A Lexicon Design for Ontology-based Question Answering

Ibrahim Soumana, Sylviane Cardey and Peter Greenfield
Centre de Recherche en Linguistique Lucien Tesnière, Université de Franche Comté,
30 rue Megevand, 25030 Besançon Cedex, Besançon, France

Keywords: Natural Language Interface for Database, NLIDB, NLI, Lexicon, Interclass Lexicon, Intraclass Lexicon, RDF Triple, Ontology, Operator, Function, Natural Language Processing.

Abstract: Data volume growth leads to new challenges for Natural Language Interfaces (NLI). With Big Data for example, NLI must not only be portable from one domain to another, but be operational simultaneously in several domains. The lexicon is an important resource that improves the system performance. In this paper, we propose an approach to design a lexicon centered on RDF (Resource Description Framework) triple. We argue that a triple centric lexicon is reusable. The lexicon is also extended to include operations and possible functions in which data can be involved. This allows increasing the complexity of questions a NLI can process.

1 INTRODUCTION

A natural language interface to databases (or ontology) allows querying a database in natural language without using formal languages such as SQL or SPARQL. Developed since the 70s (Woods et al., 1972), Natural Language Interface (NLI) has in recent years a renewed interest and new challenges. Data produced on the Internet are more and more structured. Government and private organizations are also taking the initiative to publish their publically accessible data through Open Data. Storage models more suited to the abundance and heterogeneity of data have been developed through Big Data. Natural language appears to be the most instinctive way for database querying (Chao et al., 1999). Several studies have shown that users prefer to formulate their queries in natural language (Kaufmann and Bernstein, 2007). A natural language interface to structured data can meet the need for information in the context of local infomediation (Soumana et al., 2012). The Growth in data volume and their heterogeneity reduce the performance of large scale NLI. This reduction in the performance is due to, on the one hand the words that are expressed in the question which do not always correspond to the lexicon used for understanding the concepts of database or the formal language (in which the question is translated) and on the other hand the systems are unable to identify correctly the requested information in the database. Several strategies have been used to increase the performance and portability of NLI. In this paper, we propose a triple centric approach to build a reusable lexicon for NLI. The lexicon is extended to potential operators or functions in which the data may be involved.

The rest of the paper is structured as follows. The following section introduces the concept of triple and argues for the portability of its lexicon. Section 3 summarizes the literature review of NLI focusing on how lexicon is built. Section 4 presents the methodology and an example for cardinal numbers. Section 5 ends with the conclusion.

2 TRIPLE AS PORTABLE ELEMENT

The Semantic Web has developed the RDF (Resource Description Framework) as a data model for describing the web. An RDF triple consists of *(subject, predicate, object)*. RDF defines two types of properties (predicates): *object properties* and *datatype properties*. Object properties (example 2) link two entities of the application (or instances) while *datatype properties* (example 3) link an entity (or instance) to data values.

Soumana I., Cardey S. and Greenfield P..
A Lexicon Design for Ontology-based Question Answering.
DOI: 10.5220/0004540501700175
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2013), pages 170-175
ISBN: 978-989-8565-81-5

subject	predicate	object	
City	mayor	Person	(2)
City	population	integer	(3)
Paris	population	2,243,833	(4)

Example 1 is the syntax of a triple. Example 4 is an instance of the example 3, and means that Paris has a population of 2,243,833 inhabitants. Several NLI use the triple to solve certain lexical ambiguities. For example in this triple (bank, branch, string), the subject *bank* is ambiguous. It can refer to a financial institution or a geographical entity. However with the predicate *branch*, we can say that it is the financial institution rather than a geographical entity. The triple is less ambiguous than the subject, predicate and objects taken separately. Thus the lexicon developed around a triple is also stable for various domains for the same triple. The lexicon developed from a triple can be grouped in two categories: *intraclass lexicon* and *interclass lexicon*. The *intraclass lexicon* is the lexicon that indicates exclusively, a subject, a predicate or an object. *City, population, Paris* are examples of *intraclass lexicon* because they are related to one single class (*city* refers to class city, *population* refers to the predicate population, *Paris* is an instance of the class city). The *interclass lexicon* is related to at least two elements of the triple (subject, predicate or object). For example the word *megacity* has the notion of city, population and a restriction on the size of the population. So *megacity* refers to the subject, the predicate and the object of the triple. These conditions should be taken into account when the question is translated into formal language. Words from the interclass lexicon need additional processing over and above just string matching (as an *intraclass lexicon*).

3 LITERATURE REVIEW

The extension of the lexicon of NLI can increase the portability and recall. Several methods are used to build the lexicon. The initial lexicon and the adaptation (to a new domain) lexicon can be built automatically or manually. The lexicon can be generated automatically from the database (Cardey et al., 2001), PRECISE (Popescu et al., 2004). Some systems like PANTO (Wang et al., 2007) and Aqualog (Lopez and Motta, 2004) use Wordnet (Fellbaum, 1998) to extend the lexicon. Aqualog, in addition to Wordnet, uses machine learning techniques to augment the lexicon. The single use of Wordnet does not allow adequate coverage of the lexicon because it is a general resource. In a database (or ontology), labels are sometimes coded, absent or not meaningful. Automatic recognition using Wordnet is not sufficient. A lexicon engineer is required to extend the lexicon for a new domain ORAKEL (Cimiano et al., 2008) to have a good quality and coverage of the lexicon. In ORAKEL the lexicon is built so that the parse tree of the question can match the ontology concepts. In TEAM (Grosz et al., 1987), the adaptation to a new domain is done by the database administrator.

The lexicon can also be acquired by a set of named entity annotators implemented as a web service (Ou et al., 2008; Soumana, 2010). The annotators are used to identify the concepts of the ontology. FREyA (Damljanovic et al., 2011) and QUERIX (Kaufmann et al., 2006) use user feedback in case of ambiguity. Some systems consider the web as a lexical resource. Online resources like LOD (Linked Open Data) are used to identify the concepts of database (Lopez et al., 2011; Yahya et al., 2012).

Most of the research on lexicon design consists in identifying directly the database concepts (values, properties, entities) in the question. Natural language does not always express the concepts as a bijection from the lexicon to database concepts (classes, properties or values). Except for some fuzzy quantifiers which can be part of the interclass lexicon, the main work in NLI for lexicon design is focused on building an *intraclass lexicon*. For example *big* is a fuzzy quantifier, but it is not part of the interclass lexicon because it does not have a meaning of any specific predicate. Its meaning is only scalar. The fuzzy quantifier *hot* can be part of the interclass lexicon because it has not only the notion of being scalar (object of a datatype triple) but the notion of temperature which can be linked to a specific part of the triple. The meaning has at least two dimensions (scalar and temperature) which can be linked either to the subject, the predicate or with the object of a triple.

In the next section, we present the triple centric model for lexicon design and an example of how it is built with cardinal numbers (values).

4 LEXICON DESIGN

The lexicon is build from basic types up to a triple level. For a datatype property, the object can be: numeric, string, boolean, binary or geometric (one, two or three dimensions for spatial database) values. In traditional linguistics a numeric value can be

cardinal number or *ordinal number*. The lexicon is designed in the form of hierarchical classes. The basic types (numeric, string, boolean, binary or geometric) are subclasses of the object class.

4.1 Class Hierarchy

The vocabulary that occurs within a triple can be divided into 4 classes:

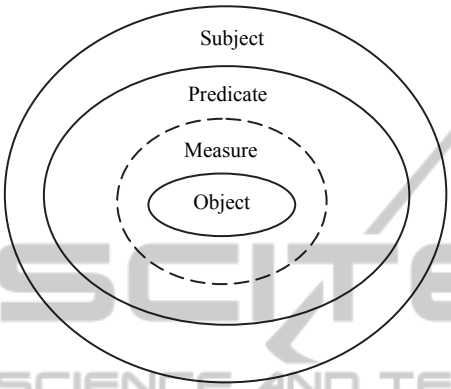

Figure 1: class hierarchy.

- Object class
- Measure class
- Predicate class
- Subject class

The measure class is optional. It occurs when the object is a quantity of measure (like numeric values). The Object class contains vocabulary whose semantics refers exclusively to the object. If the object is a numeric value which is a cardinal, the lexicon generated for this class contains all the vocabulary that expresses only cardinal numbers. So the lexicon should contain all the cardinals *(0,1, 3, etc.)*, as well as words such as *big, little* which have exclusively the notion of number. This class also contains words which refer to a cardinal like *value, number* but without any precision. Words like *long, heavy* are not parts of this class because the notion of the measure is already included: *long* is related to *length* and *heavy* is related to *weight*. The measure class contains both the *intraclass lexicon* of measure and the *interclass lexicon* object and measure. An example of the *intraclass lexicon* content for measure is: length, weight, temperature. These refer to a measure without precision. Another example of the interclass lexicon content is: *long, heavy, hot*. These refer to a measure with some value in the scale of this measure. The value is the object of the triple. For all classes, we have an *intraclass lexicon* and possibly an *interclass lexicon* which is semantically the intersection of the current class and subclasses. An example of the predicate level can be illustrated as following:

Area population integer (5)
What is the population of each area? (6)
What are the populated areas? (7)

With triple (5), we can have the question (6) and (7). The word *population* is from the intraclass lexicon of the predicate because it denotes just the predicate without any restriction. The system should display the population of all areas regardless the number of habitants. In question (7), the word *populated* refers to the predicate population and makes a restriction on the object of the triple. Not all the values are accepted. The system should not display an area with zero habitants. So the word *populated* is in the interclass lexicon of the predicate population. The reuse of the lexicon is done by inheritance. A triple can inherit the subclass lexicon of basic type like the cardinals for example, the lexicon of the class measures, the lexicon of the predicates or the lexicon of subjects. The lexicon is developed once but can be used by many applications according to the level of granularity that is necessary.

4.2 Cardinal Numbers

Cardinal number is a subclass of the class object. It has no interclass lexicon because the main class (the object class) is the lowest class in the hierarchy. The general approach is to determine the *variables*, the *instances*, the *data granularity*, and the *operations* and *functions*. Information is searched in all the syntactic granularities (part of speech tag or multiword expression). The aim of studying the *intraclass* lexicon and the *interclass* at each level of the triple is firstly to identify the concept in the question. The syntactic information will be used subsequently to compute the semantics. The following paragraphs are more focused on identifying the concepts than computing their semantics.

4.2.1 Variable of Cardinal Numbers

A variable is the string used to denote a cardinal. There is no reference to any particular value. Table 1 shows a list of the cardinal *variable* according their part of speech (POS) tag in French. Lists are not exhaustive but are given as an example. The corresponding English word is in brackets.

Table 1: A list of variables for cardinal number.

POS tag	Variable
Noun	valeur (value), nombre (number), chiffre (numeral), donnée
Verb	compter dénombrer chiffrer (to count), évaluer (to evaluate,), s'élever (to rise), calculer (to calculate), estimer (to estimate), énumérer (to enumerate)
Adverb	combien (how much/how many)

4.2.2 Instances of Cardinal Numbers

The instantiation lexicon is rich. Many adjectives and adverbs derived from these adjectives express a cardinal value; see Table 2.

Table 2: A list of cardinal instances.

POS tag	Instance
Noun	million, couple, singleton, multitude
adjective	unique, seule (alone), grand (big), considerable
Adverb	uniquement (only) considérablement (greatly)
Verb	exceller (excel)
Determinant	determinant (singular or plural), number (0, 1, 2,3 etc.), nul, aucun (no)
Pronoun	personne (nobody, none)

In the question *which students excel in math?* the system should understand the word *excel* as an instance of value linked to math.

4.2.3 Operators of Values Generation

The values generation operators allow generating more precise intervals than those that can be found in Table 2. We are interested in the intervals that the natural language can express using at most one operator. These intervals are called basic intervals. They are usually traditional half-lines in geometry. The other intervals can be expressed by composition (using two or more operators). Table 3 presents the basic interval where *a* is a cardinal. In the phrase *exceed a*, *exceed* is the operator and *a* is the argument, and the phrase *exceed a* generates an interval.

The interval [a,b] and intervals centered around a value such as $a \pm \Delta$ (e.g. around 5) where Δ is the variation (amplitude) are also considered as basic intervals according to the previous definition. The symbol "-" can be used to express an interval (e.g. 5-10 meaning from 5 to 10). However the analysis must be extended to the sentence to ensure the operator is an operator of value generation.

Table 3: Basic intervals (half-lines).

POS	[a ; +∞ [	]a ;+∞[	]- ∞ ; a]	]-∞;a[
verb		dépasser (exceed)		
preposition	dès, à partir de (from)	plus de, more than	jusqu'à (until)	moins de, inférieur à (less than)

Table 4: Others basic intervals.

POS tag	[a ;b]	$a \pm \Delta$
verb		avoisiner, approcher, frôler (to approach)
preposition	à, au (to)	autour de (around)
symbol	-	

In the sentence "*The Russian team surprised many observers by fighting back from 0-2 and 2-3"*, the symbol "-" does not indicate an interval.

4.2.4 Operators of Proportion

The operators of proportion can be classed in two categories: coefficient and ratio. The operators with a coefficient indicate the coefficient of the proportion and the sign (multiplication or division). For example double, triple indicates respectively the coefficient two and three and the sign is multiplication. Half, the quarter, the third indicate respectively two, four and three and the sign is division. Certain word (multiple, multiply, split, partition) specifies only the sign of the operation. The coefficient can be found in the context of the word. The operators which indicate a ratio are in general prepositions or symbols (e.g. 9 over 10 or 90%).

Table 5: Operators of proportion.

POS tag	Coefficient	Ratio
nom	double, triple, moitié (half), multiple	
verb	doubler (double), tripler(triple)	
adjective	double, triple, quintuple	
preposition		sur (over), parmi (among), des (of), pour cent (percent)
adverb	doublement (doubling twice),	
symbol		%, /

4.2.5 Comparison Operators

Comparison uses the same operators as those used to generate intervals. In the case of interval generation the result is an interval, for comparison, it is a boolean. Table 6 gives a possible list for equality.

Table 6: Operators of comparison (equality).

POS tag	Operators
noun	égalité (equality)
verb	égaler (equal) , correspondre (correspond), équilibrer (balance)
adjective	même (same), identique (identical), similaire (similar), équivalent (equivalent)
adverb	exactement (exactly)

4.2.6 Logical Operators

The logical operator expresses disjunction, conjunction and negation. Conjunction is marked by words or expression such as *and*, *with*, *like*, *as well as*, or the *comma*. Disjunction is expressed by words such as *or, otherwise*. Negation is marked by *not*, *without*.

Table 7: Logical operators.

POS	conjunction	disjunction	negation
conj.	et (and), aussi bien que (as well as),	or, soit (either..or)	
prep.	puis (then)		sans (without)
adv.		autrement (otherwise)	pas (not)

4.2.7 Set Operators

The arguments of set operators are sets. Union and intersection can have the same operators as conjunction and disjunction respectively. Complement is generally marked by prepositions such as *without, except* or *apart from*.

4.2.8 Arithmetical Operators

The arithmetical operators also allow increasing the complexity of questions. They make it possible to introduce into the questions elementary calculations whose parameters are known or unknown at the time of the question. For example in a database the following question: *"Which directors has a salary more than 4 times the minimum salary of the company?"* requires calculations.

Table 7: Operators for addition.

POS	addition
verb	additionner ajouter totaliser sommer (add)
noun	addition ajout totalité somme (addition)
adjective	entire, total
adverb	en tout, en tout et pour tout, plus (in all)
determ.	tout (all)
symbol	+

4.2.9 Functions

A cardinal may be involved in a function (a sequence of elementary operations). Functions can be of two types: general functions or functions related to the business processes. The general functions are functions such as *powers*, *square root*. Functions related to the business process are domain dependent.

5 CONCLUSIONS

In this paper, a design of the lexicon based on the triplet is proposed. This is structured in 4 hierarchical classes: object, measurement, predicate, and subject. The *intraclass lexicon* and *interclass lexicon* have been defined to take into account the specificity of each one. The class object is at the lowest level. It contains the basic types like the numerical type (cardinal and ordinal), string, boolean, binary and geometric type. Apart from the object class which has only a *intraclass lexicon*, all the classes of the hierarchy can have, in addition, an *interclass lexicon*. An example of the *intraclass lexicon* for cardinals with their operators has been presented. The hierarchy of classes helps to develop the lexicon once and can be reused in as many applications if necessary. The reusability is done by inheritance when the classes are semantically equivalent. A limitation of this approach is that semantic equivalence cannot be done well automatically. In example (5) population can refer to humans, animals or plants. The lexicon developed for these populations is not the same. Human intervention can help to disambiguate as is done in many NLI. Currently we are working on the others basic types and higher level classes.

REFERENCES

Cardey, S., El Abed, W., Greenfield, P., 2001. Exploiting semantic methods for information filtering. In Actes du 3ème Colloque du Chapitre français de l'ISKO

(International Society for Knowledge Organization), 5 et 6 juillet 2001, Université de Paris X : "Filtrage et résumé automatique de l'information sur les réseaux", pp.219-225.

Chao, H. L., Chen C. H., Cardey, S., 1999. Traitement automatique de la sémantique floue dans l'interrogation de base de données en langue naturel. In BULAG n°24, Université de Franche-Comté, Besançon, pp. 187-208.

Cimiano, P., Haase, P., Heizmann, J., Studer, R., 2008. Towards Portable Natural Language Interfaces to Knowledge Bases: The Case of the *ORAKEL System. Data Knowledge Engineering* (DKE), 65(2), pp. 325-254.

Damljanovic, D., Agatonovic, M., Cunningham, H., 2011. FREyA: an Interactive Way of Querying Linked Data using Natural Language. In Proceedings of 1st Workshop on Question Answering over Linked Data (QALD-1), Collocated with the 8th Extended *Semantic Web Conference (ESWC 2011), Heraklion*, pp. 10-23.

Fellbaum, C., 1998. WordNet, an Electronic Lexical Database, MIT Press.

Grosz, B. J., Appelt, D. E., Martin P. A., and Pereira, F.C.N., 1987. TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. *Artificial Intelligence*, 32:173-243.

Kaufmann, E., Bernstein, A., 2007. How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users. In *Proceedings of the 6th International Semantic Web Conference ISWC 2007*, pp. 281-294.

Kaufmann E., Bernstein A., Zumstein R., 2006. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In proceedings of the *5th International Semantic Web Conference (ISWC 2006)*, Athens, GA, pp. 980-981.

Lopez, V., Fernandez, M., Motta, E., Stieler, N., 2011. PowerAqua: Supporting Users in Querying and Exploring the Semantic Web, antic Web. *Journal of Semantic Web Interoperability*, Usability, Applicability., In Press.

Lopez, V., Motta, E., 2004. Ontology-driven Question Answering in AquaLog. In NLDB 2004 *(9th International Conference on Applications of Natural Language to Information Systems)*, Manchester, UK.

Ou, S., Orasan, C., Mekhaldi, D., Hasler, L., 2008. *Automatic Question Pattern Generation for Ontology-based Question Answering*. In Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS2008). Menlo Park, CA: AAAI Press, pp. 183-188.

Popescu, A., Armanasu, A., Etzioni, O., Ko D., YATES, A., 2004. Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability, COLING.

Soumana I., 2010. A Natural Language Interface for SPARQL by means of Hierarchical Categorisation. In Natural Language Processing and Human Language Technology 2010, BULAG n°34, PUFC, ISSN 0758 6787, ISBN 978-2-84867-312-7, pp. 169-185.

Soumana, I., Cardey, S., Greenfield, P., 2012. Use of Natural Language Interfaces and Open Data in Local Infomediation. In *Proceedings of International Conference on Management and Service Science (MASS)*, Shanghai, China, 10-12 August 2012, 4 pages, CD-ROM.

Wang, C., Xiong, M., Zhou Q., Yu, Y., 2007. PANTO: A Portable Natural Language Interface to Ontologies. *European Semantic Web Conference (ESWC2007)*, Innsbruck, Austria, pp. 473-487.

Woods, W., Kaplan, R., Webber, B., 1972. The Lunar Sciences Natural Language Information System: Final Report. *Technical report, Bolt Beranek and Newman Inc., Cambridge, Massachusetts.*

Yahya, M., Berberich, K., Elbassuoni, S., Maya, Ramanathz V. T., Weikum, G., 2012. Deep Answers for Naturally Asked Questions on the Web of Data. In *WWW'12 Proceedings of the 21st Annual Conference on World Wide Web Companion*, pp. 445-449.