Evaluación del Módulo 2

Laura Rodríguez Navas
rodrigueznavas@posgrado.uimp.es

Enero 2020

1. Explique cuáles son los dos grandes paradigmas del PLN e indique sus diferencias. Trate de ampliar la información que hay en la lección.

   Los dos grandes paradigmas del PLN son la Lingüística generativa y la Tecnología de la Lengua.

   La **Lingüística generativa**, también conocida como teoría racionalista y Chomskiana, describe las estructuras del lenguaje humano definidas en el cerebro (Lenguaje-I) a partir de las derivaciones de dicho lenguaje, las cuales se encuentran impresas en los textos.

   La **Tecnología de la Lengua**, también conocida como Ingeniería Lingüística o PLN estadístico, modela e identifica los parámetros del lenguaje mediante el análisis y procesamiento exhaustivo del Lenguaje-E, a partir de la reproducción física de este.

   La primera diferencia es que la Lingüística generativa fundamenta que el lenguaje es un mecanismo tan complejo que no puede ser adquirido por los sentidos. Contrariamente, la Tecnología de la Lengua fundamenta que la mente humana tiene la capacidad innata de establecer asociaciones, reconocer patrones y de generalizar ocurrencias de eventos que son percibidos a través de los sentidos. Por lo tanto, el lenguaje puede ser adquirido por los sentidos.

   La segunda diferencia es que la Lingüística generativa se refiere al Lenguaje-I ("lengua interna" o interiorizada) que contrasta con el uso del Lenguaje-E ("lengua exterior") de la Tecnología de la Lengua. Técnicamente el Lenguaje-I se refiere a la representación mental o conocimiento lingüístico inconsciente que un hablante tiene de su lengua y por tanto es un objeto mental. En cambio, el Lenguaje-E abarca los aspectos de lengua relacionados con su uso social, los hábitos sociolingüísticos y aspectos externos del uso de la lengua en comunidades humanas.

2. ¿Cuáles son los niveles de análisis o procesamiento de un sistema típico de PLN? No se limite a indicar los nombres, defínalos y explique las relaciones existentes entre ellos.

   Los niveles de análisis o procesamiento de un sistema típico de PLN son: Tokenización y Segmentación, Análisis Léxico, Análisis Sintáctico, Análisis Semántico y Análisis Pragmático.

   El método de **Tokenización y Segmentación** consiste en, dado un fragmento de texto, identificar las palabras y tokens, así como el conjunto de oraciones en el que se organiza el texto.

   El **Análisis Léxico** asocia a las palabras y tokens, información relacionada con su propia naturaleza y con la función que desempeñan en la oración. Como consecuencia requiere haber identificado las unidades mínimas que constituyen un mensaje (palabras y tokens), así como su organización en oraciones. Por tanto, primero se debe realizar el método de Tokenización y Segmentación.

El método de **Análisis Sintáctico** consiste en analizar cada una de las oraciones de un mensaje para determinar si se adecua a la gramática formal de una lengua.

Antes del Análisis Sintáctico se requiere identificar las unidades mínimas del lenguaje: las palabras y tokens, así como toda la información relativa a ellas. Por esta razón, primero se debe realizar el Análisis Léxico.

Además, según el analizador léxico que se utilice durante el Análisis Léxico, genera un árbol sintáctico en el que se indican las oraciones y sus relaciones.

El método de **Análisis Semántico** consiste en identificar, de la información procedente del Análisis Sintáctico, el significado de las palabras y de las figuras lingüísticas presentes en un mensaje, y representar conceptualmente el significado subyacente de este.

El Análisis Semántico está relacionado con la resolución de la ambigüedad.

En el método de **Análisis Pragmático** se trata de determinar el significado del discurso subyacente en el mensaje.

3. Indique cuáles son las dependencias que se deben tener en cuenta a la hora de desarrollar un tokenizador. No se limite a nombrar las dependencias, desarrolle su respuesta.

   Las dependencias que se deben tener en cuenta a la hora de desarrollar un tokenizador son: la Codificación de caracteres, el Idioma, Corpus y la Aplicación.

   La **Codificación de caracteres** es el método que permite convertir un carácter de un lenguaje natural (alfabeto o silabario) en un símbolo de otro sistema de representación y se basa en definir tablas que indiquen el carácter en el lenguaje natural y su correspondencia en el lenguaje del sistema informático. Ejemplos: el código Morse, la norma ASCII o la UTF-8, entre otros.

   La Codificación de caracteres es un aspecto fundamental de los sistemas informáticos para trabajar con textos, ya que estos utilizan la Codificación de caracteres para convertir y representar los diferentes lenguajes escritos del mundo a una representación que dicho sistema entienda, y así poder almacenar o transmitir dicha información.

   Uno ejemplo claro sobre su importancia es que actualmente todos los sistemas operativos Windows, Linux e iOS utilizan la Codificación de caracteres UTF-8, porqué anteriormente codificaciones diferentes provocaban problemas ya que hacían que los ficheros creados en un sistema Windows mostraran los caracteres de manera diferente en el resto de sistemas operativos.

   Cada **Idioma** define sus unidades mínimas de su gramática y esto obliga a conocerlo, para aplicar el método de Tokenización y Segmentación adecuado.

   Por ejemplo, en la mayoría de los idiomas, la separación de palabras se realiza con espacios y la separación de oraciones con puntos. Eso facilita la identificación de las palabras y tokens, así como el conjunto de oraciones en el que se organiza un texto método de Tokenización y Segmentación), frente a los idiomas que no separan las palabras y tokens con espacios, y tampoco separan las oraciones con puntos.

   En un **Corpus**, no todos los fragmentos de texto están bien formados, por lo que deben prepararse antes del método de Tokenización y Segmentación. Algunos ejemplos:

   - Deben ser eliminadas todas las etiquetas HTML de los textos que provienen de la Web.
   - Deben corregirse los textos con errores ortográficos, con el fin de que las siguientes fases de análisis trabajen con textos de la máxima calidad.

- Deben ser traducidas las onomatopeyas y abreviaturas a sus vocablos correspondientes de los textos provenientes de redes de microblogging.

Determinar en cierta manera el tratamiento más adecuado sobre las palabras y tokens, y sobre las oraciones, es un método que depende de la **Aplicación** y del procedimiento posterior.

Por ejemplo, en la lengua castellana, ¿que sería más adecuado para el tratamiento de las palabras unidas con guion como mini-USB, departamento científico-técnico, pro-Obama, etc. separar los términos o considerar un único término unido por un guion? Pues eso dependerá de su Aplicación.

4. Explique las diferencias existentes entre un analizador y un generador léxico. Se valorará positivamente la descripción de ejemplos de analizadores y generadores léxicos.

   Un analizador léxico tiene la capacidad de analizar e identificar toda la información relacionada con un término. En cambio, un generador léxico realiza la operación contraria al análisis, a partir de un lema puede generar toda la información relacionada con un término.

   El analizador léxico generalmente se basa en una máquina de estados finitos que codificada dentro de ella información sobre los posibles lexemas que pueden estar contenidos dentro de cualquiera de los tokens que maneja.

   Un ejemplo de analizador léxico es Lex. El programa propietario Lex es el analizador léxico estándar en los sistemas Unix, y se incluye en el estándar de POSIX. Lex toma como entrada una especificación de analizador léxico y devuelve como salida el código fuente implementando el analizador léxico en C. Se utiliza comúnmente con el programa Yacc, un generador léxico implementado en C, que se sirve para generar análisis sintáctico. La versión libre de Lex es Flex .

   Un ejemplo de generador léxico como Yacc, comentado anteriormente, es Bison . Bison pertenece al proyecto GNU y está disponible para prácticamente todos los sistemas operativos, y se usa normalmente acompañado de Flex aunque es compatible con otros analizadores léxicos. Su función es convertir la descripción formal de un lenguaje en un programa en C, C++, o Java que realiza análisis sintáctico.

   Tiene compatibilidad total con Yacc e igual que este, necesita un analizador léxico y normalmente se utiliza juntamente con Flex.

5. La lección se ha centrado en los analizadores sintácticos basados en Gramáticas Libres de Contexto, por lo que ¿podría describir las operaciones fundamentales de un analizador LR (0)?

   Un analizador LR (0) realiza el análisis sintáctico a través las operaciones fundamentales: desplaza, reduce y Goto.

   - La operación desplaza se produce por la lectura de un símbolo terminal válido atendiendo a las reglas de la gramática. El desplazamiento se realiza de izquierda a derecha.
   - Cuando se está al final de un elemento (regla) se utiliza la operación reduce (substituye) a los símbolos terminales por la parte izquierda de su elemento (regla). Ya que, para la reducción se aplica la derivación más a la derecha.
   - La operación Goto consiste en un desplazamiento provocado por símbolos no terminales y pueden dar lugar a una reducción.

6. Defina qué es la identificación de roles semánticos, así como qué es FrameNet. Recuerde que se valorarán positivamente la presentación de ejemplos.

   Uno de los tipos de relaciones semánticas que existen entre el significado de los términos que constituyen un enunciado, son los roles semánticos. Que identifican y relacionan los agentes de una oración, es decir, los actores y las acciones descritas.

   Enfoques de los roles semánticos:

   - Localistas
     - Fueron introducidos por Gruber en 1965.
     - Contienen un número reducido de roles.
     - Se centran en roles con relaciones entre ellos del tipo: agente-acción.
     - De cada relación simple entre los roles surgen relaciones más abstractas.
   - No localistas
     - Fueron introducidos por Fillmore en 1968.
     - Mayor número de roles.
     - Normalmente son dirigidos por subclases verbales.

   Un recurso electrónico basado en roles semánticos no localistas es FrameNet y es en gran medida una creación de Fillmore.

   Actualmente FrameNet es un proyecto está construyendo una base de datos léxica en inglés que es legible tanto por humanos como por máquinas, basada en ejemplos de anotaciones de cómo se usan las palabras en textos reales. Las más de 200.000 oraciones que contiene proporcionan un conjunto de datos de entrenamiento único para el etiquetado de roles semánticos.

   Los datos de la base de datos léxica están disponibles gratuitamente para su descarga y también se han creado bases de datos similares para otros idiomas. Por ejemplo, en castellano SFN , alemán GF y en japonés JFN.