Procesamiento del Lenguaje Natural, Revista nº 40, marzo de 2008, pp. 43-50 recibido 30-01-08, aceptado 03-03-08

From knowledge acquisition to information retrieval*

De la adquisición del conocimiento a la recuperación de información

M. Fernández Gavilanes S. Carrera Carrera M. Vilares Ferro
Computer Science Department, University of Vigo
Campus As Lagoas s/n, 32004 Ourense, Spain
{mfgavilanes,sccarrera,vilares}@uvigo.es

Resumen: Introducimos una propuesta en recuperación de información basada en la consideración de recursos sintácticos y semánticos complejos y automáticamente generados a partir de la propia colección documental. Se describe una estrategia donde el lenguaje y el dominio de documentos son independientes del proceso.
Palabras clave: adquisición del conocimiento, análisis sintáctico,extracción de términos, recuperación de información, representación del conocimiento

Abstract: We introduce a proposal on information recovery based on the consideration of complex syntactic and semantic resources which are automatically generated from the documentary collection itself. The paper describes a strategy where the language and the domain of documents are independent of the process.
Keywords: information retrieval, knowledge acquisition, knowledge representation, parsing, term extraction

1 Introduction

Efficiency in dealing with *information retrieval* (IR) tools is related to the consideration of relevant semantic data describing terms and concepts in the specific domain considered. This kind of resources are often taken from an external and generic module (Aussenac-Gilles and Mothe, 2004), which implies that we probably lose a number of interesting properties we would be able to recover if semantic processing was directly performed on the text collection we are dealing with.

In order to solve this and produce practical understandable results, we should allow easy integration of background knowledge from possible complex document representations, fully exploiting linguistic structures. So, we could compensate for missing domain-specific knowledge, which is a significant advantage for redeploying the system when no external resources are yet available. Also, access to a concept hierarchy so generated allows information to be structured into categories, fostering its search and reuse; as well as to integrate an interesting strategy to relate languages, using it as a semantic pipeline between them (Bourigault, Aussenac-Gilles, and Charlet, 2004; Aussenac-Gilles, Condamines, and Szulman, 2002).

In the state-of-the-art, methods to automatically derive a concept hierarchy from text can be grouped into *similarity-based* approaches and *set-theoretical* ones. The first type is characterized by the use of a distance in order to compute the pairwise similarity between vectors of two words in order to decide if they can be clustered (Faure and Nédellec, ; Grefenstette, 1994). Set-theoretical ones partially order the objects according to the existing inclusion relations between their attribute sets (Petersen, 2001). Both approaches adopt a vector-space model and represent a term as a vector of attributes derived from a corpus. Typically some syntactic features are used to identify which attributes are used for this purpose.

Our proposal aims to facilitate the knowledge acquisition task through a hybrid approach that combines *natural language processing* (NLP) strategies, such as shallow parsing and semantic markers, with statistical techniques and term extraction. A modular architecture allows for the addition of textual fonts on different topics and languages, providing the basis for dealing with multilingual IR. A collection of parallel texts on the

* Work partially supported by the Spanish Government from research projects TIN2004-07246-C03-01 and HUM2007-66607-C04-02, and by the Autonomous Goverment of Galicia from projects PGIDIT05PXIC30501PN, 07SIN005206PR and the Galician Network for NLP and IR.

TERM	head	expansion
sociedad de gestión ("management society")	sociedad ("society")	de gestión ("management")
inversión directa ("direct investment")	inversión ("investment")	directa ("direct")
fondo luxemburgués ("Luxembourg fund")	fondo ("fund")	luxemburgués ("Luxembourg")
sesión de subida ("rise session")	sesión ("session")	de subida ("rise")
dólar por euro ("dollar for euro")	dólar ("dollar")	por euro ("for euro")

Table 1: Example of terms extracted

economy in French and Spanish is used as a running corpus to illustrate our proposal.

2 Knowledge acquisition

Intuitively, we are interested in strategies allowing semantic relations to emerge from text, which implies grouping relevant terms in classes according to their similarity and establishing semantic links between them.

We approach this task from two different points of view. The former is a classic term-based strategy, that only takes into account lexical data. For the second, we incorporate explicit semantic hypotheses. In both cases, our framework is based on two general principles: the *distributional semantic* model (Harris, 1968) establishing that words whose meaning is close often appear in similar syntactic contexts, and the assumption that terms shared by these contexts are usually nouns and adjectives (Bouaud et al., 1995).

As a general purpose, our work has an experimental interest as a testing frame for comparing different knowledge acquisition strategies, but also considering about the possibility of complementing capabilities. In effect, as we shall see, a term-based approach allows the acquisition task to be performed automatically. Although the results so obtained cannot compare with the quality of the semi-automatic dependency-based proposal introduced later, it could serve as a starting point for this function, generating the initial set of semantic classes we need to initialize an iterative process in order to establish more complex relationships.

2.1 A term-based approach

Our starting point here is the information provided by a classic term extractor running on a tagging environment. No particular architecture has been considered at this point.

Once the extractor has provided all the base terms and, possibly, associated their syntactic and/or morpho-syntactic variations; we can differentiate between the *head* and the *expansion* of each term, often a nominal syntagm. The former is the kernel of the syntagm, usually a noun, around which we assume the meaning of the term is structured. The *expansion* is the complement of the *head*, modifying it and defining the context where it appears.

This set of identified heads provides a local look around the meaning of the text, focused on the syntagms recognized as terms. In order to extend these primary semantic links to the full text, we apply a simple recursive process by generating a hash table whose entries we baptize as *main elements*. Mains elements are all heads whose pos-tag is a noun. The key of each entry is a main element, to which we associate the list of contexts where it appears either as an *expansion* or as an *head*. As a result, we obtain a simple graph structure capturing the essential meaning of the text, as seen in Table 1.

The next step consists of grouping terms in semantic classes, filtering out non-relevant features. To deal with in practice, we go through the hash table generated, comparing different contexts by applying as a similarity[1] measure the DICE coefficient (Bourigault and Lame, 2002):

$$\text{DICE}(C_1, C_2) = \frac{|C_1 \cap C_2|}{(|C_1| + |C_2|)/2}$$

where C_1 and C_2 are contexts, and $|C_i|$ represents the cardinal of C_i, $i = 1, 2$. Intuitively, we are computing the common terms between C_1 and C_2, and then applying normalization.

At this point, the generation of classes is an iterative process. In each iteration we join

[1] we can define a similarity between entities as the number of common properties shared by them.

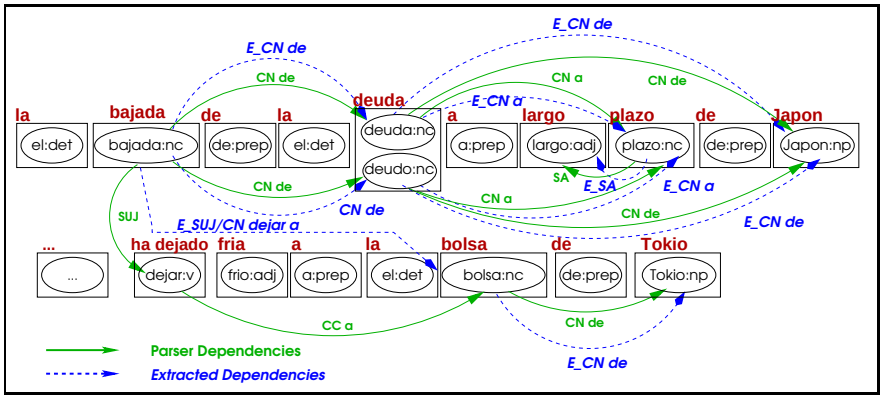

Figure 1: Graph of dependencies from a parse

the pair of main elements whose DICE value turns out to be the highest computed from the hash. So, in each step the hash table is reduced in an element and the process finishes when only DICE coefficients equal to zero can be computed. In other words, when no more context sharing is possible.

Once the iteration loop stops, entries in the hash are words semantically related together with their associated unified contexts. This hash outcome is stored in an XML[2] file, in such a way that similar elements are grouped representing a new and previously undefined semantic class. This file is later converted to an OWL[3] (Szulman and Biébow, 2004) format, in order to facilitate ulterior retrieval tasks.

2.2 A dependency-based approach

We start now from a robust parse based on a cascade of finite automata (Vilares, Alonso, and Vilares, 2004). So, we can identify relevant terms in nominal and verbal phrases, namely, those nouns and verbs relaying essential semantic information, as well as local relationships between them. As result, we obtain a graph of dependencies of the type *governor/governed*, as is shown in Fig. 1 by using dotted lines going from the governor term to the governed one.

2.2.1 Filtering out dependencies

Once these primary syntactic dependencies have been established, possibly including a number of lexical and syntactic ambiguities generating useless dependencies, we try to effectively extract the latent semantics in the document. The idea consists of compiling additional information from the corpus in order to detect and delete these useless structures.

We first introduce, from the sentence *"la bajada de la deuda a largo plazo de Japón ... ha dejado fria a la bolsa de Tokio"* in Fig. 1, some simple notations to describe parses. So, rectangular shapes, called *clusters*, show positions in the input string. Lemmas with their corresponding lexical categories are represented by ellipses baptized as *nodes*. Green arcs represent binary dependencies between words through some syntactic construction.

The parsing frame provides the mechanisms to deal with a posterior semantic phase of analysis, by avoiding the elimination of syntactic data until we are sure it is unnecessary for knowledge acquisition. So, the lexical ambiguity illustrated in Fig. 1 should be decided in favor of the first alternative[4], because we have the intuitive certainty that the word *"deuda"* is related to "debt" and not to "relative". Given that we are dealing with a specialized corpus, we should confirm this by exploring the corpus in depth. That is, in order to solve the ambiguity we only need the information we are looking for, which leads us to consider an iterative learning process to attain our goal.

In particular, we are more interested in dependencies between nouns and adjectives. This justifies filtering those dependencies, as shown in Fig. 1, following the dotted lines. So, the word *"plazo"* ("term") is connected to *"largo"* ("long"), the latter being an adjective. Furthermore, we are also interested in extracting dependencies between nouns through, for example, prepositions such as *"bolsa de Tokio"* ("Tokyo Stock Exchange") and through verbs such as *"bajada dejar_a*

[2] see http://www.w3.org/XML/
[3] see http://www.w3.org/TR/owl-features/

[4] which corresponds to "The long-term debt descent of Japan has left cold to the Stock Exchange of Tokyo".

$$1.\quad P(\text{deuda:uc:money}, [\text{\_CNde}], \text{Japón:up:country})_{\text{local(0)}} = \frac{\begin{array}{c}P(\text{deuda:uc}, [\text{\_CNde}], \text{Japón:up})_{\text{local(0)}} \\ P(\text{deuda:uc:money})_{\text{local(0)}} \\ P(\text{Japón:up:country})_{\text{local(0)}}\end{array}}{\Sigma_{X,Y}\, P(\text{deuda:uc:X})_{\text{local(0)}}\; P(\text{Japón:up:Y})_{\text{local(0)}}}$$

$$2.1\quad P(\text{deuda:uc:money}, [\text{\_CNde}], X)_{\text{global(n+1)}} = \frac{\Sigma_X P(\text{deuda:uc:money}, [\text{\_CNde}], X)_{\text{local(n)}}}{\#\text{dep}_{\text{local(n)}}(\text{deuda})}$$

$$2.\quad 2.2\quad P(Y, [\text{\_CNde}], \text{Japón:up:country})_{\text{global(n+1)}} = \frac{\Sigma_Y P(Y, [\text{\_CNde}], \text{Japón:up:country})_{\text{local(n)}}}{\#\text{dep}_{\text{local(n)}}(\text{Japón})}$$

$$2.3\quad P(\text{deuda:uc:money}, [\text{\_CNde}], \text{Japón:up:country})_{\text{global(n+1)}} = \begin{array}{l}P(\text{deuda:uc:money}, [\text{\_CNde}], X)_{\text{global(n+1)}} \\ P(Y, [\text{\_CNde}], \text{Japón:up:country})_{\text{global(n+1)}}\end{array}$$

$$3.\quad P(\text{deuda:uc:money}, [\text{\_CNde}], \text{Japón:up:country})_{\text{local(n+1)}} = \frac{\begin{array}{c}P(\text{deuda:uc:money}, [\text{\_CNde}], \text{Japón:up:country})_{\text{local(n)}} \\ P(\text{deuda:uc:money}, [\text{\_CNde}], \text{Japón:up:country})_{\text{global(n+1)}}\end{array}}{\Sigma_{X,Y}\quad \begin{array}{l}P(\text{deuda:uc:X}, [\text{\_CNde}], \text{Japón:up:Y})_{\text{local(n)}} \\ P(\text{deuda:uc:X}, [\text{\_CNde}], \text{Japón:up:Y})_{\text{global(n+1)}}\end{array}}$$

Table 2: Extraction of classes for *"deuda de Japón"*

bolsa" ("descent leave Stock Exchange").

In order to identify the most pertinent dependencies, and also using dotted lines, we focus on detecting and later eliminating those dependencies that are found to be less probable in sentences, since they include terms with a low frequency. Nodes and arcs in the resulting graph are baptized as *pivot terms* and *strong dependencies*, as is shown in Fig. 1.

A supplementary simplification phase consists of applying a simple syntactic constraint establishing that a governed word can only have one governor. So, for example, and indicated with a simple line in the sentence of Fig. 1, *"Japón"* ("Japan") is governed by *"deuda"* ("debt"), but also by *"deuda"* ("relative") and, in consequence, we should eliminate one of these dependencies. No other topological restrictions are considered and, in consequence, a governor word can have more than one governed one, as in the second interpretation of Fig. 1 ("long-term debt descent of Japan"), where *"bajada"* ("descent") is the governor for *"plazo"* ("term") and *"Japón"* ("Japan"), also indicated with a simple line. The same word could be governor and governed at the same time, this being the case of *"plazo"* ("term"), which is the governor for *"largo"* ("long"), but is also governed by *"deuda"* ("debt") in the first interpretation.

2.2.2 Term clustering

The generation of semantic classes is inspired by an error-mining proposal originally designed to identify missing and erroneous information in parsing systems (Sagot and Villemonte de La Clergerie, 2006). This technique combines two complementary iterative processes. For a given iteration, the first one computes, for each governor/governed pair in a sentence, the probability of the corresponding dependency; taking as its starting point the statistical data provided by the original error-mining strategy and related to the lexical category of the pivot terms. The second process computes, from the former, the most probable semantic class to be assigned to terms involved in the dependency. So, in each iteration we look for both semantic and syntactic disambiguation, each profiting from the other. A fixed point assures the convergence of the strategy (Sagot and Villemonte de La Clergerie, 2006).

We illustrate term clustering on our running example in Fig. 1, focusing on the dependency labeled [_CNde] relating *"deuda"* ("debt") and *"Japón"* ("Japan"). We do so by introducing both iterative processes in this particular case, talking without distinction about weight, probability or preference to refer the same statistical concept. So, from Table 2, we have that:

1. To begin with, we compute the local probability of the dependency in each sentence, which depends on the weight of each word, this in turn depending on the word having the correct lexical category. To start the process, first category assumptions, denoted by $\mathsf{P_c}$, are provided by the error-mining algorithm (Sagot and Villemonte de La Clergerie, 2006). We also take into account the initial

probability for the dependency considered, $P_{dep\ ini}$, a simple ratio on all possible dependencies involving the lexical categories concerned. The normalization is given by the preferences for the possible lexical categories involving each of the terms considered and is here represented by variables X and Y.

2. We reintroduce the local probabilities into the whole corpus locally in the sentences, in order to re-compute the weights of all possible dependencies,after which we then estimate globally the most probable ones. The normalization is given by the number of dependencies connecting the terms considered, #dep.

3. The local value in the new iteration should take into account both the global preferences and the local injection of these preferences in the sentences, reinforcing the local probabilities. The normalization is given by previous local and global weights for the dependency involving all possible lexical categories associated to each of the terms considered, and is here represented by variables X and Y.

In dealing with semantic class assignment, the sequence of steps is shown in Table 2 illustrating the computation of the probability that *"deuda"* ("debt") refers to the group of money and *"Japón"*("Japan") refers to a country, taking again the dependency labeled [_CNde] in Fig. 1, both money and country classes having been defined prior to the launch of the process in a list of semantic classes:

1. In each sentence, we compute the local probability of this dependency if *"deuda"* ("debt") and *"Japón"* ("Japan") are referring to money and a country. We start from the local weight previously computed in Table 2, and also the initial preferences of the terms involved corresponding to the classes considered[5]. The normalization is given by the probabilities for the possible classes involving each one of the terms considered, without specifying any particular class and is here represented by variables X and Y.

2. We then calculate this preference at global level, by re-introducing it to the whole corpus locally in the sentences in order to re-compute the weights of all the possible classes in the sentence. In order to obtain this, we first compute the probability in the whole corpus (2.1 and 2.2) for each term and semantic class, disregarding the right and left context, represented by variables X and Y respectively. The final probability (2.3) is a combination of the two previous ones.

3. After each iteration, we re-inject the previous global weight to obtain a new local one, by reinforcing the local probabilities. The normalization is done by the addition of the preferences corresponding to the terms and classes involved in the dependency, for all the possible semantic classes considered.

After applying these last two approaches, a hierarchy can be built according to the different elements obtained in all classes.

3 Information retrieval

Work in the field of IR increasingly aims to improve text indexing or query formulation with the help of different kinds of knowledge structures such as hierarchies or ontologies. These structures are expected to bring different targeted gains (Masolo, 2001) for example improving recall and precision or helping users to express their needs more easily.

3.1 A general approach

Generally, users have no precise idea of what they can find in a document collection, and the consideration of a hierarchical structure as a guideline to describe and organize contents could simply facilitate the two essential IR tasks, information indexing and retrieval. We propose an approach where hierarchies, built up from the semantic relations emerging from text, are used in a more unusual and promising way in combination with visualization tools for guided exploration of the information space.

In dealing with IR, concept hierarchies and documents can be related in a simple way through the indexing task, by associating

[5] this is fixed by the user if the term is in a list associated to that class. Otherwise, this probability is obtained as a ratio of the total number of classes considered.

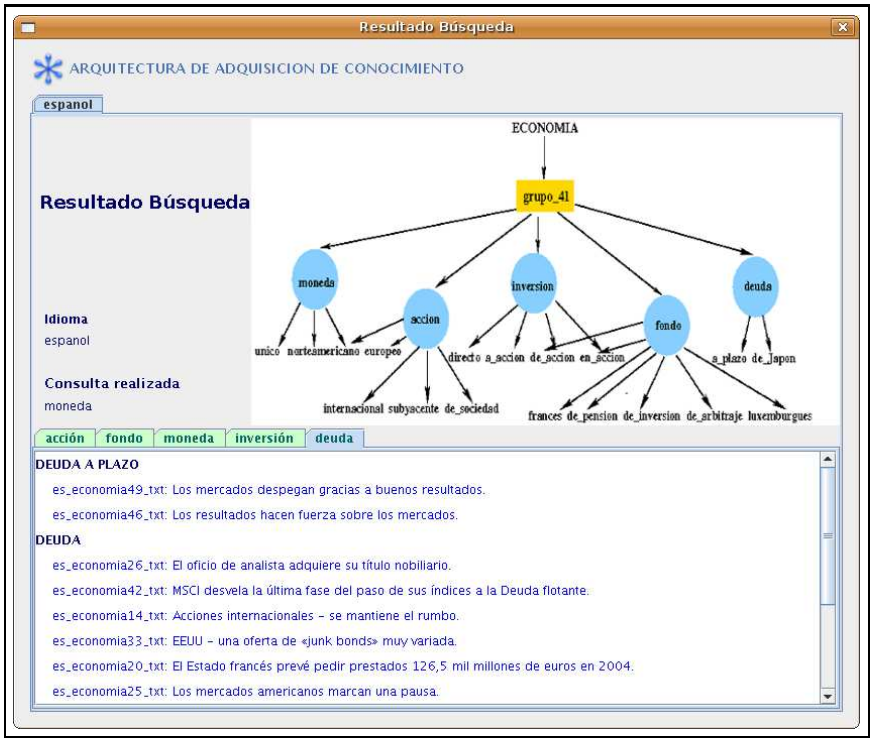

Figure 2: Sub-hierarchy for the query *"acción"* ("share") using a term-based strategy

each document to those concepts matching its content. So, in our running corpus the hierarchy is structured according to classes such as money or dates; and is automatically connected to documents after projection of the terms where they occur. We also consider a graphical interface to show these structures to the user, as is shown in Figs. 2 and 3 for our running example.

3.2 A practical approach

In practice, a major factor impacting the consideration of such an approach is the knowledge acquisition process itself. We have described two different techniques, a term-based approach and a dependency-based one, which we have integrated in a single prototype in order to define a common testing frame allowing us to effectively compare them. Although the tool can combine several domains of knowledge on a variety of different languages, we are going to focus on our running corpus by using LUCENE[6] as a standard search text engine. That is, the system identifies at parsing time the set of indexes to be considered for the effective retrieval task, using LUCENE. Once we have located them, an *expansion phase* enlarges identification of relevant terms from the conceptual structure, which will be later sent to the search engine.

In order to facilitate understanding, we illustrate the proposal through queries limited to single words. In dealing with general query sentences, these are firstly parsed to locate possible AND/OR-like operators and, in this case, we transfer them to LUCENE which can perform directly this kind of queries. In other cases, we first eliminate stop-words to later look for physical proximity and order criteria between words and, finally, re-send the query to the search engine, also after expansion.

Independently of the approach considered to generate the conceptual hierarchy, once a single-word query is introduced, we locate the corresponding class in the knowledge hierarchy we are dealing with. From this, we can identify the set of related classes, which also allows us to introduce a simple relevance criterion for the answers so obtained, based on the distance from the initial one. So, given that indexing was previously performed and system retrieves terms from these classes related to the query, we recover all the documents associated to them.

At this point, the choice of strategy impacts both the type and number of the semantic relations involved in the process described. In order to illustrate this, we study the answer given by the system for the query

[6] see http://lucene.apache.org/

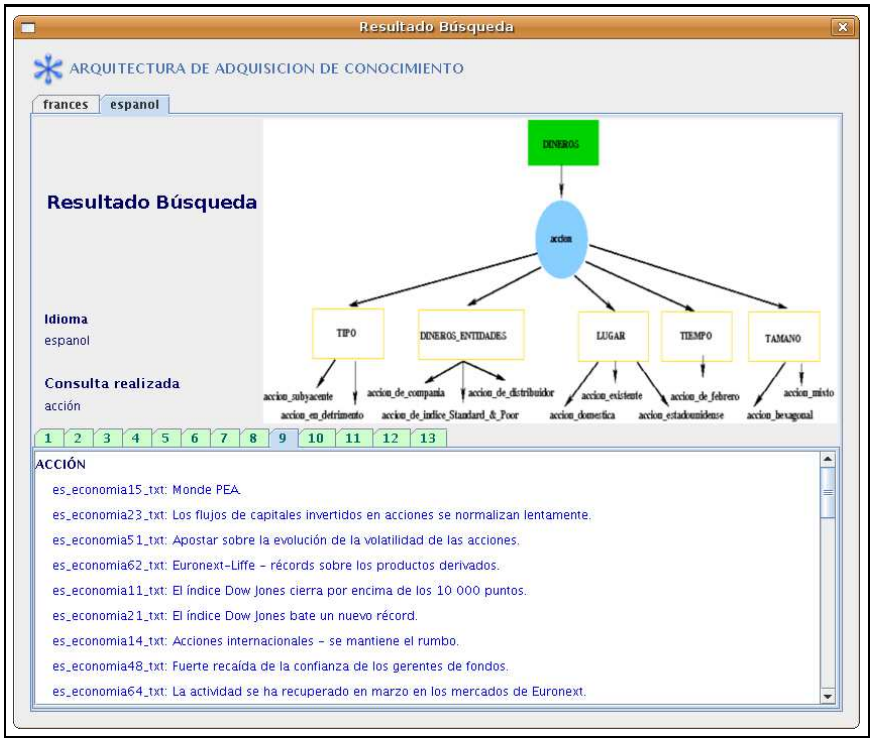

Figure 3: Sub-hierarchy for the query *"acción"* ("share") using a dependency-based strategy

"acción" ("share") first using the term-based strategy and then the dependency-based one. Focusing on the term-based approach, Fig. 2 shows the sub-hierarchy for the query, from which the system will search for the answers. The strategy groups in a class[7], the words[8] *"moneda"* ("currency"), *"deuda"* ("debt"), *"acción"* ("share"), *"fondo"* ("fund") and *"inversión"* ("investment") due to their similarities are considered high enough. Round blue shapes are heads whose expansions are indicated by arrows as in *"deuda de Japón"* ("Japan debt"), where the head *"deuda"* ("debt") points to *"de Japón"* ("of Japan"). The new class, baptized as "grupo_41", shows the way to identify the answers included at the bottom, with the documents classified according to the information retrieved and organized by their relevance and in different tabs related to the word.

Applying now the dependency-based strategy, Fig. 3 shows the sub-hierarchy considered for retrieval purposes. Classes are already defined and separated in domain concepts such as *"dineros"* ("money"), *"entidades"* ("entities") or *"paises"* ("countries"); whilst properties are similarly treated as concept features such as *"tipo"* ("type"), *"tiempo"* ("time") or *"tamaño"* ("size"). The hierarchy represents the organization of the relations between the concepts and their features. Here, the governors are represented by round blue shapes, as the word *"acción"* ("share") which is pointed to by the concept[9] *"dineros"* ("money") and is related, for example, to that of *"entidades"* ("entities"). Also, some of the properties that are related to it are *"subyacente"* ("underlying") which is a *"tipo"* ("type") property, and *"de febrero"* ("of February"), which is a *"tiempo"* ("time") one.

A particular case occurs when the governor and governed words are both concepts in a extracted parse dependency. We then represent these in the same rectangular shape using a tag governor_governed. So, in the case of *"acción de Standard and Poor's"* ("Standard and Poor's share"), it is associated to *"dineros_entidades"* ("money_entities") the governor being *"dineros"* ("money") and the governed *"entidades"* ("entities"). In this way, the query word *"acción"* ("share") is a *"dineros"* ("money") concept which is related to *"Standard and Poor's"*, which is an *"entidades"* ("entities") one by means of arrows.

If the governor is a concept and the governed is a property, only the property is rep-

[7] here represented by a rectangular yellow shape.
[8] here represented by round blue shapes.
[9] here represented in a rectangular green shape.

resented in the rectangular shape without indicating the class of the concept. In this case, the query word *"acción"* ("share") is related with different kinds of properties, such as *"de febrero"* ("of February"), which is a *"tiempo"* ("time") property; and *"subyacente"* ("underlying"), which is a *"tipo"* ("type") one.

4 Conclusion

We have introduced an IR strategy based on intelligent indexing that benefits from semantic relations between concepts in the text collection. In contrast with previous works, we generate dynamically the conceptual structure serving as a basis for the IR module, which would appear to be a promising approach exploring new knowledge domains as well as providing the user with a more flexible technique.

Although the primary purpose of this kind of hierarchies is not to classify documents, but rather to order global concepts, linking them through linguistic expressions, deductions can nevertheless be made on the texts and index creation facilitates. This factor is important because it eliminates the human factor in decision-making, this also being reflected in the ability to specify the queries launched. In effect, it is possible from these structures to infer correlation between notions present in the source text. This fact is crucial for the refinement of queries that will allow mistakes introduced by classical search engines, such as polysemy or synonymy, to be avoided.

References

Aussenac-Gilles, Nathalie, Anne Condamines, and Sylvie Szulman. 2002. Prise en compte de l'application dans la constitution de produits terminologiques. In *2e Assises Nationales du GDR I3, Nancy (F).*

Aussenac-Gilles, Nathalie and Josiane Mothe. 2004. Ontologies as background knowledge to explore document collections. In *RIAO 2004 , Avignon.*

Bouaud, J., B. Bachimont, J. Charlet, and P. Zweigenbaum. 1995. Methodological principles for structuring an ontology.

Bourigault, D. and G. Lame. 2002. Analyse distibutionnelle et structuration de terminologie, application à la construction d'une ontologie documentaire de droit. In *TAL: Traitement automatique des langues*, pages 129–150, vol 43, n 1, Paris, France. Hermès.

Bourigault, Didier, Nathalie Aussenac-Gilles, and Jean Charlet. 2004. Construction de ressources terminologiques ou ontologiques à partir de textes : un cadre unificateur pour trois études de cas. *Revue d'Intelligence Artificielle (RIA), Numéro spécial sur les techniques informatiques de structuration de terminologies, M. Slodzian (Ed.)*, 18(1/2004):87–110.

Faure, D. and C. Nédellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In Paola Velardi, editor, *LREC workshop on Adapting lexical and corpus ressources to sublanguages and applications*, pages 5–12.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Publishers, Norwell, MA, USA.

Harris, Z.S. 1968. *Mathematical Structures of Languages.* J. Wiley & Sons, USA.

Masolo, C. 2001. Ontology driven information retrieval. report of the ikf (information and knowledge fusion). eureka project e!2235.

Petersen, Wiebke. 2001. A set-theoretical approach for the induction of inheritance hierarchies. *Electr. Notes Theor. Comput. Sci.*, 53.

Sagot, B. and É. Villemonte de La Clergerie. 2006. Error mining in parsing results. In *Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 329–336, Australia.

Szulman, S. and B Biébow. 2004. Owl et terminae. In *IC: Journées Francophones dÌngénieurie des connaissances*, pages 41–52.

Vilares, J., M.A. Alonso, and M. Vilares. 2004. Morphological and syntactic processing for text retrieval. *Lecture Notes in Computer Science*, 3180:371–380.