Concept Mapping to Learn and Innovate
Proc. of Sixth Int. Conference on Concept Mapping
Santos, Brazil, 2014

NLP-IMAP: INTEGRATED SOLUTION BASED ON QUESTION-ANSWER MODEL IN NATURAL LANGUAGE FOR AN INFERENCE MECHANISM IN CONCEPT MAPS

Wagner de A. Perin, Davidson Cury & Crediné S. de Menezes, Universidade Federal do Espírito Santo, Brasil
Email: wagnerperin@gmail.com

Abstract. Concept maps are tools for knowledge representation which can have their components used as bases for inference in question-answer systems. This paper explores the limitations of a computational solution of inference in concept maps and proposes a new architecture that enables them to process questions constructed in natural language. It describes what question-answer systems are, some techniques of Natural Language Processing and their applications, and highlights the main features of the inference mechanism investigated.
Keywords: concept map, question-answer system, natural language processing.

1 Introduction

For centuries the art of questioning lingers in the minds of many researchers and philosophers. Long discussions are carried out in order to discover what is most important: to know how to answer a question, or to know how to question?

We agree with Shank and Birnbaum (1996) when stating that "a central aspect of intelligence is the need to generate questions and answer them. No entity can learn on its own without generating the need to know". So it is in education that questions play a key role, so much so that most of the tools and pedagogical approaches adopted by teachers to check learning is based on question-answer model (tests, exercises, papers etc.). However, we limit the focus of this research to the use of concept maps as a knowledge representation tool.

Concept maps are graphical tools for organizing and representing knowledge. They are composed of concepts, usually within circles, and relations between these concepts, which are indicated by lines that connect. On such lines are the linking phrases that specify the relationship between the connected concepts (Novak & Cañas, 1998). Every triple (concept, relationship, concept) we call a proposition. We consider the propositions as the smallest unit of knowledge and believe that in defining them the author is actually making a statement of an important part of his/her knowledge.

One factor that motivates us to investigate a concept map is the fact that a concept map is a tool that gives incentive to ever deeper investigations of knowledge explored. With every concept and relation represented, new questions arise in the mind of the author of the map. To assist in the elicitation of new concepts and relations, Ribeiro et al. (2011) proposed a computational architecture that suggests concepts and relationships commonly represented on maps of the topic addressed.

Those who evaluate the semantics of a concept map research extensively to identify nuances of knowledge through concepts and relationships externalized by the author of the concept map. In doing so, they use a set of questions relating to the content of the map they consider important, such as: "What does the student know about the concept A? Did the student identify the relationship between concept A and concept B? What is the term used to link the existing relations between the concepts A and B? Which concepts did the student relate to the concept A? ".

We note that the search for answers to these questions would refer to a heavy load of cognitive processing and consume a significant amount of time. To facilitate this, Perin *et al* (2012) proposes a mechanism of inference able to answer questions raised by teachers through questions and answers. However, this mechanism is limited to people who have specific expertise or have been adequately trained. We are currently working to solve this limitation. The present study seeks to propose improvements to this mechanism in order to humanize the interaction between the appraisers and the system, by allowing queries to be constructed in natural language, adopting Natural Language Processing (NLP) techniques.

To present the proposed solution, this paper is organized into five sections, including this introductory section. The second section aims to describe what the question-answer systems are and highlights the main features of the inference mechanism investigated and also points to some techniques of Natural Language Processing and their applications in this research. The third section shows the proposed solution and its integration into a inference engine. Finally, the fourth section presents some conclusions, limitations and points to possible future works.

2 Question-answer systems

Question-answer system is a research area of computer science that encompasses techniques of information retrieval and natural language processing in order to build systems for automatic answering of questions made by humans in a natural language. Overall, its architecture includes a knowledge base where the information that serves as the basis for the search for answers (documents, pages etc.) is stored.

The first question-answer system that has been reported is the BASEBALL (Green et al., 1961), a program to answer questions about baseball tournaments played one season in the American League. This system was able to answer questions made in natural language, such as: "To whom did *Red Sox* lose on 05 July?" or "How many games *Yankees* played in July? ". The BASEBALL was able to examine the question using linguistic knowledge, in canonical form, and generated a query in a structured database about BASEBALL.

The first question-answer systems were composed essentially of a *front end* that performed the analysis, interpretation and mapping of the questions written in common terms (natural language) for more specific formats to be processed by the *back end*, usually composed of relational databases. The first forays into the database for question-answer systems were abandoned in the late 1980s for reasons that included technological limitations related to the trustworthiness of the natural language processing (Clark et al., 2010).

Moldovan *et al.* (1999) proposed a taxonomy for classifying the question-answer systems divided into 5 classes, considering 3 main features: (a) the knowledge level, (b) the level of reasoning, and (c) indexing and NLP techniques used. Later, the same author characterized these systems according to the complexity of the questions and the difficulty of extracting answers (Moldovan *et al.*, 2003). We realized that the research area of question-answer systems involves the intersection of many scientific fields including NLP, information retrieval, human-computer interaction, knowledge representation, reasoning for interpretation of questions and response analysis, algorithms for finding preferred answers, extracting from audio or video sources, among others (Maybury, 2004).

Regarding the architecture of the question-answer systems, Amorim (2012) states that these are directed to: (a) a series of questions; (b) processing a variety of sources (documents, web pages, databases etc.); (c) producing answers to users. According to her the generic architecture of question-answer systems is modular and integrated as shown in Figure 1.

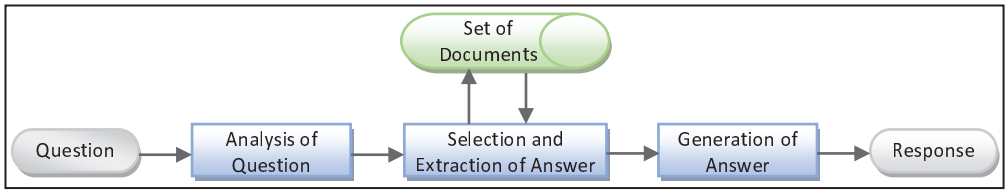

Figure 1: Basic architecture of a question-answer system. *Source: Amorim (2012)*

As shown in Figure 1, the process of a question-answer system in its classical form is: (a) Analysis of Question: The interpretation of the question asked. Building a satisfying answer to the question depends on the effectiveness of this process. Some strategies and tools are used to improve the recovery stage of the question-answer systems (Carpineto *et al.*, 2012.); (b) Selection and extraction of the answer: Retrieving the documents and extract a set of candidate answers. There are several standards used in this process such as retrieval based on relevance and retrieval based on pattern (*Clark et al.*, 2010.); and (c) Generation of answer: Process for handling text, or visual recourse, so as to make it presentable to the user, seeking to provide a better interaction with the user (Amorim, 2012).

Subsection 2.1 presents the *iMap*, the inference mechanism explored, highlighting its characteristics and limitations.

2.1 The iMap

The *iMap* (Inference from concept maps) is an intelligent tool to query the concept maps based on question-answer model. It is able to answer questions that teachers and/or specialists want to make to extract information present in a concept map, without requiring them to map all the relationships and concepts present in the map.

Since we consider each proposition in this forum as the enunciation of knowledge, the *iMap* has a knowledge base that is supplied with all the propositions presented in a concept map, where each unit of knowledge (proposition) is considered a fact. As can be seen in Figure 2, this is done by transforming the factual relationships present in the concept map into the knowledge base made in *Prolog*[1].

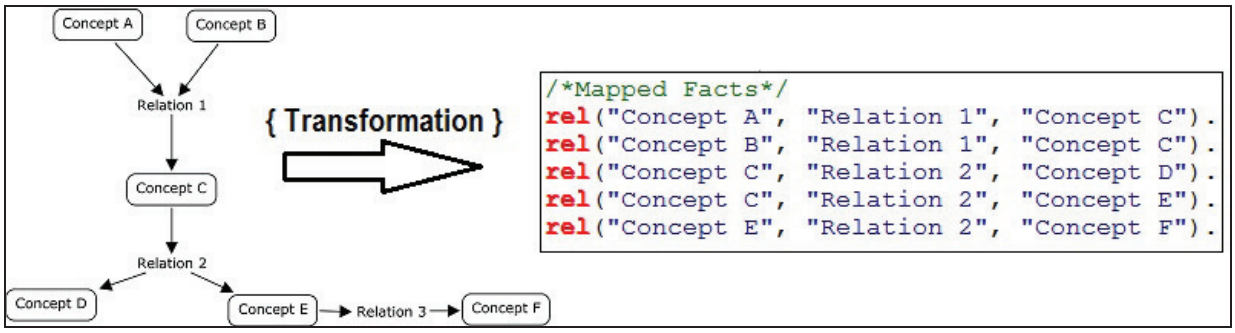

Figure 2: Transformation of propositions of a Concept Map on Facts for a Knowledge Base.

In addition to the mapped facts, the knowledge base of the *iMap* has a set of standardized rules that allow inferring answers to questions constructed by teachers and experts. Table 1 shows some examples of questions that can be made to inference engine (In Natural Language and mapped to the standard of *iMap*) and the type of response that will be presented.

Table 1: Example of questions that can be made in the *iMap*.

Question on Natural Language	Question mapped to the *iMap*	Response Type
Is there a direct relationship between concept X and concept Y?	directRelation("X", "Y").	T or F
Is there any relationship between concepts X and Y?	relation("X", "Y").	T or F
What concepts were represented in concept X?	allTarget(OUTPUT, "X").	List
Which concepts predate concept X?	allSource(OUTPUT, "X").	List
What relationships does concept X have?	whatRelations(OUTPUT, "X").	List
What relationships exist between concepts X and Y?	whatRelations(OUTPUT, "X", "Y").	List

We see great advantages in using this mechanism because it allows the teacher or expert to extract map information or even navigate their concepts without the need to step through it visually. This way they can monitor and evaluate the contents of the map without needing to spend time excessively to browse the concepts present in these maps.

The fundamental limitation that we explore in this work, however, lies precisely in the process of mapping of the questions made in natural language to the standard expected by the mechanism (see Table 1). Currently this process has no automation, which requires the teacher to write questions already in the format expected by the mechanism. We believe that this limitation reduces the field of applications of this mechanism to users who have some knowledge of Prolog or are previously trained to use it, and it presents a challenge for the new users who may be interested in using this application.

However, the translation process is not a simple task since it involves knowing, to quite some degree, in depth the two languages involved in the process. In computing environments it is even more complicated as computer scientists often are not gifted with languages and they need to perform this task producing computable solutions. Subsection 2.2 will better describe the Natural Language Processing and its challenges.

[1] ***Prolog*** is a programming language that fits the paradigm of Logic Programming and especially associated with artificial intelligence and computational linguistics.

2.2 *The Natural Language Processing (NLP)*

The Natural Language Processing (NLP) is the development of techniques and computational models for the performance of tasks that depend on information expressed in natural language (Covington, 1994; Russell, 1995). In order to occur, several of the current tasks depend functionally from the NLP, such as translation and interpretation of texts; search for information in documents; man-machine interface; etc.

According to Covington (1997), the research in NLP is directed essentially to three aspects of communication in natural language, namely: (a) sound (phonology): related to the recognition of sounds that make up words in a language; (b) structure: related to the recognition of primitive units that make up a word (morphology) and how words relate to the composition of a sentence (syntax); (c) meaning: related to the combination of a syntactic structure, the meaning of the words composing a sentence (semantic), and to the verification if this association is more appropriate in context (pragmatics).

Thus, we conclude that the NLP is an area of research that involves various disciplines of human knowledge. In this paper we propose an approach for parsing of certain types of questions, to specify a grammar and computational approach capable of processing a finite set of questions and to decide on the best translation of these questions for the universe of questions to which the *iMap* produces answers.

A grammar is a formal specification of the structure of sentences allowed by a language. The most common way to specify a grammar is to define a set of terminal symbols, the words that are permitted by the language, a set of non-terminal symbols, denoting the components of the sentences, and a set of production rules that expand the non-terminal symbols in a sequence of terminal and non-terminal symbols. Moreover, the grammar should be an initial non-terminal symbol (Rich, 1995). In computing, the technique most used for the notation of describing context-free grammars is the BNF (Backus-Naur Form). It uses a declarative syntax that allows the definition of terms of language via production rules. Each rule contains terms in which each is expanded until it reaches the terminal elements, which terms are described in literal characters.

In the context-free grammars, the left side of a rule is always a single non-terminal symbol, while the right side can contain terminal or non-terminal symbols. The context-free grammars can be used to recognize, and to check, whether a sentence belongs to the language defined by the grammar, or to generate, i.e., to construct a sentence that belongs to the language defined by the grammar.

Another NLP technique we adopted in our research is tagging. This technique is commonly used in computing to assist in the process of parsing of sentences. It involves inserting tags showing the syntactic function of each element of a sentence. There are currently several computational solutions that assist in this process, the most common being the Freeling (Atserias *et al.*, 2006) and the VISL (Bick, 2001). Because of the degree of maturity and studies that prove its practical results we use the VISL as a mechanism for tagging (Bick, 1996).

3 The *NLP-iMap*

We propose a change in the current architecture of the *iMap* inference mechanism to insert a new task to act as a translator of questions made in Natural Language into the format expected by the *Prolog* engine. Figure 3 (a) demonstrates how the questions are made in the current system architecture while Figure 3 (b) shows the insertion of the new process we are proposing and how it affects the flow of activities using this mechanism.

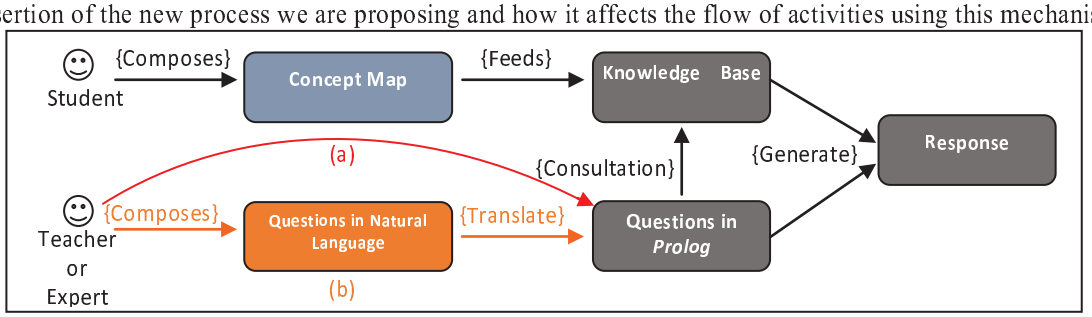

Figure 3: (a) Current (b) Proposed Conceptual Architecture for *iMap*.

3.1 *Functional Architecture of NLP-iMap*

The functional architecture proposal to the *NLP-iMap* consists of two main elements: (1) the *NLP Processor* and (2) the Solver of Correspondence; and has two functional assets: (1) the labeler VISL and (2) a correspondence table containing the intermediate grammar. This functional architecture is shown in Figure 4.

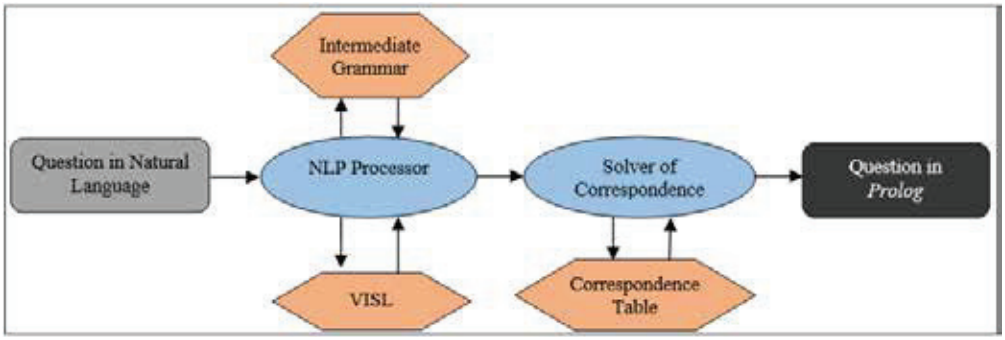

Figure 4: Functional Architecture of *NLP-iMap*.

Before defining the architecture for the *NLP-iMap*, we analyzed the components of its architecture. The first activity was to tabulate and correlate the query rules defined by *iMap* to a grammar, which we call intermediate grammar, adopted in order to reduce the semantic gap between the questions made in natural language and its corresponding translation to the standard used by *iMap*.

The semantic gap can be easily understood by people who work with translation. Often a phrase constructed in a given language has no exact correspondent in the target language. Most often, the translator builds a sentence that approximates the semantic matching of phrases in both languages. This reduces the semantic gap that distances these two languages.

As the same question can be expressed in several ways in natural language, we note that the semantic gap would be a common problem, given that the grammar of the target language is limited. This means that all questions to be constructed in natural language should be rewritten to find a match in *iMap*. Thus, when defining an intermediate grammar, the distance between the meanings of both grammars should be reduced. Table 2 shows some correspondence relations between questions in Prolog, in intermediate grammar and natural language.

Table 2: Example of questions that can be carried out to *iMap*.

Questions in *Prolog*	The corresponding *intermediate grammar*	Examples of possible matches in Natural Language
whatRelations(X, “Car”)	What relation Car	In which relationships is the car?
		Which are the relationships where the car is present?
whatRelations(O, “Car”, “Wheel”)	What relation Car Wheel	Which are the relationships that connect the car and the wheel?
		The car and the wheel are joined by which relations?
whatRelations(“have”, X, Y)	What concept have	Which pairs of concepts are connected by a relation "have"?
		"Have" is the term which bonds the connecting pairs of concepts?
howManyRelations(“Car”)	How many relation Car	How many relationships does the concept car have?
		In how many relationships is the car present?

We observed that the intermediate grammar allows the for the retrieval, from the universe of possible questions, the key elements that lead to the selection of the trigger question in the inference engine. Moreover, it allows for the definition of a context-free grammar for the language that will facilitate the work of interpretation of NLP mechanism. The BNF grammar in intermediate format can be seen in Figure 5.

```
<question> ::= <query> <type> <key>
<question> ::= what | how many | is there any
<type> ::= direct relation | relation | concept
<key> ::= <concept> | <concept> <concept> | <relation> | <relation> <concept> | <relation> <concept>
<concept>
<concept> ::= <noum>
<relation> ::= <verb>
<noum> ::= car | wheel | tire | iron | rubber | ...
<verb> ::= have | contain | elect | do | organize | ...
```

Figure 5. BNF definition of the *intermediate grammar*.

Once the intermediate grammar is defined and a parameterized table with matches in the format expected by the inference engine is constructed, it remains to specify, how the NLP processor extracts the key elements of the question in natural language to construct the corresponding question translated into intermediate grammar and how the Solver of Correspondence operates in the selection of the corresponding question in *iMap*.

3.1.1 The NLP Processor

The NLP Processor is the main element of the translation process. It is responsible for the initial translation of a question built in Natural Language for the intermediate grammar. For this, the first step of the process consists of making a web request to the VISL system requesting the tagging of the text in the original question. Figure 6 shows an output generated by the tagging process performed by VISL. In this figure we see that even if the question is constructed differently, the key elements (underlined in Figure 6) are present in the sentence subjected to both (a) and (b). These key elements are crucial as they will be used to compose the corresponding question in the intermediate grammar.

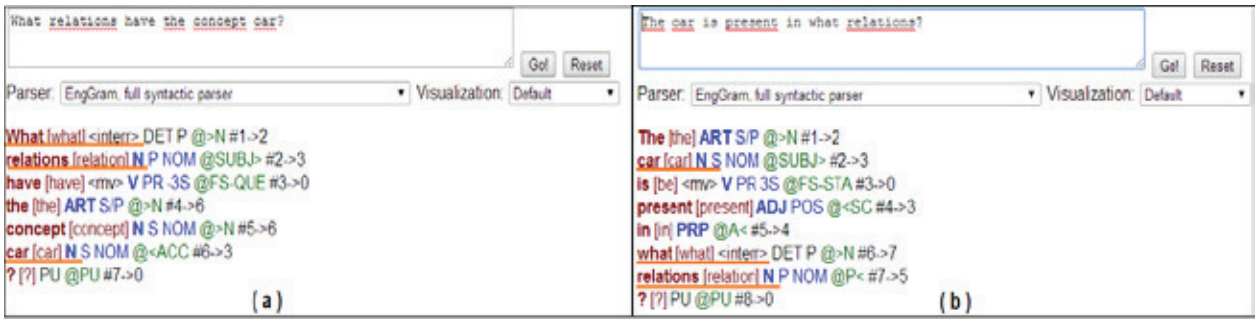

Figure 6: Output generated by VISL to the NLP Processor.

With tags of VISL, the NLP Processor initiates the construction of question in intermediate grammar, respecting their BNF. For each component of the question (see Figure 6) there is a selection parameter that takes into account the tags and the positions of the words in the sentence. Table 3 presents some parameters established for selection of each of the key elements.

Table 3: Parameters for selection elements of the intermediate grammar.

Element	Selection Parameters
Query	Tag ***<interr>***
Type	Tag **N** (noun)
	Close the delimiter of *query* (tag ***<interr>***)
	Limited to “concept” and “relation”
Key (Concept)	Tag **N**
	Do not include the delimiter of *type*
Key (Relation)	Tag **V** (Verb)
	Do not include verbs followed by adjectives (ADJ) or determinant (DET) they indicate continuity of question and not a relationship-focused.

Based on these parameters, the NLP Processor performs the building of a question in intermediate language and forwards it to the *Solver of Correspondence*.

3.1.2 The Solver of Correspondence

Each question built in intermediate grammar needs to be mapped to a corresponding question in the format expected by the inference engine. For this, the Solver of Correspondence has a table of parameterized matching, similar to that shown in the first two columns of Table 2. The Solver of Correspondence analyzes the structure of the question in intermediate grammar and extracts the required parameters for the activation of the corresponding question in the target language.

This task is performed by means of a structural analysis of the question at which a procedure that aims to extract, from a set of questions in Prolog, those which have the desired return and the number of parameters occurring in the original question (built in natural language). For this we define the following filters that are applied to the set of questions in prolog to extract only those that pass all filters:

1. *Type of question*: with this filter, we have grouped the questions by a taxonomy that defines the type of response the user wants to get. The questions may return either: concepts, relationships, logical (true or false), or statistical information. The return type is identified and the questions that do not return the desired information are removed from the set of possibilities.
2. *Type and number of parameters*: with this filter, the number of parameters and their types (concepts or relationships) identified in the original question determine which questions are candidates. Questions that do not accept these parameters are removed from the set of possibilities.
3. *Keywords:* we are defining a set of keywords that make identifying the corresponding question more accurate. In other words, this set of keywords is checked in order to ascertain which of the candidate questions respond more precisely to what the user wants.

In some cases, the question process built in natural language may: (a) have no corresponding translation in the language of *iMap* or (b) has two or more possible translations for the language of the *iMap*. In the event of (a), the Solver of Correspondence is notified, via the interface, that it is not able to answer this type of question and it suggests to the user other ways to formulate a question. With the occurrence of (b), the solver match triggers the first result. As the response generated by the engine may not be as expected, the Solver of Correspondence indicates to the user that the system has identified another way to answer that question. Thus, the user may request, at any time, the execution of other possible solutions.

3.2 The NLP-iMap today

Currently, *PLN-iMap* has a working prototype developed in Java using advanced techniques for agile development through the development paradigm oriented models. Moreover, their intelligence base is built and processed in Prolog. Figure 7 shows the operation of the prototype *PLN-iMap*. It is possible to observe the response generated by the engine to a question asked at a conceptual map where the focal question was: "What is a car?".

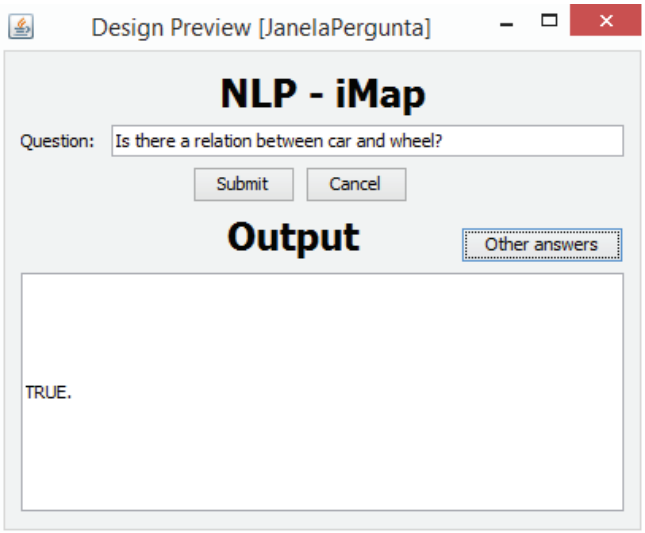

Figure 7: Prototype of *PLN-iMap*.

Current efforts, however, are being employed in order to produce a tool for final use. Thus, despite the conceptual architecture of the *PLN-iMap* and *iMap* itself will remain intact, functional architecture will suffer

some changes to suit the pattern of development known as SOA (Service Oriented Architecture). The idea is to incorporate all the features presented here to a platform construction, manipulation and management of concept maps known as *CMPaaS*[2] (Concept Map Platform as a Service), where users will have a set of solutions that will make the experience more complete when utilizing the mechanisms presented herein. In this new architecture, both *iMap* and the *PLN-iMap* are inserted as services provided by *CMPaaS* platform.

4 Final Considerations

This article aims to describe the progress of the inference mechanism in conceptual maps known as *iMap*. The goal is to become the most dynamic mechanism in processing questions built in Natural Language. Some proofs of concepts are constantly being implemented with the objective to enhance the NLP Processor and make the answers produced closer to what is expected.

As future work, we are creating a process called continuous improvement, where we are working with teachers and specialists who adopt concept maps in their work to identify a larger universe of issues that are to extract information of conceptual maps. These issues will help us to increase the base of the *iMap*, identify faults and improve the mechanisms of *NLP-iMap*. Furthermore, future versions of the tool will have embedded basic questions already available, i.e., when loading a map for investigation, the user will have, automatically, a set of answers. Moreover, he / she might suggest new questions to be incorporated into the tool. In addition, we plan to deploy an evaluation tool with which users will give feedbacks about the results of the translation processes. These feedbacks may be used in the construction of a ranking that helps to define the best possible translation for the type of question asked.

Also, we want to emphasize that all solutions explored here are being incorporated into the *CMPaaS* platform services that will extend the experience of using these tools. We look forward to be able to publicize this new platform and the results obtained from the use, in real contexts, of the mechanisms presented here.

References

Amorim, M.T., Cury, D., Menezes, C. S. Um Sistema Inteligente Baseado Em Ontologia Para Apoio Ao Esclarecimento De Dúvida. UFES – Universidade Federal do Espírito Santo, Vitória - ES, Brasil, 2012.

Atserias, J. J.; Casas, B., Comelles, E., González, M., Padró, L., Padró, M. (2006). FreeLing 1.3: Syntactic and semantic services in an open-source NLP library. In: Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006), ELRA. Genoa, Italy.

Bick, E. (1996), Automatic Parsing of Portuguese. In García, Laura Sánchez (ed.), Anais / II Encontro para o Processamento Computacional de Português Escrito e Falado. Curitiba: CEFET-PR.

Bick, E. (2001-1), The VISL System: Research and applicative aspects of IT-based learning. In: Proceedings of NoDaLiDa 2001 (Uppsala), forthcoming or internet-published.

Carpineto, C., Romano, G. A survey of automatic query expansion in information retrieval, Journal ACM Computing Surveys, Volume 44, Issue 1, New York, USA, 2012.

Clark, A., Fox, C., Lappin S. The: Handbook of computacional linguistics and natural language processing, , pp. 630-654. Wiley-Blackwell, 2010.

Covington, M. NLP for Prolog Programmers, Prentice-Hall, 1994.

Covington, M., Nute, D. and Vellino, A. Prolog Programming in Depth, Prentice-Hall, 1997.

Green, B. F., Wolf, A. K., Chomsky, C., Laughery, K. BASEBALL: An automatic question answerer. In Proceedings Western Joint Computer Conference 19, pag. 219-224, New York, USA, 1961.

Maybury, M. T. New directions in question answering, MIT Press, Stanford, USA, 2004.

Moldovan, D., Harabagiu, S.; Pasca, M., Mihalcea, R., Goodrum, R., Gîrju, R., Rus, V. LASSO: A Tool for Surfing the answer Net, In Proceeding of the Eighth Text Retrieval Conference (TREC-8), 1999.

[2] The *CMPaaS* is a service platform for construction, management and manipulation of concept maps are under development. Several solutions that facilitate the use of concept maps are being developed and incorporated into this platform in order to compose a complete set of tools to use in the Portal which is also being designed and developed by the same authors.

Novak, J. D., Cañas, A. J. The theory underlying concept maps and how to construct and use them, 1998. Disponível em: <http://cmap.ihmc.us/Publications/ResearchPapers/TheoryUnderlyingConceptMaps.pdf>.

Perin, W. A., Cury, D., Menezes, C. S. (2012). Construindo Mapas Conceituais Utilizando a abordagem iMap. In: XVII versión Congreso Internacional de Informática Educativa 2012. Santiago, Chile. Anais do TISE'2012.

Ribeiro Filho, E. L.; Tavares, O. L.; Menezes, C. S.; Cury, D. Um Estudo sobre o Incremento da Coesão e Coerência em Mapas Conceituais. Anais do 22º. SBIE. Porto Alegre/RS: SBIE, 2011. p. 1-10.

Rich, E., Knight, K. Inteligência Artificial, 2a ed., Makron Books, 1995.

Russell, S., Norvig, P. Artificial Intelligence - a modern approach, Prentice- Hall, 1995.

Schank, R., Birnbaum, L. (1996) "Aumentando a inteligência". In A natureza da inteligência, Edited by Jean Khalfa. São Paulo: Ed. UNESP, p.77-109.