ALUMNO: JUAN CARLOS MENDOZA DEL VALLE

MATERIA: INTELIGENCIA ARTIFICIAL (I.A.)

PROFESOR: JOSÉ JULIO GONZÁLEZ ÁLVAREZ

TAREA: INVESTIGACIÓN PLN

Procesamiento de lenguajes naturales

La perspectiva computacional

Como se deduce del título de este artículo, mi objetivo es estudiar el problema del procesamiento del lenguaje natural (PLN) desde una perspectiva computacional. Dicho de otro modo, e incorporando alguna idea expuesta en la introducción, sea cual sea nuestro objetivo último, la mejor manera de abordar el problema del PLN es desde una perspectiva computacional y con métodos computacionales. Pero, ¿qué comporta la adopción de la perspectiva computacional? La respuesta a esta pregunta es mucho menos evidente de lo que podría parecer al principio, ya que si pensamos que ésta debe incluir alguna referencia a los ordenadores, entonces estaremos equivocados. Es cierto que los ordenadores son dispositivos especialmente diseñados para llevar a cabo tareas que podríamos calificar de computacionales, pero no son los únicos: los cerebros también son dispositivos computacionales, y también lo son los ábacos e, incluso, nuestra mano, armada de lápiz y papel, es un dispositivo computacional. Una interesante propiedad de los procesos computacionales es que podemos analizarlos desde una perspectiva que es totalmente independiente del mecanismo que debe implementarlos. Con ello no quiero decir que la implementación sea irrelevante, que no lo es, sino que podemos aprender mucho de un problema computacional sin tener en cuenta los detalles relacionados con la implementación. Esta es, quizás, una de las enseñanzas más importantes del legado intelectual del matemático británico Alan Turing, que convendrá tener presente en todo momento.

Volvamos ahora a nuestra pregunta: ¿qué comporta la adopción de la perspectiva computacional? Bueno, aquí voy a necesitar que alguien me eche una mano, así que citaré unas palabras de uno de los más célebres precursores de las ciencias cognitivas, el psicólogo norteamericano, ya fallecido, David Marr (1982, pp. 24-25):5

«[T]he different levels at which an information processing device must be understood [are:] At one extreme, the top level, is the abstract computational theory of the device, in which the performance of the device is characterized as a mapping from one kind of information to another, the abstract properties of this mapping are defined precisely, and its appropriateness and adequacy for the task at hand are demonstrated. In the center is the choice of representation for the input and output and the algorithm to be used to transform one into the other. And at the other extreme are the details of how the algorithm and representation are realized physically—the detailed computer architecture, so to speak.»6

He destacado algunos términos en esta cita, ya que nos servirán para definir el hilo conductor que vamos a seguir a partir de ahora. El objetivo principal de David Marr era elaborar un modelo de la visión humana, pero, y aquí es donde reside la belleza de las ideas de Turing, esto es irrelevante, porque una de las ventajas del enfoque computacional es que nos ofrece un marco general desde

el cual podemos estudiar fenómenos en apariencia muy diferentes. Ya tenemos, por tanto, unas cuantas cosas importantes que decir acerca del PLN, concretamente que:

• Puede definirse como una tarea de procesamiento de la información que caracterizaremos como una proyección desde una representación de entrada en una representación de salida, de tal modo que dicha proyección pueda describirse mediante un algoritmo determinado.

Mantengo las negritas porque, en lo que queda de artículo, me ocuparé de analizar estos tres elementos con cierto detalle, poniendo de relieve las relaciones que existen entre ellos. Quisiera empezar con una analogía que nos será de utilidad más adelante. Considere el lector una actividad artística como, por ejemplo, la escultura. En un estadio preliminar de la producción de una pieza, el artista diseña su proyecto como una proyección abstracta entre un determinado bloque de materia (aún por determinar) y una forma que sólo existe en su mente o en forma de bocetos sobre papel. En un segundo estadio, el escultor debe decidir qué tipo de material va a utilizar. En esta decisión interviene toda una serie de consideraciones que no nos incumben en este momento (de tipo estético o relacionada con las preferencias y las capacidades del autor), pero, independientemente de si la escultura se va a realizar en madera, piedra o metal, esta decisión repercute definitivamente en el proceso de producción, ya que cada material debe tratarse con las técnicas apropiadas (es decir, la piedra y la madera se puede tallar, pero no fundir, por ejemplo); y viceversa, si nuestro escultor desea utilizar una técnica específica, entonces no podrá utilizar cualquier material.

La idea que me interesa derivar de este ejemplo (y del siguiente que discutiré) es que, en muchas ocasiones, la naturaleza de las entidades que participan en un proceso determina el modo en que el proceso se lleva a cabo, y viceversa, si queremos hacer algo de una manera determinada, impondremos una restricción inevitable sobre el tipo de entidades que podremos manipular. Traducida a términos computacionales, esta analogía puede reformularse como sigue: la elección de un formato o modo de representación, condicionará la elección del algoritmo, y viceversa. Veamos otro ejemplo, más próximo al caso que nos ocupa.

Esta es una historia de números. Hace ya mucho tiempo que los humanos conocen y utilizan los números, que saben de sus propiedades y aplicaciones. Para usarlos, sin embargo, hemos tenido que buscar una manera de representarlos. Por lo que yo sé, nadie ha visto nunca un número, podrían ser peludos y azules o lisos y rojos, ¡qué más da! Lo importante es que, para usarlos, tenemos que representarlos; mientras el modo de representación que elijamos conserve las propiedades relevantes de los números, que nos permiten hacer cosas tales como sumar o multiplicar, no hay problema.

En Occidente, los griegos y los romanos fueron los primeros que descubrieron muchas de las propiedades y aplicaciones de los números. Los griegos, por ejemplo, descubrieron los números irracionales, como p y Ö2, que, para desesperación de los pitagóricos, son números que no se pueden expresar como la razón de dos números enteros, como ocurre con 1/2. Resulta sorprendente que hicieran tales descubrimientos (lo cual dice mucho en favor de los matemáticos e ingenieros griegos y romanos), ya que ambos utilizaban un sistema bastante curioso para representar los números. Tomemos como ejemplo el sistema romano que todos conocemos (el griego se fundamentaba en los mismos principios).

El sistema se basa en una serie de siete símbolos que representan unos números concretos, más una serie de reglas para combinar esos símbolos a fin de poder construir símbolos más complejos

En cuanto a las reglas de combinación, todos sabemos más o menos cómo funcionan: a) un símbolo a la derecha de otro tiene función aditiva, b) un símbolo a la izquierda de otro tiene función sustractiva. Con este modo de representación, la suma y la resta son operaciones relativamente fáciles de llevar a cabo, ya que, de hecho, el sistema se basa precisamente en las nociones de suma y resta. (Una manera de pensar en los numerales romanos complejos es como un modo de escribir sumas, con signos de suma y resta implícitos que conectan los diversos símbolos.) Pero, ¿qué hay de otras operaciones más complejas como la multiplicación o la división? La multiplicación es una auténtica pesadilla; la división es, simplemente, imposible. Para que el lector pueda apreciar la verdadera dimensión del problema, veamos cómo funciona el algoritmo de la multiplicación calculando, en romanos, 32 x 46, en la tabla 2.

Ahora ya sabe el lector por qué los números romanos sólo se usan para indicar la fecha de copyright de las películas de Hollywood. No sirven para otros propósitos más prácticos. Si pensamos ahora en el sistema que utilizamos hoy en día, el sistema árabigo, veremos que, a partir de la misma idea de utilizar símbolos para representar los números, este sistema emplea un conjunto de principios para combinarlos radicalmente distinto, además de reconocer la existencia de una entidad bastante útil, el cero. Es un sistema posicional basado en la suma y la multiplicación, de modo que cada cifra, según sea su posición, nos indica la cantidad de unidades, decenas, centenas, etc. (por ejemplo, n x 1, n x 10, n x 100, etc.). Con este sistema de representación, es muy fácil efectuar cualquiera de las cuatro operaciones elementales; nos basta con elegir nuestro algoritmo favorito para verificar que, efectivamente, 32 x 46 es igual a 1472.

De vuelta al asunto que nos ocupa, vemos que nuestras historias de escultores y de números tienen moraleja: cuando caracterizamos un proceso computacional, debemos ser muy cuidadosos en el momento de escoger un formato para las representaciones de entrada y de salida, y en el momento de seleccionar un algoritmo capaz de describir el proceso de transformar las unas en las otras. No vale cualquier cosa, ya que no es cierto que la forma de nuestras representaciones no tenga ninguna repercusión sobre la manera cómo van a ser procesadas. En el momento de elegir un formato apropiado para nuestras representaciones debemos procurar no ser víctimas de la TNR («trampa de los números romanos), y la única manera de evitarlo es conocer muy bien cuáles son las propiedades formales del lenguaje de representación que vamos a utilizar. Es la única manera de determinar qué algoritmo será el más apropiado e, incluso, de saber si hay un algoritmo. Resumamos, pues, los tres elementos básicos de nuestra teoría computacional:

El procesamiento de lenguajes naturales —abreviado PLN, o NLP del idioma inglés Natural Language Processing— es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. El PLN se ocupa de la formulación e investigación de mecanismos eficaces computacionalmente para la comunicación entre personas y máquinas por medio de lenguajes naturales. El PLN no trata de la comunicación por medio de lenguajes naturales de una forma abstracta, sino de diseñar mecanismos para comunicarse que sean eficaces computacionalmente —que se puedan realizar por medio de programas que ejecuten o simulen la comunicación—. Los modelos aplicados se enfocan no solo a la comprensión del lenguaje de por sí, sino a aspectos generales cognitivos

humanos y a la organización de la memoria. El lenguaje natural sirve solo de medio para estudiar estos fenómenos. Hasta la década de 1980, la mayoría de los sistemas de PLN se basaban en un complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, sin embargo, hubo una revolución en PLN con la introducción de algoritmos de aprendizaje automático para el procesamiento del lenguaje.

Historia

La historia del PLN empieza desde 1950, aunque existe trabajo encontrado desde periodos anteriores. En 1950, Alan Turing publicó Computing machinery and intelligence el cual proponía lo que hoy llamamos test de turing como criterio de inteligencia. El experimento de Georgetown en 1954 involucró traducción automática de más de sesenta oraciones del Ruso al Inglés. Los autores sostuvieron que en tres o cinco años la traducción automática seria un problema resuelto. El progreso real en traducción automática fue más lento y después del reporte ALPAC en 1996, el cual demostró que la investigación había tenido un bajo desempeño. Más tarde investigación a menor escala en traducción automática se llevó a cabo hasta finales de 1980, cuando se desarrollaron los primeros sistemas de traducción automática estadística. Esto se debió tanto al aumento constante del poder de cómputo resultante de la Ley de Moore y la disminución gradual del predominio de las teorías lingüísticas de Noam Chomsky (por ejemplo, la Gramática Transformacional), cuyos fundamentos teóricos desalentaron el tipo de lingüística de corpus, que se basa el enfoque de aprendizaje de máquinas para el procesamiento del lenguaje. Algunos de los primeros algoritmos de aprendizaje automático utilizados, tales como árboles de decisión, sistemas producidos de sentencias si-entonces similares a las reglas escritas a mano.

Dificultades en el procesamiento de lenguajes naturales

Gramáticas Formales

Hay una clase de sistemas de generación de interés primario para los Informáticos – ellos son los sistemas conocidos como Gramáticas.

El concepto de Gramática fue originalmente formalizado por los lingüistas en su estudio de los lenguajes naturales

Los lingüistas tenían relación no sólo con la definición precisa de lo que es o no es una sentencia u oración válida de un lenguaje, sino también de dar o suministrar descripciones estructurales de las sentencias u oraciones

Uno de estos objetivos estuvo relacionado con el desarrollo de una Gramática Formal capaz de describir la lengua inglesa

Se podría pensar, que si por ejemplo, se tiene una gramática formal para describir la lengua inglesa, podríamos usar el computador en los campos que necesiten una comprensión de la lengua inglesa

Tal uso puede ser la traducción de de lenguajes o la solución computacional de problemas de enunciados

Hasta el momento actual, este objetivo sigue siendo en gran parte irrealizable

Aun no se dispone de una gramática bien definida de la lengua inglesa.

Además, existen contradicciones sobre que tipo de gramática formal seria capaz de describir al idioma Ingles.

Sin embargo, han sido alcanzados mejores resultados en la descripción de los lenguajes de computación

Por ejemplo, la Forma Backus – Naur usada para describir el lenguaje de programación ALGOL es una "gramática de libre contexto ", esto es, un tipo de gramática con la que tendremos relación en esta disciplina.

Existe costumbre de realizar diagramas o análisis (parsing) de una sentencia u oración inglesa

Por ejemplo, la sentencia u oración :"The little boy ran quickly "

se analiza (parsed) por medio de la notación de que la oración consiste de:

nome (noun phrase):

"The little boy"

seguido de la frase verbal (verb phrase)

"ran quickly"

El nombre puede ser decompuesto en nombre singular "boy" modificado por dos adjetivos:

"The" y

"little"

La frase verbal puede ser decompuesta, a su vez, en un verbo singular

"ran"

modificado por el adverbio

"quickly"

Esta estructura de la oración es indicada en el siguiente diagrama.

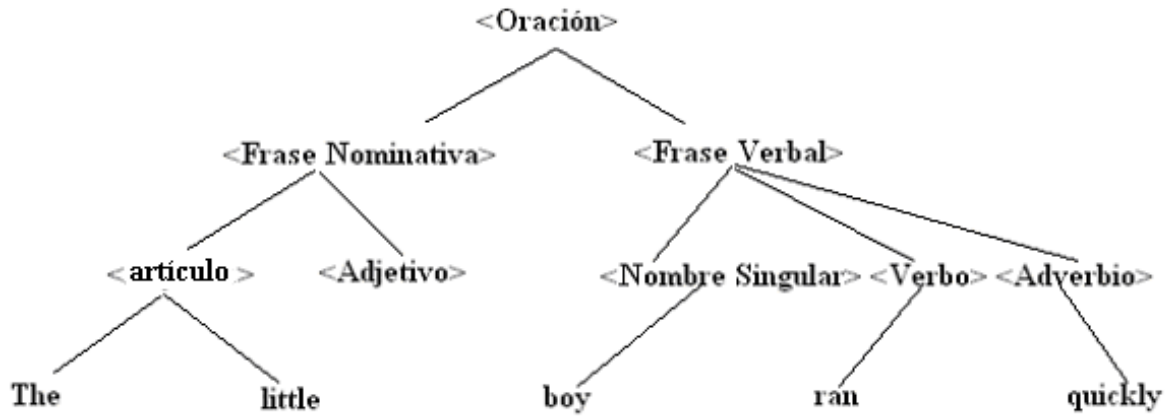

Figura Árbol Sintáctico de una oración

Se reconoce la estructura de la sentencia u oración como gramaticalmente correcta.

Si se tiene un conjunto completo de negras para analizar (parsing) todas las oraciones en idioma Ingles, entonces podríamos tener una técnica para determinar si la oración es o no gramaticalmente correcta. Sin embargo, tal conjunto de negras realmente no existe. En parte, esto se debe a que no existen reglas claras y precisas para determinar lo que constituye una oración:

<sentencia u oracion> ® < nombre> <frase verbal>

<frase verbal> ® <adjetivo> <frase nominativa>

<frase nominativa> ® <adjetivo> <nombre singular>

<frase verbal> ® <verbo singular> <adverbio>

<adjetivo> ® The

<adjetivo> ® little

<nombre singular> ® boy

<verbo singular> ® ran

<adverbio> ® quickly

La flecha indica que el elemento de la izquierda de la flecha puede generar los elementos colocados en el lado derecho de la flecha. Note que se ha encerrado entre corchetes los nombres de las partes de las oraciones, tales como, nombre, verbo, frase verbal, etc., para evitar confusión con las palabras en Ingles y las frases "nombre", "frase verbal", etc. Se puede notar que no es sólo posible verificar las oraciones por su correlación gramatical, sino también es posible generar oraciones correctas gramaticalmente. Para ello se comienza con la cantidad <oración> y se sustituye <oración> por <frase nominativa> seguida de <frase verbal> . Luego se selecciona una de las dos reglas para <frase nominativa> y se aplica, y así sucesivamente , hasta que ninguna otra

aplicación adicional de las negras sea posible. En esta forma, un número infinito de oraciones puede ser derivada – esto es, cualquier oración consistente de una cadena de ocurrencias de "the" y "little" seguido por "boy ran quickly" tal como "little the the boy ran quickly" puede generarse. La mayoría de las oraciones no tiene sentido, son gramaticalmente correctas en un sentido amplio.

Definiciones empleadas en las gramáticas formales.

Alfabeto: Un alfabeto es un conjunto arbitrario, pero finito, de símbolos.

Por ejemplo, el código de maquina se basa en el alfabeto binario A1={0,1}; otros ejemplos son A2{0,1,2,3,4,5,6,7,8,9}, A3{+,-,*,/,} etc.

Símbolos: Los elementos del vocabulario (alfabeto) de un lenguaje formal se denominan símbolos; en el caso de los lenguajes naturales los conocemos como palabras.

Componente Léxico: las ocurrencias múltiples de símbolos (o palabras) se denominan componentes léxicos.

Frase: Una frase es una secuencia de símbolos.

Gramática (sintaxis): La gramática o la sintaxis de un lenguaje define si una secuencia arbitraria de símbolos es correcta, es decir, si es una frase significativa. Decimos que una frase correcta será aceptada por el lenguaje.

Cadena: Sentencia (finita) de elementos de un cierto conjunto (alfabeto).

Producción: Las reglas para la sustitución de cadenas se denominan producciones.

Símbolos terminales: Son los símbolos que realmente aparecen en una frase.

Símbolos no terminales: Los símbolos no terminales deben ser definidos por otras producciones o reglas ; es decir, también aparecen en el lado izquierdo de las producciones. Los símbolos no terminales son variables sintácticas.

Vocabulario = alfabeto: Al igual que los lenguajes naturales, los lenguajes formales se basan en un vocabulario específico, a saber, los elementos del lenguaje.

Forma de Backus – Naur

La forma de Backus – Naur fue creada para definir la estructura del lenguaje de programación ALGOL60.

Tabla Forma Backus – Naur

Símbolo	Significado
	"se define como" **fin de definición**
|	**"or", alternativa**
[x]	**Una o ninguna ocurrencia de x**
{x}	**Número arbitrario de ocurrencias de x (0,1,2,...)**
(x | y)	**Selección (x o y)**

La forma Backus – Naur es un metalenguaje, o sea, un lenguaje con el que se pueden describir otros lenguajes. Hay algunos dialectos de la notación BNF. En la tabla se presentan algunos de los símbolos más comunes de la BNF. Con esa notación y los símbolos terminales.

T= {+,-, 0,1,2,3,4,5,6,7,8,9}

Además de los símbolos no terminales

N= {int, unsigned_int, digit}

Podemos definir los enteros con las siguientes reglas (producciones) BNF:

int à [+ | -] unsigned_int

unsigned_int à digit unsigned_int digit.

digit à 0|1|2|3|4|5|6|7|8|9|

La primera regla define un entero como un entero sin signo mas un signo inicial. Este signo puede estar ausente o ser "+" o "-". La segunda regla indica que la notación BNF permite definiciones recursivas.

Existe una descripción formal de un lenguaje si existe un número finito de reglas BNF que permiten derivar cualquier frase del lenguaje. En este aspecto, el conjunto finito de reglas anterior es una descripción formal del conjunto infinito de los enteros.

Conclusiones

El procesamiento del lenguaje natural tiene como objetivo fundamental lograr una comunicación máquina-humano similar a la comunicación humano-humano.

El empleo del lenguaje le permite al hombre trasmitir sus conocimientos, sentimientos, sensaciones, emociones, y estados de ánimo

A lo largo de la historia los lenguajes naturales han ido evolucionando, de forma paralela al desarrollo y evolución de la especie humana.

Han sido varios los sistemas informáticos inteligentes que se han desarrollado que emplean el procesamiento del lenguaje natural.

Ambigüedad

El lenguaje natural es inherentemente ambiguo a diferentes niveles:

A nivel léxico, una misma palabra puede tener varios significados, y la selección del apropiado se debe deducir a partir del contexto oracional o conocimiento básico. Muchas investigaciones en el campo del procesamiento de lenguajes naturales han estudiado métodos de resolver las ambigüedades léxicas mediante diccionarios, gramáticas, bases de conocimiento y correlaciones estadísticas.

A nivel referencial, la resolución de anáforas y catáforas implica determinar la entidad lingüística previa o posterior a que hacen referencia.

A nivel estructural, se requiere de la semántica para desambiguar la dependencia de los sintagmas preposicionales que conducen a la construcción de distintos árboles sintácticos. Por ejemplo, en la frase Rompió el dibujo de un ataque de nervios.

A nivel pragmático, una oración, a menudo, no significa lo que realmente se está diciendo. Elementos tales como la ironía tienen un papel importante en la interpretación del mensaje.

Para resolver estos tipos de ambigüedades y otros, el problema central en el PLN es la traducción de entradas en lenguaje natural a una representación interna sin ambigüedad, como árboles de análisis.

Detección de separación entre las palabras

En la lengua hablada no se suelen hacer pausas entre palabra y palabra. El lugar en el que se debe separar las palabras a menudo depende de cuál es la posibilidad que mantenga un sentido lógico tanto gramatical como contextual. En la lengua escrita, idiomas como el chino mandarín tampoco tienen separaciones entre las palabras.

Recepción imperfecta de datos

Acentos extranjeros, regionalismos o dificultades en la producción del habla, errores de mecanografiado o expresiones no gramaticales, errores en la lectura de textos mediante OCR

Componentes

Análisis morfológico. El análisis de las palabras para extraer raíces, rasgos flexivos, unidades léxicas compuestas y otros fenómenos.

Análisis sintáctico. El análisis de la estructura sintáctica de la frase mediante una gramática de la lengua en cuestión.

Análisis semántico. La extracción del significado de la frase, y la resolución de ambigüedades léxicas y estructurales.

Análisis pragmático. El análisis del texto más allá de los límites de la frase, por ejemplo, para determinar los antecedentes referenciales de los pronombres.

Planificación de la frase. Estructurar cada frase del texto con el fin de expresar el significado adecuado.

Generación de la frase. La generación de la cadena lineal de palabras a partir de la estructura general de la frase, con sus correspondientes flexiones, concordancias y restantes fenómenos sintácticos y morfológicos.

Aplicaciones

Las principales tareas de trabajo en el PLN son:

- Síntesis del discurso
- Análisis del lenguaje
- Comprensión del lenguaje
- Reconocimiento del habla
- Síntesis de voz
- Generación de lenguajes naturales
- Traducción automática
- Respuesta a preguntas
- Recuperación de la información
- Extracción de la información

REFERENCIAS

https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales
https://www.monografias.com/trabajos17/lenguaje-natural/lenguaje-natural.shtml
http://quark.prbb.org/19/019035.htm