UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

SISTEMAS DE EXTRACCIÓN DE INFORMACIÓN: SOLUCIONES INFORMÁTICAS ORGANIZACIONALES BASADAS EN DATOS NO ESTRUCTURADOS

TESIS PROFESIONAL

LISBETH MIREYA MAGAÑA LÓPEZ
CARMEN CECILIA LUZÁN HERNÁNDEZ
JOSÉ LUIS MARTÍNEZ REYES
JUAN CARLOS HERNÁNDEZ DE ANDA

MÉXICO, D.F. **2006**

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

SISTEMAS DE EXTRACCIÓN DE INFORMACIÓN: SOLUCIONES INFORMÁTICAS ORGANIZACIONALES BASADAS EN DATOS NO ESTRUCTURADOS

TESIS PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTA:

LISBETH MIREYA MAGAÑA LÓPEZ
CARMEN CECILIA LUZÁN HERNÁNDEZ
JOSÉ LUIS MARTÍNEZ REYES
JUAN CARLOS HERNÁNDEZ DE ANDA

ASESOR:
L.I. CARLOS FRANCISCO MÉNDEZ CRUZ

MÉXICO, D.F. **2006**

Integrantes
Lisbeth Mireya Magaña López
Carmen Cecilia Luzán Hernández
José Luis Martínez Reyes
Juan Carlos Hernández de Anda

Asesor
L.I. Carlos Francisco Méndez Cruz

Agradecimientos especiales al Dr. Grigori Sidorov[1] y al Grupo de Ingeniería Lingüística del Instituto de Ingeniería de la UNAM por su contribución a la realización de esta tesis.

1 Dr. Grigori Sidorov. Nació en 1965. Recibió el grado de M. en C., con especialidad en lingüística computacional, en la Universidad Estatal "Lomonósov" de Moscú, Rusia, en 1988. Recibió el título de doctor en ciencias, en la especialidad de lingüística computacional, en la Universidad Estatal "Lomonósov" de Moscú, Rusia, en 1996. Desde 1998 es Profesor Investigador titular "C", del Laboratorio de Lenguaje Natural y Procesamiento de Texto, en el Centro de Investigación en Computación del IPN. Sus áreas de interés científico son la lingüística computacional, el procesamiento automático de textos, la semántica, la sintaxis y la morfología de lenguajes naturales e inteligencia artificial. Es autor de alrededor de 40 publicaciones científicas internacionales. Pertenece al SNI.

Agradecimientos

“ Al conocer lo que Dios nos ha dado, encontraremos muchísimas cosas por las que dar gracias continuamente."

San Bernardo

Agradezco a la mejor mamá del mundo por su apoyo y ejemplo, al ser la principal responsable de todos los logros que he tenido a lo largo de la vida incluyendo esta tesis. A Laura porque siempre has sido una gran amiga. A Gris por su compresión. A todas ¡Gracias por ser mi familia!

A Daniel por tu constante apoyo e impulso, es maravilloso habernos conocido.

A todos mis familiares y amigos por estar de forma incondicional junto a mí, y por su apoyo.

Lisbeth Mireya

Agradezco con cariño a mis padres Carmen Hernández Rodríguez, Ernesto Luzán Martínez, por su apoyo, tiempo y esfuerzo brindado, y a mi hermano y abuela, Carlos E. Luzán Hernández y Cecilia Martínez, por su comprensión, a toda mi familia así como a mis amigas Marisol y Katia por ser mis amigas y estar cuando las necesito. A todos los amigos de la FCA: Alejandro M., Alfonso, Zahet, Daniela, Maura, Rosaura, Fabricio, Rebeca, Karina y Omar y todas aquellas que me faltaron, les agradezco su amistad, comprensión y confianza que tuvieron en mí, la cual siempre me hizo levantar de la adversidad. También quisiera agradecer a las instituciones que a lo largo de mi vida me apoyaron a realizar mis estudios: la Universidad Nacional Autónoma y la Secretaria de Educación Publica, así como a Dios por permitirme tener todo lo que ahora poseo y soy.

Cecilia

“Siempre define hacia donde vas, pero nunca olvides de donde vienes”

Agradezco a mis padres Clemencia y Filiberto, a todos mis familiares (incluyendo los que ya se han ido) y a todos mis amigos que he conocido desde la secundaria hasta la actualidad, así como a todos aquellos relacionados con mi vida personal, académica y laboral que me motivaron y ayudaron en la realización de esta tesis. Al programa de becas de la DGSCA-DS y a todo el personal que la forma. También agradezco a la Facultad de Contaduría y Administración de la UNAM por brindarme los conocimientos necesarios para salir adelante a través de sus profesores y de su personal técnico, administrativo y docente. Prefiero omitir nombres porque todas estas personas ya saben quienes son, y no se equivocan.

José Luis

Agradezco a DIOS por que es él quien ha estado al cuidado de mí desde siempre.

Gracias a mis padres (Juana de Anda e Inocencio Hernández) por todo lo que me han dado, sé que es más de lo que me pueda imaginar.

Gracias a mis abuelos Cirilo de Anda y Asunción Vela porque también me han dado más de lo que estoy consciente.

Gracias a mi hermosa novia Areli Mancilla por enseñarme a amar de una manera tan profunda e incondicional que no sabía existía, princesa eres el amor que nunca espere porque antes de tí no sabía que dos persona se pudieran amar así.

Gracias a mi hermano Luis Alberto por ser un gran hombre y darme un gran ejemplo, de quien siempre he recibido un respeto que no he ganado.

Gracias al ministerio Universitario de la Iglesia de Cristo en México de quien siempre encontré un apoyo y fortaleza.

Gracias especiales a: Edgar Romero y Mirta Elitania.

Juan Carlos

También agradecemos al L.I. Carlos Méndez, por ser quien nos introdujo en el mundo de la investigación y en específico en el campo de la Extracción de Información, además por la paciencia y dedicación que mostró con nosotros.

equipo de tesis

"No progresas mejorando lo que ya está hecho, sino esforzándote por lograr lo que aun queda por hacer."

Khalil Gibrán

Himno a la Universidad

Universidad Universidad
Por mi raza el espíritu hablará
Por mi raza el espíritu hablará

En el lema que adoptamos
Para nuestro laborar
El afán así expresamos:
estudiar para enseñar

Somos los educadores
Nos anima el ideal
De encender los resplandores
Del camino sin fanal

Ser para los demás
Lo suyo a todos dar
Sabiendo para prever
Previniendo para obrar

En nosotros reside el anhelo
De alcanzar la verdad y el saber
Nuestras alas presienten el vuelo
De la ciencia, el amor y el deber

Que nos guíe la voz del maestro
A alcanzar el sublime ideal
Y una mañana de luz será nuestro
De la patria diadema triunfal

Universidad Universidad
Por mi raza el espíritu hablará
Por mi raza el espíritu hablará

Romeo Manrique de Lara

«Los norteamericanos se mantienen muy firmes en su resolución de mantener pura su estirpe, pero eso depende de que tienen delante al negro, que es como el otro polo, como el contrario de los elementos que pueden mezclarse. En el mundo iberoamericano, el problema no se presenta con caracteres tan crudos; tenemos poquísimos negros y la mayor parte de ellos se han ido transformando ya en poblaciones mulatas. El indio es buen puente de mestizaje. (...) Actualmente, en parte por hipocresía y en parte porque las uniones se verifican entre personas miserables dentro de un medio desventurado, vemos con profundo horror el casamiento de una negra con un blanco; no sentiríamos repugnancia alguna si se tratara del enlace de un Apolo negro con una Venus rubia, lo que prueba que todo lo santifica la belleza. En cambio, es repugnante mirar esas parejas de casados que salen a diario de los juzgados o los templos, feas en una proporción, más o menos, del noventa por ciento de los contrayentes. El mundo está así lleno de fealdad a causa de nuestros vicios, nuestros prejuicios y nuestra miseria. (...) Los tipos bajos de la especie serán absorbidos por el tipo superior. De esta suerte podría redimirse, por ejemplo, el negro, y poco a poco, por extinción voluntaria, las estirpes más feas irán cediendo el paso a las más hermosas. Las razas inferiores, al educarse, se harían menos prolíficas, y los mejores especímenes irán ascendiendo en una escala de mejoramiento étnico, cuyo tipo máximo no es precisamente el blanco, sino esa nueva raza, a la que el mismo blanco tendrá que aspirar con el objeto de conquistar la síntesis. El indio, por medio del injerto en la raza afín, daría el salto de los millares de años que median de la Atlántida a nuestra época, y en unas cuantas décadas de eugenesia estética podría desaparecer el negro junto con los tipos que el libre instinto de hermosura vaya señalando como fundamentalmente recesivos e indignos, por lo mismo, de perpetuación. Se operaría en esta forma una selección por el gusto, mucho más eficaz que la brutal selección darwiniana, que sólo es válida, si acaso, para las especies inferiores, pero ya no para el hombre.» (José Vasconcelos, La Raza Cósmica, 1925.)

Capítulo 1

1. Introducción

La Inteligencia Artificial (IA) a grandes rasgos tiene como propósito hacer que las máquinas realicen tareas que hasta ahora sólo pueden ser efectuadas de forma óptima bajo la dirección de humanos. De la IA se desprende el Procesamiento del Lenguaje Natural (PLN), el cual se enfoca en el estudio de los mecanismos de comunicación hombre-máquina, ya que intenta simular el comportamiento lingüístico humano de una forma computacional. Y es del PLN de donde se desprenden los Sistemas de Extracción de Información (SEI), pues estos utilizan herramientas lingüísticas para hacer un análisis morfológico, léxico y sintáctico a un determinado corpus[2]. Esto sirve para seleccionar los documentos que posean los datos que son de interés, extraer y organizar la información útil.

Los Sistemas de Extracción de Información, surgen principalmente a raíz de las MUC (*Message Understanding Conferences* -véase cap. 6.2-, patrocinadas por la DARPA[3] (*Defense Advanced Research Projects Agency*) en las cuales se han definido las reglas de las tareas de extracción, dando un dominio de aplicación y proporcionando el corpus ya etiquetado (en base a herramientas de marcado, que para efectos de esta tesis se utilizará el lenguaje XML). Y es también en estas conferencias donde se realizaron los primeros sistemas de extracción de información, los cuales han mejorado los métodos utilizados y por lo tanto los resultados arrojados.

Como ya se mencionó, es necesario hacer uso de las herramientas que proporciona la Ingeniería Lingüística, como son el etiquetado, los lexicones, la lematización, el *text mining,* el reconocimiento de entidades para el análisis del corpus, para entonces caminar hacia la realización de un Sistema de Extracción de Información. Estas herramientas serán tratadas en el capítulo dos (Marco Teórico-

---

[2] Conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación.

[3] La Agencia de Investigación de Proyectos Avanzados de Defensa es una agencia del Departamento de Defensa de los E.U.A. responsable del desarrollo de nuevas tecnologías para uso militar. DARPA fue responsable de dar fondos para desarrollar muchas tecnologías que han tenido un gran impacto en el mundo, incluyendo redes de computadoras, empezando con ARPANET, que después se desarrolló como Internet.

DARPA fue creado en 1958 en respuesta al lanzamiento soviético del *Sputnik*, con la misión de mantener a la tecnología de Estados Unidos en la carrera militar por delante de la de sus enemigos.

Conceptual), para poder hacer un análisis más amplio de ellas ya que su utilización resulta primordial.

Las organizaciones comerciales y la lingüística pueden acoplarse entre sí a través de las Tecnologías de Información (TI) dando como resultado los sistemas de extracción de información, los cuales brindan una gama de herramientas o posibilidades de solución y mejora de los sistemas actuales dentro de una organización.

Hay que mencionar que la lingüística proporciona gran parte de las herramientas y métodos de análisis, aunque son propiamente la Informática y la Ingeniería Lingüística quienes se integran para poder realizar estos sistemas. La "Informática aplicada a la Lingüística" puede definirse como la aplicación de la tecnología de información para la automatización de la labor de la investigación lingüística.

Primeramente es necesario hablar del contenido del marco teórico–conceptual, en el que se abordan todos los temas considerados necesarios para poder comenzar con el estudio y fundamentar teóricamente esta tesis, tales temas son la función que se va a desempeñar de cada concepto, métodos y procedimientos de los sistemas de extracción de información dentro de los que se abordarán también los temas como la "Teoría General de Sistemas", que ayudará a definir primeramente un sistema y cuáles son los elementos que posee.

El procesamiento del lenguaje natural es un área donde surgen los sistemas de extracción de información y el cual proporciona un gran número de herramientas. El etiquetado ayuda a identificar y marcar las palabras, que resulta en un primer análisis realizado para la creación del sistema, puesto que de dicho análisis se obtienen los patrones a seguir y es a partir de aquí donde se empieza a realizar el análisis léxico y morfológico.

Por otra parte, para el análisis a realizar es necesario el apoyo en otras herramientas como los lexicones, los cuales se podrían definir como un repositorio de información léxica que incluyen información de la categoría sintáctica e interpretación semántica de las palabras. Esto es muy útil para restringir un tanto la ambigüedad. Por último el lematizador es un analizador morfológico que asocia varias palabras a una misma raíz. Todas estas herramientas mencionadas ayudan a definir y formar un sistema de extracción de información.

No es posible relacionarse con los sistemas de extracción cuando no se toman en cuenta los temas anteriores, pues para la realización de un

sistema de este tipo es necesario consultar primeramente a un experto en lingüística y a otro en el tema a tratar, entonces juntos podrán realizar un sistema óptimo.

Para la realización de un sistema de extracción de información es necesario basarse en una metodología que guíe a través de todo el proceso. Ésta consta principalmente de cuatro etapas: la primera es el proceso de tokenización, es decir segmentar el corpus, después se procede a realizar el etiquetado *POS tagger* –véase cap. 2.5.2-, entonces a partir de esto se puede realizar una construcción de expresiones regulares –véase cap. 6.3- y, por último, la extracción, organización y presentación de la información relevante.

Cabe mencionar que todo esto se debe hacer primeramente de forma manual, y después de realizarla y hacer el análisis correspondiente, se podrá realizar un sistema que lo genere de manera automática.

En todos los sistemas debe haber medidas para determinar la calidad de los resultados obtenidos. Entre las medidas que se aplican a los sistemas de extracción de información se encuentran la de *Recall* (recuperación) que indica cuánta información ha sido extraída (incluyendo aquella que no necesariamente debió haber sido extraída) y la de *precision* que indica la proporción de información correcta extraída con respecto a los resultados devueltos.

Por último pero no por ello de menor importancia, se mencionará como es que los sistemas de extracción de información que hasta el día de hoy han sido utilizados para fines de investigación, pueden proporcionar nuevas herramientas a explotar por parte de las organizaciones para una mejor toma de decisiones y el análisis de grandes volúmenes de información de una manera automática.

Pensando por ejemplo en dos supuestos; el primero de una empresa que se dedica al análisis publicitario de otras empresas y es contratada por una segunda para saber qué tanta publicidad puede realizar la competencia y en que medios se dirige, en este tipo de empresas publicitarias es necesario el análisis de grandes volúmenes de información además de otro gran número de empleados para realizarlo. El segundo supuesto consiste de un gerente en tecnología de información que necesita estar al día en la publicación de nuevo *software*, y por lo mismo necesita estar pendiente de las noticias diarias de estas nuevas publicaciones. Pensando desde el punto de vista de los sistemas de extracción de información, el análisis se haría de una forma automática y por ende implica reducción de tiempos. Hay que

mencionar que los mejores sistemas de extracción de información hasta ahora realizados por expertos en el tema no han logrado más del 75% de *precision* y *recall* -véase cap. 6.2-, y aunque se sabe que los resultados esperados en esta tesis no serán al 100%, se plantea ahorrarle al usuario entre un 75% y un 90% del trabajo.

Con el fin de comprobar la hipótesis de esta tesis, se ha decidido que a la par de realizar la misma se irá desarrollando una aplicación de un sistema de extracción de información, esto enriquecerá las aportaciones a la tesis. La aplicación ha sido nombrada "Sistema de Extracción de Información de Nuevo *Software*" (SEINS), toma primordialmente el segundo supuesto antes mencionado (productos de *software*) y cuando se haga referencia a esta aplicación a lo largo de la tesis se hará como "SEINS", aunque no es meramente un sistema de información sino una aplicación de un sistema de extracción de información. Tiene como finalidad la extracción de información de un corpus de noticias donde la búsqueda a extraer es el nombre de un nuevo *software*, el nombre de la compañía que lo desarrolla y la versión del mismo en caso de que la tenga. Esperando con esta aplicación dar una muestra de la gran gama de aplicaciones que puede dársele a los sistemas de extracción de información dentro de las organizaciones

El material bibliográfico que hasta ahora ha sido utilizado para fundamentar la parte teórica de la tesis, es en su mayoría proveniente de páginas *web* debido a que la mayor parte de la información actualizada sobre el tema se encuentra en internet.

Se buscó también información en libros muy importantes, sobre todo relacionados a la Ingeniería Lingüística donde se hallaron muy buenas exposiciones, pero la mayor parte de la información útil para la tesis provino de artículos publicados por investigadores (algunos escritos en inglés), sitios *web* y revistas publicadas en internet.

Capítulo 2

2. Marco teórico-conceptual

2.1. Sistemas de información

2.1.1. Teoría de sistemas

A principios de los años veinte, Ludwig von Bertalanffy[4] en un intento por unificar las ciencias sociales y naturales, funda una teoría general que se aplica a todos los sistemas, lo que hoy se conoce como la "Teoría General de Sistemas", y es por esa razón que se le conoce como padre de ésta. De la teoría general de sistemas se desprenden varios conceptos generales.

2.1.1.1. Conceptos generales

Se comienza por definir un sistema, ya que es fundamental para poder dar sentido a las demás definiciones. Un sistema esta formado por un conjunto de partes, elementos u objetos relacionados entre sí, que tienen objetivos o metas en común; implica integridad, totalidad y unificación de partes para lograr un funcionamiento óptimo de un conjunto de componentes que a continuación se definirán: (De la Reza 2001).

- Elementos: los elementos son los componentes de un sistema que son identificables, mismos que a su vez pueden formar un sistema propio.
- Proceso: es el resultado neto de todas las actividades que convierten los elementos de entrada en elementos de salida y generalmente se les agrega un valor o utilidad.
- Entradas: elementos sobre los cuales se aplican los recursos.
- Salida: son los resultados del proceso de conversión del sistema.
- Medio ambiente: el medio ambiente es aquel donde los sistemas tienen sus interacciones. Para diferenciar los elementos que forman parte de un sistema y los que sólo son parte del medio ambiente, es necesario delimitar las fronteras del sistema.
- Propósito o función: los sistemas adquieren un propósito o función cuando entran en relación con otros subsistemas en un contexto de un sistema más grande.
- Atributos: los atributos en un sistema son propiedades o características y pueden ser de calidad o de cantidad.
- Objetivo: es el fin al que desea llegar el sistema. Las acciones de un sistema se realizan con la idea de conseguir ese fin.
- Administración: las acciones y decisiones que tienen lugar en el sistema se atribuyen o asignan a administradores cuya

4 Biólogo austriaco, uno de los fundadores de este nuevo enfoque o paradigma, en la década de los 20.

responsabilidad es la guía del sistema hacia el logro de sus objetivos.

- Estructura: la estructura de un sistema está formada por las relaciones entre los objetos y atributos de los objetos del mismo. Puede ser simple o compleja y esto depende del número de relaciones entre los objetos del sistema, estas relaciones pueden ser disfuncionales, parasitarias, simbióticas, sinérgicas u optimizadas.
- Estado: se define por las propiedades que muestran sus elementos en un punto en el tiempo.
- Interfaz: es una conexión entre dos sistemas, la región de contacto.
- Entropía: es el movimiento de un sistema hacia un desgaste, desorden o discrepancia. Un sistema cerrado alcanza su entropía cuando se descompone (Murdick 1988:45).
- Homeostasis: es la característica de un sistema abierto para regresar a un estado estable.
- Equifinalidad: "establece que un sistema abierto debe comenzar de cualquiera de los estados iniciales y seguir alguna trayectoria para seguir una finalidad" (Murdick 1988:46).
- Multifinalidad: "implica que existen varios estados finales de modo que la elección de los medios descansa sobre las razones de llegar a un resultado" (Murdick 1988:46).

2.1.1.2. Clasificación de los sistemas

Es posible encontrar sistemas en nuestro alrededor, como los ecosistemas, o incluso dentro de nosotros mismos, como el sistema nervioso, el inmunológico, etc. Se pueden encontrar en cualquier parte y debido a esto es necesario clasificarlos. Para Robert G. Murdick (1988:37) los sistemas se clasifican en:

- Naturales: como su nombre lo dice son los creados por la naturaleza, ejemplos de éste son el ecosistema o el sistema solar.
- Artificiales: este tipo de sistemas los ha creado el hombre, ejemplos de estos son el sistema de comunicaciones o el de transporte.
- Sociales: son los integrados por personas.
- Hombre-Máquina: en estos sistemas los hombres emplean equipos en sus trabajos organizados.
- Mecánicos: estos sistemas deben tener sus propias entradas y mantenerlas, deben ser autosuficientes.
- Abiertos: estos sistemas son los que interactúan con su medio ambiente.

- Cerrados: en un sistema cerrado el ambiente no cambia, se levanta una barrera entre el sistema y el ambiente para impedir cualquier influencia, por ejemplo un vaso térmico.
- Permanentes: aquellos que duran mucho más que las operaciones que en ellos realiza el ser humano, ejemplo de éste es el sistema económico.
- Temporales: están destinados a durar cierto periodo y luego de éste desaparecen.
- Estables: sus propiedades y operaciones no varían de manera importante o lo hacen sólo en ciclos repetitivos.
- No estables: no tienen ciclos repetitivos.
- Subsistemas: son sistemas más pequeños incorporados al sistema principal.
- Suprasistemas: estos denotan sistemas extremadamente grandes y complejos.
- Adaptativos: se presentan cuando un sistema reacciona con su medio ambiente de manera que mejora su funcionamiento, logro o probabilidad de supervivencia.

2.1.1.3. Aspectos básicos de la teoría general de sistemas

Algunos aspectos básicos de la teoría de Bertalanffy son:

- Los componentes de un sistema están interrelacionados y son interdependientes.
- Se ve el sistema como un todo; no es necesario individualizarlo en cada una de sus partes ya que se puede perder de vista el conjunto del sistema.
- Todos los sistemas tienen una meta o un objetivo final.
- Los sistemas tienen entradas y salidas y estas a su vez pueden ser entradas de otros sistemas.
- Todos los sistemas transforman sus elementos de entrada en elementos de salida.
- Hay algunos sistemas cerrados que no tienen entradas, por lo tanto no tienen salidas y su supervivencia es crítica.
- El sistema debe tener alguna función o medio para regular la interacción de sus componentes, de tal modo que alcancen sus objetivos.
- Los sistemas pueden estar compuestos por sistemas más pequeños y esto forma una jerarquía de sistemas.
- En los sistemas aparecen las diferencias de tareas, entonces hay unidades especializadas que realizan tareas especializadas.
- Un sistema tiene un fin y este puede alcanzarse por diversos caminos y desde varios puntos de partida.

2.1.2. Informática

La era de la computación nace a mediados del siglo XX, cuando las ideas y modelos matemáticos hasta entonces desarrollados al fin lograron hacerse realidad mediante complejos aparatos de ingeniería electrónica. Las ciencias y técnicas de la computación por un lado y su utilización por el otro aprendieron a coexistir, dando lugar al nuevo concepto de *Electronic Data Processing* (procesamiento electrónico de datos) hacia finales de la década de 1960.

Para comprender el uso de este término en los países latinoamericanos es necesario tomar en cuenta una complicación posterior: como en inglés se usa el nombre *Computer Science*, la traducción obvia al español es "Ciencias de la Computación", concepto que en francés se conoce como "*Informatique"*. Sin embargo, estando España más cerca de Francia que de Estados Unidos, allá se le conoce como Informática. Aunque nuestro país está demasiado cerca de la influencia cultural y técnica del vecino país del norte, tampoco se puede ignorar la tradición ibérica, y la resultante, hacia la década de 1980, era un término criollo de significado equívoco: "Informática" (Levine 2001:2). En los países de habla hispana se reconoció en 1968.

La informática es el tratamiento automático de la información y el término se creó en Francia en 1962. Se dio a conocer en los países de habla hispana cerca de 1968. A continuación se dan dos definiciones de ésta (Prieto 2002:1).

- Palabra de origen francés formada por la contracción de los vocablos "información" y "automática", la Real Academia Española la define como el conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento automático de la información por medio de computadoras.
- La dada por la academia francesa es "*Informatique",* ciencia del tratamiento racional y automático de la información, considerándose ésta como soporte de los conocimientos de las comunicaciones en los campos técnico, económico y social.

El término informática se concentró desde sus inicios en la denominación de técnicas de proceso automatizado de información económica en países como España, Francia y América latina; mientras que en otros como la antigua URSS, se refería a los sistemas automatizados de búsqueda y recuperación de información en centros de documentación y bibliotecas.

La misión de la informática es detectar y satisfacer las necesidades organizacionales relativas al uso y empleo de la información. Debe

recabar y organizar los datos y procesos necesarios para el buen funcionamiento de la organización y el cumplimiento de sus objetivos. El resultado final será la creación, administración o mantenimiento de servicios y sistemas de tratamiento de información integrados y eficientes.

2.1.2.1. Problemática actual

La problemática actual de la informática es tan amplia como los quehaceres humanos que la utilizan como herramienta. La computación ha invadido la vida del hombre actual y ahora difícilmente se encuentra un área donde la computadora no juegue en algún momento un rol de participación; desde el entretenimiento, hasta el deporte, cruzando por las artes y la medicina, la informática envuelve cada vez más la vida cotidiana de las personas.

Los retos actuales son incontables:

1. Agilizar las telecomunicaciones.
2. Mejoras en la cuestión de seguridad del procesamiento de datos (antivirus, cortafuegos, *antispam*, etc.).
3. *Hardware* cada vez más rápido.
4. Aumentar la capacidad de almacenamiento.
5. Bajar los precios.
6. Profundizar en el estudio de la Inteligencia Artificial.
7. Desarrollar nuevos lenguajes de programación que permitan crear aplicaciones de manera mucho más rápida y sencilla.
8. Mejora de las interfaces de usuario.
9. Desarrollo de nuevos periféricos para gente con capacidades especiales (débiles visuales, personas con alguna obstrucción en alguna extremidad o que carezcan de ella).
10. Estabilidad en los sistemas operativos.
11. Bases de datos más potentes, rápidas y seguras.

Y esto sólo por citar algunas necesidades actuales.

Su aplicación en la vida cotidiana se puede encontrar especialmente en aparatos de diversión casera como DVD, mp3, videojuegos, etc. aunque su incursión es gradualmente mayor al grado de que en Japón se lanzó recientemente al mercado un microondas parlante que ofrece consejos de uso y recetas para el usuario.

Las tecnologías de informática pueden introducir nuevos estilos de competencia entre las empresas. Algunos de los principales cambios son los siguientes:

1. Nuevos productos
La tecnología crea productos de más rápida operación, precio más bajo y mayor calidad, o definitivamente nuevos. Incluso pueden crearse nuevos productos a la medida de las necesidades del cliente.

2. Nuevas empresas
Las tecnologías de información pueden crear por sí mismas empresas enteramente nuevas. Una compañía puede emplear su capacidad excedente de sistemas de información en el desarrollo de nuevos servicios para sus clientes fuera de su área directa de servicio.

3. Nuevas relaciones entre clientes y proveedores
Las empresas que ponen a disposición general sus sistemas de información pueden conseguir con mayor probabilidad que sus clientes sigan realizando actividades comerciales con ellas en lugar de optar por la competencia.

En la actualidad se puede ubicar a la informática dentro de los procesos de cualquier organización escolar, militar, sanitaria, social, etc., que maneje su información a través de computadoras.

Aunado a esto, el procesamiento de la información es el medio por el cual un conjunto de datos es presentado con determinadas características o atributos al receptor o destinatario. Se debe entender que después de la recepción de los datos que contienen información irrelevante; estos siguen una serie de pasos (proceso) para obtener la información, con el contenido que el individuo necesita o desea.

2.1.2.2. Requerimientos de procesamiento de datos

Los requerimientos para el procesamiento de datos se refieren al trabajo de detalle de un sistema y tiene los siguientes componentes (Burch 1994:68):

- Volumen: se refiere al volumen de los datos involucrados que deben procesarse en un tiempo determinado para lograr la meta de información. Para lograr obtener éste, es necesario contabilizar las transacciones organizacionales de la empresa.
- Complejidad: se refiere al número de operaciones de datos complicadas e interrelacionadas que se deben realizar para lograr una meta de información.
- Restricciones de tiempo: se define como la cantidad de tiempo aceptable entre el momento en que los datos están disponibles y el momento en que la información se requiere.

- Demandas computacionales: son una combinación de volumen, complejidad y restricciones de tiempo, para un requerimiento específico de información. Estas demandas computacionales pueden ser complejas si se requiere procesar un modelo grande de programación lineal o se debe dar mantenimiento en línea a una base de datos grande.

2.1.2.3. Catálogo de áreas de conocimiento

Según Levine[5] (2001:5), se definen ocho grande áreas de conocimiento en informática, que luego se subdividen en subáreas, y éstas en subsubáreas. Las subáreas y subsubáreas, según los casos, están estructuradas en grupos de temas de estudio.

Las áreas generales del conocimiento son:

1. Entorno Social.- Comprende conocimientos, normas, experiencias y motivaciones que hacen posible la buena integración de las unidades de informática y su personal en las organizaciones y en la sociedad en general. Se incluyen tópicos de administración, economía, contabilidad, derecho, sociología y psicología.

2. Matemáticas.- Brindan una excelente e imprescindible base de tipo formativo para el desarrollo de habilidades de abstracción y la expresión de formalismos, además de proporcionar conocimientos específicos fundamentales para la informática.

3. Arquitectura de computadoras.- Estudio de la teoría, técnicas y métodos para comprender el funcionamiento de los sistemas digitales y las computadoras, así como los principios físicos que los sustentan, con el objeto de formular algunas de sus especificaciones y saber integrar equipos diversos para fines particulares.

4. Redes.- Estudio de la fusión de los dominios tradicionalmente considerados como *hardware* y *software*, y las formas de distribuir y compartir recursos computacionales, procesos e información.

5. *Software* de base.- Estudio, definición y construcción de las piezas de *software* que hacen posible el funcionamiento de las computadoras en diferentes niveles operativos. Por su importancia formativa y metodológica, esta área de conocimiento resulta

5 Guillermo Levine Gutiérrez es miembro fundador del CONAIC (Consejo Nacional de Acreditación en Informática y Computación) y de la Asociación Nacional de Instituciones de Educación en Informática en México.

fundamental para el desarrollo de la industria de los programas para computadora.

6. Programación e ingeniería de *software*.- Cuerpo de conocimientos teóricos y prácticos, y conjunto de metodologías para la buena construcción de programas y sistemas de *software*, considerando su análisis y diseño, confiabilidad, funcionalidad, costo, seguridad, facilidades de mantenimiento y otros aspectos relacionados.

7. Tratamiento de información.- Tiene como objetivo presentar las diferentes filosofías, conceptos, metodologías y técnicas utilizadas para la construcción de sistemas grandes de *software*, considerando su análisis, especificaciones, diseño, programación, documentación, verificación y evaluación. Brindar elementos para lograr diseños modulares y susceptibles de ser realizados por grupos de desarrollo.

8. Interacción humano-máquina.- Es el estudio de los dominios de aplicación conducentes a lograr formas superiores de expresión e interacción entre el hombre y la computadora, con el fin de buscar mejores y novedosas maneras de integración de la tecnología en la sociedad.

2.1.3. Sistemas de información

2.1.3.1. Información

La información es un conjunto de datos en un contexto, dentro del cual obtiene un significado y de esta manera es útil para la toma de decisiones del receptor. La información debe ser emitida, procesada y recibida para ser evaluada y así sorprender y estimular la acción de aquellos quienes la reciben (Burch 1994:19-23).

2.1.3.2. Historia de la necesidad de información

La necesidad de información data desde épocas y culturas tan antiguas como el año 4500 a.C., pues en estas civilizaciones se presentan registros en tabletas de arcilla de varias formas y tamaños, estos aditamentos eran importantes porque satisfacían las necesidades de información de ingresos, desembolsos, inventarios, compras, arrendamientos, formación y disolución de sociedades y contratos (Burch 1994:21).

Por otro lado, hace más de 500 años los incas en Sudamérica, desarrollaron sistemas de bases de datos y modelos de procesamientos compuestos de miles de cuerdas con nudos denominadas *quipus* -véase figura 2.1-. Es impresionante como los nudos de cuerdas colgantes representaban el número de personas de un poblado, sus deberes, la cantidad de grano en un almacén, transacciones comerciales, poesía, registros de batallas, etc. Las personas especializadas en construir este tipo de sistemas se llamaban *quipuamayus* y estudiaban en casas de enseñanza, lo que en la actualidad comúnmente son llamadas escuelas.

Figura 2.1.- Imagen de un *quipu* (imagen tomada de web.80).

A mediados del siglo XVIII, la Revolución Industrial reemplazó al hogar y el taller como los medios básicos de la producción por fábricas. Dichas fábricas acrecentaron su tamaño y complejidad de tal forma que era imposible la obtención de información para el control de las operaciones sin la ayuda del procesamiento de datos. La aparición de los sistemas fabriles y las técnicas de producción masiva provocaron la necesidad de mayor capital, ocasionando con esto la separación del inversionista (dueño), con la gerencia (administrador). Por lo tanto la gerencia necesitaba mayor información para las decisiones internas y los inversionistas necesitaban información acerca de la organización y del desempeño de la gerencia.

2.1.3.3. La necesidad de la información en la actualidad (Siglo XX)

Con el crecimiento de los usuarios en determinadas organizaciones se generó una mayor necesidad de información. Por ejemplo los inversionistas necesitan información acerca de su estado financiero para planear transacciones futuras. Los banqueros necesitan información para analizar el desempeño y solidez de un negocio antes de realizar un

préstamo o conceder un crédito. En el gobierno se necesitan reportes financieros y operativos de los impuestos y reglamentaciones. Los sindicatos quieren saber las utilidades de las organizaciones a las que están afiliados sus trabajadores. Por otro lado las personas que también dependen de la información son quienes administran y operan las organizaciones, es decir, la gerencia y los empleados.

2.1.3.4. Atributos de la información

Para que la información sea de calidad debe tener las siguientes cualidades:

- Exactitud: la información no debe tener errores, debe ser clara y tener bases sólidas.
- Oportunidad: significa que la información debe estar lista cuando se requiere.
- Relevancia: es decir, proporcionar la información específica para quién la necesita.

De acuerdo a Bertalanffy "Un sistema es modelo de la naturaleza general, esto es, una representación conceptual de ciertos caracteres mas bien universales de entidades observadas. El uso de modelos y construcciones representativas constituye el método general de la ciencia [...] Un sistema puede definirse como un conjunto de elementos relacionados entre sí y con el medio ambiente".

Se entiende por sistema de información el medio por el cual los datos fluyen de una persona o entidad hacia otra, es decir, existe una entrada, un procesamiento de información y una salida. Por ejemplo:

- Comunicación interna.
- Líneas telefónicas.
- Sistemas de cómputo (web.01).

Un sistema de información puede surgir a partir de varios sistemas, por ejemplo en una organización existen varios sistemas como son: ventas, finanzas, mercadotecnia etc (Senn 1992:30).

Algunos autores han planteado 3 niveles en los que deben operar los sistemas de información: el asociado a las actividades operativas, el que está ligado a los aspectos tácticos y el de apoyo a las funciones estratégicas.

Desde la perspectiva operativa se requieren datos e información detallados y actualizados sobre el estado de los recursos, el avance de

las operaciones, las características de la organización y de su ambiente, etc.

Las necesidades de información para el nivel táctico pueden satisfacerse con sistemas rutinarios, aunque pueden presentarse casos en los que se requiera generar reportes e informes especiales.

La información requerida para las funciones estratégicas contiene un mayor componente de aspectos externos, cubre áreas más amplias de la organización y generalmente no requiere ser muy detallada ni actualizada. Con frecuencia se requieren pronósticos y previsiones, así como estimaciones para los cuales es difícil obtener bases sólidas.

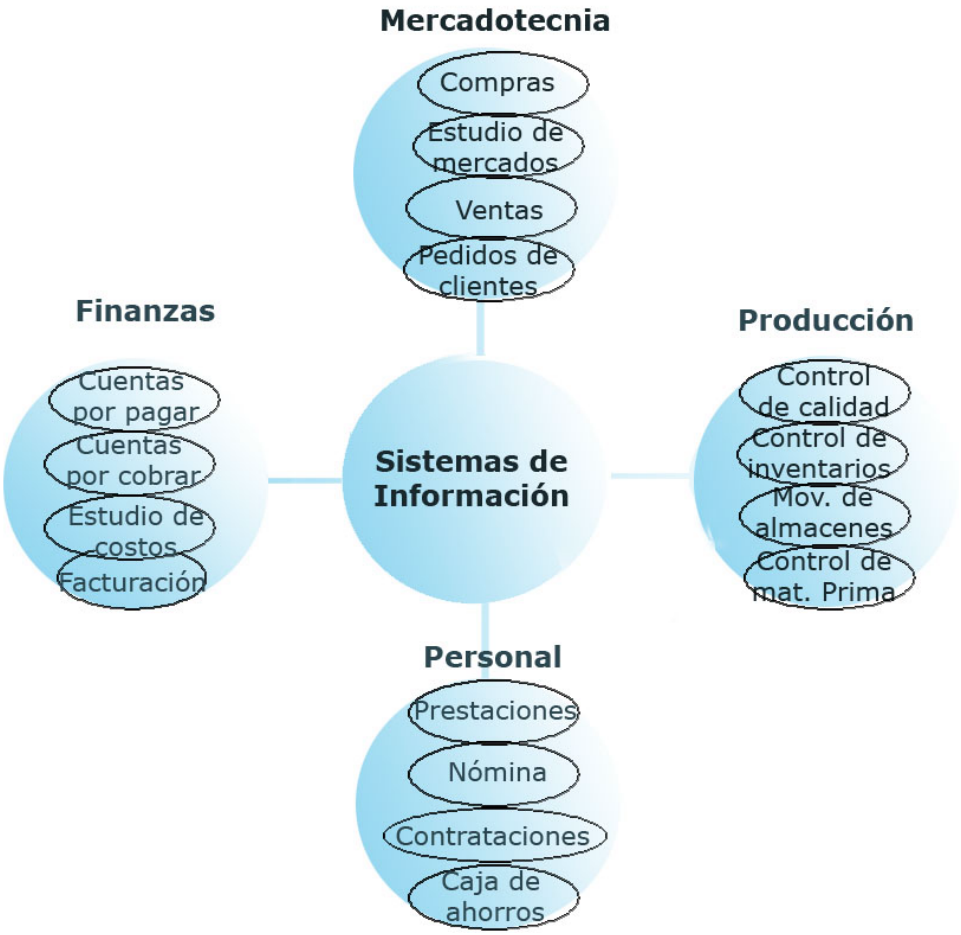


Figura 2.2.- Sistema de información visto como un sistema total.

2.1.3.5. Algunos tipos de sistemas de información

Existen tantos tipos de sistemas de información como ramas de la actividad humana donde interviene la computadora, haciendo énfasis en el área administrativa se encuentran cuatro grandes rubros en los que es posible dividir los sistemas informáticos:

- **Sistemas de procesamiento de transacciones**

Estos sistemas almacenan datos sobre el funcionamiento cotidiano de la empresa, por ejemplo la venta de productos en una empresa, si fuese una escuela un elemento importante de sus transacciones sería el registro de las calificaciones, inscripciones de los alumnos, etc.

- **Sistemas de información gerencial**

Los sistemas gerenciales son hechos para dar respuesta a preguntas como la cantidad de ventas realizadas en el mes actual o el anterior, cuál fue la razón de que se vendiera en mayor cantidad en un mes que en otro o si los precios son los más óptimos de manera rápida y exacta. Soluciones así también podrían ser obtenidas manualmente pero el tiempo y costo serían excesivos como para poder ser una opción real, además de que generalmente es necesario hacer estos procesos de manera periódica.

- **Sistema de apoyo para la decisión**

Estos sistemas también están desarrollados para la toma de decisiones y sus usuarios son los altos ejecutivos. A diferencia de los sistemas de información gerencial estos no son hechos para resolver cuestiones que se presenten de manera periódica, sino de una controversia única que además tiene carácter de presentar los datos y la información de manera poco ordenada. Estos sistemas son hechos para ser usados una sola vez, pues la naturaleza del problema que resuelve es tan específica que difícilmente se presentará otro caso igual donde pueda ser usado el sistema.

- **Sistema de información para oficinas**

Son sistemas usados por oficinistas en labores cotidianas como el procesamiento de texto, el manejo de voz o imagen a través de telecomunicaciones, etc. funciones así permiten mejorar la comunicación al interior de la empresa.

	Usuario	Uso	Propósito
Sistemas de procesamiento de transacciones	Operativos (oficinistas o relacionadas directamente con el cliente)	Muy frecuente	Realizar de manera más rápida las labores cotidianas de la empresa
Sistemas de información gerencial	Alta dirección	Periódicamente (mensual, semestral, etc.)	Tomar decisiones
Sistema de apoyo para la decisión	Alta dirección	Una sola vez	Tomar decisiones
Sistema de información para oficinas	Operativos de oficina	Frecuente	Realizar labores cotidianas de la oficina

Tabla 2.1.- Muestra comparativa de las características de los sistemas de información.

2.1.3.6. Componentes estructurales de los sistemas de información

Los sistemas están formados por componentes estructurales que se describirán a continuación (Burch 1994:58-62):

- Entrada

Es decir, todos los datos, como texto voz e imágenes, así como los medios y procedimientos utilizados para introducirse. Algunas de las entradas son: transacciones, solicitudes, consultas, instrucciones, mensajes, noticias etc.

- Modelos

Constan de formas de realizar operaciones lógicas y matemáticas para manipular las formas de entrada de datos almacenados y lograr los resultados esperados. Un modelo lógico-matemático combina datos para proporcionar una salida adecuada para una consulta, esta operación puede reducir o aumentar el volumen de los datos para obtener un reporte concreto.

- Salida

Se refiere a los productos, es decir, a la información de calidad resultante así como a los documentos necesarios para todos los niveles de la organización, desde la gerencia o incluyendo cualquier usuario tanto fuera como dentro de la misma. De la salida dependen muchos otros componentes de la organización, pero si este resultado no satisface las necesidades para las que fue creado, no sirve de nada. La salida en muchos casos puede ser la entrada para otro proceso.

- Tecnología

La tecnología es lo que realiza la unión de los complejos estructurales de la organización, es decir son las instrucciones, conocimientos, procedimientos y métodos aplicados a controlar todo el sistema.

- Bases de datos

El lugar donde se almacenan todos los datos importantes, para atender las necesidades de los usuarios -véase cap. 2.3-.

- Controles

Los sistemas están en constante peligro junto con amenazas diversas, como desastres naturales, incendios, fraudes, fallas de los sistemas, errores u omisiones. Por lo tanto es necesario un control eficiente en los registros para asegurar la integridad, además la creación de un plan de contingencias, una aplicación para los procedimientos del personal, asignación de tareas y capacitación, para asegurar buenos resultados.

Es importante estudiar estos componentes debido a que la unión de estos forma sistemas de información funcionales para satisfacer las necesidades de los usuarios.

2.1.4. Sistemas informáticos

Los sistemas informáticos son parte integral de las organizaciones, por lo que constituyen subsistemas de ellas.

Por lo tanto se puede inferir que un sistema informático es un conjunto de elementos o sistemas que utilizan la computadora para procesar información o datos para obtener objetivos específicos. Estos resultados específicos pueden ser reportes de ventas o estados financieros de una empresa (Senn 1992:31).

Los elementos componentes de un sistema informático son:

- Computadoras.
- Medios de programación (*software*), que pueden comprender: sistema operativo, programas de comunicaciones, *software* de aplicación, etc.
- Instrucciones destinadas al operador, al usuario y al proveedor de la información del sistema; las cuales tienen como objetivo reglamentar y asegurar la actividad del sistema en su conjunto.
- Información almacenada en las bases de datos.

- Funcionarios, especialistas o trabajadores en general, encargados de emitir informaciones o de utilizar la que resulta de la actividad del sistema informático.
- Sensores y captadores de información en máquinas o procesos productivos, comerciales, etc. Pueden no estar presentes en determinado sistema informático.
- Líneas y equipos de comunicaciones y enlaces entre computadoras.
- Dispositivos de almacenamiento.
- Documentos o formularios para captar la información de entrada al sistema y para reflejar la información de resultados.
- Equipos auxiliares como separadores de formas, calculadoras, fotocopiadoras, etc.

De esta manera se concluye, con respecto a los sistemas informáticos, que estos también son forzosamente sistemas de información ya que se dedican al procesamiento de la misma, pero un sistema de información no necesariamente es un sistema informático.

2.1.4.1. La información como un arma competitiva

En la actualidad, las organizaciones operan en un mundo de desaciertos gubernamentales, políticas impredecibles a nivel monetario, fiscal, impositivo y regulador, así como de ciclos de negocio como son los cambios de políticas comerciales, de competencia (tanto doméstica e internacional), de disfunciones sociales, de cambio en el mercado y de crecientes costos laborales. A todos estos problemas se enfrentan las organizaciones y, para sobrevivir y desarrollarse, se deben explotar las dimensiones de oportunidad de una gerencia informada de la diversificación de productos y servicios crecientes y de una creciente productividad (Burch 1994:34).

Por toda la problemática citada anteriormente, se ve claramente que la información es el pilar de ayuda a la gerencia para tomar decisiones que sean benéficas para la organización y justamente ésta encontrará un alto nivel competitivo. Por otro lado si una compañía no puede tener un mejor manejo de la información, quedará atrás de las que sí pueden.

Además, los sistemas de información son importantes para obtener la información a tiempo para la toma de decisiones, hay ocasiones en que la información requerida se encuentra en gran cantidad de textos, como en los textos de currículos de todos los aspirantes a un puesto en una organización, de los cuáles sólo se requieren partes concretas de esos currículos, como son: el nombre, el teléfono y la experiencia; ésta

obtención de la información se resuelve mediante sistemas de extracción de información, pues estos sistemas sólo darán como resultado información específica de los textos examinados.

2.1.5. Metodologías para el desarrollo de sistemas

La Real Academia Española define "metodología" como un conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal. Siguiendo con la misma fuente de datos, un "método" es un procedimiento que se sigue en las ciencias para hallar la verdad y enseñarla.

Con lo anterior se infiere que para resolver determinado problema se utiliza un conjunto de diversos métodos o procedimientos (metodología).

El término "desarrollo de sistemas" engloba a todo el proceso para crear un sistema informático, desde la metodología utilizada, las herramientas de apoyo (libros, internet, revistas, etc.), el material humano y técnico, los recursos de *software* que se utilizaron hasta las técnicas[6] empleadas.

La metodología se compone de los diagramas utilizados, el tipo de modelado y la forma de resolver el problema.

Existen varios tipos de metodologías para el diseño y desarrollo de sistemas, pero todas estas se basan en uno de los dos grandes modelos que existen, que son: el Modelo estructurado y el Modelo orientado a objetos.

Es decir, un desarrollador de sistemas debe crear una aplicación a través de una metodología (la forma en que va a crear el sistema y las herramientas que utilizará), esta metodología se rige en función de un modelo de desarrollo de sistemas (aunque puede tomar puntos de ambos modelos).

Las metodologías se desarrollaron para lograr un mejor diseño de acuerdo a las necesidades de los usuarios, además de hacerlo más rápido y eficiente. Estas cuentan con fases o elementos los cuales son: análisis, diseño, pruebas e implementación. Como ya se mencionó, la

6 Una técnica es un método que aplica herramientas y reglas específicas para completar una o más fases del ciclo de vida del desarrollo de sistemas. Uno de sus sinónimos más habituales es paradigma.

metodología es el estudio del método, así que dependiendo de éste será la forma en la que se utilicen estas fases o elementos, que a continuación se definen.

Análisis: en esta etapa del análisis se contestan preguntas tales como: ¿quién usará el sistema?, ¿quién deberá desarrollar el sistema?, ¿dónde y cuándo deberá usarse?, ¿qué problemática pretende resolver?. Además de realizar una planeación del proceso que se sigue para entender por qué debe ser construido el sistema, cómo y qué pretende resolver, las herramientas que se utilizarán, lo que se debe hacer para construirlo y cuánto tiempo llevará la realización del mismo.

Diseño: en esta fase se decide como deberá operar el sistema, en términos de *hardware* y *software*; cuáles serán las interfaces de usuarios y reportes, además de las especificaciones de programación. Es en esta fase donde se determina exactamente cómo debe operar el sistema.

Pruebas: en esta fase se debe revisar el sistema para verificar su operación y funcionalidad de acuerdo a las especificaciones definidas en las etapas de análisis y diseño, además de tratar de buscar todas las formas no comunes en que se puede utilizar el sistema para verificar que no hay resultados inesperados. Todo esto hace que el sistema sea confiable.

Implementación: el sistema es construido y debe estar funcionando. Requiere de mucha atención porque suele ser la más costosa de todas las fases, debido a que a veces no es posible dejar funcionando el sistema y entonces hay que regresar y hacer los cambios pertinentes.

La fase de mantenimiento no es incluida ya que se considera que cuando un sistema está terminado debe estar bien desarrollado, no debe haber algo que corregir y por lo tanto no existe el mantenimiento, puesto que hay que prevenir que el sistema siempre funcione bien. Aunque podría haber una fase de Actualización que consiste en ajustar el sistema desarrollado a las necesidades cambiantes de la organización

2.1.5.1. Metodologías basadas en el modelo estructurado

Las técnicas estructuradas son métodos formales de división de un problema de empresa en fragmentos y relaciones manejables, y la ulterior reunión de estos fragmentos y relaciones (posiblemente con añadidos y eliminaciones) en una solución informática de empresa útil

para resolver el problema. Uno de sus sinónimos es "métodos estructurados".

Este tipo de metodologías se basan en la construcción de modelos utilizando técnicas estructuradas como lo son el flujo de datos y el diagrama entidad relación. Comenzaron a surgir a finales de la década de los setenta de la mano con la programación estructurada. Dentro de las metodologías basadas en el análisis estructurado se encuentran las que a continuación se mencionarán.

En cierto sentido, como muchas veces lo enseñan en la educación media superior, las técnicas estructuradas utilizan el método de "divide y vencerás" para resolver problemas relacionados con el desarrollo de *software* y sistemas. Algunas técnicas estructuradas son:

- Programación estructurada.
- Los métodos Yourdon (DeMarco y Ward/Mellor).
- El Análisis Estructurado y Técnica de Diseño (SADT, *Structured Análisis Design Technique*, éste ha sido utilizado exitosamente por la Agencia Espacial Europea durante algún tiempo).
- El Análisis Estructurado de Sistemas y la Metodología de Diseño (SSADM *Structured Systems Analysis and Design Methodology*, éste es recomendado por el gobierno del Reino Unido para los sistemas de procesamiento de datos).
- Modelización de datos.
- Ingeniería de información.
- Método de HIPO.
- Ciclo de vida de desarrollo de un sistema.
- Metodología por prototipos (aunque ésta también contiene elementos del modelo orientado a objetos).

Se explicarán algunas metodologías del enfoque estructurado.

1. Método del ciclo de vida de desarrollo de sistemas

El propósito del ciclo de vida es planear, ejecutar y controlar el proyecto de desarrollo de un sistema. El ciclo de vida define las fases y las tareas esenciales para el desarrollo de sistemas, sin importar el tipo o la envergadura del sistema que se planea construir.

En esta metodología se realizan las etapas de "Investigación preliminar, determinación de requerimientos, diseño del sistema, desarrollo de *software*, prueba del sistema e implantación" (Senn 1992:32-37).

Estas etapas determinan los requerimientos del sistema, la ubicación lógica de los datos (archivos o bases de datos), la cantidad de transacciones y procesamiento que se realizará, la validación de los datos de entrada o salida, los departamentos que están involucrados, el tiempo en que se desarrollará el sistema y la planeación por grupos de acuerdo al proyecto.

Investigación preliminar

Inicia a partir de una solicitud para solucionar un problema, y tiene tres etapas a seguir:

a) **Aclaración de la solicitud:** debido a que algunas solicitudes por escrito no son claras, se debe examinar lo que realmente quiere el usuario. Para tener la idea concreta de lo que se espera del sistema.
b) **Estudio de factibilidad:** es decir, si se puede realizar, éste estudio se divide en factibilidad técnica, económica y operacional. La factibilidad técnica consiste en determinar si se cuentan con los recursos tecnológicos o si se tienen que adquirir. La factibilidad económica verifica si se tienen los recursos para costear el proyecto. La factibilidad operacional se refiere a la forma como será utilizado el sistema.
c) **Aprobación de la solicitud:** se refiere al momento en que la gerencia decide que un proyecto es factible y deseable, pues en ocasiones son tantas las solicitudes de los usuarios que no es posible atender todas estas peticiones en un corto tiempo.

Determinación de los requerimientos del sistema

Esta etapa describe el hecho en que el analista se entrevista con el usuario para determinar qué se hace actualmente, cómo se hace, con qué frecuencia se realiza, qué tan grande es el volumen de transacciones, cuál es la eficiencia con la que actúan, si existe algún tipo de problema etc.

Diseño del sistema

Se empieza descubriendo cuáles serán las salidas del sistema como los reportes, pantallas, las estructuras de almacenamiento que se ocuparán, la documentación con especificaciones claras y concretas del sistema.

Desarrollo de *software*

En esta etapa se puede instalar o modificar *software* propio o de terceros, o escribir programas a la medida, todo esto depende del tiempo y costo, se puede hacer con personal propio o mediante una consultoría.

Pruebas del sistema

Para asegurarse que el *software* está libre de fallas se instala de manera experimental para ver si funciona como se especificó y si realiza lo esperado. Se utilizan casos de prueba para las entradas del sistema y se analizan los resultados, se tratan de buscar todas las formas no comunes en que se puede utilizar el sistema para verificar que no haya resultados inesperados. Todo esto para que el sistema sea confiable.

2. Método del análisis estructurado

Esta metodología se enfoca en lo que el sistema realizará sin importar la forma en que se realizará, se enfoca en aspectos lógicos y generalmente emplea símbolos gráficos para describir el comportamiento, es decir, movimiento y procesamiento de datos. En esta metodología se incluyen diagramas de flujos de datos y el diccionario de datos. Este método se asigna a cualquier aplicación y además es un buen complemento para otros métodos (Senn 1992:32).

El análisis estructurado se centra en los procesos que se utilizan para realizar modelos de las necesidades del usuario en un sistema.

El análisis estructurado divide un sistema en procesos, entradas, salidas y archivos. Elabora modelos del tipo "entrada-proceso-salida" orientados a flujos para un problema o una solución de empresa.

La técnica del análisis estructurado es sencilla en su concepto. Un nuevo modelo del sistema evoluciona a partir de una serie de diagramas orientados a flujos denominados "diagramas de flujos de datos" o DFD –véase figura 2.3-.

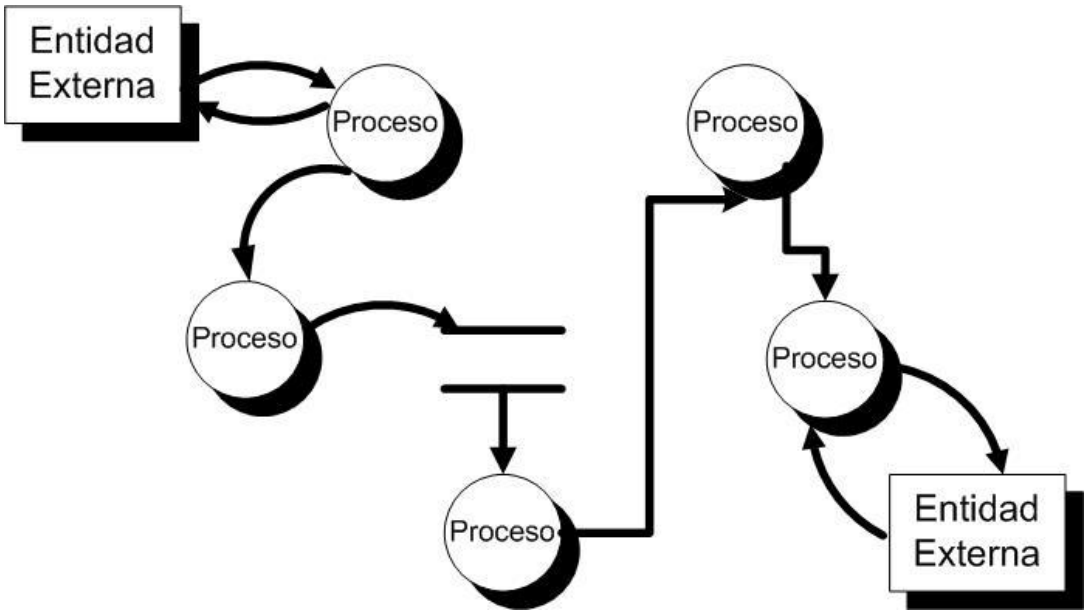


Figura 2.3.- Ejemplo sencillo de un DFD.

Los principales impulsores del análisis estructurado son Tom DeMarco, Chris Gane, Trish Sarson y Ed Yourdon.

3. Método del diseño estructurado

Las técnicas de diseño estructurado ayudan a las personas que hacen desarrollos a abordar programas complejos y de grandes dimensiones.

El diseño estructurado es una técnica utilizada para fragmentar un programa grande en un conjunto jerarquizado de módulos y obtener un programa informático más fácil de implantar y mantener (actualizar).

La idea es diseñar un programa como una distribución jerárquica descendente de módulos. Un módulo es un grupo de instrucciones: un párrafo, un bloque, un subprograma o una subrutina. La estructura descendente de estos módulos se desarrolla conforme a diversas reglas y directrices de diseño.

En un caso ideal, la lógica interna de cada módulo se escribiría por medio de técnicas de programación estructurada. Así se observa que éste tipo de técnicas puede usarse de forma combinada para mejorar la resolución de problemas.

Los principales defensores del diseño estructurado son Larry Constantine, Ed Yourdon, Meiler Page-Jones, Jean-Dominique Warnier, Ken Orr y Michael Jackson.

2.1.5.2. Metodologías basadas en el modelo orientado a objetos

Las técnicas orientadas a objetos han aparecido recientemente con el propósito de introducir cambios en la generación existente de metodologías, tal como lo exige la necesidad de actualización en el ámbito de la informática.

En dicha metodología los datos y los procesos se encapsulan en objetos. Un objeto contiene los datos y procesos que emplean o actualizan dichos datos.

Los objetos se definen desde lo abstracto a lo concreto. Por ejemplo, se puede definir un objeto abstracto de sistemas de información denominado "Reporte", que define los atributos comunes a la presencia de cualquier tipo de reporte (nombre del reporte, fecha, número de página, etc.). Después se podría definir otro objeto para un reporte específico (por ejemplo, "Reporte de proveedores"). Dicho objeto heredaría automáticamente los atributos de datos y los procesos del objeto Reporte y les añadiría los atributos de datos y los procesos exclusivos de dicho reporte específico. Este último reporte puede convertirse en un objeto dentro de otro.

La ventaja de utilizar la metodología orientada a objetos es que no se centra en una teoría en particular sino que crea una a partir de varios autores.

Las metodologías más destacadas son (Stevens 2003:59):

1. El método Grady Booch.
2. Coad – Yourdon
3. Schlaer – Mellor
4. ROOM
5. OMT de James Rumbaugh (Metodología de Modelado de Objetos)
6. OOSE (*Object Oriented Software Engineering*) de Ivar Jacobson

Existen cuatro fases en el ciclo de vida del desarrollo de *software*: iniciación, elaboración, construcción y transición.

En la etapa de iniciación, se lleva a cabo el estudio preliminar, es decir, donde se fundamenta el sistema.

En la segunda fase, de elaboración, se define hasta dónde llegará el producto y la manera como estará estructurado, en esta etapa se analizan los requisitos del sistema, las prioridades, el alcance y el comportamiento funcional y no funcional.

La tercera fase es la de construcción, es decir, cuando la aplicación llega al usuario. También se examinan los requisitos y criterios del sistema de acuerdo a las necesidades del proyecto, de esta forma los recursos se asignan para prevenir los riesgos.

La etapa de transición es cuando se proporciona el *software* a los usuarios de la comunidad (Booch 1999:29).

Algunas metodologías orientadas a objetos antes mencionadas son:

1. *Coad – Yourdon Object Oriented Analysis* (OOA)

En esta metodología se describe un método de Análisis Orientado a Objetos basado en cinco actividades principales:

- Definición de las clases y objetos.
- Identificación de estructuras.
- Identificación de temas.
- Definición de atributos.
- Definición de servicios.

Estas actividades son usadas para construir cada capa de un modelo de objetos de cinco niveles. Los objetos existen en el ámbito del problema. Las clases son abstracciones de objetos. Los objetos son instancias de clases.

2. El método Grady Booch

La metodología de Booch usa los siguientes tipos de diagramas para describir las decisiones de análisis y diseño, tácticas y estrategias, que deben ser hechas en la creación de un sistema orientado a objetos (web.71).

1. Diagrama de Clases. Consiste en un conjunto de clases y relaciones entre ellas. Puede contener clases, clases paramétricas, utilidades y metaclases. Los tipos de relaciones son asociaciones, contenencia, herencia, uso, instanciación y metaclase.
2. Especificación de Clases. Es usado para capturar toda la información importante acerca de una clase en formato texto.
3. Diagrama de Categorías. Muestra clases agrupadas lógicamente bajo varias categorías.
4. Diagramas de transición de estados.
5. Diagramas de Objetos. Muestran objetos en el sistema y su relación lógica. Pueden ser diagramas de escenario, donde se muestra cómo colaboran los objetos en cierta operación; o diagramas de instancia, que muestran la existencia de los objetos y las relaciones estructurales entre ellos.

6. Diagramas de Tiempo. Aumentan un diagrama de objetos con información acerca de eventos externos y tiempo de llegada de los mensajes.
7. Diagramas de módulos. Muestran la localización de objetos y clases en módulos del diseño físico de un sistema. Un diagrama de módulos representa parte o la totalidad de la arquitectura de módulos del sistema.
8. Subsistemas. Un subsistema es una agrupación de módulos, útil en modelos de gran escala.
9. Diagramas de procesos. Muestran la localización de los procesos en los distintos procesadores de un ambiente distribuido.

Etapas del análisis de Booch

Análisis de requerimientos

En esta etapa se define lo que quiere el usuario del sistema. Es una etapa de alto nivel que identifica las funciones principales del sistema, el alcance del modelamiento del mundo y documenta los procesos principales y las políticas que el sistema va a soportar. No se definen pasos formales, ya que estos dependen de qué tan nuevo es el proyecto, la disponibilidad de expertos y usuarios y la disponibilidad de documentos adicionales

Análisis de Dominio

Es el proceso de definir de una manera concisa, precisa y orientada a objetos la parte del modelo del mundo del sistema. Las siguientes actividades son parte de esta etapa:

- Definir Clases.
- Encontrar atributos.
- Definir herencia.
- Definir operaciones.
- Validar e iterar sobre el modelo.

Diseño

Es el proceso de determinar una implementación efectiva y eficiente que realice las funciones y tenga la información del análisis de dominio. Las siguientes actividades se plantean en esta etapa:

- Determinar la arquitectura inicial: decisiones acerca de recursos de implementación, categorías y prototipos a desarrollar.
- Determinar el diseño lógico: detalle del diagrama de clases.
- Implementación física: interfaz a dispositivos o características propias de la implementación.
- Refinamiento del diseño: incorporar el aprendizaje debido a los prototipos y cumplir con requerimientos de desempeño.

3. Metodología de modelado de objetos OMT (*Object Modeling Technique*)

La metodología OMT es una técnica de modelado de objetos, desarrollada por James Rumbaugh, que es uno de los precursores del Lenguaje Unificado de Modelado (UML). Las siglas la definen como una de las metodologías de la Ingeniería de *Software* aplicable al desarrollo orientado a objetos en las fases de análisis y diseño.

Esta metodología hace énfasis en el análisis, no en la implementación. Se centra en los datos más que en las funciones, lo que da estabilidad al proceso del desarrollo. Contiene una notación común a todas las fases a través de tres modelos que capturan los aspectos estáticos, dinámicos y funcionales que combinados proveen una descripción completa del *software* (web.48).

Fases de la metodología OMT

Análisis.- Su objetivo es desarrollar un modelo de lo que va a hacer el sistema. El modelo se expresa en términos de objetos y de relaciones entre ellos, flujo dinámico de control y las transformaciones funcionales.

Diseño del sistema.- Se define la arquitectura del sistema y se toman las decisiones estratégicas.

Diseño de objetos.- Su objetivo es refinar el modelo del análisis y proporcionar una base detallada para la implementación tomando en cuenta el ambiente en que se implementará.

4. Metodología *Object Oriented Software Engineering* (OOSE)

Jacobson plantea una metodología de desarrollo de aplicaciones orientada a objetos.

La metodología OOSE propone el desarrollo de sistemas basados en el uso de distintos modelos. El ciclo de vida que ofrece la metodología se basa en la sucesión de dichos modelos. Los modelos soportados son:

- Modelo de Requerimientos.
- Modelo de Análisis.
- Modelo de Diseño.
- Modelo de Implementación.
- Modelo de Prueba.

5. *Schlaer – Mellor Object Oriented Systems Analysis* (OOSA)

Esta metodología comienza por identificar el ámbito del problema del sistema. Cada ámbito es un mundo separado habitado por sus propias entidades conceptuales u objetos.

Los ámbitos más grandes son divididos en subsistemas. Después, cada ámbito o subsistema es analizado de forma separada en tres etapas:

- Modelo de información: es de representación gráfica y textual e identifica los objetos, atributos y relaciones. Cada objeto es una tabla y cada instancia es una fila de la tabla.
- Modelo de estados: este es una expansión del modelo de información pero además muestra la conducta de cada objeto o relación en el modelo de información.
- Modelo de procesos: este es el desarrollo de un diagrama de flujo de datos para cada método o estado.

Método del prototipo de sistemas

El diseño de prototipos es una popular técnica de ingeniería utilizada para desarrollar modelos a escala (o simulados) de un producto o de sus componentes. Cuando se aplica el desarrollo de sistemas de información, el diseño de prototipos implica la creación de un modelo o modelos iterativos de trabajo de un sistema o subsistema.

Este método se basa en la interacción con el usuario, pues la aplicación se desarrolla mediante la evolución continua con ayuda del usuario.

Se utiliza cuando el usuario tiene poca experiencia y los costos por error son altos, además se puede probar la factibilidad del sistema, pues se identifican los requerimientos del usuario, se puede evaluar el diseño y se sabrá como se utilizará la aplicación (Senn 1992:32).

Ventajas

- Se genera un prototipo con el que el usuario suele interactuar y el problema queda resuelto de manera parcial.
- El usuario puede notar los progresos en relativamente poco tiempo.
- Los siguientes prototipos se hacen con base a requerimientos reales ya que el usuario puede entender mejor que puede o no hacer el sistema.

Desventajas

- Puede haber cambios significativos entre versiones.
- Son muy costosos.
- Se debe esperar un tiempo considerable para tener el sistema terminado en su totalidad.

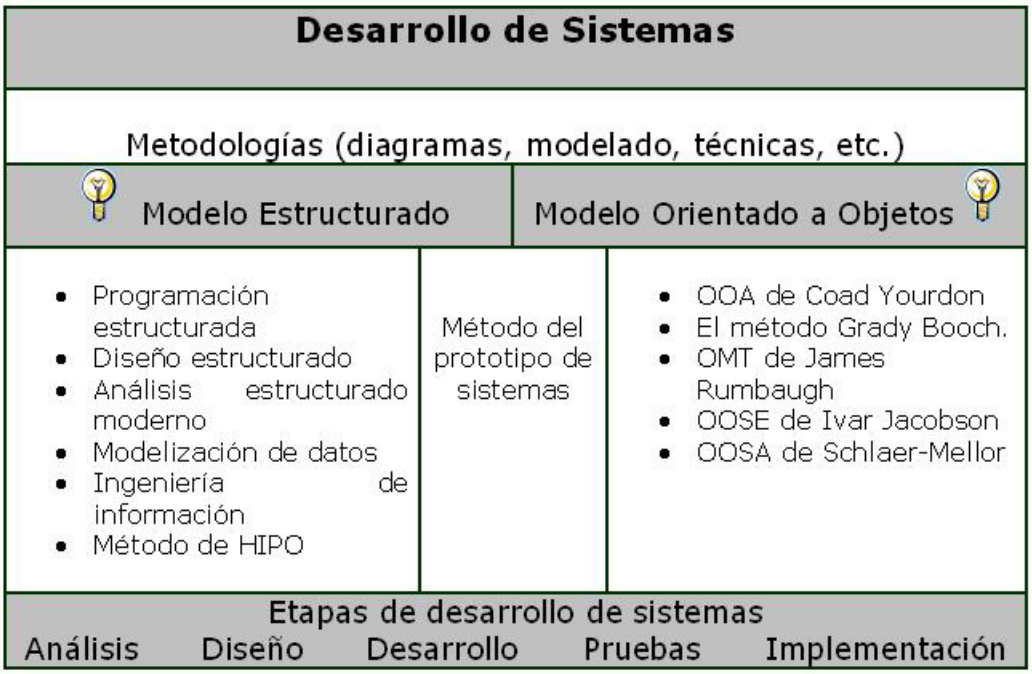


Tabla 2.2.- Cuadro sinóptico del proceso de desarrollo de sistemas.

2.2. Procesamiento de lenguaje natural

2.2.1. Definición

El Procesamiento de Lenguaje Natural (PLN) es una de las ramas más importantes de la Inteligencia Artificial, orientada a facilitar la comunicación hombre-computadora por medio del lenguaje humano o lenguaje natural. El PLN "es la disciplina encargada de producir sistemas informáticos que posibiliten dicha comunicación, por medio de la voz o del texto" (web.23). Esta disciplina data de los años cincuenta y abarca aplicaciones tan importantes como la traducción automática, la búsqueda de información en internet o el análisis de texto por medio de técnicas de carácter estadístico.

2.2.2. Objetivos del PLN

Estos pueden agruparse en tres que son los siguientes:

- Interfaces en lenguajes naturales, hoy en día es posible encontrar que modernas interfaces gráficas basadas en íconos se están volviendo más fáciles de usar y a veces superan la velocidad de escritura de muchos usuarios. Actualmente, parece que una solución más deseable para cubrir las necesidades de los usuarios sería una tecnología mixta consistente en interfaces híbridas de tipo gráfico/LN y voz/LN o voz/LN/gráfico. Los recientes avances

en el procesamiento del lenguaje oral, junto con la tecnología PLN están convirtiendo este tipo de interfaces en una realidad práctica.

- Procesamiento de textos: según se ha estimado en congresos de la IFIP[7] (*International Federation for Information Processing*), hay en todo el mundo más datos almacenados en forma de texto que en cualquier otro formato. Las ciencias de la información han abordado el problema de la recuperación probabilística, pero han tropezado con las limitaciones que plantea el sistema de palabras clave en cuanto al grado de precisión en el proceso de recuperación. Por otra parte, las necesidades de los usuarios van más allá de la recuperación de información e incluyen la extracción de los datos significativos, la elaboración de resúmenes, etc. Las actuales investigaciones en el campo del PLN intentan abordar estos problemas.

- Traducción automática (TA): el objetivo original del PLN ha tomado una vez más la delantera en cuanto a resultados científicos recientes, avances tecnológicos y aplicaciones prácticas. Diversos sistemas multilingües eficaces de TA ya están siendo explotados industrialmente y continuarán evolucionando de manera rápida en un futuro inmediato.

De estos tres se desprenden objetivos como los siguientes:

- Modelar la capacidad lingüística humana para su estudio formal (Lingüística Computacional).
- Representación del léxico, del conocimiento gramatical, morfológico y sintáctico, para los diferentes análisis y las diferentes síntesis superficiales del nivel de conocimiento lingüístico.
- Indexación y recuperación textual y oracional: estructuras de datos, agilidad de la recuperación, segmentación del texto y representación de nuevas conexiones entre palabras a medida que se encuentran.
- Crear aplicaciones que puedan manipular, interpretar y generar lenguaje humano.
- Simular el comportamiento lingüístico humano.
- Modelar la capacidad lingüística humana para su estudio formal.
- La comprensión del lenguaje natural.

[7] IFIP es una organización no gubernamental internacional, cuya creación auspició UNESCO en 1960.

2.2.3. Aplicaciones

El PLN se puede utilizar para aplicaciones típicas tales como:

- Reconocimiento del habla.- Sistemas como los de dictado automático, voz para control de dispositivos, etc.
- Generación de textos.- Generación de lenguaje a partir de una base de datos, generación de informes, explicaciones, oficios, etc.
- Recuperación de información.- Sistemas de consultas de información y desarrollo de sistemas de recuperación de páginas en la *World-Wide Web*, videos y otros formatos.
- Extracción de información.- Generar resultados de búsqueda a través de un análisis del contenido de los documentos y organizar las entidades que identifiquen a dichos resultados.
- Traducción automática.- Sistemas de traducción automática de una lengua a otra.
- Sistemas de diálogo.- Corresponde a la comunicación hombre-máquina ya sea escrita u oral, por ejemplo un sistema de comunicación pregunta-respuesta.
- Motores de búsqueda.- Sistemas de búsqueda, algoritmos, patrones y expresiones regulares.

El PLN maneja algunos recursos como los siguientes:

- Etiquetadores.- Asignan un identificador (etiqueta) a una entidad (palabra o conjunto de palabras) que ha sido identificada y es útil para mostrar el resultado de las búsquedas, consultas, generación de textos, etc.
- *Parsers*.- Son analizadores sintácticos que verifican la correcta combinación de las palabras y sus relaciones con otras para formar sentencias exactas.
- Ontologías.- La Ontología se refiere a la estructuración de conceptos de una disciplina, es decir, definir a los conceptos a los que una disciplina puede hacer referencia, por ejemplo, la Medicina engloba conceptos tales como Pediatría, Odontología, Ortopedia, etc., los cuales estos a su vez engloban a otras unidades conceptuales.
- Lexicones y bases terminológicas.- Un lexicón es el conjunto de palabras de un lenguaje (por ejemplo, un lenguaje natural –diccionario-, un lenguaje de programación –palabras reservadas-, etc.) mientras que una base terminológica es un conjunto de términos con sus respectivas definiciones.
- Lematizadores y analizadores morfológicos.- Un lematizador reduce una palabra a su raíz (lexema, por ejemplo; el lema "camin" para camino, caminar, caminante, caminaré, etc.) o agrega un "lema" a una palabra que por su naturaleza no tiene un lexema en sí, por ejemplo algunos verbos irregulares (verbo ir; voy, vine, fui, iré). La función de un analizador morfológico es descomponer los rasgos de la

palabra en su interior, es decir, si una palabra compuesta está constituida en base a dos palabras simples.

2.2.4. PLN aplicado a la informática

Es factible utilizar el Procesamiento de Lenguaje Natural para algunas aplicaciones referentes a la informática, como pueden ser:

Creación automática de textos.- La generación se realiza a partir de una representación abstracta que debe transformarse en un texto bien formado en todos sus aspectos. El principal problema que puede presentarse es que una misma oración puede representarse lingüísticamente de distintas maneras y teniendo a sus elementos en distinto orden dentro de la misma.

Comprensión del lenguaje.- A través de un sistema de diálogo, el proceso se limita a extraer una representación del significado casi exclusivamente enfocada a localizar la información necesaria en una base de datos. Para lograr este objetivo se requiere la integración de un sistema de reconocimiento automático del habla con un procedimiento de comprensión de lenguaje natural

Traducción automática.- *Systran* es un sistema avanzado utilizado por la comisión europea y que se ha incorporado a portales tales como *Altavista* y *Google*. Pero aún con esto, "los problemas de la traducción automática (TA) son los propios de la interpretación de enunciados en el lenguaje humano: por un lado se requiere conocimiento morfológico, sintáctico, léxico y semántico, mientras que, por otro, es imprescindible en ciertos casos lo que se denomina el conocimiento del mundo, información que difícilmente puede formalizarse, por el momento, en un programa informático" (Llisterri 2003).

Recuperación y extracción de información.- Las técnicas de recuperación y extracción de información, algunas de las cuales incorporan elementos tomados del PLN, constituyen una respuesta al problema de acceso automático a la gran cantidad de documentos digitalizados almacenados en bases de datos.

Para esto se utilizan dos conceptos: la Recuperación de Información (RI), que consiste en seleccionar dentro de un conjunto de elementos, aquellos que contienen la información que un usuario solicita mediante una consulta, los ejemplos mas comunes son los buscadores en internet como *Yahoo*, *Google*, etc.

La Extracción de Información (EI) es mucho más compleja que la (RI) ya que la finalidad de su búsqueda no es únicamente seleccionar los documentos relevantes sino encontrar los datos determinados en la consulta del usuario dentro del contenido de un conjunto de documentos y mostrarlos al usuario de una forma organizada. Los problemas lingüísticos que se presentan son de naturaleza muy diversa, ya que una palabra puede representar varias entidades como un nombre propio, una ciudad o un objeto y la dificultad consiste en fijar su significado con respecto a lo que el usuario desea (Llisterri 2003).

2.3. Datos no estructurados

El texto que no es manejado fácilmente en una tabla de base de datos es considerado como "no estructurado", pero esto está lejos de la verdad, según Dan Sullivan (2001), los textos que se encuentran en los memos, historietas, noticias, planes de proyectos o contratos tienen gran riqueza en cuanto a su estructura, pero no en la manera en que se acostumbra en el mundo de las bases de datos. Para ver la estructura del texto, se debe pasar más allá de filas y columnas de una estructura relacional y dar un giro a la construcción de bloques del lenguaje natural como palabras, frases y sentencias. Esta construcción de bloques se refiere a estructuras naturales, esto es, a la manera como se ha adquirido conocimiento desde que se nace y a lo largo del desarrollo del ser humano para así obtener el entendimiento necesario. Si estas palabras están en inglés, holandés, chino, belga o árabe; con una comprensión de cómo se estructuran las palabras, frases y sentencias, se podrían manipular textos y así extraer información de una forma más efectiva que si se contara con una simple manipulación de cadenas y diseño de técnicas de patrones máquina.

Los elementos estructurados del lenguaje natural no se extienden más allá de los niveles de sentencias. Afortunadamente las estructuras artificiales como los lenguajes de marcado han sido desarrolladas para proveer estructuras de alto nivel en textos. El texto que utiliza marcas extensibles es el lenguaje XML, la generalización estándar del marcado del lenguaje (SGML), y las convenciones similares son llamados documentos semiestructurados. En este caso serán llamados "datos no estructurados" y textos semiestructurados.

Con un juicio de cómo los lenguajes naturales organizan el contenido semántico de los textos, es posible dejar el mito de los textos no estructurados y construir aplicaciones que exploten el potencial de la inteligencia de negocio en los documentos.

2.3.1. El mito del texto no estructurado

Primero que nada, es necesario discernir acerca del mito de textos no estructurados. El término es aplicado a texto que no es colocado y dividido en atributos o valores, que pueden ser fácilmente encontrados dentro de una estructura tabular, como en una base de datos relacional. Por ejemplo:

El texto "Avenida Constituyentes 1600", encontrado en la columna dirección en una determinada tabla, es considerado un enunciado estructurado a causa de que es una unidad lógica simple con un fácil significado discernido, según Sullivan (2001). Por otro lado, es necesario considerar esta sentencia:

"El presidente de los Estados Unidos Mexicanos vive en Avenida Constituyentes 1600".

Esta sentencia generalmente se consideraría como dato no estructurado, pues los atributos, así como el significado de palabras y las relaciones entre palabras, no son manejadas fácilmente por bases de datos convencionales u otras herramientas de programación, sin embargo, es entendible el significado de la sentencia: el jefe del poder ejecutivo de los Estados Unidos Mexicanos reside en una específica dirección, y es posible entender esta sentencia, porque las palabras están organizadas de acuerdo a numerosas estructuras del lenguaje natural. Ahora es necesario comparar la siguiente sentencia:

"La avenida Presidentes Unidos de vida de los Estados a Constituyentes Mexicanos".

La sentencia anterior no tiene estructura alguna. No es posible realizar una figura del significado de esta sentencia sin el primer significado de las palabras y colocando el adecuado orden. El orden de las palabras es un mecanismo de estructuración del lenguaje natural.

Otros mecanismos de estructuración incluyen reglas para la creación de palabras y frases en conjunto. Existen diferentes caminos para llegar a los diferentes significados; dependiendo de que sean agregados prefijos, sufijos y restricciones de lugar para los verbos en el número y tipo de sustantivos que pueden aparecer en una sentencia.

Mientras muchos documentos de negocios son organizados en colecciones de sentencias gramaticales que se complican con los elementos estructurados previamente citados, también se pueden formar textos sin reglas gramaticales tan estrictas. Un *e-mail* es un

perfecto ejemplo: las cartas postales que se escribieron en el pasado (toda persona ha escrito alguna vez una carta) fueron frecuentemente un proceso formal, cuidadosamente dirigido por parte del escritor. Los e-*mails* son mucho menos formales y frecuentemente son muy similares a las conversaciones verbales, con suposiciones sin hablar, referencias ambiguas para otras conversaciones o textos, palabras mal escritas, abreviaturas formadas rápidamente y sentencias incompletas. Mientras los textos en los *e-mails* y otros documentos informales no son gramaticales en los niveles de sentencias, al igual que los textos sin gramática, tienen estructura suficiente en las palabras y niveles de frases para entonces producir resultados para algunas técnicas de minería de textos -véase cap. 2.6-.

Para obtener un orden más externo de la minería de textos, primero es necesario entender cómo opera la estructura del lenguaje natural.

2.3.2. Estructuras naturales

En el nivel más simple, los lenguajes naturales están compuestos de palabras y reglas para la combinación de palabras. Se debe ir de un almacén de documentos con una perspectiva de minería de textos, hacia el desarrollo de sistemas que usen reglas del lenguaje y significados de palabras para producir fácilmente, representaciones entendibles, categorizadas y manejables de textos. La lingüística, el estudio del lenguaje, y en particular un área llamada "gramática generativa" ya tienen identificadas y clasificadas muchas de las reglas y principios que sostienen al lenguaje.

Como se ha explicado, todo texto y aún el lenguaje hablado, tienen una estructura entendible según el idioma al que pertenecen

Para efectos de esta tesis, los datos no estructurados o texto libre se refieren a textos de noticias, cartas, noticia periodística, noticias de revistas científicas, etc., que no han sido procesadas por una persona o máquina para ser controlados en una base de datos.

2.3.3. ¿Por qué utilizar datos o textos no estructurados en lugar de bases de datos?

Una base de datos consiste de una colección de datos persistentes y estructurados que son usados por los sistemas para almacenar y a su

vez otorgar información, de alguna aplicación dada, para satisfacer necesidades u objetivos humanos y organizacionales.

2.3.3.1. Ventajas de las bases de datos

Se puede mencionar algunas ventajas que en la actualidad ofrecen el uso de las bases de datos:

- Compacidad.- los datos son guardados tratando de ocupar el menor espacio.
- Velocidad.- las consultas, inserciones o actualizaciones de datos son agilizadas, debido a la estructura de las bases de datos.
- Facilidad de administración.- dado que el lenguaje con el cual se manejan las bases de datos es de alto nivel, el trabajo con las mismas es mucho más sencillo.
- Vigencia (actualización de los datos).- las actualizaciones de datos son mucho más sencillas.

2.3.3.2. Beneficios del enfoque de base de datos

También se mencionan los beneficios de este enfoque:

- La redundancia puede ser minimizada: es decir los datos que se repiten pueden reducirse mediante llaves foráneas y catálogos en tablas.
- La inconsistencia puede ser evitada: gracias a las restricciones que se le pueden aplicar de manera natural a los datos que recibe cada tabla, estos pueden ser controlados de tal manera que no se repitan, cumplan cierto perfil o que contengan cierto antecedente dentro de la base de datos.
- Los datos pueden ser compartidos: una de las prestaciones más interesantes de las bases de datos es la sencilla manera en que pueden conectar diferentes usuarios al mismo tiempo, pudiendo ser de forma local (desde la misma máquina que contiene la base de datos) o remota (otra máquina diferente a la que contiene la base de datos).
- Puede definirse alguna normatividad (seguir estándares): no sólo se pueden aplicar reglas a tablas específicas, sino a la base de datos en conjunto.
- Restricciones de seguridad pueden ser aplicadas: es posible restringir el acceso a la base de datos mediante la creación de usuarios y además se le pueden asignar diferentes características y permisos a cada uno para restringir las acciones que puede hacer y que no puede hacer.
- La integridad puede ser mantenida: gracias a las reglas que se le aplican a las diferentes tablas o a la base de datos en conjunto, es

posible mantener coherencia dentro de los datos que se contengan.

- Requerimientos conflictivos pueden ser balanceados: un ejemplo de este tema serían las bases de datos con muchos usuarios, estas pueden copiarse o dividirse en varias máquinas para su administración de manera relativamente más sencilla que si estuvieran en texto bruto.

Las bases de datos son el mejor método para controlar datos y registros en donde se puede obtener fácilmente la información mediante un formulario para la captura de los datos de un cliente, de una organización, controlar los productos existentes en un inventario o también para controlar el monto de las ventas, etc.

¿Pero qué sucede cuando se quiere obtener información de noticias a partir de textos en formato electrónico o papel como la *web* o periódicos?, cualquier persona se tardaría días enteros leyendo estas fuentes, además de que tendría que invertir mucho dinero, tiempo y esfuerzo para capturar información relevante de los textos; pues aún no existe un manejador o controlador de datos para los textos no estructurados (textos en formato libre).

Si se tratara de utilizar una base de datos para manipular la información de los textos no estructurados, se tendría que contratar a un analista de sistemas de información para que haga el análisis y el diseño, varios desarrolladores para que realicen la interfaz gráfica y la programación, además de un *tester* (probador) que realice las pruebas. También se necesita una base de datos, lo cual implica un gasto en tiempo y dinero al tratar de adquirir y mantener el *software* (manejador de bases de datos) y además, para la administración y mantenimiento de dicha base se debe contratar a un especialista. Se tendría entonces que comprar equipo para que aloje los datos de la base de datos y así funcione como servidor.

En cambio, se podría analizar cómo están compuestos los textos, identificando las diferentes partes en las que se componen, es decir, identificar la estructura del lenguaje natural, para poder obtener la información relevante. La forma en que se estructura el lenguaje natural es la siguiente:

2.3.3.3. Niveles de estudio del lenguaje

El estudio de la lengua puede ser agrupado dentro de cinco principales áreas:

- Morfología: el estudio de la estructura y forma de las palabras.
- Sintaxis: estudio de la forma en que las palabras y frases forman sentencias.
- Semántica: el significado de las palabras y declaraciones.
- Fonología: el estudio de los sonidos en el lenguaje.
- Pragmática o gramática del contexto: el estudio de las frases del idioma que no pueden ser analizadas con las restricciones del análisis semántico, es decir, a la intención que se tiene al decir las cosas.

El análisis del lenguaje natural se puede utilizar para dividir los textos en sus diversos componentes y así marcar estos para identificar fácilmente la información que se desea extraer.

2.4. Lingüística computacional e Ingeniería lingüística

En la actualidad hay una confusión de términos sobre "Lingüística computacional", la "Ingeniería lingüística" y la "Informática aplicada a la lingüística", debido a factores como la reciente actividad del manejo del lenguaje por computadora o la vertiginosa evolución de las tecnologías computacionales, lo que ocasiona que los términos se vayan forjando «al calor de la batalla», es decir que todavía no se ha acuñado completamente uno, cuando ya se ha propuesto otro.

Por otro lado, el carácter interdisciplinario de los tres términos provoca "traslapes" teóricos, prácticos y terminológicos, así como la diversidad de posturas en distintas fuentes bibliográficas, producto de la visión del autor. Como mencionan Llisterri y Garrido (1998) "la ingeniería lingüística constituye un campo de trabajo interdisciplinario en el que confluyen la informática y la lingüística, de aquí que algunos autores se refieren a la "lingüística informática" o a la "informática lingüística".

2.4.1. Lingüística computacional

Comúnmente la lingüística computacional se trata como sinónimo o de la ingeniería lingüística o del procesamiento del lenguaje natural o incluso de ambos. Pero como se muestra a continuación, son actividades diferentes.

La investigación en Lingüística Computacional (LC) se ocupa de la aplicación de un paradigma computacional al estudio científico del

lenguaje humano, y a la ingeniería de sistemas para el tratamiento o análisis del lenguaje escrito o hablado (Jordán 1992).

Martin Kay (2003) brinda de forma ilustrativa la historia de la lingüística computacional. Los antecedentes de esta disciplina establecidos en 1949 están unidos a la propuesta de realizar traducción automática por computadora. La frase "*computational linguistics*" aparece a mediados de los sesenta y el término como tal es propuesto por David Hays cuando invita a dejar de lado la traducción automática para realizar investigaciones en otros aspectos del procesamiento del lenguaje.

Para el mismo Kay (2003) la lingüística computacional tiene dos objetivos. El primero sería dar avances en la teoría lingüística. El segundo sería brindar una tecnología que brinde soluciones prácticas. Grishman (1986/1991:15) define la lingüística computacional como "el estudio de los sistemas de computación utilizados para la comprensión y la generación de lenguas naturales". De igual forma Arrarte (1995:4) menciona que "se encarga del estudio de los sistemas de computación utilizados para la comprensión y la generación de lenguas naturales".

Un comentario adicional es que la lingüística computacional parece ser parte de la lingüística aplicada (Bolshakov y Gelbukh 2004:18).

Finalmente parecen acertadas las definiciones de Moure y Llisterri (1996), "bajo la denominación de lingüística computacional es posible agrupar un conjunto heterogéneo de teorías, métodos, herramientas, aplicaciones y productos que tienen en común la consideración de la lengua como un objeto susceptible de ser tratado mediante procedimientos informáticos" y la de Uszkoreit (1996), "*Computational linguistics (CL) is a discipline between linguistics and computer science which is concerned with the computational aspects of the human language faculty*".

2.4.2. Ingeniería lingüística

La definición de ingeniería lingüística dada por la Comisión Europea en el documento "Ingeniería lingüística. Cómo aprovechar la fuerza del lenguaje", es la siguiente: "la ingeniería lingüística es la aplicación de los conocimientos sobre la lengua al desarrollo de sistemas informáticos que puedan reconocer, comprender, interpretar y generar lenguaje humano en todas sus formas". En la práctica, la ingeniería lingüística consiste en una serie de técnicas y recursos lingüísticos, que se aplican,

en el primer caso, por medio de programas informáticos y que, en el segundo, constituyen una fuente de conocimientos a los que se puede acceder por medio de programas informáticos (web.73).

Hoy en día la misma Comisión Europea ha cambiado el nombre a "tecnologías para el lenguaje humano" o simplemente "tecnologías del lenguaje" (Llisterri 2004). Éstas incluyen las tecnologías del texto y las tecnologías del habla (Llisterri y Garrido 1998).

Con base en la definición y en un intento por esquematizar la actividad de la ingeniería lingüística, se observa la siguiente tabla (tabla 2.3). En ella se muestra el ámbito completo de esta disciplina.

Lengua hablada/escrita	⇨	Sistema Informático	⇨	Lengua hablada/escrita
		Procesamiento de información lingüística		
Reconocer	⇨	Comprender – Interpretar – Analizar	⇨	Generar

Tabla 2.3.- Ingeniería Lingüística

Todo sistema informático que produciría la ingeniería lingüística podría tener como entrada distintos datos lingüísticos mediante reconocimiento óptico de caracteres o reconocimiento del habla. Él mismo se encargaría de comprender, interpretar o analizar el lenguaje a través del procesamiento de la información lingüística, ya sea a nivel morfológico, sintáctico, semántico y/o pragmático. Igualmente, la salida podría ser lenguaje natural vía generación de textos o síntesis del habla.

2.4.3. Informática aplicada a la lingüística

Esta área, que combina la lingüística y la informática, es la más clara de precisar. Se puede definir esta actividad como la aplicación de la tecnología de información para la automatización de la labor de investigación lingüística. Suelen entrar en ella los programas para obtener información estadística, programas de enseñanza de lengua, construcción de recursos y atlas lingüísticos. Se trataría del "uso de

herramientas informáticas para construir elementos auxiliares para una investigación" (Moure y Llisterri 1996).

2.4.4. Interacción de las áreas

Con el fin de terminar de definir cada disciplina, se visualiza una representación esquemática que muestra la interacción entre ellas.

Informática aplicada a la lingüística				
⇩	Tecnología de información			
Lingüística	⇩	⇩	⇩	⇩
Lingüística computacional	⇨ Modelos Programas (PLN)	Ingeniería lingüística	⇨ Sistemas Tecnología	Industrias de la lengua (Comercialización)
Ciencia de la computación				

Tabla 2.4.- La ingeniería lingüística y su interacción con otras áreas

La lingüística computacional es un área interdisciplinaria de la lingüística y la ciencia de la computación dedicada a la formulación de modelos lingüísticos computables y al desarrollo de programas que los apliquen computacionalmente. Hace uso del procesamiento del lenguaje natural (inteligencia artificial) para desarrollar dichos programas.

Por otro lado, la ingeniería ligüística, o tecnología para el lenguaje humano, toma estos modelos y programas y junto con la tecnología de información que brinda la informática, construye sistemas de procesamiento de información lingüística capaces de reconocer, comprender, interpretar, analizar y generar lenguaje humano. Dichos sistemas son aplicaciones informáticas comercializables y de aplicación específica a problemas concretos; a diferencia de la lingüística computacional que brinda programas y teorías de uso general.

Finalmente las empresas dedicadas al proceso comercial de estos sistemas y de la tecnología que los acompaña es lo que se conoce como "industrias de la lengua".

Así se concluye que los tres conceptos explicados anteriormente van muy ligados uno con el otro, ya que sin los modelos lingüísticos que

proporciona la LC para uso general, no podría haber materia prima para la IL. La función de la IL es, con ayuda de la Informática, producir sistemas informáticos para la generación (entre otras funciones) de lenguaje humano.

2.5. Técnicas y recursos de la Ingeniería lingüística

A continuación se mencionan algunas de las múltiples técnicas que se utilizan en ingeniería lingüística.

- Identificación y verificación del locutor.
- Reconocimiento del habla.
- Reconocimiento de caracteres e imágenes.
- Comprensión del lenguaje natural.
- Generación de lenguaje natural.
- Generación de habla.

Los recursos lingüísticos son un elemento esencial de la ingeniería lingüística. Constituyen una de las principales formas de representar el conocimiento de la lengua, que se utiliza en los trabajos de análisis conducentes al reconocimiento y la comprensión.

El trabajo de producir y mantener recursos lingüísticos es una tarea descomunal. De la producción de los recursos se encargan centros de investigación e instituciones públicas, con arreglo a formatos y protocolos normalizados que permiten utilizarlos en muchas de las lenguas de la UE. La Asociación Europea de Recursos Lingüísticos (ELRA, "*European Language Resources Association*") produce buena parte de estos recursos.

2.5.1. Analizadores sintácticos y tokenización

Dentro del amplio ámbito de dominio del procesamiento del lenguaje natural, una de las funciones esenciales de los analizadores sintácticos o *parsers* es el análisis de cadenas de *tokens* en busca de posibles errores sintácticos[8]. Un *token* se puede definir como la unidad mínima de información con significado propio dentro de una secuencia de caracteres alfanuméricos. Estas cadenas de unidades mínimas de

8 La sintaxis, entendida en sentido amplio, es aquella parte de la gramática que se ocupa de las normas que rigen la formalización de las palabras en estructuras mayores tales como las oraciones, así como de las relaciones que establecen entre sí dichas palabras.

información o unidades léxicas son generadas previamente por el módulo lexicográfico integrado en el *parser*, encargado de identificarlas dentro de un texto o secuencia ordenada de caracteres alfanuméricos. Por su parte, la tokenización es un proceso que consiste en la descomposición, en forma de lista, de esas cadenas de *tokens* en sus unidades mínimas. Así, un programa de este tipo podría generar la siguiente lista de *tokens* a partir de la frase "¡Hola Mundo!":

[161, 72, 111, 108, 97, 32, 77, 117, 110, 100, 111, 33]

Donde cada uno de los números de la lista se corresponde con el carácter ASCII (*American Standard Code for Information Interchange*) correspondiente a cada una de las unidades mínimas de significación identificadas en la frase, en el mismo orden. Por supuesto es posible llevar a cabo el proceso inverso, y a partir de esa lista generar las cadenas de *tokens* que forman la frase en cuestión. La tokenización es por lo tanto el proceso básico que permite manejar el lenguaje natural escrito para su posterior procesamiento, con base en su descomposición en unidades mínimas de información con significado propio. En el ejemplo anterior la unidad mínima de la frase son las letras (*tokens*). Así como en otras ocasiones puede suceder que la unidad mínima de un texto sean las palabras, dependiendo de la necesidad del programador.

La mayor parte de los lenguajes de programación contemplan instrucciones específicas para llevar a cabo el proceso de tokenización de cadenas ordenadas de caracteres alfanuméricos, si bien es posible implementar alternativamente esta operación mediante otros procedimientos proporcionados por esos lenguajes.

Así, un programa que pretenda "leer" un texto, primero deberá tokenizarlo, generando una lista de los *tokens*, identificados en ese texto. A continuación, procederá a identificar unidades mayores de significado propio, contemplando por ejemplo la presencia, como elemento separador, del carácter ASCII 36[9], lo que se podría asimilar como "palabras", para finalmente, acabar identificando otras unidades de significación de orden superior, frases u oraciones. Diferenciadas las oraciones del texto "leído", el *parser* procede a realizar el análisis sintáctico propiamente dicho, identificando para ello las partes constitutivas de dichas oraciones que, a tal fin, son comparadas con patrones previamente definidos de estructuras posibles, que dependerán de la lengua de escritura del texto, y del nivel de

9 Que corresponde al espacio en blanco.

complejidad de análisis que se pretenda alcanzar, ya que contemplar todas las posibles estructuras de una lengua y sus numerosas variaciones, y representarlas mediante una serie de reglas no es una tarea precisamente sencilla.

La detección de las variaciones de posición admitidas en cada lengua, en relación con el orden de las palabras, o análisis de las transformaciones, se realiza mediante procesos de análisis estructural que tratan de identificar la estructura profunda de una oración en relación con su estructura superficial. El análisis estructural, en base a la estructura superficial (2) de una oración y, cambiando el orden de determinadas palabras, trata de determinar su posible transformación a una estructura de tipo profundo (1):

(1) Estructura profunda: "Pedro come una manzana"
(2) Estructura superficial: "Come Pedro una manzana"

La implementación del proceso de tokenización, al margen de la utilización de instrucciones específicas que transforman directamente una cadena de caracteres alfanuméricos en una cadena de *tokens*, implica la utilización de otro tipo de instrucciones cuya función es la "lectura" individual, uno a uno, de los caracteres presentes en el canal o grupo activo de entrada de datos (*input stream*) que se haya especificado, que por lo general será bien el teclado de la computadora, que es el canal activo de entrada por defecto (al igual que el canal de salida de datos, *output stream*, por defecto es el monitor de la misma), o bien un archivo de texto ubicado en la ruta que se indique.

El analizador sintáctico, con base en los constituyentes de una oración, y mediante un número finito de reglas, trata de determinar la gramaticalidad o no de un número infinito de construcciones. Un analizador sintáctico trata de ver hasta qué punto puede someterse un grupo de palabras a una estructura de reglas. Así por ejemplo, si se tiene la oración:

"Pedro come una manzana"

En primer lugar, y mediante un proceso de *tokenización*, se genera una lista de las palabras que contiene la oración. De esta lista inicial de palabras, se puede diferenciar una sublista que se corresponda con el Sintagma Nominal (SN) de la oración, y si ésta puede concatenarse con otras sublistas que según determinadas reglas se verifica como

Sintagma Verbal (SV)[10], la oración se concluye que es gramatical. Lo que importa en los constituyentes es el orden de las palabras de la oración.

El analizador sintáctico realiza el análisis secuencialmente, palabra por palabra, partiendo de una lista inicial que, siguiendo con el ejemplo de la oración expuesta, sería:

[Pedro | come | una | manzana]

El proceso de computación de las reglas del analizador sintáctico debe dar como resultado otra lista, que será una lista vacía [] si la oración inicial es gramatical (siempre en base a las reglas que tenga definidas el analizador). En definitiva, partiendo de la lista inicial de palabras, el analizador sintáctico comprueba si ésta se puede subdividir en dos sublistas, que se corresponden, respectivamente, con el SN y el SV de la oración.

2.5.2. Etiquetado

Etiquetar, primeramente, se refiere a resaltar o hacer sobresalir aquello que interesa. Para los fines de etiquetado de un corpus -véase cap. 5-, éste consiste en resaltar a través de una marca las partes que se consideran más importantes, o simplemente aquellas que interesan del mismo, un ejemplo de estas marcas es: en HTML se utiliza la etiqueta <TITLE> ... </TITLE> y esto indica que dentro de ella se encuentra el título de una página.

Para poder hacer etiquetado de un corpus se necesita: identificar las palabras o unidades léxicas a etiquetar, definir las clases de palabras refiriéndose a la gramática que quiere realizarse, definir las etiquetas con las que se van a anotar las clases de palabras y el procedimiento con el que se va a etiquetar el corpus.

2.5.2.1. Definición

Consiste en identificar a los elementos léxicos y las propiedades morfológicas de un corpus y asociarlos a la información que de ellos se

[10] Se llamará Sintagma Nominal a la estructura común donde se encuentre un sustantivo (algunas veces puede ser el "sujeto") y Sintagma Verbal a la estructura del predicado (Verbo, objeto directo, modificadores circunstanciales, etc.).

tiene en un lexicón o base terminológica, su función es dar una asignación automática de descriptores o etiquetas a las palabras.

Los etiquetadores automáticos fueron hechos desde finales de los años cincuenta debido a los problemas relacionados con la tokenización y el análisis léxico, pero uno de los problemas más grandes del etiquetado es la ambigüedad.

2.5.2.2. Aplicaciones

A continuación se enlistan varios de los principales usos que se les dan a los etiquetadores POS según Atro Voutilainen.

- Aplicaciones de tecnologías de información, la recuperación e indexación de textos pueden verse beneficiadas del etiquetado pos; los enunciados y adjetivos son mejores candidatos en términos de indexación que los adverbios, verbos o pronombres.
- Los grandes corpus textuales etiquetados (por ejemplo El corpus Británico Nacional o el corpus del Banco Inglés) son usados como datos para estudios lingüísticos.
- Beneficios en cuanto a velocidad de procesamiento del etiquetado, por ejemplo el adjetivo "bajo" (estatura, altura, etc.) tiene un significado distinto al sustantivo "bajo" (instrumento musical), a la preposición "bajo" o al verbo "bajo" (bajar) como se muestra en la siguiente oración:

"Los juguetes están bajo la cama"

En este caso la palabra "bajo" se encuentra como preposición, mientras que en la siguiente se encuentra como verbo:

(yo) "bajo a la recepción en 15 minutos"

Al problema anterior se le llama ambigüedad.

2.5.2.3. Arquitectura

Para Voutilainen la arquitectura de la mayoría de los etiquetadores es notablemente similar.

a) Tokenización. El texto de entrada es dividido en *tokens* (unidad mínima del texto, generalmente de palabras), que serán convenientes para un futuro análisis. Las marcas de puntuación, uniones de palabras y límites de pronunciación son indicadores para dividir en *tokens*.

b) Búsqueda de ambigüedad. Esto implica el uso de un lexicón -véase cap. 2.5.3- y un *guesser*[11] para los *tokens* que no están representados en dicho lexicón.
   - El lexicón puede ser una lista de palabras y de las partes posibles de una oración. Las soluciones más económicas (factibles) están basadas sobre modelos de estado finito, por ejemplo: una morfología de dos niveles, en donde las generalizaciones lingüísticas pueden ser realizadas de una forma más adecuada.

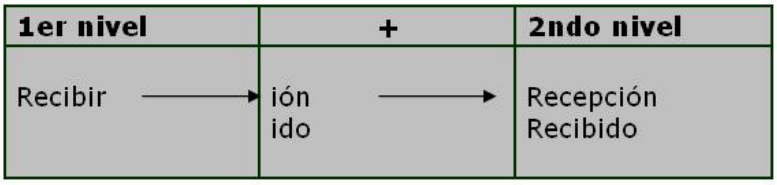


Figura 2.4.- Ejemplo de una morfología de dos niveles.

   - *Guesser*, analiza los *tokens* restantes. El diseño de los *guesser* está basado sobre el conocimiento acerca del lexicón. Si se conoce que el contenido de un lexicón es el conjunto de palabras cerradas como pronombres y artículos. Los *guesser* pueden ayudar a que el análisis del tema sea correcto y que el propósito sea alcanzado analizando sólo palabras abiertas (verbos y sustantivos).
   - Usados con un compilador, el lexicón y el *guesser* constituyen un analizador léxico que deberá proponer alternativas razonables para cada *token*.

c) Resolución de la ambigüedad. La desambiguación está basada en dos fuentes de información, ambas codificadas en un modelo formal del lenguaje, los modelos son:
   - La información acerca de la propia palabra, por ejemplo la forma de la palabra "vale" es más frecuentemente usada en un texto común como verbo (valer) que como sustantivo (ticket, boleto, etc.).

   - Debe cuidarse el uso de las palabras, prestando atención al contexto en el que se aplica, y analizando la coherencia

[11] Procedimiento para asignar una interpretación semántica a palabras desconocidas, basado en la caracterización de las palabras por su forma y contexto de ocurrencia.

entre el sustantivo y el verbo, y los artículos que estén antes o después de ellos.

2.5.2.4. Tipos de etiquetado

Hay varias formas de etiquetado:

- Fonético y fonológico: es decir, el etiquetado de los sonidos en corpus orales.
- Gramatical: se resaltan las categorías morfosintácticas, es decir, la identificación de las partes de la oración o la forma en que está constituida. Para referirse a las clases de palabras se utiliza el concepto de "partes de la oración" o sus siglas en inglés POST (*Part-of-Speech Tagging*).
- Morfológico: consiste en determinar la forma, clase o categoría gramatical de cada palabra de una oración.
- Ortográfico: errores y variantes ortográficas.
- Pragmático y estilístico: etiqueta el contexto del uso del lenguaje (un ejemplo de pragmática sería "yo me baño todos los días" equivale a decir "me baño todos los días").
- Prosódico: las señales tonales, es decir, la entonación de la voz al decir una oración.
- Sintáctico: consiste en encontrar las relaciones sintácticas que hay entre las partes de una oración en el corpus.
- Semántico: se refiere a la clasificación léxica de las palabras, además de ayudar a la desambiguación de las palabras en un corpus, ya que se le asigna a cada palabra el sentido más apropiado.
- Textual: se refiere al etiquetado de la información metalingüística, utilizando para ello por ejemplo XML.

2.5.2.5. Métodos para etiquetar las partes de una oración

- Etiquetado basado en reglas. Para esto se utiliza una base de datos en la que se encuentran las posibles ambigüedades y se explica cómo deben tratarse, por ejemplo: si se encuentra el sujeto "Yo" y después el verbo "ser", entonces éste deberá cambiarse a la forma "soy".
- Etiquetado basado en transformación: se basa en reglas para determinar qué etiqueta debe tener cuando se encuentra una palabra ambigua, además a partir de estas reglas se modifica el propio corpus.

2.5.2.6. Evaluación del etiquetado

La representación de las etiquetas puede ser evaluada de muchas formas, calculando la velocidad y la memoria, hay diversos niveles de representación y pueden clasificarse por medio de la importancia de cada uno.

Lo que hace una gran diferencia entre los sistemas etiquetadores, es la calidad lingüística que tienen. Debe haber una correcta especificación de las etiquetas asignadas antes del etiquetado de un texto.

- La cuestión informativa de un texto no es fácil medirse ya que el tamaño de la etiqueta y la ambigüedad en el texto son las medidas.
- La especificación de las etiquetas asignadas es de vital importancia para los lingüistas ya que esto garantiza entre un 97% y 100% de confiabilidad. Por eso los lingüistas, al elaborar las etiquetas, deben ser cuidadosamente específicos.
- Exactitud. El diseño de las etiquetas debe ser realizado por un lingüista, el cual podrá ser evaluado por comparaciones de varios textos selectos, esto se realizará por diversos lingüistas quienes elaborarán dictámenes de aceptación.

2.5.2.7. Principios básicos que deben aplicarse al etiquetado de corpus

Las córporas existentes difieren bastante en el tipo y cantidad de anotación y codificación que poseen. Existen siete máximas que deben aplicarse en la anotación de la córpora, máximas que son resumidas a continuación (Pérez y Chantal 2002):

1. Debe ser posible eliminar las etiquetas añadidas a un texto anotado y recuperar el texto original sin que éste sufra modificación alguna.

2. Debería ser posible también extraer las anotaciones de los textos y almacenarlas de forma independiente, por ejemplo en una base de datos relacional o en líneas paralelas al texto original.

3. El sistema de anotación usado debe estar basado en directrices documentadas y accesibles al usuario final del corpus, de modo que pueda tener acceso tanto a un listado completo de las etiquetas usadas como a las decisiones tomadas en el proceso de etiquetado.

4. Debe ser posible incluir información sobre la autoría de la codificación del texto, de forma que sea posible saber si se ha realizado manualmente (y por quién), o si se ha realizado de forma automática con o sin revisión posterior por un lingüista.

5. Se debe hacer conciencia al usuario final de que las anotaciones añadidas al corpus no son infalibles, sino que simplemente constituyen una herramienta de ayuda para el análisis. Cualquier anotación que se añada al corpus será, por definición, un acto de interpretación y de análisis del texto, por lo que es susceptible de incorrecciones e inexactitudes.

6. Los sistemas de anotación han de estar basados en la medida de lo posible en principios teóricamente neutrales y sobre los que exista un acuerdo amplio en el seno de la comunidad científica.

7. A pesar de la búsqueda de un estándar para el etiquetado, es necesario realizar algunas adaptaciones a los etiquetadores actuales. Ningún sistema de etiquetado puede ser considerado estándar.

Estas son algunas buenas sugerencias para ser implementadas en los sistemas de anotación, con el fin de evitar los problemas que estos plantean, tanto para los usuarios finales como para la reutilización del material textual etiquetado.

2.5.3. Lexicones

El conocimiento es el fundamento de cualquier sistema de comprensión del lenguaje natural, ya que las oraciones se constituyen por palabras y son éstas las que llevan asociado un conjunto de información morfológica, sintáctica y semántica necesarias en el proceso de análisis posteriores.

Para el estudio léxico es importante tener en cuenta la morfología, la cual se basa en la construcción de las palabras desde componentes básicos como son la raíz más los sufijos o prefijos, es decir, la descomposición de las palabras en una cadena de morfemas. También es importante verificar la consistencia de la semántica de las oraciones.

2.5.3.1. Información léxica

"Un lexicón[12] es un repositorio de información léxica, donde a cada unidad léxica se asocia un conjunto de información que incluye su categoría sintáctica, interpretación semántica a nivel léxico, y diversas propiedades morfológicas, sintácticas y semánticas" (Moreno 1999:44).

La construcción de un lexicón deriva problemas debidos al volumen de información como: segmentación de la oración en palabras, multiplicidad de palabras con la misma grafía (homonimia), la existencia de diversos significados para una misma palabra (polisemia), etc.

"El análisis léxico consiste en la identificación de las unidades léxicas en las oraciones que componen el texto objeto de análisis" (Moreno 1999:44).

Una forma de segmentación es considerando la palabra ortográfica como la unidad léxica, mediante la identificación de secuencias de caracteres separados por espacios en blanco o cualquier símbolo separador. Sin embargo existen palabras con distintas formas gramaticales (al=a+el, del=de+el, díselo=dí+se+lo) o también hay palabras gramaticales que son representadas ortográficamente con más de una palabra, por ejemplo: "sin embargo", "no obstante" (Moreno 1999:44).

Para cada unidad léxica del lexicón se incorpora información como (Moreno 1999:44):

- Categoría Sintáctica: etiqueta asociada a grupos de unidades léxicas, las cuales dependerán del formalismo al cual se quiera llegar. Existen categorías cerradas como el determinante[13], la preposición, etc. y abiertas como los verbos o los adjetivos.
- Características sintácticas de concordancia: género, número, persona, etc.
- Información morfológica: reglas de formación de palabras.
- Información semántica: categoría semántica, forma lógica asociada, rasgos semánticos, etc.

---

12 En términos simples, un tanto informales, puede definirse a un lexicón como un conjunto de palabras para un determinado uso.

13 Los nombres usualmente están unidos a palabras que limitan o especifican su significado. La palabra "carro" puede referirse a cualquier carro, pero si se agrega "mi carro" se está determinando su pertenencia hacia una persona. Los determinantes son parte habitual de las oraciones para acompañar a los sustantivos para especificar o limitar su significado, adicionando información de pertenencia, número, género, espacio, etc.

- Otras informaciones: restricciones selecciónales[14], tipo de complementos que una palabra rige, preposiciones que admite, etc.

Con estas características se deduce, cuando la palabra es un verbo, que es necesario conocer su número, persona y restricción seleccional para conocer el tipo de sujeto y complemento o si debe estar antecedido por cierta preposición, etc.

Una forma de representar la información léxica es mediante "Estructuras de rasgos" (*Feature Structures*). Así se les llama a las listas de atributos, las cuales pueden tener un valor atómico u otras estructuras diferentes.

Las representaciones de estructuras de rasgos sirven para expresar la generalización, capturar la herencia y evitar las redundancias.

En un sistema de lenguaje natural es necesario contar con un lexicón básico (éste a su vez puede ser específico o de propósito general) y un conjunto de reglas; estas pueden ser morfológicas o léxicas.

Las combinaciones válidas de morfemas las establecen las reglas morfológicas. Las reglas pueden ser de deletreo, de asignación, valores por defecto o de formación.

Las reglas léxicas son un complemento y pueden ser de modificación y de las estructuras de rasgos de entrega (reglas de compleción), de creación de estructuras nuevas a partir de otras llamadas reglas de multiplicación, de verificación de la consistencia de los rasgos presentes en la estructura llamadas reglas de consistencia.

Existen dos formas de adquirir información léxica: la adquisición manual y la automática.
La construcción manual del lexicón se realiza por un lexicógrafo mediante herramientas pero ésta se complica cuando la información es de gran tamaño, pues se vuelve más compleja.

Investigadores han optado por utilizar los diccionarios en soporte magnético como fuente de información léxica. Para obtener una base de datos mediante el procesamiento de la información del diccionario,

14 Forman parte de las presuposiciones (información que se infiere a partir del enunciado). Por ejemplo, en el enunciado: "si mi mascota maúlla dale de comer", se parte del hecho de que sólo los gatos maúllan, por lo tanto la mascota a la que se hace referencia debe ser un gato.

luego la extracción de información de la base de datos léxica es sencilla utilizando la información obtenida y así la representación de ésta en el lexicón.

Por ejemplo: *Acquiliex* es un proyecto realizado para extraer información de múltiples diccionarios en soporte magnético con el fin de construir una Base de Conocimientos Léxica Multilingüe (inglés, francés, italiano y español). El desarrollo del proyecto se basó en dos etapas: en la primera se construyó la base de datos léxica a partir del diccionario. Luego se vio la forma de extraer información semántica contenida en la base de datos léxica, además de herramientas para transformar, integrar y enriquecer la información.

La disponibilidad de córpora (conjunto de varios corpus) y herramientas de análisis y selección de subconjuntos facilitan la tarea de los lexicógrafos[15]; como en la construcción del diccionario *Cobuild* el cual se creó a principios de los años ochenta y en él participaron la Universidad de *Birmingham* (Inglaterra) y la editorial *Collins*, con la finalidad de investigar y describir la lengua inglesa por medio de técnicas computacionales. En los noventa aparece la adquisición automática de información léxica, pues se pueden construir lexicones de forma semiautomática.

Un lexicón guarda su información de forma estructurada en árboles y tablas. Los "*trie*" realizan un óptimo mantenimiento a los diccionarios.

El *trie* consulta la información en tiempo proporcional a la longitud de la cadena y no de acuerdo al número de palabras que existen en el diccionario, además compacta la información almacenada en el diccionario. También se expresa la relación entre las raíces con los sufijos y afijos, entre otras palabras. En los lenguajes con abundancia de formas flexionales[16] como el español y el francés, se optimiza el almacenamiento de palabras, pues una entrada más una o dos letras que corresponden a su sufijo forman otras palabras derivadas de dicha entrada.

---

15 Personas encargadas de realizar los lexicones.

16 Formas en las que se puede flexionar una "palabra". Por ejemplo Vivir, viviré, vivimos, vivirán, vivito, etc.

2.5.3.2. Representación de un lexicón

Existe un tipo de dato *trie* comúnmente llamado "árbol de letras" que se deriva de la palabra *retrieval*, es un dato que representa un conjunto de caracteres. Su estructura se basa en "N" ramas y la información de cada nodo es común a sus sucesores, de esta forma se busca la clave caracter por caracter en forma descendente en el árbol. La clave inicial es la raíz del árbol y el nodo final es un indicador de finalización de palabra. En otras palabras, cada camino desde la raíz hasta un nodo terminal es una palabra de diccionario.

Esta estructura temporal lineal está en función de la longitud de la palabra para las operaciones de búsqueda de una palabra (O(L), L=longitud de la palabra a buscar) y su máxima altura corresponde a la palabra más larga del diccionario. La complejidad obtenida está en función del número total de palabras almacenadas (O(N),N=número total de palabras almacenadas en el diccionario).

Mediante este mecanismo se observa que hay una sola entrada por palabra que viene definida por cualquier nodo del árbol. Se dice que una palabra está almacenada cuando se encuentra un nodo terminal con su lista de entradas léxicas (Moreno 1999:47-53).

La estructura de datos se puede representar mediante una tabla de dispersión o una tabla *Hash*.

2.5.3.3. Representación de un *trie* utilizando un vector

El *trie* se almacena en memoria mediante un único hecho y contiene un solo argumento, el cual es el nodo raíz del *trie*: "*trie* (raíz)".

Cada nodo del *trie* tendrá la raíz nombrada "nodotrie" y dos argumentos:

- Léxico: lista de entradas léxicas de la palabra a la que representa el nodo.
- Ramas: es una estructura formada por vectores.

Una palabra de diccionario consta del recorrido de la raíz al nodo terminal; pero es importante distinguir el nodo que delimita el final de la palabra, y los nodos intermedios o prefijos de la palabra, por ejemplo la palabra "nuestra" se forma de los nodos "n-u-e-s-t-r" que se consideran como prefijo, y no como una palabra completa. El último nodo definirá la palabra, en este caso "a", es decir se almacenarán las entradas léxicas de cada nodo hasta llegar al nodo final, para formar la palabra. Y esto se distingue por medio de la siguiente condición.

"En el caso de que un nodo no suponga el final de una palabra, el campo Léxico contendrá la lista vacía "[]". En caso de que sí corresponda al final de una palabra, entonces almacenará su lista de entradas léxicas".

El nodo raíz está vació, pues no contiene un campo léxico, entonces se convierte en una lista vacía, porque ninguno de los nodos indica el final de la palabra. A continuación se muestra la letra que representa cada nodo de un árbol.

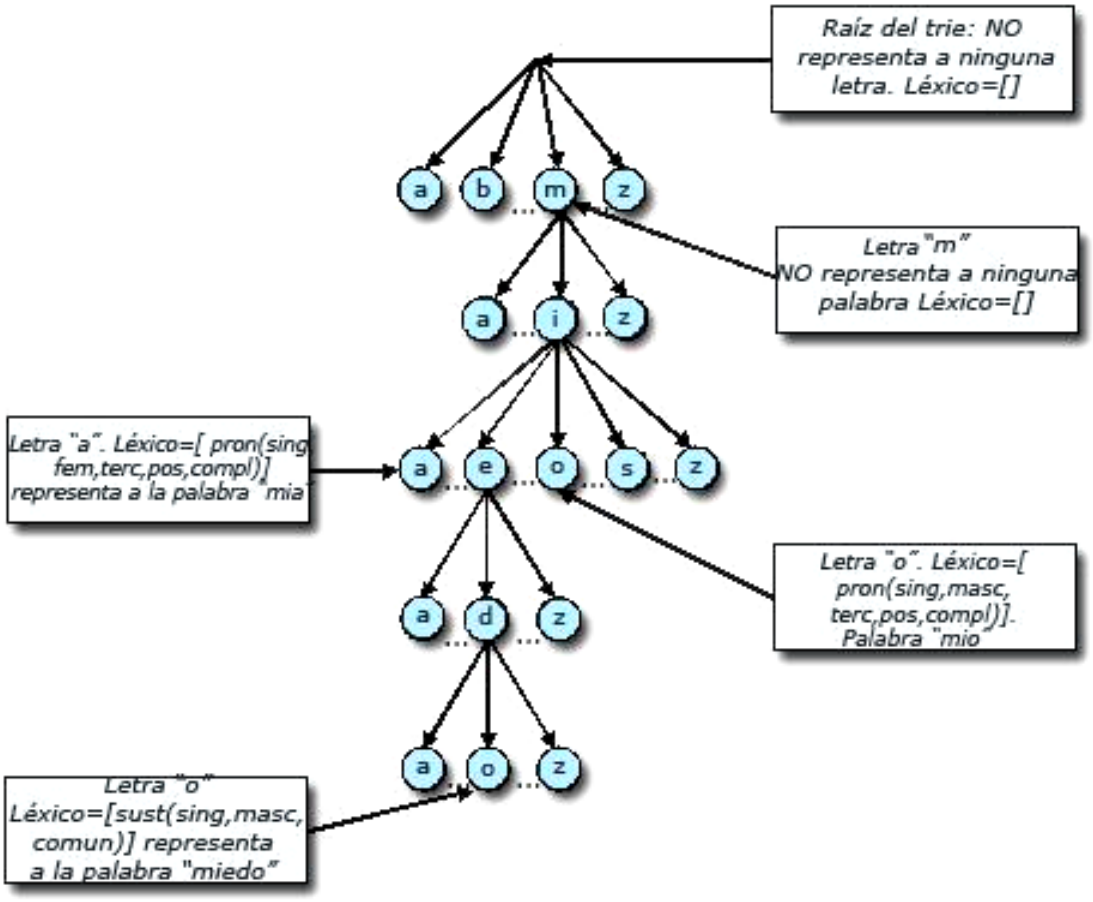


Figura 2.5.- Ejemplo de un *trie*. (imagen tomada de Moreno 1999:49)

Cuando se representa el léxico de cada palabra se hace mediante una lista estructurada, esto se puede lograr con un único acceso al diccionario; obteniendo toda la información de la palabra.

Un vector Ramas se basa en una estructura con un nombre "ramas" y con su número de argumentos igual al número total de posibles letras que pueda contener cualquier palabra almacenada en el diccionario. Cada nodo es un argumento del *trie* y tiene una estructura "nodoTrie

(Léxico2, Rama2)", cada nuevo nodo representa una letra en función de la posición que ocupe este argumento dentro de las ramas. Por ejemplo:

"El argumento número uno de Ramas referenciará a la letra "A", el número dos a la letra "B", el tres a la "C", y así sucesivamente".

Mediante las ramas se logra enlazar un nodo con todos los que desciendan de él. Algunos lenguajes de programación lo manejan así: "ramas (Rama1, Rama2…)", donde el elemento RamaJ, corresponde al nodo que desciende del actual, si no se encuentra la RamaX, entonces ya no se realiza la instancia.

2.5.3.4. Representación de un *trie* utilizando una tabla de dispersión

Cuando se ocupa una tabla de *trie,* se desaprovecha espacio cuando existe una rama con pocos descendientes, pues se reservan todas las posiciones posibles aunque sólo se ocupe una rama.

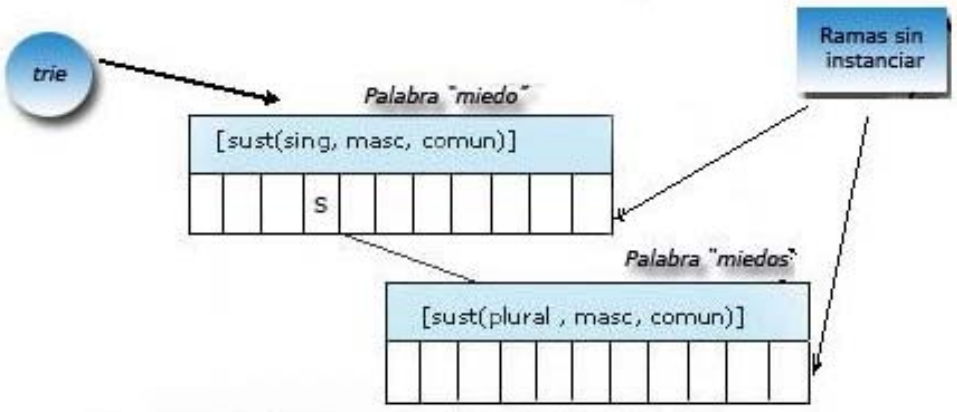


Figura 2.6.- Nodos del *trie* con la palabra "miedo" y "miedos". (imagen tomada de Moreno 1999:50)

Pues en el nodo final se representa la palabra "miedo" y sólo tiene una rama instanciada y la palabra "miedos" no tiene alguna instancia.

Una alternativa al uso de vectores espacialmente (en referencia al área que ocupa el vector), es utilizar listas de ramas ocupadas, la desventaja es el costo para llegar a las ramas (O(R), R = longitud de la lista). Pues se tiene que acceder a cada rama del *trie* perdiéndose tiempo.

Otra forma de resolver el problema del espacio ocupado es la tabla de dispersión o tabla *Hash* cerrada. El tipo de dato se basa en un vector B inferior al original con tamaño del total de letras que pueda contener una palabra guardada en el diccionario. Por ejemplo para un nodo vacío "B" se toma el valor de tres.

2.5.4. Lematización

Uno de los aspectos de la investigación en lingüística es el estudio del uso de la lengua en documentos escritos; se trata de identificar y analizar la aparición de determinadas construcciones, lo que, en gran medida, puede entenderse como una clase particular de lo que en informática se conoce como recuperación de información. En el ámbito de la recuperación de información se ha tenido desde siempre conciencia de la insuficiencia de las búsquedas exacta y parcial de las palabras de un texto, y también de la necesidad de incorporar información lingüística para una recuperación más completa. Las ya antiguas búsquedas con truncamiento parten de la hipótesis de que las distintas formas de una palabra se componen de una raíz fija acompañada de un sufijo o un prefijo variables; tal hipótesis suele ser bastante acertada para lenguas poco flexivas, pero resulta muy pobre con lenguas muy flexivas y con altas tasas de irregularidad. Las búsquedas con máscara, por similitud o con base en expresiones regulares no incorporan la adecuada información sobre la naturaleza morfológica de las palabras.

Un lematizador es un analizador morfológico el cual consiste en asociar variantes de la misma palabra con una forma raíz. La raíz puede ser pensada como la forma en que puede ser encontrada normalmente como una entrada en un diccionario. Por ejemplo, "vas","fui","iré" y "va" pueden ser asociadas con la forma raíz "ir", así como "caminante", "caminata", "camino", o "caminar" pueden ser asociadas a la raíz "camin". La lematización provee acceso al correspondiente lema en un lexicón semántico.

Un lematizador heurístico trata de remover ciertas marcas superpuestas en las palabras directamente en orden a descubrir su forma raíz. En teoría, esto implica descartar prefijos ("un", "dis", etc.) y sufijos ("ando", "iendo", etc.), aunque algunos lematizadores utilizados por motores de búsqueda solo remueven sufijos.

La estructura formal de un lexicón en línea es similar a un diccionario, consiste en un orden alfabético de lemas de acuerdo a la

siguiente estructura: los lemas son ordenados en orden alfabético de acuerdo a sus raíces. La raíz sirve como indicador tanto para el ordenamiento de los lemas durante la construcción del lexicón como para la búsqueda de cierto lema una vez que el primero se ha terminado.

2.5.4.1. Normas de lematización

Una vez terminado el proceso de lematización automática en un determinado texto, se continúa con los criterios optados para realizar la lematización manual, a pesar de que éstos no sean siempre aceptados por todos.

Las normas más importantes que se siguieron tomando como ejemplo un proyecto universitario en Europa denominado "Proyecto del diccionario de la navegación del siglo de oro[17]" acerca de la lematización son las siguientes (web.34):

1) Se ha lematizado normalmente con la grafía moderna incluso en los casos en que ésta no existía por ser esas formas más arcaicas, excepto con los nombres propios con los cuales se ha respetado, en general, su grafía.

2) La *ç* con cedilla se encuentra en las lematizaciones, (y lo mismo se haría en un diccionario), después de la serie "cu". Además, se ha adoptado el nuevo orden alfabético que la Real Academia Española ha aprobado: la "ch" dentro de la "c" y la "ll" está bajo la "l".

3) En la macroestructura se han conservado las formas normales de los repertorios lexicográficos, así, el singular para el sustantivo, el masculino singular para los adjetivos, el infinitivo para los verbos, etc.

4) Los reflexivos figuran bajo la forma verbal no reflexiva

5) En las parejas de adverbios en "-mente", la forma sin sufijo, se halla bajo el artículo respectivo con sufijo, pero dentro de la lematización y dentro del diccionario figura sin él.

6) Por lo que se refiere a la aceptación de una o de dos entradas, se han tomado las siguientes decisiones:

---

[17] El proyecto se planteó por la necesidad de realizar un tema específico del un Diccionario Histórico de la lengua española. Su objetivo es analizar todo el léxico referente a la naútica, mar, barcos y sus instrumentos.

a) Cuando en el plural de un sustantivo su acepción es divergente de la de su singular, se dan dos artículos distintos.
b) Si a una forma masculina le corresponde una femenina cuya terminación es distinta, como en "patrón/patrona", se lematizan las dos juntas, en una sola entrada, con el reenvío correspondiente de la palabra femenina.
c) Cuando un término tiene acepciones diferentes, si se emplea en masculino o en femenino, se ofrecen dos artículos.
d) No hay dos lemas distintos para los adjetivos que en el habla pueden usarse como sustantivos y, al contrario, los sustantivos que en el discurso pueden funcionar como adjetivos. En estos casos solamente figura una entrada.
e) En cuanto a la frontera entre adjetivos y participios pasados, y a pesar de que no sea muy convincente para todos, se ha preferido unirlos y dar un solo lema, excepto en los casos en que se trata de un tiempo compuesto por "ir" con el auxiliar.
f) De igual modo se actúa con los infinitivos y los infinitivos sustantivados que se han juntado en una sola entrada, menos aquellos en los que la Real Academia Española los ha separado y ha redactado dos artículos: uno como sustantivo y otro como verbo.

7) Para hacer más fácil la búsqueda de algunas formas (artículos, pronombres, contracciones, etc.) éstas se muestran en el prólogo de la obra lexicográfica y se indican los reenvíos respectivos, bajo las entradas en las que se encuentran para ayudar a los usuarios: "la", "las", "los" y "el" (artículos) bajo "el" artículo; "lo" artículo neutro, como independiente; "al" (preposición más artículo) independiente; "la", "las", "lo", "los", "le", "les" (pronombres) bajo "le"; "ello" pronombre neutro, como independiente; "del" (artículo más preposición) independiente; "dél" (artículo más pronombre) independiente; "esta", "estas", "este", "estos", "esto" bajo "este".

La técnica de lematización permite mantener la misma información semántica de los textos a tratar, disminuyendo el tamaño de los documentos a procesar. Además, al sustituir una palabra por su lema, se concentra la información semántica dándole el peso real a cada uno de los lemas que aparecen, de manera que se podría mejorar la eficiencia en la etapa de clasificación de textos.

Para conseguir una reducción del tamaño del corpus, no basta sustituir una palabra por su lema o raíz (ocuparía lo mismo en número de características), sino que se debe modificar la representación del documento. La forma de hacerlo consiste en representar cada lema

junto con el número de apariciones del mismo a lo largo del documento y no repetir el lema de la palabra en cada ocurrencia. Esta apreciación no es exclusiva para los lemas, puesto que ocurre lo mismo con las palabras. La tabla 2.5 muestra un ejemplo de reducción en el número de características diferentes de cada categoría a más de la mitad. En este ejemplo se utilizó un corpus relacionado a noticias en diferentes ámbitos en el mundo en un proyecto universitario de Europa.

Total palabras		*799.379*
	Palabras diferentes	*Lemas diferentes*
Economía	13.067	5.664
Europa	15.205	6.818
Sociedad	24.411	9.947
Deporte	27.908	14.926
Cultura	28.102	13.628
Mundo	16.122	5.415
Política	23.108	9.262
Total	*85.364*	*38.566*

Tabla 2.5.- Reducción del corpus.

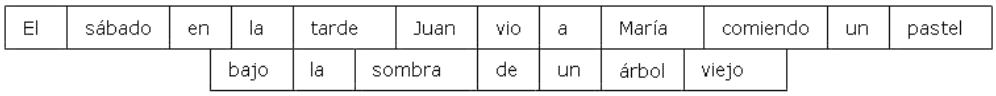
Una vez lematizado el corpus, existe la posibilidad de establecer una lista de lemas a eliminar. Para ello, se pueden utilizar los mismos parámetros usados en el caso de las palabras, es decir, la frecuencia de los lemas en los documentos o en las categorías, o bien se puede partir de la información sintáctica producida por el lematizador y eliminar aquellos lemas correspondientes a alguna categoría sintáctica concreta. Por ejemplo, se puede realizar una prueba teniendo sólo en cuenta los nombres, verbos, siglas y adjetivos, y dejando de lado el resto.

Con todos los puntos vistos en el tema "Técnicas y recursos de la Ingeniería Lingüística" -véase cap. 2.5- dentro de esta tesis, es posible construir un ejemplo sobre el uso de estas técnicas.

Se tiene el siguiente enunciado:

"El sábado en la tarde Juan vio a María comiendo un pastel bajo la sombra de un árbol viejo"

Enseguida se procede a aplicar el proceso de tokenización, resultando algo similar a lo que se muestra a continuación:



En este caso la derivación terminó con 19 *tokens*.

Ahora se buscará cada uno de los *tokens* dentro del lexicón ya definido, un ejemplo sencillo se muestra así:

Lexicón	
El	Artículo determinado
sábado	Sustantivo
en	Preposición
la	Artículo determinado
tarde	Adverbio de tiempo Sustantivo
Juan	Nombre propio
vio	Verbo
a	Preposición
María	Nombre propio
comiendo	Verbo
un	Artículo indeterminado
pastel	Sustantivo
bajo	Preposición Verbo Sustantivo Adjetivo
sombra	Sustantivo
de	Preposición
árbol	Sustantivo
viejo	Adjetivo calificativo Sustantivo

Hasta aquí todo parece sencillo, pero se puede observar que palabras como “bajo”, “tarde” o “viejo” pueden tener distintos significados por sí solos, pero dentro de la oración solo obtienen uno. A este problema se le llama ambigüedad.

Como ya se ha mencionado, dicho problema se disminuye en gran parte gracias a la ayuda de un *guesser*.

Una vez solucionado el problema anterior, se etiqueta cada uno de los *tokens* de acuerdo al lexicón. Ahora se presenta un ejemplo sencillo.

El_artdet sábado_sust en_prep la_artdet tarde_sust Juan_NP vio_vb a_prep María_NP comiendo_vb un_artind pastel_sust bajo_prep la_artdet sombra_sust de_prep un_artind árbol_sust viejo_adjcal.

2.6. Text mining

En los últimos años la minería de datos (*data mining*), ha experimentado un auge como soporte para filosofías de la gestión de la información y el conocimiento, así como para el descubrimiento del significado que poseen los datos almacenados en grandes bancos. Ésta permite explorar y analizar las bases de datos disponibles para ayudar a la toma de decisiones; además de facilitar la extracción de la información existente en los textos, así como para crear sistemas inteligentes capaces de entenderlos, a esto se denomina comúnmente como minería de textos (*text mining*).

La minería de textos es una de las más recientes áreas de investigación del procesamiento de textos. Su principal iniciador es el *Georgia Institute of Technology*, especialmente por el investigador director del *Technology Policy and Assessment Center* (TPAC), el Dr. Alan Porter. Ésta se define como el proceso de descubrimiento de patrones interesantes y nuevos conocimientos en una colección de textos, es decir, la minería de texto es el proceso encargado del descubrimiento de conocimientos que no existían explícitamente en algún texto de la colección, pero que surgen de relacionar el contenido de varios de ellos (web.92).

No debe ser confundida con las comúnmente conocidas herramientas de búsqueda de internet o la administración de las bases de datos. A diferencia de la minería de datos, la cual extrae cualquier tipo de datos dentro de grandes cantidades de información, la minería de textos es un procedimiento aplicado a grandes volúmenes de textos no estructurados. Después de que una búsqueda tradicional en algunos documentos es terminada, por ejemplo en formatos de texto, abstractos, o indexados, la minería de textos analiza la compleja relación entre dichos documentos para entregar un resultado final.

Este proceso consiste de dos etapas principales: una etapa de pre-procesamiento y una etapa de descubrimiento (web.92).

En la primera etapa, los textos se transforman a algún tipo de representación estructurada o semi-estructurada que facilite su posterior análisis, mientras que en la segunda etapa las representaciones intermedias se analizan con el objetivo de descubrir en ellas algunos patrones interesantes o nuevos conocimientos. La figura 2.7 ilustra este proceso.

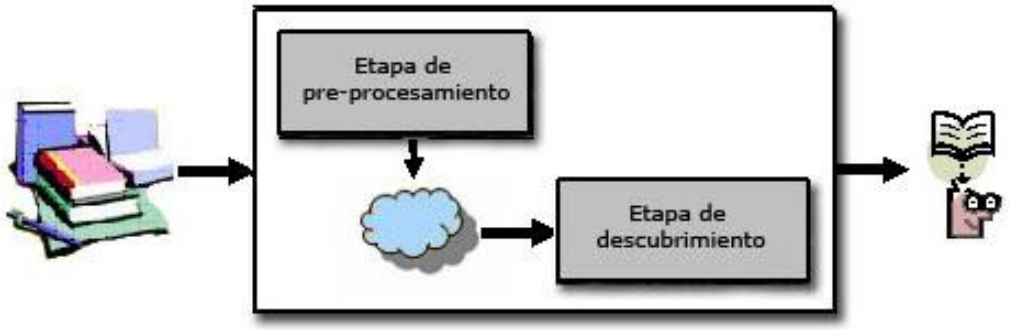


Figura 2.7.- Proceso de minería de textos (imagen tomada de web.92).

Dependiendo del tipo de métodos usados en la etapa de pre-procesamiento es el tipo de representación del contenido de los textos construida; y dependiendo de esta representación, es el tipo de patrones descubiertos (web.92).

Etapa de pre-procesamiento	Tipo de representación	Tipo de descubrimiento
Categorización	Vector de temas	Nivel temático
Full-Text	Secuencia de palabras	Patrones de lenguaje
Extracción de información	Tabla de datos	Relaciones entre entidades

Tabla 2.6.- Estado del arte de la minería de texto (imagen tomada de web.92).

La tabla 2.6 muestra los tres tipos de estrategias empleadas en los actuales sistemas de minería de texto. Como se observa, todos estos métodos limitan a un nivel temático o de entidad sus resultados, haciendo imposible descubrir cosas más detalladas como:

- Consensos, que por ejemplo, respondan a preguntas como: ¿Cuál es la opinión mayoritaria de los mexicanos sobre el gobierno del presidente Vicente Fox?.

- Tendencias que indiquen por ejemplo, si han existido variaciones en la postura de Fox con respecto a la educación.
- Desviaciones, que identifiquen por ejemplo opiniones "raras" con respecto al desempeño de la selección mexicana de fútbol.

Una idea para mejorar la expresividad y diversidad de los descubrimientos de los sistemas de minería de textos consiste en usar alguna mejor representación del contenido de los textos; por ejemplo, los grafos conceptuales (web.92.)

Esta solución involucra dos problemas diferentes -véase figura 2.8-:
- La transformación de los textos en grafos conceptuales.
- El análisis automático de un conjunto de grafos conceptuales.

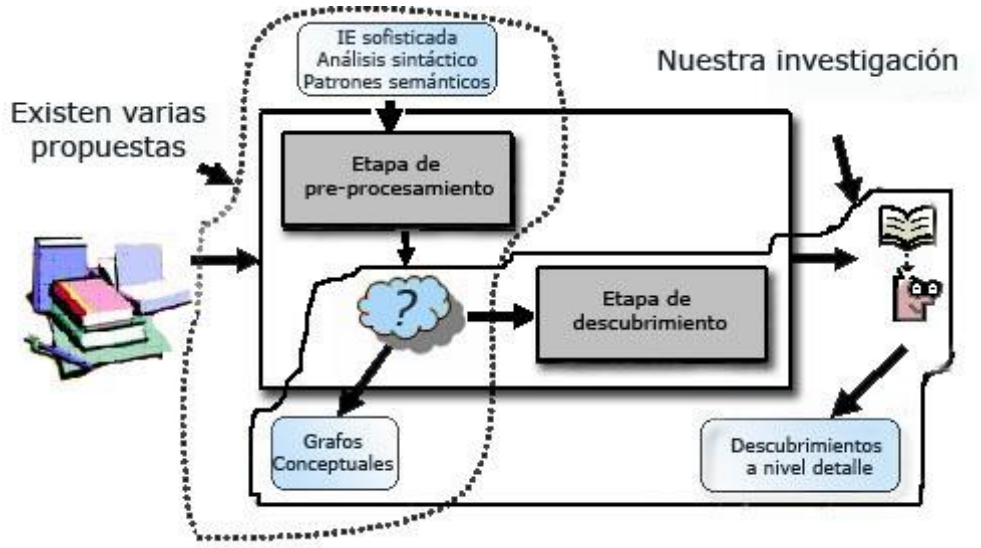


Figura 2.8.- Representación gráfica de los grafos conceptuales (imagen tomada de web.92).

La transformación de los textos en grafos conceptuales es un problema complejo vinculado con el análisis sintáctico y semántico de los textos. Algunos tipos de textos transformados automáticamente en grafos conceptuales son (web.92):

- Algunas partes de artículos científicos.
- Algunas partes de expedientes médicos.
- Algunas partes de casos legales.

Por su parte, el análisis automático de un conjunto de grafos conceptuales orientado al descubrimiento de nuevos conocimientos es

un problema poco estudiado. Solamente existen dos trabajos relacionados con el agrupamiento de grafos conceptuales (web.92).

2.7. Reconocimiento de entidades

2.7.1. Identificación de entidades

La identificación de entidades habla de la labor de identificar algún elemento de interés dentro del texto, como pueden ser fechas, lugares, sujetos o determinados verbos.

La manera más segura para cumplir este objetivo es utilizar un diccionario que contenga todo el catálogo de elementos buscados. Por ejemplo, tener una lista de compañías de *software* con su descripción, para posteriormente realizar una búsqueda de entidades con base en compañías de *software*, formando de tal manera la lista que se acote a todas las posibilidades de las entidades que se buscan.

- *Microsoft*
- *Sun*
- *Oracle*
- *Macromedia*

Identificar entidades semánticas de esta manera es sencillo, únicamente basta con buscar la palabra íntegra dentro del texto o en caso de los verbos buscar el verbo y sus variantes o buscar la raíz del verbo en caso de que el texto ya este lematizado. El verdadero problema de la identificación de entidades radica en buscar elementos que de alguna manera no se conocen, como por ejemplo identificar el nombre de una empresa dentro de un texto pero no sabiendo de que se trata del mismo, o identificar el lugar de un suceso pero sin conocer bien dicho lugar, o también podría ser identificar una fecha cuando ésta se puede presentar en varios formatos.

Para identificar elementos se utilizan técnicas donde se ubican los contenidos en base a reglas, estas reglas se utilizan antes de las reglas gramaticales y se ejecutan en forma recursiva. Por ejemplo para identificar una fecha:

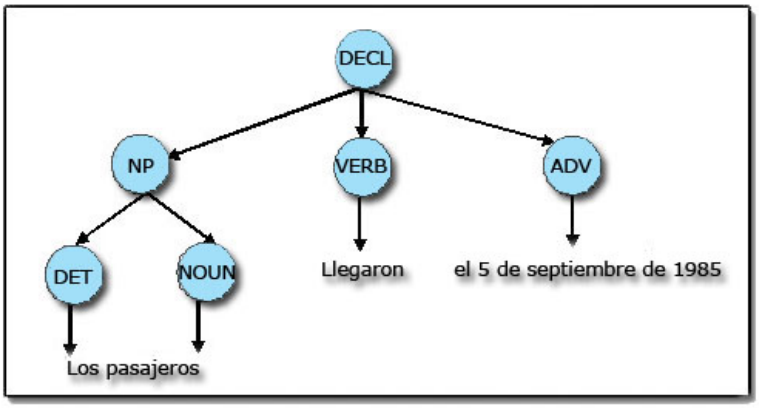


Figura 2.9.- Identificación de una fecha en base a reconocimiento de entidades.

El proceso consiste en identificar los factores semánticos y unirlos en una unidad léxica.

Lema	5_de_septiembre_de_1985
Origen	Adv. : 5 de septiembre de 1985
Descomposición	día : 5 prep.: de mes : septiembre prep.: de año : 1985
Tipo	fecha

Tabla 2.7.- Proceso de descomposición de las partes de una palabra.

En el caso de buscar un lugar como un lago o un mar generalmente se pueden ubicar gracias a que primero aparece alguna palabra describiendo el lugar.

"El lago de Chapala"

Una segunda manera de ubicar el elemento es encontrar el elemento "de" entre la descripción del lugar y el nombre.

Capítulo 3

3. Problemática

Hoy en día se cuenta con grandes volúmenes de información digitalizada a través de internet y también se sabe que cada vez son más los usuarios que tienen acceso a ella.

En otros tiempos (hace un par de décadas quizá) no era posible tener acceso a tanta información de una manera tan sencilla y ahora esto ha quedado solucionado con la aparición del internet, pero ahora aparece otro tipo de problema: es muy fácil entrar a internet pero no es tan fácil buscar la información y separar la que realmente es útil.

A través de la Recuperación de Información (RI) se facilita esta búsqueda, pero aún así no es suficiente ya que esta técnica pudiera traer la información que cumpla con las sentencias explícitas que le son suministradas para buscar, aunque no necesariamente sea la información útil que se requiere. Un ejemplo es que se puede definir "huracán" como palabra de búsqueda teniendo la idea de encontrar los huracanes más devastadores para México del año en curso, pero los documentos que traería como resultado serían todos aquellos en los cuales aparezca esta palabra, y pudiéndose encontrar títulos como; "Los huracanes más devastadores de los últimos veinte años en Haití", "la vida de Julio Salinas el huracán del norte" o incluso algo más gracioso como "El Huracán Ramírez venció en 3 caídas a *Blue Demon*". De esta manera se aclara que no toda la información que regresa es realmente útil, por decirlo de una forma, sólo se delimita la búsqueda de los documentos que obtuvo.

Hablando en específico se sabe que el desarrollo de *software* está en constante avance y en breve tiempo salen al mercado nuevas y mejores soluciones a los problemas, pero ¿qué tan fácil es estar al tanto de estos nuevos lanzamientos para poder dar mejores soluciones a través de estas herramientas?. En realidad si se quiere estar al día en este tema, primero es necesario identificar las fuentes para obtener las noticias de los desarrollos de *software* que están saliendo en el mercado, ya que hay que señalar que este tipo de información se encuentra en texto no estructurado, es decir se encuentra en formato libre, noticias, artículos etc., y no en bases de datos almacenadas y listas para ser examinadas a través de una consulta (*query*).

Después hay que leer toda esa información, verificar si es relevante o no y por último separarla y hacer un control que pueda almacenarse en una base de datos de productos de *software*, para tenerla a la mano cuando sea requerida y darle un uso provechoso que ayude a la mejora de las operaciones que se están realizando. Para llevar a cabo esto,

primeramente hay que tomar en cuenta el tiempo que se lleva en realizarlo, producido por un grupo de personas dedicadas exclusivamente a esto o tal vez contratar a una empresa que lo realice externamente.

Lo descrito anteriormente no es la única forma de realizarlo, ya que a través de los Sistemas de Extracción de Información se proporciona la pauta para realizarlo de una manera automática y rápida. Los sistemas de extracción de información han tenido hasta el día de hoy fines científicos o de investigación y no se realizan tanto de forma comercial. En esta tesis se propone a estos sistemas como herramientas de gran ayuda para las organizaciones ya que se les permite hacer búsquedas de una forma automática.

El proceso que realizan los sistemas de extracción de información es el siguiente: primero se buscan los textos candidatos, en seguida se realiza un análisis lingüístico en ellos que consiste en etiquetar dichos textos para después aplicarles un análisis, entre otros el morfológico, el sintáctico, y un proceso de desambiguación. Todo esto para separar la información útil de la que no lo es. Por último organizar la información y presentarla. Si se ve desde este enfoque es muy fácil tener un control por semana o por día de los nuevos lanzamientos de *software* o cualquier tipo de información que la organización requiera.

Cabe mencionar que la elaboración de un sistema de extracción de información es muy compleja por el tipo de análisis que se realiza a los textos, así que es muy recomendable que en su elaboración participe un experto en el tema a buscar (en este caso productos de *software*) y un lingüista. Hasta hoy en día los sistemas de extracción de información tienen un promedio de 75% de recuperación (*recall*) y precisión (*precision*), que son las dos principales medidas para estos sistemas y aunque no sea posible realizar un trabajo (hasta ahora) al 100% sí se está proponiendo una herramienta que ahorra el 75% del trabajo.

Resumiendo, esta tesis está enfocada a mostrar otras alternativas a las organizaciones, en específico los sistemas de extracción de información, así como la manera en que funcionan y cómo se pueden utilizar.

Por otro lado, pero continuando con el mismo tema, con la evolución constante de la tecnología se ha incrementado el número de programas de *software* que hoy en día ofrecen los proveedores y desarrolladores con el único fin de resolver los problemas de la vida diaria en las organizaciones, pues gracias al desarrollo del mismo se pueden realizar

tareas de manera más fácil y eficiente, ya que se automatizan las labores de tal modo que se logran alcanzar las metas en forma y tiempo, pues al tener herramientas de *software* cada uno de los integrantes de la organización ayudará a lograr los grandes objetivos empresariales. Aunado a esto existe otro gran problema acerca de la gran cantidad de información en línea que se encuentra actualmente en internet, pues existen en el mundo publicaciones de todo tipo como noticias científicas, informativas, información formal e informal y si se propusiera tener información actualizada de un cierto tema como es el objetivo de esta tesis (obtener información especifica acerca del nuevo *software* que sale al mercado), sería casi imposible y carísimo el hecho de que unas personas lean las noticias y extraigan la información que interesa como el título de la noticia o la compañía que lanzó tal o cual producto.

La idea de tener información acerca de las noticias de lanzamientos de *software* es buena, pero debido a que también se debe buscar información en textos que no tienen estructura o están en formato libre, se decidió basar la investigación en estas últimas, puesto que la forma de redactar las noticias no tiene un estándar en particular, cada persona tiene una idea diferente de los entes y por tal motivo proveen la información con respecto a su forma propia de pensar de acuerdo a su contexto social, cultural, histórico, etc. Por este motivo existen "n" formas de expresar las ideas.

En informática se dice que el texto en formato libre no tiene una estructura porque la información de estas noticias no se encuentra disponible en una base de datos. Y una gran problemática que se halla en el análisis de la tesis es que muchas noticias publicadas en internet carecen de una ortografía adecuada o de reglas gramaticales.

Por otro lado se indica que a pesar de que no tienen una estructura o están en formato libre, de acuerdo a los lingüistas el lenguaje en sí tiene su estructura propia, pues se compone de arreglos como los enunciados formados por palabras y estas a su vez por letras. Las palabras también están compuestas de lemas, prefijos y sufijos que pueden modificar el significado de una idea y tienen una gramática y una semántica de acuerdo al contexto. Las oraciones asimismo están compuestas por sustantivos, verbos, pronombres, preposiciones, adjetivos, conjunciones, artículos, etc. Y esta forma en que está estructurada la información que se encuentra en las noticias hace muy complicado el estudio y análisis de la estructura del lenguaje en los textos con los cuales se trabajará, pues se requiere de un profundo razonamiento en cada patrón hallado en el corpus formado por las

noticias de nuevo *software*. Para cada tema en particular se requiere de un minucioso escudriñamiento pues existen pistas, reglas y patrones que ayudan al proceso comunicativo y con esto encontrar datos útiles para cumplir con la necesidad de encontrar información en textos que manualmente llevaría mucho tiempo.

Por estas razones se cree que así las organizaciones o compañías tendrían a la mano información actualizada acerca del nuevo *software* para ofrecer mayores utilidades a sus clientes; además podrían desarrollar sus actividades de manera óptima pues tendrán a la mano las herramientas con funcionalidades de última tecnología y vanguardia.

Los puntos a resolver son:

- Se pueden utilizar recursos de la ingeniería lingüística dentro de las organizaciones comerciales (tokenización, analizadores sintácticos, etiquetado morfológico, lexicones, lematización) y no sólo en el campo de la investigación.
- Se puede abrir un nuevo campo de investigación y desarrollo en conceptos como buscadores en internet pero más exactos (debido a los mismos recursos lingüísticos).
- Generar un sistema de búsqueda por medio de recursos lingüísticos es una ventaja competitiva dentro de la organización, lo que le ayudará a estar un paso adelante de su competencia.
- Cómo automatizar la búsqueda en texto libre con SEI.

Capítulo 4

4. Hipótesis

La principal hipótesis en esta tesis es el hecho de que es posible utilizar sistemas de extracción de información específica en textos no estructurados como herramienta para las organizaciones empresariales y verificar su factibilidad para su uso dentro de las organizaciones.

Debido a la gran abundancia de información disponible de manera electrónica y a la gran necesidad de información específica que ayude a la toma de decisiones, se han enfocado los esfuerzos a la extracción de información en textos en formato libre, los cuales son muy extensos y si se asentara la lectura, el análisis y la captura de esta información en manos de personas, tomaría mucho tiempo, dinero y esfuerzo al hacer dichas actividades. En cambio, si se enfocan los esfuerzos a la explotación de las tecnologías de información de manera automatizada, se obtendrían resultados de manera más rápida y eficiente, todo mediante las metodologías que utilizan estándares mundiales establecidos en las MUC (Conferencias de Comprensión de Mensajes) –véase cap. 6.2- utilizadas por investigadores reconocidos.

Se analizará el fenómeno de lanzamientos de nuevo *software*, así como sus respectivas versiones mediante recursos lingüísticos para extraer la información que se requiere. De esta forma se observarán cada uno de los criterios que abarcan el formato libre.

Con el fin de llevar a la práctica todo lo descrito en los capítulos anteriores, se desarrollará una aplicación que utilice la Extracción de Información como una solución organizacional, utilizando datos no estructurados, como ya se ha citado anteriormente, llamada "SEINS".

Como ya también se ha mencionado, primero se recopilará el corpus[18], así que se ha comenzado con 120 noticias referentes al momento en que nuevas versiones de *software* han sido lanzadas (en tiempo pasado) o que se espera que salgan (en un futuro), con el fin de plantear una solución a la problemática siguiente:

Las organizaciones necesitan estar al día en lo medida de lo posible en dos aspectos; tecnología y *software*, ya que depende de estos el buen uso y aprovechamiento de la información, debido a que en el presente es el recurso más codiciado por todas las personas u organizaciones, porque se dice por ahí que la información es poder.

18 Se ha definido "corpus" como una colección muy grande de textos en formato libre, el cual contendrá ejemplos de noticias de lanzamientos de nuevo *software*.

Partiendo de esta idea y pensando en que la mayoría de las organizaciones no están en capacidad de desarrollar el *software* que esté a su medida, se necesita encontrar en el mercado la mejor solución a sus requerimientos.

Como ejemplo podría mencionarse que por tales motivos, deberán estar al pendiente de las publicaciones que se hagan de nuevo *software* para encontrar la mejor opción, así que se puede cuestionar lo siguiente: ¿cuántos productos pueden liberarse en una semana?, y de esos ¿cuántos de verdad se acercan a las necesidades que tiene la organización?. Como se acaba de demostrar, se encuentran varias problemáticas que principalmente son el exceso de información (incluso mucha que no es de importancia), y aún más, la falta de tiempo para la selección de la información que de verdad requiere de atención.

Por tal motivo se desarrollará una aplicación que seleccione de entre todas esas noticias disponibles en la *web*, la lista de los productos de *software* que se han liberado o se liberarán próximamente, sus respectivas versiones, la compañía que lo libera, y una pequeña descripción, además de un link para poder consultar a detalle las noticias que resulten de interés.

4.1. Limitaciones

Para la realización de los sistemas de extracción de información basados en expresiones regulares como es el caso actual, se necesita hacer un amplio uso de conocimientos lingüísticos para comprender lo que se encuentra en cada texto y después poder extraer entidades de él, por tal motivo es indispensable para su construcción contar con la participación de un experto en lingüística, además que para su elaboración se tiene relativamente un alto costo en tiempo y recursos a corto plazo pero con beneficios a largo plazo.

El sistema de extracción de información esta creado en un tema en específico de búsqueda (este tema como ya se ha visto trata de lanzamientos de *software*), así que si éste se cambia es necesario realizar un sistema nuevo.

Hoy en día es posible obtener excelentes resultados de búsqueda aunque estos conllevan un análisis también más complejo, y en esta situación solo fueron utilizadas expresiones regulares por lo que los resultados nos son tan óptimos como en otros sistemas de extracción

de información que utilizan algoritmos matemáticos complejos y métodos de aprendizaje de computadora.

No es posible para el método que se presenta en esta tesis, distinguir fácilmente entre noticias de *software* y *hardware* debido a la similitud de la redacción y del ámbito semántico en el que trabajan ambos (mundo de la informática).

Capítulo 5

5. Corpus textuales

5.1. Definición

Un corpus textual es un conjunto de textos reales (se pueden encontrar en la vida diaria), que cumplen con ciertas características y que forman parte de un mismo idioma, además cuentan con soporte informático, pudiendo llegar a tener varios millones de palabras.

Existen diferentes tipos de corpus. Un corpus textual se diferencía esencialmente de un corpus oral debido a que éste último está concebido a partir de información recopilada de grabaciones, diálogos, etc., para después ser transmitida a texto. En cambio, los textos escritos que pasan a formar parte de un corpus textual originalmente son concebidos en texto.

Por otra parte, un corpus textual se diferencía de un corpus de referencia porque aunque ambos están constituidos por textos, un corpus textual está conformado por textos íntegros (no han sufrido modificaciones), en cambio el corpus de referencia se integra por fragmentos escogidos de textos.

Algunos ejemplos de corpus textuales en español se encuentran en:

- Corpus Diacrónico del Español (CORDE): está integrado por todos los tipos posibles de español a través del tiempo y lugar hasta 1975 (www.rae.es).
- Archivo de textos hispánicos (Arthus): contiene textos de diferentes épocas del español.

La ventaja de los corpus radica en la posibilidad de automatizar las labores que se realizan con ellos, es decir, si se tiene que buscar una palabra dentro de un corpus de datos, esto lo podrá hacer un programa en cuestión de minutos o segundos con una precisión exacta sin espacio a errores, de esta manera se pueden programar tareas complejas que de otra forma se llevaría mucho tiempo hacerlo manualmente.

La historia de la lingüística de corpus comienza a finales de los años cuarenta con el Padre Busa con una obra de concordancias de los escritos de Santo Tomás de Aquino, pero es hasta la década de los sesenta cuando se puede considerar que comienza su auge. Al principio los corpus eran de extensiones pequeñas y se contaban las cantidades de palabras en miles pero hoy en día se cuentan en millones.

Los principales requisitos de un corpus de calidad son:

- El texto debe de estar en un soporte electrónico.

- El corpus debe ser accesible por conexión directa a la computadora donde se almacena o mediante red.
- Deben normarse de manera adecuada los canales para acceder al corpus y su uso, respetando los derechos de autor.
- Los textos deben de encontrase codificados de forma estándar.
- Los textos deben de tener un marcado adecuado que facilite su procesamiento.

Los corpus vienen a facilitar labores que de otra manera se harían manualmente, con su consiguiente grado de error humano y lentitud en el proceso. Por ejemplo: si se está investigando la cantidad de veces que se presenta una determinada estructura de oración, para su elaboración se tendría que hacer uso de una persona que leyese texto por texto identificando las estructuras de manera visual, esto llevaría mucho tiempo y estaría sujeto a una gran cantidad de errores. En cambio, si se tiene un corpus con las características antes mencionadas solo bastaría con elaborar un programa que realice la tarea de manera automática, esta alternativa ahorraría una gran cantidad de tiempo y trabajaría en un margen de error de 0%.

Una parte esencial de los corpus es su etiquetado, esto se refiere a que las colecciones de texto que se elijan deberán ser sometidas a procesos mediante los cuales las palabras se engloban dentro de etiquetas, resaltando el hecho de que se está refiriendo a un verbo, sujeto, articulo, sustantivo, etc.

La gran ventaja de los corpus electrónicos es que son reutilizables de una manera extremadamente sencilla y de igual forma se pueden unir varios corpus para formar corpus más grandes.

5.1.1. Corpus de entrenamiento

Un corpus de entrenamiento es una fracción de algún corpus o la creación de un corpus que se hace con el fin de tener un espacio en el cual se puedan hacer pruebas de alguna índole para el desarrollo de tecnologías lingüísticas. El tamaño del corpus estará dado por el alcance que se quiera dar al desarrollo, por ejemplo en un sistema de traducción automática el corpus estaría formado por los distintos tipos de textos que el traductor pretendería trasladar a otro idioma.

5.2. Etiquetado con XML

XML (*Extended Markup Language*) fue aprobado por la *World Wide Web Consortium*[19] (W3C) en su versión 1.0 en febrero de 1998, el cual es un metalenguaje, es decir un lenguaje que permite crear lenguajes de etiquetado a partir de él. XML es un conjunto de reglas para diseñar formatos de texto que permitan estructurar los datos usando etiquetas para delimitarlos. XML no es un lenguaje de programación.

XML es similar a otro subconjunto de SGML (*Standard Generalized Markup Language*) llamado HTML (*HyperText Markup Language*) el cual está especializado en la presentación de documentos para la *web*, mientras que XML es de igual forma un subconjunto de SGML pero especializado en la gestión de información para la *web*.

Los objetivos de diseño para XML son:

- XML debe ser utilizable directamente sobre internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML (*Standard Generalized Markup Language*).
- Debe permitir escribir programas que procesen documentos XML.
- El número de características opcionales en XML debe ser mínimo, preferentemente cero.
- Los documentos XML deben ser legibles a las personas y razonablemente claros.
- El diseño de XML deberá prepararse rápidamente.
- El diseño de XML deberá ser formal y conciso, es decir que sea simple.

5.2.1. ¿Qué es HTML y su diferencia con etiquetado XML?

HTML (*HyperText Markup Language*) fue creado en 1991 por Tim Berners-Lee[20]. Originalmente HTML fue desarrollado para que los

---

[19] W3C es una asociación internacional formada por organizaciones miembros del consorcio, personal y el público en general, que trabajan conjuntamente para desarrollar estándares w*eb*.

20 Científico británico del Laboratorio Europeo de Física de Partículas (CERN), decidió desarrollar un método eficiente y rápido para intercambiar datos entre la comunidad científica en internet. Para ello, combinó dos tecnologías ya existentes (el hipertexto y el protocolo de comunicaciones de Internet), creando un nuevo modelo de acceso a la información intuitivo e igualitario. Las famosas tres W han hecho posible que aprender a utilizar la Red sea algo al alcance de cualquiera.

investigadores pudieran intercambiar documentos en línea con facilidad y sin la preocupación de reglas de etiquetado tan estrictas. HTML está más enfocado en la visualización del documento que en su estructura. Por lo tanto, se eligió un número limitado de marcas que permitían etiquetar partes sencillas tales como:

- <P> - párrafo
- <B> - negritas
- <H1>- encabezados
- <Li> - listas, entre otros.

El DTD (Definición de Tipo de Documento) –véase cap. 5.2.5- de un archivo HTML es sencillo y está integrado al propio navegador para la visualización de documentos digitales.

5.2.2. Características de HTML

Algunas características importantes del lenguaje HTML son:

- Sencillez en su manejo.
- Facilidad para crear hipervínculos.
- Permite incorporar texto, imágenes, video, audio y otros formatos.

La diferencia entre HTML y XML, como lo muestra la figura 5.1, es que ambos fueron creados con fines diferentes. XML fue creado para describir y llevar datos, enfocándose en lo que los representan. HTML fue creado para desplegar datos y está enfocado en la imagen visual de los datos. En conclusión, HTML es para desplegar información y XML es para describirla.

XML y HTML Ambas son aplicaciones de SGML con fines diferentes.

Figura 5.1.- Diferencias básicas entre html y xml.

5.2.3. ¿Qué es SGML y su diferencia con etiquetado XML?

SGML (*Standard Generalized Markup Language*) es el estándar internacional para la definición, la estructura y el contenido de diferentes tipos de documentos electrónicos. Es una norma ISO8879 derivada de GML (*Generalized Markup Language*), la cual es una norma anterior adoptada de IBM como parte de su sistema de procesamiento de textos. Se dice "generalizado" porque el etiquetado debe describir la estructura de un documento y otros atributos. SGML es un estándar internacional, no propietario y abierto, que provee un método para la descripción de la estructura de documentos basándose en la relación lógica de sus partes. Provee una codificación estándar para la transmisión de documentos entre sistemas de computadoras diferentes: tales como plataformas, soportes físicos, lógicos y diferentes sistemas de almacenamiento y presentación, con independencia de su grado de complejidad.

Para el etiquetado se utiliza un conjunto de caracteres basado en el estándar ASCII, reconocido de manera universal por cualquier tipo de plataforma y sistema. Los caracteres especiales no contemplados en ASCII se transforman en representaciones ASCII y se denominan referencias de entidad. En SGML todo el etiquetado es lógico, es decir, en lugar de utilizar códigos crípticos (^C, ^D, etc.) se utilizan nombres de elementos delimitados por marcas que indican el comienzo y el final de los objetos lógicos. Estos delimitadores permiten que el *software* reconozca qué caracteres deben ser leídos en modo de etiqueta y qué otros en modo contenido. A continuación se muestra un ejemplo:

El modo etiqueta será leído así:

<title> “Consejo técnico” </title>

Que implica que el “consejo técnico” es un título y está dentro de la etiqueta <title>. Mientras que el modo contenido en la misma oración sólo diría “Consejo técnico”.

XML está basado en el estándar de SGML, como lo muestra la figura 5.2. Es una versión abreviada de SGML optimizada para su utilización en internet. Esto significa que con él es posible definir tipos de documentos propios (definir etiquetas propias) por lo que ya no se estaría dependiendo de un único tipo de documento HTML.

A XML hay que considerarlo como un SGML simple y optimizado para su utilización en internet. Los diseñadores de XML intentaron dejar fuera sólo aquellas partes que raramente se utilizan.

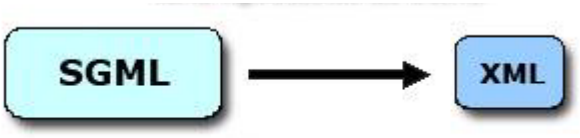


Figura 5.2.- Aplicación de SGML.

5.2.4. Tipos de etiquetado XML en general

Todos los documentos de XML deben llevar en la primera línea la versión de XML que se está utilizando. Una etiqueta comienza con el símbolo menor que "<" y termina con el símbolo mayor que ">". Después del símbolo de menor que, se escribe el nombre de la etiqueta, por ejemplo <nota>.

Hay dos tipos de etiquetas:

- Las que tienen contenido: debe haber una etiqueta de principio y una de fin, a la etiqueta de fin se agrega el símbolo "/" que siempre va después del símbolo "<". Ejemplo <nota>……</nota>.
- Las que no tienen contenido. No necesariamente tienen etiqueta de fin.

Para XML hay una diferencia entre las letras minúsculas y las letras mayúsculas, por lo que no será lo mismo <nota>…….</nota> que <Nota>…….</nota>, ya que esto último provocaría un error.

Por otra parte, si se tienen varias etiquetas seguidas, éstas deben estar organizadas o identadas.

- …….. este es correcto.
- …… sin embargo, éste no lo es.

Por otra parte XML tiene caracteres especiales reservados y cuando se quieran utilizar deberá escribirse lo siguiente:

- < <
- > >
- & &
- &quot "
- ' '

5.2.5. El DTD y las partes que lo conforman

La definición de la estructura y el contenido de un tipo de documento se realizan en un DTD (*Document Type Definition*). Es una definición de los elementos que puede haber en el documento XML y su relación entre ellos, sus atributos, posibles valores, etc. Es una especie de definición de la sintaxis del documento.

Los DTD no son mas que definiciones de los elementos que puede incluir un documento XML, de la forma en que deben hacerlo (cuáles elementos van dentro de otros) y los atributos que se les puede dar.

5.2.6. Definición de tipo de documento (DTD)

Se conoce principalmente como DTD a su traducción del inglés "*Document Type Definition*" y "*Document Type Declaration*", o definición (declaración) del tipo de documento.

Un DTD tiene definidas las entidades, elementos, atributos y lo más importante, las reglas tanto de la asociación entre entidades y elementos como del etiquetado en general. Sin ésta no se podría dar un buen manejo al texto, además de que sirve para que se comprendan mejor las etiquetas empleadas.

Algunas características de un DTD son:

- El DTD define las reglas correspondientes a las etiquetas que se han creado para un corpus.
- Un DTD define los nombres de las etiquetas y el modelo de contenido (por ejemplo, el orden de las ocurrencias y las reglas de anidación).
- El DTD se escribe en SGML y se representa como un simple archivo en el sistema.
- El DTD consta generalmente de tres partes: una etiqueta inicial, el contenido y una etiqueta final. El nombre del elemento aparece en las etiquetas inicial y final.

Además un DTD indica:

- Qué elementos pueden ser utilizados en un tipo de documento específico.
- Cuáles son obligatorios y cuáles opcionales.
- Cuáles son repetibles y cuáles no.
- En qué orden deben aparecer.
- Cómo deben anidarse los elementos que conforman un documento.

El DTD utiliza una sintaxis especial para definir la estructura de un tipo de documento, básicamente son cinco etiquetas que se definen a continuación:

<!DOCTYPE (….)> En éste se declara el tipo de documento.

<!ELEMENT (….)> En éste se hace la declaración de los elementos.

<!ATTLIST (….)> Aquí se declaran los posibles atributos de un elemento.

<!ENTITY (….)> Se declara una entidad mediante una cadena de texto más reducida.

<!NOTATION (….)> En este se declara el sistema por medio del cual se deben interpretar los datos.

Cada declaración contiene <! > y debe tener las siguientes condiciones:

- El elemento raíz del documento es el mismo tipo de documento definido en el DTD.
- Los elementos permitidos como contenido de otro elemento se respetan.
- Los elementos no pueden tener atributos no declarados en el DTD.

5.2.7. ¿Qué son las etiquetas y sus atributos y qué es el etiquetado de corpus?

Una etiqueta es una marca que delimita una región del documento en los lenguajes basados en XML.

Los atributos proveen información adicional que caracteriza, define o especifica los elementos de una etiqueta. Hay tres tipos de atributos: el tipo cadena, el conjunto de los tipos 'tokenizados' y los tipos enumerados.

El tipo cadena puede tomar cualquier cadena literal como valor. Los tipos 'tokenizados' poseen restricciones léxicas y semánticas variables. Los atributos enumerados pueden tomar una de las listas de valores proporcionadas en la declaración.

El valor del atributo va entre “”, si el valor del atributo contiene “” se usan entonces las comillas simples ‘’, si hay más de un atributo estos deben ir separados por espacios. Un atributo puede ser visto también como un elemento. Una etiqueta nunca puede tener dos atributos con el mismo nombre.

La sintaxis es la siguiente: nombre_atributo= “valor”.

El etiquetado de un corpus con XML consiste en poner marcas mediante las etiquetas para representar diferentes elementos, como pueden ser el título, el autor, los verbos, todo un párrafo en específico, etc.

5.2.8. Ejemplo de XML

Un ejemplo muy sencillo de un etiquetado con xml y su respectivo dtd se muestra a continuación:

```
<?xml version="1.0"?>
<libro>
        <titulo> Cien años de soledad </titulo>
        <disponible tiempo="24" unidad="horas"/>
        <autor> Gabriel García Márquez </autor>
        <formato> Rústica </formato>
        <publicacion>1967 </publicacion>
        <precio cantidad="9.99" moneda="euro"/>
        <descuento cantidad="5"/>
        <enlacelibro href="/exec/ISBN/84-473-0619-4"/>
</libro>
```

Su DTD correspondiente:

```
<!ELEMENT autor ( #PCDATA ) >
<!ELEMENT descuento EMPTY >
    <!ATTLIST descuento cantidad CDATA #REQUIRED >
<!ELEMENT disponible EMPTY >
    <!ATTLIST disponible tiempo CDATA #REQUIRED >
    <!ATTLIST disponible unidad CDATA #REQUIRED >
<!ELEMENT enlacelibro EMPTY >
    <!ATTLIST enlacelibro href CDATA #REQUIRED >
<!ELEMENT formato ( #PCDATA ) >
<!ELEMENT libro ( titulo | disponible | autor | formato | publicacion |
precio | descuento | enlacelibro )* >
<!ELEMENT precio EMPTY >
        <!ATTLIST precio cantidad CDATA #REQUIRED >
        <!ATTLIST precio moneda CDATA #REQUIRED >
<!ELEMENT publicacion ( #PCDATA ) >
<!ELEMENT titulo ( #PCDATA ) >
```

Capítulo 6

6. Sistemas de extracción de información

La "Extracción de Información" difiere de la "Recuperación de Información" en que su objetivo no es extraer documentos, sino extraer información útil dentro de los documentos (Jackson y Moulinier: 2002).

A diferencia de muchas formas ambiciosas del PLN, los programas de extracción de información analizan pequeños conjuntos del texto proporcionado, dichos conjuntos contienen ciertas palabras "disparadoras", y después procuran completar una forma simple que represente los objetos o eventos de interés, como un formulario, tabla, etc.

6.1. Definición

La extracción de información es una disciplina dedicada a procesar grandes cantidades de información con el objetivo de extraer únicamente los datos que cumplan con un perfil dado. Dicho de manera un poco más extendida, la extracción de información se dedica a analizar grandes cantidades de información que procesadas manualmente sería una labor muy tardada y sometida a una gran cantidad de errores humanos, es por ello que con ayuda de diversas herramientas informáticas que tienen su fundamento en la lingüística, la extracción de información posibilita el recuperar texto en un banco de datos que contenga el perfil deseado, pero no solamente porque presente palabras que describan el perfil de la búsqueda, sino porque las herramientas informáticas a partir de recursos lingüísticos serán capaces de ubicar los textos en su significado.

Los sistemas de extracción de información se engloban dentro de la Inteligencia Artificial y específicamente del Procesamiento del Lenguaje Natural. Estos extraen y organizan la información relevante o de interés en un conjunto de textos en lenguaje natural llamado corpus; estos en conjunto con la información definida a extraer forman el dominio. La extracción de información requiere un análisis morfológico, léxico y sintáctico del corpus y se basa en las entidades y relaciones en el marco de un dominio determinado.

Para su mejor explicación se citará un ejemplo práctico[21] dentro de la tesis:

[21] Cabe destacar que el texto de ejemplo se muestra de manera íntegra conforme fue obtenido de internet, por lo tanto las faltas de ortografía que aparezcan en dicha noticia son ajenas a la tesis.

Macromedia presenta Flash Communication Server
(01/04/2003 11:04): La actualización incluye HTTP Tunneling, soporte Linux, y una edición gratuita para desarrolladores.

DIARIO TI: Macromedia anuncia la inmediata disponibilidad de Macromedia Flash Communication Server MX 1.5, que incluye la funcionalidad para ampliar su posición como la primera plataforma que permite aplicaciones dinámicas de audio y vídeo interactivas y fascinantes tales como vídeo bajo demanda, emisión de eventos en vivo, charlas con una webcam, y grabación de mensajes de vídeo.

http://www.diarioti.com/gate/n.php?id=748

El *software* de extracción de información deberá tener la capacidad de ubicar "01/04/2003 11:04" como fecha y hora, "*Flash Communication Server"* como un *software*, "*Macromedia*" como una compañía, "presenta" o "inmediata disponibilidad" como la acción de dar a conocer un nuevo producto (indicador de lanzamiento o *trigger word*). Estos detalles serán acumulados con el objetivo de contrastarlos con plantillas de extracción previamente definidas, que no son otra cosa que requisitos que deberá llenar el texto que se esté analizando para poderlo identificar como información que forme parte de los datos que se están buscando (Bordón y Avanzo).

Fecha:	01/04/2003 11:04
Compañía:	Macromedia
Producto:	Flash Communication Server
Acción:	"presenta" o "inmediata disponibilidad"

Un sistema de extracción de información busca información específica en un documento según normas predefinidas (específicas del tema). Las normas son específicas de un área temática dada. Por ejemplo, si el área temática son las noticias sobre ataques terroristas, las normas pueden especificar que el sistema de extracción de información debería identificar:

- la organización terrorista que participa en el ataque.
- las víctimas del ataque.
- el tipo de ataque y
- la restante información de este tipo que puede esperarse en un documento típico del área temática.

Tales sistemas se construyen, por lo común, manualmente para una sola área temática, lo que requiere una gran cantidad de trabajo de expertos (web.02).

Una de las estrategias más comúnmente adoptadas es la Recuperación de Información, en la cual se obtienen documentos con información significativa, pero una gran diferencia con la Extracción de Información es que ésta última es una técnica de Inteligencia Artificial en la que se obtienen hechos de los documentos, es decir, extrae y organiza la información relevante e ignora la irrelevante. Las dos técnicas, son, por tanto, complementarias.

Los sistemas de recuperación de información se encargan de procesar una colección de textos y entre todos ellos seleccionar aquellos que contengan algún término relacionado con la pregunta, descartando los que no estén relacionados.

Los sistemas de extracción de información, al contrario que los sistemas anteriores, parten de una colección de textos pertenecientes todos a un mismo dominio y que contienen información considerada relevante para la aplicación. Estos sistemas tienen como objetivo principal localizar en los textos determinada información para poder rellenar una base de datos a la cuál sea posible hacer preguntas. Con ello se consigue transformar información no estructurada en información estructurada.

Los Sistemas de Extracción de Información (SEI) operan en un contexto formado por un conjunto de textos en lenguaje natural para extraer determinados conceptos que son de interés para una aplicación específica. Estos textos, en unión de la información definida para ser extraída, conforman el dominio de trabajo de un SEI.

La información a extraer por los sistemas de extracción de información se define a través de unas plantillas formadas por una serie de atributos que las caracterizan. La construcción de estas plantillas se realiza previamente y dependerán del contexto sobre el que vayan a trabajar y de la información que se desea obtener. A este contexto se le denomina escenario.

Desde la perspectiva del procesamiento del lenguaje natural, los sistemas de extracción de información son sistemas completos que deben trabajar en distintos niveles, desde el reconocimiento de palabras hasta el análisis de sentencias, y desde el entendimiento a nivel de sentencia sobre el análisis de discurso al del texto completo.

6.2. Antecedentes

En 1977, un sistema denominado THOMAS consistía en ilustrar cómo las palabras o las frases clave podían utilizarse para guiar a los usuarios en el descubrimiento de documentos de referencia útiles. Las frases clave son un tipo especialmente útil de información abreviada. Éstas condensan documentos en unas pocas palabras y enunciados, ofreciendo una descripción breve y precisa de los contenidos de un documento. Tienen muchas aplicaciones, entre las que destacan: clasificación o agrupación de documentos, interfaces de búsqueda, motores de búsqueda y construcción de tesauros[22]. Las frases clave se eligen con frecuencia manualmente, casi siempre por los autores de un documento, pero a veces por indizadores profesionales. La asignación manual de frases clave es tediosa y lleva tiempo, requiere experiencia y puede dar resultados no coherentes, de modo que los métodos automáticos benefician tanto a los que reúnen como a los usuarios de grandes colecciones de documentos. En consecuencia, se han propuesto varias técnicas automáticas.

Los sistemas de extracción de información tienen sus orígenes desde los años sesenta, pero durante más de dos décadas el desarrollo en este campo fue en extremo pobre, y es hasta finales de la década de los ochenta cuando se comienza a tener un levantamiento dentro de este campo de investigación. Esta tendencia se puede explicar de manera coherente al pensar que la computación nació durante los años cuarenta, citando el desarrollo de la primera computadora electrónica ENIAC (*Electronic Numerical Integrator and Computer*) en 1945, pero sería hasta mucho tiempo después que las computadoras dejarían de tener usos extremadamente específicos dentro de las universidades y se comenzaran a integrar a otros ámbitos del quehacer humano.

Tomando en cuenta el desarrollo computacional, es hasta casi principios de los noventa cuando se tuvo la suficiente capacidad de procesamiento para poder abordar temas como la "extracción de

22 Los tesauros son herramientas terminológicas utilizadas en el análisis y recuperación de información en bases de datos documentales con dos objetivos: controlar el vocabulario y conocer todos los términos relacionados con un concepto determinado.

Un tesauro trata de plasmar el sistema conceptual utilizado en una materia o disciplina determinada, y puede ser definido como un conjunto de términos estructurados jerárquicamente según la generalidad o especificidad de sus significados y relacionados por vínculos asociativos y de equivalencia. En un tesauro se puede conocer de un término aquellos otros que mantienen algún tipo de relación semántica con él: sus equivalentes, los que tienen una significación más amplia (TG - términos genéricos), los que representan conceptos más específicos (TE – términos específicos) y aquellos otros que por diversos motivos se le asocian (TR - términos relacionados). Existen tesauros que incluyen además equivalencias idiomáticas.

información", además de que en su génesis la computación no contaba con grandes acervos de información que crearan la necesidad de desarrollar buscadores, al hacerse la computación de uso más cotidiano por sus inminentes ventajas, la acumulación de información fue mayor, lo que pronto generó necesidades de crear motores de búsqueda efectivos que pudiesen ser de ayuda para la identificación de textos útiles para necesidades particulares, pues los procesos manuales son lentos e inexactos.

El avance de las tecnologías de información llevó a la necesidad de extraer información escrita en lenguaje natural a principios de los años ochenta. Un factor importante que motivó la investigación en este campo es el patrocinio y la organización por parte de la Agencia de Defensa de los Estados Unidos (DARPA, *Defense Advanced Research Projects Agency*) de siete conferencias de comprensión de mensajes, comúnmente conocidas como MUC (*Message Understanding Conferences*) entre 1987 y 1998 por la enorme cantidad de textos en formato electrónico disponibles. Estas conferencias fueron:

- MUC-1 (1987) textos sobre operaciones navales.
- MUC-2 (1989) textos sobre operaciones navales.
- MUC-3 (1991) noticias sobre actividades terroristas.
- MUC-4 (1992) noticias sobre actividades terroristas.
- MUC-5 (1993) noticias sobre microelectrónica y fusión de corporaciones.
- MUC-6 (1995) artículos sobre sucesión de puestos en compañías importantes.
- MUC-7 (1998) artículos sobre vehículos espaciales y lanzamiento de mísiles.

En estas conferencias se definieron las reglas de extracción de información, se estableció un tema específico para cada conferencia al que llamaron dominio y se proporcionó un corpus etiquetado sobre el que compiten sistemas, tratando de obtener los mejores resultados de extracción de información.

Esta iniciativa fue en extremo fructuosa por varias razones:

- El énfasis en tener un sistema práctico donde los investigadores por fin justificaran sus teorías.
- La disposición de un conjunto uniforme de material para entrenamiento y prueba animó la evaluación rigurosa usando un sistema acordado de métricas.

- La introducción de un elemento competitivo relacionado directamente con la retroalimentación hicieron más interesantes las investigaciones.

En estas conferencias participaron tanto elementos de clase industrial (*General Electric, Bolt Beranek & Newman*) como universidades (las universidades de *Edimburgo, Kyoto* y *Massachussets*).

Como resultado de estas conferencias se desarrollaron varios sistemas, por citar algunos ejemplos: TIPSTER (programa de investigación sobre recuperación y extracción de información del gobierno de los Estados Unidos) que surgió a raíz de la Muc-7, y CRYSTAL. Este último desarrollado en la Universidad de *Massachussets*, el cual fue implementado para inducir automáticamente un conjunto de reglas de análisis de corpus, para un dominio específico, a partir de ejemplos de entrenamiento. Este sistema crea reglas casi tan buenas como si se hubieran realizado de forma manual.

La dinámica de cada conferencia era presentar un dominio acerca del cual se presentaban diferentes sistemas que competían entre sí para observar cuáles eran los que lograban mejores resultados.

Todos los acontecimientos anteriores impulsaron la investigación por parte de diversas instancias científicas en este campo, lo que poco a poco fue formando los primeros fundamentos de lo que hoy se conoce como extracción de información.

A continuación se mencionarán algunos casos de las Mucs a detalle (web.03):

Las dos primeras conferencias se realizaron en 1987 y 1989 respectivamente, y se enfocaron a analizar operaciones navales.

Las Muc-3 (1991) y Muc-4 (1992) trataron artículos y noticias sobre terrorismo en Latinoamérica.

Para llevar a cabo este análisis se definieron los siguientes puntos:

- Se estableció una plantilla, las reglas de extracción y se aplicaron a 1000 textos.
- A efectos de evaluación, se encargó un programa de calificación semiautomática a partir de un conjunto independiente de plantillas.
- En cada evaluación anual se procesaba un nuevo conjunto de textos y se calificaba según las nuevas plantillas.
- Trabajaron conjuntamente alrededor de 17 centros de investigación.

La Muc-5 (1993) introdujo tareas más orientadas a los negocios, tales como anuncios de empresas. Se relaciona con una conferencia sobre noticias de microelectrónica y fusión de corporaciones, dicha aplicación fue desarrollada para los textos en inglés y japonés.

En 1995, la Muc-6 introdujo la extracción de entidades de nombres como un componente dentro de las tareas, por ejemplo, nombres de personas, compañías o lugares en texto libre.

En 1998, la Muc-7 demostró que la extracción de Entidades Nombradas[23] (*Entity Named Recognition*) en artículos en inglés era un problema resuelto a medias. Los mejores programas de dicha Muc alcanzaron alrededor de F=93%, comparado con el desempeño humano estimado en cerca de F=97%. La medida F es una combinación de *precision* y *recall* que se explica así:

$$F1 = \frac{2PR}{P + R}$$

Donde "R" es igual a la recuperación y "P" a la precisión.

Un punto sobresaliente es el hecho de que se reunieron tres veces para analizar las dificultades que se habían tenido, así como los progresos, e intercambiaron propuestas acerca de su futura evaluación para los sistemas en las conferencias habituales. Estás aplicaciones compartidas se comportan de la siguiente manera:

- Aceleran el intercambio de ideas.
- Es importante tener un progreso por parte de los distintos grupos por la presión de los competidores.
- Es necesario mostrarse receptivos a las ideas de los demás.
- Además se manifiestan las tendencias de la investigación (web.05).

Para evaluar un Sistema de extracción de información es necesario seguir las siguientes reglas:

---

[23] *Entity Named Recognition*: reconocimiento de entidades con nombre, consiste en identificar entidades de diferentes tipos en el texto, que corresponden con nombres de personas, lugares, organizaciones, fechas. Se realiza en dos fases, la primera lleva a cabo la detección y la segunda realiza la clasificación de las entidades detectadas en base a las clases previamente definidas.

- Cobertura o Recuperación (*recall*): es la cantidad de extracciones correctas respecto a la cantidad total de extracciones que deberían haberse obtenido.
- Precisión (precision): es la cantidad de extracciones correctas respecto a la cantidad de extracciones que el sistema ha realizado.

En los principios de los años noventa, las evaluaciones de las Conferencias sobre Comprensión de Mensajes ("*Message Understanding Conferences*" [MUC]) empezaron a financiar el desarrollo de algoritmos métricos y estadísticos para ayudar a las evaluaciones gubernamentales de las tecnologías emergentes de extracción de información. A mediados de dicha década, las evaluaciones Muc comenzaron a suministrar datos y definiciones de tareas, además de proporcionar un programa totalmente automatizado de puntuación para medir el rendimiento de las máquinas y del ser humano. Las tareas aumentaron desde la simple producción de una base de datos de acontecimientos encontrados en artículos periodísticos, a la producción de un conjunto de bases de datos de información cada vez más compleja, extraída de múltiples fuentes de noticias en diversos idiomas. Los resultados de estas evaluaciones se presentaron en congresos durante los años noventa, en los cuales los hombres que las desarrollaban y los evaluadores compartían sus resultados y los especialistas del gobierno de E.U.A. describían sus necesidades.

Para efectos ilustrativos, esta tesis se enfocará en la Muc-3, que trata de tareas de extracción de eventos, en el cual un programa debía extraer información de ataques terroristas en artículos de noticias en texto plano[24].

6.2.1. Los sistemas de la MUC-3

Los detalles de la tarea de dicha muc así como la recopilación del corpus serán omitidos, este apartado sólo se enfocará a la descripción de los sistemas y las métricas y mecanismos de desempeño de los mismos.

Un texto típico de este corpus (traducido al español) es mostrado a continuación:

24 Dicho sistema se basó en artículos en inglés.

> "En la última noche el objetivo de los terroristas fue la Planta de Licor de Antioquia. Estaban previstos para estallar 4 cohetes muy poderosos muy cerca de los tanques donde 300,000 galones del llamado crudo de Castilla, utilizado para operar los hornos, eran almacenados".

La tarea a la que se enfrentaba un programa era extraer y almacenar características específicas del incidente. Estas características incluyen marcas específicas tales como la fecha, el lugar, el objetivo o blanco del ataque, las armas utilizadas y tal vez el móvil del ataque (asesinato, sabotaje, etc.). Los programas tenían que "crear" plantillas (tablas o *templates*) para ser llenadas mostrando parcial o totalmente la descripción del mismo evento, es decir, tenían que generar una plantilla por cada evento, no múltiples plantillas representando diferentes descripciones del mismo evento encontradas en el texto.

Un ejemplo de la forma en que se debía llenar la plantilla es mostrada en la figura 6.1. Es un ejemplo muy resumido y los campos vacíos indican que no hay datos especificados sobre ese campo.

Campo	Descripción
Id de la noticia	TST-MUC3-0001
Fecha del incidente	04 Feb 90
Tipo del incidente	Incendio
Autor	"Guerrillas"
Objetivo físico	Carro tanque
Objetivos humanos	_____
Instrumento	
Lugar del incidente	"Guatemala:Peten:Flores"

Tabla 6.1.- Ejemplo de una plantilla de salida para los sistemas de la Muc-3.

Los mejores sistemas reportados en la Muc-3 reportaron resultados en el campo de 50% en recuperación y 60% de precisión para la extracción de dichos eventos. En síntesis, los sistemas podían encontrar cerca de la mitad de los datos que estaban buscando, con un rango de efectividad de menos del 50%. Durante la Muc-6, los mejores sistemas marcaban en su más alto índice 75% de recuperación y 75% de precisión, donde el desempeño parecía haber alcanzado la cima.

Un resultado de 75% en una métrica o en otra es bastante satisfactorio en la actualidad, ya que es muy difícil superar esa cifra a pesar de los avances en las investigaciones actuales en inteligencia artificial e ingeniería lingüística.

Ahora se describirán algunas técnicas principales del PLN utilizadas en las conferencias MUC y algunas otras más. Estas incluyen comparación de patrones, autómata de estados finitos[25], análisis de textos libres y modelos estadísticos.

6.2.2. El sistema Diderot

Este sistema de extracción de información se desarrolló en el laboratorio de Investigación de Cómputo de la Universidad del Estado de Nuevo México, en colaboración con la Universidad de *Brandeis*[26] y ha sido evaluado por investigadores en un periodo de 12 a 18 meses ya que el funcionamiento en inglés y en japonés es totalmente distinto. Este grupo de investigadores aplicó las tareas del procesamiento del lenguaje natural utilizado en todo el mundo. Además se practicó el análisis a más de 5000 textos. El resultado de este análisis ha sido utilizado para desarrollar y probar el sistema producido por cada sitio donde se desarrolló este proyecto. Estos sistemas en inglés y japonés han utilizado CRL[27] (Lista de Certificados Revocados).

El grupo CLR ha desarrollado métodos estadísticos para destacar las partes relevantes del texto, mediante programas que marcan y reconocen nombres de personas, organizaciones y fechas (Cowie y Guthrie).

El análisis actual ha llevado a una parte crítica, porque es controlado por palabras clave del texto. Cabe destacar que las estructuras léxicas

---

25 Un autómata de estados finitos es una máquina de estados que tiene acceso a una secuencia de símbolos de entrada (mediante una cabeza lectora).
Un autómata de estados finitos se encuentra en cada momento en un estado determinado y puede transitar a otro estado, Para ello se realizan los siguientes pasos:

- Se lee la cinta y se avanza la cabeza lectora.
- En función del símbolo leído y el estado actual tiene un comportamiento propio.

26 Se fundó en 1948 en Massachussets, es la Universidad privada más joven y la única no-confesional patrocinada por judíos en los EUA.

27 La Lista de Certificados Revocados contiene el número de serie de todos los certificados emitidos por la autoridad de certificación de la *Generalitat* Valenciana que, por algún motivo, han dejado de ser válidos de manera previa a la expiración de su periodo de validez original.

fueron tomadas de diccionarios legibles por máquinas, y la información extraída se obtuvo de córporas (varios corpus de un tema específico).

El procesamiento de textos de micro-electrónica funcionó de 12 a 18 meses, obteniendo buenos resultados para los japoneses y no tan buenos resultados para textos en inglés.

Los textos utilizados en japonés e inglés son procesados por tres pre-procesos independientes:

- Una cadena de estados finitos caracteriza las etiquetas, estas marcan: nombres, nombres de organizaciones, nombres de lugares, fechas y otros nombres dependiendo del dominio.
- Una parte del discurso etiquetado.
- Una base estadística determina el texto relevante como por ejemplo: micro es la base de la palabra microelectrónica y basándose en esta raíz que sería micro y el contexto, es posible deducir que un texto dado trata de componentes microelectrónicos por el número de veces que aparece en el texto.

Si la determinación estadística es rechazada del texto de procedencia al final del estado de salida, se produce una plantilla vacía. En otro caso, los resultados de los otros dos estados son convertidos en hechos y estos son pasados a otros refinamientos del texto como:

- Fusión: semántica de etiquetas, las cuales pueden marcar unidades de frases que son acopladas con una etiqueta pos (*Part Of Speech*), que marca palabras individuales.
- Reconocer sustantivos compuestos: este grupo de frases y palabras dentro de enunciados compuestos usan POS e información semántica.
- Los factores que intervienen son: importancia, estadísticas, discurso, transformación, referencia y reconocedor de imágenes.
- *Parser*: la información relevante del párrafo es usada para seleccionar cuáles sentencias se procesarán en otros procesos de sentencias. Las sentencias contienen marcas en los sustantivos de grupos de frases que son pasados para producir una representación parcialmente completa del contenido relevante de la semántica de la oración.
- Discernidor de referencia (*Reference resolver*): los marcos de las plantillas fueron unidas basándose en igualar nombres y sustantivos compuestos empezando con artículos definidos.
- Formatear la plantilla: esta transformación de la plantilla resuelve los marcos dentro del final de la forma de salida.

Este sistema utiliza las siguientes herramientas:

- Técnicas estadísticas de filtración.
- Etiquetado semántico.
- Parte del etiquetado del discurso.
- Combinación.
- Agrupación de la frase nominal.
- Análisis.
- Resolución de referencia (web.12).

La conferencia Muc-6 fue organizada por Beth Sundheim del grupo *Naval Research and Development*. En esta conferencia se evaluaron las siguientes tareas:

a) Reconocimiento de nombres de entidades.
b) Correferencia.
c) Elementos de las plantillas.
d) Plantillas de escenarios (web.07).

LaSIE es un sistema integrado y sencillo de extracción de información desarrollado en la Universidad de *Sheffield*[28], que ha sido utilizado para la producción de salidas especificadas en la Muc-6. No obstante, *LaSIE* ha sido desarrollado como un sistema de información de propósito general y no sólo es utilizado para dicha Muc. Las etapas en las que consiste: preproceso léxico, interpretación semántica y análisis (*parsing*), e interpretación del discurso, no corresponden directamente a alguna tarea de la Muc-6.

El texto sobre el que se realiza la extracción de información consiste en artículos del *Wall Street Journal* y la información a obtener son los cambios de puestos de trabajo producidos en empresas.

La fase uno consistió en la definición e identificación de los *tokens* (unidades o fragmentos), realizar un análisis morfológico para obtener las formas primitivas o raíces, así como la búsqueda de entidades y la relación que existe de las estructuras. Cuando se etiquetan, se identifican listas de nombres propios como por ejemplo: formatos de fecha, nombres de moneda y una lista de nombres comunes que son indicadores de entidades (Agencias, Organizaciones, etc.).

La fase dos utiliza la gramática general para dar mayor velocidad al analizador (*parser*).

28 Universidad de Inglaterra fundada en 1828, entre sus clientes se encuentran *Boeing, Rolls Royce, Unilever, Boots, AstraZeneca, GSK, ICI, Slazenger,* además de agencias del gobierno británico y organizaciones de beneficencia.

Después se da una interpretación al discurso semánticamente, gracias a la representación de instancias, clases ontológicas y sus propiedades. Las propiedades pueden ser asociaciones, reglas o valores. Para evitar problemas de correferencia[29] se comparan las sentencias con las anteriores para saber si pueden ser unidas en una sola. Las comparaciones se pueden hacer en las diferentes etapas.

Estas comparaciones son:

- Comparación de instancias con propiedades de nombre con las existentes que también tengan propiedades de nombre.
- Todas las nuevas son comparadas con todas las anteriores.
- Las instancias con pronombres son comparadas con sentencias del mismo párrafo (exceptuando si las frases con pronombre son las iniciales del párrafo, en cuyo caso se comparan con las del párrafo anterior).
- Todas las restantes son comparadas con las instancias ya existentes de los párrafos actuales y previos.
- Las correferencias de entidades se resuelven utilizando varias técnicas heurísticas[30]. Otro aspecto a destacar es que la cabecera no se procesa, ya que muchas palabras con mayúsculas pueden ser tratadas erróneamente como nombres propios y además los autores suponen que los nombres propios especificados en la cabecera lo serán más tarde en el cuerpo del texto.
- Las plantillas se recuperan del modelo del discurso formateando aquellas que tienen propiedades de nombre. Los valores de las propiedades se buscan examinando otras propiedades relacionadas. Las plantillas del escenario se realizan buscando aquellas instancias que tienen los valores para las propiedades requeridas.
- Desambiguación de la frase.

La tercera fase trata sobre la desambiguación del ámbito o límite de la frase. A continuación se muestra como se utilizó en el Sistema de Extracción de Información de Textos Notariales (EXIT). Está englobado en una fase previa a la fase de análisis léxico y el método de

29 Es la sustitución de una palabra por otra u otras que expresan o aluden al mismo concepto que aquella y a la inversa, distintos significados que se asocian a un mismo significante en momentos y espacios no excesivamente distanciados (en ocasiones en un mismo periodo).

30 De acuerdo con ANSI/IEEE Std 100-1984, la heurística trata de métodos o algoritmos exploratorios durante la resolución de problemas en los cuales las soluciones se descubren por la evaluación del progreso logrado en la búsqueda de un resultado final. Se suele usar actualmente como adjetivo, caracterizando técnicas por las cuales se mejora en promedio el resultado de una tarea resolutiva de problemas (parecido al uso del método óptimo).

reconocimiento de entidades es una fase propia tal y como aparece en la arquitectura de Grishman.

Palmer[31] considera que la desambiguación (acción de decidir el significado que tiene una palabra en un contexto) no es tarea fácil, porque requiere de conocimiento de frases completas para aplicar algunas técnicas del procesamiento del lenguaje natural.

Para determinar los límites de una sentencia se utilizan las gramáticas regulares.

La gramática intenta encontrar patrones de caracteres del tipo; "punto, espacio, letra en mayúscula", lo cual ocurre al final de una sentencia que termina en punto y seguido. En los sistemas más completos, como los modernos, se estudia la palabra completa que precede y sigue a la marca de puntuación e incluye una extensiva lista de palabras y una lista de excepciones para intentar reconocer abreviaturas y nombres propios.

El sistema de extracción de información *Alembic* (es un sistema de extracción de información utilizado a mediados de los años 80, el cual fue evaluado en la Muc-6, usando un árbol binario de decisión para su funcionamiento) contiene un extenso módulo de desambiguación de los límites de frases basado en expresiones regulares, creadas utilizando el generador de analizadores léxicos *Flex*.

A *Flex* se le conoce como una herramienta para generar programas de reconocimiento de concordancia de los patrones en un texto como son los escáneres. *Flex* lee la entrada estándar o el archivo que se le indique generando una descripción del escáner a generar.

La descripción se encuentra en forma de parejas de expresiones regulares denominadas "reglas *flex*" y genera como salida un archivo fuente que define una rutina. Este archivo se compila y se enlaza con una librería para producir un ejecutable. Cuando se arranca el archivo ejecutable, éste analiza su entrada en busca de casos de las expresiones regulares. Siempre que encuentra uno y se ejecuta el código correspondiente.

A continuación se muestra un ejemplo:

31 Martha Palmer, Universidad de Pensilvania, muy interesada en la representación de la información semántica y sus aplicaciones en el PLN.

Para hacer contacto con el uso de *flex*, la siguiente entrada especifica un escáner que siempre que encuentre la cadena "*username*" la reemplazará por el nombre de entrada al sistema del usuario:

```
%%
username    printf( "%s", getlogin() );
```

Por defecto, cualquier texto que no reconozca el analizador léxico de *flex* se copia a la salida, así que el efecto neto de este escáner es copiar su archivo de entrada a la salida con cada aparición de "*username*" expandida. En esta entrada, hay solamente una regla: "*username*" es el patrón y el "*printf*" es la acción. El "%%" marca el comienzo de las reglas.

El módulo de desambiguación de los límites de las frases utiliza una lista de 75 abreviaturas y una serie de aproximadamente 100 reglas manuales para identificar límites de sentencias, tanto como títulos, fechas y expresiones de tiempo, y abreviaturas. Alcanza una tasa de error del 0.9%.

Mark Wasson, junto con sus colegas, desarrollaron un sistema que reconoce tanto *tokens* especiales (p ej. abreviaturas, términos que no se encuentran en el diccionario como nombres propios, términos legales, financieros, modismos, etc.) como límites de frases.

Otro método para la desambiguación del límite de las frases está basado en las aproximaciones heurísticas, las cuales dependen de la disponibilidad del corpus con un buen comportamiento, una puntuación regular y pocos caracteres extraños.

Como Palmer comenta, Humphrey utilizó una red *neural feed-forward* para desambiguar el límite de las frases, aunque también utilizó una gramática regular para tokenizar el texto antes de entrenar dicha red, alcanzando resultados de alrededor del 93% de acierto.

La lengua española dispone de tres símbolos como posibles finalizadores de sentencias o frases. Estos tres símbolos son el punto, la interrogación y la exclamación o admiración. El problema que se presenta, al igual que en otras lenguas, es que no siempre funcionan como finalizadores de frases, sino que pueden desempeñar otro papel distinto. Por ejemplo, el punto será en la mayoría de los casos, final de frase, pero también puede aparecer como separador entre números (miles, millones, etc.) o como parte de la abreviatura de una palabra, etc.

6.3. Expresiones regulares

Las expresiones regulares (*regex*) proporcionan los medios para especificar o definir lenguajes regulares. A simple vista las personas que se dedican a la industria del *software* relacionan esta expresión con utilidades de patrones de comparación como *grep* en *Unix*, lenguajes de programación como Perl (*Practical Extraction and Report Language*) y herramientas de análisis léxico para compiladores de lenguajes de programación, tales como *lex*. No obstante las expresiones regulares tienen un formalismo de propósito general para escribir y comparar patrones; este formalismo no es específico para una herramienta o lenguaje de programación.

En términos simples, una *regex* representa un conjunto regular de cadenas en tres operaciones simples: adyacencia, repetición y alternancia. Por lo tanto, una expresión regular es un conjunto de patrones que sirven para representar a una cadena de caracteres, por lo que provee una caracterización finita de un conjunto infinito de dichos patrones, por ejemplo.

a(b|c)*a

representa un lenguaje infinito (conjunto de cadenas).

L={aa, aba, aca, abba, abca, acba, acca, …}

donde (b|c) significa "escoge b o c", el asterisco representa "cero o más veces" y la adyacencia de esos dos símbolos tienen un significado conjunto.

Un ejemplo de expresión no regular sería el siguiente conjunto infinito:

{ab, aabb, aaabbb, aaaabbbb, …}

por lo tanto no es una expresión regular ya que carece de un conjunto finito de elementos.

Así, una expresión regular especificando una clase de nombres propios podría mostrarse así:

{Sr.| Sra.| Srita.| Dr.} {A| B| C| …| Z} Apellido

Donde "apellido" se define mediante selección dentro de una lista de apellidos, tales como los apellidos de un directorio telefónico o una

nómina que un sistema pueda identificar. Todos los elementos que componen una expresión regular se denominan literales, como son "Sr.", "A" y ".".

Una técnica para analizar textos a través de expresiones regulares consiste en separar diferentes niveles lingüísticos de procesamiento en módulos que van seriados uno del otro como se muestra en la figura 6.1.

Figura 6.1.- Metodología típica de un SEI.

Las primeras etapas de procesamiento reconocen las entidades lingüísticas como las palabras y los límites de los enunciados y trabajan de manera regular en un dominio independiente.

Por ejemplo, para dividir una sentencia en palabras y puntuaciones, un tokenizador puede utilizar únicamente patrones lingüísticos para identificar los límites de las palabras, requiriendo ligeras o ninguna modificación cuando el sistema es cambiado a un nuevo dominio (como el idioma, ya que cada uno tiene sus propias reglas de tokenización). El etiquetado pos es un dominio dependiente sensitivo a los corpus, particularmente respecto a los nombres propios. Las etapas posteriores reconocen más patrones de dominios específicos, necesitando la identificación de objetos y eventos que serán distintos entre las aplicaciones. Así, los patrones que serán identificados diferirán a través de los dominios tanto como sean llenadas las plantillas de salida que se necesiten. De forma similar, el conocimiento requerido para crear exitosamente la plantilla final (a través de una unión de sub-plantillas) será a partir de un dominio dependiente.

Hay muchas *regex* que han sido escritas en el lenguaje de programación Perl, el cual es una síntesis bastante confiable de lo que los programas en el ámbito del lenguaje se han diseñado para alcanzar. Su sintaxis económica y las poderosas funciones de manejo del texto que tiene lo hacen una herramienta útil para el análisis inicial de texto.

Por ejemplo, es relativamente sencillo escribir un lematizador en *Perl* o un programa que conjeture la clase de sintaxis de una palabra basado en características morfológicas tales como prefijos y sufijos.

Cualquier palabra en español no capitalizada que termine con "…oso" puede ser reconocida como adjetivo, por lo que una palabra como "poder" puede ser transportada a "poderoso" en una línea de código del programa. Por ejemplo, el operador en *Perl* "=~" puede utilizarse para comparar una variable de cadena con un patrón de la forma /…/ y regresar el valor verdadero o falso según el resultado de la comparación. Así:

$cadena =~ /oso$/

compararía el valor de la cadena con la terminación "oso" (en este caso /…$/ indica fin de la cadena)

6.4. Método de desambiguación del límite de las frases

Un método para la desambiguación del límite de las frases basado en los estudios realizados por Riley y Palmer tiene el objetivo de distinguir cuando los símbolos finalizadores de frases desempeñan esa función en esa sentencia y no otra en la otra.

Debido a que el sistema EXIT trabaja en un dominio restringido, como es el de las escrituras notariales, el problema de la desambiguación de los límites de las frases queda reducido al estudio del papel que juega el punto en las frases. Es obvia esta reducción del problema ya que las escrituras notariales de compraventa no son más que la declaración de un notario dando fe de la transmisión de un inmueble. Por lo tanto no se encontrarán frases interrogativas ni exclamativas.

Es posible encontrar párrafos como el siguiente, el cual se debe segmentar en las distintas sentencias.

"La mercantil Dragados S.A. vende al Dr. Fernando García López la nave situada en la calle Almería. El precio de esta venta es de DIEZ MILLONES ($10, 000,000.00) de pesos. Dicho precio lo confiesa tener percibido la parte compradora antes de este acto y a su entera satisfacción. Los gastos e impuestos a que da lugar el presente otorgamiento, incluso el arbitrio municipal de Plus Valía, serán a cuenta y cargo exclusivo de la empresa Dragados S.A. Hago las reservas y advertencias legales a los comparecientes".

Según los estudios realizados sobre el dominio de trabajo, el símbolo del punto puede desempeñar las siguientes funciones:

a) Abreviatura (Sr., D., S.A., CIA, etc.).
b) Acrónimo (IBM., N.I.F., etc.).
c) Separador de miles 0 de decimales ($3,250.00, 7.500, etc.).
d) Finalizador de sentencias.

Este método de Palmer, se basa en el estudio de las palabras que aparecen alrededor del punto. Se va tomando *token* a *token* del texto; si alguno de esos *tokens* es un símbolo del tipo comillas o paréntesis se busca el símbolo de finalización (otras comillas o símbolo de cerrar paréntesis), no teniendo en cuenta los posibles puntos que aparezcan en medio. Se define un *token* como un elemento de la frase que puede estar formado por uno o más caracteres alfabéticos. Si se encuentra un *token*, seguido de un punto y este último seguido inmediatamente por otro *token* sin que aparezca un espacio en blanco entre el punto y el último *token*, se considera que todo forma un único *token*. Por ejemplo si se encuentra "N.I.F.", el tokenizador leerá el *token* "N", después el punto y al final el token "I". Como no existe un espacio entre los mismos se tomará todo como si el *token* estuviera formado por "NI", como posteriormente lee un punto e inmediatamente después una "F", vuelve a realizar la misma operación uniendo todos los *tokens* dando como resultado un *token* con valor "N.I.F", sin punto final. Por lo tanto el acrónimo es tomado como un único *token*. Utilizando la misma técnica cuando se trata de un punto que separa los miles y millones, como no existe espacio en blanco entre los miles y las centenas se toma todo como un único *token*.

Este método define una serie de reglas de decisión para definir si es final de sentencia o no y utiliza un diccionario de propósito general y una lista de abreviaturas. Como se ha dicho anteriormente estas reglas utilizan las palabras situadas a la izquierda y derecha del punto, dando lugar a una ventana de palabras. Concretamente la ventana de palabra está formada por las dos palabras anteriores al símbolo del punto y la posterior a él ("p-2 p-1. p+1").

Para la resolución de este problema se propusieron las siguientes reglas de decisión con base en el estudio realizado anteriormente, y para el dominio sobre el que trabaja el sistema.

>¿La palabra p+1 empieza con mayúscula?
NO: el punto no juega el papel de finalizador de frases.
SI: >¿La palabra p-1 es un tratamiento abreviado?
  SI: es final de frase

NO: >¿La palabra p-1 es un símbolo del tipo ` " ' , `) ', etc.?
NO: >¿La palabra p-1 esta en el diccionario de propósito general?
SI: es final de frase
NO: >¿La palabra p-1 está en la lista de abreviaturas?
NO: es final de frase
SI: >¿Es p-1 un tratamiento de persona abreviado?
SI: no es final de frase, ya que no puede acabar con una abreviatura del tratamiento. Es decir, no existen frases del tipo "Se realizó la venta a ese sr.". Si no que se tendría "Se realizó la venta a ese señor".
NO: >¿La palabra p-1 es un tratamiento de empresa abreviado?
NO: imposible ya que anteriormente se ha dicho que es un tratamiento.
SI: >¿La palabra p-2 es un nombre propio?
SI: es final de frase
NO: no es final de frase.

Un algoritmo similar se utilizó en el sistema "SEINS" para distinguir los puntos decimales de los puntos final y a parte.

6.5. Extracción de entidades, eventos y relaciones en textos

6.5.1. Gramáticas de contexto libre

La mayoría de los textos comúnmente encontrados contienen factores complejos que requieren métodos fuertes para análisis de los mismos, pero esto no indica que los problemas como la ambigüedad se van a solucionar. No obstante generan herramientas útiles para el análisis de palabras compuestas o sentencias de ambigüedad.

Como se mencionó anteriormente, los programas de extracción de información para las MUC comúnmente utilizaban analizadores simples, como un autómata de estados finitos para un análisis rudimentario del texto. El algoritmo CYK[32] (también llamado CKY por las siglas de sus

[32] El algoritmo de Cocke-Younger-Kasami (CYK) para análisis sintáctico de gramáticas independientes del contexto en forma normal de Chomsky fue descubierto por J. Cocke, pero fue publicado independientemente por Younger y Kasami , de ahí su nombre.

Es un algoritmo ascendente puro basado en programación dinámica. Hace uso de una matriz bidimensional indexada por posiciones de la cadena de entrada para almacenar los resultados parciales obtenidos, así el elemento A se encuentra en la Posición [i; j] de dicha

creadores Cocke, Young y Kasami) es un método de análisis un poco más robusto que tiene la capacidad computacional de un Autómata de Movimientos (*PDA*, *push-down automaton*). Un PDA es realmente un autómata de estados finitos (FSM*, Finite State Machine*) con un espacio de memoria adicional que puede ser usada como una pila (*stack*).

La pila funciona como una memoria externa que se puede utilizar como infinita[33]. La máquina puede tanto leer como escribir de la pila. La operación básica "*push"* añade un símbolo a la cima de la pila mientras que la operación "*pop*" remueve el símbolo de la cima.

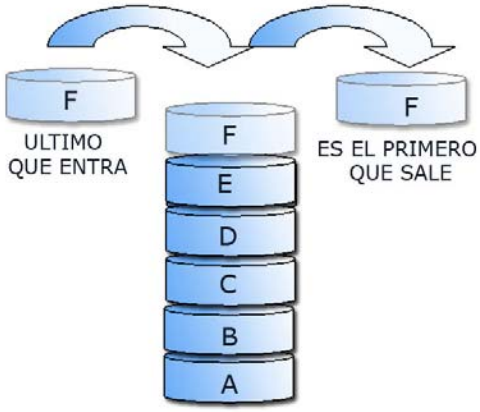


Figura 6.2.- Ejemplo de una pila.

Este arreglo permite que un algoritmo pueda utilizar Gramáticas de Contexto Libre (CFG´s) para reconocer o generar estructuras recursivas incluyendo alternadamente palabras compuestas y enunciados subordinados. La pila puede almacenar momentáneamente el contexto de la salida mientras que el analizador estudia a fondo la expresión y después reinstala de nuevo la salida cuando el análisis interno está hecho.

Así se tiene que el análisis de un enunciado como:
"*Sun Microsystems* ha liberado la última actualización de la máquina virtual de java en su versión 5 en la ciudad de Nueva York y está en espera de que los distribuidores de *software* adquieran la aplicación..."

---

matriz si y sólo si A = ai+1 : : : aj. En este contexto, se considera que el algoritmo CYK construye un conjunto de ítems.

$$\left\{ [A, i, j] \mid A \stackrel{*}{\Rightarrow} a_{i+1} \ldots a_n \right\}$$

33 Por supuesto, ninguna pila tiene almacenamiento infinito. Se dice de esta manera porque cuando el programa necesita un poco de memoria extra, un módulo del mismo provee un poco más a dicho programa.

Un PDA puede almacenar el análisis de "*Sun Microsystems*" en la pila mientras que continúa analizando la frase compuesta "ha liberado la última actualización de la máquina virtual de *java* en su versión 5 en la ciudad de Nueva York" y después recuperar su lugar para terminar de analizar la frase completa. Una FSM podría no ser confiable en su totalidad a menos que pudiera anticipar exactamente lo que el lenguaje que va a analizar le tiene preparado estructuralmente, es decir, cómo está ordenada la sintaxis del enunciado que va a analizar.

Enseguida se utiliza una convención similar a las mencionadas anteriormente. Las letras mayúsculas denotan categorías gramaticales, mientras que las minúsculas denotan palabras del lexicón en español. Las gramáticas de contexto libre difieren de las expresiones regulares en que contienen recursividad, tal como se muestra a continuación en algunos ejemplos para definir un Grupo de Sustantivos (GS).

GS = DET + SUST
GS = DET + MOD + SUST
GS = GS + PREP + GS,

La recursividad es una manera conveniente de especificar grupos complejos de sustantivos tales como:

"El lanzamiento del parche para *Dreamweber* de *Macromedia"*.

El ejemplo anterior indica que una empresa lanzó el *software* para el parche de su producto *Macromedia* debido a las necesidades de actualización de los diseñadores de sitios *web*.

Entonces un grupo de sustantivos puede ser definido de diferentes maneras como se mostró anteriormente.

Las reglas de dicha gramática también son llamadas "reglas de reescritura" debido a que el análisis sustituye una parte de la regla por otra. Así un patrón tal como

DET MOD SUST,

en una oración más larga

DET MOD SUST VERBO ADVERBIO,

puede ser reconocida como un GS y reescrita como sea necesario para generar otro GS.

GS VERBO ADVERBIO

Se aplicó otra regla de reescritura

GV = VERBO + ADVERBIO,

lo que podría resultar en el patrón

GS GV,

lo que a su vez podría resultar en la siguiente sentencia definida por la regla:

S = GS + GV

Estas gramáticas son denominadas "contexto libre" ya que no toman el contexto de la parte izquierda de la igualdad al momento de aplicar la regla. Cuando se usa

GS = DET + MOD + SUST,

no hay preocupación de que el GS (grupo de sustantivos) está seguido por un verbo o cualquier otro concepto, simplemente reconoce el patrón DET + MOD + SUST y aplica la regla.

6.5.2. Análisis con un autómata de movimientos

El algoritmo CYK se encarga de analizar las CFG y utiliza una tabla de subcadenas con una estructura definida (*wfsst, well-formed substring tables*[34]) para almacenar los resultados de las cadenas versátiles que se van formando en la sentencia. El uso de esta tabla evita la repetición del trabajo comúnmente encontrado en algoritmos menos sofisticados y por lo tanto mejora la eficiencia del análisis. De hecho CYK es un tipo de

---

34 Tablas de Cadenas Bien Formadas (*WFST, well formed string tables*). Se trata de tablas construidas dinámicamente que almacenan los resultados de los elementos no terminales reconocidos durante el proceso de análisis. Estos resultados son accesibles globalmente, no se destruyen durante el proceso de *backtracking* y pueden ser consultados y utilizados por el analizador. Para utilizar una *WFST* se debe modificar el analizador de forma que, antes de abordar el reconocimiento y la construcción de cualquier componente, se compruebe que tal componente no hubiere ya sido construido e incorporado a la tabla.

En general, el mecanismo del *backtracking* hace que al ejecutarlo se retorne al estado anterior y, por lo tanto, se pierdan todos los efectos producidos en la opción ahora rechazada.

algoritmo de programación dinámica que soluciona el problema total a través de ir resolviendo subproblemas, luego entonces a través de estas pequeñas soluciones va uniendo apropiadamente el enunciado mientras sigue buscando la solución global.

La figura 6.3 muestra la forma de trabajar del algoritmo (en la notación Pascal), tomado de un texto de una teoría autómata.

La "S" representa una cadena de palabras y la "V" (inicialmente vacía) una tabla de subcadena de tamaño "n". La tabla es accedida por subíndices en el rango de [1, n] y $V_{i,j}$ denota la celda en la columna *i* y la fila en la columna *j*.

```
begin
  for i := 1 to n do
    V[i,1] := {A|A = a /*es una regla y la i-ésima palabra de S es a*/};
  for j := 2 to n do
    for i := 1 to n - j +1 do
      begin
        V[i,j] := {};
        for k := 1 to j -1 do
          V[i,j] := V[i,j] + {A|A = B + C /*esta regla indica que B está en
V[i,k] y C en V[i+k,j-k]*/};
        end
end
```

Figura 6.3.- El algoritmo CYK.

Básicamente el primer ciclo (*loop*) completa las categorías léxicas asociadas a palabras en la secuencia. *A = a* es una regla que relaciona palabras con categorías léxicas, por ejemplo:

TVERB = lanzó

donde "TVERB" denota un verbo transitivo, es decir, un verbo que toma un objeto.

El segundo, un bloque de triple jerarquía en categorías no léxicas se combina con categorías de nivel inferior en la tabla. Este paso utiliza reglas como:

GS = DET + SUST

Considerando el siguiente ejemplo, la *wfsst* corresponde a todos los niveles excepto al primero como se muestra en la tabla 6.1. La fila 1 consiste en los ítems léxicos en la sentencia a ser analizada, en este caso se toma como ejemplo "La organización lanzó el *software*".

La siguiente fila de la tabla contiene las categorías gramaticales de los lexemas, la cual forma la primera fila de la *wssft*. Las filas subsecuentes corresponden a los más altos niveles de las estructuras sintácticas construidas de "abajo hacia arriba" (*bottom-up*) en las categorías léxicas. La última categoría de la cadena total reside en la parte izquierda de la esquina de la tabla.

	La	organización	lanzó	el	Software
	1	2	3	4	5
1	DET	SUST	TVERB	DET	SUST
2	GS			GS	
3			GV		
4					
5	S				

Tabla 6.2, Tabla de subcadenas para un análisis completo.

Así, la entrada GV en la fila 3 y columna 3 de la tabla (*wssft* [3,3]) indica que "lanzó el producto" se ha identificado como un Grupo Verbal, formado por la combinación del verbo "lanzó" con la frase de sustantivo "el *software*". Esta formación está permitida por una regla gramatical previamente definida como podría ser: GV = TVERB + GS, la cual señala que un grupo verbal consiste de un verbo seguido por un grupo de sustantivo.

La entrada S del *wssft* [5,1] revela que la cadena completa ha sido identificada como una sentencia. Ésta ha sido formada por la combinación de un grupo de sustantivo (La organización), con un grupo verbal (lanzó el *software*) mencionado anteriormente. La regla de gramática correspondiente quedaría:

S = GS + GV

CYK tolera tanto ambigüedad léxica como estructural.

- Ambigüedad léxica.- Se refiere a que una palabra puede pertenecer a más de una categoría léxica, por lo que las celdas de la primer fila de la tabla podrían contener más de una entrada.

- Ambigüedad estructural.- Es cuando un grupo de palabras puede ser analizado de más de una manera, dando resultado a un traslape de subestructuras.

Las hipótesis[35] estructurales que incorporan otra subestructura pueden usarla de cualquier manera definida por los datos de entrada y las reglas. En otras palabras, una supuesta frase que ha sido descubierta en la sentencia puede ser combinada con otro material de acuerdo a las reglas, aún si dicha hipótesis es falsa. No obstante, tales hipótesis frecuentemente son desechadas ya que no cabrán en una estructura que explique todas las palabras en la oración.

Un ejemplo que clarifique lo anterior sería:

"La empresa que lanzó la aplicación fue criticada"

Muchos analizadores sintácticos entenderían la hipótesis de que "la aplicación fue criticada" es una subsentencia de la oración completa, asumiendo una regla para formar un grupo de verbo pasivo como el siguiente : GV = BVERB3 + TVERB, donde BVERB3 se refiere a una tercera persona en singular del verbo "ser o estar", por ejemplo "es" o "eres".

	La	empresa	que	lanzó	la	aplicación	fue	criticada
	1	2	3	4	5	6	7	8
1	DET	SUST	CONJ	TVERB	DET	SUST	BVERB3	TVERB
2	GS				GS		GV	
3				GV				
4			RCLAUS		S			
5	GS							
6								
7	S							

Tabla 6.3.- Una tabla *wssft* con hipótesis competentes de las subestructuras.

Pero esta hipótesis puede estar condenada al fracaso debido a un pequeño error en la definición de las reglas, ya que si se quisiera analizar la sentencia completa daría como resultado:

(¿?: la empresa que lanzó) (S: la aplicación fue criticada))

35 Se llamará hipótesis estructural a las posibles combinaciones para formar una solución del análisis.

Siendo la correcta la siguiente:

(S:(la empresa que lanzó la aplicación)(GV: fue criticada))

"lanzó" es un verbo que debe tomar un objeto, por lo tanto no es posible formar una oración con la simple frase "La empresa que lanzó".

La utilidad del autómata de movimientos es que permite poder reconocer esta situación y evitar el error, mientras que un FSM más probablemente tendría que tomar las subsentencias y comparar sus valores. Un estudio cuidadoso de la oración por medio de un CFG detecta la presencia de oraciones relativas, de tal modo que se descubre la estructura un tanto oculta (como se mostró en la tabla anterior). Aunque "la aplicación fue criticada" de cualquier manera sería analizada como una sentencia, la hipótesis estructural que representa cruza un límite de la sentencia contenida.

La salida del análisis final ignoraría estas sentencias ya que hay una mejor hipótesis que analiza más datos. Por lo tanto, el análisis final es:

```
(S:
   (GS:
        (GS: (DET: la) (SUST: empresa))
        (CONJ: que)
        (GV: (TVERB: lanzó) (GS: (DET: la) (SUST: aplicación)))
        (GV: (BVERB3: fue) (TVERB: criticada))
)
```

La ambigüedad estructural podría parecer una ocurrencia poco repetida, pero en realidad no lo es. Aún los grupos de sustantivos pueden mostrar ambigüedad, como se observa en el siguiente ejemplo:

"La disponibilidad de la actualización por la División de Investigación de la compañía *Microsoft".*

El correcto análisis de esta frase sería:

(GS: (GS: La disponibilidad de la actualización) (CONJ: por) (GS:(GS: la División de Investigación) (CONJ: de) (GS: la compañía *Microsoft*)))

Lo que indica que la "División de Investigación" es un área de *Microsoft*, y no por ejemplo:

(GS: (GS: La disponibilidad de la actualización) (CONJ: por) (GS: la División de Investigación) (CONJ: de) (GS: la compañía *Microsoft*)))

Lo cual menciona que la actualización fue expresamente hecha por *Microsoft*, a través de una División de Investigación (aunque no necesariamente de dicha empresa).

CYK es completo en el sentido de que está garantizado encontrar todos los análisis definidos por las reglas. De esta forma, enumera cada hipótesis estructural que las reglas soportan, tanto para las partes de la oración como para la misma ya completa. Claro que no dice cómo decidir entre las hipótesis competentes, aunque cierta heurística puede ser ideada para ayudar a tomar estas decisiones. Se ha visto que es preferible una hipótesis que explique la oración completa que una subhipótesis que sólo explique partes de la misma. En ausencia de dicha hipótesis, se podrían preferir hipótesis incompletas que cuenten la cantidad de datos, por ejemplo, las palabras que mas se repiten.

El precio que se paga por tener un algoritmo muy completo es una complejidad polinomial[36]. Los ciclos de triple jerarquía de CYK establecían que el tiempo tomado para analizar una oración era una función cúbica de su longitud[37]. Esto es generalmente aceptable para una aplicación de extracción de información, donde el usuario (o desarrollador) no analiza las sentencias una por una.

6.5.3. Técnica de reconocimiento de entidades

El objetivo principal de esta tarea es el de reconocer automáticamente, por ejemplo, en textos notariales, nombres de personas, datos de inmuebles, direcciones, etc (lo que se llamará a partir de ahora "reconocimiento de entidades relevantes en el texto"). Tal y como se ha definido en la Muc-6, se entiende que una entidad es una expresión de nombres, es decir una serie de nombres propios que forman en conjunto el nombre de una organización, el de una persona, dirección, o también expresiones de tiempo (Cowie y Guthrie).

Sin embargo existe una serie de problemas para cumplir con este objetivo: el no disponer de información en el diccionario, la ambigüedad

36 Se refiere a que el tiempo utilizado por el algoritmo es una función polinomial del tamaño del problema. Es dada por una función de la forma $an^m + bn + c$, donde n es la llave de tamaño variable.

37 Esto era en el peor de los casos de análisis, que no siempre es hallado en la práctica, especialmente cuando se trataba de analizar sentencias largas usando un lexicón de palabras clave relativamente pequeño. Si se está procesando la fila i de la tabla, y j es la última fila donde se ha asignado una categoría no léxica, entonces sólo es seguir: if $i > 2j$, donde i es uniforme, e $i > 2j + 1$ de otra manera.

léxica y estructural, la ambigüedad del límite de la frase (que se acaba de tratar), etc., es decir, problemas propios del procesamiento del lenguaje natural difíciles de interpretar por una computadora.

Para ello se propuso un método de reconocimiento de entidades basado en las gramáticas SUG[38] que se consideran adecuadas para este tipo de tareas; a su vez se propone realizar análisis parciales del texto en lugar de un análisis completo que permitirá obtener la información relevante.

El objetivo de estos métodos no es sólo reconocer el grupo de expresiones que forman la entidad, es decir donde empieza y donde termina, sino también identificar el tipo de entidad de que se trata (persona, organización, dirección, etc.).

6.5.4. Análisis morfosintáctico para la extracción de información

Es importante el análisis de las entidades discursivas, esto es, unidades sintácticas a partir de información morfológica, según Chunk. Abney por su parte utiliza el adjetivo calificativo si éste aparece antepuesto al nombre (web.13).

La gramática utilizada para el análisis del español es libre de contexto según Chunk. Este tipo de análisis sirve para:

1) Desglosar un análisis por etapas o niveles para adquirir el conocimiento para el posterior análisis.
2) Definir un nivel intermedio de análisis sintáctico a partir de un análisis previo.
3) Manejar las dificultades de manera automática, por ejemplo, determinar el alcance de cada elemento; dos casos son el tratamiento del sintagma verbal[39] y desde el punto de vista teórico, incluye el verbo con sus argumentos. Otra dificultad es tener presentes los límites de las estructuras subordinadas pues no se sabe de manera exacta donde finalizan si no se tiene información de la estructura argumental tanto del verbo como de la estructura.

---

38 Gramática de Unificación de Huecos (*SUG, Slot Unification Grammar*), es un método de resolución de anáforas discursivas en textos no restringidos.

39 El sintagma verbal (SV), que funciona como predicado (P) de la oración, está constituido por un verbo, o por una expresión compleja que funcione como tal (perífrasis verbal y locución verbal), y unos complementos. El primero es necesario para que exista sintagma verbal; los complementos pueden aparecer o no, tal y como ocurre con los adyacentes del sintagma nominal.

Con esto se definió una nueva unidad sintáctica llamada chunks definida como núcleos léxicos. Esta formulación determinada presupone una configuración sintáctica aplicable al idioma inglés y a lenguas romances y muchas estructuras por tratar como los adjetivos (el hombre casado, el hijo del vecino, etc.) y la coordinación entre núcleos léxicos (sólo come pan y agua).

6.5.5. Análisis de chunks para el español

TACAT es un analizador sintáctico basado en *charts* y consiste en análisis de árboles utilizando la estrategia *bottom-up* (de abajo hacia arriba) y *left-right* (izquierda a derecha) con una gramática libre de contexto dónde se finaliza cuando no se dispone de inició y fin de la estructura.

6.5.5.1. Características del analizador respecto de la gramática

La regla de esta gramática consiste en un elemento a la izquierda y se entiende como cero y uno o más elementos a la derecha de dicha regla.

Elementos literales

Sintagma nominal (sn) se constituye por un verbo (hace), seguido de otro sintagma (meses) y se representa por:

- Hace meses.
- Hace varios meses.
- Hace cuatro meses.

Como el formalismo tiene muchas peculiaridades permite contextualizar reglas de la gramática.

Control de aplicación de las reglas en la salida

En ocasiones una o más reglas se aplican a una secuencia de palabras. Un analizador permite tener prioridades para la elección del mejor árbol creado para tener una salida del análisis óptimo. Estas reglas se establecen en orden de prioridad al final de la gramática.

Control de la salida del analizador

Es importante el control de la gramática para obtener una salida adecuada, esto se realiza mediante listas de gramática para facilitar el análisis. Si se desea mostrar sólo algunos nodos del árbol es necesario

utilizar listas *Hidden* (ocultas), *Group* (conjunto que incluye los lugares más altos del árbol), *Notop* (no aparecen los nodos cuando son los más altos del árbol) y *Flat* (ésta lista es recursiva).

Determinantes

Un determinante puede agrupar la información morfológica sobre el género y el número, sus adjetivos y posibles combinaciones y el uso de algunos adverbios como más, menos, casi, etc.

Núcleo

Los núcleos son los nombres comunes, propios y algunos pronombres.

Complementos del núcleo

Los complementos del núcleo pueden ser otro nombre o una preposición.

Existe un sistema llamado *Entity Recognition and Classification* (NERC), el cual permite reconocer entidades fuertes (pronombres personales) y débiles (nombres comunes). Los nombres comunes actúan como *trigger words* (término usado para definir que a partir de que se encuentra una cierta palabra, se realizará una acción por el sólo hecho de identificar esa palabra), acompañadas por un nombre propio. Las *trigger words* permiten identificar entidades fuertes y clasificarlas tanto semántica como morfológicamente. Por ejemplo en la entidad "El alcalde de Cuautitlán", la entidad fuerte de lugar no es el problema, sino mas bien identificar a que se está refiriendo, si a un nombre de persona, de empresa, partido político, etc., y esta información la proporciona una *trigger word*. Las *trigger words* se clasifican por tipología para identificar las entidades.

6.6. Técnicas de extracción de información

6.6.1. El uso del corpus para extracción de información y su representación

Un corpus es el texto del cual se parte para realizar el análisis, búsqueda o extracción de los datos. Dependiendo de los datos que se quieran obtener se tendrá una estructura de corpus diferente (Pérez y Chantal 2002).

Existes muchos tipos de córporas como:

- *British National Corpus* (BNC), tiene 100 millones de palabras, número de estilos, géneros y variedades de la lengua inglesa.

- Corpus de oncología: de 28 millones y medio de palabras. Se han obtenido estos de datos de situaciones comunicativas diversas entre especialistas, iniciados e intermedios y de profesores a alumnos.

En muchas investigaciones sobre extracción de información se suelen utilizar las herramientas de Michael Scott[40] llamadas *WordSmith* las cuales estudian el comportamiento de las palabras diseñadas para la terminología. Con estas herramientas se pueden realizar cálculos y análisis textuales como:

- *Word list*: listas de palabras en orden, las cuales se basan en el corpus y sirven para comparar e indexar las listas, con esto se acelera el tiempo de búsqueda.
- *KeyWords*: es una herramienta que permite extraer las palabras clave, además estudia el comportamiento de cómo éstas se distribuyen en el texto.
- *Concord*: herramienta utilizada para el análisis de patrones léxicos y agrupaciones de palabras.
- *Viewer*, *Splitter* y *Text Converter*: dividen el texto en partes más pequeñas para su análisis.

El número de *tokens* en un texto forma la longitud de la córpora, es decir, el número de palabras que componen el texto. La ratio se mide:

$$= \frac{\text{Número formas} * 100}{\text{Número palabras}}$$

A menor ratio mayor riqueza léxica y entre menor número de palabras mayor es la ratio.

También es posible establecer la media del corpus y se obtienen los *standarized type/token ratio*.

R = Riqueza léxica
N = Número de palabras en el texto
V = Número de formas en el texto
$\mathbf{V_1}$ = Número de palabras usadas una sola vez

$$R = \frac{100 \log N}{1 - (V_1/V)}$$

40 Creó para *Oxford University* el programa "*Wordsmith Tools"*.

6.6.2. Frecuencia de las formas en el corpus

El propósito de esta etapa es distinguir las palabras más frecuentes en el texto o corpus, esta repetición de palabras puede aportar mucha información tanto del tema como de las diferencias léxicas y conceptuales dependiendo del contexto de la especialidad. Algunas palabras frecuentes son en la mayoría de las ocasiones artículos, pronombres, preposiciones, etc., los cuales son excluidos y puestos en un archivo especial.

Se puede concluir que las palabras más frecuentes de un determinado corpus de oncología[41] tomado como ejemplo, se engloban en los siguientes grandes grupos:

- unidades léxicas que deben su presencia a la composición del corpus (es decir, al tipo de textos).
- unidades léxicas que deben su presencia al contenido de dichos textos.
    1) unidades léxicas del vocabulario médico general que también son de uso frecuente en la lengua general, en las que, la alta frecuencia de unidades que no son nominales hace suponer que también son importantes en la construcción del discurso científico.

Otro punto importante que hay que valorar es el número de veces que aparecen los términos científicos y de uso común, así como valorar el uso de artículos, sustantivos, además de siglas existentes de términos científicos.

Palabras clave en los textos

El motivo por el cual se cree que es importante el estudio de la frecuencia de patrones (plantillas) es el de identificar las palabras clave en el texto, es decir, si un patrón aparece en dos textos o córporas diferentes. En realidad las palabras clave no tienen que ser las más frecuentes en el texto, sino las más significativas con respecto al texto de comparación. Existen las palabras positivas que son aquellas que aparecen con mayor frecuencia en el texto y negativas, las que aparecen con menor frecuencia con relación de texto de comparación.

Esta comparación sirve para indicar la temática de los textos y son útiles para delimitar el área conceptual de la especialidad dada.

41 Parte de la medicina que trata de los tumores.

Enlaces de palabras clave

Se identifica la forma en que las palabras clave se distribuyen en el texto, así se observa qué parte del texto es más rica en contenido, pues es en donde hay más palabras clave. Se realiza un cálculo para saber la interrelación de las palabras clave y luego un nivel de concordancia de cada palabra clave.

6.6.3. Aplicación a otras áreas del PLN

Una de sus aplicaciones más interesantes es la introducción de los datos en una base de datos a partir de textos no estructurados. Es decir, que al aplicar las técnicas de extracción de información a un documento sin alguna estructura específica, sea posible obtener una plantilla o formulario (información estructurada) con la información que se tenga que guardar en la base de datos.

La extracción de información consiste en extraer de un texto o un conjunto de textos, entidades, eventos y relaciones entre ellos.

Por ejemplo, a partir de un conjunto de noticias sobre cambios de puestos directivos en empresas, podría interesar rellenar unas fichas incluyendo datos de cada evento de un cambio de una persona en un puesto dado. De modo que, por ejemplo, a partir del siguiente texto:

"Marco Rodríguez deja el puesto de vicepresidente de "Empresa Mexicana S.A." el 4 de marzo de 2006. Él será sustituido por Sandra Ayala".

El sistema debería ser capaz de deducir que:

- "Marco Rodríguez" y "Sandra Ayala" son personas, "4 de marzo de 2006" es una fecha, y "Empresa Mexicana S.A." es una organización. Esta primera tarea, que consiste en identificar entidades de diferentes tipos en el texto, se denomina *Entity Named Recognition.*
- Hay dos eventos en el texto: dejar un puesto y tomar otra persona el mismo puesto. Nótese que para saber que se refieren al mismo evento, puede ser necesario realizar la resolución de anáfora[42], descubriendo que el pronombre "Él" en la segunda frase se refiere a Marco.
- Los dos eventos tienen la misma fecha: 4 de marzo de 2006, y ambos se refieren al mismo puesto en la misma empresa. La

42 Asumir el significado de una parte de un discurso ya emitida.

persona implicada en cada uno de ellos es diferente, dado que dejar el puesto es el evento que se aplica a Marco y tomar el puesto es el que se aplica a Sandra.

En algunas ocasiones los tipos de entidades se subdividen en subtipos hasta varios niveles, de manera que puede ser necesario por ejemplo, no sólo identificar los lugares, sino distinguir cuáles son países, cuáles son ciudades y cuáles son accidentes geográficos. A continuación, entre las ciudades se podría distinguir si son o no capital de algún país; y entre los accidentes geográficos si son montañas, valles, etc. En estos casos, el problema de reconocimiento y clasificación de entidades con nombre se hace muy cercano al de población de oncologías.

6.7. EI vs. RI

Las técnicas de Recuperación y Extracción de Información actualmente son muy buscadas por los usuarios, aunque en mayor medida la primera de éstas; ya que ahora se cuenta con grandes cantidades de información digitalizada a través del internet. Estas dos técnicas, ayudan a separar y organizar toda esa información.

La Recuperación de Información (*Information retrieval*), como fue mencionado en la 42ª Reunión anual de Lingüística Computacional, consiste en: seleccionar los documentos que poseen la información requerida por el usuario, de una colección documental que puede ser pequeña o muy grande y como resultado, traerá una serie de documentos o páginas que sean potencialmente relevantes, de acuerdo con los términos de la búsqueda, ejemplo de este tipo de técnica son los buscadores como: *Yahoo* y *Google*. Es una disciplina computacional orientada a obtener textos que contengan la información deseada a partir de palabras clave con las cuales se alimenta el motor de búsqueda que las toma como objetivos a encontrar, a diferencia de la extracción de información, la recuperación de información no se preocupa por el contenido lingüístico.

La Extracción de Información va mas allá de extraer sólo documentos y por lo tanto es más compleja, en un principio se obtiene el corpus al igual que la recuperación de información, pero además, deben encontrarse y etiquetarse los datos relevantes para el usuario, descartar todo lo irrelevante y después organizar la información de una manera comprensible para él, es por eso que, para esta técnica se hace uso del

Procesamiento del Lenguaje Natural ya que se encuentran diversos problemas lingüísticos como la ambigüedad de las palabras.

De alguna manera la recuperación de información se puede clasificar como un paso previo a la extracción de la información ya que los resultados logrados por la primera muchas veces conforman el material de trabajo de la segunda.

6.8. Arquitectura

La arquitectura general para la construcción de sistemas de extracción de información es definida por Hobbs[43], y consiste en una cascada de módulos que en cada paso agregan estructura y algunas veces filtran la información relevante, a través de la aplicación de reglas, el modelo se muestra a continuación (Tellez 2005).

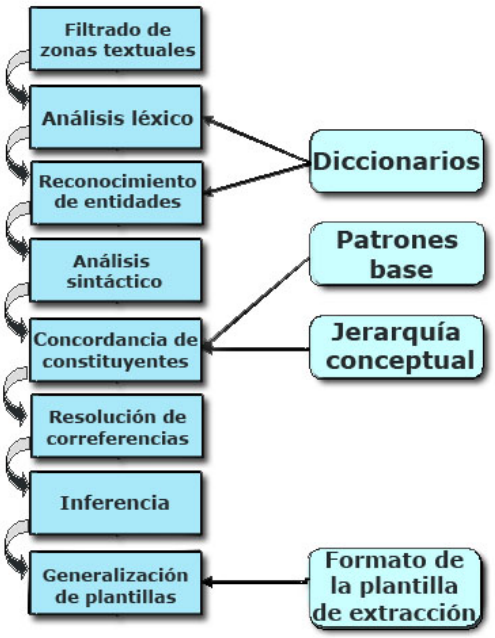


Figura 6.4.-Arquitectura general de los SEI (imagen obtenida de Tellez 2005:34).

La arquitectura de una aplicación de extracción de información no se diferencia de muchas aplicaciones informáticas como los sistemas operativos o protocolos de comunicación en su manera de estructurarse, sino que la extracción de información está apoyada en un modelo de capas que va de lo particular a lo universal.

43 J. Hobbs: escribió "*The generic information extraction system*" dentro del marco de la Muc-5.

Filtración de zonas textuales.

- División del texto en segmentos como párrafos.
- Dividir el texto en oraciones.
- Eliminar marcas en el texto ajenas al lenguaje, por ejemplo en documentos HTML serian las etiquetas o *tags*.
- Tokenización, dividir por palabras.

Análisis léxico.

- Procesa los *tokens* desde un punto de vista léxico, esto se refiere a que se busca la raíz de cada palabra quitando prefijos, sufijos, conjugaciones, etc.
- Desambiguación: trata de obtener el significado y la función específica de una palabra cuando ésta tiene muchas funciones o significados.
- Todo esto con ayuda de diccionarios.

Análisis sintáctico e interpretación semántica.

- Su objetivo es hacer un análisis tomando como base el análisis anterior y tratando de darle un significado a la oración, para esto lo más común es tratar de definir si la oración es gramaticalmente correcta, a esto se le llama análisis sintáctico total. También existe el análisis sintáctico parcial que es menos rígido que el anterior en cuanto a encuadrar un texto en un perfil estrictamente dado, pues se centra en analizar sólo frases.
- Concordancia de patrones: una vez etiquetado el documento de manera sintáctica, se procede a establecer relaciones entre el dominio de extracción y sus constituyentes.
- Relaciones gramaticales: determinan las dependencias entre los constituyentes usando patrones sintácticos flexibles.

Análisis del discurso

- Resuelve aspectos como las anáforas o elipsis (emitir en el texto una o más palabras) dentro del texto que se está procesando.

Generación de plantillas de salida

- Es finalmente empalmar la información recabada con las plantillas previamente definidas.

6.9. Algoritmos para extracción de información

Naive Bayes[44]

Este algoritmo trabaja sobre un enfoque probabilístico donde las decisiones son tomadas con base en las diferentes probabilidades generadas en función al teorema de Bayes[45] y los datos observados, dejando el espacio de que una de las variables puede ser independiente.

C4.5

El algoritmo C4.5 fue diseñado como una adición al algoritmo ID3 en el cual se basa, este último trabaja con base en árboles de decisiones donde los nodos internos forman parte de los atributos y de esa manera los textos se van desgajando.

k-Vecinos más cercanos

Un *software* de clasificación es puesto a trabajar dentro de un corpus que contiene información exclusivamente aceptada como válida, en base a los resultados de estas búsquedas genera plantillas que contienen los pasos usados para la clasificación de cada caso, así que cuando se le presenta una nueva situación lo que hace es clasificarla de acuerdo a lo ya establecido en su programación y para decidir si es una información válida o no compara la platilla generada de la clasificación y la contrasta con las que ya posee, de esta manera sabe si son válidas, sin son parecidas lo marca como dentro del rango de búsqueda.

Máquinas de vectores de soporte

En este algoritmo de alguna manera se tienen los casos exitosos y los casos fallidos, cuando surge un nuevo caso busca tender un análisis geométrico y busca si este análisis se parece más al éxito o al fracaso de los que ya tiene memorizados.

---

44 Thomas Bayes: nació en Londres Inglaterra (1702-1761), fue un brillante matemático. Sus estudios en probabilidad son reconocidos por el teorema que lleva su nombre el cual ayuda a resolver probabilidades de un suceso que se presenta como suma de diversos sucesos mutuamente excluyentes. Miembro de la *Royal Society* desde 1742, Bayes fue uno de los primeros en utilizar la probabilidad inductivamente y establecer una base matemática para la inferencia probabilística.

45 El Teorema de BAYES se apoya en el proceso inverso al que se denomina Teorema de la Probabilidad Total:

- Teorema de la probabilidad total: a partir de las probabilidades del suceso A (probabilidad de que llueva o de que haga buen tiempo) se deduce la probabilidad del suceso B (que ocurra un accidente).
- Teorema de Bayes: a partir de que ha ocurrido el suceso B (ha ocurrido un accidente) se deducen las probabilidades del suceso A (¿estaba lloviendo o hacía buen tiempo?).

6.9.1. Extracción de información vista desde la clasificación de textos

Hoy en día se cuenta con mucha información en medios de difícil acceso, por ejemplo el texto, y aún más cuando éste no está estructurado, esto conlleva a la tarea de crear aplicaciones capaces de realizar dicha función.

Una de las herramientas surgidas de esta necesidad es la recuperación de información, pero ésta tiene la desventaja de arrojar mucho material innecesario, es por ello que la propuesta de la clasificación de textos versa acerca de cómo estructurar el material para poder ser analizado.

Una de las formas mas óptimas de estructurar la información es con la ayuda de la Lingüística, que ahora se le ha nombrado extracción de información, este paradigma se basa en tratar de comprender parcial o totalmente el contenido para intentar responder preguntas acerca de lo que trata el documento, provocando que estas aplicaciones contengan un alto contenido lingüístico dado generalmente por expertos en el área, además de que las aplicaciones resultantes se vuelven tan específicas que difícilmente su desarrollo aporta experiencia a otras áreas.

La experiencia ha mostrado que una de las maneras más eficientes de desarrollar aplicaciones es mediante corpus previamente etiquetados para que el programa se entrene en adquirir los conocimientos necesarios.

El punto de vista ofrecido por la extracción de información con algoritmos de clasificación está basado esencialmente sobre una base empírica, de tal modo que no conlleve un pesado y riguroso análisis lingüístico, se pretende de esta manera evitar problemas como lo son la falta de portabilidad entre idiomas o contextos de los desarrollos hechos sobre una profunda base lingüística.

Se toma el punto de que para extraer información es necesario identificar entidades rodeadas por ciertos patrones de palabras y el identificar estas estructuras en los documentos permitirá elegir si un documento sirve o no. Con esto se deduce que únicamente es necesario reconocer estos patrones para cimentar en base a ellos la aplicación que fundamentará esta tesis. El problema versa ahora en cómo identificar los patrones y las entidades. La respuesta que se ofrece a la primera cuestión es el aprendizaje automático enfocado a estructuras regulares que permitan integrar el menor conocimiento lingüístico y con ello aumentar la portabilidad entre idiomas.

6.9.2. Técnicas de extracción de información

Durante varios de los últimos años, las compañías han venido usando las técnicas de extracción de datos para conocer las exigencias de los clientes y para proporcionarles así interacciones personalizadas. Existen varias técnicas de extracción de datos desarrolladas para identificar tendencias y nuevas oportunidades de tratamiento de datos ocultas. Varias de estas técnicas de *Data Mining*[46] se han insertado en aplicaciones de *software* que procesan complejos algoritmos para obtener un significado coherente de la información.

Mientras las aplicaciones de *Data Mining* del destinatario final están al alcance de cualquiera, estas aplicaciones no se desarrollan extensamente a través de las organizaciones debido a que habitualmente se desconoce su utilidad. Una forma de descubrir las posibilidades de la extracción de datos es comparándola con otras estrategias de espionaje industrial (*business intelligence, BI*)[47].

6.9.2.1. Aplicación de consulta personalizada

Mediante la aplicación de consulta personalizada los usuarios tienen la posibilidad de acceder a la información deseada. Por ejemplo, un usuario crea y ejecuta una consulta personalizada que contesta al interrogante: ¿qué cantidad de ingresos ha generado cada cliente durante este año?. Los resultados de esta consulta contendrían los nombres de los clientes y los ingresos para el año seleccionado. La figura 6.5 representa los resultados de la consulta personalizada.

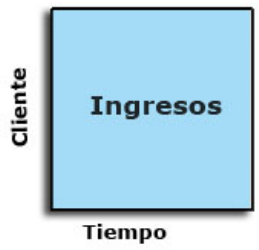


Figura 6.5.- Resultados de la consulta personalizada (imagen tomada de web.91).

46 Es la extracción de información oculta y predecible de grandes bases de datos. La idea es descubrir tendencias y comportamientos que, aplicados a la industria, puedan resultar de valor económico a la hora de implementar campañas o lanzar productos, etc.

47 Es un modelo que habla de concentrar todos los sistemas de información de una compañía con el objetivo no sólo de concentrar información sino, darle un uso realmente inteligente, todo esto enfocado al conocimiento de la empresa en sí pero enfocadas también al cliente, pues un conocimiento profundo de éste permite a las empresas desarrollar mejores productos y servicios que le dan ventajas competitivas.

Los ingresos por cliente podrían dar origen a otra pregunta: ¿qué cantidad de ingresos se generan al año?. Además, también podrían responderse otras preguntas como: ¿qué cliente generó la mayor cantidad de ingresos para la compañía? y ¿qué cliente generó la cantidad más baja?. Mientras que el resultado de la búsqueda sea útil y resuelva varias cuestiones, la tecnología BI no identificará los modelos poco habituales ni revelará las relaciones anormales. Lo que el usuario solicitó eran los ingresos por cliente para el año actual y ésta es la información que fue proporcionada; ni más ni menos (web.91).

6.9.2.2. Procesamiento analítico on line (OLAP)

Las aplicaciones OLAP permiten a los usuarios explorar y analizar manualmente la información resumida y detallada. Por ejemplo, un usuario crea y ejecuta un análisis OLAP que responde a la pregunta: ¿cuáles fueron los ingresos para cada trimestre de este año por región geográfica y cliente?. Los resultados de este análisis contendrían región geográfica, nombre del cliente, ingresos y los trimestres seleccionados. La figura 6.6 representa el resultado del análisis OLAP (web.91).

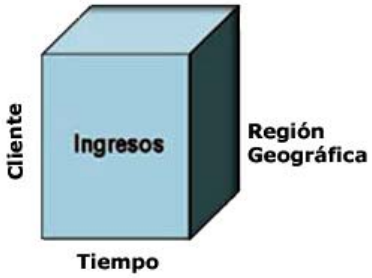


Figura 6.6.- Ejemplo de un resultado olap (imagen tomada de web.91).

Podrían formularse preguntas adicionales acerca de los datos que subrayasen a los modelos de ingresos temporales por región geográfica. Sin embargo, debe dirigir este proceso un usuario que sepa cómo tratar los datos. El OLAP sólo puede subrayar los modelos de los datos que se solicitaron. Es competencia del usuario identificar las tendencias y modelos subrayados por el análisis OLAP. Esta tecnología BI no identificará relaciones poco usuales ni revelará relaciones ocultas (web.91).

6.9.2.3.Data mining

Para describir mejor el *Data Mining* puede decirse que es una tecnología BI que cuenta con diversas técnicas para extraer información útil, oculta y comprensible de un conjunto de datos. El *data mining* posibilita el descubrimiento de tendencias y modelos ocultos en extensas cantidades de datos. El resultado de un ejercicio de *data mining* puede tomar forma de modelos, tendencias o reglas implícitas a los datos.

Existen diversas técnicas de *data mining* de las que puede hacerse uso; cada una de ellas sirve para un propósito específico y varía la cantidad de participación del usuario. La figura 6.7 representa la progresión de las técnicas de *data mining* según la participación del usuario.

Figura 6.7.-Técnicas de *Data Mining* (imagen tomada de web.91).

Las Redes Neuronales son sistemas altamente evolucionados que proporcionan modelos predictivos. Estos sistemas son muy complicados y lleva su tiempo adecuarlos para que actúen de una forma similar al pensamiento humano. Esta técnica de *data mining* se ha usado para detectar potenciales transacciones fraudulentas con tarjetas de crédito.

La Inducción es una técnica de *data mining* que deriva en reglas inherentes a los datos. Las reglas se utilizan para entender las relaciones existentes. Un ejemplo clásico es que el 50% de las veces que una persona compra pañales, también compra cerveza.

La Estadística es la base de toda técnica de *data mining* y requiere individuos altamente cualificados en matemáticas que construyan e interpreten los resultados.

La Visualización representa los datos en mapas gráficos o tridimensionales, permitiendo así al usuario que identifique tendencias, modelos y relaciones. Mientras que una imagen que se produce proporciona otra perspectiva de las relaciones entre los datos, la

visualización está a menudo incorporada en las aplicaciones de *data mining*.

Mientras la *Gartner Group*[48] enumera el lenguaje de la OLAP y de búsqueda como técnicas de *Data Mining*, el grado de implicación del usuario conlleva extensible y extremadamente más tiempo cuando se trata de identificar tendencias y relaciones ocultas. Por lo tanto, el uso de estas técnicas no resulta rentable.

Cuando un usuario utiliza una aplicación de *data mining* puede preguntar: ¿cuáles son las características distintivas de aquellos clientes que pagan en el plazo establecido?. Los resultados del ejercicio de *data mining* servirían entonces para establecer la condición de una consulta personalizada que identificase los nombres de los usuarios y la información de contacto en la base de datos para los propósitos de servicios adicionales de venta cruzada.

Las aplicaciones de consulta personalizada dan un valor superficial al existente dentro de la base de datos, mientras que OLAP proporciona usuarios con una gran capacidad de profundización y comprensión. Sin embargo, el *data mining* llega más lejos y proporciona a los usuarios conocimiento a través del descubrimiento de tendencias y relaciones ocultas. La combinación del *data mining* con una consulta personalizada o una aplicación OLAP es extremadamente poderosa y proporciona a los usuarios conocimiento sobre los datos que están siendo analizados y la capacidad de actuar según ese conocimiento. La figura 6.8 representa el valor y el propósito de estas tecnologías BI (web.91).

48 Una de las mayores empresas de consultoría, investigación y análisis sobre la industria de Tecnologías de Información global en el mundo.

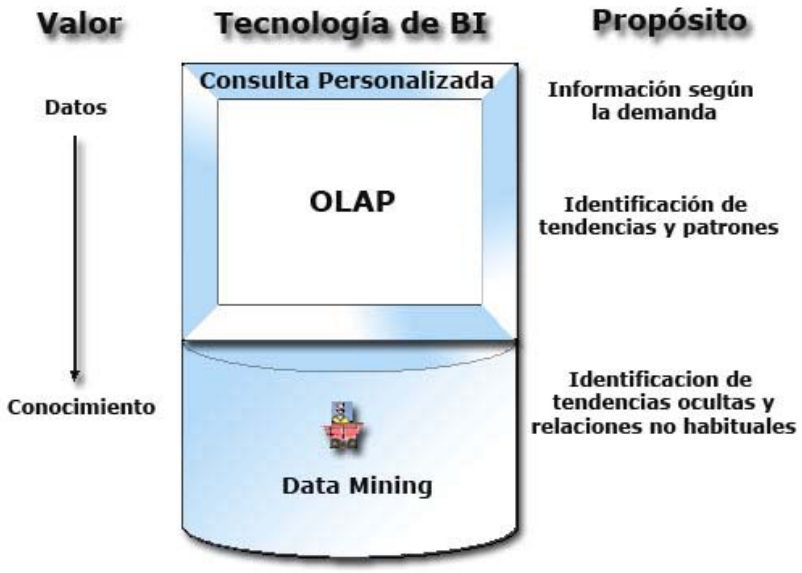


Figura 6.8.- Tecnologías de BI (imagen tomada de Web.91).

Capítulo 7

7. Metodología para el desarrollo de sistemas de extracción de información (caso práctico)

7.1. Objetivos de la aplicación

- Desarrollar una aplicación que permita recuperar entidades relacionadas al lanzamiento de nuevos productos tecnológicos de *software*. La extracción será hecha a partir de noticias en formato de texto libre.
- Poner en práctica la metodología.
- Comprobar con datos de prueba la hipótesis de esta tesis.

7.2. Análisis

7.2.1. Planeación

Se definió la siguiente planeación a seguir en la metodología para la realización del sistema "SEINS", representada en la siguiente gráfica de Gantt.

Se piensa comenzar el día 6 de mayo de 2006 y terminar el sábado 17 de junio del mismo año.

Como se puede observar, a la etapa de análisis del sistema se le van a dedicar dos semanas, mientras que el diseño se hará en una semana. La parte de desarrollo, que entre otras cosas trata mucho de "ensayo y error", se piensa terminar en poco más de tres semanas. Por último las etapas de implementación y pruebas finales tendrán una duración de dos días cada uno.

En caso de que se requiera ajustar cualquier asunto no definido u omitido por equivocación, se tiene un día de desfase.

7.2.2. Recopilación del corpus de entrenamiento

En primer lugar es importante definir la información que se va a obtener mediante el proceso de extracción de información además de la fuente donde se encuentra dicha información, a esta fuente se le llamará "corpus". El desarrollo del corpus se dio a partir de búsquedas en internet con palabras claves como "*software*", "nuevo", "presentación", "lanzamiento", "programa" y constantes visitas a páginas de noticias de tecnologías ya conocidas.

Con el corpus de entrenamiento primeramente fue realizado un análisis sobre la forma en la que se encuentran divididas las noticias así como su redacción, es decir, el tipo de palabras que se utilizan y de éstas cuáles son las más comunes, dicho en otras palabras, se comenzó analizando la estructura del corpus y alrededor de toda esta recopilación surgió lo siguiente.

Es necesario realizar un análisis de los requerimos o necesidades de la información que se necesitan, para posteriormente obtener el corpus para la extracción de información.

Para poder almacenar el corpus de manera segura (no almacenar noticias repetidas) y también para que los integrantes del equipo pudieran conocer qué páginas estaba ocupando cada uno para efectos del corpus, se realizó una pequeña interfaz *web* –véase figura 7.1- alojada en un principio en el servidor "http://ayax.iingen.unam.mx" del Instituto de Ingeniería con un formulario que obtenía los datos principales de la noticia de la página de donde se consiguió la misma, tales como el *link*, integrante del equipo que la encontró, fecha en la cual se capturó la noticia además de la publicación. Con esto se obtuvo un control formal del corpus y permitió consultar el origen de las noticias o corpus de estudio en línea.

Debido a cambios internos en el Instituto de Ingeniería de la UNAM esta página está deshabilitada, pero aún así se cuentan con los archivos que forman el corpus en archivos de txt.

Figura 7.1.- Interfaz *web* para el almacenamiento de las noticias.

En un principio se decidió que el corpus fuera sólo de productos nuevos de *software*, pero después se descubrió que también eran necesarias las actualizaciones de dichos *softwares* para que las empresas tuvieran la última versión (*service packs*, actualizaciones antivirus, parches, etc.).

Más tarde se encontró que las noticias en internet no tenían un formato estándar, por ese motivo fueron llamados "datos no estructurados" -véase cap. 2.3- ya que sus datos se encuentran en diversas partes del texto y sus elementos no tienen una estructura específica como los datos que se encuentran en las bases de datos.

Siguiendo esta pauta se vio que en la actualidad, dentro del área de informática, a diario sale al mercado *software* de todo tipo, tanto comercial como de distribución libre, por ejemplo, en las organizaciones se necesita información al día acerca de dicho *software*, así como de las actualizaciones que puedan ayudar a desarrollar cada una de las actividades de dicha corporación de manera más eficiente y rápida. Por esta razón esta tesis se basará en corpus de noticias sobre productos

nuevos de *software*, así como las actualizaciones del mismo que salen al mercado.

Ahora, se extraerá información mediante un análisis lingüístico (que si bien no tiene una estructura como una base de datos, si tiene estructura con respecto al lenguaje y sintaxis que construyen dichas noticias).

A partir del corpus recolectado de noticias sobre lanzamientos y actualizaciones de *software*, se escudriñará la estructura del lenguaje que lo compone a partir de cada uno de los elementos en los que está formado el lenguaje, tales como verbo, sustantivo, preposición, artículo, conjunción, adjetivo, etc., pues al identificar qué función tiene cada uno de estos elementos en el texto será posible saber que un sustantivo se refiere al nuevo producto de *software* y no a la compañía que lo vende, que el verbo lanzó (el más utilizado en las noticias del corpus) puede estar conjugado en diferentes tiempos pero a pesar de todo se refiere a un nuevo producto de *software* que salió (o saldrá) al mercado, ya sea comercial o no. Y ahora la pregunta es: ¿cómo identificar las partes del lenguaje?, la respuesta es mediante el etiquetado pos, que a su vez se divide en dos partes muy importantes:

- En la estructura general de cada noticia.

En otras palabras cada noticia es dividida en los elementos generales que pueden contener las noticias relacionadas con el lanzamiento o actualización del *software*, como son, la noticia en sí, la parte (párrafo) en donde se describe que se lanza o anuncia un producto o actualización de *software*, dentro de dicho párrafo se encuentra el indicador o *trigger word* (palabra que indica que es un nuevo *software*), nombre del producto, fecha de lanzamiento, versión, compañía, el título y liga de la noticia. De esta forma se le dio estructura a los elementos del corpus. Ahora se dividirá el mismo en los diferentes elementos del lenguaje.

- En identificar los elementos del discurso o etiquetado POS.

Con este etiquetado se hace referencia a identificar palabra por palabra dentro de cada una de las estructuras definidas anteriormente, es decir, dentro del disparador (párrafo o palabras que indican un lanzamiento de *software*) existen palabras que generan enunciados y estos a su vez generan ideas complejas que pueden servir para identificar la información que se desea extraer.

Por ejemplo en la oración:

"El nuevo *software SQL Server*".

La palabra "El" se sabe que es un artículo determinado, el cual define que la siguiente palabra puede ser un sustantivo (en la mayoría de los casos pero no en todos), pero aquí se observa que posteriormente hay un adjetivo el cual es "nuevo", seguido de dos sustantivos a los que se refiere tanto el artículo como el adjetivo, que en este caso son "*software*" y "*SQL Server*". Y también se sabe que el adjetivo describe a los sustantivos. Todo este análisis se realizará con ayuda de la estructura del lenguaje a partir de criterios morfológicos, sintácticos, semánticos y ortográficos.

7.2.3. Etiquetado

En un principio, pensando en la forma en la que se podrían encontrar las noticias a recopilar, se trató de definir las etiquetas, acordando que el etiquetado se hará en XML. Bajo este supuesto también se pensó que la noticia tiene un título, e inmediatamente después la compañía que lo da a conocer, por consiguiente (en la mayoría de los casos) el producto seguido de la versión (aunque la versión no necesariamente es obligatoria). La versión podía contener números enteros o decimales y después del nombre del producto debía encontrarse una descripción de éste. Prácticamente todo lo demás no es de interés para el propósito del análisis, o por lo menos no en este momento.

Después de reunir el corpus y hacer un análisis del mismo el equipo se fue dando cuenta de que el primer formato de etiquetado (bosquejo del primer etiquetado) no tenía un buen diseño, ya que las noticias no caían en una regla general, sino más bien en algunos de los supuestos que fueron planteados (aunque en otros ni siquiera se encuadraban). Se comprobó que la estructura de las noticias era más complicada, por ejemplo, uno de los primeros supuestos era que se iba a encontrar la información más relevante (título, nombre del producto, compañía que lo desarrolló y breve descripción del mismo) en un sólo párrafo y que en la gran mayoría de los casos sería el primero, pero al analizar cada una de las noticias se fue aclarando el hecho de que la información, en muchos casos, no estaba en un sólo párrafo y mucho menos en el primero.

Bosquejo del primer etiquetado semántico:
<link>http://www.zonanat.com/SoftWare/Noticias/Octubre_05/lanzamiento_de_mysql_5_251005.htm**</link>**
<public>Zona N.A.T</public>
Lanzamiento de MySQL 5.0
2005
<noticia><lanzamiento><fecha><adv>Ayer**</adv>**
<adv>24/10/2005**</adv></fecha> <pron tipo=se>**se**</pron>** **<vb>**lanzó**</vb>** la nueva versión **<producto><sus tipo=np>**MySQL**</np> <num>**5.0**</num></producto>** que es el servidor de base datos de código abierto más popular en Internet con más de 6 millones de instalaciones funcionando.**</lanzamiento>**
MySQL 5.0 combina fiabilidad y funcionamiento, siendo la solución más rentable en bases de Datos para las empresas y particulares.
Esta nueva versión trae nuevas funcionalidades como una herramienta para poder trasladar datos desde otros servidores de Bases de datos como SQL Server de Microsoft, access, un asistente remoto de gestión para poder arrancar y parar remotamente MySQL, nuevas herramientas visuales para controladores para MySQL's ODBC, Java and .NET...
Esta versión estará disponible para Linux, Windows, Solaris, OS X, FreeBSD, Hp-ux, IBM AIX 5L..., bajo doble licencia pudiendo elegir entre la GPL o la comercial de MySQL.
Este nuevo lanzamiento supone un gran avance respecto a la ultima versión la 4.1.14 y posiblemente consiga muchos más adeptos de los que actualmente tiene.
</noticia>

También fue saliendo a la luz que conformar un corpus de 120 elementos no era suficiente, así que se decidió incrementarlo a 240, por lo que la definición de un DTD -véase cap. 5.2.5- en estos casos es primordial, ya que sin éste no se puede comenzar a realizar el etiquetado.

Se ha podido definir que lo que interesa rescatar de las noticias es la fecha de la publicación de la noticia, su título, la compañía que lo da a conocer, el nombre del producto, la versión del mismo, la fecha en que se lanza (o se lanzó, para el caso del corpus de entrenamiento) y una breve descripción, la cual sería la palabra (o conjunto de palabras) que indiquen que se trata de la publicación de un nuevo producto de *software*, aunque no necesariamente se encuentran todos estos elementos en todas las noticias. Por último se trató de identificar en la noticia todo lo que no sea de relevancia[49], como pueden ser las

49 Cabe mencionar que se llamará información irrelevante a los párrafos dentro de la noticia que en un principio muestran demasiada información para aquellas personas no interesadas en esa noticia. Para una persona interesada en una noticia (o varias) sobre un

personas encargadas de hacer la presentación, el lugar, descripción de las versiones anteriores el nombre de las personas que desarrollaron el producto, etc., y por esta necesidad se realizó un DTD que se puede consultar en el anexo A –véase cap. 11-.

7.2.4. Estándar de etiquetado Eagles

Expert Advisory Group on Laguage Engineering Standars es una iniciativa de la UE coordinada por expertos, que se ocupa de evaluar los modelos existentes para después elaborar recomendaciones de estandarización encaminadas a armonizar los trabajos que se realicen en el ámbito de la ingeniaría lingüística.

7.2.4.1. Etiquetado morfosintáctico

Se entiende por etiquetado morfosintáctico a la anotación de la categoría a la que pertenecen las palabras en un corpus. Se llamará indistintamente a las clases de palabras como "partes de la oración" o *Part-Of-Speech Tagging* (etiquetado de las partes de la oración). Generalmente para el inglés se dice *Tagging*, cuando se refiere al etiquetado de las partes de la oración, más que a cualquier otro tipo de etiquetado (sintáctico, semántico, etc.).

Existen cuatro puntos secuenciales a considerar para obtener el etiquetado de las partes de la oración

- Identificación de las palabras o unidades léxicas a etiquetar.
- Definición de las clases de palabras, desde el punto de vista gramatical, que quiere realizarse.
- Definición de las etiquetas con las que se van a anotar las clases de palabras.
- Procedimiento con el que se va a etiquetar el corpus (web.24).

7.2.4.2. Métodos para etiquetar

Existen diferentes métodos para etiquetar las partes de la oración, a continuación se describen tres formas (web.24).

- Etiquetado basado en reglas: mediante una base de datos con reglas de desambiguación que indican, por ejemplo, que una palabra ambigua es sustantivo, en lugar de verbo, cuando va

---

software específico que está buscando, es claro que toda la información de esa noticia es relevante.

después de un determinante (por ejemplo, "siempre sigo mi camino."). El método basado en reglas consta de dos etapas. En la primera etapa, se utiliza un programa para identificar las posibles partes de la oración de cada palabra, a partir de un lexicón en donde a cada palabra le corresponde su o sus partes de la oración. En la segunda etapa, un programa con un conjunto de reglas para desambiguar las palabras.

- Etiquetado estocástico: se utiliza un corpus entrenado para calcular la probabilidad de que una palabra tenga cierta etiqueta dado un contexto determinado.

  El fundamento de los métodos estocásticos se basa en una selección de "escoja la etiqueta más probable de esta palabra", basada en el enfoque *Bayesiano*. Para una oración o secuencia de palabras dadas, los algoritmos basados en las cadenas de Marcov seleccionan la secuencia de etiquetas que maximicen la siguiente fórmula:

  P (palabra | etiqueta) * P (etiqueta | n etiquetas previas)

  Los modelos basados en las cadenas de Marcov seleccionan una secuencia de etiquetas para una oración completa, más que para una palabra sola.

- Etiquetado basado en transformación: el más conocido es el *Brill tagger*, que comparte características de los dos algoritmos anteriores. Se basa en reglas para determinar cuando una palabra ambigua debe tener cierta etiqueta, y a la vez tiene un componente de aprendizaje en donde las reglas son automáticamente inducidas de un corpus entrenado previamente.

Existen varios estándares para etiquetar los corpus de manera morfológica, entre otros está el sistema de etiquetado del *British National Corpus* (BNC), el cual es un método muy reconocido para etiquetar (Pérez y Chantal 2002).

Expert Advisory Group on Language Engineering Standards (*Eagles*) es un grupo encargado de la asignación de estándares para el etiquetado morfológico de las palabras, también llamado analizador morfológico. Éste es patrocinado por la Unión Europea desde 1993. *Eagles* ha apoyado a grupos de expertos para el desarrollo de estándares pre-normativos. Actualmente existen cien centros de investigación, entre ellos, organizaciones industriales, asociaciones

profesionales y redes de investigación, todos ellos trabajan para la creación de métodos para la descripción, representación, evaluación y asesoría de los recursos lingüísticos y la tecnología asociada en los dominios de procesamiento del lenguaje natural (web.18).

El estándar de *Eagles* se encarga de la anotación morfosintáctica de lexicones y corpus para todas las lenguas europeas, es decir, abarca la sintaxis de varias lenguas, lo cual hace que cada quien adapte este etiquetado de acuerdo a su propia lengua, es decir, algunos atributos no pueden especificarse para el etiquetado del español; por esta razón, para efectos de esta tesis se adaptaron algunas etiquetas de dicho estándar de acuerdo al idioma español (web.16).

Eagles proporciona una serie de recomendaciones para reconocer las partes de la oración en tres niveles:

- Características obligatorias: son aquellas partes de la oración básicas que deben ser anotadas en cualquier etiquetado de las partes de la oración. *Eagles* reconoce las siguientes como principales: sustantivo, verbo, adjetivo, pronombre/determinante, artículo, adverbio, adposición (preposición), conjunción, numeral e interjección.
- Características recomendadas: aquellas categorías gramaticales ampliamente reconocidas y que deben ser anotadas de ser posible. Por ejemplo, para el sustantivo: número, género y tipo (común o propio, por ejemplo).
- Características opcionales: aquellas que pueden ser usadas para propósitos específicos, pero que no son lo suficientemente importantes para ser consideradas obligatorias o recomendadas.

7.2.5. Etiquetado con *Eagles*

A continuación se describe cada una de las categorías que fueron utilizadas para el análisis morfosintáctico de ésta tesis, es decir, el estándar para etiquetado *Eagles* (web.19).

Categorías

No.	Etiqueta de Eagles	Significado en español
1.	N [noun]	Sustantivo
2.	V [verb]	Verbo
3.	AJ [adjective]	Adjetivo
4.	PD [pronoun/determiner]	Pronombre
5.	AT [article]	Artículo
6.	AV [adverb]	Adverbios
7.	AP [adposition]	Preposiciones
8.	C [conjunction]	Conjunción
9.	UN [numeral]	Numeral

A continuación se describen los atributos que fueron utilizados para cada una de las categorías mencionadas en la tabla anterior.

***Nouns* (N)**

Para los sustantivos se tomaron y adaptaron los atributos que *Eagles* estableció y que a continuación se despliegan.

(i)	Type (Tipo):	1. Common (común)	2. Proper (propio)
(ii)	Gender: (género)	1. Masculine (masculino)	2. Feminine (femenino)
(iii)	Number (Número):	1. Singular (singular)	2. Plural (plural)

Por ejemplo, "país" se etiquetaría así: país_N111

N	1	1	1
país	i. tipo: común	ii. género: Masculino	iii. número: singular

***Verbs* (V)**

A continuación se muestra la manera de cómo se representaron los atributos para cada verbo.

(i)	Person: (Persona)	1. First (primera)	2. Second (segunda)	3. Third (tercera)	
(iii)	Number: (número)	1. Singular (singular)	2. Plural (plural)		
(v)	Verb form / Mod: (modo)	1. Indicative (indicativo)	2. Subjunctive (subjuntivo)	3. Imperative (imperativo)	4. Condicional (condicional)
		5. Infinitive (infinitivo)	6. Participle (participio)	7. Gerund (gerundio)	
(vi)	Tense: (tiempo)	1. Present (presente)	2. Imperfect (imperfecto)	3. Future (futuro)	4. Past (pasado) Pretérito perfecto simple
(vii)	Voice: (voz)	1. Active (activa)	2. Passive (pasiva)		

Por ejemplo, en el verbo: "bailamos", su representación mediante la etiqueta sería: V12111

V	**1**	**2**	**1**	**1**	**1**
bailamos	i. persona: primera	iii. número: plural	v. modo: Indicativo	vi. tiempo: presente	vii. voz: activa

***Adjectives* (AJ)**

Aquí se muestra cada uno de los atributos acordados para los adjetivos.

(i)	Degree: (grado)	1. Positive (positivo)	2. Comparative (comparativo)	3. Superlative (superlativo)
(ii)	Gender: (género)	1. Masculine (masculino)	2. Feminine (femenino)	
(iii)	Number: (número)	1. Singular (singular)	2. Plural (plural)	
(v)	Type: (tipo)	1. Posesivo	2. Demostrativo	3. Cuantitativo

Por ejemplo; en el enunciado "la casa bonita", para la palabra "bonita" la etiqueta sería: AJ1212.

AJ	1	2	1	2
bonita	i. grado: positivo	iii. género: femenino	v. número: singular	vi. tipo: demostrativo

Pronouns and Determiners **(PD)**

Los pronombres personales tienen formas diferentes según la función: yo, me, mi, conmigo.

Equivalen a formas de sustantivos, adjetivos e incluso a adverbios.

(i)	Person: (Persona)	1. First (primera)	2. Second (segunda)	3. Third (tercera)
(ii)	Gender: (género)	1. Masculine (masculino)	2. Feminine (femenino)	
(iii)	Number: (número)	1. Singular (singular)	2. Plural (plural)	

Por ejemplo la palabra "mí" quedaría como se muestra a continuación:

PD	1	0	1
mí	i. persona: primera	iii. género: ninguno	v. número: singular

Articles **(AT)**

Los artículos contienen atributos opcionales específicos del lenguaje y a continuación se despliegan algunas características. Son morfemas independientes.

(i)	Article-Type: (tipo de artículo)	1. Definite (definido)	2. Indefinite (indefinido)
(ii)	Gender: (género)	1. Masculine (masculino)	2. Feminine (femenino)
(iii)	Number: (número)	1. Singular (singular)	2. Plural (plural)

Por ejemplo la palabra "la":

PD	1	2	1
la	i. tipo: definido	iii. género: femenino	v. número: singular

***Adverbs* (AV)**

Los adverbios son invariables, es decir no admiten morfemas gramaticales, tan solo en algunos casos excepcionales el grado comparativo.

(i)	Degree: (grado)	1. Positive (positivo)	2. Comparative (comparativo)	3. Superlative (superlativo)

***Adpositions* (AP)**

Las adposiciones son el conjunto de las preposiciones y las posposiciones. En el idioma español sólo existen las preposiciones. Estas últimas no admiten morfemas, son invariables, son morfemas independientes. A continuación se menciona cada uno: a, ante, bajo, cabe, con, contra, de, desde, en, durante, entre, excepto, hacia, hasta, mediante, para, por, salvo, según, sin, so, sobre y tras.

(i)	Type: (tipo)	1. Preposition (preposición)

***Conjunctions* (C)**

Las conjunciones sirven para unir oraciones o ideas y a continuación se despliegan sus atributos.

(i)	Type:	1. Coordinating (coordinado)	2. Subordinating (subordinado)

Como una observación, se hace mención de que aquellos atributos que no aparecen completos en las categorías del etiquetado, es porque para efectos del etiquetado manual no eran necesarios, por ejemplo, en el etiquetado de los atributos del verbo no aparece el número dos, debido a que era innecesario.

7.2.6. Ejemplo de etiquetado pos en el corpus

A continuación se muestra un ejemplo del resultado del proceso manual del etiquetado pos en una noticia del corpus (conformado por 240) con base en el estándar *Eagles* para el idioma español.

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE corpus SYSTEM ".\lanzamientosU.dtd">
<corpus>
<noticia>
<link>http://diariored.com/blog/001356.php</link>
NOTICIA
Informática
<titulo>Así será Messenger 8 </titulo>
Guillem Alsina <fechaLanzamiento>22/03/2006</fechaLanzamiento>, 13:42:50
enviar a un amigo comentar (0) imprimir
La próxima versión del conocido programa de mensajería instantánea modifica ligeramente su nombre para integrarse con los servicios Live que Microsoft quiere potenciar, pasando a llamarse Windows Live Messenger.
<compania>Microsoft_N201</compania> ha_V31121 lanzado_V01602 la_AT121 <version>beta_N121</version> del_C2 futuro_N111 <producto>Messenger_N201</producto> sin_C2 la_AT121 habitual_AJ0210 fanfarria_N121 y_C1 publicidad_N121 que_C1 acostumbra_V31111 a_AP1 rodear_V00500 a_AP1 los_AT112 productos_N112 estrella_N121 de_AP1 la_AT121 compañía_N121 de_AP1 Redmond_N201 -y_C1 eso_AV0 que_C1 sin_C2 lugar_N111 a_AP1 dudas_N122, Messenger_N201 es_V31111 uno_AT211 de_AP1 sus_PD302 productos_N112 estrella_N121 en_AP1 Internet_N201-. Hasta ahora, el acceso a la beta estaba cerrado solamente a un grupo de betatesters seleccionados por invitación, pero ahora Microsoft la ha disponibilizado para todo el mundo.
Cambios gráficos
La estética de Messenger cambia -como en cualquier otra nueva versión de una aplicación de Microsoft, las mejoras en la interfaz gráfica son indispensables- para integrarse más con el aspecto que presentará el futuro Windows Vista y que se empieza a entrever en la gama de productos Live. Esta nueva faz que presenta el programa se ve desde el momento de arrancarlo.

Aparte de los cambios en el grafismo, la disposición de la interfaz no sufre grandes variaciones. Eso si, las formas más cuadradas predominan sobre las más redondeadas de versiones anteriores.
Otras mejoras que presenta la versión beta del próximo Messenger son:
Poder entablar conversaciones con nuestros contactos aunque tengamos el estado de desconectado, de forma que podemos pasar inadvertidos si no deseamos hablar con un contacto "indeseado" en aquel momento.
Disponibilidad de un "contestador automático" o buzón donde se dejarán mensajes a los contactos que no estén online en aquel momento, y que ellos podrán leer posteriormente sin tener que entrar en el buzón de correo (opción "Enviar un mensaje instantáneo para usuarios no conectados")
Carpetas compartidas que facilitan el intercambio de documentos y el trabajo en grupo entre los usuarios de Messenger. Será, sin lugar a dudas, una de las opciones más bien recibidas por parte de los usuarios corporativos.
Posibilidad de iniciar directamente una conferencia de audio mediante la opción "Llamar al PC de un contacto".
Opción de ver imágenes procedentes de la cámara web de un contacto directamente y desde la ventana principal de la aplicación.
¿Soporte para Mac OS X?
La tendencia en la compañía de Bill Gates es la de ir disminuyendo progresivamente el soporte a la plataforma de Apple, excepto en productos clave para la propia Microsoft como es la suite Office. Buen ejemplo de ello es el abandono primero del Internet Explorer para Mac y, más recientemente, del reproductor multimedia Windows Media Player.
Ante esta situación cabe preguntarse si Messenger va a ser el siguiente en sufrir los recortes, ya que la versión para Mac OS no dispone actualmente de todas las funcionalidades que tiene su homóloga para PCs con Windows, por lo que el futuro inmediato podría apuntar a la "congelación" del desarrollo de este software para pasar a abandonarlo definitivamente.
En cualquier caso, los usuarios de Mac (al igual que los de Linux u otros sistemas operativos) podrán continuar utilizando aplicaciones alternativas para hablar con sus conocidos a través de la Red.
Más información:
Windows Live Ideas
http://ideas.live.com
Descarga de la versión beta de Windows Live Messenger
http://ideas.live.com/signup.aspx?versionId=0eccd94b-eb48-497c-8e60-c6313f7ebb73
</noticia>
</corpus>

Como se puede observar en la noticia anterior, se encuentran varios elementos definidos en el dtd (la definición del dtd se muestra en el siguiente sub capítulo), como en este caso son:

- Corpus.- Indica y delimita todo el cuerpo de la noticia.
- Noticia.- Jerárquicamente se encuentra dentro de la etiqueta corpus.

- Link.- Se trata de la url de origen de la noticia, únicamente se copió del navegador y se pegó en el archivo txt.
- Título.- Se refiere al título del lanzamiento de la noticia.
- FechaLanzamiento.- Puede ser cualquier formato de fecha, como la fecha de publicación de la noticia y no necesariamente la fecha de lanzamiento, aunque pueden haber varias fechas dentro de la noticia.
- Compañía.- Empresa que desarrolla o lanza el producto.
- Producto.- Nombre del producto de *software* que se lanzó.
- Versión.- Versión del producto, que puede ser un numeral, una combinación de números y letras o una palabra (beta, prueba, etc.).

Las dos primeras líneas corresponden a:
<?xml version="1.0" encoding="iso-8859-1"?>.- Versión del editor de xml[50] que fue utilizado.
<!DOCTYPE corpus SYSTEM ".\lanzamientosU.dtd">.- Ubicación del DTD al que se hace referencia para el etiquetado.

Bajo un simple análisis es notorio que esta última forma de etiquetado semántico es un tanto diferente al primer bosquejo del etiquetado -véase primer bosquejo en cap. 7.2.3-, por las problemáticas y necesidades que fueron surgiendo al ir coleccionando el corpus.

Después del etiquetado semántico se continúa con las partes de la oración. Se puede observar que en la noticia anterior hay un fragmento de texto como el siguiente:

…….. ha_V31121 lanzado_V01602 la_AT121 AT121
<version>beta_N121</version> del_C2 futuro_N111
<producto>Messenger_N201</producto> ……

corresponde al estándar de *Eagles* referente al verbo "haber" (ha: verbo, 3ra persona, singular, modo indicativo, tiempo imperfecto y voz activa), "lanzar" (lanzado: verbo, sin persona, singular, modo participio, sin tiempo y en voz activa) y al artículo determinado "la" (artículo definido, femenino, singular), etc.

También se pueden definir varias palabras clave dentro del párrafo etiquetado, entre las que se encuentran "beta", "disponibilizado" y "lanzado"[51]. Pero hay otras a lo largo de la noticia como "próxima" y

[50] En este caso se utilizó un editor para xml llamado XML *Writer*.

[51] Se debe recordar que todas las palabras se buscan en el lexicón con base en su lema.

"versión" que también son *trigger words*. Así, en el *gazetteer* de empresas se buscaría el *token* "*Microsoft*" para definir que es una empresa (entity named recognition).

Para efectos del etiquetado manual, cabe recalcar que solamente se etiquetó el enunciado gráfico (comprendido entre una letra mayúscula hasta un punto) que indica que se trataba de un nuevo *software*, dejando momentáneamente a un lado el resto de la información (que en dado caso pudiera ser información irrelevante).

La mente humana puede identificar un enunciado gráfico que indique un nuevo lanzamiento de *software* en base a criterios como:

- Contiene el mayor número de *trigger words.*
- Contiene todos o casi todos los elementos del DTD que son necesarios para el procesamiento de los *tokens*, como pueden ser la compañía, producto, versión y/o fecha de lanzamiento.
- Generalmente el título de la noticia es donde mayoritariamente aparecen todos estos elementos, pero el ejemplo anterior no fue el caso.

7.2.7. Desarrollo de la investigación y proceso de análisis

El proceso de análisis del "SEINS" se divide en tres partes principales; la primera parte es llamada "proceso de investigación", la segunda se basa en el análisis de los datos de forma manual y la tercera consta de la utilización de programas de *software*, algunos ya desarrollados como java y otros adaptados (XML, lematizadores y lexicones) a las necesidades del proceso de extracción de información de noticias sobre el lanzamiento de productos nuevos.

Se decidió que se iba a etiquetar el texto tanto con XML como con *Eagles*, con XML únicamente se iban a marcar cuestiones fundamentales como el título de la noticia, la fecha, versión del *software*, el autor de la publicación y la ubicación en internet básicamente, y las etiquetas *Eagles* se marcarían con un guión bajo sucediendo la palabra a etiquetar. Esto hace innecesario un procesamiento de etiquetas XML riguroso como al principio se había pensado -véase primer bosquejo en cap. 7.2.3-.

En el proceso de investigación se obtuvo la información para fundamentar la misma, así como las bases teóricas para llevar a cabo el proceso de extracción de información[52].

Para el análisis de los datos de forma manual se partió de la recolección del corpus, es decir, la obtención de noticias en línea (en la *web*) sobre lanzamientos de nuevos productos o actualizaciones de los mismos dentro de la www con buscadores como *google*, *yahoo* y *altavista* entre otros.

En un principio el corpus textual estuvo conformado por 120 noticias de distintos tamaños, redacción, estructura y formatos. Pero se empezó a hacer notorio que dicha cantidad no era suficiente, así que se decidió ampliarla a 240 noticias, lo que ayudaría a obtener métricas más precisas al momento de mostrar los resultados y permitiría un mejor análisis para construir después las expresiones regulares.

Lo que procedió fue el análisis y etiquetado manual, el cual se dividía básicamente en dos partes: el etiquetado POS (morfológico) y el etiquetado semántico.

El etiquetado semántico consiste en dividir mediante etiquetas xml que se establecieron previamente, las partes de la oración de las noticias de acuerdo a la estructura general que se muestra a continuación. Dicha estructura se estableció mediante un análisis minucioso de las partes que componen a las noticias. Estas noticias tratan sobre eventos de lanzamientos de nuevos productos de *software* y a continuación se muestra un esquema de la jerarquía de las etiquetas propuestas.

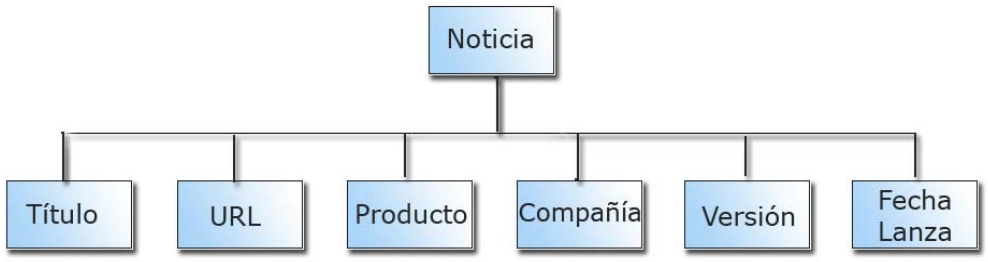


Figura 7.2.- Elementos que se definieron en el DTD.

52 Material bibliográfico, internet, publicaciones de científicos y expertos en la materia, etc.

Posteriormente se hizo el etiquetado morfológico, que consiste en identificar cada una de las palabras de la oración según su estructura gramatical, a esta forma de etiquetar también se le llama etiquetado POS (*Part Of Speech*), -véase sección de *Eagles* en cap. 7.1.4-.

En esta etapa se tokeniza cada palabra, es decir, se divide cada palabra y se marca de acuerdo a su definición gramatical de la lengua española, si es un verbo, sustantivo, artículo, preposición, conjunción, adjetivo, adverbio, etc., mediante las etiquetas de *Eagles* adaptadas para el español. De esta forma se obtiene el corpus etiquetado.

Mediante el análisis del corpus etiquetado se obtienen aportaciones importantes que a continuación se mencionan.

Aportaciones posibles del corpus para un futuro:

- El corpus obtenido puede servir para análisis de investigaciones posteriores por parte de alumnos de la facultad de contaduría y administración en la carrera de Licenciatura en Informática, o alumnos de algunas ingenierías.
- Entrenamiento de aprendizaje de computadoras para extracción de información.
- Para análisis estadísticos del contenido lingüístico de dicho corpus.
- Generar etiquetadores automáticos mediante los patrones encontrados durante el etiquetado y escudriñamiento del corpus.
- Una consultoría de *software* pudiera querer conocer cuál es su principal competencia y mediante este corpus lo puede lograr. Aunque sólo se limitaría a datos y fechas anteriores al año 2006.
- El corpus puede servir como un archivo de almacenamiento de sitios *web* dedicados a la publicación de noticias de *software*.

7.2.7.1. Uso del corpus

El etiquetado manual es útil para el análisis de la estructura lingüística.

- ¿Qué verbos se utilizan en este corpus?
- En el caso de las compañías a las que se hace referencia, ¿Cómo se nombran?
- ¿Cómo es la estructura del párrafo?, ¿toda la información que se va a extraer se puede encontrar en un sólo párrafo o en varios?.
- Observar que existen mayúsculas al principio del párrafo, en la primera palabra y después de cualquier punto, así como en la mayoría de los casos para referirse a nombres propios.

- ¿Cuáles son las *trigger words* que se podrían definir?

Mediante el análisis, se pueden generar expresiones regulares (indicadores) para realizar búsquedas por patrones, es decir, generar indicadores o elementos clave para cada elemento a encontrar. Por ejemplo, "*Microsoft* lanza *SQL Server*" (Nombre compañía + verbo + Nombre producto).

Obtener medidas de precisión y recuperación las cuales son medidas de evaluación (*Precision and Recall*).

De esta manera se pretende obtener el algoritmo que proporcione la forma de extraer la información requerida y por lo tanto llegar a la solución.

La tercera etapa consiste en establecer una metodología que permita extraer la información requerida mediante un proceso automático.

Una vez que se ha obtenido el corpus, se selecciona una noticia y se procede con los siguientes pasos:

- Tokenizar: se encarga de la división del corpus palabra por palabra y de esta manera obtener el lema de cada una, es decir, obtener la raíz de la palabra.

- Etiquetado POS: en este paso se llevará a cabo el etiquetado morfológico o etiquetado de *Part of speech* (partes de la oración) mediante una herramienta de etiquetado automático.

- Etiquetado semántico: se utiliza un lexicón y una ontología (en este caso se trata de un *gazetteer* de compañías con las empresas desarrolladoras más importantes del mundo), es decir, un mapa conceptual del tema a tratar. Aquí ya se debe obtener el algoritmo de solución.

- Llenado de los *templates*: en esta etapa se llenan los *templates* (plantillas) con la información que se debe obtener o extraer, es decir, la información final que se va a mostrar al usuario.

Lo importante es recopilar material acerca del tema a investigar, en otras palabras, formar el corpus que sirva de modelo durante la investigación. La importancia de esto radica en que la única forma de conocer profundamente el objeto de estudio es analizándolo detenidamente y buscando dentro de él patrones lingüísticos, es por ello que el corpus de entrenamiento se debe etiquetar completamente «a

mano» y obtener estadísticas con base en dicha recopilación de datos. Esto permite que durante la recolección y el etiquetado, uno mismo se pueda dar cuenta de los patrones de la lengua que caracterizan a lo que se está estudiando, es así como resalta el hecho de que hay palabras que denotan ciertas acciones a la hora de publicar un lanzamiento de un nuevo programa, *software* o aplicación; lo mismo ocurre con la posición de la información relevante dentro de la noticia y cómo las circunstancias más importantes siempre o casi siempre se encuentran redactadas al principio de la oración. Entonces sólo un análisis manual y minucioso podrá sacar a la luz las estructuras que se están buscando dentro de los textos donde se usará el programa en la práctica.

También es necesario entender que sobre ese corpus se desarrollarán algunos programas de prueba que no tienen la finalidad siquiera de pasar a formar parte del desarrollo final, sino que únicamente se utilizan para analizar los datos de muestra e inspirar el producto final.

Por lo tanto el etiquetado del texto es muy importante porque aquí es posible conocer los patrones.

7.2.7.2. Expresiones regulares para extracción de enunciados

El léxico *PAROLE* del español

Parole es un proyecto financiado por la UE bajo el IV Programa Marco para la creación y desarrollo de recursos léxicos (corpus y diccionarios electrónicos) a gran escala para catorce lenguas europeas con una estructura de codificación común en SGML (*Standard Generalized Markup Language, Internacional, Standard* 8879).

Los léxicos *Parole* siguen las recomendaciones del grupo *Eagles*. El modelo *Parole* es, por lo tanto, un modelo descriptivo, flexivo (permite acomodar diferentes niveles de granularidad en sus descripciones) y neutral en cuanto a teorías lingüísticas se refiere.

El léxico *Parole* incluye cuatro niveles de descripción: (i) nivel morfológico, (ii) nivel sintáctico, (iii) nivel semántico, y (iv) nivel relacional que permiten relacionar la información codificada.

Las expresiones regulares que se van a utilizar para efectos de programación se basan en dicha categoría al momento de etiquetarse mediante el etiquetador automático.

Cabe hacer mención que *Parole* es una categoría derivada de *Eagles* para el idioma español. Por lo tanto se utilizó el estándar de *Eagles* para el etiquetado manual, pero el etiquetador automático y el lexicón que se van a utilizar toman la categoría *Parole* aunque básicamente ambos utilizan la misma estructura sintáctica -véase atributos de *Eagles* en cap. 7.1.5-, es decir, verbos, sustantivos, adjetivos, etc. Lo único que cambia es la definición de la etiqueta, por ejemplo, para los artículos *Eagles* se utiliza el afijo "AT", mientras que *Parole* utiliza "T".

Como nota adicional se menciona que las etiquetas Xn (donde "n" es un número consecutivo del cero al seis) hacen referencia a palabras no encontradas en el lexicón. Dichas etiquetas se explicarán a mayor detalle más adelante dentro del apartado "Etiquetas adicionales".

Patrones para identificar el producto

Una expresión regular o patrón que se definió para identificar un producto de *software* (a través de su nombre) en un principio fue la que se muestra a continuación:

DP3C.00 | TD(M|F)S0 | DI3(M|F)S00 nuev(a|o) .* X1|X3 ha lanzado | creado | liberado | presentado | anunciado | publicado .* X1|X3

Lo anterior indica que un patrón muy confiable para encontrar un nuevo producto es:

Un Pronombre personal ó un Artículo ó un Determinante + "nuevo" ó "nueva" + conjunto de palabras + Nombre propio + "ha" + lanzado ó creado ó liberado ó presentado ó anunciado ó publicado + conjunto de palabras + Nombre propio.

Con el patrón anterior, un texto como "Una nueva liberación de *software* se dio a conocer por la compañía *Microsoft,* que ha lanzado ahora *Messenger Plus*" podría hacer concordancia.

Patrones para identificar compañías

la compañía .* X1|X3 .* ha lanzar_V0I(S|F|P)3S0| crear_V0I(S|F|P)3S0|liberar_V0I(S|F|P)3S0|presentar_V0I(S|F|P)3S0|anunciar_V0I(S|F|P)3S0|publicar_V0I(S|F|P)3S0|colocar_V0I(S|F|P)3S0.

El patrón consiste en:
"la" + "compañía" + conjunto de palabras + Nombre propio + conjunto de palabras + "ha" + lanzado ó creado ó liberado ó presentado ó anunciado ó publicado ó colocado.

Como ejemplo para esta expresión regular se podría citar:
"La compañía de *Redmond* en EE.UU., Microsoft, ha liberado *Windows Vista*".

Patrones para identificar versiones

- X1 + X4 : nombre propio + nombre propio
- versión + X4 : "versión" + nombre propio
- versión + (b|B)eta : "versión" + "beta" ó "Beta"

Como ejemplos de versiones se pueden citar:

- Messenger Plus.
- versión Plus.
- ha salido la nueva versión beta de Messenger plus.

Expresiones regulares en java:

Caracter	Significado
^	Principio de la cadena
$	Final de la cadena
.	Cualquier carácter excepto salto de línea
*	Operador de repetición 0 ó más veces
+	Operador de repetición 1 ó más veces
?	Operador alternativo: una vez ó ninguna
|	Alternativa
( )	Agrupar expresiones
[ ]	Conjunto de caracteres
{ }	Modificador de repetición
\	Permite presentar un metacaracter como un carácter ordinario

Los caracteres antes citados se utilizan en java, por lo tanto fueron tomadas para su representación en la aplicación a desarrollar.

Etiquetas adicionales

Así mismo se definieron algunas etiquetas denominadas "X", debido a que se asignan a palabras que no fueron encontradas en el lexicón en español, como son entre otras, *software*, *hardware*, nombres propios de empresas, compañías, productos, etc.

La etiqueta "X1" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón[53] utilizado y que además comiencen con una letra mayúscula y continúe con el resto de las letras en minúsculas. Con esta etiqueta se pretenden definir nombres propios de compañías, productos, personas, etc.

"X2" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón y que estén formadas solamente por minúsculas.

"X3" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón y que estén formadas solamente por mayúsculas.

"X4" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón v que estén formadas por combinaciones de mayúsculas, minúsculas y numerales. Por ejemplo, productos con nombres poco comunes.

"X5" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón y que estén formadas por la combinación de letras mayúsculas y minúsculas.

"X6" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón y que estén formadas solamente por números, por ejemplo, las fechas o versiones de los productos.

"X0" se aplicará a todas aquellas palabras que no estén contenidas dentro del lexicón y que tampoco se engloben dentro de alguna de las anteriores.

7.3. Diseño

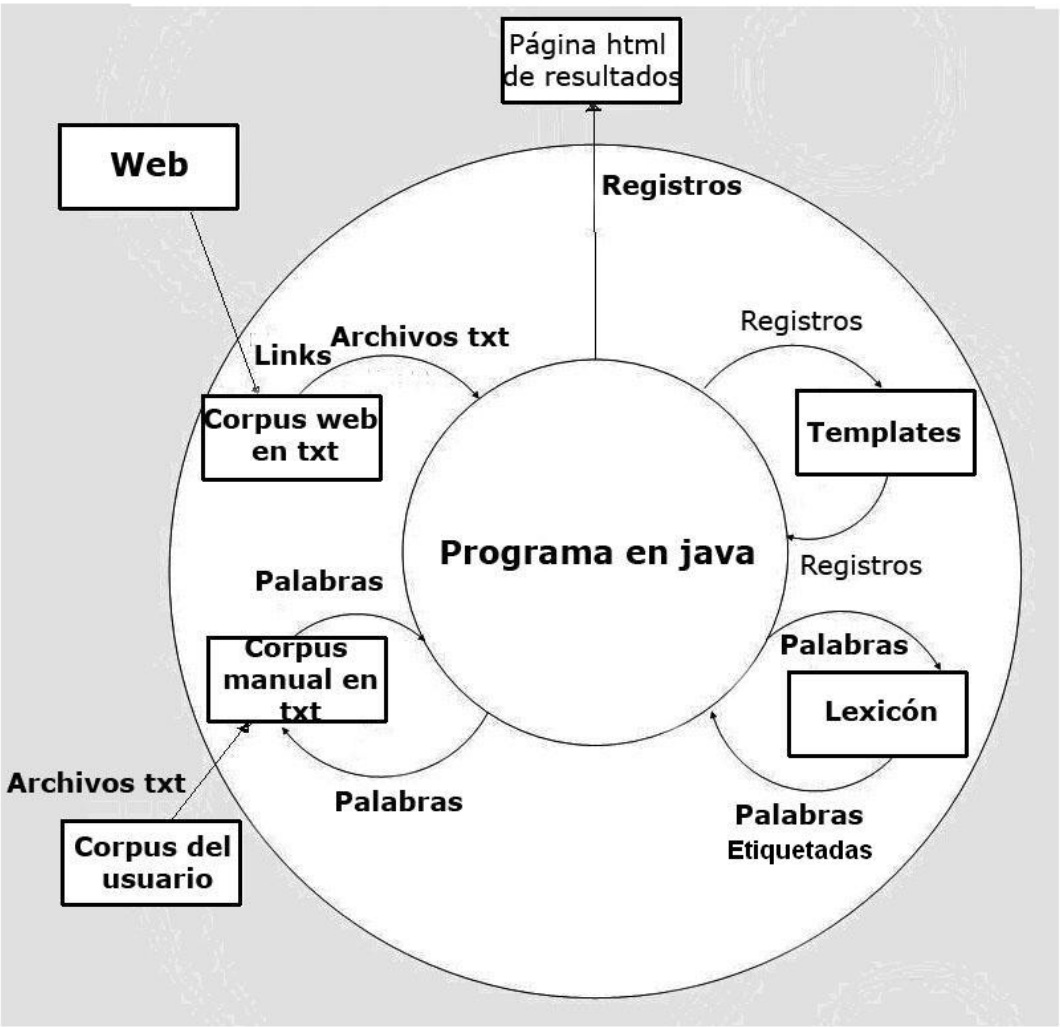
El DFD nivel 0 que se ha definido para desarrollar la aplicación es el siguiente:

[53] Dicho lexicón contiene todas las palabras del español, incluyendo nombres propios de personas, ciudades y países. Aunque hay que mencionar que el lexicón que se utilizó para el desarrollo de la aplicación, no contiene algunas palabras como *software* o ninguno de los anglicismos.

El DFD nivel 0 que se acaba de mostrar consiste básicamente de dos entradas: noticias desde el internet y noticias ingresadas manualmente por el usuario a través de archivos con formato txt.

Las entradas desde el internet consisten en direcciones url que el sistema va a analizar en la *web*, dichas direcciones ya se han definido como confiables donde generalmente se dedican a publicar noticias de este tipo.

Así mismo también se definió el DFD nivel 1, que se muestra a continuación:



En el DFD nivel 1 se muestra de una rápida manera los módulos que componen la aplicación.

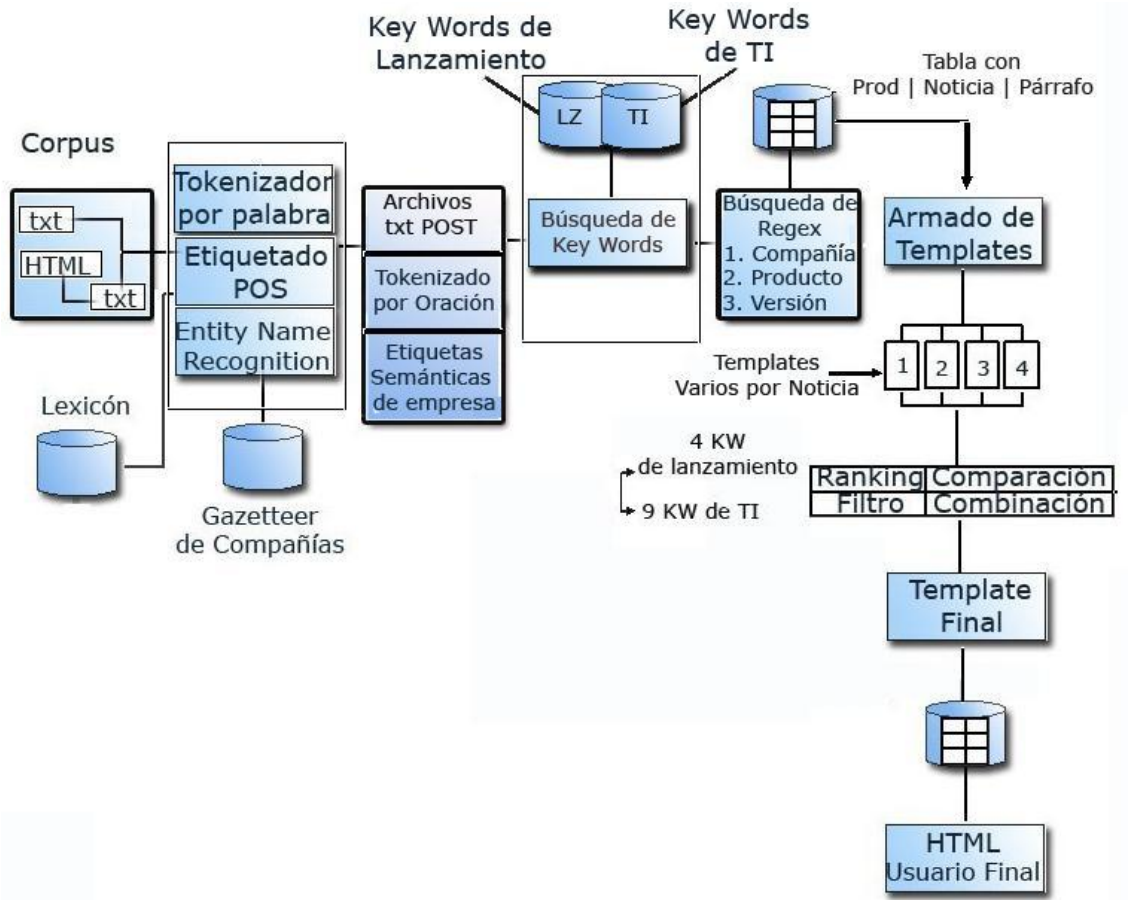
Las direcciones url de las entradas desde la *web* se guardan en archivos html y se convierten en archivos txt cuando se etiquetan, uno por noticia. En caso de que el usuario ingrese los archivos txt manualmente, el sistema se ahorra el proceso de almacenar las url, debido a que ya están convertidas en archivos de bloc de notas (txt).

En seguida el programa trabaja internamente, haciendo consultas a un lexicón del alfabeto español y a un *gazetteer*[54] de compañías de *software* para el etiquetado pos.

[54] Se denominó *gazetteer* a un listado de compañías de *software*.

Después de haber hecho los procesos anteriores, genera unas plantillas (*templates*) que más tarde saldrán a la vista del usuario a través de una interfaz sencilla en html.

Por último, también se definió un flujograma donde se describe con más detalle lo que el programa en java realiza, como se mostró en el DFD 1.



La explicación del flujograma es la siguiente:

1.-Se establecen los datos que servirán de entrada, como ya se mencionó pueden ser a través del internet (que sería lo más común) o manualmente por medio de archivos de texto.

Los datos de entrada se establecerán en base a sitios *web* confiables, es decir, aquellos sitios que se dediquen a publicar lanzamientos de *software*, aunque también pueden abarcar noticias de cualquier otro tipo, pero predominando todo lo referente a las Tecnologías de Información.

2.-Mediante las entradas de los datos se forma el corpus, cada una de las noticias que se lee desde internet se transforma, por así decirlo, en un archivo de bloc de notas.

El corpus de noticias no tiene un número definido de elementos, es variable de acuerdo a la cantidad de url's de noticias que en ese momento haya en la página que se dedica a publicar nuevos lanzamientos de *software*.

3.-Posteriormente se hace la limpieza de las etiquetas propias del lenguaje html, es decir, la aplicación SEINS lee noticia por noticia, eliminando todas las etiquetas html (<title>, <a>, <body>, <p>, etc.) para dejar únicamente el texto de la noticia sin algún formato.

Después se limpiaron todos los signos de puntuación como son "¿ ?", "! ¡", comas, punto y coma, etc., exceptuando todos los "puntos finales" y los "punto y a parte" de cada enunciado gráfico. Como se mencionó en el capítulo de "Método de desambiguación del límite de las frases" -véase cap. 6.4-, se dio un tratamiento especial a los puntos decimales en base a números próximos que tenga, en caso de no tenerlos, se considera como final del enunciado.

4.-Ahora se continúa con el proceso de tokenización, que en un primer caso se realizará por párrafos y después por palabras, esto lo hará el programa en java.

5.-El etiquetado se hace en base al código fuente del programa de java y utiliza un lexicón para el español y un *gazetteer* más de empresas de *software*. El primero contiene todas las palabras del idioma español y el segundo una gran lista de compañías de *software*.

Después de tokenizarse la noticia, cada palabra (*token*) se busca dentro del lexicón, con cerca de 1,010,000 elementos definidos en todas sus posibles conjugaciones (en el caso de verbos) o lemas, en caso de que se encuentre contenida, se le asigna su etiqueta correspondiente en base a su morfología (verbo, sustantivo, adjetivo, preposición, etc.).

Si en esta primer búsqueda no se encontró alguna palabra contenida en el lexicón, se busca en otro lexicón alterno derivado del primero con únicamente las palabras acentuadas ortográficamente (todas aquellas que llevan acento) pero que en dicho lexicón alterno se omitieron sus acentos de manera intencional, debido sobretodo a que un error ortográfico muy común en la redacción de las noticias, es que los redactores suelen olvidar la acentuación de las palabras. En caso de que

ahora sí se encuentre la palabra contenida en el lexicón alterno de palabras sin acentos, se etiqueta conforme le corresponde.

Alternadamente a la búsqueda de los dos lexicones, se busca cada palabra en el *gazetteer* de compañías de *software*, dando preferencia a este último etiquetado ya que hay palabras como Adobe que son empresas pero que también se incluyen en el lexicón. A este proceso del *gazetteer* se le llama "Reconocimiento de Entidades Nombradas" y corresponde a un etiquetado semántico ya que de cierta manera, se asocia una entidad del mundo (empresas de *software*) a una palabra y esta misma se reconoce como un ente (empresa).

El *gazetteer* de compañías de *software* también se estableció para reducir de cierta manera la cantidad de elementos que se etiquetaban con "X", ya que así se delimitaba mucho más la cantidad de palabras y los nombres propios que pudieran aparecer, estadísticamente tendrían mayor probabilidad de ser productos o compañías ya que los países y muchos nombres están contenidos en el lexicón para el español, las empresas tendrían su propio *gazetteer*, dejando una mayor posibilidad a los productos. Como ejemplo se puede mencionar que la palabra "Adobe" o "Pegaso", contenidos en el lexicón del idioma español, aparecían etiquetados como sustantivos[55], siendo que en una redacción de noticias de *software* era un 99% de posibilidades de que fuera nombre de compañías.

El *gazetteer* de empresas tiene alrededor de 300 registros, faltando de normalizar alrededor de 1000 más con las que ya se cuentan almacenadas sin formato y que se encuentran en un archivo de excel.

[55] Según la RAE: Adobe.-Masa de barro mezclado a veces con paja, moldeada en forma de ladrillo y secada al aire, que se emplea en la construcción de paredes o muros. Pegaso es un caballo alado en la mitología griega.

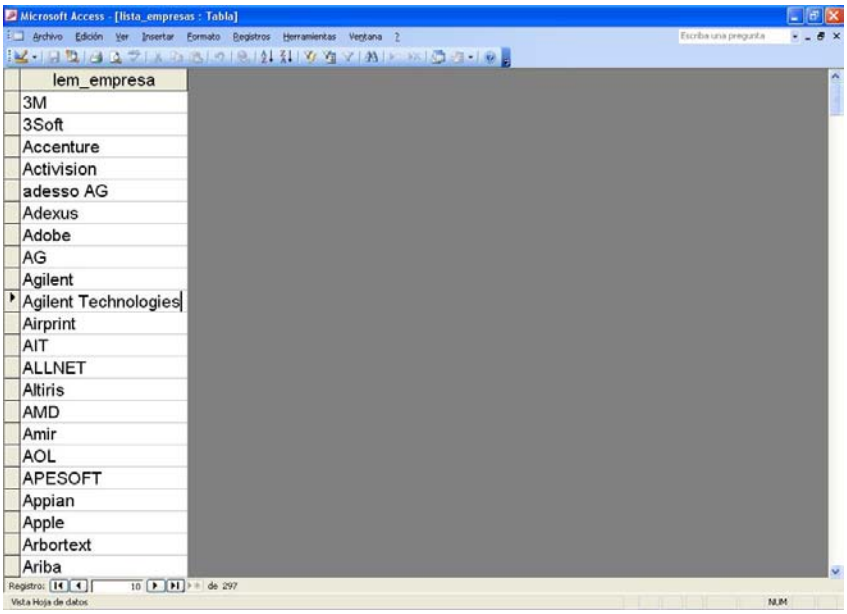


Figura 7.3.- Vista de la tabla donde se almacenan las compañías de *software*.

Si todavía no es encontrada la palabra en alguno de los lexicones, se etiqueta con el elemento "X" correspondiente definido en el capítulo "Expresiones Regulares para extracción de enunciados" -véase cap. 7.1.7.2- en base a expresiones regulares que se definieron en el programa de java.

Después de haber definido las etiquetas de las palabras, los archivos de salida (noticias tokenizadas y etiquetadas) se almacenan en otros archivos (un archivo txt por noticia) con el mismo nombre pero solamente añadiéndole el texto "FaseDos" y sustituyendo los puntos por guiones bajo en el mismo nombre, con lo que se hace el proceso inverso a la tokenización. Por ejemplo, se procesa el archivo "Sql Server.txt" y sale como resultado un archivo "Sql Server-txt-FaseDos".

Como resultado de esto se tienen los archivos etiquetados y tokenizados por oración y las etiquetas de las empresas (etiquetado semántico) dentro de los mismos archivos.

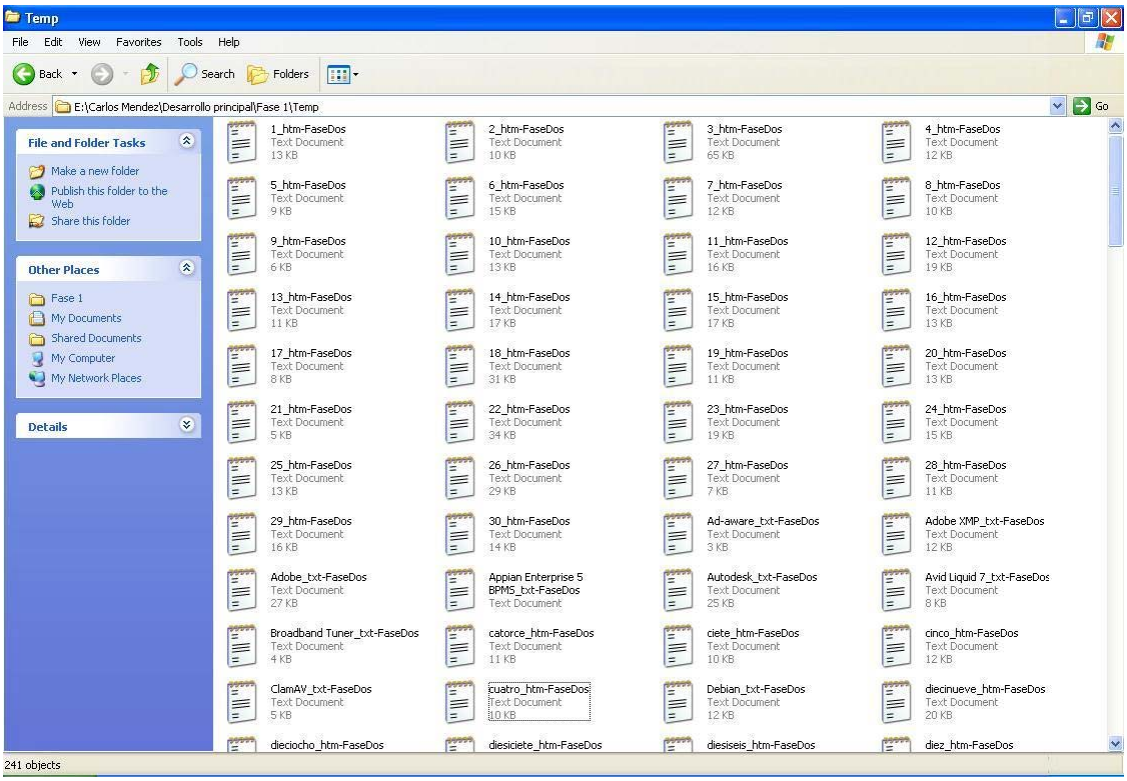


Figura 7.4.- Vista del corpus que se tiene una vez que se aplica el proceso de tokenización y etiquetado.

6.-Se continúa con la búsqueda y comparación de las expresiones regulares dentro del corpus generado por el proceso de etiquetado.

Dentro de la programación en java se estableció una serie de palabras clave que van a funcionar como *trigger words*. Estas palabras clave son:

Palabras clave de informática:

- software
- ordenador
- computadora
- hardware
- digital
- tecnología
- informática
- PC
- XML
- CPU
- documento
- red

Palabras clave de lanzamiento:

- lanzar
- crear
- liberar
- presentar
- anunciar
- beta
- Beta
- publicar
- colocar
- crear
- programa
- lanzamiento
- nuevo
- nueva
- versión
- disponible
- disponibilidad
- solución
- producto

Expresiones regulares usadas

Las expresiones regulares se dividieron en dos apartados, unas para identificar que se trata de una noticia relacionada con las Tecnologías de Información (en base a las palabras clave de tecnologías de información) y la otra para identificar lanzamientos de *software* (en base a las palabras clave de lanzamientos).

Esto se hizo para discriminar noticias de lanzamientos de productos que no tengan relación alguna con el *software*.

Entonces se vuelve a hacer una tokenización pero ahora por palabra incluyendo su etiqueta y lema (ya realizados en la etapa anterior), cada palabra clave del programa se busca y compara con cada una de las palabras contenidas en el archivo generado, la búsqueda se hace en base al lema de la palabra clave.

El método "indexOf()" de java devuelve una posición dentro de una cadena de un carácter específico. Dicho método busca empezando desde el principio de la cadena y es la que va a permitir trabajar con expresiones regulares.

Las otras funciones importantes en la programación de la aplicación son "Match()" y "stringtokenizer", la primera recibe la expresión regular como argumento y su función es identificar si la palabra del texto de la noticia coincide con el parámetro, por ejemplo, si empieza con

mayúsculas, minúsculas, etc. Mientras que la segunda permite dividir un *string* en *substrings* o *tokens*, en base a otro *string* (normalmente un caracter) separador entre ellos denominado delimitador.

Por mencionar algunos ejemplos de la búsqueda, cuando se tratan de verbos, se analiza por su lema, mientras que cuando son adjetivos o adverbios se identifica la palabra completa.

Las expresiones regulares son comparadas con algún patrón que coincida dentro del contenido de la noticia, cuando este patrón coincide exitosamente, el resultado se almacena en una tabla específica de la base de datos dependiendo de la función de la *regex*, por ejemplo, los resultados de las expresiones regulares para encontrar compañías son almacenadas en una tabla llamada "empresa".

7.-En la etapa de almacenamiento de los resultados, los resultados de las *regex* exitosas son enviados a las tablas de la base de datos.

Entre las tablas que se encuentran dentro de la base de datos están:
"empresa".-Recibe el nombre de todas las compañías que las expresiones regulares lograron obtener.
"producto".-Contiene el nombre de los productos que las expresiones regulares lograron rescatar.
"version".-Las versiones que se pudieron obtener.
"palabras_c".-El número de palabras clave de lanzamientos de *software* que había en cada noticia, así como el número de párrafo donde se encontraron.
"palabras_ti".- El número de palabras clave relacionadas con la Informática.
"paginas".- Se almacenan los sitios donde se hizo la búsqueda cuando el sistema es ejecutado para armar el corpus desde internet.
"corpus".-Almacena cada uno de los títulos de los archivos que conforman el corpus.
"resultados".-Almacena las salidas de las empresas que fueron encontradas, las versiones y los productos. Esta última tabla es la que va a formar las plantillas de salida.

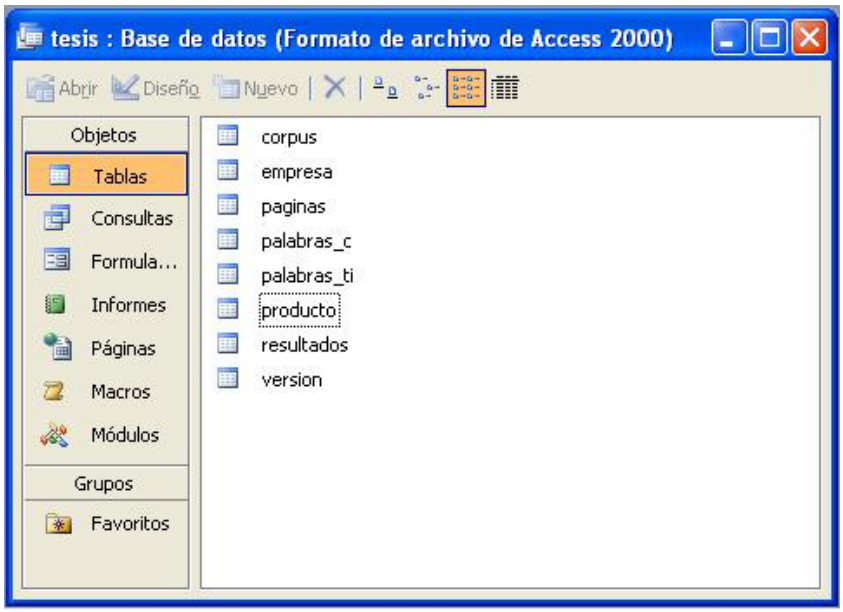


Figura.-7.5.- Tablas de la base de datos en las que se almacenan los registros de entrada y salida del sistema SEINS.

Todos los registros se almacenan por párrafos, es decir, cantidad de palabras o registros almacenados por párrafo individual.

8.-Después del almacenamiento de las palabras en las tablas, se tiene que hacer un ordenamiento (filtrado) de los registros.

Este filtrado se hace mediante un conteo de las palabras clave, ya que mientras más palabras clave haya contenidas en la noticia, más confiable se puede catalogar, aunado todo esto a la identificación de la compañía, producto y versión en el mismo texto.

Si la noticia contiene menos de 4 palabras clave de lanzamiento o menos de 9 palabras de tecnología no se considera confiable y no se toma en cuenta.

Después se continúa con el armado de *templates* por párrafo, esto ocasiona que cada noticia genere varias plantillas para después hacer un nuevo filtrado de estos de acuerdo a la confiabilidad de cada *template*, únicamente es considerado el que alcance el mayor puntaje de acuerdo a un *ranking* por palabras, cantidad de veces que se repiten, las compañías, productos y versiones, así como por su ubicación en el texto (casi siempre es el primer párrafo) que se hace a cada párrafo.

El *template* con mayor ranking es el que se almacena en la tabla de resultados.

Figura 7.6.- Tabla de almacenamiento de la salida lista para ser filtrada y ordenada para después mostrarse en html.

La figura anterior muestra una prueba realizada a una sola noticia. El primer campo resalta el nombre del archivo (unoSi.htm). El segundo campo indica que fue encontrada una sola palabra clave en ese *template* (párrafo). Después se indica que en toda la noticia hubo 5 palabras clave relacionadas a lanzamientos de *software* y 14 relacionadas a tecnologías de información. El quinto campo muestra el nombre de las empresas de acuerdo a los resultados de las *regex*. El siguiente campo se refiere al producto. El séptimo indica la versión que fue recuperada y continúa con el número del párrafo dentro de la noticia (en este caso es el número 37). Por último se muestra una casilla de verificación que se activa conforme se va mostrando el resultado en la interfaz de html.

Los resultados de los párrafos (en este caso enunciados gráficos) se van a mostrar en tablas html.

7.4. Desarrollo

Se decidió llevar a cabo la programación en el lenguaje Java, yendo de lo particular a lo general, debido a las ventajas que este programa representa sobre otros lenguajes, tales como:

- Portabilidad: es decir que el mismo paquete binario que se genera al compilar el código se puede transportar de un sistema operativo a otro o de una arquitectura de computadora a otra.
- Múltiples funciones: Java cuenta ya con varias librerías de funciones desarrolladas llamadas API´s.
- En el campo laboral agrega un mayor valor extra saber programar en Java que en otros posibles lenguajes que se pudieron utilizar en este desarrollo (como *Python*).

- Utilizar el modelo orientado a objetos nativo de Java sirve para seccionar el trabajo de una manera sencilla (dividir el problema).
- Cuenta de manera predeterminada con varias clases que permite trabajar con expresiones regulares.

Las etapas del desarrollo del sistema se explicarán a continuación.

7.4.1. Módulo 1

La página "index.html" del sitio *web* donde se publican las noticias es analizada por el sistema. Se omite absolutamente todo el contenido con la única opción de las palabras dentro de las etiquetas que marcan los hipervínculos (<a>) y es en base a estos hipervínculos como se van a tomar esas noticias más específicas para almacenarlas en el corpus de entrada del sistema, tomando como referencia las rutas relativas al momento del almacenamiento, como se muestra en la figura 7.7.

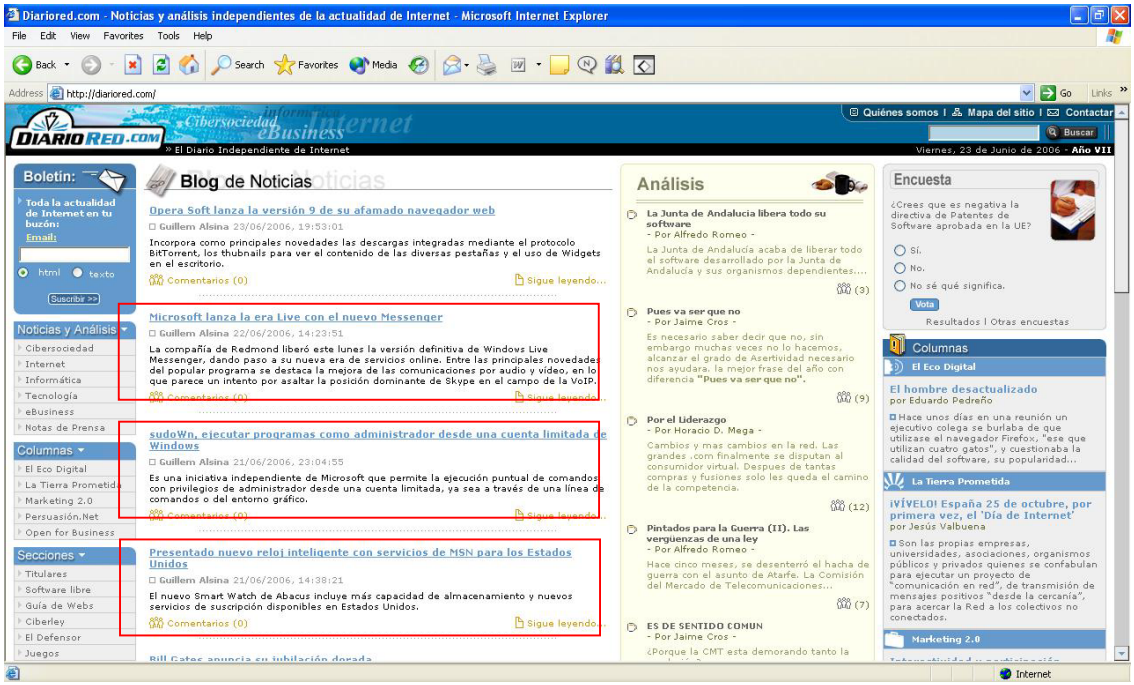


Figura 7.7.- Página de inicio de un portal dedicado a la publicación de nuevas noticias de *software*.

La figura 7.8 muestra un sitio *web* que publica el lanzamiento del producto de *Microsoft Messenger* en su versión 8.

Esta noticia fue tomada de la dirección http://diariored.com/blog/001356.php, derivada del portal mostrado en la figura anterior y es el link que va a generar un archivo con el texto de la noticia de ese link.

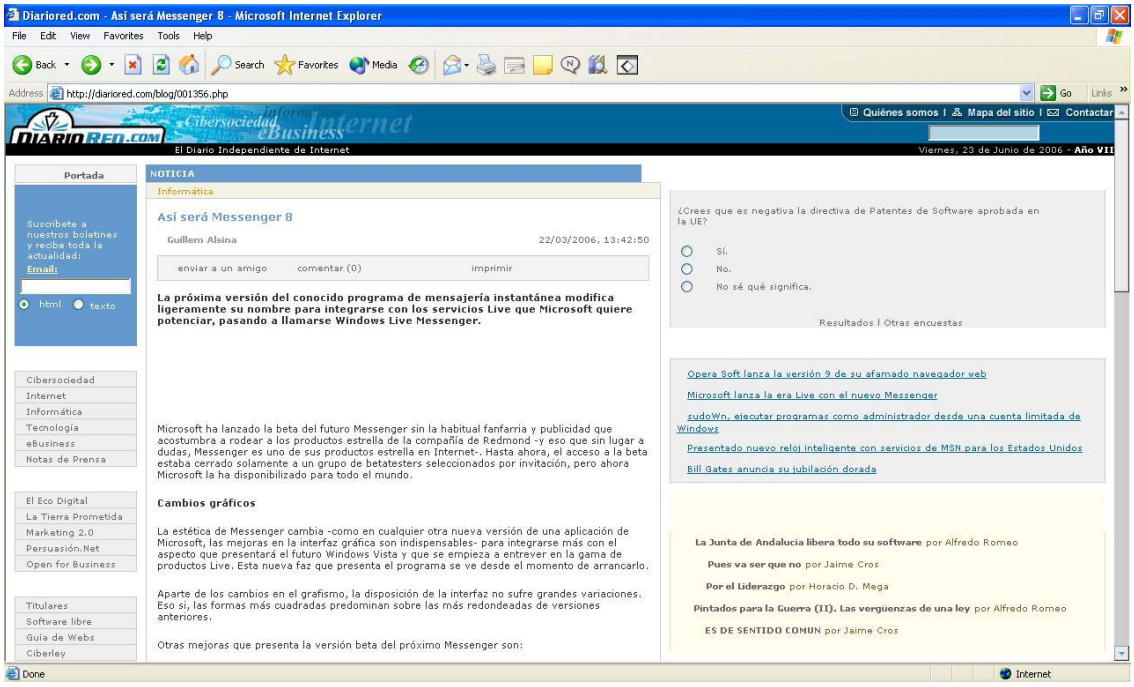


Figura 7.8.- Noticia en internet sobre el lanzamiento de *Messenger* 8.

El nombre del archivo de entrada se guarda con el nombre de la *url* de la noticia, pero por el formato de nombres de archivo que se ocupan en los txt, se omitieron los símbolos "?", "/", ".", "&", etc.

Durante la **recolección del corpus** y **limpieza del etiquetado html**, se tuvo un problema con la codificación de las páginas *web*. Esto hace referencia a la codificación mediante ASCII o Unicode, dependiendo de los navegadores como *Mozilla, Internet Explorer, Firefox, etc.,* que utilizan uno u otro tipo. El problema se debe a que java trabaja sobre archivos con codificación Unicode[56] y muchos caracteres especiales como la "ñ" o los acentos se representan de distinta manera en ASCII y en unicode. Pero este problema no se presentará cuando el corpus sea armado desde internet.

56 El lenguaje Java utiliza el conjunto de caracteres Unicode, que incluye no solamente el conjunto ASCII sino también carateres específicos de la mayoría de los alfabetos. Así, es posible declarar una variable que contenga la letra ñ:

int año=1999

Al momento de la limpieza de etiquetas html en los archivos de las noticias, se presentaron problemas con las etiquetas que normalmente abarcan más de un renglón, esto es, que empiezan en un determinado lugar de la noticia y su etiqueta que cierra se encuentra varios renglones después, como ejemplos se pueden mencionar las etiquetas <style>, <script>, <img>, <a>, <table>.

El problema anterior se solucionó buscando un carácter "<" y omitiendo todo su contenido hasta que se encuentre el carácter "/>" pero de su misma etiqueta.

En la programación, también se buscaron etiquetas especiales de html que no necesariamente llevan los caracteres "<>", previamente definidos en una lista, como son entre otros: "<" (<), ">" (>), "&" (&), "" ("), "™" (®), "©" (©), "espacio", ( ), á ñ etc. Estas etiquetas también serán eliminadas.

Ahora se muestra la misma noticia limpia de etiquetas html.

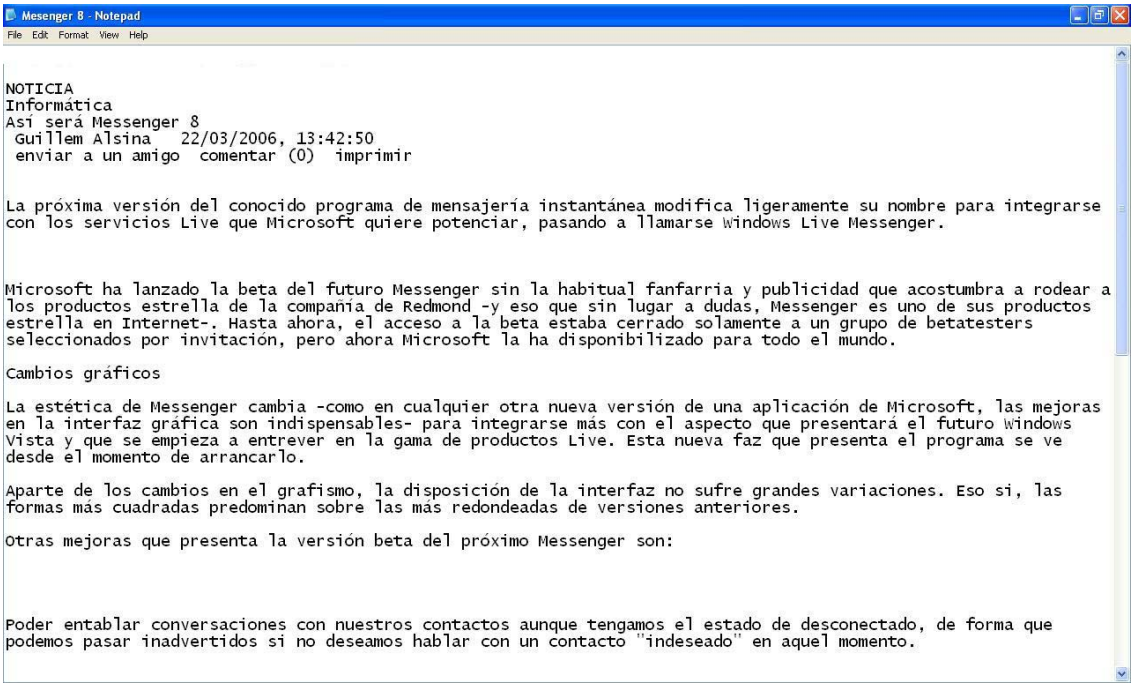
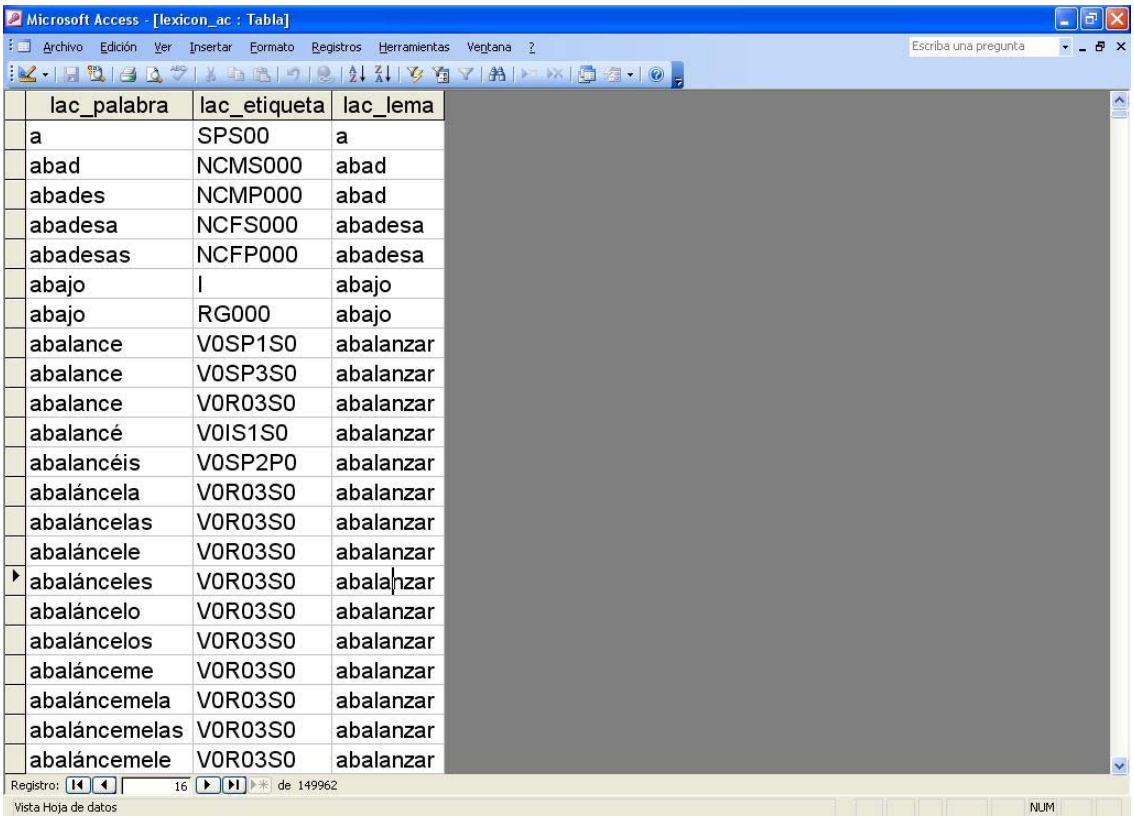


Figura 7.9.- La noticia ha sido almacenada en un archivo txt después de haber sido limpiada de html.

7.4.2. Módulo 2

Se continúa con los procesos de **tokenización, lematización y etiquetado pos**.

El lexicón que se utilizó se encuentra almacenado en varias tablas de *Microsoft Access*. Se accede a la base de datos por medio de un ODBC definido para hacer la conexión entre *Java* y la misma.



lac_palabra	lac_etiqueta	lac_lema
a	SPS00	a
abad	NCMS000	abad
abades	NCMP000	abad
abadesa	NCFS000	abadesa
abadesas	NCFP000	abadesa
abajo	I	abajo
abajo	RG000	abajo
abalance	V0SP1S0	abalanzar
abalance	V0SP3S0	abalanzar
abalance	V0R03S0	abalanzar
abalancé	V0IS1S0	abalanzar
abalancéis	V0SP2P0	abalanzar
abaláncela	V0R03S0	abalanzar
abaláncelas	V0R03S0	abalanzar
abaláncele	V0R03S0	abalanzar
abalánceles	V0R03S0	abalanzar
abaláncelo	V0R03S0	abalanzar
abaláncelos	V0R03S0	abalanzar
abalánceme	V0R03S0	abalanzar
abaláncemela	V0R03S0	abalanzar
abaláncemelas	V0R03S0	abalanzar
abaláncemele	V0R03S0	abalanzar

Figura 7.10.- Vista de un fragmento de la tabla donde se almacena el lexicón para el idioma español.

El nombre de los archivos de entrada que contienen cada una de las noticias es almacenado en otra tabla de la misma base de datos de *Access*.

Luego entonces, el programa en java se conecta a dicha tabla para extraer el listado de los nombres del corpus.

Al momento de leer el contenido de los archivos, se divide el texto en párrafos a través de todo signo de punto (.) que cumpla con un patrón, que consiste en:

- Punto y a parte.
- Dos o más espacios continuos.

El punto final se etiqueta con otra marca especial definida en la programación denominada "_FIDEOR" (final de oración) y posteriormente se le asigna también un salto de línea (\n). Esto se hace debido a que en un fragmento como:

"…….el nuevo producto."

podrían encontrarse las dos primeras palabras de este fragmento dentro del lexicón ("el" y "nuevo"), pero no se puede encontrar "producto." ("producto" "punto").

En cambio, sí se podrían encontrar: "el" "nuevo" "producto" ".".

Para esto, todo punto no decimal se forzó para que fuera punto y a parte a través del salto de línea, formando enunciados gráficos a través de oraciones en lugar de trabajar por los párrafos que venían predefinidos en la redacción de las noticias.

Una vez definidos los enunciados, se continúa con la búsqueda de palabras en el lexicón.

En un principio la búsqueda de las palabras en el lexicón para efectos de etiquetado era demasiado tardada, alrededor de 14 horas para un corpus de solamente 60 archivos. Esto se debió a que cada una de las palabras se buscaba en todo el lexicón, empezando desde la letra A y continuando hasta que la encontrara.

Palabras como "actualización", "antes", "ahora", etc., no representaban tanto problema porque eran encontradas rápidamente, pero cuando tocaba a palabras como "*Verisign*" o "veces" la búsqueda era muy tardada, porque la segunda se encuentra casi al final del lexicón, pero la primera, por ser una compañía de *software*, no se encuentra en los dos primeros lexicones (con acentos y sin acentos) sino hasta el *gazetteer* de compañías.

Para asegurar una mayor rapidez, se indexó el campo "palabra", que contiene en sí la palabra, de cada tabla (dentro de las propiedades de *Access*), lo que permitió un mejor desempeño cambiando a siete horas aproximadamente.

Entonces lo que se hizo fue dividir tanto el lexicón general como el lexicón de palabras sin acentos en tablas de acuerdo su primera letra, por lo tanto se tuvieron 29 tablas (27 para las letras del alfabeto, 1 de las compañías y otra más de las palabras sin acentos), una para las palabras que comenzaran con la letra "A", otra tabla con las palabras que comenzaran con "B" y así sucesivamente.

Lo anterior mejoró notablemente la rapidez del proceso, reduciendo la búsqueda y etiquetado de 60 noticias a sólo 7 minutos aproximadamente.

Finalmente aparece la noticia etiquetada en base a los lexicones

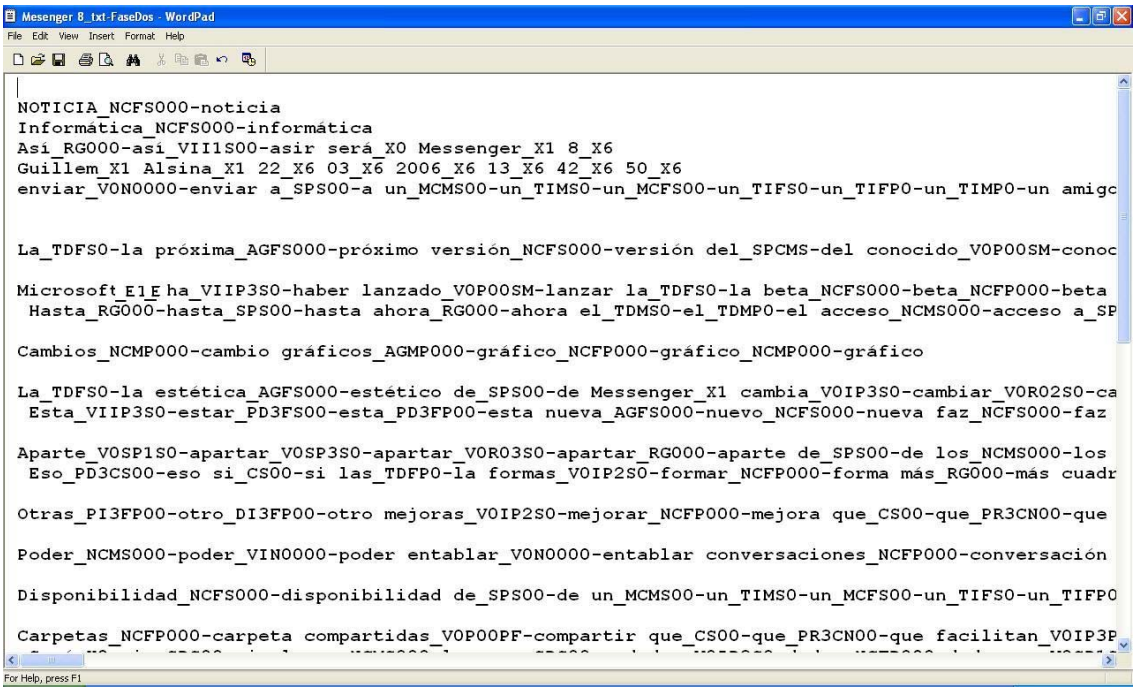


Figura 7.11.- Archivo de la noticia después del proceso de etiquetado.

Como se puede observar en el ejemplo anterior, hay un fragmento de texto como el siguiente:

Microsoft_E1_E ha_VIIP3S0-haber lanzado_V0P00SM-lanzar la_TDFS0-la beta_NCFS000-beta_NCFP000-beta del_SPCMS-del futuro_AGMS000-futuro_NCMS000-futuro Messenger_X1 sin_SPS00-sin la_TDFS0-la habitual_AGIS000-habitual_AGIS000-habitual fanfarria_NCFS000-fanfarria y_CC00-y publicidad_NCFS000-publicidad que_CS00-que_PR3CN00-que acostumbra_V0IP3S0-acostumbrar_V0R02S0-acostumbrar a_SPS00-a rodear_V0N0000-rodear a_SPS00-a los_NCMS000-los productos_NCMP000-producto estrella_V0IP3S0-estrellar_V0R02S0-estrellar_NCFS000-estrella_NP00000-estrella de_SPS00-de la_TDFS0-la compañía_NCFS000-

compañía de_SPS00-de Redmond_X1 y_CC00-y eso_PD3CS00-eso que_CS00-que_PR3CN00-que sin_SPS00-sin lugar_NCMS000-lugar a_SPS00-a dudas_V0IP2S0-dudar_NCFP000-duda Messenger_X1 es_VIIP3S0-ser uno_V0IP1S0-unir_PI0FS00-uno_PI0FP00-uno_MC00000-uno_PI0MS00-uno_PI0MP00-uno de_SPS00-de sus_PS0P000-sus productos_NCMP000-producto estrella_V0IP3S0-estrellar_V0R02S0-estrellar_NCFS000-estrella_NP00000-estrella en_SPS00-en Internet_X1 ._FIDEOR
Hasta_RG000-hasta_SPS00-hasta ahora_RG000-ahora el_TDMS0-el_TDMP0-el acceso_NCMS000-acceso a_SPS00-a la_TDFS0-la beta_NCFS000-beta_NCFP000-beta estaba_VIII1S0-estar_VIII3S0-estar cerrado_V0P00SM-cerrar_AGMS000-cerrado solamente_RG000-solamente_RG000-solamente a_SPS00-a un_MCMS00-un_TIMS0-un_MCFS00-un_TIFS0-un_TIFP0-un_TIMP0-un grupo_NCMS000-grupo de_SPS00-de betatesters_X2 seleccionados_V0P00PM-seleccionar por_SPS00-por invitación_NCFS000-invitación pero_CC00-pero_RG000-pero ahora_RG000-ahora Microsoft_E1_E la_TDFS0-la ha_VIIP3S0-haber disponibilizado_X2 para_V0IP3S0-parar_V0R02S0-parar_V0SP1S0-parir_V0SP3S0-parir_V0R03S0-parir_SPS00-para todo_DI3FS00-todo_PI3FS00-todo_DI3FP00-todo_PI3FP00-todo_DI3MS00-todo_NCMS000-todo_PI3MS00-todo_RG000-todo_DI3MP00-todo_PI3MP00-todo el_TDMS0-el_TDMP0-el mundo_NCMS000-mundo ._FIDEOR

En éste se puede observar que la palabra *Microsoft* tiene asignada la etiqueta "_E1_E" porque así se estableció en el programa al momento de encontrarla en el *gazetteer*, y cada una de las palabras encontradas en uno de los dos lexicones para el español que se tienen, llevan la estructura "palabra_etiqueta-lema" (ha_VIIP3S0-haber, lanzado_V0P00SM-lanzar).

Ahora se continúa con la comparación de las expresiones regulares, las cuáles son hipótesis o suposiciones de nuevos productos de *software*, y que obtendrán la compañía, producto y la versión en el contenido de la noticia.

Esos patrones se buscan en toda la noticia y no son restrictivos, es decir, al coincidir un patrón de la noticia con una expresión regular, se sigue buscando ese misma *regex* en toda la noticia y no se detiene donde haya coincidido por primera vez.

```
StringTokenizer stk = new StringTokenizer(linea, " ");

while(stk.hasMoreTokens())
{
buf_car1=stk.nextToken();
//Se buscan las palabras clave de lanzamiento
```

```
buf_car2=buf_car1.toLowerCase();

if(buf_car2.indexOf("lanzar") >= 0 || buf_car2.indexOf("crear") >= 0 ||
buf_car2.indexOf("liberar") >= 0 || buf_car2.indexOf("presentar") >= 0 ||
buf_car2.indexOf("anunciar") >= 0 || buf_car2.indexOf("beta") >= 0 ||
buf_car2.indexOf("Beta") >= 0 || buf_car2.indexOf("publicar") >= 0 ||
buf_car2.indexOf("colocar") >= 0 || buf_car2.indexOf("lanzar") >= 0 ||
buf_car2.indexOf("crear") >= 0 || buf_car2.indexOf("liberar") >= 0 ||
buf_car2.indexOf("programa") >= 0 || buf_car2.indexOf("presentar") >= 0 ||
buf_car2.indexOf("anunciar") >= 0 || buf_car2.indexOf("publicar") >= 0 ||
buf_car2.indexOf("colocar") >= 0 || buf_car2.indexOf("lanzamiento") >= 0 ||
buf_car2.indexOf("nuevo") >= 0 || buf_car2.indexOf("nueva") >= 0 ||
buf_car2.indexOf("versión") >= 0 || buf_car2.indexOf("version") >= 0 ||
buf_car2.indexOf("disponible") >= 0 || buf_car2.indexOf("disponibilidad") >=
0 || buf_car2.indexOf("solución") >= 0 || buf_car2.indexOf("producto") >= 0)
{
sql_actualizar="insert into palabras_c(pc_pagina,pc_parrafo) values
('"+pagina+"',"+pa2+")";
res=actualizar.executeUpdate(sql_actualizar);

if(res==0)
{
System.out.println("Error en la actualización de las palabras clave de la
página: "+pagina);
}
else
{
System.out.println("Palabra clave de lanzamiento:
"+buf_car1.substring(0,buf_car1.indexOf("_")));
}
}
//Se termina de buscar las palabras clave de lanzamiento

//Se buscan las palabras clave de informática
if(buf_car2.indexOf("software") >= 0 || buf_car2.indexOf("ordenador") >= 0
|| buf_car2.indexOf("computadora") >= 0 || buf_car2.indexOf("hardware")
>= 0 || buf_car2.indexOf("digital") >= 0 || buf_car2.indexOf("tecnología") >=
0 || buf_car2.indexOf("tecnologia") >= 0 || buf_car2.indexOf("informatica")
>= 0 || buf_car2.indexOf("pc") >= 0 || buf_car2.indexOf("cpu") >= 0 ||
buf_car2.indexOf("XML") >= 0 || buf_car2.indexOf("-documento") >= 0 ||
buf_car2.indexOf("-red") >= 0 || buf_car2.indexOf("multimedia") >= 0)
{
sql_actualizar="insert into palabras_ti(pti_pagina) values ('"+pagina+"')";
res=actualizar.executeUpdate(sql_actualizar);

if(res==0)
{
```

```
System.out.println("Error en la actualización de las palabras clave de la
página: "+pagina);
}
else
{
System.out.println("Palabra clave de tecnología:
"+buf_car1.substring(0,buf_car1.indexOf("_")));
}
}
//Se termina de buscar las palabras clave de informática

//Se buscan patrones para identificar el producto

//Se busca el patrón:
//renovar|producto|disponible|disponibilidad|Beta|beta|nueva|nuevo|ofrecer|d
esarrolladores|crear|liberar|presentar|anunciar|publicar|lanzar|version|versión
|puesto .* X1|X3|X4|X5
if((p1b || p1c) && (buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3")
>= 0 || buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0))
{
pro1=pro1+buf_car1.substring(0,buf_car1.indexOf("_X"))+" ";
p1c=true;
}
else
{
if(p1c)
{
p1c=false;
pro1=pro1.trim();

sql_insert="insert into producto(pro_pagina,pro_producto,pro_parrafo)
values('"+pagina+"','"+pro1+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la inserción del producto 1: "+pro1);
System.out.println("De la página: "+pagina);
}

System.out.println("Producto 1: "+pro1);
pro1="";
}
}

if(p1a)
{
p1b=true;
```

```
}
else
{
p1b=false;
}

if(buf_car1.indexOf("renovar") >= 0 || buf_car1.indexOf("producto") >= 0 ||
buf_car1.indexOf("disponible") >= 0 || buf_car1.indexOf("disponibilidad") >=
0 || buf_car1.indexOf("Beta") >= 0 || buf_car1.indexOf("beta") >= 0 ||
buf_car1.indexOf("nueva") >= 0 || buf_car1.indexOf("nuevo") >= 0 ||
buf_car1.indexOf("ofrecer") >= 0 || buf_car1.indexOf("desarrolladores") >= 0
|| buf_car1.indexOf("crear") >= 0 || buf_car1.indexOf("liberar") >= 0 ||
buf_car1.indexOf("presentar") >= 0 || buf_car1.indexOf("anunciar") >= 0 ||
buf_car1.indexOf("publicar ") >= 0 || buf_car1.indexOf("lanzar") >= 0 ||
buf_car1.indexOf("versión") >= 0 || buf_car1.indexOf("version") >= 0 ||
buf_car1.indexOf("puesto") >= 0 || buf_car1.indexOf("informar") >= 0)
{
p1a=true;
}
else
{
p1a=false;
}

//Se busca el patrón:
//nueva|nuevo|software X1|X3|X4|X5

if((p2a || p2b) && (buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3")
>= 0 || buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0))
{
pro2=pro2+buf_car1.substring(0,buf_car1.indexOf("_X"))+" ";

p2b=true;
}
else
{
if(p2b)
{
p2b=false;
pro2=pro2.trim();

sql_insert="insert into producto(pro_pagina,pro_producto,pro_parrafo)
values('"+pagina+"','"+pro2+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la inserción del producto 2: "+pro2);
```

```
System.out.println("De la página: "+pagina);
}

System.out.println("Producto 2: "+pro2);
pro2="";
}
}

if(buf_car1.indexOf("renovar") >= 0 || buf_car1.indexOf("producto") >= 0 ||
buf_car1.indexOf("disponible") >= 0 || buf_car1.indexOf("disponibilidad") >=
0 || buf_car1.indexOf("software") >= 0 || buf_car1.indexOf("Beta") >= 0 ||
buf_car1.indexOf("beta") >= 0 || buf_car1.indexOf("nueva") >= 0 ||
buf_car1.indexOf("nuevo") >= 0 || buf_car1.indexOf("ofrecer") >= 0 ||
buf_car1.indexOf("desarrolladores") >= 0 || buf_car1.indexOf("crear") >= 0
|| buf_car1.indexOf("liberar") >= 0 || buf_car1.indexOf("presentar") >= 0 ||
buf_car1.indexOf("anunciar") >= 0 || buf_car1.indexOf("publicar ") >= 0 ||
buf_car1.indexOf("lanzar") >= 0 || buf_car1.indexOf("versión") >= 0 ||
buf_car1.indexOf("version") >= 0 || buf_car1.indexOf("puesto") >= 0 ||
buf_car1.indexOf("informar") >= 0)
{
p2a=true;
}
else
{
p2a=false;
}

//Se busca el patrón:
//X1|X3|X4|X5|publica|X6 .* X1|X3|X4|X5
if((p3b || p3c) && (buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3")
>= 0 || buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0))
{
pro3=pro3+buf_car1.substring(0,buf_car1.indexOf("_X"))+" ";

p3c=true;
}
else
{
if(p3c)
{
p3c=false;
pro3=pro3.trim();

sql_insert="insert into producto(pro_pagina,pro_producto,pro_parrafo)
values('"+pagina+"','"+pro3+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
```

```
{
System.out.println("Error en la inserción del producto 3: "+pro3);
System.out.println("De la pagina: "+pagina);
}

System.out.println("Producto 3: "+pro3);
pro3="";
}
}

if(p3a)
{
p3b=true;
}
else
{
p3b=false;
}

if(buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3") >= 0 ||
buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0 ||
buf_car1.indexOf("publica") >= 0 || buf_car1.indexOf("pública") >= 0 ||
buf_car1.indexOf("_X6") >= 0)
{
p3a=true;
}
else
{
p3a=false;
}

//Se busca el patrón:
//X1|X3|X4|X5 software|un|una|es|X4|X6

if(p4a && (buf_car1.indexOf("software") >= 0 || buf_car1.indexOf("un") >= 0
|| buf_car1.indexOf("una") >= 0 || buf_car1.indexOf("es") >= 0 ||
buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X6") >= 0))
{
pro4=pro4.trim();
sql_insert="insert into producto(pro_pagina,pro_producto,pro_parrafo)
values('"+pagina+"','"+pro4+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la inserción del producto 4: "+pro4);
System.out.println("De la pagina: "+pagina);
}
```

```
System.out.println("Producto 4: "+pro4);
}

if(buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3") >= 0 ||
buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0)
{
pro4=pro4+buf_car1.substring(0,buf_car1.indexOf("_X"))+" ";
p4a=true;
}
else
{
pro4="";
p4a=false;
}
//Se termina de buscar patrones para identificar el producto

//Se buscan patrones para identificar la compañía

//Se busca el patrón: buf_car1.indexOf("_E") >= 0 ||
//presentar|compañía|software|X1|X3|X4|X5 .* X1|X3|X4|X5
if((c1b || c1c) && (buf_car1.indexOf("_E") >= 0 || buf_car1.indexOf("_X1")
>= 0 || buf_car1.indexOf("_X3") >= 0 || buf_car1.indexOf("_X4") >= 0 ||
buf_car1.indexOf("_X5") >= 0))
{
com1=com1+buf_car1.substring(0,buf_car1.indexOf("_"))+" ";
c1c=true;
}
else
{
if(c1c)
{
c1c=false;
com1=com1.trim();

sql_insert="insert into empresa(emp_pagina,emp_nombre,emp_parrafo)
values('"+pagina+"','"+com1+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la inserción del empresa 1: "+com1);
System.out.println("De la pagina: "+pagina);
}

System.out.println("Empresa 1: "+com1);
com1="";
}
```

```
}

if(c1a)
{
c1b=true;
}
else
{
c1b=false;
}

if(buf_car1.indexOf("presentar") >= 0 || buf_car1.indexOf("compañía") >= 0
|| buf_car1.indexOf("compañia") >= 0  || buf_car1.indexOf("software") >= 0
|| buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3") >= 0 ||
buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0)
{
c1a=true;
}
else
{
c1a=false;
}

//Se busca el patrón:
//compañía|presentar X1|X3|X4|X5
if((c2a || c2b) && (buf_car1.indexOf("_E") >= 0 || buf_car1.indexOf("_X1")
>= 0 || buf_car1.indexOf("_X3") >= 0 || buf_car1.indexOf("_X4") >= 0 ||
buf_car1.indexOf("_X5") >= 0))
{
com2=com2+buf_car1.substring(0,buf_car1.indexOf("_"))+" ";
c2b=true;
}
else
{
if(c2b)
{
c2b=false;
com2=com2.trim();

sql_insert="insert into empresa(emp_pagina,emp_nombre,emp_parrafo)
values('"+pagina+"','"+com2+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la insercion del empresa 2: "+com2);
System.out.println("De la pagina: "+pagina);
}
```

```
System.out.println("Empresa 2: "+com2);
com2="";
}
}
if(buf_car1.indexOf("compañía") >= 0 || buf_car1.indexOf("compañia") >= 0
|| buf_car1.indexOf("presentar") >= 0)
{
c2a=true;
}
else
{
c2a=false;
}
//Se busca el patrón:
//X1|X3|X4|X5 .*
vender|lanzar|crear|liberar|presentar|anunciar|publicar|colocar|informar|ha|q
ue|dar|desarrollar|ofrecer|empresa
if(c3b && (buf_car1.indexOf("empresa") >= 0 || buf_car1.indexOf("que") >=
0 ))//|| buf_car1.indexOf("vender") >= 0 || buf_car1.indexOf("lanzar") >= 0
|| buf_car1.indexOf("crear") >= 0 || buf_car1.indexOf("liberar") >= 0 ||
buf_car1.indexOf("presentar") >= 0 || buf_car1.indexOf("anunciar") >= 0 ||
buf_car1.indexOf("publicar") >= 0 || buf_car1.indexOf("colocar") >= 0 ||
buf_car1.indexOf("informar") >= 0 || buf_car1.indexOf("ha") >= 0 ||
buf_car1.indexOf("dar") >= 0 || buf_car1.indexOf("desarrollar") >= 0 ||
buf_car1.indexOf("ofrecer") >= 0))
{
com3=com3.trim();
sql_insert="insert into empresa(emp_pagina,emp_nombre,emp_parrafo)
values('"+pagina+"','"+com3+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);
if(res == 0)
{
System.out.println("Error en la inserción de la empresa 3: "+com3);
System.out.println("De la página: "+pagina);
}
System.out.println("Empresa 3: "+com3);
}
if(c3a)
{
c3b=true;
}
else
{
com3="";
c3b=false;
}
```

```
if(buf_car1.indexOf("_E") >= 0 || buf_car1.indexOf("_X1") >= 0 ||
buf_car1.indexOf("_X3") >= 0 || buf_car1.indexOf("_X4") >= 0 ||
buf_car1.indexOf("_X5") >= 0)
{
com3=com3+buf_car1.substring(0,buf_car1.indexOf("_"))+" ";
c3a=true;
}
else
{
c3a=false;
}
//Se busca el patrón:
//X1|X3|X4|X5
acabar|vender|lanzar|crear|liberar|presentar|anunciar|publicar|colocar|inform
ar|ha|que|dar|desarrollar|ofrecer|empresa
if(c4a && (buf_car1.indexOf("acabar") >= 0 || buf_car1.indexOf("vender") >=
0 || buf_car1.indexOf("lanzar") >= 0 || buf_car1.indexOf("crear") >= 0 ||
buf_car1.indexOf("liberar") >= 0 || buf_car1.indexOf("presentar") >= 0 ||
buf_car1.indexOf("anunciar") >= 0 || buf_car1.indexOf("publicar") >= 0 ||
buf_car1.indexOf("colocar") >= 0 || buf_car1.indexOf("informar") >= 0 ||
buf_car1.indexOf("ha") >= 0 || buf_car1.indexOf("que") >= 0 ||
buf_car1.indexOf("dar") >= 0 || buf_car1.indexOf("desarrollar") >= 0 ||
buf_car1.indexOf("ofrecer") >= 0 || buf_car1.indexOf("empresa") >= 0))
{
com4=com4.trim();
sql_insert="insert into empresa(emp_pagina,emp_nombre,emp_parrafo)
values('"+pagina+"','"+com4+"',"+pa2+")";
res=actualizar.executeUpdate(sql_insert);
if(res == 0)
{
System.out.println("Error en la inserción de empresa 4: "+com4);
System.out.println("De la página: "+pagina);
}
System.out.println("Empresa 4: "+com4);
}

if(buf_car1.indexOf("_E") >= 0 || buf_car1.indexOf("_X1") >= 0 ||
buf_car1.indexOf("_X3") >= 0 || buf_car1.indexOf("_X4") >= 0 ||
buf_car1.indexOf("_X5") >= 0)
{
c4a=true;
com4=com4+buf_car1.substring(0,buf_car1.indexOf("_"))+" ";
}
else
{
com4="";
c4a=false;
}
```

```
//Se termina de buscar patrones para identificar la compañía

//Se buscan patrones para identificar la versión

//Se busca el patrón:
//versión X4|X6
if(v1a && (buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X6") >= 0))
{
sql_insert="insert into version(ver_pagina,ver_version,ver_parrafo)
values('"+pagina+"','"+buf_car1.substring(0,buf_car1.indexOf("_"))+"',"+pa2
+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la insercion de la version 1: "+buf_car1);
System.out.println("De la pagina: "+pagina);
}

System.out.println("Version 1:
"+buf_car1.substring(0,buf_car1.indexOf("_X")));
}

if(buf_car1.indexOf("versión") >= 0 || buf_car1.indexOf("version") >= 0 )
{
v1a=true;
}
else
{
v1a=false;
}

//Se busca el patrón:
//versión (b|B)eta
if(v2a && (buf_car1.indexOf("beta") >= 0 || buf_car1.indexOf("Beta") >= 0))
{
sql_insert="insert into version(ver_pagina, ver_version, ver_parrafo)
values('"+pagina+"','"+buf_car1.substring(0,buf_car1.indexOf("_"))+"',"+pa2
+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la inserción de la version 2: "+buf_car1);
System.out.println("De la página: "+pagina);
}
```

```
System.out.println("Version 2:
"+buf_car1.substring(0,buf_car1.indexOf("_")));
}

if(buf_car1.indexOf("versión") >= 0 || buf_car1.indexOf("version") >= 0 )
{
v2a=true;
}
else
{
v2a=false;
}

//Se busca el patrón:
//X1|X3|X4|X5 X4|X6
if(v3a && (buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X6") >= 0))
{
sql_insert="insert into version(ver_pagina,ver_version,ver_parrafo)
values('"+pagina+"','"+buf_car1.substring(0,buf_car1.indexOf("_"))+"',"+pa2
+")";
res=actualizar.executeUpdate(sql_insert);

if(res == 0)
{
System.out.println("Error en la inserción de la version 3: "+buf_car1);
System.out.println("De la página: "+pagina);
}

System.out.println("Version 3:
"+buf_car1.substring(0,buf_car1.indexOf("_")));
}

if(buf_car1.indexOf("_X1") >= 0 || buf_car1.indexOf("_X3") >= 0 ||
buf_car1.indexOf("_X4") >= 0 || buf_car1.indexOf("_X5") >= 0)
{
v3a=true;
}
else
{
v3a=false;
}
//Se termina de buscar patrones para identificar la versión
}
}
leer.close();
}
conexion.close();
```

El algoritmo de comparación de expresiones regulares (*regex*) funciona de la siguiente manera:

Algunas de las expresiones regulares que se definieron para la búsqueda en el contenido de la noticia son las siguientes:

Productos

- renovar|producto|disponible|disponibilidad|Beta|beta|nuevo|nueva|ofrecer|desarrolladores|crear|liberar|presentar|anunciar|publicar|lanzar|version|versión|puesto|informar .* X1|X3|X4|X5
- renovar|producto|disponible|disponibilidad|software|Beta|beta|nuevo|nueva|ofrecer|desarrolladores|crear|liberar|presentar|anunciar|publicar|lanzar|version|versión|puesto|informar X1|X3|X4|X5

- X1|X3|X4|X5|publica|X6 .* X1|X3|X4|X5

- X1|X3|X4|X5 software|un|es|una|X4|X6

Compañías

- presentar|compañía|software|X1|X3|X4|X5 .* X1|X3|X4|X5

- compañía|presentar X1|X3|X4|X5

- X1|X3|X4|X5 .* vender|lanzar|crear|liberar|presentar|anunciar|publicar|colocar|informar|ha|que|dar|desarrollar|ofrecer|empresa

- X1|X3|X4|X5 acabar|vender|lanzar|crear|liberar|presentar|anunciar|publicar|colocar| informar|ha|que|dar|desarrollar|ofrecer|empresa

Versiones

- versión|version X4|X6

- versión (b|B)eta

- X1|X3|X4|X5 X4|X6

Con base en cada *token* que conforma la *regex*, se va buscando uno por uno dentro de toda la noticia. En caso de que un token de la *regex* se encuentre dentro del texto, se enciende una bandera (variable *booleana*) y sigue comparando para encontrar el siguiente *token* correspondiente a la misma *regex*. Si el siguiente *token* es encontrado, se enciende una segunda bandera y el proceso sigue comparando hasta

que haya encontrado el último *token* de la *regex*. La cantidad de comparaciones es directamente proporcional a la longitud de la expresión.

Para que una *regex* sea almacenada en la variable y se califique como exitosa, deben estar encendidas todas las banderas una tras otra y esto se logra a través de una estructura de control denominada "*If-then*".

De lo contrario, en el momento en que una bandera de la *regex* no se encienda, entonces esa expresión no resultó exitosa y se continúa analizando la siguiente dentro del texto de la noticia.

Las variables donde se almacena el contenido de las *regex* al momento de ser exitosas se envían a distintas tablas de la base de datos dependiendo de la expresión regular.

Los resultados de las variables para las expresiones regulares de "compañía" se envían a una tabla denominada "empresa", y lo mismo pasa con las *regex* para el "producto" y la "versión".

7.4.3. Módulo 3

Ahora sigue el **ordenamiento de los resultados**, el cual se realizará en base a la cantidad de palabras clave, la compañía, producto y versión.

Una vez definidos los posibles párrafos de salida, se hace un conteo para ir eliminando los *templates* que no vayan cumpliendo con los requisitos (compañía, producto, versión y palabras clave) teniendo una jerarquización.

El orden de jerarquía que se sigue es el siguiente:
1.-Cuando se encuentran "compañía", "producto" y "versión", se ordena por número de palabras clave, entonces esta plantilla es la que se muestra.
2.-Cuando se encuentran sólo el producto y la versión entonces también se ordenan por palabras clave.
3.-Cuando sólo se encuentra compañía y producto.
4.-Sólo la compañía y versión.
5.-Cuando sólo se identificó el producto.
6.-Cuando sólo se identificó la compañía.
7.-Cuando sólo se identificó la versión.

Primeramente, las palabras clave se van a enumerar por enunciado gráfico. Con base en el enunciado de la noticia que contenga la mayor cantidad de palabras clave se va a buscar ahí mismo la compañía, el producto y la versión (mediante las expresiones regulares de cada una).

Los resultados obtenidos se mostrarán en *templates* de html.

7.4.4. Módulo 4

Este módulo consiste en la presentación de la información recuperada a través de una interfaz.

Figura 7.12.- Vista de la interfaz inicial de la aplicación.

La lista desplegable que indica "Elige el origen de los datos" se refiere a que se puede ejecutar el proceso desde "Internet" (que es la opción predeterminada) o desde la "PC". Para realizar la opción de esta última, se debe tener almacenado el corpus ya sea en archivos de texto o en archivos de bloc de notas dentro del directorio de trabajo del sistema, en un directorio especificado donde se almacena el corpus de entrada.

El botón de "Iniciar Procesamiento" es claro que inicia todo el proceso para hacer el análisis sobre el corpus, mostrando una pantalla como en la figura 7.12.

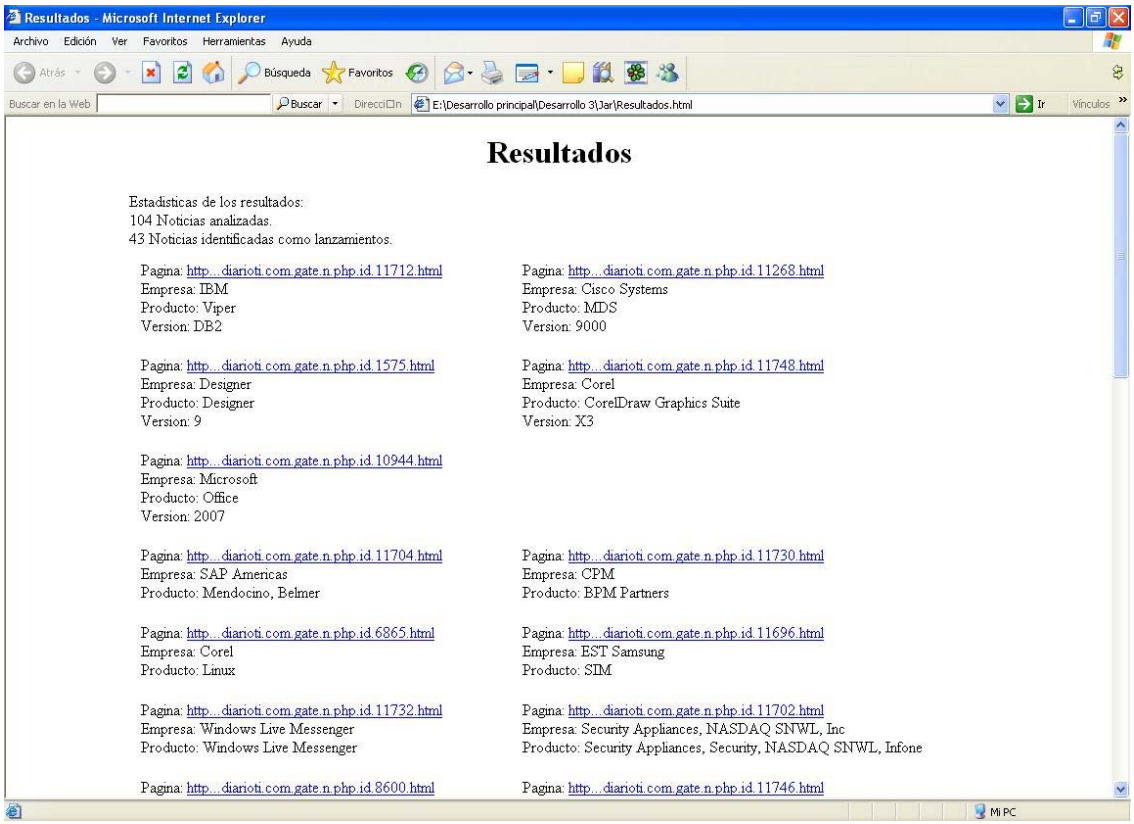


Figura 7.12.- Pantalla de resultados.

Como se puede observar en la pantalla anterior, se muestran registros acomodados en tablas que tienen la descripción de la página donde se analizó la noticia, la empresa que lanzó un nuevo producto y el nombre del mismo.

7.5. Pruebas

Como una limitante se encontró que cuando el programa se corre en internet y halla noticias hechas únicamente en flash estas serán ignoradas.

Las pruebas que se realizaron sobre el funcionamiento del sistema se dividen en 3 partes:
1.-Prueba inicial con las primeras expresiones regulares definidas.
2.-Prueba con las expresiones regulares mejoradas.

3.-Prueba con un corpus final.
4.-Prueba en internet.

Primer prueba: expresiones regulares iniciales

El primer experimento se llevó a cabo con un corpus elaborado de 60 noticias en formato html que contenían lanzamientos de *software* de los que logró reconocer las siguientes cifras:

Entidades	Reconoció	Correctas	Incorrectas
Empresas	74	68	16
Productos	15	9	6
Versiones	21	21	0

Expresiones regulares que se usaron:
Productos:

- Beta|beta|nuevo|nueva|ofrecer|desarrolladores|crear|liberar|presentar|anunciar|publicar|lanzar|version|versión|puesto|informar .* X1|X3|X4|X5.
- nueva|nuevo|software X1|X3|X4|X5.
- beta|Beta pública .* X1|X3|X4|X5.
- versión|version X6 .* X1|X3|X4|X5.
- X1|X3|X4|X5 software|un|es|una|X4|X6.

Compañías:

- compañía .* X1|X3|X4|X5.
- X1|X3|X4|X5.*lanzar|crear|liberar|presentar|anunciar|publicar|colocar|informar|ha|que| dar|desarrollar|ofrecer|empresa.
- X1|X3|X4|X5 lanzar|crear|liberar|presentar|anunciar|publicar|colocar|informar|ha|que|dar| desarrollar| ofrecer|empresa.
- software de X1|X3|X4|X5.
- software X1|X3|X4|X5 de X1|X3|X4|X5 (las ultimas X son las compañías que se buscan).

Versiones:

- versión|version X4|X6.
- versión (b|B)eta.

Muchas de las expresiones regulares en esta primera fase de prueba eran muy restrictivas, por lo que eran muy exactas pero en detrimento de la cantidad de entidades recuperadas, es decir, daban pocos resultados pero muy exactos, por lo que se decidió tomar como una medida para solucionar este problema hacerlas más generales.

En algunas expresiones regulares faltaban más verbos y adjetivos que puedan completar de una mejor forma estos modelos ya propuestos para así también recuperar más entidades. Se trabajó en la investigación dentro del corpus de prueba en la búsqueda de una mayor cantidad de verbos y adjetivos que vengan a enriquecer la efectividad de las expresiones regulares.

Segunda prueba: expresiones regulares mejoradas

Tomando en cuenta la experiencia pasada se buscó subsanar los errores anteriormente señalados.

El corpus trabajado ahora es de 241 noticias:

Entidades	Reconoció
Empresas	3104
Productos	5310
Versiones	1137

Las expresiones regulares usadas fueron:

Productos

- renovar|producto|disponible|disponibilidad|Beta|beta|nuevo|nueva|ofrecer|desarrolladores|crear|liberar|presentar|anunciar|publicar|lanzar|version|versión|puesto|informar.* X1|X3|X4|X5.
- renovar|producto|disponible|disponibilidad|software|Beta|beta|nuevo|nueva|ofrecer|desarrolladores|crear|liberar|presentar|anunciar|publicar|lanzar|version|versión|puesto|informar X1|X3|X4|X5.
- X1|X3|X4|X5|publica|X6 .* X1|X3|X4|X5.
- X1|X3|X4|X5 software|un|es|una|X4|X6.

Compañías

- presentar|compañía|software|X1|X3|X4|X5 .* X1|X3|X4|X5.
- compañía|presentar X1|X3|X4|X5.
- X1|X3|X4|X5.*vender|lanzar|crear|liberar|presentar|anunciar|publicar|colocar|informar|ha|que|dar|desarrollar|ofrecer|empresa.
- X1|X3|X4|X5 acabar|vender|lanzar|crear|liberar|presentar|anunciar|publicar|colocar|informar|ha|que|dar|desarrollar|ofrecer|empresa.

Versiones

- versión|version X4|X6.
- versión (b|B)eta.
- X1|X3|X4|X5 X4|X6.

La cantidad de entidades recuperadas aumentó pero la precisión disminuyó. De los resultados se deduce que convertir las expresiones regulares más generales y menos restrictivas, además de aumentar los adjetivos y verbos para su procesamiento, hace que la cantidad de entidades recuperadas aumente considerablemente pero la exactitud se reduzca, ahora se trabajará sobre los verbos y adjetivos para poder localizar la manera en que se puedan hacer más exactas las expresiones regulares.

Prueba con corpus final

Con base en la experiencia de los anteriores experimentos, por primera ocasión se probó el programa en términos de noticias de nuevos lanzamientos de *software* recuperadas y no de entidades recuperadas.

El corpus ahora es de 70 noticias que se forman de la siguiente manera:
10 noticias ajenas a tecnologías o lanzamientos de productos de *software* (espectáculos, deportes, política, etc.).
20 son de tecnologías de información pero no son lanzamientos de *software*.
10 son lanzamientos de productos completamente ajenos a *software*.
30 son lanzamientos de *software*, que es la parte que el sistema SEINS debe extraer.

Noticias de nuevos lanzamientos contenidas en el corpus:	30
Noticias recuperadas:	26
Noticias recuperadas correctas:	19

Precision	73.03%
Recall	86.66%

En esta ocasión se recuperaron 26 noticias de las 30 relacionadas al lanzamiento de *software*, resultando en una recuperación del 86.66% y una precisión de poco más del 73%, es decir, se extrajo bastante de la información requerida con una buena exactitud.

Experimento final en internet

Este experimento es el primero en ser realizado en un ambiente de prueba real, no manipulado previamente.

Sitio que se utilizó para la prueba: http://www.diarioti.com

Total de noticias publicadas el día de la prueba (junio-06) que eran accesibles directamente desde la página principal del sitio: 104.

Cantidad de noticias descargadas en dicho portal *web*: 104

Resultado del análisis manual:
Lanzamientos de *software* contenidos en el portal *web*: 26

Resultados arrojados por SEINS:
Noticias que reconoció como lanzamientos de *software*: 43

Análisis de resultados del SEINS:
Resultados correctos: 13
Resultados incorrectos: 30

Muchas de las noticias extraídas contienen información de lanzamientos de otros tipos, por ejemplo dentro de los resultados arrojados por SEINS se encontraron ocho lanzamientos de nuevo *hardware*, el resto de los lanzamientos identificados fueron dados porque dentro de algunas noticias se lanzaba un nuevo servicio, una nueva certificación, se presentaba una nueva estrategia de negocios o a un nuevo presidente de alguna empresa. Solo en seis casos de las noticias extraídas no se refiere a algún tipo de lanzamiento de ninguna índole.

Debido a los resultados anteriores se determina que la aplicación SEINS presenta un problema de semántica, pues los resultados indican que le es difícil ubicar noticias de lanzamientos de nuevo *software* y diferenciarlas de noticias de lanzamientos de nuevos productos o acontecimientos ajenos al *software*.

7.6. Implementación

Durante el proceso de implementación también se hicieron pruebas finales, para determinar la confiabilidad del sistema en base a su recuperación y precisión.

Las pruebas de implementación consistieron en probar el sistema ya en un ambiente preparado, listo para trabajar.

Dichos resultados ya se registraron en la parte de pruebas de este capítulo.

Cuando el sistema SEINS se vaya a implementar en producción en alguna empresa, solo se necesita tener la máquina virtual de java instalada, así como Microsoft Access y un navegador *web*.

7.7. Actualización

La actualización básicamente consiste en ir añadiendo los sitios *web* que se dediquen a la publicación de noticias de *software*.

Dentro del directorio de trabajo de la aplicación se encuentra la base de datos en *Access* donde se encuentran almacenados el lexicón de palabras, el *gazetteer* de compañías y los resultados del proceso de extracción de información. También se encuentra una tabla denominada "páginas", donde se almacenan los sitios de confianza para ser analizados por el sistema SEINS.

Esta tabla contiene dos campos: pag_index la cual guarda la url completa del sitio *web* (por ejemplo, contando su index.html) mientras que la otra almacena el dominio de dicho portal *web* (pag_dominio).

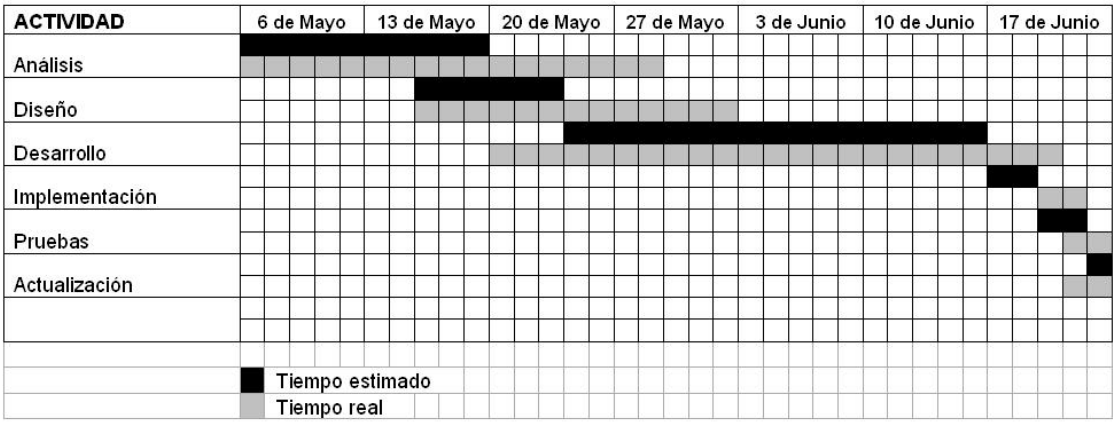
Figura 7.13.- Tabla donde se almacenan los sitios que el sistema va a analizar para la extracción de resultados.

Por otra parte, las actualizaciones complementarias que en un momento se requerirían podrían ser las referentes al *software* que se utiliza para el sistema, tales como *Microsoft Access* o la máquina virtual de java.

Por el momento el sistema es capaz de trabajar en plataformas *Windows* con cualquier tipo de navegadores (aunque ya se mencionó que hay algunos que utilizan el estándar ASCII y otros el Unicode, lo que puede generar problemas de incompatibilidad).

No se podría afirmar que funcione a plenitud en plataformas *Linux* ya que no ha sido probado en las mismas, además de que se tendría que escoger una aplicación diferente a *Access* para almacenar los datos e instalar la máquina virtual de java.

Al término de la aplicación, los pasos de la gráfica de Gantt establecidos en el capítulo 7.1 -véase gráfica de Gantt en cap. 7.1.1- se cumplieron de la manera en que se muestra a continuación:



Capítulo 8

8. Aplicaciones de los sistemas de extracción de información en las organizaciones

En la actualidad la cantidad de información manejada por las empresas es enorme, refiriéndose tanto a la información generada en el interior como a la recibida de terceros (bancos, proveedores, clientes potenciales, etc.) y también a información importante del medio ambiente (información de la competencia, política, sociedad, mercados locales y mundiales, etc.). Todo esto causa gran dificultad al momento de tomar decisiones y desarrollar planes de acción, pues la tarea de recabar la información necesaria de manera rápida y exacta es una labor difícil debido a que en la mayor parte de las ocasiones la información necesaria no está estructurada en alguna base de datos, sino que se encuentra en documentos de texto, y es en esta área donde la extracción de información encuentra su utilidad ya que se especializa en hacer búsquedas en archivos cuyo contenido esté elaborado en lenguaje natural.

8.1. Aplicaciones organizacionales

8.1.1. La información y la organización

Una organización, para funcionar, se compone de los siguientes elementos (Burch 1994:24-30):

- El área de trabajo: las personas de una organización se unen para lograr objetivos en común, en este caso para ofrecer un producto o servicio. Para lograr el objetivo del trabajo se requieren personas con habilidades físicas y mentales de acuerdo a las tareas a realizar.

  a) Los trabajadores operacionales: participan en la construcción de las materias primas para llegar a los productos terminados en las compañías manufactureras.

  b) Trabajadores de la información: son aquellas personas que para realizar su trabajo necesitan de información, trabajan con ella, la crean, la procesan, la distribuyen, interpretan y analizan dicha información. Estas personas manejan los mensajes, las llamadas telefónicas, etc. Así estudian y preparan reportes, dirigen o asisten a reuniones, e inician y dan seguimiento a las actividades. A su vez los trabajadores de la información se dividen en:

i. Usuarios primarios de la información: son todos aquellos gerentes que utilizan información para el control, planeación y toma de decisiones.
ii. Usuarios secundarios: son los usuarios y proveedores de la información, como los contadores.
iii. Otro tipo de usuarios son las personas que dan soporte a la información, como secretarias, programadores, operadores de computadoras, especialistas en tecnología, administradores de bases de datos, analistas de sistemas etc.

A consecuencia del incremento de la información, han aumentado los trabajadores de la misma, pero en algunas ocasiones el equipo para dar soporte a los trabajadores es insuficiente, sin embargo, son necesarios los sistemas de información para proporcionar la misma de manera rápida para todos los tipos de usuarios.

- La cultura

Se define como el ambiente diario que se siente y se observa en las personas que están involucradas en el trabajo. También se conoce como el conjunto de conocimiento acumulado durante la vida de la organización y este se ve reflejado en las promociones, recompensas, correctivos y decisiones, es decir, el comportamiento que ha adoptado la gente de determinada organización. Cada organización tiene su cultura corporativa y es lo que la define de las demás.

- La base de los activos

La base del activo de una organización está formada por las personas, el dinero, las máquinas, el material y los métodos. En este caso sólo se describirán los activos financieros y operacionales.

Los activos financieros son los que se pueden convertir en efectivo fácilmente, estos son activos que proveen la energía de inversión a la organización. Los activos operacionales son todos aquellos activos tangibles e intangibles utilizados para producir y distribuir un producto o servicio.

- Los interesados y afectados:

En el mundo organizacional existe un intercambio constante entre el ambiente y cada organización, incluso aquellas que son cerradas. En primer lugar se requiere información de la imagen que se tiene de la organización en los diferentes sectores de la sociedad, además de la calidad de vida de la misma. Muchos sectores de la sociedad están

involucrados de alguna forma con la organización, por ejemplo, en el impacto ambiental que provoca dicha organización como la contaminación del aire o el agua. También hay preocupación en la calidad y seguridad de los productos y servicios, en las barreras comerciales, en los precios y las prácticas de empleo.

Todos los componentes anteriores deben perseguir el mismo objetivo, además de estar sincronizados entre sí. La información en la organización permite lograr y tener un estado de unidad y armonía.

8.1.2. Calidad y utilidad de la información

La base para el diseño de un sistema de información es saber qué información específica se requiere por parte del usuario. Por ejemplo un usuario requiere un estado de flujo de efectivo y otro conocer las ventas de alguna región, por eso se debe identificar los requerimientos de información de cada individuo para satisfacer sus necesidades de información (Burch 1994:66).

8.2. Aplicaciones generales de los SEI

Cualquier empresa donde se maneje un volumen alto de datos puede hacer uso de la extracción de información.

Por ejemplo:

- El desarrollo de sistemas para extraer información contenida en los correos electrónicos que se reciben o envían en la empresa. De principio parece una labor sencilla buscar entre los correos de una organización si cada individuo administra su cuenta, pero pensando en bancos o aseguradoras cuyos ejecutivos a la hora de planear necesitan estar al tanto de toda la información recibida acerca de algún tema en específico o emitida de la empresa hacia sus proveedores o de la empresa a los clientes la labor de recolectar esta información se vuelve difícil, tardada e inexacta, pero teniendo un sistema automático los resultados serían rápidos y altamente exactos.

- El desarrollo de un sistema de información que haga búsquedas dentro de los datos contables de la empresa (facturas, estados de cuenta, informes financieros, nómina, etc.). Pedir el reporte desglosado de pérdidas o ganancias de algún periodo o área en específico o buscar los costos de compra de un artículo en

particular son labores que tardaría mucho tiempo en elaborar el departamento contable, desde horas hasta días. Si la organización es realmente grande podrían incluso ser semanas, pero si se contase con un sistema que extrajera información específica sólo indicando los detalles de lo necesario se ahorraría mucho tiempo.

- Con un sistema de información se podría monitorear de manera sencilla el flujo de información oficial (si éste se realiza a través de circulares u oficios) dentro de la organización acerca de un tema o proceso que se tenga que realizar a través de varios departamentos, o mirar en los registros pasados y observar con fines de auditoría y mejoramiento de procesos.

Pensando en sistemas como los esbozados anteriormente, se pueden desarrollar aplicaciones mucho muy específicas ubicándolas dentro de las necesidades de la empresa, el giro de ésta y las ventajas competitivas que se desee obtener.

Por otro lado existen funciones dentro de las empresas o incluso empresas dedicadas cuyos elementos principales indudablemente responden a la labor de extracción de información, por ejemplo:

- Una empresa que mantiene un portal de búsqueda, inminentemente tiene áreas donde puede aplicarse la extracción de información. Un buscador en internet hoy en día trabaja examinando la palabra como si fuera una cadena de texto, por lo tanto para los buscadores actuales la palabra "circulación" es interpretada como circulación sanguínea, circulación vial, etc. En cambio un buscador basado en tecnología lingüística solo arrojaría resultados de un solo tema, el que se esté buscando en ese momento y discriminando los demás documentos, que aunque contengan la palabra no pertenezcan al tema buscado.

- También se tienen organizaciones dedicadas enteramente al monitoreo de la información de las campañas mercadológicas, las cuales, si utilizaran una herramienta que les hiciera la búsqueda de documentos en lenguaje natural en internet, las encumbraría sobre su competencia.

8.3. Ventajas de los sistemas de extracción de información

Para poder hablar de las ventajas de los sistemas de extracción de información, primero hay que hablar de las tareas humanas que vienen

a apoyar o sustituir cotidianamente. Para hacer una búsqueda en el internet acerca de los productos de cierta marca o compañía, se necesitaría de una persona que estuviera al tanto mediante múltiples búsquedas revisando cientos de páginas *web*, de las cuales seguramente un alto índice resultarían en páginas de poco interés para el propósito, esto sería tardado, costoso y sujeto a muchas fallas, en cambio si lo realizara un sistema seria rápido, barato y exacto.

Bajo un supuesto de que como ejecutivos se tiene la inquietud de estar al tanto de la opinión de la propia empresa en medios de difusión por el internet (periódicos *online*, *bloggs*[57] especializados, etc.). Para esta labor se necesitaría tener a una persona que realizara dicha tarea con sus consiguientes repercusiones en salario, tiempo de espera y errores humanos, no así si se contara con un sistema que hiciera esta tarea.

En resumen son tres las principales ventajas de un sistema de extracción de datos:

- Reduce costos.
- Reduce el tiempo de espera.
- Aumenta el grado de exactitud.

57 También llamados *weblog* o simplemente *blog*, es una página que alguien escribe en internet de forma personal, que puede ser utilizada como bitácora de actividades, investigaciones, diario personal, etc.

Capítulo 9

9. Conclusiones

9.1. Del análisis realizado al corpus se puede concluir lo siguiente

Las noticias del corpus obtenido están almacenadas en formato libre, así se tiene que mientras en algunas de ellas la fecha de publicación es lo primero que se muestra, en otras se encuentra después del titulo, al final de la noticia o algunas ni siquiera la tienen.

Un aspecto muy presente que se pudo observar es que la versión no siempre es un número inmediatamente precedido por el del nombre del producto, sino que puede ser un número precedido de varias frases o palabras introductoras, por citar un ejemplo; "*Messenger* después de varios meses de anunciarlo ha liberado hoy la versión 7.0", la versión puede ser una palabra en sí, por ejemplo, hay versiones "beta", "alfa", etc.

Otros puntos importantes que se encontraron son:

- En algunos casos se halló que el nombre del producto sólo está en el título y lo mismo ocurre con la versión.

- La mayoría de las noticias no incluyen la página de publicación oficial dentro de la redacción de la misma.

- Es muy probable encontrar que la mayoría de las noticias omiten uno o más elementos que son considerados importantes (antes mencionados), es decir, pueden no tener todo lo que se requiere, como la fecha de publicación, el nombre de la compañía que lo libera etc.

- En algunos casos la etiqueta que identifica la información irrelevante (a la que en un principio se denominó "complemento" pero que después se omitió en el dtd) era la de mayor volumen en casi todas las noticias.

- Del análisis del corpus se pretende rescatar primeramente, las partes que componen cada noticia, como ya se ha mencionado. Ahora la tarea es hacer un análisis lingüístico de éste.

- Se ha concluido que la parte sustanciosa, es decir donde se engloba toda o por lo menos la mayoría de la información a extraer, se identifique como "lanzamiento" con una etiqueta homónima, entonces se haría el análisis lingüístico de esta parte

únicamente. Para realizar este análisis se decidió utilizar el estándar de etiquetado *Eagles* adaptado para el español.

- El nombre de las páginas que se publican muchas veces está contenido dentro del dominio del sitio, es decir, una noticia publicada en:

  http://www.libertaddigital.com/php3/noticia.php3?cpn=1276261684
  El autor es: Libertaddigital.

  http://diarioti.com/gate/n.php?id=10924
  El autor es: diarioti.

- Otras tantas veces el título de la noticia está en la etiqueta *title* del código fuente en *html* de la página que publica la noticia. Por ejemplo:

<*title*>*Palm* lanza junto a *Microsoft* y *Verizon* un móvil basado en el sistema operativo *Windows* - Libertad Digital</*title*>

Y el título de la noticia es: "*Palm* lanza junto a *Microsoft* y *Verizon* un móvil basado en el sistema operativo *Windows*".

- En muchas ocasiones la fecha de lanzamiento o publicación del artículo es la única que aparece en el documento.

- El titulo de las noticias generalmente aparece dentro de las páginas con una fuente mayor o remarcada dentro del documento. Debido a que hay varias maneras de resaltar una fuente dentro de un documento o si se hace a través de hojas de estilo (CSS), el grado de dificultad para su manejo se eleva considerablemente. Es una buena referencia a evaluar pero en cuestión de programación es un reto fuerte.

- Se debe elaborar un listado de empresas (lexicón) para definir elementos como punto de referencia en los artículos, ya que casi todo el *software* publicado es de una empresa conocida. Por ejemplo:
  - *Oracle*.
  - *Microsoft*.
  - *Sun*.
  - *Cisco*.

- Es necesario crear un listado de programas que sea interesante para seguir en sus actualizaciones:
    - *Windows*.
    - *FireFox*.
    - *Open Office*.
    - *Real Player*.

- También se debe tomar en cuenta que muchas de las compañías se llaman igual que el producto que desarrollaron, por lo que había un problema a la hora de definir si eran productos o compañías, por ejemplo:
    - *Oracle* vende la base de datos *Oracle*.
    - *Netscape* distribuye el navegador *Netscape*.
    - *Realnetworks* que también se puede encontrar como simplemente *Real* comercializa *Real Placer*, que también se puede encontrar como simplemente *Real*.

- En casos así para identificar el *software,* generalmente el producto viene acompañado de su correspondiente versión. Ejemplo:
    - *Oracle* 10g
    - *Netscape* 8
    - *Real Player* 10.5

- Es necesaria la creación de una lista de verbos y frases que funcionen como disparadores (*trigger words*).
    - Lanzamiento.
    - Presentación.
    - Nuevo.
    - Actualización.

- Era necesario, aunque un poco más extenso y por lo tanto no se tomó en cuenta durante el análisis y desarrollo del sistema, elaborar un listado de palabras o frases que puedan invalidar un resultado como bueno.
    - Nuevo *hardware*.
    - Nuevo servicio.
    - Presenta *hardware*.

- Elaborar un listado de páginas confiables que puedan ser las principales suministradoras de noticias, ya que salió a relucir que no todas las páginas contienen información seria y muchas contienen datos inexactos, caducos o replican exactamente las publicaciones de otras páginas, así un intento consiste en elaborar un listado de fuentes seguras:

- http://diarioti.com/
- http://www.terra.es/tecnologia/
- http://mx.news.yahoo.com/

- La elaboración de una lista de fuentes seguras también facilitará la elaboración de los programas encargados de encontrar las fechas, el párrafo de lanzamiento, títulos, etc.

- El programa resultante deberá ser capaz de realizar búsquedas en sitios de internet o a través de los resultados arrojados por buscadores como *Google* mediante ciertas consultas.

- Se decidió reemplazar los caracteres "=" por "--" y los caracteres "&" por "__" dentro de las URL de la etiqueta "<link>" (en el etiquetado con xml) ya que causaban dificultad a la hora de abrir el documento XML en navegadores como *Explorer* o *FireFox*.

  Ejemplo:
  Antes
  - http://www.infobae.com/notas/nota.php?Idx=126188&IdxSeccion=100623

  Después
  - http://www.infobae.com/notas/nota.php?Idx--126188__IdxSeccion--100623

En cuanto al tamaño de las noticias, había algunas que eran muy pequeñas y se consideró que no proporcionaban la suficiente información como para poder ser tomadas en cuenta. Este tipo de discriminación se hizo al leer y observar la extensión del texto, con ello se evaluó rápidamente a través de un escaneo visual si el texto era digno de tomarse en cuenta o no. Computacionalmente la solución para descartar noticias como estas fue establecer un criterio de longitud mínima de la noticia a través del conteo de palabras por enunciado gráfico usados en el texto.

Pero ya al momento de la programación se omitió esta posibilidad ya que cuando una noticia desde su url se limpia de todas las etiquetas html, queda una gran cantidad de párrafos muy pequeños generados por la misma estructura html, por ejemplo la etiqueta "*title"* (título) en html normalmente es muy pequeña, así como la etiqueta "a" genera hipervínculos que normalmente no son muy grandes.

9.2. Datos estadísticos del análisis del etiquetado del corpus

Con el fin de identificar los nuevos lanzamientos de productos de *software*, y para los fines del etiquetado morfológico manual, se etiquetó solamente el tramo comprendido entre una letra mayúscula y un punto (ya sea seguido, aparte o final). Este tramo de párrafo se identificó en base a palabras que indicaran que se trataba de un nuevo lanzamiento de *software*, tales como "nuevo", "lanzar", "liberar", etc. y se va a llamar "Enunciado Gráfico". También se llamarán "Palabras Gráficas" a todo el conjunto de cadenas de caracteres (palabras) dentro del corpus.

No todas las noticias tienen todos los elementos que se decidió extraer en un principio, algunas carecen del fabricante, "compañía" (como se ha nombrado a la etiqueta de dicho elemento), versión o en algunos casos no se encuentra el nombre del producto en el cuerpo de la noticia (aunque a veces se puede localizar en el título, pero en algunas otras no se encuentra en algún lado dentro de la noticia), en estos casos se localiza la descripción del producto pero no el nombre.

Durante el etiquetado manual de las noticias se encontraron bastantes problemas, así como con varios casos para analizar, los cuales son enumerados a continuación:

1. La estructura más común en el 90% del corpus de noticias es:

- Título.
- Fecha de publicación de la noticia (aunque ésta puede aparecer en cualquier parte del cuerpo de la noticia).
- Breve introducción de la misma.
- Complemento y descripción del nuevo producto.

2. La mayoría de los verbos que se utilizan en la redacción de las noticias se encuentran conjugados en tercera persona en número singular.

3. La fecha de lanzamiento casi nunca se publica, o por lo menos no de una manera explícita, aunque en la mayoría de los casos cercanos al 90%, aparece alguna fecha de referencia (fecha de publicación de la noticia, fecha aproximada del lanzamiento, etc.).

4. Otro problema que se encontró, es el hecho de que hay párrafos completos en mayúsculas, lo que dificulta la diferencia con los nombres propios, especialmente de compañías desarrolladoras de *software* y sus productos (*Microsoft, Adobe, Visual Basic*, etc.).

5. La forma en la que el etiquetado va a distinguir un párrafo de otro es a través de la combinación:

"Letra mayúscula" "contenido de palabras" "punto" "No contenido de palabras"

es decir, todo lo que se encuentre entre una letra mayúscula (con o sin sangría) y un punto (y aparte o final).

6. Pero el problema surge cuando hay compañías o productos que en su nombre contienen un punto, tal es el caso de las tecnologías como *Visual .NET*, *Microsoft .Inc.*, o las mismas versiones de los productos, por ejemplo, *Gentoo Linux* versión 2006.0 o *Thunderbird* 1.5.

7. Pueden haber caracteres o símbolos especiales como ®, ©, etc., lo que también es difícil de identificar y etiquetar morfológicamente.

8. En un principio se consideró declarar en el dtd etiquetas como <pub> (empresa o portal *web* que hizo la publicación de la noticia) que posteriormente se fueron dejando de utilizar poco a poco, o como <fechaLanzamiento> que en el transcurso del análisis fue notorio que tal vez hubiera sido mejor dejarlo simplemente como <fecha>, ya que no siempre se publicaba la fecha exacta de lanzamiento.

9. Algunos títulos de las noticias no tienen punto y a aparte, por lo tanto, resultaba difícil establecer el final del párrafo de ese título.

10. Antes de iniciar el anuncio en concreto del lanzamiento de un producto, había ocasiones en que éste era precedido por un pequeño preámbulo del mismo, por ejemplo: "Con el fin de apoyar a los diseñadores de *software*, la mayor empresa desarrolladora del mundo, así como sus socios principales en este rubro han lanzado……."

11. Las faltas de ortografía no sólo son un problema al leer un documento, sino también al etiquetar, ya que una palabra mal escrita prácticamente no existe en el lexicón, por lo tanto no puede asociarse una etiqueta a la misma, como puede ser "nuesta" en lugar de "nuestra", aunque el razonamiento humano inmediatamente intuye que la palabra está mal escrita, mas no así un programa de computadora (a menos que se especifique con algoritmos muy complejos y bien diseñados).

12. Para identificar un producto o compañía de *software*, normalmente se podía hacer por medio de las palabras que empezaran con letras mayúsculas, pero el problema es que había productos que empezaban en minúsculas, como *uBrowser*.

13. El problema más grande fue sin duda el de ambigüedad, por ejemplo, cómo diferenciar entre las palabras "descarga" como sustantivo y como verbo, ya que la clasificación que tenga cambia completamente el significado de esa parte del párrafo.

A continuación se muestran las mayores cantidades de palabras que aparecieron dentro del párrafo que indicaba un lanzamiento (párrafo principal), en un corpus de 240 noticias, referentes o que indicaban nuevo *software*. Se debe tomar en cuenta que estas palabras se derivan de su lema, por ejemplo, puede aparecer "lanzar", "lanzamiento", "lanzado", "lanzará", etc.

Trigger Word	**Num. de veces que aparecen en el párrafo etiquetado manualmente**
lanzamiento	99 veces
nueva	87 (se puede asociar a otros sustantivos como versión, actualización, etc.)
anunció	77
disponible	58
presenta	28 (problemas de ambigüedad, ya que puede ser tanto verbo como sustantivo)
desarrollará	24 (problemas de ambigüedad, ya que puede ser tanto verbo como sustantivo)
beta	17
liberar	14
crear	10
acaba	9
mejoras	9
actualización	8
próxima	5
publicado	5
reciente	3
renovada	1

Las palabras antes mencionadas permitieron elaborar una lista de palabras clave (*trigger words*) a identificar dentro del texto, que si se unen entre sí formando oraciones (p ej., *Microsoft* **anunció** el **nuevo**

lanzamiento de su **versión beta** de *Windows Vista*) proveen grandes indicios de un nuevo *software* en el mercado.

El título de la noticia fue el párrafo donde mayormente indicaba un nuevo lanzamiento de *software*, ya que 198 títulos de noticias (de 240 analizados) presentaban alguna o varias de las palabras identificadoras de lanzamientos antes mencionadas.

La palabra "lanzamiento", como se puede observar, fue la que más veces apareció en la sección del párrafo que indicaba un nuevo *software*, pero también fue la que más veces apareció en el título, ya que apareció en 98 títulos, le siguen "nueva" con 76, "presenta" con 14 y "disponible" con 13. Aunque hay que mencionar que en menor grado aparecían otras palabras indicadoras como "publicar", "liberar" o "anunciar".

Otros indicadores de lanzamiento pueden ser los verbos conjugados en el tiempo futuro, los cuales pueden ser muy útiles debido a que expresan una pronta acción a realizar.

También sobresalió el hecho de que hubo algunas noticias donde aparecían dos lanzamientos diferentes de *software*, es decir, la noticia contenía dos nuevos productos.

Por otra parte sólo se encontraron 22 veces "hoy" y 19 palabras más entre las que destacan "recientemente" o "ahora" en lo que a adverbios de tiempo se refiere dentro de los 240 corpus analizados. Generalmente la fecha de lanzamiento aparece mediante adverbios de tiempo, en raras ocasiones, aparece un día exacto.

Referente a las fechas de lanzamiento del producto:
En total aparecieron 204 fechas dentro de las 240 noticias.
172 del total fueron fechas de publicación de la noticia (de las cuales la gran mayoría se presentaron en los primeros párrafos de la noticia y escasos 4 al final de la misma).
28 del total fueron fechas explícitas de lanzamiento en base a adverbios de tiempo ("hoy" y "recientemente").

Cabe destacar que el formato de las fechas es muy variado, como pueden ser los siguientes:

- dd / m / aaaa
- dd / mm / aaaa
- dd / mm / aa
- día de la semana / mes del año / aaaa

- mm / dd / aaaa
- aaaa / mm / dd
- día de la semana / mes del año / aa, etc.

Todos estos indicadores de lanzamiento se encuentran distribuidos de la siguiente manera:
151 en el primer párrafo de las noticias.
43 en el segundo párrafo.
11 en el tercer párrafo.
2 en el cuarto párrafo y
4 en el quinto párrafo.

Los párrafos con menor cantidad de caracteres fueron de doce y ocho palabras.

Se encontraron algunos casos en que los elementos buscados (como la compañía, nombre del producto y la versión) de la noticia se dividían en distintos párrafos dentro del texto. Pero una vez analizada la información de manera adecuada, se llegó a la conclusión de que se necesitaban extraer los enunciados gráficos en donde se encontrara el mayor número de indicadores (que por lo general era el primer párrafo).

9.3. Conclusiones del diseño y desarrollo del sistema "SEINS"

En la etapa de diseño, como se muestra en la última grafica de Gantt, se hicieron grandes cambios, lo que llevó a invertir más tiempo en esta etapa de lo que se tenia previsto, pero gracias a ello se obtuvieron mejores resultados. Es probable que un experto no hubiera incurrido en este tipo de demoras.

En la etapa de desarrollo se comenzó a trabajar sobre un primer diseño elaborado, aunque conforme se avanzó en el desarrollo fue notorio que las expresiones regulares establecidas eran demasiado generales, y como consecuencia de esto se decidió incrementarlas en número, tratando así de obtener mejores resultados, esto a su vez modificó el algoritmo de diseño, retrasando el desarrollo y la implementación, la ventaja fue que se decidió iniciar en paralelo la etapa de diseño y desarrollo.

En un principio se pensó incluir en el algoritmo solamente las expresiones regulares, pero conforme se fue avanzando en el desarrollo

se decidió incluir una etapa de *Ranking* en la que se asignó una calificación conforme a las palabras de lanzamiento (*trigger words)* y de tecnología, tratando de hacer diferencia de las noticias de lanzamiento de *software* y de las que sólo hacen alusión a otro. De esta manera aumentó la precisión. Esto también para evitar que regresara como resultado noticias de lanzamiento de productos de belleza, películas, electrodomésticos, etc.

Sólo se utilizaron para la construcción del algoritmo expresiones regulares, y es por eso que se obtienen bajos resultados de *recall* y *precision*, es posible que si en la elaboración del sistema de extracción de información se utilizaran recursos más sofisticados de búsqueda y desambiguación, se obtendrían resultados de búsqueda muy exactos con hasta un 95% de *precision* y *recall*.

Los sistemas de extracción de información como se planteó en un principio, proporcionan varias ventajas como son la reducción de tiempo en el análisis de un corpus, pues una vez construido el sistema a realizar, el análisis es cosa de minutos.

Los sistema de extracción de Información también son tema de estudio hoy en día, es decir que aún es posible que surjan nuevos algoritmos que ayuden a mejorar los resultados de éstos y tal vez también de una manera más sencilla. Además de que no debe parecer extraño que este tipo de sistemas en unos cuantos años se vuelvan herramientas básicas de los buscadores de internet que hasta ahora solo utilizan algoritmos de recuperación de información como base en sus sistemas de búsqueda.

9.4. Técnicas alternas a la extracción de información

¿Existe alguna otra solución, alterna a la extracción de información?

En realidad la extracción de información en textos no estructurados no es nueva, sin embargo los sistemas actuales de extracción de información no son del todo precisos por la complejidad que existe al momento de analizar el lenguaje natural y la forma en cómo está compuesto.

Existen varios métodos para resolver el problema de la obtención de información de un tema específico en textos no estructurados; como lo son el aprendizaje automático y la minería de textos, entre otros. En el caso de esta tesis la metodología que se utilizó para la extracción de

información fue realizar un análisis minucioso para obtener patrones mediante un etiquetado morfológico de las palabras para generar expresiones regulares a partir del corpus, de esta forma se pudo resolver el problema, sin embargo, como ya se ha mencionado, existen varias técnicas para la extracción de información que podrían mejorar los resultados.

De esta manera se observa la gran importancia que tienen los sistemas de extracción de información en la actualidad, ya que de no existir estos, muchas organizaciones gastarían mucho tiempo, recursos humanos y económicos en analizar la gran cantidad de textos y tratar de obtener la información necesaria para la toma de decisiones en la organización, así mismo se debería capturar la información obtenida para almacenarla y procesarla.

Aprendizaje automático de la extracción de información

Una de las principales desventajas de la tecnología de extracción de información es la escasa portabilidad que tiene con sistemas existentes a nuevos dominios e idiomas. En general, la portabilidad implica reajustar manualmente el conocimiento lingüístico que depende del dominio, por ejemplo: diccionarios, gramáticas, patrones de extracción entre otros. Desde finales de los noventa a la fecha las investigaciones se enfocan en el uso de métodos empíricos para automatizar y reducir el alto costo de la portabilidad, los esfuerzos se concentran principalmente en el uso de técnicas de aprendizaje automático para adquirir de forma automática los patrones de extracción útiles para tratar con un lenguaje y dominio particular, y que además es una de las tareas más costosas en el desarrollo del sistema.

9.5. Conclusiones generales

La necesidad de la información data desde tiempos de los chinos (ábaco) o egipcios en Asia y los incas en Sudamérica, pasando por la etapa de la Revolución Industrial con máquinas mecánicas para los cálculos, la era moderna con el manejo de computadoras y su rápido y constante desarrollo y seguirá siendo necesaria mientras exista el hombre.

A lo largo de la investigación realizada y dando respuesta a la hipótesis planteada se pudo concluir que se cumplió con ella al definir conceptualmente los Sistemas de Extracción de Información (SEI), investigar como se elaboran los mismos y lograr demostrar que son una herramienta con posibilidad de ser utilizada para optimizar los recursos

dentro de las organizaciones, específicamente con los datos no estructurados, pues proporcionan opciones de búsqueda que hasta ahora sólo han sido usados con fines de investigación, pero que como ya se demostró existe una gran variedad de organizaciones que pueden hacer uso de los SEI. Debido a que áreas como ventas, contabilidad y producción, entre otras, generan altos volúmenes de información, su análisis manual es cada vez más difícil por lo que en corporativos extensos ya es una necesidad el tratamiento automático de documentos en lenguaje natural.

Los SEI proporcionan varias ventajas como son la reducción de tiempo en el análisis de un corpus, es decir en los textos en los que se tiene información a extraer, pues una vez construido el SEI éste realiza el análisis en unos cuantos minutos. Otra gran ventaja que ofrecen los SEI, es que a partir del análisis y el reconocimiento de entidades, puede extraerse la información que se desea y mostrarla en plantillas. Algo que se demostró y aplicó es que esta información puede introducirse en una base de datos y lograr así tener una estructura específica sobre ella.

Los Sistemas de Extracción de Información, son tema de estudio hoy en día, es decir que aún es posible el hecho de que surjan nuevos algoritmos que ayuden a mejorar los resultados de estos sistemas y tal vez también de una manera más sencilla. Además no debe parecer extraño que este tipo de sistemas en unos cuantos años se vuelvan herramientas básicas de los buscadores de internet que hasta ahora solo utilizan sistemas de recuperación de información como base en sus algoritmos de búsqueda.

Los SEI requieren varios tipos de análisis a un corpus, entre ellos y de gran importancia para obtener un mejor *recall* y *precision* es el morfosintáctico. Este tipo de análisis proporciona bastante información sobre los datos a extraer, aunque el tipo de análisis que se debe usar en un SEI es criterio de quien lo realiza. Este tipo de análisis no es tan imprescindible como sí lo es el realizar el etiquetado y su ventaja es una comprensión más exacta del lenguaje. Hay técnicas de análisis más apegadas a las matemáticas en las cuales los textos son despojados de su carácter lingüístico y son tratados como sólo cadenas de caracteres, estas técnicas también tienen sus ventajas pero en cuestión de lenguaje el análisis morfosintáctico se acerca más a la compresión del mismo por parte de la computadora. Ejemplo de la premisa anterior es el concepto de palabras clave o *trigger words,* estas pueden ser sustantivos, verbos, adjetivos en algún determinado género, etc., pero el hecho de usarlos implica que previamente dicha palabra fue marcada (etiquetada) de

alguna manera para identificarla con sus atributos lingüísticos, lo que pone a la computadora en entendido de la función morfosintáctica de la palabra.

El corpus es fundamental en los SEI, a lo largo de este trabajo de investigación se hizo notorio que en realidad es indispensable y no se puede prescindir de él, ya sea que se arme de forma manual o que el SEI lo arme de forma automática. Los corpus son instrumentos lingüísticos indispensables para el desarrollo de sistemas de extracción de información, en las empresas gracias a la creciente automatización de los procesos de procesamiento e intercambio de información es relativamente sencilla la fase de armado de corpus para los desarrollos de Extracción de Información (EI) dentro de una compañía. Aunado a esto XML es otra herramienta que brinda gran versatilidad al procesamiento y reutilización de los corpus por su carácter multiplataforma y su facilidad para construir estructuras propias para ordenar el texto.

La frecuencia de las formas en el corpus es una forma de darse cuenta del tipo de redacción que tiene un corpus y comprobar que lo que en el área de la informática se conoce como texto libre o no estructurado, en realidad sí tiene una estructura sintáctica y gramatical de la cual se puede hacer uso para la extracción de información. Un ejemplo de esto son las palabras clave que como se mostró son un pilar para diferenciar cuáles textos formarán parte del corpus y cuáles no, es decir, qué textos forman parte del área a extraer bajo el supuesto en el que se decida armarse de forma automática, porque si se hace de forma manual sería muy fácil identificar las diferencias. Un usuario no necesita este tipo de palabras pero para un sistema o una aplicación son indispensables.

La arquitectura de los SEI consiste de varios niveles o capas y esta arquitectura es una metodología en sí para su construcción. Aunque es decisión de la persona responsable de realizarlo resolver si se harán todas las capas o se prescindirá de alguna de ellas.

Existen varios Algoritmos para la realización de los SEI, como son K-Vecinos cercanos, C4.5 y la clasificación de textos, cabe mencionar que en función del algoritmo que se decida usar serán generados los resultados, puede hacerse una combinación de algunos de ellos, en este caso se utilizaron las *regex* como principal algoritmo de solución. Dado que este método es uno de los más sencillos se obtuvieron bajos resultados al generarse el corpus de forma automática, aunque los resultados obtenidos en un corpus ya recopilado son muy buenos.

Gracias al trabajo de investigación de la tesis y al desarrollo de la aplicación que la fundamenta, también se ha concluido que el Procesamiento del Lenguaje Natural (PLN) es un área poco desarrollada en México debido a la poca investigación que existe, salvo algunos casos excepcionales como la UNAM o el IPN, pocas son las instituciones de educación superior que le dan interés a dicha área.

Esto se puede constatar con base en que la mayor parte de las fuentes para la elaboración de la tesis son del extranjero, especialmente de Estados Unidos e Inglaterra, y la poca información existente en español proviene de España, muy ligada a Europa por su misma ubicación geográfica, y también porque las herramientas utilizadas son foráneas, como el lexicón del idioma español. España cuenta con bastante investigación en PLN, lingüística computacional e ingeniería lingüística gracias al apoyo que le da a estas áreas la Unión Europea.

A lo largo de esta investigación sirvió de gran utilidad la búsqueda de indicadores de lanzamientos (disparadores) en fragmentos de textos, además se reflexionó sobre el hecho de que la tarea de identificar indicadores clave de manera manual resulta complicada y tardada, pues se requiere tokenizar (dividir palabra por palabra), etiquetar (identificar la estructura morfológica de cada palabra), identificar en dónde se encuentra el fragmento con el que hay que trabajar (dónde está la información de mayor interés) y a continuación generar una plantilla para unir la información final, por lo cual se concluye que hacerlo de manera automática resulta mucho más útil, pues es más rápido y fácil para el ser humano en general.

A pesar que el término extracción de información puede parecer un concepto nuevo no lo es, pues tiene sus comienzos en los años sesenta, aunque su auge comienza a partir de los ochentas y noventas debido al aumento en las capacidades de equipos de cómputo, además de que las necesidades de información también aumentaron y fue importante su desarrollo, pues los procesos manuales eran lentos.

La extracción de información en la investigación en términos generales ha trascendido profundamente tanto que la Agencia de Defensa de los Estados Unidos (DARPA, *Defense Advanced Research Projects Agency*) ha patrocinado siete conferencias de comprensión de mensajes, conocidas como MUC (*Message Understanding Conferences*). Estas conferencias tienen una gran importancia en los SEI, tanto así que investigadores y científicos de todo el mundo acordaron intercambiar conocimiento con respecto a sus trabajos sobre extracción de

información. Debido a estas conferencias se han desarrollado reglas específicas las cuales sirvieron de guía, pues al haber varias organizaciones interesadas en ellos y al haber un lugar de discusión en donde estos sistemas se puedan retroalimentar mediante la competencia, el intercambio de ideas, sugerencias y evaluación constante, se lograron mejores métodos, algoritmos y sistemas.

Se puede definir a las MUC (*Message Understanding Conferences*) -véase cap. 6.2- como el inicio de los sistemas de extracción de información y como el inicio de la creación de reglas que se deben seguir para elaborar los mismos, haciendo una analogía serían algo así como la creación del modelo OSI (Modelo de Referencia de Interconexión de Sistemas Abiertos) para las comunicaciones de datos hecha por la ISO[58] (Organización Internacional de Estándares) o a diferencia de la clonación humana, la cual ya ha sido desarrollada en varias universidades del mundo pero todavía no hay una ley que limite o permita la manipulación genética. Por lo anterior es posible asegurar que el desarrollo de los SEI comenzó plenamente en los años ochenta con la creación formal de dichas conferencias.

En la actualidad, se comprobó mediante el desarrollo de la tesis y la aplicación, que la mayoría de las empresas tienen un concepto sobre el PLN relacionado únicamente a la investigación científica y no como una herramienta competitiva útil. La razón por la que se pudo corroborar esto es que los integrantes del equipo de tesis fuimos en algunas ocasiones a entrevistas de trabajo, y al mencionarle a los entrevistadores la opción de titulación consistente en una tesis respaldada por una aplicación en java mediante PLN, expresiones regulares e Ingeniería Lingüística, les asombró bastante, ocasionando que preguntaran más sobre el tema ya que no les era tan clara la idea de que se pudiera hacer eso con dichas herramientas.

El PLN es una rama de la Inteligencia Artificial que se encarga de producir sistemas informáticos -véase cap. 2.2.1- que posibiliten la comunicación hombre-máquina por medio de la voz o del texto a través del lenguaje humano, de ahí su relación con el desarrollo de la tesis de investigación y la aplicación, ya que el objetivo de ésta última es fundamentar la teoría de la primera, que a su vez tiene como apoyo el PLN.

---

58 ISO es una organización no gubernamental de los institutos nacionales de estándares de 157 países bajo la consigna de un miembro por país, con una Secretaría Central en Ginebra, Suiza que coordina todo el sistema.

Los recursos que maneja el PLN -véase cap. 2.2.3- son los utilizados para el desarrollo de la aplicación, un etiquetador, un *parser* (en este caso desarrollado en java), la lista de *triger-words* definida por las palabras clave de tecnologías de información y de nuevos lanzamientos de software , lexicones y *gazetteers*, lematizadores y analizadores morfológicos.

En ésta tesis se destacó la diferencia entre la Recuperación de Información (RI) y la EI debido a que en la primera sólo se obtienen textos con los patrones dados, como sucede en el buscador *Google*. Se está más familiarizado con dicho término (RI) que con el de extracción de información, debido a que éste último se refiere al hecho de obtener información específica de los textos. La razón de elegir lanzamientos de *software* es porque en algunas organizaciones como los bancos, son trascendentales las operaciones para el manejo de dinero, por lo tanto necesitan tener programas y sistemas robustos tanto en proceso como en seguridad además de estar a la vanguardia en tecnología y eficiencia. Por esto, a los altos directivos les es muy útil estar informados acerca del nuevo *software* lanzado periódicamente para tener actualizados sus sistemas y funcionen adecuadamente en tiempo y forma para la correcta toma de decisiones.

A pesar de que esto parece sencillo no es una tarea fácil, pues no obstante que estamos relacionados con el lenguaje (se utiliza diariamente) la extracción de información en textos no estructurados es muy compleja, porque cada ser humano utiliza el lenguaje de acuerdo a su contexto, el cual influye en la manera de expresar sus ideas y por eso se dice que la composición del lenguaje es infinita. Por lo anterior es difícil encontrar patrones lingüísticos que ayuden a identificar los elementos que se deben tomar en cuenta para descubrir los indicadores que pudieran ayudar a obtener las entidades de las noticias de lanzamientos de nuevo *software* como se mencionó anteriormente: "compañía", "producto" y "versión".

Por eso se resolvió que la extracción de información se encuentra dentro del procesamiento del lenguaje natural, debido a que se estudia o simula un comportamiento del hombre, en este caso el lenguaje. Al ser parte de la inteligencia artificial es más complejo y en caso de no realizarse este tipo de investigaciones, el personal encargado de leer y obtener la información deseada invertiría mucho más tiempo y esfuerzo.

Las herramientas utilizadas para el desarrollo de esta investigación han servido para llevar a cabo un análisis minucioso para obtener "disparadores" en los fragmentos de texto, pues al identificar éstos, se

pudo distinguir los perfiles específicos de las noticias de nuevo *software*, al identificar estos fragmentos y aplicarles herramientas de la lingüística se pudo obtener información significativa en las noticias, como identificar que *Microsoft* es una compañía de *software* y no un nombre de persona o un objeto.

Gracias a las investigaciones realizadas previamente otra de las conclusiones surgidas fue la necesidad de definir previamente lo que se buscaría en las noticias, en este caso la compañía que lanzó el *software*, el nombre del mismo y su versión. Estos elementos fueron localizados mediante el estudio del contexto. Los elementos que se deben buscar forman las plantillas que deben ser llenadas mediante los elementos encontrados en la noticia, los cuales van desde palabras hasta sentencias.

La extracción de información consiste en analizar la estructura interna de un texto mediante búsqueda de patrones, algoritmos de aprendizaje, comparaciones semánticas (para definir el tema donde se ubique el texto, por ejemplo deportes, política, cine, etc.) y eliminar información no útil para desplegar sólo la información que sea de interés, a diferencia de la recuperación de información donde únicamente se extraen los textos con las palabras que sirvieron de criterios de búsqueda. Con esto también se ha concluido que en una primera instancia la RI es de cierta manera útil y es el primer paso para solucionar el problema de análisis de textos, pero si se desea una salida más exacta se tendría que hacer uso de la EI (y a su vez del PLN y de la ingeniería lingüística), pero en caso de que todavía se deseara una salida mucho más exacta, entonces procedería el uso de algoritmos matemáticos muy complejos sin escala a errores, métodos de aprendizaje continuo, máquinas y autómatas de estados finitos

Uno de los problemas más grandes a resolver fue el de la ambigüedad, ya que el lenguaje hablado humano es tan extenso que hay palabras que dependiendo de su ortografía y semántica tienen uno u otro significado, pero lo que de cierta manera sirvió para considerar lo anterior es la misma ortografía, ya que por ejemplo palabras como "el" y "él" se diferencian por su acento ortográfico; el primero hace referencia al artículo determinado masculino singular y el segundo al pronombre personal. Algo con que no cuenta nuestro análisis es el "reconocimiento del habla", donde los patrones a comparar son frecuencias de sonidos y únicamente se puede hacer diferencia de las palabras para disminuir la ambigüedad en base a la entonación de las mismas y obviamente no en su ortografía, algo que es mucho más difícil

y requiere mayor investigación, mejores herramientas y mucho más tiempo.

Según Dan Sullivan (2001) los textos en los memos, historietas, noticias, etc. tienen una estructura sintáctica, lingüística y ortográfica, y en eso todo mundo está de acuerdo porque normalmente un enunciado se forma de sujeto, verbo y predicado y cada uno de estos tiene otros elementos que los conforman, pero el problema es cuando esta estructura no se respeta por diversos factores como una mala redacción por parte del escritor o faltas de ortografía, para lo cual ningún sistema está preparado en la actualidad, es decir, ningún sistema va a corregir un estilo de redacción, pero se puede llegar a esto en el futuro.

A pesar de que no se desarrolló un sistema informático, sino sólo una aplicación, el enfoque metodológico que se utilizó fue básicamente el Ciclo de vida clásico de un sistema debido a la definición de los modelos de requerimientos, análisis, diseño, desarrollo, pruebas e implementación, aunque cabe decir que se utilizaron los diagramas clásicos del modelo estructurado como los diagramas de flujo o las gráficas de Gantt y un lenguaje orientado a objetos, tal es el caso de Java. Por tal motivo también se mostró que en la actualidad es muy difícil seguir la rigidez de un solo modelo, ya que muchas veces implícitamente se ocupan fracciones del otro modelo (orientado a objetos o estructurado). Simplificando lo anterior se podría decir que para construir un edificio se necesitan en primer instancia ingenieros y arquitectos, pero dicen por ahí que cuando un edificio lo hace un arquitecto "solito se cae", y cuando lo hace un ingeniero se debe derrumbar de lo feo que queda, por lo tanto para terminar la obra se necesitan de ambos conocimientos, de lo contrario surgen resultados no esperados o se hace mas difícil el camino para lograrlo.

Hoy en día es imposible encontrar sistemas de extracción de información que arrojen resultados al 100% de exactitud, para muestra basta un botón que se explicará a continuación.

Annie (*A Nearly-New IE System*) es uno de los muchos sistemas de EI desarrollado con una tecnología llamada GATE (*General Architecture for Text Engineering*). Fue diseñado para ser un SEI portable (utilizable en distintas aplicaciones con diferentes tipos de textos y cada uno con distintos propósitos) y sus principales implicaciones son:

- Trabajar sobre documentos con distintos formatos, desde mensajes con faltas de ortografía distintivos en una redacción de correo electrónico hasta estructuras como XML y HTML.

- El sistema debe ser capaz de procesar grandes volúmenes de datos sin interrupción y a alta velocidad. Esto significa que debe escalar desde relativamente pequeñas computadoras personales corriendo sistemas operativos de escritorio hasta supercomputadoras corriendo procesos paralelos.
- Los desarrolladores del sistema pueden adaptar el mismo a nuevas circunstancias con un esfuerzo mínimo.
- Los datos en múltiples lenguajes alrededor del mundo también deben ser procesados. Este problema incluye la edición y publicación de diversos caracteres y la conversión de diversas codificaciones en *Unicode* (web.90).

Annie utiliza algoritmos de estados finitos y un lenguaje llamado Jape[59]. Para demostrar la afirmación anterior se utilizó un demo de *Annie* publicado en internet que se encarga de mostrar un reconocimiento de entidades en textos.

Dicho demo utiliza un conjunto predeterminado de componentes y recursos de EI y por lo tanto su alcance puede variar. Así mismo estructuras complejas de html pueden evitar que el sistema analice el texto que contiene (como tablas, marcos o formularios). Para utilizar el demo simplemente basta insertar la dirección URL de la página *web* a examinar dentro de un cuadro de texto como se muestra en la siguiente figura.

59 *Java Annotation Patterns Engine*.- *JAPE* es un lenguaje de comparación con modelos. El LHS (*Left Hand Side*) de cada norma contiene modelos que deben cumplirse y el RHS (*Right Hand Side*) contiene detalles de anotaciones (y excepcionalmente características) para ser creadas. Por ejemplo, una regla puede reconocer un nombre de pila (del módulo de un *gazetteer*) seguido por un nombre propio (del *POS tagger*) y anotar este modelo como una persona.

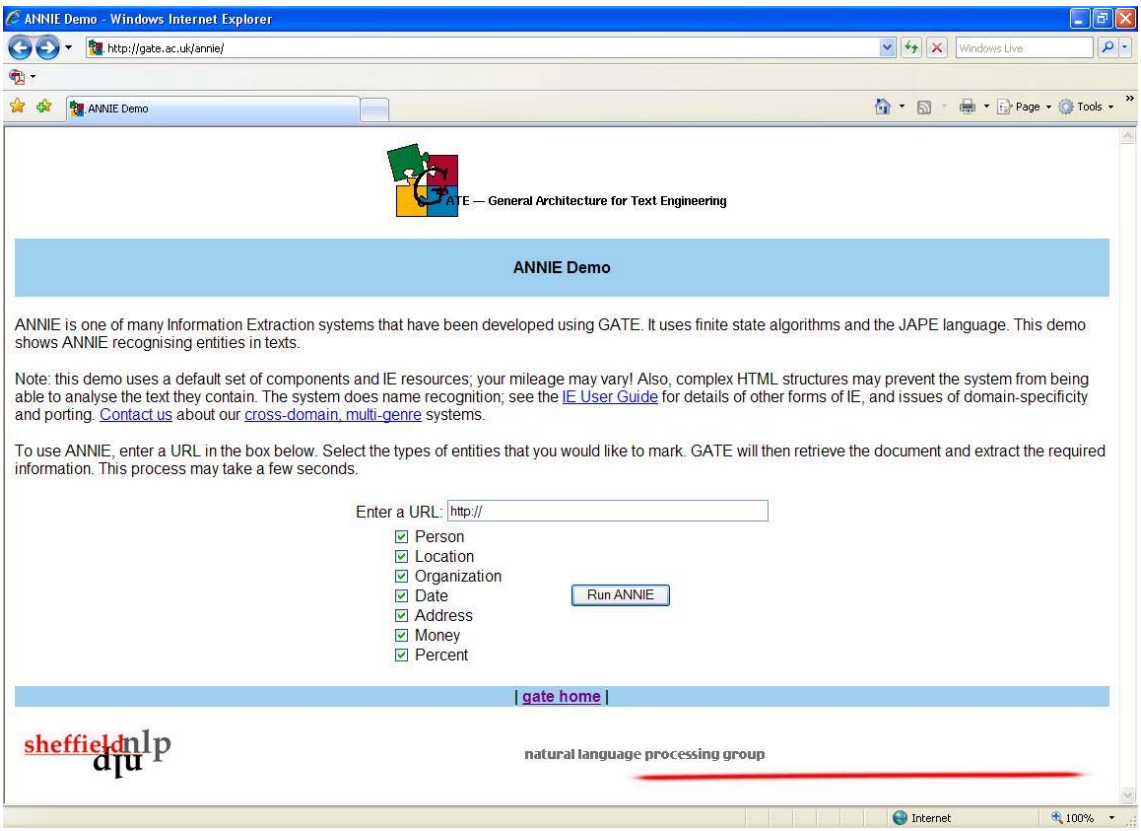
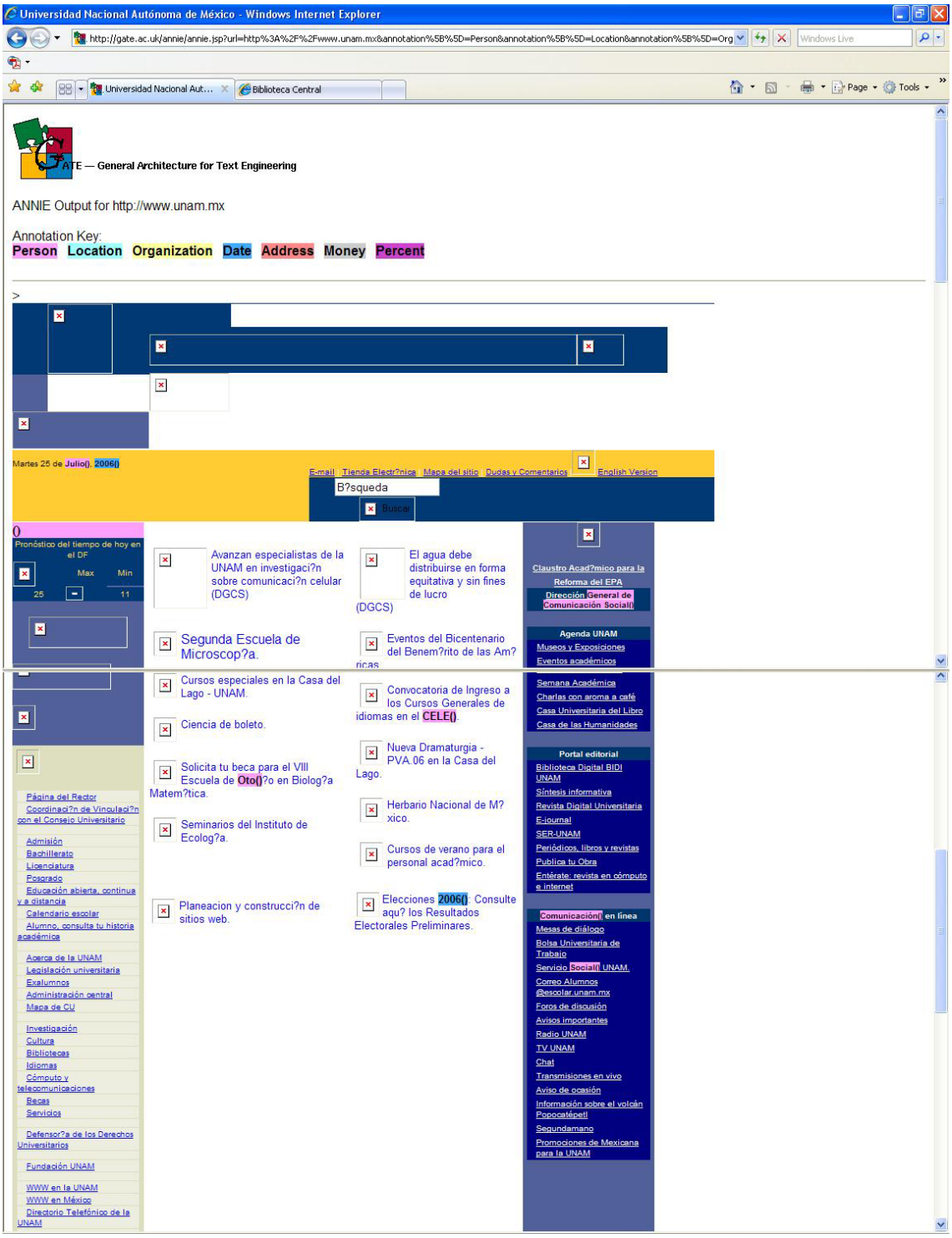
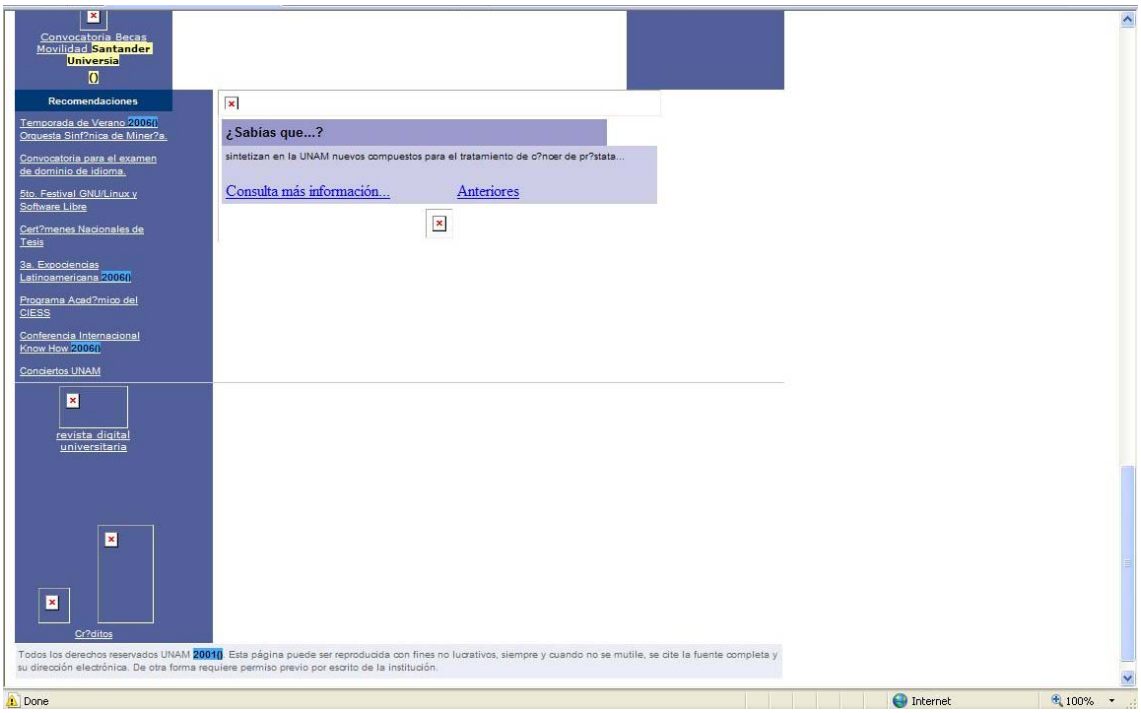


Figura 9.1.- Página *web* donde se ejecuta el programa *Annie*.

En el sistema a ejecutarse se tienen varias opciones para ser reconocidas, como personas, lugares, organizaciones, fechas, direcciones, cantidades de dinero y porcentajes.

Una vez ejecutado en la página http://www.unam.mx los resultados arrojados se muestran a continuación


Universidad Nacional Autónoma de México - Windows Internet Explorer
ATE — General Architecture for Text Engineering
ANNIE Output for http://www.unam.mx
Annotation Key:
Person Location Organization Date Address Money Percent
Martes 25 de Julio(), 2006()
E-mail | Tienda Electr?nica | Mapa del sitio | Dudas y Comentarios | English Version
B?squeda
Buscar
Pronóstico del tiempo de hoy en el DF
Max Min
25 11
Avanzan especialistas de la UNAM en investigaci?n sobre comunicaci?n celular (DGCS)
El agua debe distribuirse en forma equitativa y sin fines de lucro (DGCS)
Segunda Escuela de Microscop?a.
Eventos del Bicentenario del Benem?rito de las Am?ricas
Claustro Acad?mico para la Reforma del EPA
Dirección General de Comunicación Social()
Agenda UNAM
Museos y Exposiciones
Eventos académicos
Cursos especiales en la Casa del Lago - UNAM.
Convocatoria de Ingreso a los Cursos Generales de idiomas en el CELE().
Ciencia de boleto.
Nueva Dramaturgia - PVA 06 en la Casa del Lago.
Solicita tu beca para el VIII Escuela de Oto()?o en Biolog?a Matem?tica.
Herbario Nacional de M?xico.
Seminarios del Instituto de Ecolog?a.
Cursos de verano para el personal acad?mico.
Planeacion y construcci?n de sitios web.
Elecciones 2006(). Consulte aqu? los Resultados Electorales Preliminares.
Semana Académica
Charlas con aroma a café
Casa Universitaria del Libro
Casa de las Humanidades
Portal editorial
Biblioteca Digital BIDI UNAM
Síntesis informativa
Revista Digital Universitaria
E-journal
SER-UNAM
Periódicos, libros y revistas
Publica tu Obra
Entérate: revista en cómputo e internet
Comunicación() en línea
Mesas de diálogo
Bolsa Universitaria de Trabajo
Servicio Social() UNAM.
Correo Alumnos @escolar.unam.mx
Foros de discusión
Avisos importantes
Radio UNAM
TV UNAM
Chat
Transmisiones en vivo
Aviso de ocasión
Información sobre el volcán Popocatépetl
Segundamano
Promociones de Mexicana para la UNAM
Página del Rector
Coordinaci?n de Vinculaci?n con el Consejo Universitario
Admisión
Bachillerato
Licenciatura
Posgrado
Educación abierta, continua y a distancia
Calendario escolar
Alumno, consulta tu historia académica
Acerca de la UNAM
Legislación universitaria
Exalumnos
Administración central
Mapa de CU
Investigación
Cultura
Bibliotecas
Idiomas
Cómputo y telecomunicaciones
Becas
Servicios
Defensor?a de los Derechos Universitarios
Fundación UNAM
WWW en la UNAM
WWW en México
Directorio Telefónico de la UNAM



Como se puede observar las personas las identifica de color rosa, los lugares los marca en color verde, las organizaciones de amarillo, las fechas de color azul, las direcciones de anaranjado y las cantidades de dinero y porcentajes de color gris y morado respectivamente.

Algunos errores que son posibles corroborar son que marca "Julio" como persona, pero teniendo en cuenta que también puede ser el nombre de una persona podría estar en lo correcto, aunque este no es el caso (parte superior izquierda). No obstante hay aciertos que identifica perfectamente que son "2006" (fecha), "General de Comunicación Social" o "CELE", (personas) y "Santander Universia" (que se podría tomar como organización) sin embargo hay otros como "UNAM" que debería reconocer al instante y no lo hizo. También es notorio que no reconoce caracteres especiales como los acentos y las letras "ñ", obviamente no toma en cuenta tampoco las imágenes.

Para confirmar lo anterior se puede ejecutar *Annie* desde el *link* http://gate.ac.uk/annie/ y escribiendo http://www.unam.mx/ en su respectivo cuadro de texto.

Por lo visto anteriormente queda demostrado que no hay un SEI exacto en todos sus sentidos. *Annie* es un sistema desarrollado por un grupo de aproximadamente quince investigadores del "Grupo de Procesamiento de Lenguaje Natural" encabezado por el científico

investigador en Ciencias de la Computación de la Universidad de *Sheffield*; Hamish Cunningham.

Se decidió desarrollar una aplicación para fundamentar la tesis debido a que pareció importante el hecho de que se pudiera hacer extracción de información con Ingeniería Lingüística y con PLN, además de que era todo un reto. También se decidió que ésta fuera hecha en Java por la gran funcionalidad de clases y funciones que vienen en el *kit* y que permiten trabajar fácilmente con la manipulación de los datos, como es la clase *stringtokenizer* (permite dividir cadenas en subcadenas a través de ciertos parámetros).

La aplicación desarrollada seria clasificada como un sistema de información gerencial –véase cap. 2.1.3.5- debido a que se enfoca hacia los gerentes y los datos que muestra sólo influyen una vez, siendo después inservibles.

La identificación de entidades también se hizo presente en la etapa de desarrollo del sistema, aunque no en base a reglas gramaticales ni factores semánticos –véase cap. 2.7.1- sino más bien en base a un *gazetteer* de compañías de *software*, con lo cual se redujo en gran medida el número de entidades no encontradas en el lexicón (etiquetadas con "X") y permitió definir a las posibles empresas dentro de las expresiones regulares.

Se estudiaron y utilizaron principalmente tres de los cinco niveles de estudios del lenguaje –véase cap. 2.3.3.3- que son Morfología, Sintaxis y Semántica.

El etiquetado de las palabras primero para el análisis manual y posteriormente para el diseño de la aplicación consistió en el etiquetado de las partes de la oración (POST) a través de un lexicón, el tipo de etiquetado en el análisis manual y en el proceso del sistema fueron el Gramatical y el Morfológico –véase cap. 2.5.2.4-, mientras que el reconocimiento de entidades nombradas estuvo a cargo de la comparación de las palabras dentro de un *gazetteer.*

También se siguieron con mucha exactitud los principios básicos de un etiquetado automático de corpus –véase cap. 2.5.2.7- ya que se puede recuperar el corpus original eliminando las etiquetas y almacenándolas en una ubicación alterna, es posible tener acceso al listado de las etiquetas a través del lexicón utilizado, la información sobre el etiquetado del corpus se encuentra definido en el código fuente

del programa y el lingüista que revisó dicho proceso fue el propio asesor.

El lexicón del idioma español utilizado para el proceso de lematización en el desarrollo de la aplicación también sigue una serie de algoritmos explicados en los capítulos 2.5.3.2 y 2.5.3.3 consistentes en la definición de los árboles de letras (*tries*) porque como se puede observar en la figura 7.10, cada raíz de la palabra (letra inicial) va tomando un nodo distinto de forma sucesiva hasta formar la palabra completa, lo que da como resultado la formas de cada palabra (singular o plural y masculino o femenino, etc.). Para una mejor referencia véase figura 2.5 dentro del cap. 2.5.3.3.

Como nota adicional, el lexicón no fue creado dentro del proceso del desarrollo del sistema, sino que se utilizó uno ya elaborado por un investigador del Centro de Investigación en Computación (CIC) del IPN. Pero lo que sí se elaboró fue el algoritmo para el proceso de lematización. En este caso el proceso de lematización no tuvo como causa la reducción del corpus, sino simplemente identificar las palabras de acuerdo a su categoría sintáctica para el posterior manejo y comparación con las expresiones regulares.

Las diferencias que se pudieron establecer entre *data mining*, *text mining* y extracción de información consisten en que la primera hace consultas a estructuras de datos definidas como son las bases de datos, produciendo una salida de las relaciones semánticas entre tablas que pudieran existir y reuniendo las ventajas de varias áreas como la estadística, la inteligencia artificial, la computación gráfica, las bases de datos y el procesamiento de datos, principalmente usando como materia prima estas últimas. El text mining consiste en encontrar información interesante (no trivial, oculta, desconocida anteriormente y potencialmente útil) en grandes conjuntos de datos textuales. Es encontrar información semántica y abstracta en la forma más superficial de los datos textuales. Como hurgar dentro del texto para sacar la información más útil. La extracción de información tiene exactamente el mismo fin que el text mining.

No solo se creó una aplicación que realiza extracción de información sobre el internet, sino que también se desarrollaron implícitamente pequeños sistemas derivados del principal como puede ser un "limpiador de etiquetas de html", un sistema "comparador de expresiones regulares" o un tokenizador.

Los SEI son muy distintos al desarrollo de sistemas tradicionales que se conocen como un sistema de altas bajas y cambios (el registro a un concurso por internet o un sistema de ventas), ya que en los primeros se requiere constante investigación lingüística e informática además de utilizar asiduamente recursos de la Inteligencia Artificial, mientras que en los segundos su alcance es finito (se tiene un principio y un fin dentro de la metodología para elaborarlos) y los recursos que se utilizan están al alcance de cualquier persona (compiladores, bases de datos, navegador de internet, etc.). Para analizar textos libres es difícil trabajar con autómatas de estados finitos que realicen sus labores con la mayor rigurosidad de procedimiento con respecto a las partes de la oración, ya que el lenguaje natural aunque es regido por reglas gramaticales, éstas son tan abiertas que es muy difícil encontrar textos elaborados bajo estructuras de oraciones ya definidas.

Para concluir y en forma de despedida, los procesos de investigación y elaboración de la tesis así como de la aplicación nos proporcionaron la satisfacción de haber contribuido al desarrollo de la investigación en los campos de la Lingüística y PLN dentro de la UNAM. También nos ayudaron a formar un compromiso más estricto de cumplimiento con nuestras labores en la tesis, puntualidad, indagación en los temas que teníamos que estudiar e investigar y de indagar cómo solucionar el problema de extraer información en textos con datos no estructurados, es decir, la formación de un carácter formal y útil a la sociedad.

Anexos

10. Anexos

Anexo A

DTD final para el etiquetado manual a través de xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT corpus (noticia)>
<!ELEMENT noticia (titulo | link | producto | compania | version | fechaLanzamiento | pub)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT producto (#PCDATA) >
	<!ATTLIST producto tipo CDATA #IMPLIED >
<!ELEMENT compania (#PCDATA)>
	<!ATTLIST compania tipo CDATA #IMPLIED >
<!ELEMENT fechaLanzamiento (#PCDATA)>
	<!ATTLIST fechaLanzamiento tipo CDATA #IMPLIED >
<!ELEMENT pub (#PCDATA)>
<!ELEMENT version (#PCDATA)>
	<!ATTLIST version tipo CDATA #IMPLIED >
```

Índice Alfabético

Índice alfabético

S

T

U

V

W

X

Bibliografía

11. Bibliografía

Fuentes bibliográficas:

ALONSO PARDO, MIGUEL A. 2000. Interpretación tabular de autómatas para lenguajes de adjunción de árboles. Universidade da Coruña. Documento electrónico en web.84

ARRARTE CARRIQUIRY, GERARDO. 1995. "Tendencias actuales de la ingeniería lingüística en Europa", en *II Jornadas de Informática e Investigación Operativa*.

ARREGI O, E I. FERNÁNDEZ, Clasificación de documentos escritos en euskara: impacto de la lematización. Universidad del País Vasco Documento electrónico en web.35

BERTALANFFY, LUDWIG VON Y A. ROSS, G. M. WEINBERG. 1987. Tendencias en la Teoría de Sistemas, Madrid: Alianza Universidad.

BLANCO ENCINOSA L. E I. GUTSZTAT GUTSZTAT. 1997. Sistemas informáticos. Teoría, métodos de elaboración, técnicas, herramientas. Tomo I, México: IPN.

BOLSHAKOV, IGOR Y A. GELBUKH. 2004. Computational Linguistics. Models, Resources, Applications, México: IPN-UNAM-FCE.

Booch, Grady y J. Rumbaugh, I. Jacobson. 1999. El Lenguaje Unificado de Modelado, México: Addison Wesley.

BORDONI LUCIANA Y E. D'AVANZO, The IPTS Report, Perspectivas para la integración de la minería de textos y la gestión del conocimiento: ENEA. Documento electrónico en web.11

BURCH, G. JOHN, G. GRUDNITSKI. 1994. "1 El recurso de la Información" en *Diseño de Sistemas de Información Teoría y Práctica*, México.

CARBONELL, JAIME. 1992. El procesamiento del lenguaje natural, tecnología en transición**.** Carnegie Mellon University. Documento electrónico en web.77

COWIE JIM Y L. GUTHRIE, W. JIN, R. WANG, T. WAKAO, Description of the Diderot system as used for MUC-5. New Mexico State University Documento electrónico en web.15

DE LA REZA, GERMÁN. 2001. Teoría de Sistemas, México: Porrúa.

FEBLES RODRÍGUEZ, JUAN PEDRO Y A. GONZÁLEZ PÉREZ. 2002. Aplicación de la minería de datos en la bioinformática. ACIMED
Documento electrónico en web.37

GELBUKH, ALEXANDER Y G. SIDOROV. Analizador Morfológico Disponible: un Recurso Importante para PLN en Español. México: IPN.
Documento electrónico en web.93

GELBUKH, ALEXANDER Y G. SIDOROV. Approach to construction of automatic morphological analysis systems for inflective languages with little effort. México: IPN.
Documento electrónico en web.94

GRISHMAN, RALPH. 1986/1991. Introducción a la lingüística computacional, Madrid: Visor.

HAUSSER, ROLAND. 1999. Foundations of computacional linguistics: man machine communication in natural language, Berlín: Springer-Verlag.

JACKSON, METER E I. MOULINIER. 2002. Natural language processing for on line applications: text retrieval, extraction and categorization, Philadelphia: John Benjamins B.V.

JORDÁN, ÁNGEL G. 1992. Lenguas y tecnologías de la información. Carnegie Mellon University.
Documento electrónico en web.74

KAY MARTÍN. 2003. "1 Introduction", en *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford: Oxford University Press, pp. XVII-XX.

LEVINE GUTIÉRREZ, GUILLERMO. 2001. Computación y programación moderna. Perspectiva integral de la informática, México: Pearson Education.

LLIDO ESCRIVÁ, DOLORES MARÍA. 2002. Extracción y recuperación de información temporal. Escola Superior de Tecnologia y Ciencias Experimentals. Departament de Llenguatges i Sistemes Informàtics. Universitat Jaume I.
Documento electrónico en web.53

LLISTERRI, JOAQUIM. 2003. Lingüística y tecnología del lenguaje, Lynx. Panorámica de Estudios Lingüísticos (Departament de Teoria dels Llenguatges, Universitat de València) 2: 9-71
Documento electrónico en web.26

LLISTERRI, JOAQUIM. 2004. "Las tecnologías lingüísticas en España", en *El español en el mundo. Anuario del Instituto Cervantes 2004*, Madrid: Instituto Cervantes – Círculo de Lectores – Plaza & Janés, pp. 229-251.
En
http://cvc.cervantes.es/obref/anuario/anuario_04/llisterri/default.htm

LLISTERRI, JOAQUIM Y J. GARRIDO ALMIÑANA. 1998. "La Ingeniería Lingüística en España", en *El español del mundo. Anuario del Instituto Cervantes. 1998*, Madrid: Instituto Cervantes – Arco Libros, pp 299-391.
En
http://cvc.cervantes.es/obref/anuario/anuario_98/llisterri/

LUCAS, HENRY. 1984. Sistemas de Información, Madrid: Paraninfo.

MITKOV, RUSLAN (ed). 2003. The Oxford Handbook of Computational Linguistics, New York: Oxford University Press.

MONTÉS Y GÓMEZ, MANUEL. Minería de texto: Un nuevo reto computacional, México: IPN.
Documento electrónico en web.38

MORA, JOSÉ LUIS Y E. MOLINO. 1999. Introducción a la informática, México: Trillas.

MORENO BORONAT, LIDIA. 1999. "1 Preliminares del Lenguaje Natural", en *Introducción al Procesamiento del Lenguaje Natural*, Alicante: Universidad de Alicante.

MORENO BORONAT, LIDIA.1999. "2 Análisis Léxico", en *Introducción al Procesamiento del Lenguaje Natural*, Alicante: Universidad de Alicante.

MOURE, TERESA Y LLISTERRI, JOAQUIM. 1996. "Lenguaje y nuevas tecnologías. El campo de la lingüística computacional", en *Avances en lingüística aplicada*, M. Fernández Pérez (Coord.), Santiago de Compostela: Universidad de Santiago de Compostela, Servicio de Publicaciones e Intercambio Científico (Avances en, 4), pp. 147-228. En http://homepage.mac.com/joaquim_llisterri/publicacions/llisterri_moure_96.html

MURDICK, ROBERT Y J. MUNSON. 1988. Sistemas de información administrativa, México: Prentice Hall.

O'LEARY, TIMOTHY Y L. O'LEARY. 1997. Computación Básica, México: McGraw Hill.

PAZ GARCÍA, INGRID. Ensayo de un sistema de extracción de información (técnica de inteligencia artificial) en un centro de información especializado en sanidad vegetal, Cuba: Dirección Nacional Gaviota S.A.

PEREZ GUERRA, JAVIER. 1998. Introducción a la lingüística de corpus: un ejercicio con herramientas informáticas aplicadas al análisis textual, Santiago de Compostela: Torculo Editions.

PÉREZ HERNÁNDEZ, M. CHANTAL. 2002. Explotación de los córpora textuales informatizados para la creación de bases de datos terminológicas basadas en el conocimiento. Volumen 18, Universidad de Málaga.
Documento electrónico en web.14 y web.17

PIVAN, JOHN. 1981. Teoría de Sistemas Aplicada, México: Trillas.

PRIETO, ALBERTO Y A. LLORIS. 2002. Introducción a la Informática, Madrid: Mc Graw Hill.

SANTANA SUÁREZ, OCTAVIO Y Z. HERNÁNDEZ FIGUEROA, G. RODRÍGUEZ RODRIGUEZ, L. LOSADA GARCÍA. 2005. Una herramienta de recuperación morfoléxica aplicada a *Microsoft Word*, Universidad de Las Palmas de Gran Canaria.
Documento electrónico en web.04

SENN, JAMES A. 1992. Análisis y Diseño de Sistemas de Información, México: McGrawHill.

SENN, JAMES A. 1990. Sistemas de Información para la Administración, Georgia: Grupo Editorial Iberoamericana.

STEVENS, P. Y R. POOLEY. 2003. Utilización de UML en la Ingeniería del Software con Objetos y Componentes, España: Pearson Education S. A.

SULLIVAN, DAN. 2001. "2 Understading the Structure of Text: The Foundation of Text-Based Business Intelligence" en *Document WareHouse and Text Mining*, New York: Wiley.

TAMAYO, MARIO. 1990. Metodología Formal de la Investigación científica, México: Limusa.

TÉLLEZ VALERO, ALBERTO. 2005. Extracción de Información con Algoritmos de Clasificación, Tonantzintla Puebla: INAOE.
Documento electrónico en web.39
USZKOREIT, HANS. 1996. What is Computational Linguistics. Computational Linguistics Department, University of the Saarland.
En http://www.coli.uni-sb.de/~hansu/what_is_cl.html

VOUTILAINEN ATRO. 2003. "2 Processes, Methods and resources, cap. 11 Part-of-speech Tagging", en *The Oxford Handbook of Computational Linguistics*, Ruslan Mitkov (ed.), Oxford: Oxford University Press, pp. 220-221.

WHITTEN, JEFFREY Y L. BENTLEY, V. BARLOW. 2003. Análisis y diseño de sistemas de información, México: McGraw Hill.

Fuentes de internet:

web.01
Universidad Tecnológica de la Mixteca. Sistemas de Información.
http://mixteco.utm.mx/~mmoreno/PAD/pa2.pdf#search=sistemas%20de%20informacion-21/09/2005
Fecha de consulta a dicha página para la elaboración de la tesis: 02/02/2006 (formato – dd/mm/aaaa-)

web.02
The open archive for Library and Information Science.
Ensayo de un sistema de extracción de información (técnica de inteligencia artificial) en un centro de información especializado en sanidad vegetal. Cuba
http://eprints.rclis.org/archive/00003017/01/2004_27.pdf 12/11/2005

web.03
Grupo de Estructuras de Datos y Lingüística Computacional. Estado del arte de las MUC
http://www.gedlc.ulpgc.es/docencia/seminarios/pln/Extraccion_de_informacion/tsld009.htm
15/11/2005

web.04
Grupo de Estructuras de Datos y Lingüística Computacional.
http://www.gedlc.ulpgc.es/art_ps/art47.pdf
7/02/2005

web.05
GPLSI. Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.
http://64.233.187.104/search?q=cache:U3OxvATpCgcJ:gplsi.dlsi.ua.es/gplsi/articulos/a1998/novatica.ps+MUC-3&hl=es&lr=lang_es
16/11/2005

web.06
GPLSI. Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. Departamento de Lenguajes y Sistemas Informáticos. Extracción de información. Universidad de Alicante
http://gplsi.dlsi.ua.es/index.php?opc=211

web.07
GPLSI. Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante.
http://gplsi.dlsi.ua.es/gplsi/publicaciones/trabajosInves/Munoz_ti.ps
16/11/2005

web.08
GPLSI. Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. Sistema de Recuperación y Extracción de Información Notarial
http://gplsi.dlsi.ua.es/~rafael/IRn/firanet2.html

web.09
Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información.
http://gplsi.dlsi.ua.es/gplsi/articulos/a1997/Art_caep.ps
28/02/2006

web.10
GPLSI. Grupo de Investigación en Procesamiento del Lenguaje Natural y Sistemas de Información. Departamento de Lenguajes y Sistemas Informáticos. Sistema de procesamiento del lenguaje natural SUPAR

(*Slot Unification Parser for Anaphora Resoultion*). Universidad de Alicante.
http://supar.dlsi.ua.es/supar/ayuda.html#_Toc12340454

web.11
European Commission. Joint Research Centre. Institute for Prospective Technological Studies.
http://www.jrc.es/home/report/spanish/articles/vol68/ICT2S686.html
16/11/2005

web.12
Computer Science. New York University.
http://www.cs.nyu.edu/cs/faculty/grishman/muc6.html
16/11/2005

web.13
CLIC. Centre de Llenguatge i Computació. Parc Científic de Barcelona
http://clic.fil.ub.es/personal/civit/PUBLICA/jotri03-b.PDF
17/11/2005

web.14
Estudio de Lingüística del Español. Comité de Redacción.
http://elies.rediris.es/elies18/6.html
15/11/2005

web.15
ACL Anthology. A Digital Archive of Research Papers in Computational Linguistics.
http://acl.ldc.upenn.edu/M/M93/M93-1015.pdf
18/11/2005

web.16
Departament de Llenguatges i Sistemes Informatics. UPC Universitat Politécnica de Catalunya.
http://www.lsi.upc.es/~nlp/tools/parole-sp.html
15/04/2006

web.17
Grupo Estudios de Lingüística del Español.
http://elies.rediris.es/elies18/233.html
12/10/2005

web.18
Grup d'Investigació en Lingüística Computacional. Universitat de Barcelona.
http://www.ub.es/gilcub/castellano/proyectos/investigacion/eagles.html
15/04/2006

web.19
Department of Linguistics and Modern English Language. Lancaster University.
http://www.ilc.cnr.it/EAGLES96/annotate/node17.html#recpu
15/04/2006

web.20
Tejedores del *Web*. Tutoriales y alojamiento *web.*
http://www.tejedoresdelweb.com/307/article-5671.html

web.21
Wikipedia. Enciclopedia libre en línea.
http://es.wikipedia.org/wiki/URL
22/05/2006

web.22
Programación en Castellano. Expresiones regulares.
http://www.programacion.com/java/articulo/expresionesreg/
23/05/2006

web.23
Universidad Europea de Madrid. *Laureate International Universities*.
http://www.esi.uem.es/~jmgomez/pln/
19/09/2005

web.24
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Anotación morfosintáctica.
http://iling.torreingenieria.unam.mx/CursoCorpus2005/3_3_Anotacion_morfosintactica.html
15/04/2006

web.25
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Objetivos del PLN.
http://iling.torreingenieria.unam.mx/CursoPLN/segunda_sesion.pdf
19/09/2005

web.26
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Lingüística y tecnología del lenguaje.
http://iling.torreingenieria.unam.mx/lecturasprohibidas/TecnoLing_Lynx02.pdf
19/09/2005

web.27
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Lingüística de corpus / Diseño y análisis de corpus textuales.
http://iling.torreingenieria.unam.mx/CursoCorpus2005/default.html
12/10/2005

web.28
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Lingüística de corpus / Diseño y análisis de corpus textuales.
http://iling.torreingenieria.unam.mx/CursoCorpus2005/
12/10/2005

web.29
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Text mining.
http://www.iling.unam.mx/mineriadetextos/Text%20Mining%20Sum.doc

web.30
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Tipología de corpus
http://iling.torreingenieria.unam.mx/CursoCorpus2004/1_2_Clasificaci%C3%B3n.html
20/10/2005

web.31
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Delimitación del corpus.
http://iling.torreingenieria.unam.mx/Conferencias/Corpus/Capitulodos.htm
20/10/2005

web.32
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Descripción del corpus lingüístico.
http://iling.torreingenieria.unam.mx/Protocolo/protocolo1.html
20/10/2005

web.33
Grupo de Ingeniería Lingüística. Universidad Nacional Autónoma de México. Herramientas de control del vocabulario científico-técnico: Glosarios y tesauros del CINDOC.
http://www.iling.unam.mx/mineriadetextos/Herramientas.pdf

web.34
Istituto di Lingüística Computazionale. Consiglio Nazionale delle Ricerche. Area della Ricerca di Pisa
http://www.ilc.cnr.it/leneso/descripcion-proyecto.doc
18/09/2005

web.35
IXA Taldea. Euskal Herriko Unibertsitatea. Informatika Fakultatea
http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1023985947/publikoak/cfd_3.pdf
18/09/2005

web.36
LANIA Laboratorio Nacional de Informática Avanzada A. C.
http://www.lania.mx/biblioteca/seminarios/basedatos/pano2.html
04/01/2006

web.37
Biblioteca Virtual en Salud. Aplicación de la minería de datos en la bioinformática.
http://www.bvs.sld.cu/revistas/aci/vol10_2_02/aci03202.htm

web.38
Coordinación de Ciencias Computacionales. Instituto Nacional de Astrofísica, Óptica y Electrónica. México.
http://ccc.inaoep.mx/~mmontesg/publicaciones/2001/MineriaTexto-md01.pdf

web.39
Coordinación de Ciencias de la Computación. Instituto Nacional de Astrofísica, Óptica y Electrónica. México.
http://ccc.inaoep.mx/~mmontesg/tesis%20estudiantes/TesisMaestria-AlbertoTellez.pdf
04/10/2005

web.40
Coordinación de Ciencias Computacionales. Instituto Nacional de Astrofísica, Óptica y Electrónica.

http://ccc.inaoep.mx/~labvision/doo/proy/T32.pdf#search='metodologias%20orientadas%20a%20objetos
12/03/2005

web.41
Universidad Anáhuac Xalapa. Veracruz.
http://www.uax.edu.mx/monica/xml.html
16/12/2005

web.42
W3C *Proposed Recommendation*. Oficina Española.
http://personal2.iddeo.es/polycolor/xhtml1.htm#xml
15/12/2005

web.43
World Wide Web, Consortium. Oficina Española.
http://www.w3c.es/Consorcio/
27/02/2006

web.44
El Consorcio World Wide Web (W3C) Oficina Española. Tokenización.
http://www.w3c.es/Divulgacion/Guiasbreves/TecnologiasXML

web.45
Latindex. Sistema Regional de Información en Línea para Revistas Científicas de América Latina, el Caribe, España y Portugal.
http://64.233.179.104/search?q=cache:7Y8dFBFkaIUJ:www.latindex.unam.mx/ponencias/ppts/Taller-Isabel%2520Galina.ppt+etiquetado+%2B+xml&hl=es
15/12/2005

web.46
Biblioteca Digital Universitaria de la DGSCA. Universidad Nacional Autónoma de México.
http://www.bibliodgsca.unam.mx/tesis/tes7cllg/sec_25.htm
16/12/2005

web.47
Biblioteca Digital Universitaria de la DGSCA. Universidad Nacional Autónoma de México.
http://www.bibliodgsca.unam.mx/tesis/tes7cllg/sec_26.htm
16/12/2005

web.48
Posgrado de Ciencia e Ingeniería de la computación. Universidad Nacional Autónoma de México.
http://www.mcc.unam.mx/~cursos/Objetos/Omt/omt.html

web.49
Dirección General de Servicios de Cómputo Académico. Universidad Nacional Autónoma de México.
http://www.revista.unam.mx/curriculum.html
03/05/2006

web.50
Dirección General de Servicios de Computo Académico. Universidad Nacional Autónoma de México.
http://sistemas.dgsca.unam.mx/publica/pdf/metodologias.PDF

web.51
EMC Rachète Captiva Software.
http://www.swt-concept.com

web.52
Fórum Barcelona 2004. La extracción de la información.
http://www.barcelona2004.org/esp/banco_del_conocimiento/documentos/ficha.cfm?IdDoc=1784
16 /09/2005

web.53
Escola Superior de Tecnologia y Ciencias Experimentals. Departament de Llenguatges i Sistemes Informàtics. Universitat Jaume I. Tesis doctoral, Dolores Llido.
http://www.tdx.cesca.es/TESIS_UJI/AVAILABLE/TDX-0630104-124212//llido.pdf
16/09/2005

web.54
Fundación Ciencias de la Documentación, Cuba.
http://www.documentalistas.com/web/biblios/articulos/20/2004_27.pdf
16 /09/2005

web.55
Departamento de Ingeniería Informática. Universidad Autónoma de Madrid.
http://www.ii.uam.es/~ealfon/esp/research/ie.html

web.56
Data Mining Institute. España.
http://www.estadistico.com/arts.html?20020506

web.57
Ubik World Domination. Programación Neurolingüística.
http://www.ubik.com.ar/pnl/milton.html
27/02/2006

web.58
Autoritat de Certificació de la Comunitat Valenciana.
http://www.accv.es/html-gestion/crl/crl_c.htm
14/02/2006

web.59
The University Of Sheffield.
http://www.shef.ac.uk/about/
15/02/2006

web.60
Distrito de Inteligencia Artificial. Portal de Carlos H. Von der Becke.
http://www.geocities.com/ohcop/heuristi.html
15/02/2006

web.61
Fórum Barcelona. Lingüística Computacional. Comunicación y lenguaje en la era digital.
http://www.barcelona2004.org/esp/banco_del_conocimiento/documentos/ficha.cfm?idDoc=1556
15/02/2006

web.62
The Linux Documentation Project.
http://es.tldp.org/Manuales-LuCAS/FLEX/flex-es-2.5.html
17/02/2006

web.63
Nutrición y Composición Corporal. LABORATORIO DE NUTRICION APLICADA. Universidad Complutense Madrid. Teorema de BAYES (estadística Bayesiana).
http://nutriserver.com/Cursos/Bioestadistica/Teorema_Bayes.html

web.64
Grupo Ibermática. *Business Intelligence.*
http://www.ibermatica.com/ibermatica/businessintelligence/

web.65
Gartner Group.
http://www.gartner.com/

web.66
GestioPolis.
http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/bintna.htm

web.67
Monografías. Portal *web* de tesis, documentos, publicaciones y recursos educativos. *Data Mining.*
http://www.monografias.com/trabajos/datamining/datamining.shtml

web.68
La web del programador.
http://www.lawebdelprogramador.com/

web.69
Real Academia Española.
http://www.rae.es/

web.70
Espacio dedicado a la programación lógica y la recuperación de información, con una atención especial al lenguaje Prolog y otros lenguajes afines, pertenecientes al paradigma lógico y declarativo.
http://programacionlogica.blogspot.com/2006_02_01_programacionlogica_archive.html

web.71
Department of Computing Science. Faculty of Science. University of Alberta. Canada.
http://www.cs.ualberta.ca/~pfiguero/soo/metod/ood.html

web.72
Instituto Tecnológico de Buenos Aires.
http://www.itba.edu.ar/capis/rtis/articulosdeloscuadernosetapaprevia/petronio8.pdf.

web.73
Department of Computer Science. University of Chile, Republic of Chile.
http://sunsite.dcc.uchile.cl/~abassi/WWW/Lengua/ingenieria.html#wile

web.74
Instituto Cervantes. Centro Virtual Cervantes.
http://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/ponenc_ajordan.htm

web.75
Canal audio visual. Sitio de exhibición de obras audiovisuales independientes (Cortometrajes, 3D, Flash, fotografía y audio), tutoriales multimedia (*Flash, Premiere, LightWave, Director, DreamWeaver*), y bases de datos con información sobre empresas, *webs*, productoras y festivales, leyes audiovisuales, preguntas tipo test.
http://www2.canalaudiovisual.com/ezine/books/acjirINFORMATICA/1info01.htm

web.76
Ametzagaiña. Taldea. En el mundo de la comunicación.
http://www.ametza.com/castellano/procesamiento.htm

web.77
Congreso de Sevilla. Instituto Cervantes. Centro Virtual Cervantes.
http://cvc.cervantes.es/obref/congresos/sevilla/tecnologias/ponenc_carbonell.htm

web.78
Asociación Nacional de Instituciones de Educación en Informática, A.C.
http://aniei.org.mx/portal/modules.php?&name=modeloslic&op=areas&func=pi23

web.79
Natural Language Processing and Information Retrieval Group at UNED.
http://nlp.uned.es/~anselmo/parole.html

web.80
Aldea Educativa.
http://www.aldeaeducativa.com/images/quipus.gif

web.81
Asociación de Técnicos de Informática.
http://www.ati.es/rubrique.php3?id_rubrique=148

web.82
Grupo de Ciencias de la Computación. Universidad Nacional de Córdoba. Minería de datos en texto semi-estructurado. Tratamiento de avisos clasificado.
http://www.cs.famaf.unc.edu.ar/~pln/Proyectos/MineriaTexto/files/page4_1.pdf

web.83
Grupo de Inteligencia Artificial. Departamento de Arquitectura de Computadores y Ciencia de la Computación e Inteligencia Artificial. Escuela Sup. de Ciencias Exp. y Tecnología. Universidad Rey Juan Carlos.
http://platon.escet.urjc.es/grupo/docencia/automatas/apuntes/capitulo5.pdf

web.84
COLE Research Group. COMPILADORES Y LENGUAJES. Facultad de Informática de la Universidad de Coruña y de la Escuela Superior de Ingeniería Informática de la Universidad de Vigo.
http://coleweb.dc.fi.udc.es/cole/library/ps/Alo2000a_B.pdf

web.85
Universidad del País Vasco, Campus de *Gipuzkoa*.
http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/primero.htm
23/06/2006

web.86
Universidad del País Vasco, Campus de *Gipuzkoa*.
http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/colecciones/stringtokenizer.htm
28/06/2006

web.87
Enciclopedia libre multilingüe.
http://es.wikipedia.org/wiki/DARPA
28/06/2006

web.88
Sistema para un análisis morfológico automático para el español
www.cic.ipn.mx/~sidorov/agme/index.html

web.89
ISO (Organización Internacional de Estándares)
http://www.iso.org/iso/en/aboutiso/introduction/index.html#two

web.90
Gate: General Architecture for Text Engineering from the University of Sheffield
http://gate.ac.uk/ie/annie.html

web.91
Portal estadístico de habla hispana dependiente del *Data Mining Institute*
http://www.estadistico.com/arts.html?20020506

web.92
Instituto Nacional de Astrofísica, Óptica y Electrónica
http://ccc.inaoep.mx/~mmontesg/publicaciones/2001/MineriaTexto-md01.pdf

web.93
Laboratorio de Lenguaje Natural y Procesamiento de Texto, Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional (IPN)
http://ccc.inaoep.mx/~hrl_04/versionesFinales/Sidorov_14.pdf#7

web.94
System for automatic morphological analysis of Spanish
http://www.cic.ipn.mx/~sidorov/agme/index.html