A Model for Concepts Extraction and Context Identification in Knowledge Based Systems

Andre Bortolon, Hugo Cesar Hoeschl, Christianne C.S.R. Coelho, Tania Cristina D'Agostini Bueno
IJURIS – E-Gov, Juridical Intelligence and Systems Institute
Lauro Linhares, St. 728. 105. Trindade. 88036-002
Florianopolis, SC, Brazil
bortolon@ijuris.org, metajur@digesto.net, ccsrcoelho@aol.com, tania@ijuris.org
Home Page: http://www.ijuris.org

Abstract. Information Retrieval Systems normally deal with keyword-based technologies. Although those systems reach satisfactory results, they aren't able to answer more complex queries done by users, especially those directly in natural language. To do that, there are the Knowledge-Based Systems, which use ontologies to represent the knowledge embedded in texts. Currently, the construction of ontologies is based on the participation of three components: the knowledge engineer, the domain specialist, and the system analyst. This work demands time due to the various studies that should be made do determine which elements must participate of the knowledge base and how these elements are interrelated. In this way, using computational systems that, at least, accelerate this work is fundamental to create systems to the market. A model, that allows a computer directly represents the knowledge, just needing a minimal human intervention, or even no one, enlarges the range of domains a system can maintain, becoming it more efficient and user-friendly.

Keywords: Artificial Intelligence, Information Retrieval, Knowledge-Based Systems.

1 Introduction

The huge amount of documents on Internet has become a challenge to everyone that tries to find any information on any subject. [1] developed a method that allows us to estimate the size of Internet in 8.25 billions of pages on June 2003. The same work says the most known search engine, Google, has indexed only 37% of those

pages. In 2005, [2] says Google has more than 8 billion pages indexed. Assuming that Google has maintained the rate of indexed documents on the size of Internet, it is not exaggerated to affirm that Internet has around 21.6 billion of documents. These documents deal with a wide variety of subjects, giving different, or even opposite, points of view on them.

In general, Information Retrieval (IR) systems work with indexes to represent the documents. These indexes can be built with or without a controlled vocabulary. Controlled vocabularies are lists of all important words in a specific domain. However, to build a controlled vocabulary is an expensive task, demanding much time and people. Besides, the absence of a domain specialist can produce low quality vocabularies, lowering the system's performance.

Ontologies are used to extend the controlled vocabularies, allowing knowledge engineers to relate terms among them. But, considering the Internet, it is almost impossible to build ontologies that represent all domains, besides all the time that is necessary to execute the representation.

The objective of this work is to develop a computer model to automatic extraction of terms from a random set of text documents, aiming at finding concepts and identifying the context that the document belongs. Or, at least, to provide information to a domain specialist and/or a knowledge engineer to build an ontology on one or more domains, simultaneously. In this case, the time spent in the construction of the ontology can be highly reduced.

This paper presents the preliminary results of the work. Its studies aren't completed yet, needing more detailed research to finish it.

Section 2 shows the technologies that base the model, section 3 shows the proposed model, and section 4 presents the conclusions obtained up to now and future works.

2 Involved Technologies

2.1 Information Retrieval

Information Retrieval (IR) [3, 4] is the traditionally applied technique to retrieve textual documents to a specific problem. However, unlike its name suggests, IR do not retrieve the information in the sense it delivers the facts that satisfy some necessary information. According to [3], "An information retrieval system does not inform (that is, change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request."

To represent the documents, IR systems can work with either a logical or a complete representation. The former uses a keyword list, which is previously built by a human. The latter uses all words from the document. But, both have problems. The first one needs well-trained specialists to represent adequately the domain that the document belongs. The lack of keywords ends by degrading the performance, since

the system cannot answer some queries, even they are related to the domain. The second representation maintains all the words, which can retrieve documents that are not related to the context. So, it is necessary to find more refined techniques to represent the documents aiming at contextualized searches.

2.2 Knowledge Engineering

According to [5], Knowledge Engineering (KE) is the process of construction of a knowledge base. In a simplified way, a knowledge base can be understood as a set of representations of facts on the world.

The construction of a knowledge base is done through a process called knowledge acquisition, where the knowledge engineers work together with domain specialists on the study and gathering of rules and concepts that are relevant to that domain. In the knowledge representation, a domain is some part of the world on which it is desired to express some knowledge. Normally, the knowledge engineer is not a domain specialist. He/she just needs one to support the study of knowledge acquisition.

There are many processes of knowledge engineering. Here, it is focused on the process used in Dynamically Contextualized Knowledge Representation (DCKR) methodology [6]. This process is well described in [7]. This work focuses on the problems that a knowledge engineering team has to create a knowledge base if both KE team and the domain specialists are not synchronized and shows a methodology to obtain this synchronization.

Briefly, the process can be described as a sequence of steps [7]. First one is to divide the domain in subdomains, as much as necessary. After that, knowledge engineers elaborate a conceptual map of the subdomain that will be worked on. Then, they identify the domain's usual vocabulary, visualize the results and identify the relevance of these results to the domain. The last step is to insert the terms and relations in the system. [7] says, “Although the Intelligent Systems has been demonstrated their importance and maturity, their use still has some challenges for their wide dissemination and implantation. The knowledge acquisition stage still is manual and subjective.”

One alternative to accelerate the time is to build an automatic system that can extract the terms of the documents and relate them. So, the KE team would only have to examine the results of the system and approve or decline them.

2.3 UNL

The Universal Networking Language (UNL) [8] can be defined as a digital metalanguage for the description, storage and dissemination of information independently of machine or natural language. It has been developed by United Nations University (UNU) since 90s and, later, distributed to various research centers around the world. UNL works in the same way as an interlingua, that is, all the natural languages can be converted to UNL and UNL can be converted in any natural language.

UNL works with the premise that the most important information in a sentence is the concept in it. This concept is represented through Universal Words (UW) and Relations. Both UWs and Relations are universal, that is, all languages can represent them. So, the sentence "the dog runs" is represented by the UNL sentence:

agt(run(icl>do).@entry, dog(icl>animal).@def),

where "run(icl>do)" and "dog(icl>animal)" are UWs that represent the concepts "run" and "dog", respectively, and "agt" represents the relation between those UWs, indicating the second concept is the agent of the first one.

UNL's structure is based on three basic elements:

Universal Words: are the UNL vocabulary, representing a concept related to a word. Divided in two parts: headword, corresponding the concept; and, constraint list, representing the interpretation of the UW, that is, in which domain the UW is inserted. All the UWs are unified in a Knowledge Base. This Knowledge Base is hierarchical, allowing a better classification of the words.

Relations: relate two Universal Words through their syntactic behavior. They intent to describe the objectivity of the sentence. Examples: "agent", "object", "and".

Attributes: on the contrary of relations, the Attributes describe the subjectivity of the sentence. They show what is been said from the speaker's point of view.

Beside these mechanisms, UNL has some tools that are responsible for the translation process. The most important are the Enconverter and the Deconverter. The former is responsible for translation the natural language to UNL. The latter does the inverse.

Due to its organization, UNL has a great potential to represent element that can be used in both Information Retrieval and Knowledge Based Systems.

3 A Model to Automatic Extraction of Terms and Contexts

The model described here intends to identify terms automatically and join these terms trying to determine contexts. To do this task, the model follows these steps:

1) Separation of text in sentences;
2) Separation of sentences in tokens;
3) Classification of tokens;
4) Mapping the tokens;
5) Building of the relationship tree;
6) Calculation of relationship's weights;
7) Calculation of relationship's proximity value.

3.1 Separation of text in sentences

In first step, all the sentences are separated to facilitate next steps. According to Brazilian grammar, described in [9], a sentence "is any linguistic enunciation that has complete sense. It can have one or more words, with or without a verb." Examples: "Hi." "Attention!" "The house is yellow."

So, it is necessary to find all punctuation marks that indicate the end of a sentence: periods (.), exclamation points (!), and question marks (?). In some cases, colons (:) and semi-colons (;) can finish sentences, but they are not considered in this model. Finding all those punctuation marks, it is available a list of sentences.

For instance, considering the text: "The Palestinian president, Yasser Arafat, died in Paris. His body will first be carried to Cairo, Egypt, to the funeral reverences. The burial will be in Hamalah, Cisjordania." It should be divided in three sentences: (1) The Palestinian …Paris; (2) His body … reverences; and (3) The burial … Cisjordania.

The next six steps are applied sentence by sentence, since the process is very similar to syntactic analysis.

3.2 Separation of sentences in tokens

Separation in tokens is a common process in IR indexing and Natural Language Processing systems. Here, it is used in same way.

3.3 Classification of tokens

Each token is classified in four categories:

Number: every sequence of number characters, with or without separators. Ex.: "87", "1.439,26[1]";

Dates: every sequence of number characters with date separators (both "-" and "/"). Ex.: "25/04/2006";

Punctuation marks: all Brazilian Portuguese punctuation marks. Ex.: periods (.), commas (,), parentheses (());

Words: every sequence that cannot be classified in the previous ones. So, it is possible to classify all elements that appear in the text.

3.4 Mapping the tokens

Fourth step is to map each token to one of the ten morphological categories from Brazilian Portuguese grammar, such as: Nouns, Adjectives, Articles, Pronouns, Verbs, Numerals, Adverbs, Preposition, Conjunctions, and Interjections. This map is done comparing each token with a dictionary.

If a token has more than one entry in the dictionary, all the entries become candidates. For instance, the word "meio[2]" can belong to four different categories: Numeral, Noun, Adjective, and Adverb. To solve this problem, it is necessary to verify the words that are related to the ambiguous word in the sentence, establishing a context. This process is executed in the next step.

Other problem that appears in the mapping process is when the token doesn't have an entry in the dictionary. This situation occurs due to four main reasons. The first is the proper nouns. Normally, names of people, places, etc. don't appear in

[1] Brazilian number format.

[2] In English, "half".

dictionaries. There are special dictionaries to deal with people names, places, and so on, but they cannot be complete, specially related to people, since parents are very creative people when naming their children. Considering an elegant and well-done texts, proper nouns start with a capital letter. So, this can be considered as a rule to identify them in the texts.

The second reason for tokens without entries is those that correspond to words with morphologic inflections, such as, plural, verbs (in Portuguese, all persons have a specific inflection), and gender (there are different suffixes to masculine and feminine). Two approaches can be used to solve the problem: insert all inflections of the word in the dictionary, indicating the correspondence with the main word; or, build a set of rules to replace suffixes and find the main form of the word.

Tokens that correspond to words incorrectly spelled or with typing errors are the third reason. The most common causes for typing errors are:

Missing character: a missing character in any part of the word. Ex.: "snd" instead of "send";

Extra character: an extra character in any part of the word. Ex.: "sensd" instead of "send";

Wrong character: a wrong character typed. Ex.: "semd" instead of "send";

Changed character: two characters typed inversely. Ex.: "sned" instead of "send";

Absence of accent marks: the word doesn't have the accent marks. Ex.: "coleção" is typed as "colecao". This error is very common in search engine queries. Although, in regular documents, its occurrence is low.

The most common way to solve this problem is using an algorithm that generates all the words that should be the actual word, considering all the possible errors. But, this approach is questionable, since a simple five-letter word has 295 possible words (except absence of accent marks) as candidates. Other technique uses heuristics to find the most common errors (for instance, the absence of one "s" in words with "ss"). One other technique is to use probabilistic rules to determine the most similar entries to the given token.

The last reason is the tokens that aren't a word, so no entry can be related to them. For instance "abcde". In these cases, the word is mapped to a noun.

The option of mapping all the unknown tokens as nouns comes from the fact that it is almost impossible to have a complete dictionary of proper nouns. So, if a token isn't an incorrect or inflected word, very probably it is a proper noun. The documents used as a test base confirm it, since they don't have words without any meaning in their body.

Therefore, the result of the third step is a list of tokens and their corresponded grammatical category. For instance, the sentence "Peter broke the window with a rock." has the following list[3]:

Peter={(Peter; noun)}
broke={(brake; verb), (broke; adjective)}

[3] The English categories for the words come from [10]. The examples are illustrative. The model has just been tested in Brazilian Portuguese.

the={(the; article), (the; adverb), (the; preposition)}
window={(window; noun)}
with={(with; preposition)}
a={(a; noun), (a; article), (a; preposition), (a; verb)}
rock={(rock; verb), (rock; noun)}

3.5 Building of the relationship tree

The relationship between words is a process that can be done through syntactic analysis. Among all the current available technologies, the chosen one is UNL, because it can perform both syntactic and semantic analysis. But, UNL rules for Portuguese aren't completely ready, yet. So, the solution was build a structure based in UNL, but with simpler representation and mechanism.

To build this structure, a study was performed to find which are the elements in language that can be put together to form terms. It was analyzed 1,042 terms of the KMAI System's ([11]) ontology. As expected, 100% of terms have a noun as part. These nouns were mainly related to other nouns and adjectives. More than 90% of terms with more than two words have a preposition relation the two other words. The conclusion was prepositions could be used as the element of relation between two words. The words that don't have a preposition between them are related with a underscore (_).

This simpler structure was improved using the UNL attributes that indicates number, time, concept (definite or indefinite nouns and negation), and a special one necessary to Portuguese, gender.

Therefore, the relation's structure is described as:

relation(word1.attributes, word2.attributes),

where *relation* is the preposition between the words or the character "_", *word1* and *word2* are the words and *attributes* are the attributes of each word. For instance, the previous sentence "Peter broke the window with a rock.", becomes[4]:

_(Peter, brake.@past)
_(brake.@past, window.@def)
with(window.@def, rock.@indef)

3.6 Calculation of relationship's weights

To determine which relationships correspond to terms, it is necessary to calculate some weights to the relations. In this model, it is used the same weights that are used in regular IR systems, such as, term frequency (tf) and inverse document frequency (idf). Since the weights are based in the relations, they become relation frequency (rf) and relation's inverse document frequency (ridf). They are calculated by the same formula:

[4] Again, the example is illustrative. The model has just been tested in Brazilian Portuguese.

$$rf_i = \frac{n_i}{\sum_k n_k}$$

where n_i is the number of times the relation appears in the document and n_k is the number of relations of the document.

$$idf_i = \log_2\left(\frac{N}{n_i}\right)$$

where N is the number of documents and n_i is the number of documents with the relation.

3.7 Calculation of relationship's proximity value

The last step is to establish a proximity value between the relations, aiming at the creation of contexts. To do this, it is used a model based in the statistic co-occurrence of words, described in [4]. There, the similarity coefficients between two terms are based on coincidences in the term associations in the documents from the collection. The documents are represented by a matrix based in the vector-space model, where the rows are the documents' individual vectors and the columns identify the associations of terms and documents. In the model described in the paper, relations between two words are represented in the columns rather than terms. So, the matrix becomes as shown in Table 1.

Table 1. Matrix of association of terms

	R_1	R_2	…	R_k	...	R_m
D_1	rf_{11}	rf_{12}	…	rf_{1k}	…	rf_{1m}
…	…	…	…	…	…	…
D_n	rf_{n1}	rf_{n2}	…	rf_{nk}	…	rf_{nm}

The similarity between two relations can be calculated by the formula:

$$sim(REL_k, REL_l) = \frac{\sum_{i=1}^{n} rf_{ik} \cdot rf_{il}}{\sum_{i=1}^{n} rf_{ik}^2 + \sum_{i=1}^{n} rf_{il}^2 - \sum_{i=1}^{n} rf_{ik} \cdot rf_{il}}$$

where rf_{ik} indicates the frequency that relation i appears in document k and n is the number of documents in the base.

The similarity value indicates the probability that the two relations have to be related each other, since it indicates how many times one relation appeared and other also did. Doing the calculus for many relations can create cohesion between the relations, generating groups of relations that might be contexts.

Considering the matrix showed in Table 2:

Table 2. Example Matrix

	(1)	(2)	(3)	(5)	(6)	(10)
D_1	1	1	1	1	1	1
D_2	0	1	1	0	0	0
D_3	0	1	1	0	0	0
D_4	0	0	1	0	0	1
D_5	0	0	0	0	1	0

The similarities between the relations are disposed in Table 3.

Table 3. Similarities between relations

	(1)	(2)	(3)	(5)	(6)	(10)
(1)	X	0.33	0.25	1	0.5	0.5
(2)	0.33	X	0.75	0.33	0.25	0.25
(3)	0.25	0.75	X	0.25	0.2	0.5
(5)	1	0.33	0.25	X	0.5	0.5
(6)	0.5	0.25	0.2	0.5	X	0.33
(10)	0.5	0.25	0.5	0.5	0.33	X

To really find a context, it is necessary to put a minimum threshold to initiate the grouping. Tests have been done trying to determine this value, but no result was obtained yet. Basically, the test is to get one context and select a number of documents on it. So, extract the relations and calculate the similarity between that. Also, it is necessary to find other relations that should be considered in other contexts. After that, get other context related to the first and calculate the frequency of the relations from the first one appears and compare the values. Last, get a third context that does not have any relation with the first and do the same process. So, we can get some average from the three values and test in a generic set of documents.

4 Conclusions

Since the research is not finished yet, there are not so many results achieved up to now. But, the reducing of time to build a initial set of terms to analyze and build an ontology has already been evidenced. The main reason for it is that the simple structure of the model create documents that allow a fast recognizing of related words, creating a lot of candidate terms. Usage of weights also highlight the terms that happens with more frequency and which are normally related.

The model is also ready to be used in a Information Retrieval System, improving the representation and the results to the users.

Also, the model can be used in any IR system that uses UNL to structure the documents in the base, since the representation is strongly based in UNL.

References

[1] SOARES, António; BARROSO, João; BULAS-CRUZ, José. Estimativa da PIW através de Motores de Pesquisa de Grande Escala. In: Conferência IADIS Ibero-Americana WWW/Internet 2004. Madrid, 2004.

[2] ARNOLD, Stephen A. The Google Legacy. Chapter 3. Infonortics. 2005. Available at: http://www.infonortics.com/publications/google/technology.pdf.

[3] VAN RIJSBERGEN, C. J. Information Retrieval. Second Edition. Butterworths. London, 1979.

[4] SALTON, C.; MCGILL, M. Introduction to Modern Information Retrieval. McGraw-Hill, New York, 1983.

[5] RUSSELL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach. Prentice-Hall. New Jersey, 1995.

[6] HOESCHL, H. C. Sistema Olimpo: tecnologia da informação jurídica para o Conselho de Segurança da ONU. Tese de Doutorado (Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, 2001.

[7] BUENO, Tânia Cristina D' Agostini; Engenharia da Mente: Uma Metodologia de Representação do Conhecimento para a Construção de Ontologias em Sistemas Baseados em Conhecimento. Tese de Doutorado (Engenharia de Produção). Universidade Federal de Santa Catarina. Florianópolis, 2005.

[8] UCHIDA, Hiroshi; ZHU, Meiying; DELLA SENTA, Tarcisio; The UNL, A Gift for a Millennium. UNU Institute of Advanced Studies. Tokyo, 1999.

[9] FARACO, Carlos Emílio; MOURA, Francisco Marto de; Língua e Literatura. 23ª Edição. Ática. São Paulo, 1995. v. 3.

[10] Merriam-Webster Online Dictionary. http://www.m-w.com. Accessed at: 26/04/2006.

[11] KMAI. Knowledge Management with Artificial Intelligence. Software. http://www.kmai.com.br.