « *milestone* » est un facteur calculé et est liée à la spécificité des concepts. Sa gamme se situe entre zéro et un.

  $$\text{milestone(c)} = \frac{1/2}{k^{depth(c)}}$$

  Où :
  Depth(c) est la profondeur du nœud c dans la hiérarchie
  k est une constance généralement de valeur 2.
  La distance est alors calculée comme suit :

  $$dc(c1, c2) = dc(c1, msca(c1, c2)) + dc(c2, msca(c1, c2))$$

  Où : $msca(c1, c2)$ est le plus proche parent commun de c_1 et c_2

$$dc(c, msca) = milestone(msca) - milestone(c)$$

- *nam* : c'est le logarithme d'une formule du chemin le plus court entre les deux concepts, et la profondeur de la taxonomie moins la profondeur du concept *msca* [19]. Son domaine dépend de la profondeur de la taxonomie.

$$Sim_{nam}(c_1, c_2) = log\left((Sim_{Rada} - 1)^{\alpha} * \left(Dc - depth\left(msca(c_1, c_2)\right)\right)^{\beta} + K\right)$$

Pour des raisons de performance, nous avons introduit un nouvel outil : la matrice de proximité sémantique (**Fig. 2**). La matrice de proximité sémantique est une matrice carrée, dans laquelle chaque cellule correspond à la similarité sémantique entre chaque paire de concepts qui se trouvent dans l'index construit à partir de l'ensemble des documents. Ainsi, les similarités sémantiques sont utilisées au travers cette matrice de proximité sémantique dans le but d'enrichir, mutuellement, les représentations vectorielles.

6 Processus d'expérimentation

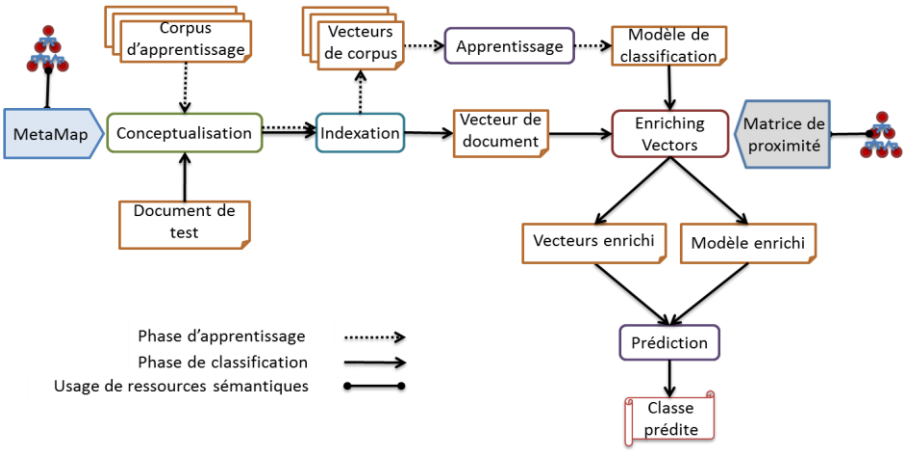

Fig. 3. L'enrichissement sémantique en utilisant « *Enriching Vectors* »

Afin d'évaluer l'effet de la méthode « *Enriching Vectors* » sur le processus de classification de texte à l'aide de Rocchio, nous utilisons la plate-forme expérimentale illustrée par la **Fig. 3**. Cette plate-forme utilise Rocchio pour l'apprentissage et pour la prédiction. L'étape de la conceptualisation est réalisée en amont par l'outil MetaMap avant d'effectuer l'étape d'indexation. Ainsi, les mots dans les documents du corpus sont entièrement substitués par les concepts trouvés par MetaMap ce qui permet d'indexer le corpus en tant que BOC. Pendant l'étape d'enrichissement, le vecteur du document de test est comparé à chacun des centroïds appris pendant l'apprentissage dans l'espace de concepts. Ils sont alors mutuellement enrichis en utilisant la matrice de proximité sémantique de l'une des cinq mesures de similarité sémantique. Après

cet enrichissement, les vecteurs traités sont moins espacés dans l'espace et partagent plus de caractéristiques communes (concepts). Enfin, l'étape de prédiction applique l'une des mesures de similarité classiques et les résultats sont ensuite évalués.

Dans ces expériences, la plate-forme exécute l'apprentissage une fois. Durant la classification, nous utilisons pour chaque expérimentation une des cinq mesures de similarités sémantiques pour l'enrichissement (cdist, lch, nam, wup, zhong) et une variante de Rocchio en utilisant une des cinq mesures de similarité classiques.

7 Résultats

Les résultats détaillés des exécutions qui sont liés à chaque mesure de similarité sémantique sont regroupés afin d'analyser l'impact d'« Enriching Vectors » sur l'efficacité des cinq variantes de Rocchio. Les résultats des cinq variantes sont illustrés par la **Fig. 4**. Les résultats de l'expérimentation conduisent à soulever les points suivants :

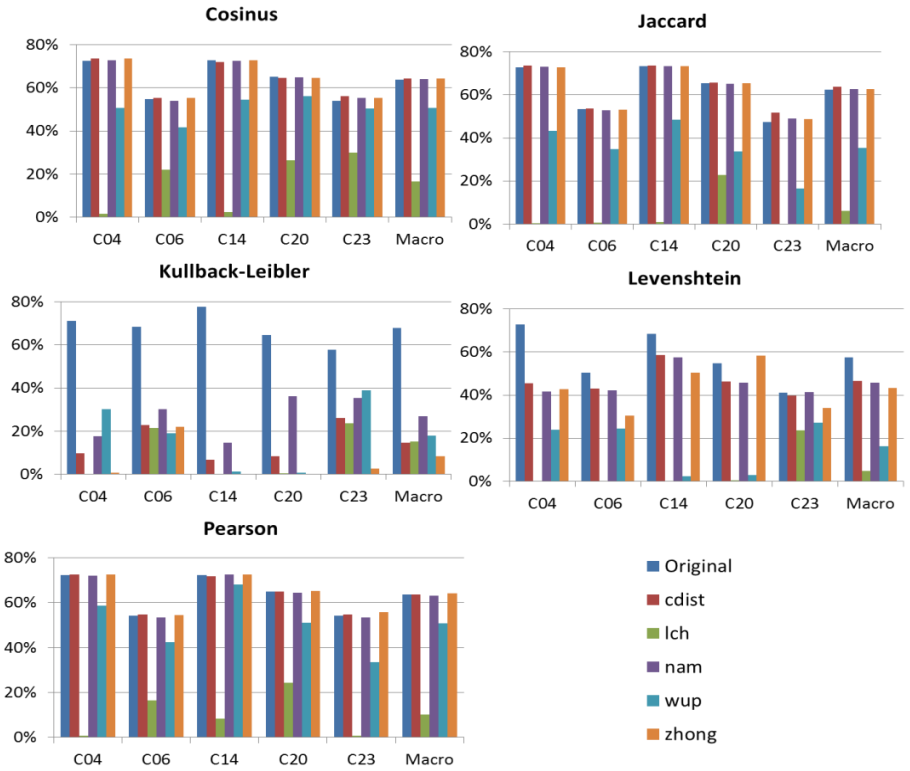

Fig. 4. F1-Measure avant et après l'application de la méthode « *Enriching Vectors* » en utilisant cinq mesures de similarité sémantique

Tout d'abord, dans tous les cas, l'utilisation des mesures de similarités lch et wup a causé une détérioration des performances de Rocchio tandis que les autres mesures de similarité ont montré des améliorations. Notons que le seul aspect que partagent cdist,

nam, et zhong est d'avoir un domaine de fonction (entre 0 et 1) par rapport à lch et wup ce qui peut justifier l'influence différente qu'ils peuvent avoir sur la représentation de texte. La meilleure performance globale a été obtenue à l'aide de la variante de Rocchio utilisant cosinus et zhong avec une macro moyenne de la F1-mesure de (64,33 %). Cette valeur est plus élevée que celle rapportée dans [6], qui est de (59,1%), où les auteurs ont testé « *Enriching Vectors* » sur un petit corpus extrait d' Ohsumed en utilisant le classifieur kNN.

Deuxièmement, on distingue deux groupes de variantes de Rocchio selon leur performance après l'application d'*« Enriching Vectors »* : le premier groupe contient Cosinus, Jaccard et Pearson et le second contient KullbackLeibler et Levenshtein. La principale différence entre ces deux groupes est que le premier évalue la similarité entre les vecteurs en utilisant leurs concepts communs tandis que le second dépend du nombre de leurs concepts discriminants afin d'évaluer leurs similarités. En général, « *Enriching Vectors »* vise à réduire la dispersion de la représentation des textes dans l'espace de concepts, ce qui semble aider le premier groupe dans l'évaluation des similarités. Au contraire, cet enrichissement semble être néfaste pour l'évaluation des similarités qui se basent sur les concepts discriminants entre les vecteurs.

Troisièmement, lorsqu'un système de classification classique obtient une faible valeur de la F1-mesure, *« Enriching Vectors »* a pu améliorer cette valeur. En effet, c'est le cas de la classe (C23) dont les résultats sont détaillés dans la **Table 2**. Le gain maximal a atteint (9,45%) dans le cas de la variante de Rocchio utilisant Jaccard après l'enrichissement des vecteurs en utilisant la mesure cdist. Ces résultats sont similaires à nos observations lors de l'application de la conceptualisation [9]. En fait, la classe « C23 » est sémantiquement très large par rapport aux autres classes et la représentation enrichie, par des concepts similaires, des documents et du modèle abouti à une meilleure identification de cette classe, ce qui a conduit à de meilleurs résultats.

Enrichissement	Cosinus		Jaccard		Kullback-Leibler		Levenshtein		Pearson	
BOC Original	**53,96**		**47,40**		**57,69**		**41,03**		**54,20**	
cdist	56,17	***+4,10****	51,88	***+9,45****	26,04	-54,86	39,69	-3,28	54,73	***+0,97****
lch	29,84	-44,69	0,00	-100,00	23,71	-58,89	23,50	-42,74	0,67	-98,76
nam	55,37	***+2,63****	49,16	***+3,71****	35,46	-38,52	41,32	***+0,69***	53,30	**-1,65**
wup	50,46	-6,48	16,61	-64,97	38,95	-32,48	27,15	-33,83	33,47	-38,25
zhong	55,26	***+2,41****	48,73	***+2,79****	2,58	-95,52	33,89	-17,41	55,73	***+2,82****

Table 2. Résultats de classification des documents de la classe C23. Les valeurs sont des F1-mesure (pourcentage). Les * signifient que les améliorations sont significatives selon McNemar

Enfin, il semble bénéfique à la classification basée sur Rocchio d'appliquer *« Enriching Vectors »* avant la prédiction car le comportement du classifieur semble être modifié et peut améliorer son efficacité. Cependant, le bénéfice obtenu est fonction de la mesure de similarité sémantique utilisée pour l'enrichissement et également de la mesure de similarité utilisée pour la prédiction. Par conséquent, il est nécessaire d'investiguer expérimentalement les bénéfices obtenus pour vérifier si *« Enriching Vectors »* est utile dans un contexte particulier.

8 Conclusion

A travers des expériences dans le domaine biomédical avec le corpus Ohsumed, l'ontologie UMLS, et la méthode de classification supervisée Rocchio, nous avons essayé d'estimer l'impact d'une stratégie d'enrichissement sémantique pour la classification supervisée de textes.

L'enrichissement, réalisé après l'entraînement et avant la prédiction des classes, est basée sur la méthode *« Enriching Vectors »*. Les résultats obtenus après l'enrichissement sont meilleurs de ceux obtenus sur les BOCs sans enrichissement pour plusieurs classes de documents. Cette amélioration est significative particulièrement pour la classe « C23 » qui est une classe large et hétérogène difficile à classer. Néanmoins, ces améliorations dépendent très largement de la mesure de similarité sémantique utilisée dans l'enrichissement et de la mesure de similarité utilisée pour la prédiction. Nous avons constaté également que l'enrichissement sémantique mutuel des vecteurs est bénéfique en utilisant les mesures de similarité qui se basent sur les caractéristiques communes entre les vecteurs comparés.

Dans de futurs travaux, nous avons l'intention de tester d'autres familles de mesures de similarité sémantique comme les mesures basées sur le contenu d'information (IC) ou basées sur les caractéristiques en les testant sur Ohsumed et sur d'autres corpus médicaux, comme celui de « TREC genomics » ou de « i2b2 ».

9 Références

[1] S. Albitar, S. Fournier, and B. Espinasse, "The Impact of Conceptualization on Text Classification," presented at the Proceedings of the 13th international conference on Web Information Systems Engineering, Paphos, Cyprus, 2012.

[2] S. Aseervatham and Y. Bennani, "Semi-structured document categorization with a semantic kernel," *Pattern Recogn.*, vol. 42, pp. 2067-2076, 2009.

[3] S. Bloehdorn and A. Hotho, "Boosting for text classification with semantic features," presented at the Proceedings of the 6th international conference on Knowledge Discovery on the Web: advances in Web Mining and Web Usage Analysis, Seattle, WA, 2006.

[4] E.-H. Han and G. Karypis, "Centroid-Based Document Classification: Analysis and Experimental Results," presented at the 4th European Conference on Principles of Data Mining and Knowledge Discovery, 2000.

[5] D. Ó. Séaghdha, "Semantic classification with WordNet kernels," presented at the Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers, Boulder, Colorado, 2009.

[6] L. Huang, D. Milne, E. Frank, and I. H. Witten, "Learning a concept-based document similarity measure," *J. Am. Soc. Inf. Sci. Technol.*, vol. 63, pp. 1593-1608, 2012.

[7] P. Wang and C. Domeniconi, "Building semantic kernels for text classification using wikipedia," in *14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Las Vegas, Nevada, USA, 2008, pp. 713-721.

[8] M. Mohler and R. Mihalcea, "Text-to-text semantic similarity for automatic short answer grading," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, 2009, pp. 567-575.

[9] S. Albitar, S. Fournier, and B. Espinasse, "Conceptualization Effects on MEDLINE Documents Classification Using Rocchio Method," in *Web Intelligence*, ed, 2012, pp. 462-466.

[10] A. Huang, "Similarity measures for text document clustering," presented at the Sixth New Zealand Computer Science Research Student Conference, , Christchurch, New Zealand, 2008.

[11] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam, "OHSUMED: an interactive retrieval evaluation and new large test collection for research," in *17th annual international ACM SIGIR conference on Research and development in information retrieval*, Dublin, Ireland, 1994, pp. 192-201.

[12] UMLS®. (2013). *Unified Medical Language System*. Available: http://www.nlm.nih.gov/research/umls/

[13] A. R. Aronson and F. M. Lang, "An overview of MetaMap: historical perspective and recent advances," *J Am Med Inform Assoc*, vol. 17, pp. 229-236, May-Jun 2010.

[14] T. Pedersen, S. V. S. Pakhomov, S. Patwardhan, and C. G. Chute, "Measures of semantic similarity and relatedness in the biomedical domain," *J. of Biomedical Informatics*, vol. 40, pp. 288-299, 2007.

[15] J. E. Caviedes and J. J. Cimino, "Towards the development of a conceptual distance metric for the UMLS," *J. of Biomedical Informatics*, vol. 37, pp. 77-85, 2004.

[16] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," presented at the Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Las Cruces, New Mexico, 1994.

[17] C. Leacock and M. Chodorow, "Combining Local Context and WordNet Similarity for Word Sense Identification," in *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, C. Fellbaum, Ed., ed: The MIT Press, 1998, pp. 265-283.

[18] J. Zhong, H. Zhu, J. Li, and Y. Yu, "Conceptual Graph Matching for Semantic Search," presented at the Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces, 2002.

[19] H. Al-Mubaid and H. A. Nguyen, "A Cluster-Based Approach for Semantic Similarity in the Biomedical Domain," in *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, 2006, pp. 2713-2717.

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Peuplement automatisé d'ontologies par analyse des programmes scolaires

Mahdi Gueffaz[1], Jirasri Deslis[1], Jean-Claude Moissinac[1]
[1] Institut Mines-Télécom; Télécom ParisTech;
CNRS LTCI
46, rue Barrault
75634, Paris Cedex 13
{mahdi.gueffaz, jirasri.deslis, jean-claude.moissinac}@telecom-paristech.fr

Résumé. La construction d'ontologies et l'annotation documentaire sont des traitements très coûteux en temps et en ressources. Plusieurs travaux cherchent à mettre en place des solutions basées sur l'utilisation d'outils linguistiques pour extraire semi automatiquement ou automatiquement les informations pertinentes. Le volume d'information des programmes scolaires est important et une assistance à leur transformation en une ontologie apparaît nécessaire. Le but de nos travaux de recherche est le développement d'outils de création et d'enrichissement d'ontologies avec une assistance semi-automatique. Dans le cadre du projet ILOT[1] (Innovative Learning Object for Teaching) nous proposons une approche composée de trois phases. Une phase de peuplement d'ontologie, une phase d'enrichissement avec des ontologies propres et des ontologies extérieures et une dernière phase pour l'exploitation de données contenues dans l'ontologie dans le cadre des enseignements. Cet article porte principalement sur la première phase.

1 Introduction

Les enseignants sont amenés à manipuler une grande quantité d'informations numériques (texte, documents multimédias, documents composites). Le web sémantique peut faciliter les démarches d'apprentissage en aidant à faire face à la multiplicité et à la complexité des données à traiter. Il offre une extension au web actuel, pour que l'accès aux données pertinentes soit facilité par des automatismes. Il organise et structure l'énorme quantité d'informations contenues dans le web.

Les ontologies sont utilisées pour formaliser la connaissance dans le Web sémantique. Elles sont représentées par un langage qui permet de spécifier une conceptualisation, définie comme une version simplifiée d'un domaine que nous voulons représenter de façon formelle en utilisant des concepts et leurs relations [1]. Un des objectifs des ontologies est la facilitation des échanges de connaissances entre les humains, entre humain et machine ou entre machines.

Dans le cadre du projet ILOT, nous avons décidé de travailler sur la création d'ontologies à partir de corpus de ressources pédagogiques. Dans un premier temps,

[1] http://ilot.wp.mines-telecom.fr/

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

nous avons élaboré une méthodologie en travaillant à la création d'ontologies pour la représentation des programmes scolaires français.

Notre corpus est composé d'une centaine de documents (d'une trentaine de pages par matière) et sa transformation en une ontologie manuellement serait une activité difficile qui nécessiterait des compétences à la fois dans la représentation des connaissances avec les ontologies et des connaissances sur les domaines couverts (pédagogie, histoire, mathématiques...etc.).

Dans cet article, nous proposons une méthode de création et d'enrichissement d'ontologie semi-automatique afin d'enrichir les possibilités de l'enseignant en s'appuyant sur le contenu des programmes scolaires. Dans la section 2, nous présentons les travaux relatifs à cette méthode. La section 3 donne une description du corpus utilisé pour la création de nos ontologies. Nous décrirons dans la section 4 la mise en place de notre solution. Ensuite, nous exposons les résultats de nos premières expérimentations. Enfin, nous commentons ces résultats afin d'apporter des améliorations dans les travaux futurs.

2 Travaux existants

Nous trouvons des travaux voisins dans les domaines de la formation, en particulier la formation en ligne. Plusieurs travaux de recherche ont été identifiés dans le domaine du e-learning exploitant des ontologies. Ils contribuent à confirmer l'intérêt de modéliser un champ de formation pour améliorer les outils de cette formation. Dans cette catégorie, citons : le projet Web Sémantique et E-Learning [2] [3] [4] qui propose des concepts, des méthodes et des outils sur les environnements informatiques pour l'apprentissage humain; le projet de développement de l'environnement de conception de curriculum et de cours présenté dans [5] utilise quatre ontologies constituant la base du curriculum qui exploite une ontologie de capacités, une ontologie d'objectifs, une ontologie de ressources, et une ontologie de liens.

Face à la masse croissante d'informations numériques exploitées, des méthodes automatiques de conception d'ontologie ont été proposées. Différents types d'approches sont distingués selon les types de données en entrée : à partir de texte, de dictionnaires, de base de connaissances, de schémas semi structurés et de schémas relationnels. [6].

L'application Text-To-Onto [7], développée à l'Institut AIFB de l'Université de Karlsruhe, sert à extraire à partir de corpus ou de documents Web des données pour la conception d'ontologie et permet également la réutilisation d'ontologies existantes [8]. OntoBuilder [9] permet de construire une ontologie à partir de ressources Web. L'extraction de l'ontologie à partir de fichiers XML est suivie d'une phase de raffinage guidée par l'utilisateur.

Notre corpus des programmes scolaires français est composé de fichiers XML dont nous voulons obtenir une représentation sous forme d'une ou plusieurs ontologies. Les travaux précédents ont été une source d'inspiration pour notre méthodologie. Dans la littérature, nous avons aussi trouvé des travaux de recherche qui proposent des outils de transformation de fichiers XML en ontologies.

Les travaux de [10] proposent une approche de construction d'ontologie à partir d'un schéma XML et transforment le document XML en graphe RDF. Dans [11] est proposée une approche similaire pour la création d'ontologie OWL à partir de schémas XML. Dans cette approche, les classes OWL sont définies à partir des types complexes du schéma XSD, c'est-à-dire les éléments définis dans le XSD qui contiennent d'autres éléments ou ont au moins un attribut. Quand un élément contient un autre élément, une propriété d'objet (ObjectProperty) est créée entre les classes OWL correspondantes. Les propriétés de type de données (DataTypeProprety) sont définies à partir des attributs XML et à partir des éléments contenant seulement un littéral et pas d'attribut.

[12] proposent un outil X2OWL de transformation de documents XML en ontologie OWL. L'ontologie générée ne contient que les concepts et les liens entre concepts (ObjectProperty). Cette méthode est basée sur le schéma du document XML pour générer la structure de l'ontologie. Cette méthode inclut aussi une étape de raffinement permettant la restructuration de l'ontologie.

L'ensemble de ces travaux ont servis de base à la mise au point de la méthode que nous décrivons dans la suite de cet article.

3 Description du corpus

L'utilisation du programme scolaire dans notre démarche destinée à indexer des ressources éducatives s'est appuyée sur des expériences menées avec des utilisateurs et des résultats de recherche de projets européens dans le domaine de l'éducation, qui nous ont confortés dans l'idée d'utiliser des ontologies.

En premier lieu, citons le retour des expériences du portail Learning Resource Exchange for schools, dans le cadre du projet ASPECT. Cette expérimentation, dont un des objectifs est de vérifier comment les enseignants cherchent et trouvent des ressources, a été effectuée auprès des 44 enseignants venant de plusieurs pays européens [13]. Elle démontre en effet que « 85 % des enseignants définissent la qualité des ressources comme la correspondance entre le contenu et le programme éducatif qu'ils traitent. L'indexation par point de programme améliore la pertinence des résultats de recherche ». [14]. Cela conforte notre proposition de s'appuyer sur nos ontologies pour indexer, en phase de création, les ressources produites dans la plateforme ILOT.

Par ailleurs, nous pouvons également nous référer aux expérimentations du projet européen Intergeo, dédié à éliminer les freins à l'adoption de la géométrie dynamique par les enseignants et à l'utilisation des ressources existantes à travers l'Europe. L'ontologie des compétences GeoSkills [15] dont les informations proviennent de l'extraction des informations du programme scolaire dans le domaine de la géométrie a été ainsi mise en place afin de résoudre les barrières de la langue et obtenir un référencement adapté aux pratiques professionnelles des enseignants.

Notre corpus d'expérimentation est constitué des programmes scolaires de toutes les matières au niveau Collège et Lycée, mis à disposition par le Ministère de l'Education

nationale sur le portail *eduscol*[2]. Le corpus contient des documents PDF et des documents HTML que nous avons transformés manuellement en fichiers XML. Ces programmes ont été conçus comme la trame de la pédagogie pour les enseignants. Ces programmes, consultés par tous les enseignants, deviennent ainsi un langage commun propre de la communauté.

Le corpus contient plusieurs matières à enseigner : Histoire-géographie-éducation civique, Arts plastiques, Education musicale, Education physique et sportive, Langues et cultures de l'Antiquité, Langues vivantes, Français, Histoire des Arts, Mathématiques, Physique-chimie, Sciences de la vie et de la Terre, Technologie. Quantitativement, le corpus est composé de 62 documents en format pdf et en html. Dans l'ensemble, les documents comprennent deux grandes parties. La partie introductive contenant les objectifs et la mise en œuvre des programmes et la partie des contenus de l'enseignement.

Cette dernière partie contient les entités pédagogiques (thème, démarche, capacité, description, etc.) qui seront extraits pour la construction des ontologies des programmes scolaires. Certains programmes comme Histoire, Mathématiques, Sciences de la vie et de la Terre, Technologie ont une représentation semi-structurée sous forme de tables contenant ces informations. Cette mise en forme nous aide à déterminer les relations et/ou définir la hiérarchisation des entités pédagogiques entre elles.

4 Approche

[16] définit le cycle de vie de la génération automatique d'ontologie comme un processus composé de cinq étapes :

- Extraction : fournit les informations à partir du corpus pour constituer l'ontologie ; dans notre processus, une extraction semi-automatique des textes des fichiers PDF a permis une première structuration en XML ;
- Analyse : en partant des résultats de la première étape, cette étape utilise l'analyse morphologique ou lexicale, l'analyse sémantique pour détecter les synonymes, les homonymes et d'autres relations de ce type; de telles techniques nous permettent d'annoter le fichier XML pour en distinguer certaines parties ;
- Génération : cette étape porte sur la formalisation d'un modèle, par exemple avec OWL. C'est l'étape la plus importante du processus. Dans notre projet, nous construisons plusieurs ontologies pour les différents programmes scolaires regroupés par matière. Partant d'une ontologie de base, nous la peuplons par extraction d'information du fichier XML. L'ontologie de base est conçue manuellement pour tous les différents programmes scolaires. La conception de l'ontologie de base a été élaboré par une études détaillée des différents programmes scolaires de chaque matière de tout niveau.
- Validation : toutes les étapes précédentes peuvent introduire des concepts et des relations erronés, pour cela une phase de validation automatique et/ou

[2] http://eduscol.education.fr/pid23391/programmes-de-l-ecole-et-du-college.html

humaine est nécessaire. Nous avons procédé à une validation humaine, notamment avec les raisonneurs de l'outil Protégé ;

- Evolution : une ontologie ne doit pas être une représentation statique d'un domaine, mais doit évoluer avec lui ; cette phase n'a pas encore été abordée dans notre dispositif.

Une phase d'enrichissement avec des ontologies externes est en cours d'élaboration ; nous verrons ci-dessous qu'une première ébauche de cela est obtenue par l'utilisation de DBPedia Spotlight.

4.1. Phase de peuplement

Une classification sur la génération d'ontologie a été proposée dans [16]. Cette classification a regroupé les expériences et les outils en quatre catégories comme suit :

- Conversion ou traduction : il s'agit de logiciels qui assurent la transformation d'un format classique de représentation d'information (par exemple XML) vers une ontologie par des processus limités de traduction de format. C'est dans cet esprit que nous avons décomposé notre travail : la deuxième étape de notre traitement consiste à passer d'une représentation XML à une représentation RDF calquée sur la sémantique du XML initial et pas encore enrichie par des connaissances sur le domaine de connaissances visé.
- Basé sur les extractions : des techniques d'extraction ont été développées afin d'extraire d'informations pour générer une ontologie. La plupart des expériences sont axées sur des sources non structurées, comme des documents textuels ou des pages web et mettent en œuvre des techniques de traitement du langage naturel (TAL). Ces expériences nous disent que la récupération de concepts structurés de documents non structurés nécessite toujours une assistance humaine et que les techniques d'extraction de textes en langage naturel peuvent être utilisées en complément d'autres représentations existantes de connaissances structurées. Ce type d'outil inspire en partie la première étape de notre méthode : marquage sémantique local d'éléments dans un fichier source XML.
- Basé sur des connaissances externes : cette catégorie concerne les applications qui construisent ou enrichissent une ontologie de domaine par des ressources extérieures. Cette catégorie est classée avec les approches d'intégration des dictionnaires externes, des ontologies existantes ou des connaissances structurées (WordNet ou DBpedia). Ce type d'outil inspire une partie de la première étape de notre traitement (où nous cherchons des 'capacités' au sens des travaux de Bloom).
- Framework : regroupe les outils d'édition d'ontologie comme Protégé ; il s'agit là seulement d'une assistance à la création manuelle ou à la vérification d'ontologies. Protégé nous a servi à établir par nous-mêmes quelques ontologies de base pour notre travail.

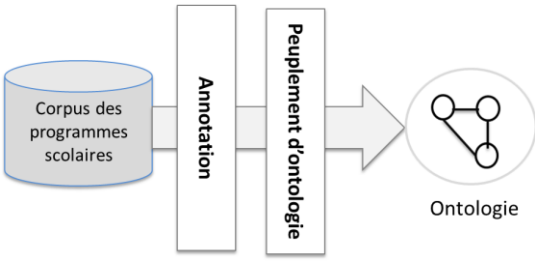

Figure 1. Architecture de la phase de peuplement d'ontologie.

Dans les programmes scolaires figurent des descriptions d'objectifs associés à chacune des sections des programmes. Comme c'est l'usage pour décrire ce type d'objectifs, les verbes d'action proposés par Bloom [17] est largement utilisées par la communauté des enseignants pour formuler les objectifs pédagogiques à atteindre. A la base, il s'agit d'une taxonomie composée de verbes à l'infinitif caractérisant des grandes catégories auxquels sont ensuite associés d'autres verbes caractérisant une hiérarchie de sous-catégories. Nous nous sommes inspirés de la taxonomie de Bloom proposée par l'European Schoolnet[3].

Nous avons choisi une représentation dans le langage d'ontologie OWL. Plusieurs raisons ont guidé ce choix. D'abord une raison pragmatique de cohérence en terme d'outils techniques et conceptuels utilisés par notre équipe. Ensuite, parce que OWL nous a permis d'adjoindre clairement la description de nombreux synonymes aux verbes de Bloom (nous verrons que cela nous permet d'améliorer le rappel dans nos traitements automatisés). Enfin, cette approche, bien intégrée aux technologies du web sémantique, nous parait faciliter de futures évolutions de nos outils, par exemple pour la prise en compte de stratégies pédagogiques (par exemple, en associant un objectif pédagogique à des méthodes connues pour l'atteindre et qui seraient décrites dans cette ontologie ou dans d'autres).

Notre ontologie de Bloom est composée de 83 classes réparties dans une hiérarchie à partir de 6 classes principales (analyser, appliquer, créer, évaluer, mémoriser, comprendre). A ces classes ont été associés les labels des verbes correspondants, en français et en anglais, De plus, une référence aux mots anglais correspondants dans WordNet constitue une annotation qui peut aider à l'exploitation de ces verbes.
L'architecture de notre outil de peuplement d'ontologie est composée de deux parties principales (voir Figure 1) :

1. Le traitement des données (Annotation) : une étape de lemmatisation est appliquée pour la reconnaissance des mots de manière automatique sous différentes variations. Une autre étape de sélection de mots a été effectuée grâce à la mesure TF.IDF définie dans [18] [19]. Elle permet de proposer des 'topics' les plus porteurs de connaissances dans un domaine particulier, en mettant en évidence des mots singulièrement importants pour un programme. Enfin, nous utilisons notre ontologie 'de Bloom', qui sert à classifier les actions pédagogiques sous forme de verbes à la forme infinitive ; nous

[3] http://europeanschoolnet-vbe.lexaurus.net/lexaurus/browse

l'avons créée d'après les travaux de Bloom. Ces étapes nous permettent d'annoter sémantiquement nos documents sources. (voir Figure 2)

2. Peuplement d'ontologie : la deuxième étape, nous permet de peupler l'ontologie de base automatiquement en s'appuyant sur les annotations effectuées sur le corpus. A chaque matière, nous associons une ontologie de base, créée manuellement à l'aide de l'éditeur d'ontologie Protégé après une analyse des concepts généraux utilisés par le programme (thèmes, capacités,…). Nos différentes ontologies de base sont très voisines et nous prévoyons de les unifier.

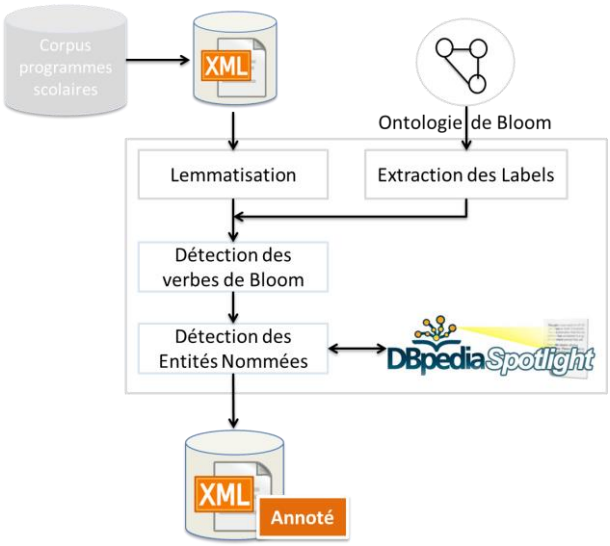

Figure 2. Etape d'annotation du corpus.

La lemmatisation est réalisée grâce à l'outil TreeTagger [20]. Cet étiqueteur grammatical permet de lemmatiser efficacement les phrases en français. Après lemmatisation, tous les mots sont représentés par leur forme générique. Cette étape nous permet de détecter les verbes de Bloom dans le corpus afin de localiser les capacités (un verbe+un objet). Les verbes de Bloom détectés dans le corpus sont marqués par des balises ajoutées au fichier XML d'entrée :

Tableau 1. La balise OntoClass.

<OntoClass uri="uri_vb_Bloom " > vb_Bloom </OntoClass>

La valeur de l'attribut URI de la balise OntoClass est la référence du concept détecté dans l'ontologie de Bloom.

Après la détection de tous les verbes de Bloom, une étape de reconnaissance de toutes les entités nommées est lancée. Cette étape utilise l'API DBPedia SpotLight dont les bons résultats sont démontrés [21]. Les entités nommées détectées sont marqués par des balises ajoutées au fichier XML d'entrée :

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Tableau 2. La balise NamedEntity.

<NamedEntity type="type_EN" uri="uri_ EN "> EN</NamedEntity>

Nous récupérons grâce à DBpedia Spotlight le type (personne, lieu, homme politique, …etc.) de l'entité nommée détectée et aussi son URI dans DBpedia.

La deuxième étape est l'étape de peuplement de l'ontologie de base. Une telle ontologie a été définie pour chaque matière du programme scolaire. La deuxième étape consistera à peupler l'ontologie de base avec des individus extraits du corpus XML annoté (voir Figure 3).

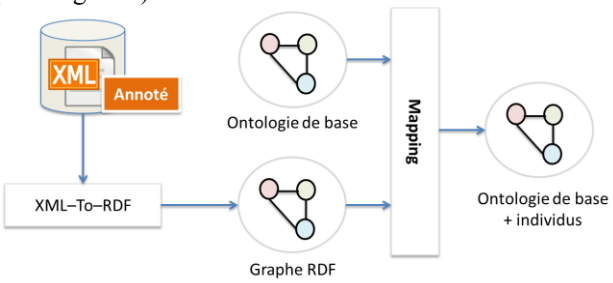

Figure 3. Etape de peuplement de l'ontologie de base.

Dans la phase de peuplement, une étape de transformation de notre corpus XML annoté en triplet RDF est accomplie par la méthode décrite dans les travaux de [22]. Cette transformation utilise un fichier XSLT. Les éléments XML sans élément fils sont représentés en sujet avec un prédicat « rdf:value ».. Un exemple de transformation d'une portion d'un document XML en triplet RDF (turtle) de la balise section et de la balise titre est présentée dans le Tableau 3 ci-dessous,

Tableau 3. Transformation XML vers RDF.

```
…
<section>
        <titre>Thème 4-LE MONDE DEPUIS LE DEBUT DES ANNÉES 1990</titre>
        <connaissances>CONNAISSANCES  Les principales lignes de force de
        la géopolitique mondiale depuis le début des années 1990.
        </connaissances>
        …
</section>
…
```

```
…
<http://www.ilot.org/#programme/matiere_3/section_7/section_6>
mp:titre
<http://www.ilot.org/#programme/matiere_3/section_7/section_6/titre> . ;

<http://www.ilot.org/#programme/matiere_3/section_7/section_6/titre>
rdf:value
"Thème 4 - LE MONDE DEPUIS LE DEBUT DES ANNÉES 1990" .
…
```

La balise *section* a pour prédicat *mp:titre* car la balise titre est un fils de type texte (balise sans fils) de cette balise. La balise titre est de type texte dans le fichier XML et dans ce cas elle aura le prédicat *rdf:value* et un objet avec comme valeur le texte de la balise titre. Le document RDF généré est pauvre sémantiquement et peut être enrichi

par un mapping vers une ontologie OWL. L'étape de mapping est nécessaire pour peupler l'ontologie de base avec les données du graphe RDF ou pour ajouter aux triplets RDF générés des liens vers l'ontologie de base. L'ontologie de base nous permet d'avoir un modèle cohérent de la structure des programmes scolaires.
Sur la base des travaux de [23] définissant un mapping entre l'ontologie DBpedia et Wikipédia, nous proposons le mapping de nos documents aux formats RDF vers l'ontologie de base. Dans les triplets RDF, nous cherchons ceux avec le prédicat « mp :nom_propriété » mp est le namespace associé à notre document XML annoté. Le « nom_propriété » désigne le nom de la propriété qu'on trouve dans le RDF. On aura, par exemple, mp:titre, mp:theme…

Tableau 4. Le mapping vers la classe Theme de la propriété titre.

```
{{ TemplateMapping
mapToClass = ops:theme;
mappings = {{ PropertyMapping
        templateProperty = mp:titre;  ontologyProperty = ops :titre;
}}
}}
```

La ligne mapToClass spécifie l'URI de la classe de l'ontologie de base à instancier. Dans la partie propertyMapping on définit l'attribut templateProperty spécifie l'URI de la propriété qui nous intéresse et qui correspond à un prédicat du document RDF et l'attribut ontologyProperty spécifie le nom de la propriété de la classe qui recevra les données. Ces deux lignes, nous permettent de faire correspondre les valeurs de nos triplets RDF avec la propriété d'une classe.

Tableau 5. La requête SPARQL générée automatiquement.

```
SELECT ?vti
WHERE {
      ?th mp:titre ?ti .
      ?ti  rdf :value ?vti .
}
```

A partir du mapping défini ci-dessus, une requête SPARQL est générée automatiquement pour extraire les données des documents RDF. La requête SPARQL du Tableau 5 est générée à partir du mapping du Tableau 4.
Deux types de propriétés sont distingués dans notre construction d'ontologie :les propriétés d'objet (ObjectProperty) permettent de relier des instances à d'autres instances ; les propriétés de type de donnée (DataTypeProperty) permettent de relier des instances à des types de données (entier, chaine de caractère…etc.).
Dans les travaux de [24], le mapping des propriétés de type ObjectProperty n'est pas présenté pour le peuplement de l'ontologie DBpedia. Le mapping entre les informations des articles Wikipédia et l'ontologie DBpedia n'y est décrit que pour une ou plusieurs propriétés de type DataTypeProperty pour une classe donnée.
Nous proposons dans notre approche, une étape supplémentaire afin de remédier aux limites du mapping précédent. Pour obtenir les propriétés reliant les instances de classe dans notre ontologie de base, après chaque instanciation de chaque classe x de

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

notre ontologie par des individus, nous récupérons toutes les propriétés de type ObjectProperty de la classe x. Ensuite, avec une requête SPARQL, nous établissons le lien avec les autres instances des classes liées avec la classe x.

Tableau 5. Algorithme de la première étape de peuplement.

```
Algorithme de peuplement
input: Ontology O1, RDF graph G, Mapping M;
output: Ontology O1+individus
Pour chaque classe C cible d'un mapping définie dans M
    Pour chaque mapping PM de propriété  défini dans M pour cette classe
        Ajouter dans l'ontologie un individu I de classe C
        R est la liste de résultats de la requête SPARQL créée à partir de PM
        Pour chaque résultat r de la liste R
            Créer une propriété de I
                de type défini par le champ ontologyProperty  de PM et de valeur r
```

La figure 4 montre un exemple de peuplement de l'ontologie de base « histoire et géographie » avec des données récupérées depuis le document du programme scolaires de 5eme. Le concept *capacités* est défini par les concepts *Bloom* et *Topic*. Le concept topic peut avoir un ou plusieurs *éléments*. Ces éléments peuvent être des entités nommées. La capacité récupérée du document « *Raconter une journée de Louis XIV* » est composée d'un verbe de Bloom et d'un Topic. Le topic « une journée de Louis XIV » est constitué d'une entité nommée « Louis XIV ». On voit ainsi que nous pourrons relier l'acquisition de cette capacité à des descripteurs liés à notre ontologie de Bloom et à des descripteurs externes liés à Louis XIV.

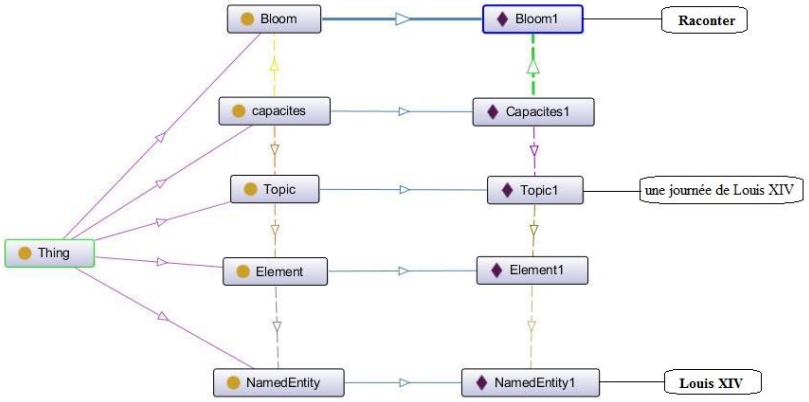

Figure 4. Exemple de peuplement d'ontologie avec une capacité et un topic.

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

4.2. Phase d'enrichissement

Pour enrichir les ontologies de base du programme scolaire, nous utilisons des collections de données ouvertes (open data). Nous pouvons distinguer deux types de telles collections : des données générales et des données spécifiques.

4.2.1. Enrichissement à l'aide des 'Catégories' de Wikipédia

La première collection concerne les différents types de données extraits de DBpedia, particulièrement Wikipédia Catégorie, destiné à classer les articles de Wikipédia. Ce type de donnée utilise les vocabulaires [24]. Depuis sa création, le graphe des 'Wikipédia catégories', est exploité comme objet de recherche à part entière. Nous pouvons citer les travaux récents comme la constitution d'une ressource sémantique issue du treillis des catégories de Wikipédia [25] et l'usage de catégorie pour la conception d'un système de recommandation inter-domaines [26].

Dans notre étude de cas, nous utilisons la classification hiérarchique de 'Wikipédia catégorie' pour enrichir les ontologies de programme scolaire de base. Par exemple, dans l'ontologie de l'Histoire des arts, une des classes primaires est la classe « Domaine artistique » contenant les informations concernant les six domaines artistiques : Arts de l'espace, Arts du langage, Arts du quotidien, Arts du son et Arts du spectacle vivant. Le programme officiel propose des listes d'exemples concrets pour chaque domaine. Les exemples des Arts de l'espace sont architecture, urbanisme, arts des jardins, paysages aménagé. Dans l'ontologie Histoire des Arts de base, ces derniers deviennent des concepts faisant partie des sous-classes de la classe «Arts de l'espace».

Un des scénarios d'enrichissement de l'ontologie grâce à 'Wikipédia catégorie' est la proposition semi-automatique de nouvelles entités associées. Concrètement, le système propose l'arborescence de « Catégorie : architecture » à l'utilisateur qui souhaite personnaliser son ontologie 'Histoire des Arts'. La figure 1 représente l'arborescence de « Catégorie : architecture », catégorie mère, et les chemins à valider par l'enseignant jusqu'à la sous-catégorie souhaitée. Le chemin contient ainsi : Architecture > Style_architectural > Architecture_gothique. A l'étape finale, l'enseignant peut valider des entités souhaitées et ajouter la « Catégorie : Architecture_gothique » en tant que nouvelles sous-classes du domaine Architecture dans son ontologie dérivée de l'ontologie de base d'Histoire des Arts.

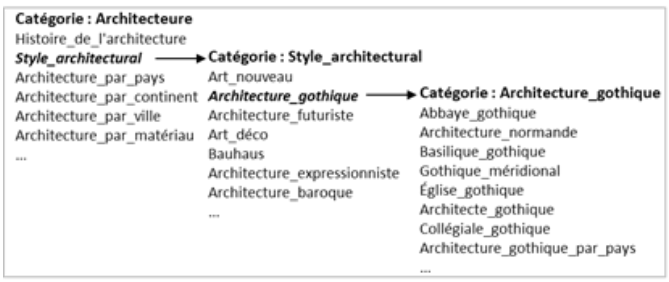

Figure 5. Arborescence de « Catégorie : Architecture » dans DBpédia.

L'usage de 'Wikipédia catégorie' illustré dans le scénario précédent pourra effectivement s'appliquer à l'enrichissement de l'ensemble des ontologies du programme scolaire de façon plus générique.

Partant d'une étude manuelle humaine des possibilités, nous avons entrepris une systématisation de la démarche avec des possibilités de propositions et de validation des enrichissements.

4.2.2. Enrichissement avec des données spécifiques

Le deuxième type de données que nous pouvons exploiter pour la phase d'enrichissement concerne les données ouvertes mises à disposition par les institutions publiques. Nous avons identifiés trois jeux de données ayant à voir avec nos ontologies de programmes scolaires sur la plateforme data.gouv.fr. Nous analysons les possibilités de connexion entre ces données et les concepts de nos ontologies. Il s'agit de « Histoire des arts-Notices textes » du Ministère de la Culture et de la Communication, d'une liste de 156 dossiers pédagogiques produits par le Centre national d'art et de culture Georges Pompidou et enfin de la collection des données sur la plateforme *data.bnf.fr*. Ces données pourront être exploitées pour l'enrichissement des ontologies du programme scolaire telles qu'Histoire des Arts, Histoire et Français.

A titre d'exemple, « Histoire des arts-Notices textes » est un document CSV peu structuré qui contient environ 4777 références utiles pour le programme d'Histoire des Arts. Nous l'avons transformé en XML, puis par un traitement XSLT, nous avons structuré le document. A partir du schéma XML du document, nous avons tiré une ontologie. En calquant la méthode sur ce qui proposé dans les sections précédentes, nous avons produit un ensemble de triplets RDF représentant les références disponibles. Puis nous avons pu établir des liens entre les thèmes indiqués dans ce document et les thèmes présents dans notre ontologie du programme scolaire d'Histoire des Arts.

A l'aide des outils Datalift [27], nous avons publié notre fichier XML, puis exporté une représentation RDF. Nous pouvons ainsi faire des interrogations SPARQL de ces connaissances sur l'Histoire des Arts ; voici un exemple de requête SPARQL :

```
SELECT DISTINCT ?titre WHERE {
  ?tags ha:tag "Normandie" .
  ?row ha:tags ?tags .
  ?peinture ha:tag "Peinture" .
  ?row ha:tags ?peinture .
  ?row ha:titre ?titre
} LIMIT 100
```

qui nous donne le titre de toutes les références qui ont pour tags les mots "Peinture" et "Normandie".

Le fait de mettre en correspondance les ontologies du programme scolaire avec d'autres données structurées, traitées et préparés par les professionnels du domaine a un double avantage. Il facilite, en premier lieu, la tâche de l'enseignant pour trouver des ressources éducatives de bonne qualité et fiables correspondants aux programmes

officiels. Il est, en outre, bénéfique pour les producteurs des données culturelles, la réutilisation de leurs ressources étant considérée comme un moyen de la valorisation du patrimoine par l'éducation. Nous sommes également en train d'identifier et d'analyser d'autres données culturelles produites par les collectivités locales. L'objectif du repérage de telles données est d'exploiter ces données dans le contexte de l'adaptation du programme scolaire pour la valorisation des patrimoines régionaux. Cette action fait partie des démarches pédagogiques fortement recommandées par le programme de l'Histoire des Arts.

4.3. Perspective: phase d'exploitation

C'est la dernière phase de notre méthode qui servira d'interface entre l'enseignant et les programmes scolaires. Notre formalisation est actuellement constituée de plusieurs ontologies de chaque matière. Nous comptons rapidement créer une ontologie intégrant l'ensemble, ce qui facilitera l'établissement de liens entre programmes.
Nous avons déjà créé des interfaces qui exploitent nos ontologies pour présenter les programmes en mettant en évidence les capacités à acquérir et les liens avec des ressources tirées de DBPedia. La perspective est de rendre d'autres ressources externes exploitables pour un enrichissement tant du travail du professeur en phase de préparation de cours que du parcours des cours par les élèves.

5 Conclusion

Dans cet article, nous avons proposé une méthode de création d'ontologie à partir d'un corpus de programmes scolaire français. Pour cela, nous avons définis une ontologie de base pour chaque matière afin d'éviter des problèmes d'incohérence sur la génération automatique de l'ontologie.

Dans cette approche, nous avons utilisé l'ontologie de Bloom pour la détection des capacités dans notre corpus et l'API DBpedia SpotLight [21] pour la détection des entités nommées. Les documents XML ont été ensuite transformés en RDF qui seront mappé vers nos ontologies de base. Ces ontologies vont être enrichie et exploité semi-automatique afin d'enrichir les possibilités de l'enseignant en s'appuyant sur le contenu des programmes scolaires.

Dans nos travaux futurs, nous enrichissons l'ontologie de Bloom avec des synonymes de verbes (167 synonymes) pour chaque classe de verbe afin de permettre la détection d'autres capacités dans notre corpus ce qui augmentera la précision dans notre approche.

Références

1. Gruber, T., Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43, pp. 907-928. 1993.

2. Hérin, D., Sala, M., & Pompidor, P., Evaluating and Revising Courses from Learning Web Resources. ITS'2002. 2002.
3. Pompidor, P., Sala, M., & Hérin, D., An incremental method for extraction of pedagogical knowledges on the web. SW-WL'2003 & EIAH'2003. 2003.
4. Sala, M., Pompidor, P., & Hérin, D., Aid to the Semantic Maintenance of the Web Site. IADIS WWW/Internet'03. 2003.
5. Nkambou, R., Frasson, C., & Gauthier, G., CREAM-Tools: An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems. In Authoring Tools for Advanced Technology Learning Environments: Toward coste effective, adaptative, interactive, and intelligent educational software, pp. 93-138. 2003.
6. Hernandez, N., & Mothe, J., TtoO: une méthodologie de construction d'ontologie de domaine à partir d'un thésaurus et d'un corpus de référence. Rapport de recherche, IRIT/RR 2006-04--FR, IRIT. 2006.
7. Cimiano, P., & Völker, J., Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. (L. N. Springer, Éd.) Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), pp. 227-238. 2005.
8. Maedche, A., & Staab, S., Ontology Learning for the Semantic Web. IEEE Intelligent Systems, Special. 2001.
9. Roitman, H., & Gal, A., OntoBuilder: fully automatic extraction and consolidation of ontologies from web sources using sequence semantics. EDBT'06 Proceedings of the 2006 international conference on Current Trends in Database Technology, pp. 573-576. 2006.
10. Ferdinand, M., Zirpins, C., & Trastour, D., Lifting XML Schema to OWL. Web Engineering Lecture Notes in Computer Science Volume 3140, pp. 354-358. 2004.
11. Bohring, H., & Auer, S., Mapping XML to OWL Ontologies. In Leipziger Informatik-Tage, vol. 72, pp. 147–156. 2005.
12. Ghawi, R., & Cullot, N., Building Ontologies from XML Data Sources. DEXA '09. 20th International Workshop on Database and Expert Systems Application, pp. 480-484. 2009.
13. Gras-Velazquez, A., Teachers and content packaging standards. Initial conclusions from the ASPECT evaluation. Récupéré sur Adopting Standards and Specifications for Educational Content: http://aspect-project.org/node/84. 2010.
14. Gómez de Regil, R., Retour d'expérience sur le pilote ASPECT. Récupéré sur http://www.lom-fr.fr/scolomfr/communication/conferences.html. 2011.
15. Desmoulins, C. Construction avec des enseignants d'une ontologie des compétences en géométrie, Geoskills. Ingénierie des connaissances (IC 2010). http://www-limbio.smbh.univ-paris13.fr/GBPOnto/data/documents/2010/5desmoulins.pdf
16. Bedini, I., & Nguyen, B., Automatic Ontology Generation: State of the Art. University of Versailles Technical report. 2007.
17. Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. Taxonomy of educational objectives: Handbook I: Cognitive domain. New York: David McKay, 1956.
18. LUHN, H. P. (1958). The automatic creation of literature abstracts. IBM Journal on Research and Development, 2(2).
19. SPÄRCK JONES, K. (1972). A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation, 28
20. Schmid, H., Probabilistic part-of-speech tagging using decision trees. In International Conference on New Methods in Language Processing. 1994.
21. Mendes P.N., Jakob M., García-Silva A., Bizer C. D., DBpedia Spotlight: Shedding Light on the Web of Documents. In the Proceedings of the 7th International Conference on Semantic Systems (I-Semantics 2011). Graz, Austria. 2011.
22. Breitling, F., A standard tranformation from XML to RDF via XSLT. Astronomical Notes, pp. 755-760. 2009.

23. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P., & Bizer, C., Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web Journal. 2013.
24. Torres, D., Molli, P., Skaf-Molli, H., & Diaz, A., Improving wikipedia with DBpedia. In Proceedings of the 21st international conference companion on World Wide Web, pp. 1107-1112. 2012.
25. Collin, O., Gaillard, B., & Bouraoui, J.-L., Constitution d'une ressource sémantique issue du treillis des catégories de Wikipedia. TALN 2010-Session Posters. 2010.
26. Fernández-Tobías, I., Kaminskas, M., Cantador, I., & Ricci, F., A generic semantic-based framework for cross-domain recommendation. HetRec '11 Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems, pp. 25-32. 2011.
27. Scharffe, F., Atemezing, G., Troncy, R., Gandon, F., Villata, S., Bucher, B., & Vatant, B., Enabling linked-data publication with the datalift platform. In Proc. AAAI workshop on semantic cities. 2012.

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

IR^2 : Using External Indexes to Expand Document Representations for Ad-hoc Retrieval

Davide Buscaldi[1]

LIPN - Laboratoire d'Informatique de Paris Nord, CNRS, (UMR 7030)
Université Paris 13, 93430 Villetaneuse, France
davide.buscaldi@lipn.univ-paris13.fr

Extended Abstract

In the last years, the attention of many researchers in the field of Natural Language Processing has been focused on *semantic* similarity methods. Such methods differ from "classical" similarity methods in the sense that similarity is calculated not only on the basis of surface features, like characters, frequencies in some text collection, but also using deep linguistic analysis methods, such as parsing, disambiguation, Named Entity recognition. The Semantic Textual Similarity (STS[1]) task has been proposed at the SemEval campaigns since 2012, in order to foster research on this topic. Within SemEval, several semantic similarity measures have been proposed, using external knowledge [1], corpora [2], syntactic dependencies [3]. The aim of the work presented in this paper is to study whether this kind of measures can be used effectively in Information Retrieval (IR) tasks such as ad-hoc retrieval. We focused on the IR-based measure introduced for our participation to SemEval2013 [4], which resulted to be the best of the 9 features used in our system, with a Pearson correlation of 0.541 on the test collection.

The IR-based measure considers two texts p and q as input queries to an IR system S, with a document collection D indexed by S. We assume that p and q are similar if the documents retrieved by S for the two texts are ranked similarly. Let be $L_p = \{d_{p_1}, \ldots, d_{p_K}\}$ and $L_q = \{d_{q_1}, \ldots, d_{q_K}\}$, $d_{x_i} \in D$ the sets of the top K documents retrieved by S for texts p and q, respectively. Let us define $s_p(d)$ and $s_q(d)$ the scores assigned by S to a document d for the query p and q, respectively. Then, the similarity score is calculated as:

$$sim_{IR}(p,q) = 1 - \frac{\sum_{d \in L_p \cap L_q} \frac{\sqrt{(s_p(d) - s_q(d))^2}}{\max(s_p(d), s_q(d))}}{|L_p \cap L_q|} \quad (1)$$

if $|L_p \cap L_q| \neq \emptyset$, 0 otherwise. We used the Lucene[2] 4.4 as search engine with BM25 similarity. The K value for the IR-based similarity measure was set to 70 after some tests on the STS 2012 data.

[1] http://ixa2.si.ehu.es/sts/
[2] http://lucene.apache.org/core

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

We studied the possibility to use the IR-based measure to calculate similarities between document and queries in an IR system. We figured out two major issues that need to be addressed: the first one, the fact that the measure was conceived to compare a sentence to another one and not a sentence (the query) to a set of sentences (the text). The second, that the reference collection D indexed to calculate the similarity measure may have a different coverage with respect to the document collection that we are indexing (let it be C). Instead, we suggest to use the document lists retrieved using the IR-based measure to enhance the existing document representations. The proposed indexing process is as follows:

foreach *document* d_c *in* C **do**
  Transform d_c in a set of sentences $S_c = \{s_1, \ldots, s_n\}$;
  foreach s_i *in* S **do**
    Obtain the list of relevant documents $L_{s_i} = \{r_1, \ldots, r_K\}, r_i \in D$ and their scores ;
  end
  Merge all lists L_{s_i} in a list L_{d_c} (keep the highest score if a document occurs more than once);
  L_{d_c} to enhance the representation of d_c;
  foreach r_k *in* L_{d_c} **do**
    Add the *id* of document r_k and its weight to the representation of d_c ;
  end
end

Therefore, the ids of the documents in D are terms that are added to the representation of a document $d_c \in C$, weighted according to the the BM25 scores obtained for the sentences composing d_c. Similarly, a set of weighted documents $L_q \subset D$ can be obtained for a query q and compared using the IR-based similarity to rank documents in C.

We carried out a first evaluation of this setting by indexing the AQUAINT-2[3] and the English NTCIR-8[4] collections as reference D. The test collection C was composed by the Robust WSD CLIR collection[5], without WordNet senses. We calculated the normalized discounted cumulative gain (nDCG) over the 159 queries with three different set-ups: using only the terms in C and Lucene with BM25 scores (baseline), using only the terms from D and the IR-based measure (IRsim), and an hybrid approach in which the score of each document is the average between its BM25 score and the IRsim score calculated on its expanded terms (hybrid), obtaining respectively 0.562, 0.381 and 0.564 for average nDCG with the three configurations. Image 1 shows the results obtained with bm25 and IRsim for the first 32 queries of the test set (we chose only the first 32 for reason of space).

[3] http://www.nist.gov/tac/data/data_desc.html#AQUAINT-2
[4] http://metadata.berkeley.edu/NTCIR-GeoTime/ntcir-8-databases.php
[5] http://ixa2.si.ehu.es/clirwsd/

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE,
SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

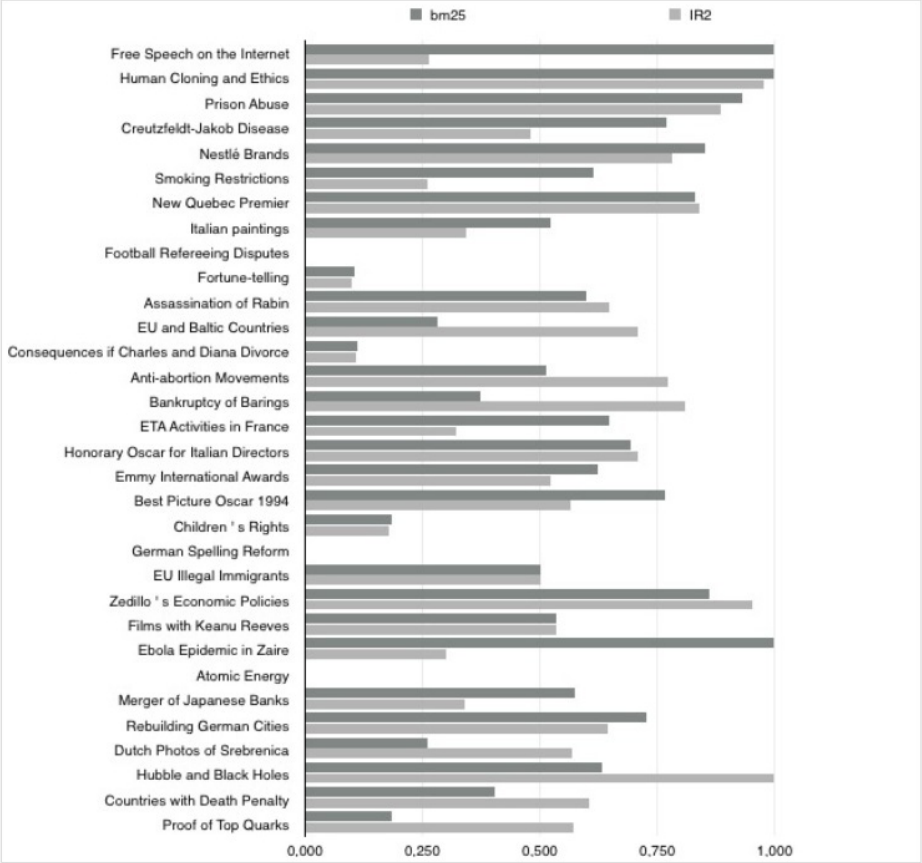

Fig. 1. Results obtained for the first 32 queries in the test set.

The results in Figure 1 show that in some cases the hybrid scoring allowed to obtain a significant improvement over the base BM25 score, but we are still studying the reason of such improvements. We are studying some queries where the improvement was remarkable, like query 193-AH: 'EU and Baltic countries'. In Table 1 we compare the top 5 results retrieved by the base system with BM25 and the system with the hybrid weighting.

In the case of query 193-AH it is possible to observe that the use of document annotations allowed to establish a link between "Latvia" and "baltic countries". We suppose that the links that can be found depend strongly from the reference collection used.

Although we were able to obtain a slight improvement with the hybridation, the difference is not statistically significant. Given the naive assumptions that we made, especially with regard to the composition of sentence-based scores into a

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

bm25		
rank	*docID*	*title*
1	**GH950613-000167**	Baltic states join queue for European membership
2	LA121194-0308	EU PLANS TO ADMIT EX-SOVIET BLOC NATIONS
3	GH950217-000132	Kinnock backs code of safety for ferries
4	GH950724-000097	New European realism
5	GH950414-000146	Peace in sight over Spanish armada
hybrid		
1	**GH950613-000167**	Baltic states join queue for European membership
2	**GH950612-000097**	Baltic deal
3	**GH951028-000091**	Latvia woos EU
4	**GH951014-000120**	Latvia woos EU
5	GH950107-000124	Lawyer is Baltic choice

Table 1. Results obtained for query 193-AH: *EU and baltic countries*. IDs in bold indicate that the related document is relevant to the query.

document score, we suppose that this result can be improved, for instance using text compression and automatic summarization algorithms. We would also like to investigate the use of a reference collection with a broader coverage, such as Wikipedia, taking inspiration from Explicit Semantic Analysis by [2].

References

1. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st national conference on Artificial intelligence - Volume 1. AAAI'06, AAAI Press (2006) 775–780
2. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proceedings of the 20th international joint conference on Artifical intelligence. IJCAI'07, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (2007) 1606–1611
3. Bär, D., Biemann, C., Gurevych, I., Zesch, T.: Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In: Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), Montreal, Canada (June 2012) 435–440
4. Buscaldi, D., Le Roux, J., Garcia Flores, J.J., Popescu, A.: Lipn-core: Semantic text similarity using n-grams, wordnet, syntactic analysis, esa and information retrieval based features. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Atlanta, Georgia, USA, Association for Computational Linguistics (June 2013) 162–168

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Integrating Terms Hierarchy into Dirichlet Language Model

Mohannad ALMasri*, KianLam Tan*, Jean-Pierre Chevallet**, Philippe Mulhem***, and Catherine Berrut****

* Université de Grenoble, ** UPMF-Grenoble 2, *** Centre National de la Recherche Scientifique, **** UJF-Grenoble 1
LIG laboratory, MRIM group, Grenoble, France {mohannad.almasri, kian-lam.tan,jean-pierre.chevallet,philippe.mulhem,catherine.berrut}@imag.fr

Abstract. Most Information retrieval systems (IRSs) use the intersection between document and query in order to retrieve relevant documents to a given query. Term mismatch problem appears when users use different terms form terms used in the index to express their needs. Indexing term specificity is one face of term mismatch problem where a user query contains more general indexing terms from terms in the index. In this paper, we present an approach to capture the specificity of terms by incorporating the hierarchical information between indexing terms into a Language Model. Experiments on different CLEF corpora from the medical domain show an improvement in retrieval performance. We show that this improvement is independent to the length of documents and queries within the tested collection.

1 Introduction

Specificity is a semantic property that can be applied to index terms: an indexing term[1] is more or less specific as its meaning is more or less detailed and precise. For instance, in the medical domain, the terms "*B-Cell*" and "*T-Cell*" are more specific than "*Lymphocyte*", or in other words, we say that "*B-Cell*" and "*T-Cell*" are types of "*Lymphocyte*" in the adaptive immune system. Therefore, when a user query contains the term "*Lymphocyte*", then, a document talks about "*B-Cell*" or "*T-Cell*" is relevant to this query. Another example from the same domain, documents talk about "*Veins*" or "*Arteries*" are relevant to the query "*Blood Vessel*", where "*Veins*" and "*Arteries*" are types of "*Blood Vessel*". A retrieval model that depends on the intersection between document and query cannot capture this kind of relation between indexing terms. In order to take these hierarchical relations between query and document terms, we should incorporate them in the retrieval model. We need an external knowledge or resource [2] further from query and document to identify these hierarchical relations between terms.

[1] Indexing terms differ from system to another, so it can be : word, noun phrase, n-gram, or concept [5].
[2] like thesaurus or ontology

Classical indexing techniques represent documents and queries as a bag of words or phrases without taking into account the semantics, the meaning or the correlation between these words. The main disadvantage of these techniques is that they depend on the text signal, and not on the meaning [5, 12]. For example, in the medical domain, the two phrases "Atrial Fibrillation" and "Auricular Fibrillation" have the same meaning. However, when we use phrases to represent a document and a query, if one phrase appears in a document and a different one appears in a query, that leads to mismatch problem. So over the last 20 years, several approaches attempted to use available knowledge bases and natural language processing techniques in order to overcome this problem and produce more meaningful answers [1]. These approaches represent documents and queries by means of concepts. This representation is obtained using conceptual indexing. Conceptual indexing is the process of mapping text into the concepts of an *external resource*. Therefore, it needs a resource out of documents and queries which contains concepts and the information about them. In our study, we use concepts as indexing terms.

In this paper, we consider the problem of concepts specificity within in the Language Model (LM). Language Model for IR has been proven to be a very effective method for text retrieval [15, 20]. The extension that we propose in this paper is to integrate concept hierarchy into the Dirichlet language model. This extension is easily applied to other smoothing methods. Our proposed method has the following advantages: a) it is easy and simple to generate concept similarity based on the hierarchy of concepts from an external resource b) we propose a light weight integration in the Dirichlet smoothing that improves the retrieval performance. The rest of this paper is organized as follows. Firstly, we present the problem of term mismatch in Section 2. Then, we discuss several approaches to solve the problem of term intersection in Section 3 followed by our approach in Sections 4. Finally, we conclude our results and present the future work in Sections 5 and 6.

2 Term Mismatch Problem

Several techniques [13] have been proposed to tackle term mismatch problem. Among these techniques: relevance feedback [11, 17], dimension reduction [16, 10, 7, 2, 8], and statistical translation model [3, 9].

Relevance feedback involves the user in the IR process in order to improve the final result. There are three types of relevance feedback: 1) explicit feedback, 2) implicit feedback and 3) pseudo or blind feedback [13]. Rocchio algorithm [17] is the classic algorithm for implementing explicit feedback which enables the user to select relevant documents in order to reformulate the original query. Query Reformulation is made by adding terms extracted from the selected documents. Implicit feedback incorporates user behavior like clicks, in order to predict relevant documents to reformulate the query, while blind feedback provides a method for automatic local analysis. Blind feedback automates the manual part of the Rocchio algorithm without an extended interaction with the user. This method

performs normal retrieval and finds an initial set of relevant documents and makes the assumption that the top k ranked documents are the most relevant. Lavrenko and Croft [11] proposed an approach to estimate a relevance model with no training data. The main problem for implicit and explicit relevance feedback is that they should rely on accurate ways of finding term relation in order to avoid the problem of query drift.

Dimension reduction is the process of reducing the number of data dimensions that represents a query and a document in cases where the query and the document refer to the same concept but using different terms. This can be achieved by using thesaurus [8], concept based approach [2], stemming [16, 10], and latent semantic indexing [7]. All these techniques proposed different strategies to reduce the chances that the query and document refer to the same concept but using different terms. In later development, Peng et al.[14] performed stemming according to the context of the query which helps to improve the accuracy and the performance of retrieval compared to the query independent stemmers such as Porter[18] and Krovetz[10]. Deerwester et al.[7] proposed to solve the dimension reduction by representing the terms and the documents in a latent semantic space where the terms that are similar in the space tend to be the terms that not only co-occur in the documents, but also appear in similar contexts.

Statistical Translation Model is a model where all of the translations are generated on the basis of a statistical models. This idea is based on information theory where a model estimates the probability of translate a document to a user query according to the probability distribution $P(u|v)$, which gives the probability that a word, v can be semantically translated to a word, u [19, 3]. Unfortunately, Statistical Translation Model requires training data and some relevant query-document pairs where the documents are relevant to the query.

3 Exploiting term similarity

Most of the approaches to solve the problem of term mismatch described in Section 2 faced the same problem: how to select the best term and assign the best weight to the corresponding term?. No single solution has been proved to be the best.

Some approaches have been proposed using LM such as the work of Karimzadehgan and Zhai [9], Berger and Lafferty[3] who use Statistical Translation Model. The main difference between these two works is that Berger and Lafferty try to identify the most important concepts apears in a verbose query [3] while Karimzadehgan and Zhai used mutual information to generate term links[3] [9].

In some ways, we can consider that the proposed approach [9, 3] are related to the proposition from Crestani [6] where the idea is to consider the similarity between each query term and all document terms. The results obtained by Karimzadehgan and Zhai [9] showed that integrating the term similarity and LM is more efficient and more effective than the existing approaches in information retrieval.

[3] Term links refer to the relationship between two terms in a vocabulary

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

However, Karimzadehgan and Zhai [9] noticed that the self-translation probabilities lead to non-optimal retrieval performance because it is possible that the value of $P(w|u)$ is higher than $P(w|w)$ for a term, w. In order to overcome this problem, Karimzadehgan and Zhai [9] defined a parameter to control the effect of the self-translation.

In a nutshell, we can remark that 1) the normalization of the mutual information is rather artificial and requires a parameter to control the effect of the self-translation, and 2) the regularization of the initial transition probabilities may look uncertain.

4 Proposed Approach

Our model uses concepts as indexing terms. In other words, queries and documents are represented by concepts. Then, we use hierarchical relations from an external resource to build the Concept Similarity Matrix. This matrix contains semantic similarities between each two concepts computed from an external resource. Our goal is to integrate the Concept Similarity Matrix into LM in order to overcome the mismatch problem. After the reviews of Crestani [6], Karimzadehgan and Zhai [9] and Zhai [19], we propose the approach as shown below:

- In the case that there is a query concept does not appear in the document(Mismatch): we consider the most similar document concept to this query concept in the matching process. We use concept similarity matrix to find the most similar concept.
- We propose to exploit hierarchical relations between concepts which is defined in the external knowledge to define the semantic link between concepts rather than probability approaches in order to avoid the problem of self-translation Karimzadehgan and Zhai [9].

In order to build the Concept Similarity Matrix, we find the links between all the vocabulary V which is the set of all concepts. We make the assumption that the two concepts are considered to be linked to each other if both concepts belong to the same hierarchy in the external resource. Assume a query concept c, and c' refers to a document concept:

$$c, c' \in V, 0 \leq Sim(c, c') \leq 1 \tag{1}$$

1. $Sim(c, c') = 0$, there is no link between the concept c and c'
2. $Sim(c, c') < 1$, there is a link between the concept c and c'
3. $Sim(c, c') = 1$, there is an exact match between the concept c and c'

4.1 Extended Dirichlet Smoothing

The LM approach in IR is proposed by Ponte and Croft [15]. The basic idea of LM is to assume that a query q, which is generated by a probabilistic model based on a document d, as shown below:

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

$$P(d|q) \propto P(q|d).P(d) \tag{2}$$

$\propto$ means that the two sides give the same ranking. $P(q|d)$ the query likelihood for the given document d matches with the query q. If we consider that every document is equally relevant to any other query, then we can discard $P(d)$ and we can rewrite the formula after adding the log function as:

$$logP(d|q) = \sum_{c \in V} \#(c;q).logP(c|d) \tag{3}$$

where $\#(c;q)$ is the count of concept c in the query q and V is a set of vocabulary. Assuming a multinomial distribution, the simplest way to estimate $P(c|d)$ is the maximum likelihood estimator:

$$P_{ml}(c|d) = \frac{\#(c;d)}{|d|} \tag{4}$$

where $|d|$ is the total length of the document d. Due to the data spareness problem, the maximum likelihood estimator directly assign *null* to the unseen concept in a document. Smoothing is a technique to assign extra probability mass to the unseen concept in order to solve this problem.

Basically, Dirichlet [21] is one of the smoothing technique based on the principle of adding an extra pseudo concept frequency: $\mu P(c|C)$. The Dirichlet smoothing is obtained by taking into account the extra pseudo concept frequency distribution:

$$P_{\mu}(c|d) = \frac{\#(c;d) + \mu P(c|C)}{\sum_c \#(c;d) + \mu} \tag{5}$$

where C is the whole collection. The main idea of this proposal is to integrate links between concepts which are represented by Concept Similarity Matrix into the current Dirichlet formula. First, we assume that for a query concept $c \in q, c \notin d$, there is a document concept $c' \in d$ can play its role during the matching process. More specifically, we consider that if c does not occur in the initial document d but occurs in the *document* d_{ext}, which is the result of extending d according to the query and some knowledge [4], the probability of the concept c' is defined according to the extended document d_{ext}.

The knowledge provides a similarity function $Sim : V \times V \rightarrow [0, 1]$, that denotes the strength of the similarity between the two concepts (the larger the value, the higher the similarity between these two concepts). We propose that: $\forall c, c' \in V, Sim(c, c') = 1$ if exact matching between c with c', and $\forall c, c' \in V, Sim(c, c') = 0$ if c is not at all semantically related to c'.

In order to avoid any complex extension, we assume that a query concept c, must only impact occurrences of one document concept, so:

1. If a query concept c occurs in a document d, then the concept will not change the length of the document.

[4] The knowledge refers to the Concept Similarity Matrix

2. If a query concept c does not occur in a document d but the concept c contains a link with c' (concept from document), then we define: :

$$c^* = argmax_{c' \in d} Sim(c, c')$$

as the concept from the document will serve as the basic count of the pseudo occurrences of c in d:

$$\#(c^*; d).Sim(c, c^*)$$

this pseudo occurrences of the concept c are then included into the size of the extended document.

3. If a query concept c does not occur in the document and does not show any link to the document concepts , then this concept will not change the length of the document as the first case.

According to the previous three cases, the expression of $|d_{ext}|$ can be done:

$$|d_{ext}| = |d| + \sum_{c \in q} \#(c^*; d).Sim(c, c^*) \tag{6}$$

Note that we propose to extend the document according to the query. We extend the document by query concepts which are not in the document but they are linked to at least on document concept. Now, the extended Dirichlet Smoothing leads to the following probability for the concept c of the vocabulary V in the extended document d_{ext} according to a query q, and note that $p_\mu(c|d_{ext})$ is defined as:

1. if $c \in d \cap q$:

$$P_\mu(c|d_{ext}) = \frac{\#(c; d) + \mu P(c|C)}{|d_{ext}| + \mu} \tag{7}$$

2. $c \notin d \cap q$ and if $\exists c^* \in d \setminus q; Sim(c, c^*) \neq 0$:

$$P_\mu(c|d_{ext}) = \frac{\#(c^*; d).Sim(c, c^*) + \mu P(c^*|C)}{|d_{ext}| + \mu} \tag{8}$$

with $c^* = argmax_{c' \in d} Sim(c, c')$.

3. $c \notin d \cap q$ and if $\nexists c^* \in d \setminus q; Sim(c, c^*) \neq 0$

$$P_\mu(c|d_{ext}) = \frac{\#(c; d) + \mu P(c|C)}{|d_{ext}| + \mu} \tag{9}$$

with $c^* = argmax_{c' \in d} Sim(c, c')$.

In the specific case where all the query concepts occur in the document, we have $|d_{ext}| = |d|$, and that leads to $p_\mu(c|d) = p_\mu(c|d_{ext})$.

4.2 Concept Similarity Matrix

We propose to use a lightweight way to build the Concept Similarity Matrix using the concept hierarchy from an external resource. The similarity between two concepts is the inverse of a distance between these two concepts in the concept hierarchy [18]. We use the path length or the number of links in the hierarchy between two concepts as distance.

The similarity score is inversely proportional to the number of nodes along the shortest path between the concepts. The shortest possible path occurs when the two concepts are directly linked. Thus, the maximum similarity value is 1:

$$Sim(c, c') = \frac{1}{distance(c, c')}, distance(c, c') > 0 \quad (10)$$

We also tried other similarity metrics like Leacock and Resnik but we obtained best performance improvement using this path metric.

5 Experimental Setup

5.1 Indexing Terms

Documents and queries is mapped to UMLS concepts using MetaMap. UMLS is a multi-source knowledge base in the medical domain, whereas, MetaMap is a tool for mapping text to UMLS concepts. Using concepts allows us to investigate the semantic relations between concepts, so it allows to build our Concepts Similarity Matrix. To build this matrix, we only consider, the ISA relations between concepts from the different UMLS concept hierarchies. If we have two concepts in multiple concept hierarchies we consider the shortest path.

5.2 Corpora

Five corpora from CLEF are used. Table 1 shows some statistics about them.

- Image10, Image11, Image12: contain short medical documents and queries.
- Case2011, Case2012: contain long medical documents and queries.

5.3 Results

All the experiments are conducted using the XIOTA engine [4]. The performance was measured by Mean Average Precision (MAP). The approaches used for experiments are as follows:

- DIR-BL (baseline): language model with Dirichlet smoothing.
- DIR-CSM: Extended Dirichlet smoothing after integrating the Concepts Similarity Matrix(CSM).

Table 1. Corpora statistics. *avdl* and *avql* are average length of documents and queries. Number of general concepts inside the queries, or in other words the number of concepts which has the potential to be subsumed at matching time.

Corpus	**#d**	**#q**	**avdl** (words)	**avql** (words)	**Number of Concepts in the Queries**	**Number of General Concepts**
Image2010	77495	16	62.12	3.81	186	109
Image2011	230088	30	44.83	4.0	374	198
Case2011	55634	10	2594.5	19.7	516	219
Image2012	306530	22	47.16	3.55	204	132
Case2012	74654	26	2570.72	24.35	1472	519

Results of our Dirichlet smoothing extension are summarized in Table 2. We first observe the consistent performance improvement achieved for our five target collections, which confirms our belief that integrating hierarchical relations from an external resource improves relevance model estimation. Second, the improvement occurs in the studied collection is independent the length of these collection. It seems to be similar for both types of collection: 1) short documents and short queries, 2) long documents and long queries.

Table 2. MAP of Extended Dirichlet smoothing after integrating Concept Similarity Matrix: our approach outperforms the baseline result for all studied collections. The gain obtained could be related to the number of general concepts inside the queries, or in other words the number of concepts which has the potential to be subsumed at matching time.

Corpus	General Concepts Rate	**DIR-BL**	**DIR-CSM**	**Gain**
Image2010	59%	0.2571	0.3049	+19%
Image2011	53%	0.1439	0.1540	+7%
Case2011	42%	0.1493	0.1597	+7%
Image2012	65%	0.1039	0.1131	+9%
Case2012	35%	0.1788	0.1861	+4%

Table 3 show some statistics about our three cases in the extended model during the matching between documents and queries. We see in this table: 1) the number of shared between documents and queries. 2) the number document concepts linked to unmatched query concept.3) the number of document concepts which they can not be linked to a query concepts. These number are over all queries and documents in the studied collections.

6 Conclusions

We present a model to exploit the indexing term hierarchy in order to capture the specificity of indexing terms during the retrieval time. Our experimental re-

Author-produced version of the paper presented at 6ème atelier Recherche d'Information SEmantique RISE, SDNRI 2014, March 18th, 2014, Nancy, France.
The original publication is available at https://sites.google.com/site/frenchsemanticir/documents

Table 3. Statistics during the matching between documents and queries: 1) number of shared between documents and queries. 2) number document concepts linked to unmatched query concept.3) number of document concepts which they can not be linked to a query concepts.

Corpus	#Shared	#Linked	#Not linked
Image2010	2,138,561	668,148	11,607,361
Image2011	2,492,692	1,275,058	82,285,162
Case2011	6,725,714	932,321	21,049,109
Image2012	2,874,031	1,556,122	91,169,348
Case2012	27,940,254	3,681,351	78,255,835

sults indicate that the proposed approach to extend Dirichlet smoothing using Concept Similarity Matrix based on hierarchical information from an external resource in the medical domain is more effective than the term intersection approach. This extension is suitable for any situation where such a kind of this mutual information between indexing terms is available. For future work, we would like to validate our extension using mutual information between indexing terms extracted from other external resource and maybe in different ways rather than hierarchical relations. In addition, we think that with more mutual information we will have a higher degree of knowledge to build the link between two indexing terms.

References

1. Mustapha Baziz, Mohand Boughanem, and Nathalie Aussenac-Gilles. Conceptual indexing based on document content representation. CoLIS'05, 2005.
2. Michael Bendersky and W. Bruce Croft. Discovering key concepts in verbose queries. SIGIR '08, pages 491–498, New York, NY, USA, 2008. ACM.
3. Adam Berger and John Lafferty. Information retrieval as statistical translation. SIGIR '99, pages 222–229, New York, NY, USA, 1999. ACM.
4. Jean-Pierre Chevallet. X-iota: An open xml framework for ir experimentation. volume 3411 of *Lecture Notes in Computer Science*, pages 263–280. Springer Berlin Heidelberg, 2005.
5. Jean-Pierre Chevallet, Joo-Hwee Lim, and Diem Thi Hoang Le. Domain knowledge conceptual inter-media indexing: Application to multilingual multimedia medical reports. CIKM '07, pages 495–504. ACM, 2007.
6. Fabio Crestani. Exploiting the similarity of non-matching terms at retrieval time. 2:25–45, 2000.
7. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.
8. Yufeng Jing and W. Bruce Croft. An association thesaurus for information retrieval. pages 146–160, 1994.
9. Maryam Karimzadehgan and ChengXiang Zhai. Estimation of statistical translation models based on mutual information for ad hoc information retrieval. pages 323–330. ACM, 2010.
10. Robert Krovetz. Viewing morphology as an inference process. pages 191–202. ACM Press, 1993.

11. Victor Lavrenko and W. Bruce Croft. Relevance based language models. SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.
12. Jimmy Lin and Dina Demner-Fushman. The role of knowledge in conceptual retrieval: a study in the domain of clinical medicine. SIGIR '06, pages 99–106, 2006.
13. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
14. Fuchun Peng, Nawaaz Ahmed, Xin Li, and Yumao Lu. Context sensitive stemming for web search. SIGIR '07, pages 639–646, New York, NY, USA, 2007. ACM.
15. Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. SIGIR '98, pages 275–281. ACM, 1998.
16. M. F. Porter. Readings in information retrieval. chapter An algorithm for suffix stripping, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.
17. Gerard Salton, editor. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood, Cliffs, New Jersey, 1971.
18. Dominic Widdows. *Geometry and Meaning*. Center for the Study of Language and Inf, November 2004.
19. ChengXiang Zhai. *Statistical Language Models for Information Retrieval*. Now Publishers Inc., Hanover, MA, USA, 2008.
20. Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. 22(2):179–214, April 2004.
21. Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. 22(2):179–214, April 2004.