Exploration des Données du Défi EGC 2016 à l'aide d'un Système d'Information Logique

Peggy Cellier, Sébastien Ferré, Annie Foret, Olivier Ridoux

Exploration des Données du Défi EGC 2016 à l'aide d'un Système d'Information Logique

Peggy Cellier*, Sébastien Ferré**, Annie Foret**, Olivier Ridoux**

* INSA Rennes, IRISA — ** Université Rennes 1, IRISA — {prenom.nom}@irisa.fr

Résumé. Nous présentons dans cet article les méthodes employées et les résultats obtenus en réponse au Défi EGC 2016. Notre approche repose d'une part sur des chaînes automatiques de traitements linguistiques en français et en anglais utilisant le plus possible des ressources et outils publics et d'autre part sur un environnement d'exploration des données basé sur les systèmes d'information logiques ; ces systèmes exploitent une généralisation des treillis de concepts formels appliquée aux données attribut-valeur ou au web sémantique.

1 Introduction

Le Défi EGC 2016 propose d'exploiter un fichier texte contenant des descriptifs de publications EGC. Nous avons choisi d'exploiter ces données en utilisant les méthodes de l'analyse de concepts logiques (Ferré et Ridoux, 2004) via les outils Camelis et Sparklis. Ces méthodes étant de nature purement symbolique il faut commencer par extraire des données les traits symboliques que nous voulons exploiter de façon à s'affranchir des variations linguistiques dans la donnée : flexions, synonymie, voire même langue. Dans une première partie nous décrivons la chaîne de traitement utilisée pour le nettoyage et l'enrichissement du jeu de données fourni (section 2). Dans une seconde partie nous décrivons comment les données enrichies peuvent être explorées à l'aide de systèmes d'information logiques (section 3).

2 Nettoyage et enrichissement du jeu de données

2.1 Nettoyage et traitements linguistiques

Jeu de données : Articles RNTI. Le jeu de données fourni pour le Défi [1] est un fichier texte contenant 1937 lignes et où chaque ligne représente un article de recherche publié entre 2004 et 2015. Pour chaque article, 8 champs peuvent être renseignés : *series*, *booktitle*, *year*, *title*, *abstract*, *authors*, *pdf1page* et *pdfarticle*.

Filtrage et nettoyage du jeu de données. Dans un premier temps le fichier de données a été converti d'un encodage Windows à l'encodage UTF8 et les articles ont été filtrés grâce au champ *booktitle* pour ne conserver que les 1103 articles publiés à la conférence EGC entre

1. `RNTI_articles_export.txt` à l'adresse `http://editions-rnti.fr/?m=articles_export`

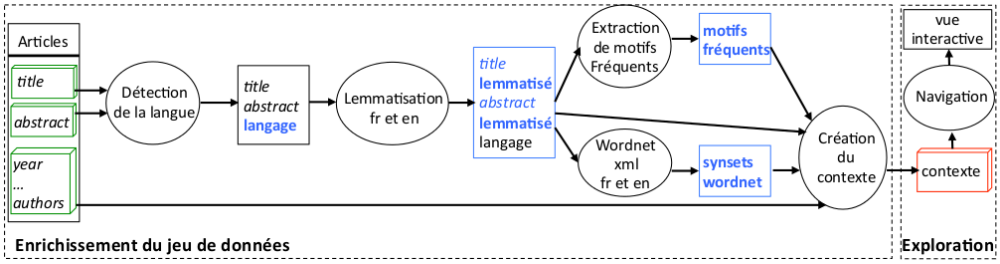


FIG. 1 – *Chaîne de traitement*

2004 et 2015, conformément à la consigne du Défi. Cependant, cette sélection aurait pu être reportée à la phase exploratoire (section 3) pour comparer les publications EGC et non EGC.

Détermination de la langue. Les articles d'EGC peuvent être rédigés en anglais ou en français. La méthode utilisée pour reconnaître la langue utilisée s'appuie sur la commande `curl` avec utilisation de `open.xerox`. Cette étape a été effectuée en considérant les champs *title* et les premières phrases des résumés (champs *abstract*), sans leur ponctuation.

Lemmatisation des résumés et des titres. Les titres et les résumés ont été lemmatisés à l'aide de l'outil TreeTagger [2] (Schmid, 1995) en s'appuyant sur la détermination de la langue. TreeTagger permet de collecter plusieurs information pour un mot : le mot lui-même, le lemme, des Part-Of-Speech (POS) tags. Toutefois pour le Défi, seuls les lemmes ont été conservés.

2.2 Enrichissement avec les synsets

Wordnet [3] est un réseau lexical pour l'anglais, qui sert aussi de référence pour d'autres langues, où des codes *synset* regroupent des ensembles de synonymes. Pour l'anglais, nous avons utilisé la version 3, sous la forme XML proposée par Lapalme (2014). Pour le français, nous avons exploité une autre ressource XML, appelée WoNef, accessible à `wonef.fr` et qui permet de relier les unités de sens dans les deux langues (par les codes synset). Le contexte produit présente les unités de sens par leur codes et ensembles de mots en anglais, permettant de regrouper des mots et de fusionner les deux langues à ce niveau sémantique. Cela pourrait s'étendre à d'autres langues, pour peu que les ressources existent.

2.3 Extraction des suites de lemmes fréquentes

L'objectif de l'extraction des suites de lemmes fréquentes dans les résumés est d'obtenir un ensemble de mots-clés/thématiques associés aux articles. Cette extraction se fait en 3 étapes.

1. Éliminer les mots « vides » dans les résumés. Pour le français nous avons utilisé la liste de mots vides de Jean Véronis [4] et pour l'anglais celle de BaseX [5].

2. `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`
3. `wordnet.princeton.edu`
4. `http://clu.uni.no/corpora/1999-1/0042.html`
5. `http://files.basex.org/etc/stopwords.txt`

2. Appliquer l'outil DMT4SP (Nanni et Rigotti (2007)) [6] avec les paramètres suivants : séquences de longueur 2 à 5, support minimum (minsup) = 5. Selon ces paramètres, 841 motifs sont extraits.
3. Parmi ces 841 motifs, certains ne sont pas pertinents car très communs dans un article de recherche, par exemple {avoir être proposer}, {présenter nouveau approche} ou encore {utiliser technique}. Ainsi les motifs contenant les mots suivants sont supprimés [7] :
   – verbes : avoir, être, présenter, present, utiliser, permettre, pouvoir, montrer, ... ;
   – noms : méthode, approche, résultat, algorithme, cadre, travail, article, papier, ... ;
   – expressions/autres : eg, ie, ne, plus.
   Après cet élagage, il reste 309 motifs comme {classification superviser}, {classification non superviser}, {support vector machine} ou encore {data mining}.

En résumé : Le schéma de la figure 1 résume la méthode employée avec les flux de traitements. La chaîne de traitement fournit des fichiers enrichissant les données de départ, ces fichiers sont aussi transformés en fichiers de *contextes logiques* pour être chargés dans un outil de gestion de contextes logique comme Camelis (Ferré, 2009), ainsi qu'en données RDF pour être explorés par des outils du Web sémantique tels que Sparklis (Ferré, 2014) [8].

3 Accès aux données enrichies et exploration conceptuelle

Nous proposons deux modes d'exploration des données enrichies : exploration d'un treillis de concepts logiques via Camelis ; et exploration d'un point d'accès SPARQL via Sparklis.

3.1 Exploration des suites de lemmes fréquentes avec Camelis

Camelis est basé sur l'analyse de concepts logiques (LCA, (Ferré et Ridoux, 2004)), une extension de l'analyse de concepts formels (FCA, (Ganter et Wille, 1999)). Cette théorie établit une bijection entre un ensemble d'informations élémentaires constituant un *contexte logique* et un treillis de *concepts logiques*. Un contexte logique est défini par un ensemble fini d'objets $\mathcal{O}$, et pour chaque objet o_i, un ensemble fini de descriptions logiques $d(o_i)$, chaque description étant une formule logique. Dans le cadre du Défi, le contexte logique a été obtenu à partir du document initial et de ses enrichissements (voir section 2) où les objets sont les articles. Un *concept logique*, noté c, est un couple formé d'une extension $ext(c)$ (un ensemble d'objets) et d'une intension $int(c)$ (une formule) tel que les éléments de $ext(c)$ sont exactement ceux qui vérifient $int(c)$. Ces concepts forment un treillis selon l'ordre d'inclusion des extensions.

Camelis (version 1 [9]) permet d'explorer le treillis de concepts pour former progressivement une requête. Le scénario d'exploration typique est de partir de la requête `all` (valeur logique = *true*) et de la raffiner progressivement en choisissant des critères de sélection dans un *arbre de navigation*. Les requêtes ainsi formées sont des formules logiques de plus en plus fortes qui déterminent des concepts dont l'extension est de plus en plus petite. La requête (panneau

6. `http://liris.cnrs.fr/~crigotti/dmt4sp.html`
7. Liste complète : `http://www.irisa.fr/LIS/results-fr/defiEGC2016/liste`
8. Toutes les données produites pour le défi sont disponibles à l'adresse : `http://www.irisa.fr/LIS/results-fr/defiEGC2016/`
9. `http://www.irisa.fr/LIS/ferre/camelis/`

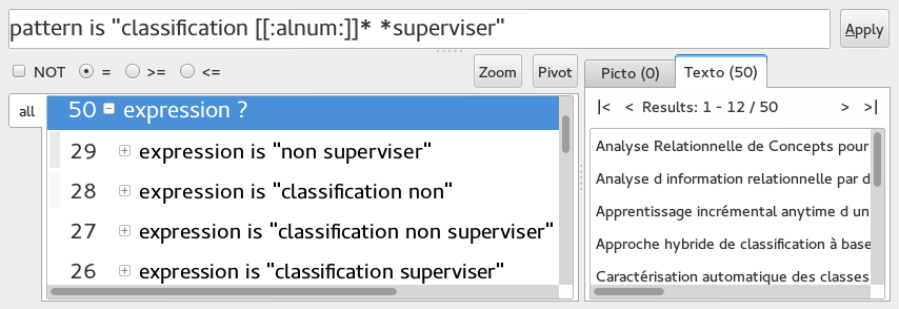


FIG. 2 – *Pattern et expression*

du haut de la figure 2), l'arbre de navigation (panneau de gauche), et l'extension (panneau de droite) sont constamment synchronisés, et sont tous intéractifs ; une sélection ou une modification de l'un des trois éléments cause automatiquement la mise à jour des deux autres. Une propriété fondamentale de cette navigation est qu'il est impossible de former une requête inconsistante (d'extension vide) en ne suivant que les critères de sélection proposés par l'arbre de navigation.

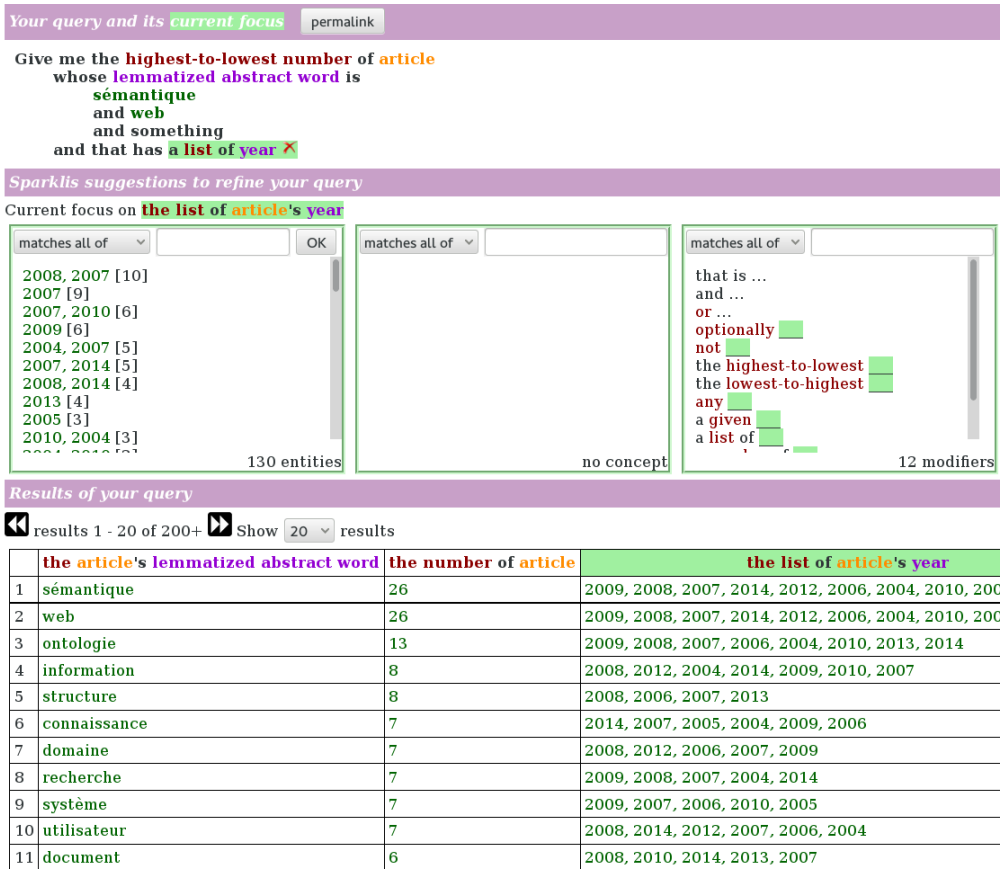
Camelis affiche les critères de sélection de l'arbre de navigation en les préfixant de la cardinalité de leur extension. Cela permet de faire des inférences quantitatives alors même que l'analyse de concepts logiques est fondamentalement symbolique. Par exemple, en examinant la propriété `pattern`, ordonnée par cardinal décroissant, nous constatons que ces motifs sont les plus fréquents : *base ... donnée ; fouille ... donnée ; classification ... supervisée.* Il faut alors vérifier la réalisation de ces motifs où le ... représente un *gap* (représenté par l'expression "[[:alnum :]]* *" dans la figure 2) qui pourrait être le mot *non*. La sélection du motif *classification ... supervisée* puis l'examen de la propriété `expression` qui contient la réalisation des motifs permet de voir qu'en effet le *gap* peut contenir le mot *non* et que c'est la *classification non supervisée* qui est majoritaire (voir figure 2).

3.2 Point d'accès SPARQL et exploration avec Sparklis

Nous avons modélisé une partie des informations présentées à la section 2 en RDF, le modèle de représentation de connaissances du web sémantique (Hitzler et al., 2009). Chaque article y est décrit par son titre, par sa liste d'auteurs et par chacun des auteurs individuellement, par son année, par la langue du titre, par son résumé et par chaque mot lemmatisé du résumé. Afin de rendre ces données publiquement accessibles, un point d'accès SPARQL a été créé [10].

Sparklis [11] (Ferré, 2014) est une application Web destinée à faciliter l'interrogation de points d'accès SPARQL. Il aide les utilisateurs à construire des requêtes complexes sans avoir besoin de connaître ni le langage SPARQL, ni le schéma des données. En effet, le système guide l'utilisateur pas à pas dans la construction des requêtes et présente la requête et les sug-

10. `http://lisfs2008.irisa.fr/defiEGC2016/sparql`
11. `http://tinyurl.com/qbq5qjo`



	the article's lemmatized abstract word	the number of article	the list of article's year
1	sémantique	26	2009, 2008, 2007, 2014, 2012, 2006, 2004, 2010, 200
2	web	26	2009, 2008, 2007, 2014, 2012, 2006, 2004, 2010, 200
3	ontologie	13	2009, 2008, 2007, 2006, 2004, 2010, 2013, 2014
4	information	8	2008, 2012, 2004, 2014, 2009, 2010, 2007
5	structure	8	2008, 2006, 2007, 2013
6	connaissance	7	2014, 2007, 2005, 2004, 2009, 2006
7	domaine	7	2008, 2012, 2006, 2007, 2009
8	recherche	7	2009, 2008, 2007, 2004, 2014
9	système	7	2009, 2007, 2006, 2010, 2005
10	utilisateur	7	2008, 2014, 2012, 2007, 2006, 2004
11	document	6	2008, 2010, 2014, 2013, 2007

FIG. 3 – *Capture d'écran de Sparklis montrant les mots-clés associés à Web sémantique, avec les années de publication*

gestions en langue naturelle. Le guidage garantit que les requêtes construites sont correctes aux niveaux lexical (bon vocabulaire), syntaxique et sémantique (pas de réponses vides).

À titre d'illustration des possibilités offertes par Sparklis, la figure 3 montre les mots-clés les plus fréquemment associés aux mots-clés "web" et "sémantique". La liste des années de publication correspondantes est également affichée. D'un point de vue SPARQL, la requête implique deux agrégations : le nombre d'articles et la liste des années pour chaque mot-clé.

4 Conclusions et perspectives

Nous avons proposé une chaîne de traitements qui produit des données qui peuvent être explorées par des outils de l'analyse de concepts logiques. Les traitements composant cette chaîne ne constituent pas une préconnisation ; ce sont juste des exemples qui permettent un enrichissement linguistique (lemmatisation et synonymes) en français et en anglais. Les don-

nées enrichies peuvent alors être explorées à l'aide des outils Camelis ou Sparklis pour une exploration sûre (jamais de réponse vide, tout ce qui est accessible l'est en suivant l'arbre de navigation) généralisant en particulier les interrogations de type hiérarchique, et bases de données.

Un aspect important de ce travail avec des étapes automatisées est sa réutilisabilité :

- pour de nouvelles années de la revue ;
- pour d'autres données à caractéristiques proches.

Enfin, ce résutat peut être prolongé et complété de plusieurs façons :

- améliorer la qualité des informations obtenues, appliquer d'autres outils linguistiques pour repérer les expressions de façon plus riche et plus précise ;
- exploiter d'autres informations comme le nom de laboratoire et la répartition géographique des auteurs ;
- compléter et exploiter davantage la version RDF ;
- intégrer des taxonomies et de l'inférence (entre patterns et entre unités de sens), avoir ainsi plus de liens de navigation directs.

Références

Ferré, S. (2009). Camelis : a logical information system to organize and browse a collection of documents. *Int. J. General Systems 38*(4).

Ferré, S. (2014). Expressive and scalable query-based faceted search over SPARQL endpoints. In P. Mika et T. Tudorache (Eds.), *Int. Semantic Web Conf.* Springer.

Ferré, S. et O. Ridoux (2004). An introduction to logical information systems. *Information Processing & Management 40*(3), 383–419.

Ganter, B. et R. Wille (1999). *Formal Concept Analysis - Mathematical Foundations*. Springer.

Hitzler, P., M. Krötzsch, et S. Rudolph (2009). *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC.

Lapalme, G. (2014). Wordnet en XML-HTML. In *TALN 2014 - Atelier RLTLN*.

Nanni, M. et C. Rigotti (2007). Extracting trees of quantitative serial episodes. In *Knowledge Discovery in Inductive Databases*, LNCS, pp. 170–188. Springer Berlin Heidelberg.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. In *In Proceedings of the ACL SIGDAT-Workshop*, pp. 47–50.

Summary

This article explains the methodology we used and the results we obtained to answer the challenge in Défi EGC 2016, on data-mining articles published in this conference since 2004. Our approach is based on a fully automated pre-treatment based on public natural language processing ressources for both French and English and on an exploration tool based on logical information systems; these systems rely on formal concept analysis generalized to key-value data or to the semantic web.