RECUPERAR INFORMACIÓN CON CONOCIMIENTO

Sara Carrera Carrera[1] **Milagros Fernández Gavilanes**[2] **Manuel Vilares Ferro**[1]

[1] Departamento de Informática, Universidad de Vigo
Campus As Lagoas s/n, 32004 Ourense, {sccarrera,vilares}@uvigo.es
[2] Departamento de Computación, Universidad de A Coruña
Campus de Elviña s/n, 15071 A Coruña, {mfgavilanes@udc.es}

Resumen

Introducimos una propuesta en recuperación de información basada en la consideración de recursos sintácticos y semánticos generados automáticamente a partir de la propia colección documental. Efectuamos un estudio comparativo entre dos estrategias que pueden actúar de forma complementaria. En la primera se clasifican los conceptos relevantes en el dominio a partir de una extracción de términos. En la segunda analizamos las dependencias sintácticas existentes en la colección documental, de manera a obtener una asociación de los elementos que la conforman.

Palabras Clave: Adquisición del conocimiento, procesamiento del lenguaje natural, recuperación de información.

1. INTRODUCCIÓN

El contínuo incremento de textos disponibles en formato digital hace necesaria la creación de herramientas que capturen el conocimiento, de forma a poder manipularlo y explotarlo. Nos estamos refiriendo a la necesidad de automatizar la adquisición del conocimiento, para mejorar la calidad de los sistemas de *recuperación de información* (RI), suministrándoles estructuras conceptuales que reflejen el contenido de un documento.

La eficiencia de las herramientas en este campo se vincula con la observación de los datos semánticos relevantes descritos en los términos y conceptos del dominio considerado. Este tipo de recursos se suele importar de módulos externos y genéricos [1]. Consecuencia de esto es la inoperancia en el tratamiento de dominios aún sin explotar, la dificultad de mantenimiento y la imposibilidad de abordar el procesamiento de colecciones dinámicas como son las ligadas a Internet. Para solucionar esta conyuntura y producir resultados prácticos y comprensibles, debemos permitir la integración del conocimiento subyacente en los documentos; intentando aprovechar las estructuras lingüísticas que le sirven de soporte.

El estado del arte actual distingue dos corrientes para plantear el problema descrito. Se pueden agrupar en acercamientos basados en *similaridad* y aproximaciones basadas en *conjuntos teóricos*. El primer tipo utiliza una medida de distancia para calcular la similaridad entre pares de vectores de palabras [3]. En la segunda aproximación, los conjuntos teóricos ordenan de manera parcial los objetos de acuerdo a la posible existencia de relaciones de inclusión entre sus atributos [6]. Ambas técnicas adoptan el modelo de espacio vectorial y representan un término como un vector de atributos, identificados mediante características sintácticas.

Nuestra propuesta se centra en facilitar la tarea de adquisición de conocimiento a través de una aproximación híbrida. Combinamos técnicas propias del *procesamiento del lenguaje natural* (PLN) tales como el análisis sintáctico superficial y los marcadores semánticos por una parte, con técnicas de aprendizaje basadas en métodos estadísticos. Aunque es posible imaginar que el ámbito ideal de aplicación de una herramienta de este tipo sería el representado por Internet, haciendo que la tarea de RI fuese más precisa y completa, nuestra propuesta se expondrá sobre un sencillo corpus paralelo de economía en francés y español con el objeto de facilitar su descripción.

2. EXTRAER EL CONOCIMIENTO

De manera intuitiva, nuestro interés se centra en recopilar las relaciones semánticas que surjan del texto. Esto conlleva la organización de los términos en clases de acuerdo a su similaridad. Hay que destacar el propósito experimental de nuestro trabajo, que no es otro que el de establecer un marco de evaluación de dis-

tintas técnicas de adquisición de conocimiento, una de ellas basada en información léxica y la otra en información sintáctica. Se considera, además, la posibilidad de complementar ambas capacidades. Generando a partir de la extracción de términos un conjunto de clases semánticas que se utilizarían para inicializar el proceso de clasificación por dependencias. En cualquier caso, nuestro marco de trabajo se apoya en el modelo de distribución semántica de Harris [4] que establece que aquellas palabras que aparecen en los mismos contextos son semánticamente próximas.

2.1. ORDENACIÓN DE TÉRMINOS

En esta sección se describe la puesta en marcha de un prototipo rápido de clasificación de terminología. Partimos de un extractor de términos de propósito general. El corpus de entrada para esta tarea será analizado con un etiquetador léxico que asigna a cada palabra una categoría y sus posibles lemas. El extractor de términos nos proporciona el conjunto de términos base, junto con sus variantes sintácticas o morfo/sintácticas, combinando técnicas lingüísticas y estadísticas.

En un primer momento recogemos todos aquellos términos cuya frecuencia de aparición sea mayor a un umbral dado, estos datos se guardan en un fichero XML[1]. En este punto es necesario definir una restricción sobre los términos, así consideramos que cada término está formado por una *cabeza* y una *expansión*. La cabeza es el núcleo alrededor del cual asumimos que se estructura el significado del término mientras que la expansión es su modificador, tal como se ilustra en el Cuadro 1. Consideraremos para el proceso sólo aquellas *cabezas relevantes*, esto es, cabezas que bien sean sustantivos, bien nombres propios o bien acrónimos. Estimamos tales categorías léxicas como las que aportan mayor información semántica, sobre un enfoque en régimen nominal del problema.

La idea parte de utilizar los contextos en los cuales se usan los términos, y aproximar aquellos cuyo contexto es similar. Así, dado un elemento y un término cualquiera donde aparece ese elemento, su contexto será bien la cabeza bien la expansión que lo acompaña. El proceso de clasificación se basa en un algoritmo iterativo que mide la similaridad entre los contextos de aparicición de las cabezas relevantes.

Cuadro 1: Ejemplo de la extracción de términos

Término	**cabeza**	**expansión**
sociedad de gestión	sociedad	de gestión
fondo luxemburgués	fondo	luxemburgués
sesión de subida	sesión	de subida

Generamos una tabla de colisión cuyas entradas son las cabezas relevantes que llamaremos *elementos principales* y el valor de cada entrada es una lista de los contextos de aparición del elemento principal correspondiente. Partimos de una estructura global que captura el significado esencial del texto. El siguiente paso consiste en agrupar los términos en clases semánticas. Para ello, recorremos la tabla de colisión comparando los distintos contextos aplicando como medida de similaridad[2], el coeficiente DICE, definido como sigue:

$$\text{DICE}(C_1, C_2) = \frac{|C_1 \cap C_2|}{(|C_1| + |C_2|)/2} \qquad (1)$$

donde C_1 y C_2 son los contextos de dos términos y $|C_i|$ representa el cardinal de $C_i,\ i = 1, 2$. Intuitivamente, calculamos los contextos comunes entre los términos y aplicamos normalización. En cada paso del procedimento, se unen pares de elementos principales cuyo valor para el coeficiente DICE es el mayor en esa iteración. De este modo, la tabla de colisión se reduce en un elemento en cada iteración. El proceso finaliza cuando ningún contexto es compartido y el coeficiente es cero en todos los casos.

Una vez acaba el algoritmo, las entradas en la tabla de colisión son palabras relacionadas semánticamente junto con los contextos que comparten. Su lectura final produce una salida que será almacenada en un fichero OWL[3].

2.2. ORDENACIÓN DE DEPENDENCIAS

Partimos de un análisis sintáctico robusto basado en una cascada de autómatas finitos [8]. Identificamos términos relevantes en frases nominales y verbales, es decir, aquellos sustantivos y verbos que aportan la información semántica esencial junto con las relaciones entre ellos. Como resultado obtenemos un grafo de dependencias del tipo *gobernante/gobernado*, como se muestra en la Figura 1.

2.2.1. Depurando las dependencias

Los analizadores sintácticos trabajan con datos léxicos que aportan la información sobre las categorías gramaticales y los lemas de las palabras. En nuestro caso, el analizador sintáctico nos devuelve un conjunto de dependencias entre pares de palabras. Sin embargo, tanto a nivel léxico como a nivel sintáctico pueden surgir una serie de ambigüedades que es necesario solventar para extraer la semántica latente en los documentos. La idea consiste en recopilar información del corpus para detectar y eliminar este tipo de estructuras.

[1] ver http://www.w3.org/XML/

[2] definimos la similaridad entre entidades como el número de propiedades comunes compartidas por ellas.

[3] ver http://www.w3.org/TR/owl-features/

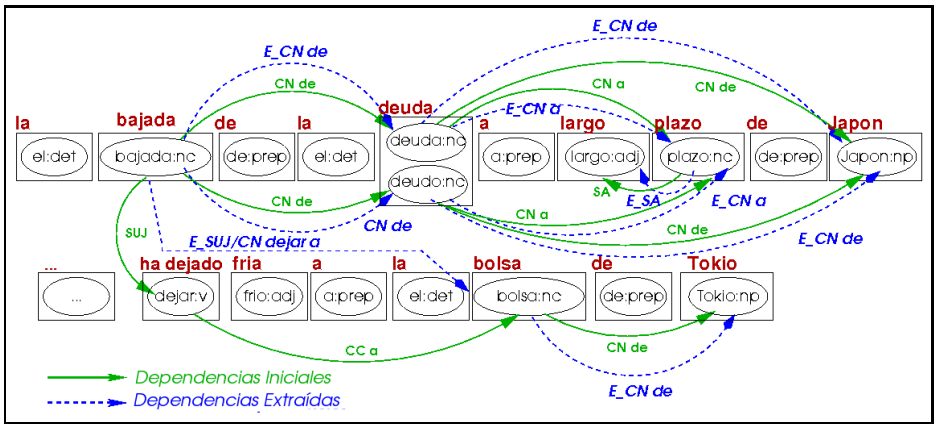

Figura 1: Grafo de dependencias del análisis sintáctico

Introducimos la notación y nomenclatura, a través de un ejemplo que utilizaremos en el resto de este apartado. Tomamos la frase *"La bajada de la deuda a largo plazo de Japón ... ha dejado fría a la bolsa de Tokio"*, de la Figura 1. Las formas rectangulares son llamadas *grupos* que indican las posiciones que ocupan los distintos elementos en la frase. Los lemas junto con su correspondiente categoría gramatical se representan mediante elipses llamadas *nodos*. Con líneas continuas se representan dependencias binarias entre palabras a través de alguna construcción sintáctica.

Tratamos de llegar a la fase de análisis semántico sin descartar ninguna posibilidad a dicha interpretación. Ello permite generar una estructura semánttica sobre la que aplicar, sin restricciones previas, procesos de adquisición del conocimiento que supondrían un filtrado de interpretacions inviables o inadecuadas. Así, la ambigüedad léxica ilustrada en la Figura 1 en el grupo correspondiente a la palabra *"deuda"*, debería decidirse en favor de la primera alternativa porque tenemos la certeza intuitiva de que *"deuda"* se refiere a la obligación que una persona tiene de pagar un dinero y tal situación no se refiere a un familiar. Al trabajar con un corpus especializado, ésto podrá confirmarse explorando el mismo en profundidad. Ello conlleva la consideración de un proceso iterativo convergente en un punto fijo en el que cada paso de la cadena secuencial re-inyecta en la siguiente iteración la información previa obtenida en forma de peso probabilístico [2].

Nuestro interés se centra por una parte en las dependencias entre sustantivos y adjetivos. Esto justifica el hecho de guardar las dependencias con líneas discontiuas mostradas en la Figura 1. Así, la palabra *"plazo"* (sustantivo) se relaciona con *"largo"* (adjetivo). Por otra parte, centramos nuestro interés en la extracción de dependencias entre sustantivos unidos por una preposición como *"bolsa de Tokio"* o unidos por un verbo, como en la dependencia *"bajada dejar_a bolsa"*, extraída de dos relaciones: *"bajada"* con *"dejar"* y *"dejar"* con *"bolsa"*.

El algoritmo intenta eliminar este tipo de ambigüedades añadiendo una fase de simplificación que aplica una restricción sintáctica simple, a saber, en una frase una palabra dada sólo puede tener a otra como gobernante. Así vemos en el ejemplo de la Figura 1, que *"Japón"* es gobernado por *"deuda"* (obligación de pago de dinero) y por *"deuda"* (familiar o pariente). Es necesario eliminar una de estas dependencias. No se considera ninguna otra restricción topológica y por consiguiente una palabra gobernante puede tener más de un gobernado, como el caso de "bajada" como gobernante de *"deuda"* y de *"bolsa"*. Además, una palabra puede ser a la vez gobernante y gobernado, como en *"plazo"*; gobernante de *"largo"*, pero también gobernada por *"deuda"*.

2.2.2. Agrupando los términos

La generación de clases semánticas se inspira en una propuesta de exploración de errores[4] originalmente diseñada para identificar la información errónea en entornos de análisis sintáctico [7]. Esta técnica combina dos procesos iterativos complementarios. Dada una iteración, el primero calcula, para cada par gobernante/gobernado en una frase, la probabilidad de la correspondiente dependencia. El segundo proceso calcula, a partir del primero, la clase semántica más probable para ser asignada a los términos de la dependencia. Así en cada iteración se procede a una desambiguación tanto a nivel sintáctico como a nivel semántico.

Ilustramos la agrupación de términos en nuestro ejemplo de ejecución de la Figura 1, centrándonos en la dependencia etiquetada como [_CNde] que se refiere a *"deuda"* y a *"Japón"*. Introducimos ambos procesos

[4] en terminología anglosajona, *error mining*.

Cuadro 2: Extracción de clases para *"deuda de Japón"*

$$1.\quad P(\text{deuda:uc:moneda, [\_CNde], Japón:up:países})_{\text{local(0)}} = \frac{\begin{matrix}P(\text{deuda:uc, [\_CNde], Japón:up})_{\text{local(0)}}\\ P(\text{deuda:uc:monedas})_{\text{local(0)}}\\ P(\text{Japón:up:países})_{\text{local(0)}}\end{matrix}}{\Sigma_{X,Y}\, P(\text{deuda:uc:X})_{\text{local(0)}}\; P(\text{Japón:up:Y})_{\text{local(0)}}}$$

$$2{,}1\quad P(\text{deuda:uc:monedas, [\_CNde]}, X)_{\text{global(n+1)}} = \frac{\Sigma_X\, P(\text{deuda:uc:monedas,[\_CNde]}, X)_{\text{local(n)}}}{\#\text{dep}_{\text{local(n)}}(\text{deuda})}$$

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

$$2{,}3\quad P(\text{deuda:uc:monedas, [\_CNde], Japón:up:países})_{\text{global(n+1)}} = \begin{matrix}P(\text{deuda:uc:monedas, [\_CNde]}, X)_{\text{global(n+1)}}\\ P(Y, \text{[\_CNde], Japón:up:países})_{\text{global(n+1)}}\end{matrix}$$

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

iterativos para este caso particular y hablaremos sin distinción de peso, probabilidad o preferencia para referirnos al mismo concepto.

1. Empezamos calculando la probabilidad local de la dependencia en cada frase, que depende del peso de cada palabra que a su vez depende de que su categoría léxica sea la correcta. El peso inicial asignado a una categoría es calculado por el algoritmo de exploración de errores. También tenemos en cuenta la probabilidad inicial de la dependencia observada que es una simple proporción de todas las posibles dependencias implicadas en las categorías consideradas. La normalización se calcula en base a la preferencia por una posible categoría léxica de los términos implicados.

2. Re-inyectamos localmente en la frase las probabilidades calculadas sobre el corpus en conjunto, para así re-calcular el peso de todas las posibles dependencias, después de lo cual podremos estimar globalmente las más probables. La normalización viene dada por el número de dependencias que conectan los términos considerados.

3. El valor local en la nueva iteración debería tener en cuenta tanto las preferencias globales como la re-incorporación local de esas preferencias en la frase. La normalización se calcula a partir del peso local previo y del peso global de la dependencia actual, teniendo en cuenta todas las posibles categorías léxicas asociadas con cada uno de los términos considerados.

Con respecto a la asignación de clases semánticas, la secuencia de pasos se puede ver en el Cuadro 2, que muestra la probabilidad de que la palabra *"deuda"* se refiera al grupo de `Monedas` y *"Japón"* al grupo de `Países`. Las clases `Monedas` y `Países` se definen *a priori* en una lista de clases semánticas:

1. En cada frase se calcula la probabilidad local de la dependencia para saber si sus elementos pertenecen a una u otra clase. Iniciamos el proceso a partir del peso local calculado y teniendo en cuenta la preferencia inicial de que los términos envueltos se correspondan a las clases consideradas[5]. La normalizacion viene dada por las probabilidades de todas las posibles clases incluyendo cada uno de los términos considerados que aquí se representa por las variables X e Y.

2. Entonces calculamos este peso a nivel global, reinyectando el resultado obtenido a nivel de corpus, para así poder re-calcular los pesos de todas las posibles clases en la frase. Esto se consigue calculando primero la probabilidad a nivel de corpus (2.1 y 2.2) para cada término y clase semántica, sin tener en cuenta ni el contexto izquierdo ni el derecho, representado por las variables X e Y respectivamente. La probabilidad final (2.3) es una combinación de las dos anteriores.

3. Después de cada iteración re-introducimos el valor global previo para obtener un nuevo valor local. La normalización se hace sumando las preferencias correspondientes a los términos y clases implicadas en la dependencia, para todas las posibles clases semánticas consideradas.

Una vez aplicadas éstas dos aproximaciones es posible construir una jerarquía teniendo en cuenta los elementos obtenidos en cada clase. Esta jerarquía, como en

[5]Las probabilidades se obtienen como una proporción de clases consideradas. Si el término se encuentra en una lista asociada, éste parte de una probabilidad mayor.

el caso de los términos, es almacenada en un fichero OWL.

3. CONOCIMIENTO Y RI

Las corrientes de investigación actuales en el campo de la RI centran sus esfuerzos en la mejora de los procesos de indexación y en la gestión inteligente de las consultas. El uso de diferentes tipos de estructuras de conocimiento tales como jerarquías u ontologías brindan claras ventajas para este cometido [5]. La cobertura y la precisión mejoran y las peticiones del usuario pueden concretarse de forma automática.

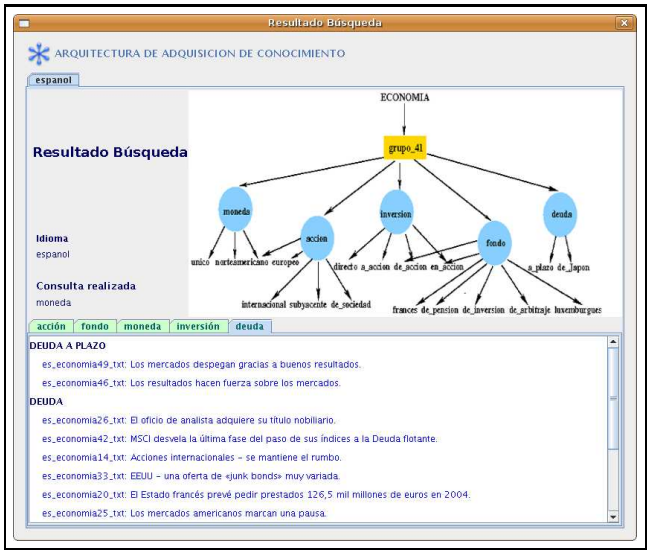

Figura 2: Sub-jerarquía para la consulta *"acción"* usando una estrategia basada en términos

3.1. PROPUESTA DE RI

Es necesario plantear nuevos métodos a la hora de crear y manipular los índices, haciéndolos más flexibles y ricos semánticamente. Esto debería extenderse al tratamiento de consultas, pues a menudo el usuario no tiene una idea exacta de la información contenida en la colección documental, lo que condiciona fuertemente sus interrogaciones. Una estructura jerárquica permite guiar las dos tareas aquí tratadas, el proceso de recuperación y la manipulación de consultas.

Nuestro acercamiento se apoya en una jerarquía de conceptos construida a partir de las relaciones semánticas que emergen del texto. Una herramienta de visualización permite la exploración del espacio de información; acadando así un entorno adecuado para la presentación de los documentos relevantes a una consulta. El punto de partida de esta propuesta es el de aprovechar, en el momento de la recuperación, la vinculación existente entre la jerarquía de conceptos y los textos.

El corpus tratado genera una estructura conceptual organizada alrededor de clases. A través de la proyección de las consultas se obtiene una conexión automática entre la estructura y la colección documental. El todo se apoya en una interfaz de visualización que vemos en las Figuras 2 y 3, donde se muestran la sección de la jerarquía relacionada con una determinada petición y en la parte inferior los documentos seleccionados.

3.2. PLANTEAMIENTO PRÁCTICO

En la práctica nos interesa analizar cómo es posible, partiendo de la adquisición de conocimiento, explotar las estructuras creadas y mejorar así los resultados obtenidos en la recuperación de documentos. En los apartados previos se han descrito dos técnicas de adquisición de conocimiento, una basada en términos y otra basada en dependencias ambas se integran para delimitar un marco de evaluación.

El motor de búsqueda utilizado es LUCENE[6], que asume la creación del índice una vez finalizado el análisis sintáctico. La jerarquía de conceptos entra en juego en la fase de consulta, en lo que hemos denominado *fase de expansión*. Ésta consiste en comparar la consulta del usuario con las estructuras del grafo conceptual y descubrir así la existencia de elementos relacionados con la petición. En caso afirmativo, estos elementos se toman de la jerarquía y se buscan en el índice, recuperando los documentos relevantes.

En caso de introducir una petición con más de una palabra, el proceso se lanza sin intervención de la jerarquía. Una consulta de este tipo es filtrada por una lista de parada y a continuación pasada al motor de búsqueda. Las peticiones que incluyan operadores lógicos, como AND y OR, son tratadas directamente por LUCENE. En cualquier otro caso, en un primer momento se lleva a cabo la expansión de la consulta haciendo uso de la estructura conceptual para, seguidamente, pasar los resultados al índice.

Una vez introducida una petición, el sistema busca la clase correspondiente en la estructura de conocimiento y recupera los conceptos asociados con esa clase. Esta función se ejecuta indistintamente de si estamos operando sobre la jerarquía de términos o sobre la jerarquía de dependencias. Sin embargo, los resultados varían en función de la estrategia elegida, pues el número de relaciones semánticas implicadas cambiarán si estamos considerando una u otra aproximación. Para ilustrarlo, describiremos la respuesta del sistema a la consulta *"acción"* planteada para ambas estrategias.

Nos centramos en la Figura 2 resultado de la consulta

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

usando la ontología por términos. Vemos que *"acción"* pertenece a la misma clase que *"deuda"*, *"moneda"*, *"fondo"* e *"inversión"*. Estos elementos son cabezas de términos y se representan en un círculo. Sus expansiones vienen dadas por las flechas, como en *"acción de sociedad"*. El cuadro da nombre a la clase, un nombre genérico y asignado automáticamente.

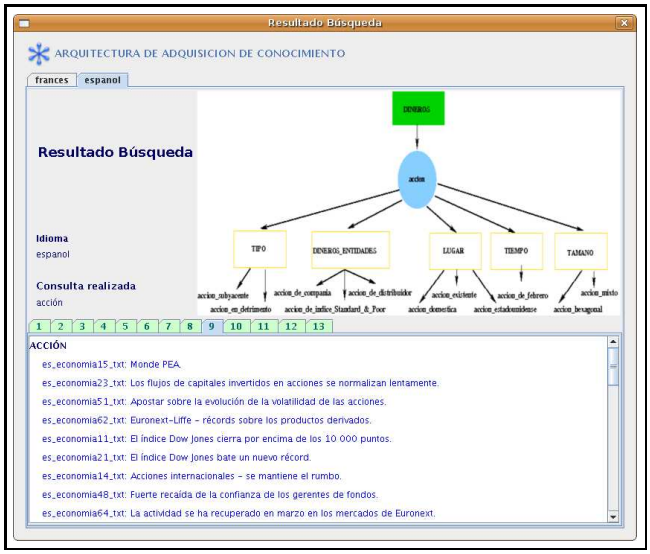

Figura 3: Sub-jerarquía para la consulta *"acción"*, usando una estrategia basada en dependencias

Nos centramos, ahora, en la Figura 3 resultado de la consulta usando la ontología por dependencias. Se definen dos tipos de clases: `conceptos` y `propiedades`. Las propiedades se consideran características de los conceptos. La interpretación es la siguiente: el gobernante es la consulta actual y está representado por un círculo. El cuadro superior indica la clase a la que pertenece el gobernante, en este caso es un concepto, `Dineros`. Los rectángulos representan el tipo de relación que se establece entre el gobernante y sus gobernados. Así, ambos pueden ser conceptos como *"acción"* y *"Standard and Poor's"* donde la relación tiene por nombre `Dineros_Entidades`. El gobernante puede ser un concepto y el gobernado una propiedad como en *"acción subyacente"*. Esta interpretación se puede extender al caso en el cual el gobernante es una propiedad y el gobernado un concepto.

4. CONCLUSIÓN

Hemos descrito una estrategia en el ámbito de la RI, basándonos en una indexación inteligente que se beneficia de las relaciones semánticas entre los conceptos presentes en la colección de textos. En contraste con acercamientos previos, elegimos una técnica que implica la ejecución de un número importante de tareas en modo no supervisado. Presentamos un conjunto de ejemplos en el campo de la economía.

Generamos de forma dinámica una estructura de conceptos que sirve de base para el módulo de RI, aproximación interesante para explorar nuevos dominios de conocimiento. Se trata de eliminar, en lo posible la traza de subjetividad introducida por el programador, reduciendo el factor humano en la toma de decisiones. Se procede, además, a un refinamiento de las consultas en base a la estructura conceptual.

Agradecimientos

Trabajo parcialmente subvencionado por el MEC en el proyecto HUM2007-66607-C04-02 y por la Xunta de Galicia en los proyectos PGIDIT05PXIC30501PN, 07SIN005206PR y la Red Gallega para el Procesamiento del Lenguaje y la RI.

Referencias

[1] Aussenac-Gilles, N. and Mothe J. 2004. Ontologies as background knowledge to explore document collections. In *RIAO*, pp. 129–142.

[2] Fernández M., E. de la Clergerie and Vilares M. Knowledge acquisition through error mining. 2007 In *RANLP, Proceedings*, pp. 220–224.

[3] Grefenstette, G. 1994. Explorations in Automatic Thesaurus Discovery. isbn: 0792394682

[4] Harris, Z.S. 1968. *Mathematical Structures of Languages*. J. Wiley & Sons.

[5] Masolo, C. 2001. Ontology driven information retrieval. Report of the IKF, 3180:371–380

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

[7] Sagot, B. and É. Villemonte de La Clergerie. 2006. Error mining in parsing results. In *Proc. of the 21st Int. Conf. on Computational Linguistics*, pages 329–336.

[8] Vilares, J., M.A. Alonso, and M. Vilares. 2004. Morphological and syntactic processing for text retrieval. *LNCS ISSN:0302-9743*, 3180:371–380.