Una aproximación declarativa a la clasificación de documentos

Francisco P. Romero[1] **Pascual Julián Iranzo**[2] **Mateus Ferreira-Satler**[2] **Juan Gallardo-Casero**[2]

[1] Escuela Universitaria de Ingeniería Técnica Industrial de Toledo,
Universidad de Castilla La Mancha,
FranciscoP.Romero@uclm.es
[2] Escuela Superior de Informática de Ciudad Real
Universidad de Castilla La Mancha,
Pascual.Julian@uclm.es, msatler@gmail.com, Juan.Gallardo@alu.uclm.es

Resumen

En este trabajo se presenta una aproximación a la clasificación de documentos sin necesidad de realizar un proceso de entrenamiento previo. En este caso el clasificador se define a partir de las descripciones semánticas de los conceptos relacionados con cada una de las categorías preexistentes. Para ello se utilizan relaciones de proximidad entre conceptos extraídas de ontologías y tesauros de propósito general. La base declarativa del proceso de clasificación es un mecanismo de unificación borrosa que facilita un proceso de búsqueda flexible de los conceptos en los documentos independiente de la base de conocimiento semántica que sirva de soporte.

Palabras Clave: Clasificación de Documentos, Unificación Borrosa, Búsqueda Flexible, Relaciones de proximidad.

1. INTRODUCCIÓN

La clasificación es el proceso de búsqueda de una serie de modelos o funciones que permiten distinguir entre una serie de clases o conceptos establecidos a priori sobre los datos. Este modelo será utilizado para identificar la clase de los objetos que aún no hayan sido clasificados [5]. La clasificación de documentos, ubicada dentro del aprendizaje supervisado, es una tarea que consiste en determinar la clase o clases a las que pertenece un documento [11].

Dada la naturaleza de la clasificación de documentos, el proceso se realiza habitualmente en varios pasos. En primer lugar, se cuenta con ejemplos dados de documentos pertenecientes a las clases predefinidas. Todos estos ejemplos son etiquetados con respecto a una de estas clases para formar el conjunto documental de entrenamiento. El algoritmo de aprendizaje correspondiente realizará un proceso de inducción obteniendo como resultado una descripción de cada una de las clases definidas. Esta descripción será posteriormente utilizada para clasificar de forma automática los nuevos documentos, por lo que de ella depende el grado de exactitud del sistema clasificador. Uno de los problemas que presentan estos sistemas se refiere al aumento exponencial del volumen del conjunto de entrenamiento cuando se pretende lograr un grado de exactitud suficiente ante una estructura compleja y ante ejemplos difíciles de clasificar a priori.

En este trabajo se propone prescindir del procedimiento de entrenamiento del clasificador. Con esta finalidad se utilizará como base la descripción semántica de los conceptos vinculados a cada una de las categorías predefinidas, es decir, a partir del concepto que define cada categoría se obtendrán las relaciones semánticas positivas que tenga con otros conceptos. La utilización de un proceso de recuperación de información basado en conceptos nos permitirá superar las dificultades inherentes a la recuperación basada únicamente en aspectos lexicográficos [3] siempre y cuando se cuente con una definición correcta y concreta del propio concepto para lo cual se utilizarán tesauros y ontologías de propósito general como Wordnet [12] y Concept Net [10]. Estas descripciones conceptuales serán la entrada a un mecanismo operacional de unificación borrosa que permite la búsqueda flexible de conceptos en documentos [8].

El resto del artículo está organizado de la siguiente manera. La sección 2 incluye una breve semblanza sobre relaciones de proximidad entre conceptos. En la sección 3 se presenta el lenguaje Bousi∼Prolog que proporciona los mecanismos necesarios para poder realizar la clasificación de los documentos de forma declarativa. En la sección 4 se presenta el experimento realizado con el fin de verificar el funcionamiento de esta aproximación. Por último, se finaliza con las conclusiones y el trabajo futuro en la sección 5.

2. RELACIONES DE PROXIMIDAD ENTRE CONCEPTOS

La proximidad entre conceptos permite definir cuán cerca o cuán similar es un concepto en relación a otro, es decir, permite identificar si dos conceptos se relacionan semánticamente. Normalmente, los valores de similitud varían dentro de un rango de 0 y 1.

Existen muchas propuestas y métodos en la literatura que tratan de estimar un valor de similitud entre conceptos. En [9] se presenta un estudio comparativo de algunos de estos métodos, destacando las ventajas y desventajas de cada uno de ellos. Para comparar las metodologías, los autores propusieron una taxonomía de propuestas, como muestra la Figura 1.

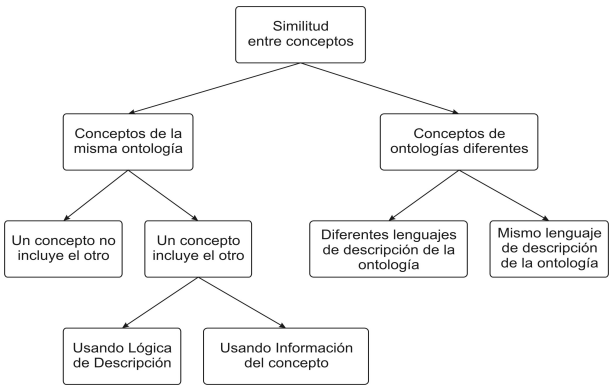

Figura 1: Taxonomía de propuestas para medir la similitud entre conceptos.

Dado que todo conjunto de conceptos relacionados puede ser incluido en una ontología, las propuestas se dividen en métodos para conceptos de la misma ontología y métodos para conceptos en ontologías diferentes. El primer tipo de métodos se basa en la idea de que un concepto incluye el otro, es decir, un concepto es una subclase o una superclase del otro en la jerarquía de la ontología. Si esto ocurre, las propuestas se subdividen en métodos basados en Lógica de Descripción y métodos basados en informaciones de los conceptos, como por ejemplo, sus propiedades. Las propuestas que soportan conceptos de diferentes ontologías se separan en métodos que soportan ontologías descritas en el mismo lenguaje y métodos que soportan ontologías descritas en diferentes lenguajes.

En [16], los autores proponen un modelo para medir la similitud semántica entre conceptos con base en estudios filosóficos y de inteligencia artificial. El modelo usa características de conjuntos para describir conceptos. Un concepto consta de muchas características, al que llaman conjunto de características (features set). A estos conjuntos se aplican determinadas reglas y fórmulas para derivar la similitud de los conceptos.

Wang et al. [14] propone un modelo de similitud basado en el Análisis de Conceptos Formales (Formal Concept Analysis). La idea es que cuanto más objetos y atributos sean compartidos por dos conceptos, más similitud hay entre ellos. El método presentado combina características e informaciones estructurales en la toma de decisión, para intentar acercarse a la forma del discernimiento humano.

En [2], Formica presenta un método para evaluar la similitud de conceptos que tiene en cuenta la taxonomía y la estructura de los conceptos. Se basa en la asociación entre probabilidades y conceptos de la jerarquía. Se define la idea de similitud entre contenido de información, que se combina con la estructura de los conceptos y con una variable de peso definida por un experto en el dominio para generar un valor de similitud entre conceptos.

Para obtener la distancia o proximidad semántica entre conceptos será necesario recurrir a tesauros y ontologías como WordNet o Concept Net. Las relaciones de proximidad utilizadas en este trabajo están basadas en relaciones entre conceptos que proveen de forma directa o indirecta estos recursos de uso no específico.

En concreto, las medidas de proximidad conceptual que van a ser utilizadas como base de conocimiento para realizar las tareas de búsqueda flexible dentro del sistema Bousi∼Prolog van a ser las siguientes.

- *Analogía estructural*[10]: No es una medida estrictamente semántica, sino que indica un grado de similitud en cuanto a las características del concepto representado por la palabra. Dos conceptos serán análogos si tienen características y funcionalidades similares.

- *Proximidad contextual*[10]: Refleja la proximidad entre dos conceptos en base a sus contextos habituales. Dado que Concept Net está concebida como una red semántica, la vecindad contextual alrededor de una palabra o nodo viene dada por la extensión del radio desde el nodo origen a los diferentes nodos relacionados. Se calcula contabilizando los caminos entre los dos nodos y la dirección de los arcos. Como resultado se aporta una medida de mutua información entre dos nodos.

- *Relaciones de similitud basadas en la sinonimia*: En [13] se presenta una fórmula basada en el coeficiente de Jaccard con el fin de calcular la similitud entre dos términos a partir de los conceptos que representan utilizando como base las relaciones semánticas existentes en WordNet.

3. EL LENGUAJE BOUSI∼PROLOG Y LA BÚSQUEDA FLEXIBLE DE SOLUCIONES

Bousi∼Prolog (BPL, para abreviar) [6, 7, 8] es un lenguaje de programación lógica donde la noción de "aproximación" se gestiona a un nivel sintáctico por medio de relaciones de proximidad. El objetivo princial de su diseño ha sido flexibilizar el proceso de repuesta a preguntas. Actualmente existen dos implementaciones de BPL:[1] una implementación de alto nivel [6], desarrollada mediante un metaintérprete escrito en Prolog, y otra de bajo nivel [7] que esencialmente es una adaptación de la Máquina Abstracta de Warren (WAM — Warren Abstract Machine) [15] (implementada en Java) para que pueda incorporar un algoritmo de unificación borroso.

Desde el punto de vista operacional, Bousi∼Prolog extiende el lenguaje Prolog mediante un mecanismo de *resolución SLD débil* que es una adaptación del principio de resolución SLD clásico donde la unificación sintáctica se reemplaza por un algoritmo de unificación borrosa (o débil) basado en relaciones de proximidad. Informalmente este algoritmo de *unificación débil* establece que dos términos $f(t_1, ..., t_n)$ y $g(s_1, ..., s_n)$ unifican débilmente con respecto a una relación de proximidad $\mathcal{R}$ si f y g son próximos (esto es, si $\mathcal{R}(f, g) > 0$) y cada uno de sus argumentos t_i y s_i unifican débilmente. Por consiguiente, el algoritmo de unificación no produce un fallo cuando se confrontan dos símbolos sintácticamente distintos, siempre que sean próximos, sino un éxito con un cierto grado de aproximación que es el mínimo entre el grado de aproximación, $\mathcal{R}(f, g)$, de los símbolos en cabeza y los grados de aproximación con que unifican débilmente cada uno de sus componentes. Así pues, Bousi∼Prolog computa tanto respuestas computadas (sustituciones) como grados de aproximación.

En esencia, la sintaxis de Bousi∼Prolog es como la de Prolog pero enriquecida con el símbolo predefinido "∼∼" empleado para describir las relaciones de proximidad mediante lo que denominamos una *ecuación de proximidad* de la forma: `<symbol> ~~ <symbol> = <degree>`. Aunque, formalmente, una ecuación de proximidad representa una entrada de una relación borrosa binaria arbitraria, su lectura intuitiva es que dos constantes, símbolos de función n-arios o símbolos de predicado n-arios son próximos con un cierto grado. Informalmente, empleamos el símbolo predefinido "∼∼" como una notación comprimida para la clausura simétrica de una relación borrosa binaria arbitraria (esto es, una ecuación de proximidad $a \sim\sim b = \alpha$ puede entenderse en ambos sentidos: a es próximo a b y b es próximo a a con grado α). Por consiguiente, en su forma más simple, un programa Bousi∼Prolog es una secuencia de hechos y reglas Prolog seguido por una secuencia de ecuaciones de proximidad.

En este trabajo se han aprovechado las notables características del lenguaje Bousi∼Prolog para la búsqueda flexible de soluciones a la hora de implementar el programa `inspect.bpl` (véase la Figura 2), que inspecciona una secuencia de documentos almacenados en un fichero cuya estructura interna es conforme al formato estándar SMART (véase la Figura 3). Este programa comprende una ontología de términos, modelada mediante ecuaciones de proximidad, y una serie de predicados, donde el predicado `inspect/3` constituye la interfaz principal con el usuario. De manera resumida, puede decirse que el predicado `inspect/3` lee, palabra por palabra, un fichero, denominado `FileName`, buscando palabras que son próximas (según la ontología definida) a una palabra clave, denominada `Keyword`. El término `Keyword` es un tópico que caracteriza el documento. Como resutado de la inspección, el predicado `inspect` devuelve un registro con datos estadísticos, denominado `DataAccount`, con la siguiente estructura:

```
[[texNumber(1)|L1], [texNumber(2)|L2], ...]
```

Esto es, existe una sublista para cada texto almacenado en el fichero `FileName`. Cada sublista `Li` almacena una secuencia de triples `t(X,N,D)`, donde X es un término próximo o similar al término `Keyword`, con grado `D`, que ocurre `N` veces en el texto `texNumber(i)`. Para realizar esta búsqueda de términos próximos o similares a uno dado, el programa se apoya en el mecanismo de unificación borrosa implementado en el núcleo del lenguaje Bousi∼Prolog . Más concretamente, el predicado `processRemainderText` (véase la Figura 2) utiliza un operador de unificación débil, denotado por "∼∼", que es la contrapartida difusa al operador de unificación sintáctica del lenguaje Prolog.

Una vez terminada la inspección de los textos, los datos contenidos en el registro `DataAccount` se utilizan en el análisis de los textos, con objeto de lograr su clasificación.

4. EXPERIMENTOS

En este apartado se presenta el experimento que se ha realizado sobre una colección estándar con el fin de validar la aproximación propuesta. El experimento es útil a su vez para determinar qué relaciones de proximidad entre documentos representan mejor el conocimiento de un experto al clasificar un documento en su correspondiente categoría.

[1] http://www.inf-cr.uclm.es/www/pjulian/bousi.html

Reuters[2] es una colección de noticias que han sido formateadas y preparadas para su utilización en la evaluación de sistemas de categorización de documentos. Esto significa que está orientada al descubrimiento de los criterios que ha seguido el experto humano para clasificar un documento en una o varias categorías.

En concreto, esta colección consiste en 21.758 artículos generados por la agencia de noticias Reuters durante el año 1987. De entre estos documentos, únicamente algo más de la mitad de ellos (aprox. 12.900) tienen asignada una categoría de las 118 posibles definidas por la agencia, siendo posible asignar un documento a más de una categoría. Cada una de las categorías tiene un nivel diferente de importancia, existiendo categorías con menos de 10 documentos asignados y categorías que engloban a más de 1.000 documentos.

Para este experimento se han seleccionado de forma aleatoria 6 categorías de entre las 118 posibles. Además de esta selección de categorías hemos seleccionado aquellos documentos más significativos siguiendo el criterio propuesto en [1], con lo cual se utilizan un total de 160 documentos que pueden estar clasificados en una o varias categorías de forma simultánea (ver Tabla 1).

Tabla 1: Distribución de los documentos por clases.

Categoria	**Documentos**
wheat	71
sugar	36
rice	24
cotton	20
copper	18
lead	14

4.1. Procedimiento y resultados

Con la finalidad de realizar el experimento de clasificación de este conjunto de documentos a partir de las relaciones de proximidad conceptual anteriormente especificadas se definen los siguientes dos conjuntos de referencia.

- El conjunto T contiene las etiquetas que representan a cada una de las clases, de esta forma T = {*wheat, sugar, rice, cotton, copper, lead*}. Cada una de las etiquetas individuales es identificada mediante t_i.
- El conjunto D contiene los documentos que van a ser clasificados. Se representará con d_j a cada uno de los documentos del conjunto, mientras que t_{ij} representará la etiqueta teórica correspondiente al documento d_j. Por otro lado, t'_{ij} representa a la etiqueta obtenida tras el proceso de búsqueda flexible obtenida en el paso 3 del proceso que se explica a continuación. Como ejemplo de uno de estos documentos, se dispone del siguiente texto (ya procesado previamente) que está vinculado a la etiqueta "*wheat*":

```
agriculture,department,report,farm,own,
reserve,national,average,price,loan,
release,price,reserves,matured,bean,
enter,corn,sorghum,rates,bean,potato
```

Una vez establecidos estos conjuntos se realiza un proceso de clasificación de todos los documentos para cada una de las relaciones de proximidad especificadas en el subapartado anterior. Los pasos a seguir serán los siguientes:

1. Obtención de los grados de proximidad según la relación elegida para cada uno de las categorías predefinidas. Por ejemplo,

```
wheat~~bean=0.315.  bean~~corn=0.48.
wheat~~corn=0.315.  bean~~potato=0.5.
```

2. Se carga el programa resultante del proceso anterior en Bousi∼Prolog dentro del programa `inspect.bpl` con el fin de que sea utilizada para realizar las búsquedas flexibles que permitirán clasificar los documentos.

3. Buscar dentro de cada documento d_j cada una de las etiquetas t_i, con ello se obtiene una serie de grados de presencia de t_i en el documento d_j que vendrá representado por w_{ij} ($w_{ij} = g(t_i, d_j)$). Siendo g el predicado de búsqueda flexible que permite a Bousi∼Prolog encontrar t_i en d_j sin necesidad de que éste aparezca explícitamente.

```
BPL> sv g(wheat,L)
 L= [ (potato, 1, 0.107),
      (bean, 2, 0.315),
      (corn, 1, 0.315)]
w = 0,864
```

4. La etiqueta ganadora t'_{ij} corresponderá a aquella que obtenga un mayor peso ($t'_{ij} = arg\ max(w_{ij})$). Ésta será comparada con la categoría teórica (t_{ij}) y si coincide será una clasificación correcta. El porcentaje de clasificaciones correctas nos arrojará el grado de bondad que ofrece el uso de la ontología en el proceso de clasificación.

[2] www.daviddlewis.com/resources/testcollections

El proceso se ha realizado tanto con cada una de las relaciones semánticas definidas anteriormente como sin utilizar aportación semántica alguna, es decir, buscando únicamente mediante la igualdad sintáctica. Los resultados obtenidos se pueden observar en el tabla 2. La utilización de la relación de analogía estructural ha arrojado un aceptable número de elementos no clasificados que pertenecen casi al 50 % a los documentos que pertenecen a la categoría *"lead"* lo que no es sorprendente dado que la descripción del concepto utilizada no es suficientemente completa. Algo similar sucede al utilizar la proximidad contextual. En este caso, tanto el número de no clasificados como de errores cometidos es aún mayor. Los mejores resultados los ha ofrecido la utilización de la relación descrita en [13] ya que cuenta con una definición de los conceptos de mayor calidad y completitud. Otra conclusión que se puede obtener es la mejora considerable que se obtiene en la eficacia del proceso de clasificación al utilizar relaciones semánticas ya que se logra prácticamente doblar el número de documentos correctamente clasificados con respecto a la utilización exclusiva de la igualdad sintáctica.

Tabla 2: Resultados obtenidos.

Relación	**Correctos**	**Incorrectos** No Clasif/Errores
Analogía	74 %	16 %/10 %
Prox. Context.	59 %	30 %/11 %
Sinonimia	83 %	6 % / 11 %
Igualdad Sint.	40 %	48 %/ 12 %

5. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado una aproximación a la clasificación de documentos no basada en un proceso de entrenamiento sino en la definición mediante relaciones semánticas de los conceptos que describen a cada una de las clases objetivo. A partir de esta definición, la utilización de un lenguaje de programación lógica como Bousi∼Prolog permite realizar una búsqueda flexible de los conceptos que representan a las categorías dentro de los diferentes textos a clasificar. Este proceso arroja como resultado la clasificación en una o más categorías de cada uno de los documentos. El rendimiento obtenido en los experimentos es satisfactorio siempre teniendo en cuenta la dependencia existente con respecto a las relaciones semánticas utilizadas. Esto ha generado la aparición de ciertos problemas cuando la definición del concepto correspondiente a la categoría no es suficientemente completa.

Como trabajo futuro, es necesario en primer lugar, abordar el problema de escalabilidad del sistema de búsqueda flexible, ya que no permite realizar búsquedas sobre colecciones con más de 200 documentos. Una vez resuelto este problema se podrán realizar experimentos en los que se cuente con un volumen tanto de documentos como de relaciones semánticas mucho mayor. A su vez, será interesante comprobar la eficacia de la utilización de relaciones semánticas más complejas como las proporcionadas por la librería WordNetSimilarity[3] o las estudiadas en [4] basadas en la versión más reciente Concept Net. Otro proceso a llevar a cabo será la evaluación de la mejora de la eficacia en el proceso de clasificación mediante la utilización de las ontologías de propósito específico mezcladas con las ontologías de propósito general ya mencionadas.

Agradecimientos

La investigación realizada está financiada por los proyectos TIN2007-67494 y TIN2007-65749 del Ministerio de Ciencia e Innovación, los proyectos PEIC09-0196-3018 y PII1I09-0117-4481 de la Junta de Comunidades de Castilla-La Mancha.

Referencias

[1] C. Apté and F. Damerau and S.M. Weis, Automated Learning of Decision Rules for Text Categorization, ACM Transactions on Information Systems, pp. 23-30, (1994).

[2] A. Formica. Concept similarity by evaluating information contents and feature vectors: a combined approach. Commun. ACM. 52(3) p. 145-149. (2009).

[3] P. Garces and Jose A. Olivas and F. P. Romero. Concept-Matching IR Systems Versus Word-Matching Information Retrieval Systems: Considering Fuzzy Interrelations for Indexing Web Pages Journal of the American Society for Information Science and Technology, 57(4):564-576, 2006.

[4] S. Guadarrama and M. Garrido Concept-Analyzer: A tool for analyzing fuzzy concepts In: Proceedings of Information Processing and Management Uncertainty Conference (IPMU'2008). 1084-1089. (2008).

[5] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann. (2001).

[3] http://www.d.umn.edu/mich0212/cgi-bin/similarity/similarity.cgi

[6] P. Julián, C. Rubio and J. Gallardo. Bousi∼prolog: a Prolog extension language for flexible query answering. In: ENTCS, vol 248, pp. 131-147. Elsevier, Amsterdam (2009).

[7] P. Julián and C. Rubio. A similarity-based WAM for Bousi∼Prolog. In: LNCS, vol 5517, pp. 245–252. Springer, Heidelberg (2009).

[8] P. Julián and C. Rubio. A Declarative Semantics for Bousi∼Prolog. In: Proceedings of 11th International ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming (2009).

[9] D.N. Le, A.E.S. Goh, Current Practices in Measuring Ontological Concept Similarity, in Proceedings of the Third International Conference on Semantics, Knowledge and Grid. IEEE Computer Society. p. 266-269. (2007).

[10] H. Liu and P. Singh. ConceptNet: A Practical Commonsense Reasoning Toolkit. BT Technology Journal, Vol. 22(4). (2004).

[11] D. Meretakis and D. Fragoudis and D. Lu and S. Likothanassis. Scalable associationbased text classiffication. In Proc. 9th ACM Int. Conf. Information and Knowledge Management, Washington, US, pp. 5-11. (2000).

[12] G.A. Miller and R. Beckwith and C. Fellbaum. Introduction to wordnet: an on-line lexical database. International Journal of Lexicography 3(4):235–244. (1990).

[13] A. Soto and J.A. Olivas and M.E. Prieto. Fuzzy Approach of Synonymy and Polysemy for Information Retrieval. Proceedings of the International Symposium on Fuzzy and Rough Sets, ISFUROS-2006, Santa Clara, Cuba. (2006).

[14] L. Wang and X. Liu. A new model of evaluating concept similarity. Knowledge-Based Systems, 2008. 21(8): p. 842-846.

[15] D. H. Warren. An Abstract Prolog Instruction Set. Technical note 309, SRI International, Menlo Park, CA., October (1983).

[16] C. Wu and D. Dai and Y. Wan, Ontology Concept Similarity in Semantic Query, in Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery. IEEE Computer Society. p. 24-28. (2008)

```
%%%%%%%%%%%
%% inspect(FileName, Keyword, DataAccount),
%% inspect(i, i, o)

inspect(FileName, Keyword, DataAccount) :-
        seeing(OLD),
        see(FileName),
        write('Processing file, '), write(FileName),
        write('. This may take a while ...'), nl,
        processFile(Keyword, DataAccount),
        seen,
        see(OLD).

%%%%%%%%%%%
%% processFile(Keyword, DataAccount)
%% processFile(i, o)

processFile(Keyword, DataAccount):-processDocs(Keyword, [], DataAccount).

/*
processDocs/3,
lee una palabra, comprueba que es un delimitador
de identificador de documento ".I",
lee el identificador del documento
y eventualmente llama a processText/4.

processText/4,
lee una palabra, comprueba que es un delimitador de texto
".W" y lee y analiza el texto.
*/
%%%%%%%%%%%
%% processText(Keyword, AccAccount, NewAccAccount, EOF)
%% processText(i, i, o, o)

processText(Keyword, AccAccount, NewAccAccount, EOF) :-
        read_word(Word1,EOF1),
        (EOF1 = false ->
           (isTextDelimiter(Word1) ->
                processRemainderText(Keyword, [], TextAccount, EOF2),
                NewAccAccount = [TextAccount|AccAccount], EOF = EOF2;
                write('This file does not conform the standards.'),
nl, NewAccAccount = [['ERROR']|AccAccount], EOF = EOF1
           );
           write('End of file reached.'), nl,
           NewAccAccount = AccAccount, EOF = EOF1
        ).

%%%%%%%%%%%
%% processRemainderText(Keyword, TextAccAccount, TextAccount, EOF)
%% processRemainderText(i, i, o, o)

processRemainderText(Keyword, TextAccAccount, TextAccount, EOF) :-
    read_word(Word1,EOF1),
    name(Word1name, Word1),
    (EOF1 = false ->
    (isDocIdentifierDelimiter(Word1) ->
TextAccount = TextAccAccount,
EOF = EOF1;
(Keyword ~~ Word1name = AD ->
  (insert(t(Word1name,1,AD), TextAccAccount, NewTextAccAccount),
  processRemainderText(Keyword,NewTextAccAccount,TextAccount,EOF)
  );
  processRemainderText(Keyword, TextAccAccount, TextAccount, EOF)
)
    );
      TextAccount = TextAccAccount, EOF = EOF1
    ).
```

Figura 2: Fragmento del programa `inspect.bpl`.

```
.I 1
.W
<Texto Documento 1>
.I 2
.W
<Texto Documento 2>
          .
          .
          .
.I N
.W
<Texto Documento N>
```

Figura 3: Estructura de los documentos analizados.