Natural Language-based User Interface for Knowledge Management System. A Computational Linguistics Approach.

Mario Monteleone, Maria Pia di Buono and Federica Marano
University of Salerno, Fisciano (SA) – Italy
mmonteleone@unisa.it, mdibuono@unisa.it, fmarano@unisa.it

Abstract. This research wants to show how it is possible to convert natural language (NL) queries into formal semantic ones, by means of a procedure which allows to semi-automatically map natural language to formal language. More specifically, focusing on voice and/or keyboard-based natural language user interfaces, this research wants to explain how to simplify and improve human-computer natural language interaction and communication. Also, in a more wide perspective, it wants to individuate a method for the creation of Natural Language Processing (NLP) applications finalized to the achievement of Question-Answering (QA).
The NLP activities sketched in this research fall inside Lexicon-Grammar (LG) theoretical and practical framework, which is one of the most consistent methods for natural language formalization, automatic textual analysis and parsing. This framework is independent from those factors that are crucial within other approaches, as those concerning the interaction type (voice or keyboard-based), the length of sentences and propositions, the type of vocabulary used, and the restrictions due to users' idiolects.
Another feature is the possibility to process unstructured, semi-structured or structured information retrievable from either knowledge management system (KMS) or on-line repository, also considering that all other approaches mainly use interfaces which dialogue with structured data.
This approach allows to overcome users' limits about domain ontology knowledge, and to define relationships between search terms to be considered.

Keywords: Natural Language Interface, Knowledge Management System, Lexicon-Grammar, SeRQL, Cultural Heritage

1. Introduction

Building natural language interfaces (NLIs) is not only answering questions on the basis of a given database or knowledge base, but also accessing structured data in the form of ontologies and unstructured data.

Therefore, in this paper, we propose a framework for converting natural language (NL) queries into formal semantic ones, by means of a procedure which allows to semi-automatically map natural language to formal language. More specifically, focusing on voice and/or keyboard-based natural language user interfaces, this research wants to explain how to simplify and improve human-computer natural language interaction and communication. Also, in a more wide perspective, it wants to individuate a method for the creation of Natural Language Processing (NLP) applications finalized to the achievement of Question-Answering (QA).

The NLP activities sketched in this research fall inside Lexicon-Grammar (LG) theoretical and practical framework, which is one of the most consistent methods for natural language formalization, automatic textual analysis and parsing. LG method gives us the theoretical basis to imagine and work towards a linguistically motivated system in which any type of user is able to obtain the exact information he/she is looking for. This aim seems easy to obtain, but the first trial, not yet surmounted, is to digit a query using sentences in natural language. Nowadays, humans usually make efforts in "translating" that query

into proper keywords, or even into non-acceptable[1] sequences of nouns and/or adjective which they never would use in ordinary communication. A second more important trial is that generally speaking outputs are full of noise, so humans have to filter results to obtain the information they need. In order to achieve effective IR and IE results, any KM system, whether closed or open (i.e. the World Wide Web), could avoid most of the noise if it worked with ontologies developed taking into account syntactic, lexical and semantic rules (under W3C criteria); or also, if it could be linked to data and document repositories of to extract proper and updated information (IST, or Information Storage Techniques), therefore making the Web more semantic. On the basis of such premises, the system outlined here will stick to the following steps:

1. a linguistic analysis inside an NLP environment;
2. an iterative transformation finalized to the accomplishment of a machine-readable query;
3. the execution of a query against a knowledge base;
4. the display of results.

It is worth stressing that the processing phase shown in step 2 is of crucial importance for reconstructing conceptual relationships among query terms, and also in order to retrieve a meaningful value from subjects' text sequences. We will see that as for query words, the matching process to the ontology concepts is also based on domain labels which semantically tag (i.e. denote/connote) each entry of simple-word and multi-word electronic dictionaries[2]. Also, within our NLP environment, finite-state automata (FSA) and finite-state transducers (FSTs) are used to:

- recognize and classify word relationships inside the query propositions entered/chosen by the user;
- parse lexical ambiguity.

Indeed, FSA and FSTs are typically applied to locate morph-syntactic patterns inside corpora and extract matching sequences, in order to build indices, concordances, etc. This FST/FSA-based method, which is already available inside our NLP environement, is also likely to be used to automatically recognize any kind of text pattern.

And again with reference to this environment, an API represents the ideal solution to:

- build an interface providing procedures callable by means of external processes;
- drive the application for translating NL queries into Sesame RDF query Language (SeRQL).

1.1. Background

For several years, we will see that similar projects and demonstrations proposed data management solutions based on NLIs. Existing proposals share similar goals focusing on the development of applications that satisfy required flexibility in order to support the user's view of a given domain. Many of these works have been focused on the use of machine learning algorithms for mapping NL questions to query languages. Indeed, automatic interpretation of natural language is very difficult to achieve, for a specific obstacle encountered in NLIs is the resolution of ambiguity, a problem which is mentioned in various overviews on NLIs [1,8].

Different design approaches have been used for implementing several tools which present various levels of expressivity and user-friendliness.

An example is SemSearch [22] a semantic search engine which uses classes or instances as queries. This system requires a deep knowledge of the specific domain ontology.

Another ontology-driven system is Aqualog [23], which applies parsing and WordNet and is based on a controlled language with learning mechanisms.

At the same time, Orakel [6] supports questions with quantification, conjunction and negation, but it necessitates a customization for being ported to a new domain.

Querix [21] converts natural language into SPARQL and it solves language ambiguities dialoguing with users.

Anyway, our approach is founded on a not statistically-based linguistic formalization which ensures a low degree of ambiguity, a low loss of meaning and an accurate matching between linguistics structures, domain concepts and programming language.

[1] According to [4], acceptable phrases or sentences of a given language are built according to the agreed-upon syntactic rules of that language.

[2] A detailed definition of electronic dictionaries is given in 2.1, 3.2.

2. Methodology and Tools

Our linguistic methodology is based on the Lexicon-Grammar (LG) theoretical and practical analytical framework. LG theory was set up by the French linguist Maurice Gross during the '60s [14,15]. It assumes that natural language formal description must start from the observation of lexicon and of lexical entry combinatory behaviours, encompassing syntax and, also, lexicon. It differs from the best known among current linguistic theories, i.e. Chomsky's deep grammar and its various offspring [4,5], which is strictly formalist and syntax-based. LG has also reached important results in the domain of automatic textual analysis and parsing, with the creation of software and lingware fully oriented toward NLP, such as NooJ[3], and the oldest softwares already used in LG framework INTEX and UNITEX[4].

As previously mentioned, LG invests lexicon, and especially the concepts of "meaning unit", "lexical unit" and "word group". Of course, the first problem in the Multiword Units (MWU) treatment is the identification of strings of words properly representing strings of "words related to each other". Subsequently, we interpret and formalize the syntactic structure of the collected MWUs by classifying them [19] as Part of Speech patterns[5] (POS) and analyzing their semantic properties (Semantic Tagging). Then we define when a MWU is used compositionally or non-compositionally. Linguistic Resources (LRs) developed in this way are used in NLP applications and are useful to achieve effective semantic tagging. Nowadays, LG describes both Indo-European languages (French, Italian, Portuguese, Spanish; English, German, Norwegian; Polish, Czech, Russian, Bulgarian; Greek) and others (Arabic; Korean; Malagasy; Chinese; Thai...). The fundament of LG framework is the "simple sentence"[6] is the minimal linguistic meaning context that can be analyzed; on the basis of this "simple sentence" it is possible to achieve concrete studies on natural languages. In addition, the study of simple sentences is achieved by analyzing the so-called rules of co-occurrence and selection restriction, i.e. distributional and transformational rules based on predicate syntactic-semantic properties.

Transformational rules (active/passive, positive/interrogative, etc.) highlight mutual relationships between simple sentences as observed by Zellig Harris [20] starting from the bloomfieldian notion of morpheme and from method of commutation or equivalence between different morphemes' available lexical stuffs [2]. LG theory is prevalently based on the concept of Operator-Argument Grammar [17][7]. As previously stated, LG range of analysis invests lexicon, and especially the concept of MWUs as "meaning unit", "lexical unit" and of "word group", for which LG identifies four different combinatorial behaviours (see also [10]):

- Combinations with a high degree of variability of co-occurrence between words. In this case we have combinations based on open distribution with a compositional and signified meaning;
- Combinations with a low degree of variability of co-occurrence between words. In this case we have combinations based on constrained distribution;
- Combinations with zero or almost zero degree of variability of co-occurrence between words. In this case we have combinations based on fixed distribution;
- Combinations without variability of co-occurrence between words. In this case we have proverbs.

Relations between these mentioned classes could be interpreted not only as relations between separated classes, but also as relations between poles of a continuum. We give here some examples of these combination classes:

a) (combinations at point 1.)

- Verbal structures: (Max, Mary, your nephew,...) looks at (a book, the river, Eva,...)
- Nominal structures: (clean, dirty,…) water
- Adverbial structures: with (elegance, love, devotion,...)

b) (combinations at point 2.)

- Verbal structures: (Max, Mary, your nephew,...) dries (the clothes, the laundry,…)
- Nominal structures: (mineral, sparkling, natural,…) water
- Adverbial structures: from one (moment, day, year,...) to the other

c) (combination at point 3.)

[3] See http://www.nooj4nlp.net/pages/nooj.html.

[4] More information on the website http://www-igm.univ-mlv.fr/~unitex/.

[5] According to Manning and Schütze [24] we consider POS "a part of the grammar of a language which includes the lexical entries for all the words in the language and which may also includes other information".

[6] In LG, a simple sentence is a context formed by a unique predicative element (a verb, but also a name or an adjective) and all the necessary arguments selected by the predicate in order to obtain an acceptable and grammatical sentence. For more specification on simple sentence definition [14].

[7] Regarding the topic see also the Valency Theory developed by the French linguist Lucien Tesnière [32,33].

- Verbal structures: (Max, Mary, your nephew,...) bends his elbow
- Nominal structures: heavy water, arsenic water
- Adverbial structures: in no uncertain terms

d) (combination at point 4.)

- Proverbs: Walls have ears.

From a semantic point of view and for disambiguation tasks, we observe that types (c) and (d) may also have "idiomatic" interpretations, or rather interpretations that are not semantically compositional (i.e. not coming from a compositional computation of each lexical element meaning). Probably, some of these fixed and idiomatic combinations are the result of metaphoric and metonymic drifts, which have been lexicalized.

Starting from these assumptions, we may deduce that the use of the four mentioned combination types originates from the need for incisive and immediate communication processes rather than for ordinary ones. While metaphor and metonymy, as any figure of speech, involve an additional operation of decoding and interpretation, fixed and idiomatic combinations are used as a single block: they are semantic shortcuts, and it is not necessary to know the meaning of each element of the linguistic sequences they are conveyed by. It is important to stress that in LG, all types of lexical entries can be formalized, coherently inserted inside linguistic databases (i.e. electronic dictionaries), and used within NLP routines, as for instance information retrieval and parsing. To clarify, the LR built in this way and managed using the above-mentioned criteria, are useful to effective semantic tagging. Furthermore, our research is part of a complex LG study on speciality languages (see also [11],[16]).

2.1. Resources and Tools

Our LRs consist of electronic dictionaries morphologically and semantically tagged; local grammars in the form of Finite State Transducers/Automata (FST/FSA); and tables presenting lexical entry syntactic-semantic properties.

An electronic dictionary is a lexical database homogeneously structured, in which the morphologic and grammatical characteristics of lexical entries (gender, number and inflection) are formalized by means of distinctive and non-ambiguous alphanumeric tags. As for the differences existing between electronic and computerized dictionaries, it has been high-lighted [35] that the term "computerization" has somehow confused the two categories. Print modernization processes requires that the texts of conventional paper dictionaries are typographically composed on computerized media, but this computerization process does not affect the content of these dictionaries, which remains un-changed. So, both paper and computerized dictionaries are only used by humans having a solid and already existing expertise. On the contrary, electronic dictionaries are only used by computers within specific software routines, and are managed by specialized human users. Therefore, the data included inside electronic dictionaries are formalized by means of codes which are not intelligible to common readers. On such basis, it is also possible to state that a classic paper dictionary is not fully (re)usable within automatic textual analysis.

All electronic dictionaries built according to LG descriptive method form the DELA[8] System, which works as a linguistic engine embedded in automatic textual analysis software systems, and parsers. DELA electronic dictionaries are of two types:

- simple word (DELAS 135,000 simple words and DELAF 1,200,000 inflected simple words), which include lexical units semantically autonomous and formed by sequences of characters delimited by blanks. This is the case of words such as home and chair;
- compound word (DELAC 154,000 compound words and DELACF 480,000 inflected compound words collected in dictionaries of specific domains), which include lexical units composed of two or more simple words having an overall meaning. This is the case of sequences such as *nursing home*, and *rocking chair*.

As already stated, terminological entries are mainly lemmatized in compound word electronic dictionaries.

Together with electronic dictionaries, local grammars are used in NLP routines. Local grammars are useful to cope with specific characteristics of natural language; more appropriately, local grammars design is based on syntactic description, which encompasses transformational rules and distributional behaviours [18]. We develop local grammars in the form of FSA/FST [29,30].

To develop and test electronic dictionaries and local grammars we use the software NooJ.

NooJ is a complex NLP environment in which it is possible to automatically read digitized texts and retrieve from them specific linguistic patterns in the

[8] Dictionnaire Électronique of LADL (Laboratoire d'Automatique Documentaire et Linguistique).

form of concordances. NooJ engine is based on the DELA system of electronic dictionaries, on LG syntactic tables and on FSA/FST, developed in the form of graphs and used in LG to parse texts.

3. Experiment and Results

Starting from this NLP theoretical and practical framework, in this project we propose to build an User Interface for KMS which takes as input a NL query from a user, converts it to a SQL query based on domain semantics and database schema, retrieves appropriate data from the database and returns the output to the user. The basic mechanism involves the following iterative transformation:

- the system acquire domain semantics from terminological electronic dictionaries in form of lexical databases;
- it recognizes a NL query by means of local grammars which formalize the query in a linguistic structure;
- it transduces it in an SeRQL path expression.

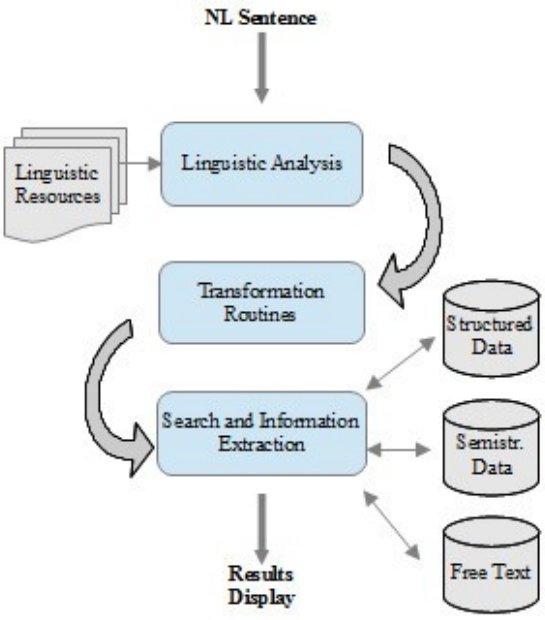


Fig. 1. System Architecture.

As depicted in Figure 1, the process starts with a linguistic analysis, which is based on well defined Linguistic Resources. Then, transformation Routines are applied to map NL into a RDF-triple graph. Finally, the returned SeRQL query is executed against a knowledge base, in order to extract information and present them to users.

Due to all these premises, the process here depicted produces a hybrid architecture, both into the NL analysis and in the usable document base. As we will see, NL analysis is hybrid as it copes with strings which are not composed only by words, but also by morph-grammatical tags (i.e. *N* for any noun, *V* for any verb, and so on). Also, it is hybrid because it may employ three types of information sources, i.e.: (i) unstructured text, (ii) semi-structured and (iii) structured data.

3.1. Domain modeling and ontology

We have chosen the Archaeological domain to test the applicability of our approach. This choice allows us to demonstrate that the modularity of our architecture may be applied to a domain which is variable by type and properties and is semantically interlinked.

As for ontologies, the formal definition we rely upon is the one given by the International Council of Museums - Conseil Interational des Musees (ICOM – CIDOC) Conceptual Reference Model (CRM), which defines that "a formal ontology (is) intended to facilitate the integration, mediation and interchange of heterogeneous cultural heritage information" [9]. CIDOC CRM is composed by 90 classes (which includes subclasses and superclasses) and 148 unique properties (and subproperties). The object-oriented semantic model and its terminology are compatible with Resource Description Framework (RDF). Actually, this ontology was already available and is constantly developed. At the same time**,** our methodology shows that a given linguistic knowledge can be reused independently from the domain to which it pertains. Actually, domain ontologies refer to mid- and upper-level ones, which tend pragmatically to be standardized. Logically, such process indirectly involves also low-level ontologies, and this allows the reuse of linguistic resources regardless the domain in which they were developed or to which they pertain.

Therefore, LG electronic dictionaries and local grammars may together represent the linguistic (lexical, morpho-syntactic and semantic) engine of the KMS [25]. In order to clarify this approach, it becomes necessary to describe the LRs we used to develop our system.

3.2. Lexical and Ontological Databases Construction

The development and management of a lexical and ontological database in form of electronic dictionary consist of three main steps [25]:

- *Lexical acquisition*. During this on-going phase, MWUs are extracted from corpora and/or certified glossaries and continuously updated.

- *Morpho-grammatical and syntactic tagging*. Each lexical entry is given an inflectional paradigm, in order to be inflected. The following string gives a sample of this morpho-grammatical formalization procedure:

ordine dorico, N + NA + FLX = C523 + DOM = RA1EDEAES + ENG = The Doric order

The tag "N" (noun) indicates the grammatical function of the whole compound. The elements that form the morphologic and grammatical patterns of each compound structure - "NPN" (noun + preposition + noun), "FLX=C523" indicates the gender and the number of the compound; also it gives instruction on how to derivce all its possible inflected forms; "DOM=RA1EDEAES" stands for Archaeological Artefacts – Building – Architectural Elements – Structural Elements (terminological tag referring to the electronic dictionary of Archaeological Artefacts) - are followed by the English translation.

- *Testing on corpora*. The dictionary is used to automatically analyze and process large corpora.

3.2.1. Formalization of Lexical Structures

In order to acquire information on compound words formation processes, we identify the typologies of MWU structure in the dictionary, as shown in the following table:

Table 1

Sample of morpho-syntactic POS of MWUs

N° of constituents in the lexical unit	POS tags	Example
bi-gram	NA NN …	freccia foliata (RA1SUOIL) ascia piccone (RA1SUOA-RAB) …
tri-gram	NPN NPN …	fregio con coronamento (RA1EDEAES) freccia di balestra (RA1-SUOARAL) …
fourth-gram	NAPN …	ansa cornuta a manubrio (RA1SUOIL) …
fifth-gram	NPNPN …	antefissa a testa di Gorgone (RA1EDMC) …
…	…	…

The following example represents an excerpt extracted from the Italian dictionary of Archaeological Artifacts[9]:

freccia di balestra, N + NPN + FLX = C45 + DOM = RA1SUOARAL

freccia foliata, N + NA + FLX = C556 + DOM = RA1SUOIL

fregio con coronamento, N + NPN + FLX = C12 + DOM = RA1EDEAES

fregio dorico, N + NA + FLX = C523 + DOM = RA1EDEAES

fuseruola biconica, N + NA + FLX = C547 + DOM = RA1SUOCF

fusto a spirale, N + NPN + FLX = C7 + DOM = RA1EDEAES

3.3. Local Grammars Construction: the Transition from NL to RDF Graph

We have seen how a linguistic pre-processing phase may be achieved to formalize natural language strings into reusable linguistic structures. Such structures have the form of knowledge databases, which are transformed into local grammars (FSA/FSTs) for mapping NL query to RDF, and constructing a virtual graph capable to retrieve coherent information.

[9] It's important to specify that our domain dictionaries, collected in the DELAC system, cover about 180 different semantic tags. The most important dictionaries are those of Informatics (54,000 entries ca.), Medicine (46,000 entries ca.), Law (21,000 entries) and Engineering (19,000 entries ca.). Each dictionary has been created and verified under the supervision of domain experts. Subset tags are also previewed for those domains that include specific subsectors. This is the case of Archaeological Artefacts dictionary (9,200 entries ca.) , for which a generic tag RA1 is used, while more explicit tags are used for object type, subject, primary material, method of manufacture, object description.

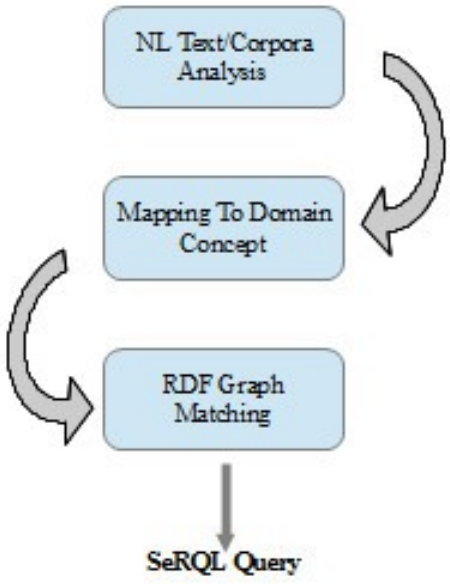


Fig. 2.Transformation Routine.

Figure 2 shows the process for converting NL text in a SerQL Query. During this process, LRs are used for analyzing corpora to retrieve phrase recursive structures, in which combinatorial behaviours and co-occurrence between words identify properties, also denoting a relationship.

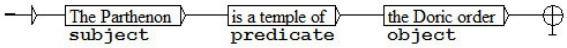


Fig. 3. Simple FSA/FST with RDF Graph.

Figure 3 is a sample of an automaton showing an associated RDF graph for the following sentence:

The Parthenon (subject) is a temple of (predicate) the Doric order (object)

According to our approach, electronic dictionaries entries (simple words and MWUs) are the subject and the object of the RDF triple [Fig. 4].

Fig. 4. Subject/object extracted from FSA/FST.

Further, our NLP environment allows us to transform our graph as follows:

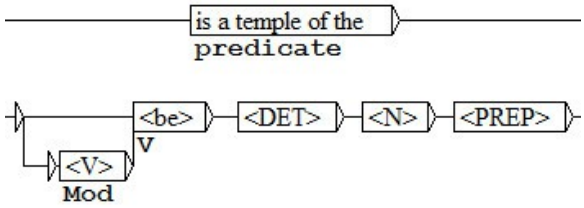


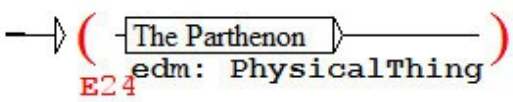
Fig. 5. Predicate extracted from FSA/FST.

As we can see, figure 5 represents the predicate of our triple, in which we have reported our original word combination together with the morph-grammatical description of such combination, in which linguistic tags introduce a high degree of variability, i.e. describe also other instances of the class. Thanks to this, the FSA we built can recognize other combinations of words having the same distribution inside the same class, as for instance:

Notre Dame is a cathedral of the Gothic style

Words in angle brackets stand for lemma forms. When the word form is set between angle brackets, the software locates all the word forms that are in the same equivalence set as the given word form (generally all inflected, derived forms, or spelling variants of a given lexical entry).

Furthermore, a subgraph can freely be embedded in more general graphs. Graph embedding allows the reuse of subgraphs in more than one context. At a more theoretical level, it introduces the power of recursion inside grammars. Subgraphs may also be used to represent a semantic class and can be encoded in a dictionary with specific semantic features. Such features are represented by tags which denote/connote all lexical entries of electronic dictionaries. Such can also be used in the definition of local grammars. Also, electronic dictionaries entries identify a class instance of RDF schemas:[10] in fact, we may use a variable to create the rdf:class through the domain labels, and the rdfs:subClassOf through the subsector labels.



[10] For our examples we refer to Europeana Data Model (EDM) v. 5.2.3 and CIDOC Conceptual Reference Model (CRM) Ver. 5.0.

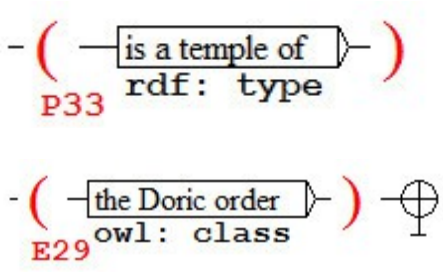


Fig. 6. Sample of the use of the FSA variables for identifying classes for subject, predicate and object.

In Figure 6 we develop an FSA with a variable which applies to the sentence the following classes and property:

- E 24 indicates "Physical Man-Made Thing" class;
- P33 stands for "Used specific technique" property;
- E29 indicates "Design or Procedure" class.

So , the FSA variables transform our sentence into:

The Pantheon (E24) *used specific technique* the Doric order (E29).

The role pairs *physical man-made thing/name* and *design or procedure/type* are trigged by the RDF predicate *is (a) temple of*.

Applying the automaton in Fig. 5 (built using the high variability of lexical class and not of the original form) we can recognize all instances included in E24 and E29 classes, the property of which is P33.

As we have seen, we choose to use affirmative sentences for mapping linguistic structures to corresponding concepts in the domain ontology. This is due to the fact that generally speaking affirmative sentences are more predictable and reusable from the point of view of word distribution. Therefore, such feature:

- grants a coherent identification and extraction of ontological constrains;
- simplifies the process of NL question-answering procedure, because it is based on a consistent reusable repository of pre-constituted sentence descriptions.

FSA/FSTs may also be used to account for all possible grammatical transformation of a given word combination. In our case, this feature allows to transform affirmative sentence into an interrogative one. During this process the semantic representation is preserved.

The differences between such alternative versions do not pertain to syntax, but to lexicon, therefore FSA is useful for mapping various types of interrogative sentence. Due to the transformation routines, the sentence, presented in Fig. 3, may have these forms:

Is the Parthenon a temple of the Doric order?
What kind/type of temple is the Parthenon?

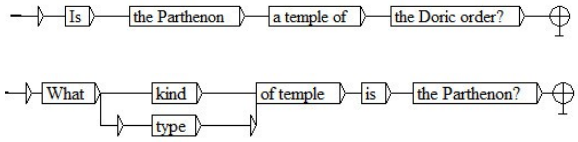


Fig. 7. FSA used for recognizing interrogative sentences.

Fig. 7 shows the matching between the structures identified in Fig. 6 and the new ones, also alternative versions of the interrogative sentence are given.

3.4. Information extraction

Querying information in a RDF framework means to specify path expressions. Our architecture aims to be useful with the query language SeRQL.

Our specific interest is based on various practical observations:

SeRQL uses a path expression syntax which is based on the graph nature of RDF. The path is composed by a collection of nodes and edges and it has an arbitrary length[11].

Indeed, when user queries for two or more triples with identical subject and predicate, the subject and the predicate do not have to be repeated. A multi-value node and branches can be used:

{subj1} pred1 {obj1, obj2, obj3}

This path expression is equivalent to:

{subj1} pred1 {obj1},
{subj1} pred1 {obj2},
{subj1} pred1 {obj3} [3]

[11]Most current RDF query languages define path expression of length 1 and use them to find combination of triples in an RDF graph.

This procedure is very close to the linguistic NL features of transformation, deletion and reduction, which are present in sentence pairs/triples as:

1. *Max eats an apple → Max eats*
2. *Max washes Max's hair → Max washes his hair*
3. *Max eats an apple + Max eats a banana → Max eats an apple and a banana*
4. *Max has a walk → Max walks*

In SeRQL we can apply a restricted form of disjunction through optional matching, and also use existential quantification over predicates and Boolean constraints.

In Fig. 7 we have the two concepts "Production" and "Design or Procedure" associated with different attributes.

Therefore, a SQL interpretation of the question in Fig. 7 could have the following form*:*

*SELECT **
FROM production
WHERE name="Parthenon"
*SELECT **
FROM design
*WHERE type="*order"*

The information is displayed by merging the two results.

In SeRQL, instances of concepts are identified by variables in the subject position of an RDF triple and returns sets of RDF statements.

The query presented in Fig.7 can be solved with the following sample (i.e. prototype) path expression:

*SELECT **
FROM
edm:PhysicalThing {PhysicalThing}
owl:ObjectProperty {rdf:about="P33.used_specific_technique"}
rdfs:range {rdf:resource="E29.Design_or_Procedure"}
WHERE
PhysicalThing LIKE "Parthenon"

Where {Production} is a variable representation of Subject; owl:ObjectProperty is the predicate; and rdfs:range is the variable representation of the Object.

4. Discussions and Conclusions

In this paper, we approach the problem of converting NL queries into programming language.

Our framework is based on a robust definition languages, which extend and create grammars and lexicons.

The aim is to generate metadata representation from natural language inputs. The program outputs RDF graph and SeRQL query representations of a sentences, clauses, and phrases. Furthermore, our architecture ensures a high degree of portability; indeed the specifications are designed to allow the processing of highly complex sentences and phrases of any language and covering any vocabulary.

Future work will provide a deep evaluation of our prototype system. We will apply an evaluation processes for estimating quality and efficiency of sentence mapping and information extraction. We will also perform a comparative evaluation of our system to other results in this area.

References

[1] Androutsopoulos I., Ritchie G. D., and Thanisch P., Natural language interfaces to databases: an introduction, in: Journal of Language Engineering, 1(1), Cambridge University Press, 1995, pp. 29-81.

[2] Bloomfield L., Language. Henry Holt, New York, 1933.

[3] Broekstra, J., Kampman A., An RDF Query and Transformation Language, in: Semantic Web and Peer to Peer, S. Staab & H. Stuckenschmidt, eds., Springer Berlin Heidelberg, Berlin, 2006, pp 23-29.

[4] Chomsky N.A., Syntactic Structures. Mouton, The Hague, Paris, 1957.

[5] Chomsky N.A., Aspects of the Theory of Syntax, MIT Press, Cambridge, Massachusetts, 1965.

[6] Cimiano, P., Haase, P., Heizmann, J., Porting natural language interfaces between domains: an experimental user study with the orakel system, in: IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces, New York, ACM, 2007, NY, USA, pp. 180-189.

[7] Cimiano P., Minock, M., Natural Language Interfaces: What Is the Problem? - A Data-Driven Quantitative Analysis. NLDB, 2009, 192-206.

[8] Copestake A. and Sparck Jones K., Natural language interfaces to databases, in: Knowledge Engineering Review, 5(4), Special Issue on the Applications of Natural Language Processing Techniques,1989, pp. 225-249

[9] Crofts N., Doerr M., Gill T., Stead S., Stiff M., eds., Definition of the CIDOC Conceptual Reference Model, Version 5.0, 2008.

[10] De Bueris G., Elia, A., eds., Lessici elettronici e descrizioni lessicali, sintattiche, morfologiche ed ortografiche. Plectica, Salerno, 2008.

[11] Elia A., Le verbe italien. Les completives dans les phrases à un complement, Schena-Nizert, Fasano di Puglia – Parigi, 1984.

[12] Elia, A., Vietri, S., Postiglione, A., Monteleone, M., Marano, F., Data Mining Modular Software System, in: Proceedings of The 2010 International Conference on Semantic Web & Web Services, WorldComp 2010 Conference, Arabnia H.R., Marsh A., Solo A.M.G., eds., CSREA Press, 2010, 127-133.
[13] Giordani, A., Moschitti, A., Semantic mapping between natural language questions and sql queries via syntactic pairing, in: NLDB '09: Proceedings of the 13th international conference on Natural Language and Information Systems, 2009.
[14] Gross M., Grammaire transformationnelle du français. – I – Syntaxe du verbe, Larousse, Paris, 1968.
[15] Gross M., La construction de dictionnaires électroniques, Annales des Télécommunications, vol. 44, n° 1-2: 4-19, CENT, Issy-les-Moulineaux/Lannion, 1989.
[16] Gross M., Méthodes en syntaxe, régime des constructions complétives, Hermann, Paris, 1975.
[17] Harris Z.S., A Grammar of English on Mathematical Principles, John Wiley and Sons, New York, USA, 1982.
[18] Harris Z.S., Co-occurrence and transformation in linguistic structure., Language 33, 1957, pp. 293-340.
[19] Harris Z.S., Papers in Structural and Transformational Linguistics. Reidel, Dordrecht, 1970.
[20] Harris Z.S., Transformations in Linguistic Structure. Proceedings of the American Philosophical Society 108:5, 1964, pp. 418-122.
[21] Kaufmann, E., Bernstein, A., Zumstein, R., Querix: A natural language interface to query ontologies based on clarification dialogs, in: 5th International Semantic Web Conference (ISWC 2006), Springer, 2006, pp. 980-981.
[22] Lei, Y., Uren, V., Motta E., SemSearch: a search engine for the semantic web, in: Managing Knowledge in a World of Networks, Springer Berlin / Heidelberg, 2006, pp. 238-245.
[23] Lopez, V., Motta, E., Ontology driven question answering in Aqualog, in: NLDB 2004 (9th International Conference on Applications of Natural Language to Information systems), Manchester, UK, 2004.
[24] Manning C.D. and Schütze H., Foundations of Statistical Natural Language Processing, The MIT Press Cambridge, Massachusetts, London, England, 1999.
[25] Marano F., Exploring Formal Models of Linguistic Data Structuring. Enhanced Solutions for Knowledge Management Systems Based on NLP Applications, PhD Dissertation, University of Salerno, Italy, 2012.
[26] Monteleone, M., Lessicografia e dizionari elettronici. Dagli usi linguistici alle basi di dati lessicali, Fiorentino & New Technology, Napoli, 2004.
[27] Popescu, A.M., A Etzioni, O., A Kautz, H., Towards a theory of natural language interfaces to databases, in: Proceedings of the 2003 International Conference on Intelligent User Interfaces, Miami, Association for Computational Linguistics, 2003, pp. 149-157.
[28] Postiglione A., Monteleone M., Marano F., Monti J., Napoli A., Electronic Dictionaries for Information Retrieval, Automatic Textual Analysis and Semantic-Based Data Mining Software, in: Database, Corpora e Insegnamenti Linguistici, Schena Editore – Alain Baudry et Cie, Bari-Parigi, 2012.
[29] Silberztein, M., Dictionnaires électroniques et analyse automatique de textes, Masson, Paris, 1993.
[30] Silberztein M., NooJ Manual, Available for download at: www.nooj4nlp.net, 2002.
[31] Tablan, V., Damljanovic, D., Bontcheva, K., A Natural Language Query Interface to Structured Information, in: Proceedings of the 5th European semantic web conference on The semantic web: research and applications, Springer-Verlag Berlin, Heidelberg Springer-Verlag Berlin, Heidelberg, 2008, pp. 361-375.
[32] Tesnière L., Esquisse d'une syntaxe structurale, Klincksieck, Paris, 1953.
[33] Tesnière L., Éléments de syntaxe structurale, Klincksieck, Paris, 1959.
[34] Vietri, S., Dizionari elettronici e grammatiche a stati finiti. Metodi di analisi formale della lingua italiana, Plectica, Salerno, 2008.
[35] Vietri, S., Elia, A., D'Agostino, E., Lexicon-grammar, Electronic Dictionaries and Local Grammars in Italian, in: Syntaxe, Lexique et Lexique Grammaire. Volume dédié à Maurice Gross, Laporte E., Leclere, C., Piot, M., Silbertzein, M., eds., Lingvisticae Investigationes Supplementa n. 24, John Benjamins, Amsterdam/Philadelphia, 2004, pp. 125-136.
[36] Zettlemoyer, L.S., Collins, M., Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars, in: UAI, 2005, pp. 658-666.