UNIVERSIDAD DE MATANZAS "CAMILO CIENFUEGOS"

FACULTAD DE INGENIERÍA INFORMÁTICA

Título: Interfaz de Acceso en Lenguaje Natural a la Información de la Universalización de la Enseñanza Superior.

Autor: Ing. Katia Vila Rodríguez.

Coator: MSc. Antonio C. Fernández Orquín.

Matanzas, 2005.

Resumen.

El sistema responde consultas en lenguaje natural a una base de datos. En él se describen las bases que se siguen para su implementación, características principales, así como las técnicas que se emplearon para traducir las consultas en lenguaje natural a sentencias en el lenguaje formal de interrogación de la base de datos. Procesa consultas en lenguaje castellano utilizando *Prolog*, éstas son evaluadas en una base de datos que contiene información estadística y geográfica de la Universalización en Matanzas, con lo que se resuelve la ausencia de facilidades para lograr una sencilla adquisición de información por usuarios no expertos en informática. Entre los resultados más relevantes se encuentra la utilización del lenguaje natural para la interacción del usuario con la información disponible en la base de datos. Tiene una gran novedad práctica, ya que no se ha podido comprobar en la bibliografía consultada la existencia de una interfaz de acceso en lenguaje natural a base de datos para algún dominio en nuestro país. Además se puede considerar que también contiene aspectos novedosos en el ámbito teórico.

Índice

Introducción

El proceso de Universalización de la Enseñanza Superior en Cuba es un proceso joven, por tanto en la actualidad presenta dificultades en cuanto al tratamiento de toda la información disponible. Se aprecia la necesidad de buscar fórmulas para organizar toda la información relacionada con este proceso para así facilitar la adquisición de información concreta o estadística sobre estos datos. Además está latente la necesidad de elaborar alternativas de fácil acceso a la información por parte de los usuarios no expertos en el uso de las TIC's logrando una eficiente información sobre el proceso de Universalización de la Enseñanza Superior de todos los actores vinculados al mismo o de los interesados en el tema.

El objetivo general planteado fue el de elaborar una interfaz que reconozca un conjunto de oraciones en lenguaje natural destinado a la obtención de información de carácter estadístico y geográfico del proceso de Universalización en Matanzas.

El *procesamiento del lenguaje natural* es una de las ramas principales de la *Inteligencia Artificial*, y estudia una propiedad importante de la inteligencia humana: su capacidad de comunicarse por medio del lenguaje. Una parte importante del diálogo humano consiste en hacer preguntas y recibir respuestas y este proceso, realizado de forma ordenada y guiado por una finalidad, constituye uno de los procesos básicos del aprendizaje humano.

El trabajo tiene una gran actualidad, ya que el procesamiento del lenguaje natural se considera una de las bases para el desarrollo de la sociedad de la información del futuro. Por lo que se realizan muchísimas investigaciones en pro de lograr sistemas y aplicaciones cada vez más eficientes en la tarea de procesar el lenguaje natural; además se desarrollan muchos formalismos para resolver problemas lingüísticos como la ambigüedad, la elipsis, la anáfora, etc. Particularmente, el trabajo representa una contribución al incipiente desarrollo de sistemas de PLN en Cuba. Y el principal aporte práctico es una interfaz que facilita el proceso de adquisición de datos estadísticos de la Universalización de la Enseñanza en Matanzas, usando lenguaje natural.

Desarrollo

El PLN, es hoy en día, una parte esencial de la Inteligencia Artificial, que permite una comunicación hombre-máquina mucho más fluida y menos rígida que los lenguajes formales, a través de la investigación y formulación de mecanismos efectivos computacionalmente. Entiéndase por lenguajes formales, aquellos que el hombre ha desarrollado para expresar las situaciones que se dan en cada área específica del conocimiento humano, como la lógica, la matemática, la física, etc.

Covington define el PLN, en [Covington, 1994], como “el uso de las computadoras para entender los lenguajes naturales humanos, como el Español, el Inglés, el Francés, entre otros”. Por “entender” se refiere a que las computadoras sean capaces de reconocer y usar información expresada en lenguaje humano; no que las computadoras piensen, sientan o tengan inteligencia como los humanos.

Se considera que el lenguaje natural (LN) en la comunicación hombre-máquina es por un lado ventajoso, con respecto a otros medios de comunicación; en la medida en que el usuario no tiene que esforzarse para aprender como interactuar a través de él a diferencia de otros medios como son los lenguajes de comando o las interfaces gráficas. Por otro lado su uso también puede ser considerado un obstáculo porque la computadora tiene una limitada comprensión del lenguaje. Por ejemplo, el usuario no puede hablar sobrentendidos, ni introducir nuevas palabras, ni construir sentidos derivados, tareas que se realizan espontáneamente cuando se utiliza el lenguaje natural. Pero la mejor manera de ganar fluidez y eficacia en las interacciones hombre-máquina es consiguiendo que éstas sean lo más naturales posibles, a través del lenguaje, por lo que cada día se trabaja más en el objetivo de minimizar las desventajas que pueden proporcionar su uso.

I. Aplicaciones y áreas de principal interés del PLN

Las aplicaciones del procesamiento del lenguaje natural tradicionalmente se agrupan en dos áreas, según [Moreno, 1999], que son las aplicaciones basadas en diálogos y las aplicaciones basadas en el tratamiento masivo de información textual, las cuales serán tratadas a continuación.

I.1. Aplicaciones basadas en diálogos.

Estas aplicaciones son las correspondientes a la comunicación hombre-máquina, ya sea de forma escrita u oral. En esta área se han desarrollado diversos sistemas, que se pueden dividir en tres grandes grupos:

1 Sistemas de acceso a Base de Datos:

Se pueden definir como sistemas de pregunta/respuesta locales en los cuales el conocimiento está almacenado establemente en una base de datos, ya sea, estructurada o lógica. Su objetivo consiste en compilar las expresiones producidas por los usuarios en LN a una forma directamente interpretable por un sistema de gestión de base de datos utilizando su lenguaje de consulta formal.

2 Sistemas de acceso a otros dominios:

Son sistemas que actúan sobre dominios de información heterogéneos: interfaz con sistemas expertos, acceso a sistemas operativos, sistemas tutores o de asesoramiento, etc.

3 Sistemas de diálogo inteligente:

Son sistemas que examinan el aspecto multimodal de la comunicación que se establece entre el usuario y el ordenador, en toda su globalidad. Es decir, intentan formalizar aspectos tales como las intenciones y deseos del usuario, el conocimiento y las creencias acerca de ese conocimiento y la relación entre el conocimiento y la acción.

I.2. Aplicaciones basadas en el tratamiento masivo de la información textual.

Aquí se enmarcan las aplicaciones que corresponden al procesamiento de texto escrito, tal como libros, periódicos, informes, correo, etc. Esta área surge por la necesidad de disponer de herramientas para tratar la creciente cantidad de información textual disponible, ya sean mensajes, noticias, revistas electrónicas o diccionarios.

Es importante destacar que los Sistemas de Acceso a Base de Datos son más eficientes que otras aplicaciones dentro de estas dos áreas, porque el nivel de complejidad es menor, es decir, las computadoras no requieren mucho conocimiento acerca del mundo real.

Por lo que se puede concluir que la eficacia del PLN, depende de los límites que pongamos a la necesidad del conocimiento externo y la experiencia humana.

II. Estado del Arte

En los años 70 comienzan las primeras interfaces en LN a Base de Datos como el sistema LUNAR de Woods que permitía interrogar en inglés a una base de datos sobre las muestras de materiales recogidos en misiones de exploración espacial.

Respecto a las aplicaciones, en los años 80, comienza la construcción de una serie de sistemas cada vez más sofisticados en el campo de las interfaces con Bases de Datos, como TEAM, CHAT-80, ORBI [Pereira, 1982], USL que constituye una interfaz interactiva con un sistema de gestión de bases de datos relacionales, su objetivo es traducir las frases de entrada escritas en lenguaje natural a sentencias del lenguaje formal de interrogación de la base de datos [De Sopeña, 1983] y otros.

En términos generales de todas las áreas del PLN, en los años 90 y la actualidad, se han recuperado formalismos ya introducidos en los años 80, realizándose extensiones de éstos. Las extensiones consisten en la representación de las estrategias requeridas para el análisis y eliminación de la ambigüedad y de otros problemas del lenguaje. Estas aproximaciones permiten expresar más aspectos

del lenguaje y de manera más efectiva y eficiente, pero no han resuelto completamente el problema del LN ya que el uso del lenguaje es demasiado variado para poder representarlo completamente por medio de unas reglas y cada tarea del PLN presenta nuevos problemas no resueltos.

Las Interfaces de lenguaje natural para bases de datos pueden combinar los procesos de emparejamiento de patrones y análisis de palabras claves; para comprender la pregunta por parte del usuario, encontrar la forma lógica asociada y acceder a la base de datos para brindar la respuesta deseada.

Se puede decir mucho a favor de los sistemas de palabras claves, en el sentido de que ellos ignoran las palabras irreconocibles, lo que contrasta con la utilización de gramáticas semánticas, ya que ellas si analizan todas las palabras y sus combinaciones, por tanto es mucho más rígido en este aspecto que los sistemas que usan palabras claves. Además otra característica de importancia en los sistemas de palabras claves es que se pueden incrementar las reglas de traducción de una manera sencilla y eficiente; en cambio las gramáticas semánticas presentan dificultades a la hora de ampliar la gramática con nuevas construcciones, por lo que se vuelve complicado el hecho de incrementar la cobertura del lenguaje por parte del sistema.

III. Solución y Análisis de los Resultados.

Se presenta en detalle el trabajo principal realizado. Primeramente se especifican los requerimientos funcionales del sistema. Luego se realiza la descripción de la solución, destacando la arquitectura, y los módulos del sistema. Y por último se señalan los resultados del entrenamiento realizado a dicho sistema.

III.1. Requisitos funcionales del sistema.

1. Interpretar la consulta realizada por el usuario.
2. Capturar el significado del dominio, a través de las reglas de traducción.
3. Mostrar la respuesta obtenida de la base de datos.

III .2. Descripción de la solución.

Seguidamente se expondrán los aspectos más importantes en la implementación de la interfaz de acceso en lenguaje natural a la base de datos. Para ello se usaron las nociones de emparejamiento y sustitución, predefinidas en el compilador de *Prolog*, para obtener respuestas a preguntas. También se utilizaron reglas de simplificación y traducción para transformar las oraciones a su forma lógica y éstas a la forma lógica en términos de la base de datos.

III.2.1. Arquitectura del sistema.

La Figura 1 presenta la arquitectura del sistema de forma detallada y adaptada a los procesos que realiza el sistema.

Las preguntas realizadas por el usuario desencadenan una serie de procesos que finalizan en la obtención de las respuestas. A continuación se explican detalladamente cada uno de los procesos.

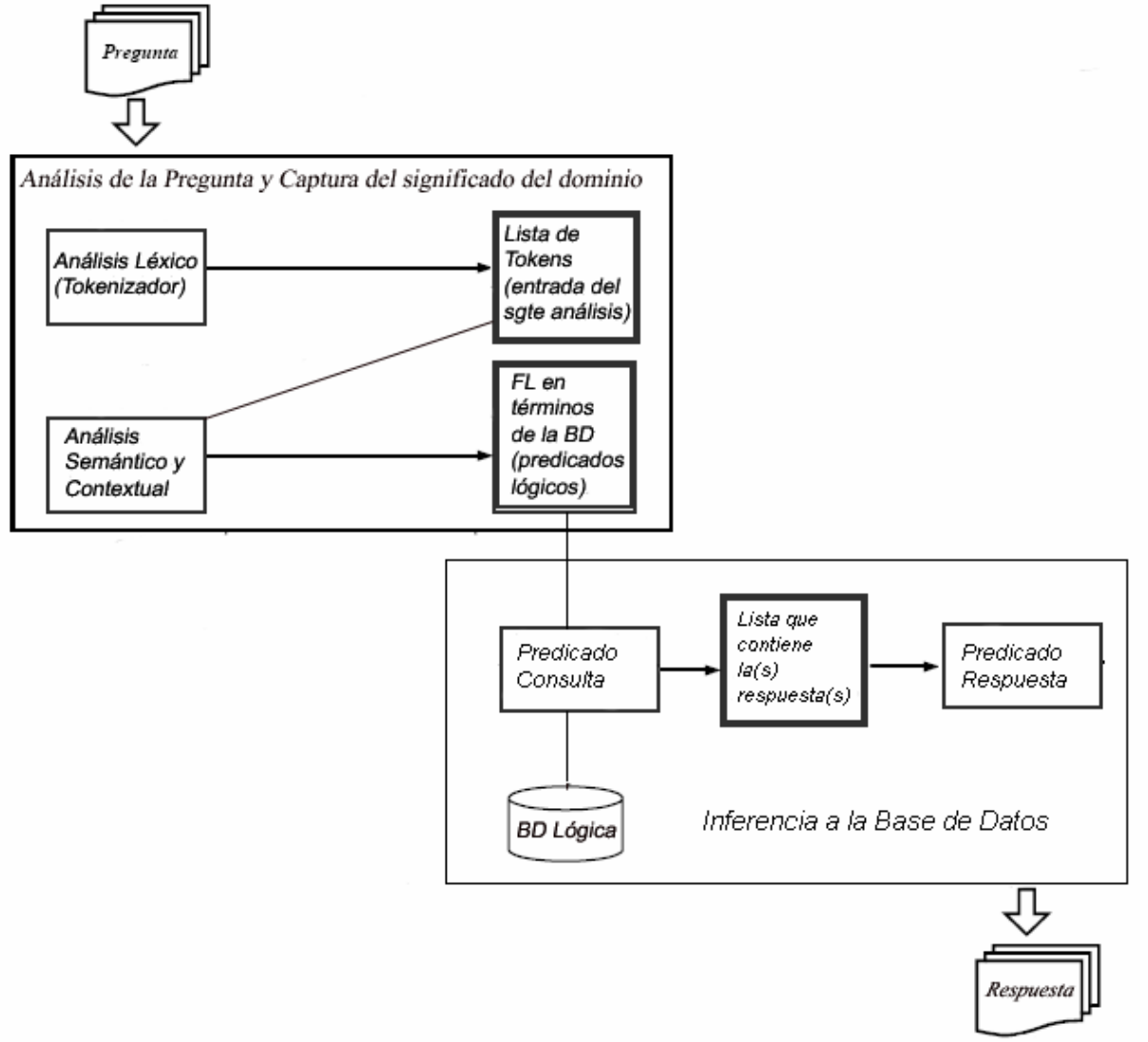

Figura 1. Arquitectura del sistema.

1. Análisis Léxico.

El análisis léxico consiste en la identificación de las unidades léxicas dentro de las oraciones que componen el texto objeto de análisis. Una aproximación simplificada consiste en considerar la palabra ortográfica como la unidad léxica en cuyo caso la función de éste consiste en identificar secuencias de caracteres separadas por espacios en blanco u otros símbolos separadores (coma, punto y coma, etc.).

El análisis léxico se le realiza a la sentencia entrada por el usuario. Lo que se hace no es más que eliminar los espacios en blanco, signos de puntuación, las tildes en

las vocales acentuadas y sustituir todas las mayúsculas por sus minúsculas correspondientes. La salida es una lista de palabras, que se utilizará en el proceso que sigue a continuación. Para lograr con efectividad lo planteado se implementó un tokenizador, del cual se comentará posteriormente.

Un *tokenizador* realiza lo que se puede llamar un análisis lexicográfico o análisis lineal (*scaning*), su tarea fundamental consiste en leer de izquierda a derecha los caracteres, introducidos a través del teclado, que componen la pregunta realizada por el usuario y producir como salida una secuencia de "*tokens*" o símbolos con un significado para la aplicación.

2. Análisis Semántico.

En el momento de emparejar la oración completa a un patrón, un sistema de palabras claves busca una palabra específica en la oración y responde a cada palabra de una manera determinada.

A partir del conjunto de tokens generados por el analizador léxico, el analizador semántico empareja cada palabra clave reconocida dentro de la lista, o combinaciones de éstas con sus respectivas formas lógicas. Con lo que se conforma el significado de la sentencia. Para lograr esto lo primero que se hace es definir el tipo de oración, analizando la primera palabra clave que se encuentra en la lista. Luego es necesario analizar la palabra clave siguiente a la primera que indicará el tipo de respuesta que se desea obtener, ejemplo: "Qué SUM tienen la carrera de Industrial" y "Qué carreras tiene la SUM de Matanzas". Se observa que en la primera la respuesta que se espera es un listado de SUM y en la segunda un listado de carreras; por lo que sí interesa en estos casos el orden de la palabras claves *SUM* y *carrera*.

En la fase de análisis semántico se ignoran las palabras ruidosas, las que no tienen ninguna información semántica importante para obtener la FL (Forma Lógica: es la representación del significado independiente del contexto). Para realizar el análisis semántico se implementaron reglas de simplificación y traducción que se detallan más adelante.

3. Análisis contextual.

El análisis contextual es el proceso por el cual la FL asociada a la oración (resultante del análisis semántico), es transformada a su equivalente FL en términos de la BD. Esta última será directamente evaluada sobre la BD, proporcionando la respuesta asociada a la consulta realizada por el usuario.

En este caso las FL's asociadas a la oración son transformadas por el análisis contextual en predicados lógicos capaces de interrogar a la BD lógica. Las reglas de traducción son también las que se encargan de llevar a cabo este proceso.

4. Proceso de inferencia a la base de datos.

Finalmente, se ejecuta la consulta conformada por predicados lógicos obtenidos

del proceso anterior. Los resultados son almacenados en una lista. Luego se muestra la solución, para ello se analiza el formato que debe tener la respuesta, es decir, si la pregunta del usuario fue de cantidad, el sistema debe devolver precisamente un número que refleje la cantidad de resultados obtenidos. En un fichero se almacenan las consultas que el sistema no puede resolver porque no tiene conocimiento suficiente para ello; así se garantiza la futura ampliación de la base de conocimiento del sistema.

III.2.2. Descripción de los módulos

Son dos los módulos principales del sistema el *tokenizador*, que se encarga de realizar el análisis léxico y el de *traducción*, donde se encuentran las palabras claves definidas para el dominio y la implementación de las reglas de simplificación y de traducción.

1. Tokenizador.

En la aplicación se implementa un tokenizador (programa que divide una frase en palabras como átomos *Prolog*) en el fichero tokeniza.pl. Concretamente a partir del predicado *tokeniza(L)* se devolverá en *L* una lista de átomos *Prolog* correspondientes a las palabras leídas desde teclado en *leeFrase(S)* con *S* una lista de códigos ASCII. Por lo que se puede afirmar que su función es leer la frase introducida por el usuario a través del teclado y devolver una lista de palabras, eliminando así los espacios en blanco o cualquier símbolo sin interés para la aplicación. En la Figura 2 se muestra un fragmento del algoritmo del tokenizador.

```
tokenizador(L) :-
  leeFrase(S), !, tokenizadorAux(S, [], L), !.

/*****************************************************************************
  Extrae de S (lista de códigos ASCII) una palabra en Pal (string). Devuelve la lista restante en SF
*/
leePalabra([], [], Pal, Pal).

leePalabra(S, SF, Pal, PalF) :-
  S=[CS | RS],
  (caracterPalabra(CS) ->
    ( append(Pal, [CS], PalFT),
      leePalabra(RS, SF, PalFT, PalF) )
  ;
    (length(Pal,0) ->                                   % Significa que no hay palabra a almacenar
      (CS == 32 ->                                      % Se quitan de la cadena de entrada
        ( quitaBlancos(RS,RST),
          leePalabra(RST, SF, Pal, PalF) )
      ;
        ( PalF=[CS],                                    % Se almacena como palabra (podría ser :;,...)
          SF=RS )
      )
    ;
      ( PalF=Pal, SF=S )
    )
  ).

/*****************************************************************************
  Marca los posibles caracteres de una palabra
*/
caracterPalabra(C) :- 97 =< C, C =< 122.      % Minúsculas
caracterPalabra(C) :- 65 =< C, C =< 90.       % Mayúsculas
caracterPalabra(C) :- 48 =< C, C =< 57.       % Números
caracterPalabra(95).                          % Subrayado
%--------------------------- Cód. ASCII de ANSI ----------------------------------------------
caracterPalabra(160).                         % á
...
```

Figura 2 Fragmento del *Tokenizador*

2. Módulo de Traducción.

Este módulo se encuentra implementado en el fichero queryBD.pl.

Las reglas de traducción son las que realizan tanto el proceso semántico como el contextual de la consulta introducida por el usuario. Ya que va tomando cada token, buscando si es una palabra clave, si lo es la empareja con su FL y a la vez con su interpretación en términos de la BD evaluándose directamente sobre ésta. Así se proporciona la respuesta asociada a la consulta realizada por el usuario. Por tanto, las reglas de traducción son las que hacen la mayoría del trabajo.

La salida del traductor será una consulta *Prolog* que permitirá seleccionar posteriormente los registros apropiados. La cadena de palabras debe ser traducida a una consulta o pregunta *(query)* de *Prolog*, trabajando en la cadena palabra por palabra y adicionándole algo a la consulta por cada significado.

En la Figura 3. se muestran algunos ejemplos de reglas de traducción que se explican a continuación.

```
/*---- Para identificar el tipo de pregunta
              y el tipo de respuesta que se espera-----*/

traducir([W,W1 | RW], X, (RestoQueries, Query)) :-
  accion(W, W1, X, Query),
  !,
  traducir(RW, X, RestoQueries).

/*----- Para traducir una comparación utilizando
               dos operadores como: mayor que---------*/

traducir([Arg1, Op1, Op2, Arg2 | RW], X, (Q1, Q2, Q3, RestoQueries)) :-
  comparacion(Op1, Op2, Y, Z, Q3),
  argumento(Arg1, X, Y, Q1),
  argumento(Arg2, X, Z, Q2),
  !,
  traducir(RW, X, RestoQueries).

/*---- Saltar una palabra (ruidosa) desconocida----*/

traducir([_ | RW], X, Query) :-
  !, traducir(RW, X, Query).

/*---- Final de la lista de palabras---------*/

traducir([], _, true).
```

Figura 3. Ejemplos de Reglas de Traducción

III.3. Resultados del Entrenamiento.

Al sistema le han sido aplicados tres entrenamientos, a continuación se analizan los resultados obtenidos en cada uno de ellos.

El primer entrenamiento que se le aplicó al sistema fue de quince preguntas de las cuales seis no se pudieron interpretar de manera correcta, por lo que el resultado del entrenamiento arrojó un 60% de efectividad. Las preguntas con problemas fueron:

1. Qué matricula tiene Matanzas del sexo femenino.
2. Qué departamentos tiene Cultura Física.
3. Muestre la matrícula de Matanzas.
4. Qué carreras tiene Matanzas.
5. Donde se ubica Matanzas.
6. Cuáles SUM tiene Matanzas.

En las seis oraciones, se reconocen claramente los problemas de ambigüedad. Ejemplo: en la primera oración no se puede decidir con la información que se brinda si Matanzas se refiere al municipio o a la provincia, en la segunda la ambigüedad en el significado semántico se encuentra en la palabra clave (de tipo *test)* Cultura Física, que puede significar tanto la facultad como la carrera de Cultura Física. En la tercera y sexta oración ocurre lo mismo que en la primera con respecto a la palabra clave Matanzas. En la cuarta y la quinta ocurre lo mismo que en la primera oración pero además se adiciona otro posible significado semántico que puede ser la SUM o sede de Matanzas.

Estos problemas fueron rápidamente resueltos, se modificaron todas las reglas de traducción que permitían emparejar a un *test* con su FL sin estar acompañado por algún argumento. En la solución todas las reglas de traducción tienen como mínimo un argumento, por lo tanto si el usuario no lo específica no obtendrá respuesta a sus interrogantes. Además, en la guía disponible de cómo se debe acceder al sistema se le señala a los usuarios que deben ser más específicos en el momento de introducir la consulta porque de lo contrario no obtendrán ninguna respuesta a sus preguntas. A continuación se muestra cómo deberían reformularse las consultas que dieron problema anteriormente:

1. Qué matrícula tiene el municipio de Matanzas del sexo femenino.
2. Qué departamentos tiene la facultad de Cultura Física.
3. Muestre la matrícula de la SUM de Matanzas.
4. Qué carreras tiene la provincia de Matanzas.
5. Donde se ubica la sede de Matanzas.
6. Cuáles SUM tiene la provincia de Matanzas.

En el segundo entrenamiento de nuevo se aplicaron quince preguntas, de ellas tres no se lograron analizar completamente, hubo una mejora de un 20% con respecto a la efectividad del entrenamiento anterior, es decir, se alcanzó un 80%. Las siguientes preguntas fueron las que provocaron esa pérdida de efectividad:

1. Muestre la matrícula de la provincia de matanzas.
2. Qué asignaturas dirige el departamento de contabilidad.
3. Cuáles asignaturas tiene la carrera informática.

Hay que señalar, primero que todo, que esas oraciones presentan una estructura similar a las que presentaron problemas en el entrenamiento anterior, por no contemplar todos los detalles necesarios en las reglas de traducción. Pero la dificultad que se muestra es otra, lo que sucedía era que en los *test* declarados hasta el momento no se tenían en cuenta las repeticiones. Por ejemplo se tenían dos *test* para matanzas *test*('matanzas',49) y *test*('matanzas',4), el primero se

refería al municipio y el segundo a la provincia. Entonces al usuario realizar la pregunta “Muestre la matrícula de la provincia de matanzas.”, el *test* matanzas emparejaba con el primero que hallaba, que en este caso, era el código 49, que correspondía al municipio y no correspondía a ninguna provincia.

La solución que se desarrolló fue sencilla y efectiva, primero se especificó en cada *test*, con problemas de ambigüedad en su significado, el sentido semántico asociado a él; por ejemplo: *test*(municipio,'matanzas',49), *test*(provincia, 'matanzas', 4).

El segundo paso fue modificar las FL’s asociadas a los *test* en las reglas de traducción, quedando así: argumento(Arg1,X,Y,Query), *test*(Arg1,W, Y); por ejemplo, para la sentencia que venimos analizando quedaría de la siguiente manera: argumento(provincia,X,Y,mat_prov(X,Y)), *test*(provincia,matanzas,4).

En el tercer y último entrenamiento que se efectuó se aplicaron todas las preguntas recopiladas de las entrevistas realizadas a los usuarios del sistema, éstas se pueden ver en los *Anexos*. Se comprobó que el sistema funciona de manera óptima para todas esas consultas. Por lo que se puede afirmar que el sistema trabaja bien para el conjunto de patrones que caracterizan esas preguntas.

Conclusiones.

Con la realización de este trabajo se logró resolver la inexistencia de un mecanismo que permitiera una fácil adquisición de la información sobre la Universalización en Matanzas por usuarios inexpertos en la Informática.

Se logró elaborar una interfaz que reconoce un conjunto de oraciones en lenguaje natural destinado a la obtención de información de carácter estadístico y geográfico, a través del uso del análisis de las palabras claves.

Se definió la arquitectura y se desarrollaron los módulos necesarios para la implementación de la interfaz.

Se implementó el análisis léxico y semántico, los cuales son independientes del ámbito de aplicación. De esta forma para que esta aplicación funcione correctamente en otro dominio sólo sería necesario definir las nuevas palabras claves.

De los estudios llevados acabo y los resultados del entrenamiento realizado, se puede concluir que:

- El uso de herramientas y técnicas de PLN en sistemas de acceso en lenguaje natural a base de datos es imprescindible en el proceso de análisis de la pregunta, ya que facilita la información necesaria para lograr una buena precisión en la búsqueda de la respuesta.

- El uso de las palabras claves es una buena alternativa de entre las posibles técnicas analizadas, destaca su sencillez y los buenos resultados que ofrece en cuanto al tiempo de procesamiento, ya que no ofrecen obstáculo las palabras con variaciones porque se desprecian.

- La lógica de predicado como mecanismo para la representación del conocimiento, permite realizar una definición fácil del conjunto de sentencias que pueden inferirse de una base de conocimiento determinada.

- Para construir modelos computacionales que realicen inferencias a una base de conocimiento en un tiempo razonable pueden implementarse reglas de traducción que logren convertir la forma lógica en el lenguaje de interrogación correspondiente al sistema de gestión de bases de datos que se utilice.

El futuro de las aplicaciones del Procesamiento del Lenguaje Natural todavía no está claro, será en los próximos años cuando se determine su evolución. Lo que si queda claro es que cualquier pequeño paso hacia adelante en la tarea de hacer más fluidas las interacciones hombre-máquina será también un paso adelante en la construcción de la sociedad de la información del futuro. Por lo que se considera este trabajo un pequeño aporte al incipiente desarrollo de aplicaciones de PLN en Cuba.

Bibliografía.

[Allen, 1995]. J. Allen. *Natural Language Understanding* . 2nd ed. Benjaming Cummings series in Computer Science. 1995.

[Cazorla, 1999]. Miguel A. Cazorla, Otto Colomina, Francisco Escolano, Domingo Gallardo, Ramón Rizo y Rosana Satorre. *Técnicas de Inteligencia Artificial.* Universidad de Alicante, Textos Docentes. 1999.

[Covington, 1994] M. A. Covington. *Natural Language Processing for Prolog Programmers*. Prentice Hall. 1994.

[De Sopeña, 1983]. Luís De Sopeña. *USL: Un Sistema para Interrogar en Castellano a Base de Datos Relacionales.* Procesamiento del Lenguaje Natural, Revista SEPLN nº 1, octubre de 1983

[Ferrández, 1999] Antonio Ferrández. *Sistemas de Pregunta y Respuestas.* Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante. 1999

[Llopis, 2003] F. Llopis. *IR-n: Un sistema de Recuperación de Información basado en Pasajes.* Tesis Doctoral. 2003.

[Moreno, 1993a] L. Moreno, M. Palomar, M. Pastor. *Interpretación de la Comparación en consultas a una Base de Datos geográfica a través de la Lógica.* Procesamiento del Lenguaje Natural, Revista SEPLN nº 13, Pág. 259-277, Febrero 1993.

[Moreno, 1999]. Lidia Moreno, Manuel Palomar, Antonio Molina y Antonio Ferrández (1999). *"Introducción al Procesamiento del Lenguaje Natural"*, Servicio de publicaciones de la Universidad de Alicante, Alicante.

[Norvig, 1991]. P. Norvig. *Paradigms of artificial intelligence programming: Case studies in Common Lisp*. San Mateo, California: Kaufmann, 1991.

[Pereira, 1982] L.M..Pereira. *ORBI- An Expert System for Environmental Resource Evaluation through Natural Language.* Informe Interno (Univ. Nova de Lisboa), 1982.

[Sobrino, 2001]. Alejandro Sobrino. *Interrogación, en lenguaje natural, de una base de datos lógica.* Departamento de Lógica y Filosofía Moral, Universidad de Santiago de Compostela. 2001.

Anexos

A1. Preguntas frecuentes recopiladas y usadas en el entrenamiento.

COMPARACIÓN

1. Cuáles municipios tienen matricula mayor que 250.
2. Mostrar las SUM con matricula menor que 120.
3. Qué SUM tienen cantidad de TV igual a 5.
4. Qué SUM tienen cantidad de profesores mayor a 80.
5. Mostrar las asignaturas con horas de conferencia mayor que 30.
6. Mostrar asignaturas con horas de conferencia igual a 160.
7. Mostrar las asignaturas con horas de clases practicas mayor que 30.

INTERROGATIVAS GENERAL

8. Qué asignaturas imparte el colectivo de informática industrial.
9. Cuáles asignaturas imparte el colectivo de sistemas digitales y aseguramiento de programas.
10. Cuáles asignaturas dirige el departamento de física.
11. Qué asignaturas dirige el departamento de contabilidad.
12. Cuáles asignaturas tiene la carrera de ingeniería química.
13. Qué asignaturas tiene la carrera de ingeniería informática por crd?.
14. Qué asignaturas tiene la carrera de ingeniería informática por cesp?.
15. Qué asignaturas tiene la carrera de ingeniería informática por cursos especiales?
16. Qué asignaturas dirige el departamento de contabilidad por crd?.
17. Cuáles asignaturas imparte el colectivo de matemática aplicada por crd?.
18. Cuáles asignaturas imparte el colectivo de matemática aplicada por cursos especiales?.
19. Cuáles asignaturas tiene la carrera de ingeniería informática en 2 semestre 1 año.
20. Cuáles asignaturas tiene la carrera de ingeniería informática en segundo año 1er semestre.
21. Qué asignaturas dirige el departamento de informática en 2 semestre 1 año.
22. Qué asignaturas dirige el departamento de informática en segundo año 1er semestre.
23. ¿ Qué asignaturas tiene la carrera de ingeniería informática en segundo año?.
24. ¿ Qué asignaturas dirige el departamento de contabilidad en 3er año?.
25. Cuáles asignaturas imparte el colectivo de matemática aplicada en 1 año?.
26. Qué departamentos pertenecen al área de ingeniería química y mecánica.
27. Qué departamentos pertenecen al área de cultura física.
28. Qué departamentos tiene la facultad de cultura física.
29. Cuáles departamentos integran la facultad de ingeniería informática.

30. Qué CES tiene el municipio de matanzas.
31. Cuáles SUM tiene la provincia de matanzas.
32. Qué carreras tiene la facultad de cultura física.

33. Cuáles carreras tiene el área de ingeniería química y mecánica.
34. Qué carreras tiene la SUM de limonar.
35. Qué centros tiene el municipio de matanzas del tipo otros.
36. Qué centros tiene la provincia de matanzas del tipo IPVCE.
37. Que matrícula tiene el municipio de Matanzas de nuevo ingreso.
38. Cuál matrícula tiene la sede de Cárdenas en la carrera de Estudio Socioculturales para nuevos ingresos de maestros emergentes.
39. Qué SUM tienen la carrera de Industrial.
40. Qué carreras tiene la SUM de Matanzas.
41. Cuál matricula tiene la provincia de matanzas en la carrera de comunicación social.
42. Qué matricula tiene la sede de Matanzas en la carrera de Contabilidad.
43. Cuál matricula tiene la sede de matanzas en la carrera de psicología para trabajadores sociales.
44. Cuál matricula tiene la provincia de matanzas en la carrera de psicología para ME.
45. Qué matricula tiene el municipio de matanzas para sexo femenino.
46. Qué matricula tiene la SUM de cárdenas para color de piel mestiza.
47. Qué matricula tiene la SUM de perico para trabajadores sociales.

ENUNCIATIVAS

48. Mostrar las asignaturas que imparte el colectivo de técnicas de programación de computadoras.
49. Mostrar las asignaturas de la carrera de ingeniería mecánica.
50. Enumerar las asignaturas que imparte el colectivo de sistemas digitales y aseguramiento de programas en 2 semestre 1 año.
51. Listar las asignaturas que imparte el colectivo de sistemas digitales y aseguramiento de programas en segundo año 1er semestre.
52. Mostrar las carreras que tiene el área de ingeniería informática.
53. Mostrar la matrícula que tiene la sede de Matanzas en la carrera de Comunicación Social para continuantes.
54. Mostrar la matrícula que tiene la provincia de Matanzas en la carrera de Comunicación Social para continuantes.
55. Mostrar la cantidad de profesores que tiene el municipio de cárdenas de profesor titular.
56. Mostrar la matricula de la provincia de matanzas para continuantes.
57. Mostrar la matricula del municipio de perico para ciencias sociales.

IMPERATIVAS

58. Muestre las asignaturas que tiene la carrera de ingeniería informática por crd.
59. Exponga los centros que tiene el municipio de matanzas.
60. Ofrezca los centros de estudio que tiene la provincia de matanzas.
61. Muestre la cantidad de profesores que tiene la SUM de cárdenas con categoría de instructor.
62. Muestra las carreras que tiene la sede de cárdenas.
63. Cita las SUM que tienen laboratorio.
64. Diga la matricula que tiene la provincia de matanzas para instructores de arte.
65. Muestre la matricula de la provincia de matanzas.
66. Diga la matricula del municipio de jovellanos para nuevos ingresos.

INTERROGATIVAS DE CANTIDAD

67. Cuántas SUM tienen laboratorios.
68. Cuántas SUM tienen cantidad de profesores igual a 60.
69. Cuántos municipios tienen matricula mayor que 250.
70. Cuántas SUM tienen matricula menor que 120.
71. Cuántas SUM tienen cantidad de videos igual a 5.
72. Cuántas asignaturas tienen horas de conferencia mayor que 30.
73. Cuántas asignaturas tienen horas de conferencia igual a 160.
74. Cuántas las asignaturas tienen horas de clases practicas mayor que 30.
75. Cuántas asignaturas tiene la carrera de ingeniería informática.
76. Cuántas asignaturas imparte el colectivo de matemática aplicada.
77. Cuántas asignaturas dirige el departamento de contabilidad.
78. Cuántas carreras tiene la facultad de ingeniería química y mecánica.
79. Cuántos CES tiene el municipio de matanzas.
80. Cuántos CES tiene la provincia de matanzas del tipo IPVCE.
81. Cuántos departamentos tiene la facultad de ingeniería informática.

INTERROGATIVAS DE LUGAR

82. Donde se ubica la sede de Matanzas.
83. Donde esta la SUM de Cárdenas.