Vers une base de connaissances biographiques : extraction d'information et ontologie

Laurent Kevers* et Cédrick Fairon*

* Cental, Université catholique de Louvain (UCL)
Place Blaise Pascal, 1 - 1348 Louvain-la-Neuve - Belgique
laurent.kevers@uclouvain.be - cedrick.fairon@uclouvain.be

Résumé. Le projet B-Ontology a pour but l'extraction, l'organisation et l'exploitation de connaissances biographiques à partir de dépêches de presse. Sa réalisation requiert l'intégration de diverses technologies, principalement l'extraction d'information, les ontologies et bases de connaissances, les techniques de data mining. Cet article propose un aperçu des choix réalisés dans le cadre du projet. Cette démarche permet également de définir un environnement d'outils utiles pour les applications d'extraction et de gestion de connaissances.

1 Introduction

B-Ontology est un projet de recherche appliquée dont l'objectif est de construire le prototype d'une application capable d'extraire et d'organiser de l'information biographique. Cette information sera exploitée dans le cadre du processus de rédaction d'une agence de presse. L'agence Belga diffuse quotidiennement plus de 250 dépêches en deux langues (français et néerlandais). Cette masse textuelle représente environ 70.000 mots par jour (25 millions de mots en un an) par langue. Dans ce projet, nous nous intéresserons aux informations qui concernent les personnes, les organisations et les événements dans lesquels elles interviennent. Le résultat est stocké dans un ensemble de données structurées facilement consultable. Des systèmes comparables existent déjà (NewsExplorer[1], KIM [2]) mais ne couvrent cependant pas toutes les fonctionnalités désirées ici et sont souvent uniquement adaptés aux textes en anglais.

La première partie exposera les méthodes d'extraction d'information. La deuxième s'attardera sur le choix de l'organisation des données. Une troisième partie, présentera une réalisation concrète, mais limitée, de la base de connaissances et quelques aspects de data mining.

2 Extraction d'information

2.1 Définitions des entités et du formalisme d'annotation

L'extraction d'information passe par l'annotation sémantique du texte. Cette tâche nécessite avant tout une bonne définition des types d'entités recherchées. On définit le concept

[1] http ://press.jrc.it/NewsExplorer/home/en/latest.html
[2] http ://www.ontotext.com/kim/index.html

d'*entité* de manière assez large, dépassant ainsi la conception habituelle de l'entité nommée présentée par Chinchor (1998). Dans le cas d'une personne ou d'une organisation, elle peut être constituée uniquement du nom (l'entité nommée au sens strict), mais peut également être accompagnée d'un ensemble d'informations complémentaires. Celles-ci, que l'on définit comme des *entités associées*, sont souvent accolées à l'entité nommée proprement dite. En les regroupant, on obtient une entité complexe. Une nomencalture contenant les principaux types d'entités a été mise au point (voir tableau 1).

codes	commentaires	codes	commentaires
ORG	organisation	NUM	valeur numérique
ORG+INF	org. informelle	NUM+ORD	ordinal
ORG+SUB	partie d'une org.	NUM+FIN	val. financière
PERS	personne	NUM+HUM	val. à unité humaine
PRO	profession	NUM+HUM+PLACE	nombre d'habitants
PLACE	lieu	NUM+SCORE	résultat structuré
PLACE+SUPRA	supranational	TIME	valeur temporelle
PLACE+COUNTRY	pays	TIME+DATE	date précise
PLACE+REGION	région	TIME+PERIOD	période
PLACE+TOWN	ville	EVENT	événement
PLACE+HUM	habitant	RULE	loi, traité, convention
PLACE+ADDRES	adresse	RES	résultat

TAB. 1 – *Codes sémantiques définissant les types d'entités.*

L'annotation adopte le formalisme des dictionnaires DELA, présenté par Gross (1989),Courtois (1990) et Silberztein (1993). Une entité est encadrée par des accolades. On y trouve la forme du texte, un lemme, ainsi qu'une série de codes grammaticaux ou sémantiques (*codes d'entité*) : *{forme,lemme.code-1+...+code-n}*. La partie lemme est facultative et peut être remplacée par un identifiant afin de relier les multiples occurences (éventuellement formulées de manières différentes) d'une entité. Le nombre de codes n'est pas limité, mais il est obligatoire d'en attribuer au moins un. Afin de distinguer les différentes parties d'une entité complexe, on les place entre crochets ('[' et ']') et on leur attribue un code (placé à la fin de la chaîne de caractères et commençant par le signe '#') qui caractérise le type de l'information. Ce code, dénommé *code d'entité associée*, peut être un code classique d'entité (voir tableau 1) ou un code spécifique (voir tableau 2). Les éléments *non significatifs* ou non décrits sont encadrés à l'aide des crochets mais aucun code ne leur est attribué.

codes	commentaires	codes	commentaires
SUB	sous structure d'une organisation (ORG+SUB)	NAT	nationalité
INF	organisation informelle (ORG+INF)	AGE	âge
PRO	profession	REL	religion
FCT	fonction	POL	politique
NAME	nom (d'une ORG, d'une PERS)	TITLE	titre
ALIAS	surnom	PLACE	lieu
DESCR	description	RNK	classement
MODT	modificateur temporel		

TAB. 2 – *Codes sémantiques spécifiques pour les entités associées.*

Tous les codes d'entité (tableau 2) peuvent se retrouver imbriqués dans n'importe quel autre code. Certaines combinaisons ne seront cependant que peu ou jamais rencontrées en raison de

la sémantique trop éloignée des codes en question. Le nombre de niveaux d'imbrication n'est pas limité. À un niveau donné, plusieurs éléments de code identique peuvent coexister.

2.2 Extraction par grammaires locales et transducteurs

L'approche privilégiée consiste en l'utilisation de méthodes linguistiques qui offrent des possibilités d'analyses très fines. Les principaux types de ressource d'extraction sont les transducteurs à états finis (ou graphes) et les dictionnaires électroniques. Les transducteurs constituent un formalisme simple permettant une description très précise des entités à extraire. Pour des raisons de modularité et de facilité dans la construction des graphes, l'extraction s'effectue en plusieurs phases successives (cascade). Le processus mis en œuvre est décrit de manière plus approfondie dans Kevers (2006). L'objectif est de repérer et de catégoriser des entités de plus en plus complexes, telles que celle illustrée ci-dessous :

```
{ [Agé de] [48 ans#AGE] [,] [l'] [ [ex-#MODT] [président#FCT] [du] [ [département IT]
[du] [ [groupe de presse#DESCR] [anglais#NAT] [NewCorp#NAME] #ORG] #ORG+SUB] #PRO]
[,] [le] [socialiste#POL] [français#NAT] [François Pignon#NAME] ,.N+PERS}
```

Pour dépasser le stade de l'annotation des groupes complexes, il faut s'intéresser à l'extraction d'événements. Cela nous mène aux limites d'utilisation des transducteurs. L'expression d'un événement s'étale généralement sur plusieurs phrases, voire sur la dépêche entière. De plus, la richesse du langage naturel implique qu'un fait précis peut souvent être formulé de multiples manières. Il devient difficile de construire des motifs d'extraction pour des éléments si étalés et variables. Une méthode d'analyse syntaxique de surface va être développée pour faire face à ces difficultés.

3 Principes d'organisation et de stockage de l'information

L'information extraite doit être organisée en une ensemble de données structurée facilement adaptable. Cela permet de mettre en place un système minimal, progressivement étendu, qui extrait des données limitées et les stocke dans une structure peu élaborée. Cela implique l'ajout et/ou la modification de types d'entités et de types de relations entre ceux-ci.

La possibilité d'organiser les types d'entités de manière hiérarchisée est également un point intéressant. À mesure que la structure de stockage va se complexifier, il sera judicieux d'arranger les types d'entités de cette manière. Ce principe permet aussi d'utiliser le mécanisme d'héritage ainsi que d'avoir différents niveaux de généralité pour les concepts.

Enfin, nous désirons disposer d'une structure sur laquelle il est possible de réaliser des inférences. La masse de données pourra ainsi être enrichie grâce à la mise en commun des informations extraites et des connaissances de raisonnement incluses au système.

4 Ontologie et base de connaissances

4.1 Un choix : l'ontologie

L'ontologie, concept bien connu dans le domaine du web sémantique, correspond aux exigences exprimées ci-dessus. Elle est définie par Gruber (1993) comme *an explicit and formal*

spécification of a conceptualization. La construction d'une ontologie revient donc à exprimer de manière formelle la perception que l'on a d'un domaine. Cette formalisation (ici de l'information biographique) sert ensuite de modèle pour le stockage de données réelles.

Dans une ontologie, les concepts s'organisent en classes et disposent de propriétés. Celles-ci se rapportent à une autre classe ou à un type de données particulier. Les classes peuvent être organisées de manière hiérarchique et diverses restrictions peuvent être exprimées. Les données réelles qui actualisent les classes sont apellées des instances. L'ensemble des données qui *remplissent* l'ontologie constitue une base de connaissances.

4.2 Choix technologiques pour l'implémentation

Un des principaux langage utilisé pour définir les ontologies est OWL (McGuinness et van Harmelen, 2004). Il existe en trois versions, d'une puissance expressive, mais aussi une lourdeur de manipulation, de plus en plus élevée : Lite, DL et Full. OWL DL constitue un bon compromis. OWL repose sur un ensemble de couches de base : RDFS, RDF, XML et les URIs.

Parmi les outils de développement et de maintenance d'ontologies, nous avons décidé d'utiliser Protégé[3]. Il s'agit d'un logiciel *open source* qui offre une interface graphique permettant de manipuler une ontologie de manière conviviale ainsi qu'une API Java permettant le développement d'applications accédant aux ontologies et aux bases de connaissances. D'autres APIs telles que Jena[4] sont également disponibles. Le guide proposé par Horridge et al. (2004) décrit les concepts et l'utilisation de OWL dans Protégé.

En ce qui concerne les capacités de raisonnement, les raisonneurs OWL (ou *DL-reasoner*) et les langages de règles constituent les principales possibilités. Il est important de souligner qu'en la matière, il n'existe pas réellement de solution *standard*. Beaucoup de langages et d'implémentations cohabitent, sans qu'aucun ne s'impose pour l'instant. Ce contexte induit un certain flou quant au choix de la technologie adéquate.

Les raisonneurs OWL disposent de capacités d'inférence qui sont essentiellement utilisées pour vérifier l'intégrité d'une ontologie ou d'une base de connaissances, ainsi que pour déterminer si une classe est une sous-classe d'une autre classe (ce qui permet d'établir la hiérarchie des classes). Ils se basent sur la logique de description qui est un sous ensemble de la logique des prédicats. Citons entre autres Racer[5], Pellet[6], Hoolet[7] ou encore FaCT++[8].

L'utilisation d'un langage de règles (aussi appelé *règles de Horn*) permet quant à elle d'effectuer des inférences plus complexes. SWRL (Horrocks et al., 2004) semble être la solution émergeante. Il s'agit d'un sous ensemble de la logique des prédicats, mais disjoint de la logique de description.

Enfin, l'implémentation de clients ou de modules aditionnels pour protégé peut se faire à l'aide du langage Java. Cela permet de décliner les clients sous diverses formes : en interface graphique «classique», en application web[9] ou encore en service web[10].

[3] http ://protege.stanford.edu
[4] http ://jena.sourceforge.net
[5] http ://www.racer-systems.com
[6] http ://www.mindswap.org/2003/pellet/
[7] http ://owl.man.ac.uk/hoolet/
[8] http ://owl.man.ac.uk/factplusplus/
[9] Par exemple via Tomcat (http ://tomcat.apache.org)
[10] Par exemple : Axis (http ://ws.apache.org/axis/)

4.3 Structure partielle de l'ontologie biographique

Un premier prototype partiel sert de base au système complet. La définition de l'ontologie commence par la spécification des classes (figure 1, cadre A). Pour l'information biographique, nous avons deux entités principales (*Person* et *Organization*), des entités accessoires (*Country*, *Town*, *Religion*, *Function*, *Title*, *Politic* et *Religion*) et des événements (*Divorce*, *Employment*, *Lawsuit*, *Sentence* et *Wedding*). Ces listes devront bien entendu être complétées. Chaque type d'entité est caractérisé par des propriétés dont la valeur est un autre type d'entité ou un littéral. A titre d'exemple, *Organization* est détaillée à la figure 1 (cadre B).

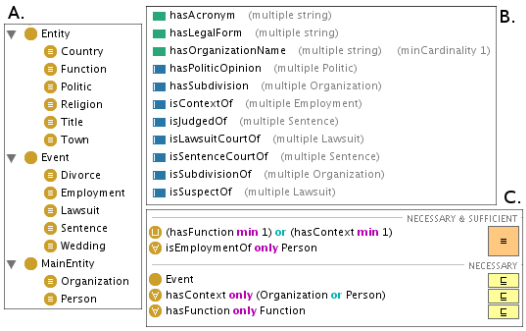

FIG. 1 – *Liste des entités, définition de «Organization» et conditions pour «Employment».*

Les conditions associées aux classes sont des *conditions nécessaires* (si une instance appartient à une classe, elle doit vérifier ces conditions) ou des *conditions nécessaires et suffisantes* (si une instance est membre d'une classe, elle doit remplir ces conditions ; si elle remplit les conditions elle est membre de la classe). Elles font appel à la restriction existentielle ($\exists$; *hasProperty some Class* décrit les instances qui ont au moins une relation *hasProperty* avec une instance de *Class*) ou à la restriction universelle ($\forall$; *hasProperty only Class* décrit les instances dont toutes les relations *hasProperty* s'appliquent uniquement à des instances de *Class*).

Pour *Employment* (figure 1, cadre C), il est spécifié que *a)* toute instance de *Employment* est aussi une instance de *Event* ; *b)* si une instance de *Employment* a une relation *hasContext*, elle porte nécessairement sur une instance des classes *Organization* ou *Person* ; *c)* si une instance de *Employment* a une relation *hasFunction*, elle porte nécessairement sur une instance de la classe *Function* ; *d)* une instance de *Employment* possède au moins une relation *hasFunction* ou une relation *hasContext* ; *e)* si une instance de *Employment* a une relation *isEmploymentOf*, elle porte nécessairement sur une instance de la classe *Person* ; *f)* pour qu'une instance soit membre de la classe *Employment*, il suffit qu'elle soit reliée à une (ou plusieurs) instance(s) de *Person* (et uniquement à cette classe) par la relation *isEmploymentOf* ET qu'elle possède soit au moins une relation *hasFunction* soit au moins une relation *hasContext*.

L'inférence est possible grâce à l'adjonction de règles. SWRL permet par exemple d'exprimer le lien entre frères (et/ou soeurs) : *Person(?x)* $\wedge$ *Person(?y)* $\wedge$ *Person(?a)* $\wedge$ *Person(?b)* $\wedge$ *hasParent(?x, ?a)* $\wedge$ *hasParent(?x, ?b)* $\wedge$ *hasParent(?y, ?a)* $\wedge$ *hasParent(?y, ?b)* $\wedge$ *differentFrom(?x, ?y)* $\wedge$ *differentFrom(?a, ?b)* $\rightarrow$ *isSiblingOf(?x, ?y)*. Cette règle crée une proriété *isSiblingOf* chez x (vers y) et y (vers x) si les conditions sont vérifiées.

5 Conclusion

L'extraction d'information à partir de textes en langage naturel permet d'extraire des données biographiques complexes. Elles peuvent être organisées sous une forme structurée dans une base de connaissances dont l'architecture est définie grâce à une ontologie. Le maintien de la cohérence et la catégorisation de données nécessitent une spécification précise des contraintes sur les classes. Pour enrichir le contenu de la base via un processus d'inférence, des règles doivent être construites. Tout ces éléments forment un ensemble suffisant pour créer une application d'extraction et de gestion des connaissances.

Références

Chinchor, N. (1998). Muc-7 named entity task definition, version 3.5. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, Virginia. Morgan Kaufmann Publishers, Inc.

Courtois, B. (1990). Un système de dictionnaires électroniques pour les mots simples du français. *Langue Française 87*, 11–22.

Gross, M. (1989). La construction de dictionnaires électroniques. *Annales de Télécommunications 44*, 4–19.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition 5(2)*, 199–220.

Horridge, M., H. Knublauch, A. Rector, R. Stevens, et C. Wroe (2004). *A practical guide to build OWL ontologies using the Protégé-OWL plugin and CO-ODE tools* (1.0 ed.).

Horrocks, I., P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, et M. Dean (2004). SWRL : A semantic web rule language combining owl and ruleml. W3C Member Submission.

Kevers, L. (2006). L'information biographique : modélisation, extraction et organisation en base de connaissances. In P. Mertens, C. Fairon, A. Dister, et P. Watrin (Eds.), *Verbum ex machina, actes de la 13eme conférence sur le traitement automatique des langues naturelle (TALN06)*, pp. 680–689. Presses Universitaires de Louvain.

McGuinness, D. L. et F. van Harmelen (2004). Owl web ontology language overview. W3C Recommendation.

Silberztein, M. D. (1993). *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*. Paris : Masson.

Summary

The aim of the B-Ontology project is extraction, organisation and exploitation of biographical data from press dispatches. It requires the integration of various technologies, mainly information extraction, ontologies and knowledge bases, data mining. This paper gives a quick look on the choices taken for the project. This work also try to define a framework usefull to build knowledge extraction and management applications.