2. Conceptos básicos

En este capítulo se describen brevemente los conceptos básicos divididos en dos secciones: por una parte, el procesamiento del lenguaje natural (PLN) en donde se detallan las nociones propias de esta rama de la computación; y, por otra parte, las herramientas computacionales utilizadas.

2.1. Procesamiento de lenguaje natural

La inmensa cantidad de información que se maneja diariamente en el ámbito académico, empresarial y de gobierno requiere que sea procesada debidamente para ahorrar tiempo y recursos en su interpretación y análisis. Por tanto, la ingeniería en computación, en especial la inteligencia artificial, abordan este problema a través del procesamiento del lenguaje natural (PLN).

El PLN es una rama de las ciencias de la computación, relacionada con la inteligencia artificial y la lingüística. El PLN elabora sistemas computacionales para la comunicación eficiente entre personas y máquinas a través de lenguajes naturales. Diseña mecanismos para que los programas ejecuten o simulen la comunicación. Implica aspectos cognitivos, de memoria y de comprensión del lenguaje.

Entre las aplicaciones del PLN destacan la minería de textos y la recuperación de información. La minería de textos se refiere a un conjunto de técnicas estadísticas o lingüísticas, que permiten la extracción en textos de información de manera automática. La recuperación de información (RI) “es la elaboración de sistemas para la búsqueda y selección de documentos que cumplan ciertos criterios señalados por un usuario” (Alarcón, 2009: 30). La extracción de información (EI) “se encarga de desarrollar sistemas para la búsqueda y selección de datos específicos sobre eventos, entidades o relaciones a partir de un conjunto de documentos”. (Alarcón, 2009: 30).

Sosa (1997) declara que “el PLN se concibe como el reconocimiento y utilización de la información expresada en lenguaje humano a través del uso de sistemas informáticos”.

2.1.1. Extracción terminológica

La extracción terminológica trata sobre la identificación de términos. Para esto, utiliza la terminología. Según Cabré (1993: 82):

> La terminología puede ser entendida, por un lado, como "el conjunto de principios y de bases conceptuales que rigen el estudio de los términos", y por otro, como "una materia de intersección que se ocupa de la designación de los conceptos de las lenguas de especialidad", cuyo objetivo, a grandes rasgos, es la denominación de los conceptos.

Sager (1993: 35), por su parte, define estas tres concepciones de la terminología de la siguiente forma:

1. El conjunto de prácticas y métodos utilizados en la recopilación, descripción y presentación de términos;
2. una teoría, es decir, el conjunto de premisas, argumentos y conclusiones necesarias para la explicación de las relaciones entre los conceptos y los términos;
3. un vocabulario de un campo temático especializado.

La terminología surgió gradualmente como una disciplina lingüística separada cuando personas como Wüster opinaron que los términos deberían ser tratados de manera diferente que las palabras del lenguaje general (Pearson, 1998):

1) Porque en contraste con la lexicología, donde la unidad léxica es el punto de inicio, el trabajo de la terminología empieza desde el concepto. Un concepto consiste en un agregado de características que pueden ser reconocidas como comunes a un número de objetos individuales y que pueden usarse como medios para el ordenamiento mental y la comunicación. Un concepto es un elemento del pensamiento.

2) Porque los terminólogos se interesan en el vocabulario aislado. Los términos son diferentes de las palabras no solo en cuanto a su significado sino a su naturaleza y uso. Hay una correspondencia entre el término como una etiqueta y el concepto como un constructo mental, e idealmente, un término se refiere únicamente a uno y solo un concepto de un campo determinado.

3) Los terminólogos se ocupan de imponer normas para el uso del lenguaje. Esto llevó a la creación de vocabularios estandarizados que se usarían como un medio para representar las estructuras conceptuales que llevan cada campo de conocimiento.

El dominio de una materia especializada contiene una serie de conceptos o constructos mentales que están representados por términos. La relación entre conceptos y términos está estandarizada. La relación entre conceptos está representada por relaciones lógicas y ontológicas que se utilizan para construir sistemas jerárquicos de conceptos (Pearson, 1998).

La EI elabora sistemas para la extracción automática de términos. La EI se vale de sistemas basados en reglas lingüísticas o en reglas estadísticas para el análisis de textos que permitan obtener candidatos a términos. También se ha dedicado a la obtención de contextos definitorios para inferir el significado de los términos de acuerdo con sus características, sus atributos o sus relaciones con otros términos. Así, se convierte en un proceso enfocado a extraer relaciones semánticas (RSs) y contextos definitorios (CDs).

2.1.2. Contextos definitorios

"Un contexto definitorio es un fragmento textual en donde se obtiene información para conocer el significado de un término. Son unidades discursivas que vehiculan información predicativa sobre un término en un dominio de conocimiento específico." (Alarcón, 2009: 39)

Los contextos definitorios no solo aportan información sobre el significado del término sino que también permiten conocer las relaciones semánticas que existen con otros términos y de esta manera establecer una red conceptual del campo del conocimiento al que pertenecen.

De Bessé (1991): entiende por contexto al punto de inicio de cualquier trabajo terminográfico. El contexto es el entorno lingüístico de un término conformado por un enunciado, es decir, las palabras o frases alrededor de dicho término, y que persigue dos funciones básicas: aclarar el significado de un término e ilustrar su funcionamiento.

Por tanto, los contextos constituyen un elemento esencial para la descripción de un concepto y resultan indispensables para redactar una definición.

De Bessé distingue los CDs como aquellos contextos donde se aporta información sobre los atributos de los términos. Diferencia los contextos conceptuales como aquellos que se refieren a características sobre las relaciones conceptuales de los términos, en tanto los materiales proveen instrucciones sobre el alcance de los términos y la forma en que éstos operan en un contexto determinado.

La importancia de los contextos definitorios y las relaciones semánticas en las prácticas terminográficas radica en la necesidad de comprender un término y situarlo frente a otros de su campo, a través del análisis de las situaciones reales en las que estos se definen dentro de la comunicación del lenguaje especializado.

Un contexto definitorio (CD) es un fragmento textual que, mediante patrones verbales, patrones pragmáticos, tipografía y nexos, engloba un término y su definición la cual puede ser extraída automáticamente.

De Alarcón, Bach y Sierra (2008:2).

> Se entenderá por contexto definitorio a todo aquel fragmento textual de un documento especializado donde se define un término. Estas unidades están formadas por un término (T) y una definición (D), los cuales se encuentran conectados mediante un patrón definitorio (PD), por ejemplo verbos como definir o entender. Opcionalmente, un contexto definitorio (CD) puede incluir un patrón pragmático (PP), esto es, estructuras que aportan condiciones de uso del término o que matizan su significado, por ejemplo en términos generales o en esta investigación.

En la figura 2.1 se ve como un CD, contiene un término (T), un patrón definitorio (PD) y una definición (D), adicionalmente puede o no contener un patrón pragmático (PP).

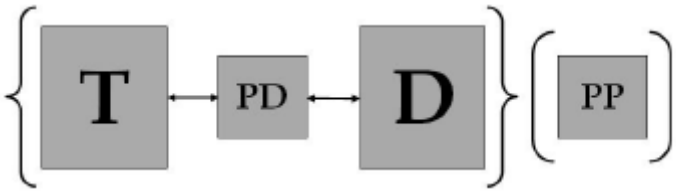

Figura 2.1. Estructura de un CD (Sierra, 2009)

Existen dos tipos de patrones definitorios: los patrones tipográficos, y los patrones sintácticos.

Sierra (2009:18) comenta sobre los patrones tipográficos:

> La tipografía de un texto es un recurso que sirve como ayuda visual para identificar fácilmente los elementos importantes y diferenciarlos del resto del texto común. En muchos casos, los términos tienden a ser frecuentemente resaltados. Muchas veces ocurre que la definición también se encuentra señalizada con algún elemento tipográfico o con alguna tipografía específica. En este sentido, los patrones tipográficos se utilizan ya sea para resaltar a los elementos constitutivos mínimos de los CDs o bien para conectar dichos elementos.

Y sobre los patrones sintácticos (Sierra, 2009:19):

> Un camino para extraer CDs de manera automática en textos de especialidad consiste en identificar las estructuras sintácticas recurrentes tanto de los elementos mínimos constitutivos como de los conectores que unen a estos dos elementos. Alarcón (2009) describe dos patrones sintácticos que sirven para conectar el término con su definición.

> Cuando dichos conectores tienen como núcleo un verbo, tenemos entonces un patrón verbal definitorio (PVD). Cuando se emplean otro tipo de formas sintácticas cuya finalidad es establecer una reformulación de una idea o concepto, y que se utilizan para esclarecer el significado de un término, tenemos marcadores reformulativos.

Una clasificación de los contextos definitorios es por el tipo de su definición y pueden ser (Alarcón, 2009: 135-137):

- *Analíticos.* En los cuales se presenta tanto el *género próximo* al cual pertenece el término, como la *diferencia específica* que lo diferencia de otros elementos de su campo de conocimiento; un ejemplo extraído del corpus de Alarcón (2009):

  > *De acuerdo con el enfoque integral expuesto, <t>el sistema de gestión</t> se concibe como <d><genus>una organización,</genus> <differentia>cuyo funcionamiento busca lograr ciertos objetivos a través de la operación de los diversos subsistemas interrelacionados que la componen</differentia></d>.*

- *Funcionales.* En estos contextos no se incluye el género próximo pero se indica la función del término. P. ej. de Alarcón, (2009):

  <t>Las escalas de Likert</t> se usan habitualmente <d>para <función>cuantificar actitudes y conductas</función></d>.

- *Extensionales.* Como en el caso anterior, en estos contextos tampoco se presenta el género próximo, sin embargo se enumeran características que conforman las partes de la entidad. P. Ej de Alarcón, (2009).

  > *El terminal utilizado es <t>el videoteléfono</t>, que consta básicamente de <d><partes>una pantalla, cámara, teclado, micrófono, altavoz</partes></d>.*

- *Sinonímicos.* En este tipo de contextos se presenta otro término que puede equivaler semánticamente al término definido. P. ej. de Alarcón (2009):

En la mujer, <t>el conducto vaginal<t> se llama también <d><termino equivalente>conducto de Nuck</termino equivalente></d>.

Recapitulando, las partes constitutivas de un CD son el término y su definición; ambos elementos se encuentran conectados por un patrón definitorio (PD) el cual puede ser tipográfico: por medio de marcadores discursivos como son que el texto esté en negritas o subrayado o que este precedido por dos puntos, y algunos otros símbolos (Patrón tipográfico definitorio); o sintáctico: por medio de verbos definitorios y sus nexos (Patrón Verbal Definitorio). Pudiendo contener patrones pragmáticos en su estructura.

Cabré (1993) define a los patrones pragmáticos como un tipo de información relevante para situar al término dentro del contexto en el cual aparece. Esta información describe el uso de los términos y precisa las condiciones de uso o de alcance de dicho término, como son el ámbito temático, la ubicación geográfica, las instituciones que utilizan el término, el nivel de especialidad, o la frecuencia de uso, entre otras características pragmáticas.

Un ejemplo de contexto definitorio obtenido de es:

<t> Las escalas de Likert </t><PVD>se usan</PVD> <PP>habitualmente</PP> <Nx>para</Nx><d> cuantificar actitudes y conductas.</d> (Alarcón, 2009: 136).

2.1.3 Etiquetado de partes de la oración

Las partes de la oración (también conocidas como POS –de sus siglas en inglés, *part of speech*-, clases de palabras, clases morfológicas, categorías gramaticales y etiquetas léxicas) son muy importantes para el PLN por la información que proporcionan de una palabra y las palabras adyacentes.

Las partes de la oración se utilizan para la recuperación de información (RI) y para extraer verbos o sustantivos de diferentes textos. La localización automática de las partes de la oración es útil en el análisis sintáctico, en algoritmos de desambiguación, en análisis sintácticos superficiales de palabras, para encontrar nombres, datos, fechas en textos y para otras entidades de la extracción de información. Finalmente, los corpus que han sido marcados con partes de la oración se usan para la investigación de la lengua.

Sierra y Alarcón (2010:2) definen al etiquetado de partes de la oración de la siguiente manera:

> El siguiente nivel en PLN que sirve para la búsqueda de información es el análisis de las partes de la oración y su etiquetado.
>
> Este proceso consiste en identificar la categoría gramatical (ya sea verbo, adjetivo, sustantivo, etc.) de cada palabra y asignarle una etiqueta que será utilizada posteriormente en los sistemas inteligentes de extracción de información. Este proceso sirve además para otros niveles superiores de análisis en PLN. Por ejemplo, si queremos encontrar imágenes que hagan referencia al instrumento musical bajo, es probable que también se recuperen objetos cuya característica implique la palabra bajo como adjetivo o preposición.

Para Méndez (2009:72):

> El problema de la identificación automática de categorías gramaticales puede entenderse como el problema de asignar cierta etiqueta, que corresponde a una categoría, a cada palabra de un corpus (cf. Charniak 1996/1993)[1]. Para Voutilainen (1999a: 3)[2] las etiquetas son símbolos descriptivos que se asignan a palabras en un texto ya sea de forma manual o automática. Entonces estas etiquetas codificarían el nombre de la categoría y los rasgos morfosintácticos asociados.

A las etiquetas que se utilizan para marcar las palabras se les llama etiquetas de partes de la oración o etiquetas morfosintácticas. Cuando en un corpus se decide con qué etiquetas se va a etiquetar, se tiene un conjunto de etiquetas de corpus. Existen dos consideraciones fundamentales para dichas etiquetas: la especificidad de las categorías, que es el nivel de detalle con el que será etiquetados y por otro lado las etiquetas en sí y cómo se codifican dichos rasgos en ellas (Méndez, 2009). En la tabla siguiente se muestra el conjunto de etiquetas del Corpus del Español Mexicano Contemporáneo (CEMC):

[1] Charniak, Eugene. 1996/1993. *Statistical language learning*. Cambridge, Massachusetts: The MIT Press.

[2] Voutilainen, Atro. 1999ª. "*Orientación*", en Hans Halteren (ed.) Syntactic Wordclass Tagging, Dordrecht. Netherlands: Klluwer Academic, 3-7.

Etiqueta	Categoría
0	Ambigua
1	Adverbio
2	Adjetivo
3	Conjunción
4	Preposición
5	Pronombre
6	Artículo
7	Contracción
8	Nominal
9	Verbo
A	Apoyos conversacionales
B	Nombres propios
C	Otros (cifras, errores y palabras que comenzaban con mayúscula)

Tabla 2.1 Conjunto de etiquetas del CEMC Méndez (2009: 75)

Existe un estándar para las etiquetas gramaticales que es multilingüe y vale la pena mencionar: el estándar EAGLES (Expert Advice Group for Language Engineering Standars). Se ha trabajado en una serie de lineamientos para el etiquetamiento de POS de diversas lenguas (danés, alemán, inglés, francés, holandés, griego, italiano, portugués y español) y la posibilidad de adaptarse a las particulares de alguna lengua en específico (Leech & Wilson 99).

La codificación EAGLES se basa en un conjunto de letras para las categorías (atributos obligatorios), rasgos morfosintácticos (atributos recomendados) y particularidades de cada lengua (atributos opcionales), ordenadas en posiciones consecutivas.

Para el etiquetado con partes de la oración se utilizan varios algoritmos, desde las reglas elaboradas a mano, los métodos probabilísticos (etiquetado HMM y etiquetado de máxima entropía), hasta el etiquetado basado en la transformación y en la memoria para la etiquetación de partes de la oración (Jurafsky, 2007).

De los muchos algoritmos que existen dos se han aplicado a este trabajo: el algoritmo de etiquetado de Brill y el TreeTagger.

2.1.3.1. Algoritmo de Brill

El modelo Brill es un etiquetador basado en reglas que se desempeña con igual eficiencia que los modelos probabilísticos, obviando las limitaciones comunes de los enfoques basados en reglas. Las reglas se asumen automáticamente. Se requiere menos información almacenada, menos reglas, y se facilita encontrar e implementar mejoras al etiquetado y al traslado de un conjunto de etiquetas o género de corpus a otro. El algoritmo funciona encontrando automáticamente las reglas de etiquetado y reparando su debilidad, por lo que mejora su rendimiento.

El sistema requiere de un corpus previamente etiquetado para entrenar al algoritmo. Se utiliza el 90% para el proceso descrito a continuación, 5% para la selección de los parches (reglas de etiquetado manuales) que arrojan menos resultados y 5% para pruebas.

El etiquetador inicia asignando al corpus de trabajo la etiqueta POS más probable de acuerdo con el corpus de entrenamiento, sin considerar información contextual; toma en cuenta las palabras que inician con mayúscula como nombres propios. Las palabras que le son desconocidas de acuerdo con el corpus de entrenamiento las etiqueta de acuerdo con las 3 últimas letras de las mismas.

El etiquetado anterior arroja resultados cercanos al 90% (Brill 1992: 113), posteriormente se procede al análisis de los errores etiquetando el 5% del corpus de entrenamiento (corpus de parches) con las categorías encontradas y comparándolas con las que ya fueron asignadas por un experto. Con los errores encontrados se procede a aplicar una serie de reglas (parches) previamente elaboradas que alimentan al sistema.

Se aplican dichos parches y se compara cuántos errores se corrigieron; el sistema repite este proceso hasta encontrar cuáles son las reglas que eliminan la mayor cantidad de errores y las guarda.

Posteriormente, cuando se inserta al sistema un texto no etiquetado, se procede etiquetando primero con las categorías gramaticales más probables y después aplicando cada parche, en orden, hasta producir la salida final.

El etiquetado tiene la particularidad de elegir solamente las mejores reglas para el etiquetado, por lo que permite ingresar una gran cantidad de reglas sin que se afecte el resultado final, lo que facilita la elección de las reglas más sencillas y más efectivas para el etiquetador.

El etiquetador Brill es portátil. Es transferible a otros conjuntos de etiquetas o géneros y a otros idiomas. Si el etiquetador se utilizara en un corpus diferente se encontraría un conjunto distinto de parches adecuado.

En este etiquetador basado en reglas, la información se captura con menos de ocho reglas, facilitado el desarrollo posterior del etiquetador. La información contextual se expresa de manera compacta (Brill, 1992).

2.1.3.2. TreeTagger

TreeTagger es un etiquetador POS probabilístico, robusto y portable, trabaja de manera similar a un etiquetador de trigramas con algunas diferencias que explicaré a continuación. Para funcionar requiere de un lexicón de palabras completas, un lexicón en forma de árbol de sufijos y una categoría default.

Del corpus de entrenamiento etiquetado se calculan las probabilidades de las categorías de acuerdo a los trigramas, para esto se normaliza procurando que a los trigramas raros o mal formados se les asigne un valor diferente de cero, pero muy bajo y se normalizan las probabilidades.

El etiquetador estima la transición de probabilidades generando un árbol binario de decisión a partir del corpus de entrenamiento. El árbol de decisión automáticamente determina el tamaño apropiado del contexto que se utiliza para definir las probabilidades de transición.

Un trigrama se determina siguiendo la senda marcada por el árbol hasta que se alcanza la última hoja. Cada hoja contiene un vector de probabilidades de categorías. Posteriormente, se recorta el árbol de trigramas siguiendo una medida de cantidad de información, para eliminar hojas y nodos con poco contenido de información. Como otros etiquetadores probabilísticos, TreeTagger determina la mejor etiqueta utilizando el algoritmo de Viterbi (Forney, 1973).

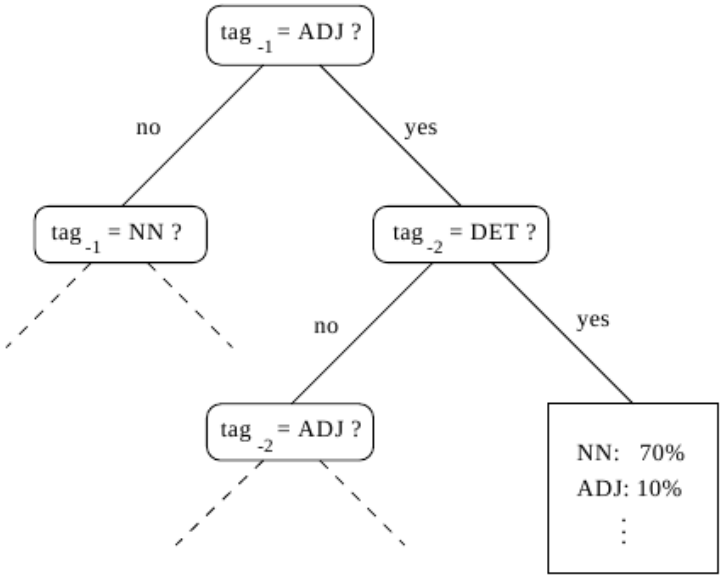

Figura 2.2 Un árbol de decisión de TreeTagger (Schmid, 1994:3)

TreeTagger requiere, a priori, las probabilidades de las categorías y para esto se apoya de dos lexicones que son, el primero, una lista de palabras completas con sus respectivos vectores de probabilidad, el segundo, un árbol de sufijos en cuyos nodos existen caracteres y en sus hojas los vectores de probabilidad de las categorías. Además, con un análisis de los lexicones se obtiene un vector default de probabilidades que permite desambiguar palabras desconocidas.

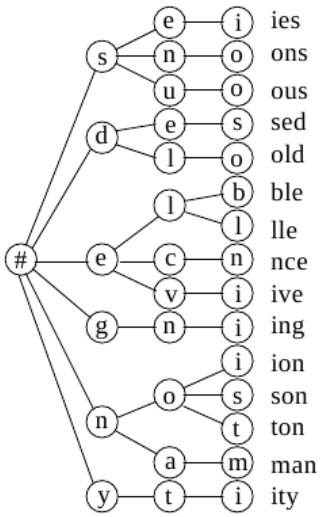

Figura 2.3 Árbol de sufijos de longitud tres (Schmidt, 1994:5)

El algoritmo difiere de otros métodos probabilísticos en la forma que las probabilidades de transición son estimadas, específicamente con un árbol de decisión. TreeTagger es robusto respecto al corpus de entrenamiento y no pierde precisión con un corpus de entrenamiento pequeño como en otros etiquetadores por trigramas. TreeTagger corta el tamaño del contexto donde es necesario para obtener estimados de probabilidad confiables (Schmidt, 1994).

2.1.4 Lematización

La lematización se puede entender como el proceso informático de reducir una palabra a su forma más simple, llamada lema. A veces se confunde el término lematización en español con *stemming,* el cual consiste en truncar los morfemas funcionales de una palabra. Para esto Jurafsky (2006:53) define:

> We will introduce some related algorithms in this chapter. In some applications we don't need to parse a word, but we do need to map from the word to its root or stem. For example in information retrieval and web search (IR), we might want to map from foxes to fox; but might not need to also know that foxes is plural. Just stripping off such word endings is called stemming in IR. We will describe a simple stemming algorithm called the Porter stemmer.
>
> For other speech and language processing tasks, we need to know that two words have a similar root, despite their surface differences. For example the words sang, sung, and sings are all forms of the verb sing. The word sing is sometimes called the common lemma of these words, and mapping from all of these to sing is called lemmatization.

Además Srivastava (2009:15) da una definición más sencilla:

> A simple example of semantic similarity mapping is stemming, that consists of removing inflection from words.

Como bien menciona Jurafsky, el algortimo de Porter es el algoritmo más conocido de lematización. De palabras del propio Porter (1980:1), que trabaja removiendo sufijos:

> Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term CONNECT. In addition, the suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is always advantageous…

> …Assuming that one is not making use of a stem dictionary, and that the purpose of the task is to improve IR performance, the suffix stripping program will usually be given an explicit list of suffixes, and, with each suffix, the criterion under which it may be removed from a word to leave a valid stem.

El algoritmo de Porter trabaja con una serie de reglas para eliminar los sufijos de las palabras. Primero se calcula el número de consonantes y vocales en cada palabra y se le asigna un número. Después se aplican las reglas escritas de acuerdo al tamaño de la palabra y la sustitución de dicha regla. Un ejemplo de una regla es:

Tamaño	Sufijo	Reemplazo de sufijo	Ejemplo
m>0	tional	Tion	conditional > condition

Tabla 2.2 Ejemplo de regla en el algoritmo de Porter (Porter 1980:2).

2.1.5 Corpus

Un corpus se define en su forma más simple como cualquier recopilación de muestras de lenguaje. Torruella y Llisterri (1999) entienden que, para la filología, un corpus es una recopilación de textos bien organizada. Cabe decir que, actualmente, los corpus cuentan con soporte informatizado y herramientas computacionales que ayudan en su procesamiento. Para Sierra (2008:446) un corpus es:

> Un conjunto de datos reales y aceptables, debidamente ordenado, codificado y organizado, de diferentes textos recopilados, pertenecientes a un código lingüístico determinado, oral o escrito.

Comúnmente se menciona que un corpus debe contar con las características de variedad y representatividad de la muestra de lenguaje que estén representando (Sierra, 2008).

2.1.5.1 Corpus Técnico del IULA

El corpus técnico del IULA (Cabré y Vivaldi, 1997) es un corpus creado por el Instituto Universitario del Lingüística Aplicada de la Universidad Pompeu Fabra en Barcelona. Este corpus cuenta con alrededor de 10 millones de palabras en español. Para acceder a este, se utiliza la herramienta de búsqueda *Bwananet* que permite una búsqueda básica, estándar y compleja.

Cuenta con diferentes áreas de especialidad en diferentes idiomas, como español, catalán,

inglés, francés y alemán. A continuación muestro una tabla de la cantidad de palabras por idioma y área de especialidad:

Área	Catalán	Español	Inglés	Francés	Alemán	Total
Derecho	1 463 000	2 085 000	431 000	44 000	16 0000	4 039 000
Economía	1 776 000	1 091 000	274 000	78 000	27 000	3 246 000
Medio ambiente	1 506 000	1 062 000	599 000	230 000	429 000	3 826 000
Informática	655 000	1 277 000	338 000	194 000	83 000	2 497 000
Medicina	2 619 000	4 077 000	1 555 000	27 000	198 000	8 476 000
Total	8 019 000	9 542 000	3 197 000	573 000	753 000	22 084 000

Tabla 2.3. Número de palabras por lengua y ámbito (Cabré y Bach, 2004:174)

Es importante mecionar que este corpus fue utilizado por el Ecode como corpus de entrenamiento y de pruebas. Una característica del corpus es que se encuentra etiquetado con POS.

2.2 Herramientas utilizadas y el lenguaje Perl y sus módulos

Las herramientas que utilicé, además del lenguaje de programación *Perl*, fueron *antiword* y *elinks*. El programa *antiword* se encarga de la extracción de texto a partir de archivos Word; por otro lado, *elinks* es un explorador web de consola que extrae texto de URL, HTML y XML.

2.2.1 Herramientas externas

Si bien todo el sistema está hecho en Perl, se utilizaron programas externos para funciones específicas que resultó mucho más fácil utilizarlas que programar su funcionalidad. Estas son:

2.2.1.1 Antiword

Antiword[3] es un programa libre distribuido bajo la licencia de software libre GPL, el cual tiene la funcionalidad de extraer el texto del formato de archivo utilizado por Microsoft Word.

Microsoft únicamente mantiene los formatos de su suite de oficina para los sistemas Windows y Mac y no son abiertos ni existe un manual con sus características, por lo que en el caso del Ecode fue necesario buscar una herramienta que lo permitiera y Antiword resultó ser adecuado para proveer esta funcionalidad al Ecode para poder extraer contextos definitorios de archivos realizados en Ms Word.

2.2.1.2 Elinks

Elinks[4] es un navegador web de consola para sistemas tipo Unix creado en 2001 a partir del explorador de consola Links.

Tiene la particularidad de que, además de proveer todas las funcionalidades de un navegador, permite el volcado del texto de la página de entrada hacia la salida estándar de los sistemas tipo Unix. También permite extraer el texto de archivos HTML y XML, además de los sitios web que se le indiquen.

Esta función permite que Ecode obtenga el texto ya sea de una dirección de internet remota o de archivos de usuario en formato HTML o XML para su posterior procesamiento.

2.2.2 Perl y el procesamiento de texto

Perl se define como un lenguaje de programación de propósito general orientado tanto a programación estructurada como programación en objetos y enfocado a la manipulación de texto. Perl se describe como un lenguaje eficiente, sencillo y completo.

Perl es un lenguaje de programación de gran utilidad para el procesamiento de lenguaje natural, debido a la gran cantidad de herramientas, tanto propias del lenguaje como módulos de terceros, razón por la que se eligió en un principio para realizar el Ecode y se mantiene utilizando dicho lenguaje durante el presente trabajo.

---

[3] Sitio web de Antiword: http://www.winfield.demon.nl/
[4] Sitio web de Elinks: http://elinks.cz/

2.2.2.1 Módulos y extensiones de Perl utilizados

Durante la realización del sistema se utilizaron diferentes módulos de Perl. Un módulo es un conjunto de funciones que proveen una cierta funcionalidad, en este caso Perl y su capacidad de procesamiento. A continuación describo brevemente los módulos que se utilizaron en el sistema.

2.2.2.1.1 Perl::DBI

Perl::DBI [5] (Data Base Interface) es un módulo que permite al programador un ambiente estándar que admite la comunicación a diversos manejadores de bases de datos.

En este caso en particular utilicé este módulo para comunicar al Ecode con la base de datos de Describe, de esta manera se utiliza Perl:DBI para comunicarse con la base, extraer el texto bruto y almacenar los resultados de acuerdo con los parámetros que Describe requiere.

2.2.2.1.2 Perl:: Lingua::Stem::Es

Perl::Lingua::Stem::Es[6] (Porter_stem_Es.pm). Se encarga de lematizar el texto en español con el algoritmo de Porter. Esto se hace porque en desarrollos futuros se pueden necesitar los contextos definitorios lematizados para hacer algún tipo de procesamiento; por ejemplo, en Molina (2009) se lematizan los contextos definitorios para agruparlos por su similitud.

2.2.2.1.3 Perl::File::Type

Perl::File::Type[7] se encarga de detectar el tipo de algún archivo de acuerdo con la información contenida, ya sea en la cabecera de este o en la estructura de los datos que contiene. Se utiliza en Ecode para detectar el tipo de archivo de entrada y tratarlo según sea el caso.

---

[5] El sitio web de Perl::DBI es: http://dbi.perl.org/
[6] Para Perl::Lingua::Stem::ES: http://search.cpan.org/~jfraire/Lingua-Stem-Es-0.04/
[7] Y para Perl::File::Type es: http://search.cpan.org/~pmison/File-Type-0.22/

Cabe mencionar que dentro de esta biblioteca no estaban bien diferenciados algunos tipos de archivo, en especial de texto plano, lo que ocasionaba problemas al Ecode. Por ello tuve que modificar una pequeña parte de la librería para que entregara los archivos de texto limpios y facilitar su posterior procesamiento.

2.2.2.1.4 Perl::CAM::PDF

El módulo Perl::CAM::PDF[8] sirve para leer y escribir archivos en formato PDF, este módulo se apega estrictamente a las especificaciones de PDF de Adobe, por lo que archivos que no se apeguen pueden no ser leídos correctamente.

En el caso particular del Ecode se utiliza para aceptar archivos de entrada tipo PDF, se les extrae el texto utilizando este módulo y posteriormente se procesan como cualquier otro texto.

2.2.2.1.5 Perl::Text::Iconv

Perl::Text::Iconv[9] es el módulo encargado de proveer una interfaz entre Perl y el programa estándar de UNIX iconv, que tiene la función de convertir entre diferentes tipos de codificaciones de caracteres.

Ecode lo utiliza para tratar de convertir de cualquier codificación en el texto de entrada hacia UTF-8, que es con la que trabaja por defecto.

Cabe mencionar que la detección de codificaciones es un asunto de extrema complejidad y que este módulo no detecta todas las codificaciones ni todas las variantes de ellas, por lo que, si bien hace un trabajo adecuado, no es tan confiable y puede convertir mal las cadenas de caracteres.

[8] El sitio web es: http://search.cpan.org/~cdolan/CAM-PDF-1.55/
[9] Su sitio es: http://search.cpan.org/~mpiotr/Text-Iconv-1.7/