

Técnicas de Procesamiento del Lenguaje Natural. Primera parte. Fundamentos

MÁSTER EN INTELIGENCIA ARTIFICIAL
UNIVERSIDAD DE ALICANTE

Borja Navarro Colorado

14 de diciembre de 2023

Índice general

1	Introducción	5
2	Procesamiento lingüístico	7
2.1.	Introducción	7
2.2.	El texto	8
2.3.	Procesamiento computacional: representación formal y método de análisis	18
2.4.	Conclusiones	23
2.5.	Lecturas opcionales	24
3	Análisis categorial	25
3.1.	Unidades de comunicación básica. La palabra. <i>Type, token</i> y lema.	25
3.2.	Lematización y <i>stemming</i>	26
3.3.	Análisis morfológico y categorial.	27
3.4.	Ambigüedad categorial y proceso de análisis	30
3.5.	Recursos.	30
3.6.	Lecturas opcionales	31
4	Sintaxis	33
4.1.	Análisis de constituyentes	33
4.2.	Análisis de dependencias	35
4.3.	Estrategias de análisis	40
4.4.	Herramientas	42
4.5.	Lecturas opcionales	42
5	Semántica	43
5.1.	Significado como representación lógica	44

5.2. Semántica léxica	44
5.3. Semántica oracional	50
5.4. Lecturas opcionales	55
6 Semántica vectorial	57
6.1. Lecturas	57
6.2. Introducción	58
6.3. La semántica distribucional	58
6.4. Espacio vectorial como modelo de representación formal . .	60
6.5. Interpretación semántica: distancia y similitud.	68
6.6. Conclusiones	69
6.7. Situación actual	69
6.8. Herramientas y recursos	69
7 Temas abiertos en PLN	71
7.1. Introducción	71
7.2. Más allá de la alquimia	71
7.3. Composicionalidad	72
7.4. Metáforas	72
7.5. Neural-Symbolic AI	73
7.6. Texto narrativos	73
Bibliografía	75

Introducción

Este documento¹ es la guía de estudio de la primera parte de la asignatura “Técnicas de Procesamiento del Lenguaje Natural” del *Máster en Inteligencia Artificial* de la Universidad de Alicante.

En el documento hallarás las ideas principales de cada tema (en modo más o menos esquemático) y lecturas asociadas para ampliar y completar la información. El documento está redactado en español, pero las lecturas asociadas serán en inglés.

Web de la asignatura: <https://jaspock.github.io/tpln2324/cl/>

¹Agradecimientos a Juan Antonio Pérez Ortiz por la revisión del texto y los oportunos comentarios realizados, que sin duda han mejorado la calidad del texto. Cualquier error y omisión es responsabilidad mía.

Introducción al procesamiento de un texto

En este capítulo veremos:

- Qué es un texto.
- Cómo se procesa desde un punto de vista lingüístico-teórico.
- Principales problemas computacionales: representación formal y métodos.

2.1. Introducción

Con “Procesamiento del Lenguaje Natural” (PLN) se hace referencia a todos aquellos aspectos de la inteligencia artificial relacionados con la capacidad de comunicación humana mediante una lengua natural.

En general, un sistema de PLN es una emulación computacional de la capacidad humana para generar e interpretar textos en un idioma concreto. El sistema trata de generar o interpretar textos tal y como lo haría un ser humano. Esto no quiere decir que los métodos para crear/interpretar textos deban ser los mismos métodos cognitivos utilizados por los humanos. Como se verá a lo largo de la asignatura, las técnicas de PLN pueden imitar o estar inspiradas en (lo que conocemos de) los modelos lingüísticos y cognitivos humanos, o simplemente utilizar modelos computacionales propios. En cualquier caso, el resultado es un texto y/o su interpretación “como si lo hubiera hecho un humano”; es decir, que un humano no pueda decir si el texto o su interpretación la ha realizado otro humano o una máquina.

Así, un sistema de PLN puede actuar en tres escenarios:

1. Sistemas en el que el texto es la entrada (*input*) del sistema. En este caso, el proceso computacional es un proceso de interpretación automática. Ejemplo de este escenario son sistemas de detección de opiniones y emociones, sistemas de extracción de información, sistemas *text-to-image* o *text-to-video* (generación automática de imágenes o vídeo, respectivamente, a partir de un texto), etc.
2. Sistemas en los que el texto es la salida (*output*) del sistema. En este caso, el proceso computacional es la generación o creación del texto. Ejemplos de este escenario son los sistemas de descripción de imágenes, la generación de texto a partir de plantillas de información o la creación automática de poesía, entre otros.
3. Sistemas en los que tanto la entrada como la salida serán textos. En estos casos deben darse ambos procesos: la interpretación y la generación (o a la inversa, según la finalidad del sistema). Ejemplos de sistemas texto-texto son los sistemas de traducción automática, resumen automático, sistemas de diálogo (chat humano-máquina), etc.

El término “procesamiento del lenguaje natural” es hoy el más común para esta área. Desde la lingüística se le suele denominar aún “Lingüística Computacional” (LC), que es término más tradicional. En ocasiones se utiliza LC para los aspectos más lingüísticos del PLN (gramáticas computacionales, análisis de rasgos, anotación de corpus, etc.). Otros términos que se utilizan son “procesamiento del lenguaje humano”, “ingeniería lingüística” o “comprensión del lenguaje natural”, entre otros.

Como vemos, los sistemas de PLN giran en torno al texto, que es la unidad de comunicación lingüística humana. Antes de profundizar en las técnicas de PLN para procesar textos, se va a exponer qué es un texto y cómo es su procesamiento desde un punto de vista lingüístico (teórico-cognitivo). En algunos casos se hará alusión a sus implicaciones computacionales (que se tratarán con más detalle en siguientes capítulos).

2.2. El texto

El texto es la principal unidad de comunicación humana.

Cuando hablamos de un texto no nos referimos solo al texto escrito, sino a cualquier comunicado concreto y perceptible que se emita en una situación comunicativa determinada y que utilice signos lingüísticos (una lengua natural) como principal medio de transmisión de significado.

Así, un texto puede ser una sola palabra emitida de manera oral. Por ejemplo, un grito de "socorro."^{en} una situación de peligro. O también un texto puede estar formado por miles de palabras en formato escrito. Por ejemplo, toda la novela *El ingenioso hidalgo don Quixote de La Mancha* es un único texto. En ambos casos estamos ante un texto.

El texto, en este sentido, es un objeto concreto perceptible por los sentidos. En concreto, un texto puede ser percibido por el oído (texto oral), por la vista (texto escrito) o por ambos a la vez. Luego se comentará más sobre esto.

La comunicación

Este objeto texto tiene sentido dentro de una situación comunicativa, es decir, solo dentro de una situación comunicativa ese objeto se puede interpretar, donde se le puede asignar un significado.

Sin querer profundizar aquí en el esquema general de una situación comunicativa (que asumimos es algo conocido), se deben tener en cuenta los siguientes aspectos:

- Todo texto es creado por un agente productor (en principio humano, pero ver luego) con una intención comunicativa concreta, es decir, con la finalidad de conseguir algo con ese texto.
- Para que el texto tenga sentido, debe ser interpretado por un agente receptor (en principio humano también, pero no solo). Este agente receptor interpreta el texto también con una finalidad propia, para conseguir algo de esa interpretación.
- La intención de productor y receptor no tienen por qué coincidir, ni si quiera deben ser compatibles.
- El texto como objeto físico necesita un soporte por el que se transmite. Independientemente de que sea digital o analógico, lo importante es si el soporte es auditivo (habla) o visual (texto escrito).

Un sistema de Procesamiento del Lenguaje Natural asume uno de esos papeles: puede ser el agente productor o el agente receptor. La única diferencia es la intención: los sistemas de PLN no tienen intención comunicativa en sí misma, sino que la intención sigue siendo humana.

Componentes de un texto y criterios de textualidad

Hasta ahora hemos considerado el texto como un objeto físico perceptible dentro de una situación comunicativa. Sin embargo, el texto se considera tal y no un simple objeto cuando:

1. un ser humano lo percibe como texto y por tanto con capacidad para ser interpretado, y
2. un ser humano lo interpreta y, con ello, le aporta significado.

Si el texto no significa, es decir, no hay un ser humano que lo interprete y genere un significado a partir de ese objeto físico (el sonido o la imagen), no podemos hablar de texto propiamente dicho.

En esta consideración el texto ya no es tanto un objeto físico como un objeto cognitivo. En tanto que entidad cognitiva, un texto es un signo: una señal visual o acústica cargada de significado,¹ y por tanto que se puede interpretar a partir de un código de interpretación.

Veamos, por ejemplo, una simple palabra: FUEGO. En su materialidad visual, es solo una secuencia de caracteres:

F U E G O

Nosotros lo percibimos como caracteres interpretables, pero para la máquina son secuencias de bits. En código ASCII sería:

70 86 69 71 79

Cuando una mente humana ve esa secuencia de caracteres, si conoce el código interpretativo (en este caso, la lengua española), es entonces capaz de interpretarlo: es capaz de aplicarle un significado. En este caso, el significado es el concepto cognitivo de “fuego”, algo así como la imagen mental del fuego. Más o menos (según cada persona), lo que hay en la Figura 2.1²:

Un texto es un signo, por tanto, porque ante una secuencia de sonidos o ante una imagen, una mente humana es capaz de generar una imagen mental, relacionar esos sonidos o esas imágenes con conceptos. El texto como objeto físico no es el texto en sí mismo, sino solo una parte de él.

Así, desde un punto de vista ya lingüístico y cognitivo, el texto se define como un complejo sígnico relacional.

¹https://encyclopaedia.herdereditorial.com/wiki/Recurso:Eco,_Umberto:_el_signo

²<https://pixabay.com/es/photos/fuego-chimenea-madera-negro-fuego-1159157/>



Figura 2.1: fuego

Es un complejo s gnico porque est  formado por un conjunto de signos, no solo uno. Los caracteres, fonemas, palabras, etc. son signos. En un texto aparecen interrelacionados de tal manera que a todo ese conjunto de signos se le puede asignar un significado completo y coherente.

Es relacional porque todo ese conjunto de signos est  basado en la relaci n entre un objeto y su interpretaci n, es decir, entre un significante (el objeto percibido que puede ser interpretado) y un significado (la interpretaci n que realiza un ser humano, la “imagen” mental). En el caso anterior, la secuencia de caracteres FUEGO es el significante y el concepto de fuego el significado.

A partir de estos dos conceptos de significante y significado, la ling  stica moderna ha desarrollado cuatro componentes b sicos de todo texto: la imagen mental del objeto perceptible, una estructura formal que especifica la relaci n entre los signos que forman el texto, una estructura sem ntica (significado) a partir del significado de cada elemento formal del texto y sus relaciones, y la imagen mental del estado de cosas expresado en el texto (referente).

Imagen mental del texto

El texto es un concepto cognitivo, por tanto el objeto f sico en s  mismo no es el texto como se dijo antes, sino la imagen mental que un ser humano realiza de ese objeto f sico (de los sonidos o de la imagen de las letras). Es esa imagen mental la que ser  interpretada. El primer paso interpretativo est  ah : en la consideraci n de esos sonidos o im genes como elementos ling  sticos que pueden ser interpretados.

Desde un punto de vista computacional, los sistemas de PLN deben realizar tambi n este paso. La foto de un texto no es en s  un texto porque no es interpretable: son solo p xeles. Para que sea interpretable (por la m quina), se debe transformar la imagen de cada letra en su correspondiente car cter, es decir, transformar los p xeles en c digo ASCII (o UTF-8 o el sistema de codificaci n que sea). Hasta que no est  codificada la imagen de cada car cter,

no podemos hablar de texto digital propiamente dicho, sino de la imagen de un texto. Este proceso computacional lo realizan los sistemas de OCR (*optical character recognition*). Si la entrada es sonido se debe realizar un proceso similar pero más complejo: primero se debe discriminar qué sonidos son lingüísticos (y por tanto interpretables) del resto de sonidos (ruido). Una vez aislada la cadena fónica, se debe segmentar y asignar a cada sonido su codificación correspondiente (ASCII o lo que sea). Esta tarea se denomina *authomatic speech recognition* (ASR)³. Hoy día, cualquier móvil cuenta ya con un sistema de ASR.

Estructura formal

El segundo componente es la estructura formal. La estructura formal de un texto son las relaciones que se establecen entre las diferentes unidades del texto. Determinar la estructura formal de un texto implica, primero, definir las unidades lingüísticas y, segundo, las relaciones que haya entre ellas.

Hay que tener clara la diferencia entre la estructura formal que se asume tiene todo texto, de la estructura formal que nosotros podemos representar, bien sea de manera teórica o bien de manera computacional. La representación que hagamos de la estructura formal de un texto siempre será una representación parcial de la estructura inmanente del texto. Toda representación de la estructura formal estará sesgada y determinada por el modelo teórico asumido (tipos de unidades, tipos de relaciones, etc.), el modelo formal, así como por la subjetividad del intérprete o, en el caso de la representación computacional, el propio modelo de representación digital de la información (sistema binario, etc.).

El primer elemento que se debe determinar para representar al estructura formal de un texto es la unidad básica de representación. La lingüística, por tradición, asumen como unidad lingüística principal la palabra (la unidad léxica), si bien es un concepto de difícil definición.

A partir de la palabra, se establecen tres niveles estructurales: la micro-estructura, la meso-estructura y la macro-estructura.

Micro-estructura: las unidades mínimas de tamaño inferior a la palabra. Básicamente son los fonemas en el texto oral o los caracteres en el texto escrito (en tanto que unidad mínima, que no se puede dividir, pero sin significado en sí mismo) y los morfemas (combinación mínima de una o más letras con capacidad significativa, es decir, que pueden modificar el significado de la palabra). En términos lingüísticos, este nivel estructural corresponde a la morfología.

³<https://huggingface.co/docs/transformers/tasks/asr>

Por ejemplo, la palabra CANTARÍA está formada por ocho caracteres en tanto que unidades mínimas (C A N T A R Í A) y por cuatro morfemas:

1. CANT- es la raíz léxica y la que aporta el sentido general de “cantar”;
2. -A- es la vocal temática que simplemente indica que el verbo pertenece a la primera conjugación (es decir, se deriva según las regularidad de este tipo de verbos);
3. -RÍ- es la desinencia que aporta gran parte de la información gramatical: tiempo condicional (es decir, que la acción no se ha producido y solo se producirá bajo unas condiciones), modo indicativo (que es información real y verificable), aspecto imperfectivo (que la acción no se ha finalizado, si bien en este caso tampoco se ha iniciado al ser tiempo condicional); y
4. -A que indica la persona que realizaría la acción: en este caso esta información es ambigua pues podría ser tanto la persona que emite la palabra (el “yo” de la primera persona) como un tercera persona sin especificar (un “él”).

Excepto la vocal temática, el resto de morfemas aporta algún tipo de información semántica, por mínima que sea. La lingüística teórica ha propuesto otras agrupaciones intra-léxicas, como las sílabas. En todo caso, estas unidades son agrupaciones de caracteres a partir de algún criterio.

En la meso-estructura la unidad principal es la palabra. En este nivel se considera tanto las palabras de manera aislada como su combinación en oraciones. En términos lingüísticos, este nivel estructural corresponde a la sintaxis.

“Palabra” es un concepto difícil de definir. Por un lado, las palabras en un texto pueden aparecer unidades (como en DÁSELO, que en realidad son tres palabras: “dar” (verbo), “se” y “lo” (ambos pronombres)); o una misma palabra puede aparecer separada (como las formas compuestas de los verbos: “he cantado” es solo una palabra, pretérito perfecto del verbo “cantar”).

Desde un punto de vista computacional, más que con palabras se trabaja con la unidad “token”. Un *token* es una secuencia de caracteres separadas por blanco. Este criterio luego se matiza juntando algunos *tokens* (como los dos *tokens* que forman un mismo verbo tipo “he cantado”) o separando otros (como los signos de puntuación, que se separan de la palabra anterior, o amalgamas como los clíticos en español, que es el caso de “dáselo”).

Todos los *tokens* de un texto que sean iguales se considera que pertenecen la mismo *type* o tipo. Si en un texto aparece, por ejemplo, cinco veces la secuencia casa, se dice que son cinco *tokens* del *type* CASA. Esta es la base de las

frecuencias léxicas, así como de los *word embeddings* de los que dependen las redes neuronales, como se expondrá más tarde.

Por otro lado, todos los *types* que comparten significado léxico se consideran variaciones morfológicas del mismo lema. El lema es la forma de nombrar una palabra y todas sus derivaciones morfológicas. En el caso de los verbos, el lema suele ser la forma de infinitivo; y en el caso de los nombres la forma de masculino singular. Así, por ejemplo, si en un texto aparecen *types* como “cantaría, cantábamos, canteré”, etc.; se dice que todos sus *tokens* pertenecen al lema “cantar”, es decir, todos transmiten el significado básico de “cantar” (con las variaciones gramaticales correspondientes). Esta es la base del análisis semántico léxico que veremos más tarde.

Además de por su lema, los *tokens* de un texto se pueden agrupar según el tipo de palabra a la que pertenecen: su categoría gramatical o *part of speech*. Desde Aristóteles se han propuesto diferentes categorías para clasificar las palabras según su forma y comportamiento verbal. Las clases más comunes son nombre, verbo, adjetivo, adverbio, pronombre, determinante, preposición, conjunción y exclamación.

Determinar la clase de cada palabra es el primer paso para determinar cómo se agrupan entre sí: el análisis sintáctico. Unas palabras se agrupan con otras según su categoría gramatical y sus rasgos morfológicos. Por ejemplo, un determinante y un nombre se agrupan formando un sintagma nominal, que puede incluir también un adjetivo. Esto es posible solo si concuerdan en género y en número: “la casa verde” es un sintagma nominal, pero no lo es “la casas verde” porque el nombre está en plural, mientras que el determinante y el adjetivo están en singular. Las palabras se agrupan hasta alcanzar el nivel superior: la oración.

Los sistemas de PLN actuales, más que agrupaciones en sintagmas lo que detectan es el tipo de relación sintáctica entre dos palabras. Es el llamado análisis de dependencias que se expondrá más tarde.

Finalmente, la macro-estructura. En este nivel la unidad mayor es el propio texto. La macro-estructura se refiere a todas las unidades superiores a la oración (pero inferiores a la unidad texto) y a las relaciones entre ellas. Estas unidades se llaman, de manera general, frases textuales y pueden estar formados por una o más oraciones. Es difícil delimitarlas desde un punto de vista general: así como en un idioma se puede determinar qué es una palabra o qué es una oración, no es posible determinar desde el propio sistema lingüístico qué es una frase textual. Esta será cualquier unidad supra-oracional en un texto concreto que tenga alguna marca formal que la diferencie de otras frases textuales. Puede ser, por ejemplo, el párrafo en un texto escrito; una secuencia de oraciones separadas por pausa o por marcadores discursivos en

un texto oral, o los turnos de palabra en una conversación (oral o digital).⁴

Toda esta organización arquitectónica en tres niveles (micro-, meso- y macro) radica en la palabra como unidad estructural principal. Más o menos todas las teorías lingüísticas se ajustan a este planteamiento.

Los sistemas de PLN, por influencia de la Lingüística teórica, también asumen esta estructura con la palabra como unidad principal (el *token*, para ser precisos). Este modelo resulta útil porque es muy claro para el ser humano. Sin embargo no tiene por qué ser el más apropiado para un sistema computacional. Los sistemas de PLN se deben ajustar a la unidad lingüística más apropiada desde el punto de vista computacional. Hoy día, hay sistemas neuronales que utilizan el carácter como unidad principal. Esto tiene la ventaja de que es una unidad lingüística que se puede delimitar sin ningún tipo de ambigüedad, los sistemas resultantes, sin embargo, no son sistemas transparentes para el ser humano.

Sea cual sea el modelo considerado y las unidades definidas, se suele asumir que la estructura formal primaria de un texto será una estructura jerárquica, donde una unidad de orden superior está formada por unidades de orden inferior y sus relaciones: una palabra está formada por los morfemas y sus relaciones, una oración está formada por las palabras que la componen y sus relaciones, etc. Esta organización jerárquica se asume por el **principio de composicionalidad**,⁵ que será tratado en el siguiente punto. Junto a esa estructura jerárquica (o como alternativa) se pueden establecer otras relaciones, sobre todo relaciones lineales: las que se establecen en el texto por el propio orden de aparición de las palabras.

Estructura semántica (significado)

La estructura semántica de un texto es su significado. Este es el tema más controvertido tanto en Lingüística como en PLN, pues no hay una definición clara de qué es el significado. Al significado hay diferentes aproximaciones.

Al igual que con la estructura formal, debe quedar claro desde el principio que una cosa es la semántica del texto (esa representación cognitiva que hace la

⁴No se consideran aquí como unidades textuales el conjunto de oraciones relacionados con una misma idea o con un mismo tema porque estas unidades responden a un criterio semántico, no formal, y por tanto entrarían dentro del ámbito del significado que se verá después.

⁵El principio de composicionalidad establece que el significado de una unidad compleja se crea en función del significado de sus unidades inferiores y de las relaciones entre ellas. Ver <https://plato.stanford.edu/entries/compositionality/>

mente humana al interpretar el texto) y otra cosa distinta es la representación que como humanos podemos hacer de ese significado. Esa representación siempre será parcial y dependerá, entre otros aspectos, de los modelos teóricos y formales aplicados.

Para establecer el significado de un texto, los modelos teóricos y muchos modelos computacionales parten, como se comentó antes, del principio de composicionalidad. Este principio establece que el significado de una unidad compleja (el texto) está en función del significado de sus partes y de las relaciones entre ellas. Esta es la razón por la que, en los modelos explicativos lingüísticos, se plantea primero una estructura formal y luego una estructura semántica. La primera establece cuáles son esas unidades y cómo se relacionan entre sí, y la segunda cómo, a partir de cada unidad, se va generando el significado global.

En términos generales, por tanto, el significado de una texto vendría dado por las relaciones entre los significados de cada oración; y el significado de cada oración por el significado de las palabras que la forma y sus relaciones. Hay algunos aspectos que se deben tener en cuenta.

En primer lugar, una cosa es el significado que una palabra en general y otra el significado de la palabra en un texto concreto. Una palabra puede tener multitud de significados (como se puede ver en cualquier diccionario), pero en un texto concreto, en un *contexto* concreto, en principio asumirá solo uno de esos posibles significados. Así, el significado de una palabra está determinado tanto por sus posibles significados como por el significado de las palabras con las que aparece en el texto (las palabras del contexto). Este aspecto semántico se ha tratado en Procesamiento del Lenguaje Natural con el problema denominado *word sense disambiguation*, cuyo objetivo es determinar el significado de una palabra en un contexto concreto a partir de la lista de posibles significados.

En segundo lugar, junto a este significado de tipo denotativo (relación de una palabra con un concepto) hay también un significado connotativo: todas aquellos aspectos subjetivos relacionados con las palabras y su uso. Esa relación de las palabras y las emociones subjetivos se ha tratado en PLN con la tarea de análisis de sentimientos y opiniones.

En tercer lugar, este significado que más o menos está sistematizado en el idioma (es decir, que aparece en diccionarios, etc.) no es el único significado que una palabra o expresión lingüística puede tener. Una parte importante de la interpretación semántica de un texto depende de significado inferido, es decir, de aspectos semánticos que no están expresados de manera explícita en el texto pero que una mente humana puede inferir a partir de su conocimiento del mundo o de relaciones lógicas. Las implicaciones textuales, por ejemplo, se refiere a toda esa información que no está explícitamente expresada en el

texto pero que están implicada en él. Por ejemplo, que una persona ronque implica que está dormida. Por eso esta oración tiene sentido:

“Ronca tanto que no descansa bien por las noches y luego no puede conducir”

Esa oración habla de una persona (solo las personas conducen) que duerme mal (solo se ronca cuando se duerme). Sin embargo, en el texto no aparece ni el concepto de persona ni la palabra “dormir”. Ambos están implicados en el texto y lo activamos a partir de nuestro conocimiento de mundo. Desde un punto de vista computacional, toda esa información debe ser recuperada de alguna manera.

El significado de un texto está determinado también por el propio orden y disposición de las palabras y unidades en el texto. No es solo, por tanto, las relaciones formales (sintácticas) entre las palabras las que determinan el significado del texto, sino también el orden en que aparecen, es decir, el contexto concreto. Los *word embeddings* que se verán en el siguiente tema, base de los actuales modelos neuronales, dependen mucho del orden de aparición de las palabras.

En definitiva, interpretar un texto es crear la imagen mental de los seres, estados, procesos, acciones e ideas expresadas en ese texto (la imagen mental del referente), una imagen mental que debe estar cohesionada y debe ser coherente. La coherencia es un factor clave en la interpretación del texto, pues incluso ante información contradictoria, el ser humano siempre intenta dar coherencia a la información transmitida en el texto.

Además de todos estos aspectos textuales, en la interpretación del texto juegan un papel muy importante aspectos pragmáticos: intención por la que se genera o se interpreta el texto (el “para qué”), hipótesis interpretativas (qué tipo de texto esperamos según quién hable, por ejemplo), preferencias interpretativas (qué texto queremos o nos gustaría generar, oír, leer, interpretar según la situación) o incluso el estado físico y mental. Todo ello afecta a los procesos interpretativos y productivos.

Hay toda una teoría lingüística para analizar estos aspectos: la teoría de los actos de habla [2, 28]. Esta teoría parte de la idea de que comunicarse en sí es un acto (el acto de hablar). Este acto tiene tres niveles:

1. acto locutivo: el propio hecho de hablar, de comunicarse;
2. acto ilocutivo: la intención con la que se dice algo (felicitar, agradecer, ordenar, prometer, aconsejar, pedir, suponer, etc.);
3. acto perlocutivo: la consecuencia de lo que se dice. Esta consecuencia es del mundo real, externa al texto.

Imagina una situación comunicativa de un grupo de personas en una habitación, y una persona A, dirigiéndose a otra persona B, dice: -¿Puedes cerrar la ventana? Acto seguido, la persona B se levanta y cierra la ventana. En esta situación comunicativa, el acto locutivo es la propia acción de decir de la persona A, el acto ilocutivo es dar una orden (en forma de pregunta por cuestiones educativas y culturales, pero una orden a fin de cuentas), y el acto perlocutivo el hecho de que la persona B cierre la ventana.

Hay veces en que el acto locutivo y el ilocutivo no coinciden. Estos casos se denominan actos de habla indirectos. En la misma situación anterior, la persona A dice -Hace frío aquí, ¿no? Ante lo que la persona B se levanta y cierra la ventana. En este caso, el acto ilocutivo es una simple descripción, pero de manera indirecta es una orden o una solicitud.

Estos aspectos pragmáticos deben ser también modelados computacionalmente para tener agentes de IA inteligentes. Si bien hay muchos aspectos de la comunicación lingüística humana sin una modelización computacional válida, sí se han realizado grandes avances en los últimos años, sobre todo con de las redes neuronales.

A lo largo de la asignatura se irán exponiendo los principales avances en PLN para modelar la comunicación lingüística humana. Antes se van a exponer cuestiones generales como los modelos de representación formal o las principales técnicas de procesamiento.

2.3. Procesamiento computacional: representación formal y método de análisis

En la sección anterior se han mostrado algunos de los problemas que se estudian desde la Lingüística para dar cuenta de la interpretación o creación de un texto. Estos modelos son modelos explicativos, dado que intentan explicar cómo es eso de que a partir de unos sonidos o caracteres, una mente humana pueda llegar a un significado conceptual e incluso a realizar alguna acción.

Los sistemas de PLN toman esos modelos teóricos como marco general, pero no son una emulación computacional de éstos. Los modelos computacionales de procesamiento lingüístico deben estar adaptados a su medio computacional: deben ser modelos eficaces adaptados a los modos propios de procesamiento de información de un ordenador. En muchos casos, como luego veremos, los modelos más eficaces son modelos opacos, es decir, no se puede explicar cómo hace ese procesamiento. En la medida que el proceso sea correcto, son modelos válidos para el PLN y para la IA.

Desde un punto de vista muy general, todo sistema de PLN se caracteriza por dos aspectos:

1. Cómo representa tanto la información lingüística como la información conceptual. Para que la máquina puede comprender y procesar esa información, debe ser formal y no ambigua. Se requieren, por tanto, modelos formales de representación capaces de capturar una información que es subjetiva por naturaleza.
2. Cómo se procesa la información: el método de análisis propiamente dicho. Cómo unos datos de entrada (caracteres) se transforman al final en una representación conceptual, en otro texto, en una serie de acciones, etc. Y al inversa: cómo a partir de unos datos se genera un texto comprensible para un ser humano.

Modelos de representación formal

Al igual que en otras áreas de la IA, en PLN hay dos paradigmas generales de representación de la información lingüística y conceptual: el paradigma simbólico y el paradigma conexionista.

Paradigma simbólico

Los modelos simbólicos [14] se caracterizan por ser similares (o estar inspirados en) los modelos teóricos y, por ello, son modelos transparentes y explicativos: muy claros para el ser humano. Éste puede ver y comprender qué hace el sistema en cada momento y por qué. En contra, no son los modelos de representación más eficaces, como se verá más tarde.

Efectivamente, los modelos simbólicos codifican el conocimiento lingüístico de manera explícita. Este conocimiento es el que el ser humano ha creado para explicar el lenguaje: información morfológica, sintáctica, semántica, etc. Al igual que en Lingüística teórica, cada unidad lingüística o cada relación entre las unidades se codifica mediante símbolos. Así, por ejemplo, dada esta oración:

A Madrid iré este fin de semana.

Un modelos simbólico indicaría por ejemplo que “Madrid”, “fin” y “semana” son nombres. Para ello especifica una etiqueta (por ejemplo, N), es es un símbolo en sí mismo para representar la categoría “nombre”. Y lo mismo con el resto de categorías gramaticales. Las relaciones sintácticas también se

ID	token	categoría
1	a	PREP
2	madrid	N
3	iré	V
4	este	DET
5	fin	N
6	de	PREP
7	semana	N

Cuadro 2.1: Representación simbólica de información categorial.

representan mediante etiquetas, que igualmente son símbolos que representan esa relación. Por ejemplo, se puede especificar la etiqueta (el símbolo) *su* para representar la relación de sujeto entre un verbo y un nombre, etc. También se pueden especificar etiquetas para marcar el significado: por ejemplo, marcar “Madrid” con su código de Wikidata (<https://www.wikidata.org/wiki/Q2807>). Este código sería el símbolo que representa el significado del *token* “Madrid” en ese texto concreto.

Los modelos de representación simbólica hacen uso, por tanto, de etiquetas que, a modo de símbolos, representan la información lingüística o conceptual del texto de manera explícita y transparente. Estas etiquetas actúan como identificadores únicos, por lo que no tienen ambigüedad. Su relacionan con el texto de entrada se expresa mediante lenguajes formales como, por ejemplo, lenguajes de marcado (XML) o formatos tabulares.

Así, un modelo simbólico podría representar la información categorial de la oración anterior con la tabla del Cuadro 2.1:

En la base de estos modelos simbólicos está la hipótesis de que la mente humano es simbólica, que trabaja con símbolos (como es el propio lenguaje). Así, una máquina capaz de trabajar (modelar, interpretar, transformar...) símbolos es una máquina inteligente.

En el próximo capítulo se expondrán los principales modelos de representación simbólica del PLN en la actualidad. Hay que indicar, con todo, que si bien el modelo de representación simbólica ha sido el modelo dominante en PLN, en los últimos años, con el desarrollo de las redes neuronales, se está pasando a modelos de representación tipo conexionista.

Paradigma conexionista

El paradigma conexionista [29], también llamado sub-simbólico, es el modelo de representación propio de las redes neuronales artificiales (RNA) y hoy, gracias a los amplios modelos de lenguaje, es el paradigma de represen-

tación predominante. Este paradigma solo tiene sentido y se puede comprender dentro de una RNA. Como recordarás una RNA, aparte de una entrada y una salida, está formada por unidades de procesamiento conectadas entre sí llamadas neuronas.

En el modelo conexionista, las primitivas semánticas son “sub-símbolos”. Se consideran símbolos porque también representan información, pero es una representación “de grano fino”: es decir, no hay una relación directa entre un símbolo y un concepto (como ocurre con el modelo simbólico), sino que la información se representa mediante las conexiones entre neuronas. Cada sub-símbolo tiene un contenido semántico, denota un aspecto del mundo, pero la naturaleza de ese contenido semántico no es conceptual (no hay relación directa con un concepto, tal y como lo entendemos los humanos), sino que ese contenido semántico es sub-conceptual: está por debajo del nivel conceptual consciente. A ojos humanos, los conceptos son neuronas enviando y procesando información, pero no se puede relacionar directamente (o no siempre) la activación de unas neuronas con conceptos concretos (lingüísticos o generales).

Este modelo, con las máquinas actuales, se ha mostrado muy eficaz (eficiente no tanto) y ha resuelto muchos de los problemas del Procesamiento del Lenguaje Natural que hace apenas diez años, con modelos de representación simbólica, parecían irresolubles. En contra, son modelos opacos para el ser humano: podemos saber que el sistema de PLN realiza tareas de procesamiento lingüístico, pero no podemos saber exactamente cómo las realiza ni cómo representa la información, ya que maneja la información a un nivel sub-conceptual. En PLN, esa información sub-conceptual está representada mediante vectores, como se verá después. Desde el punto de vista explicativo (aplicación de PLN para conocer mejor cómo funciona la mente humana) no son modelos apropiados, pero desde el punto de vista computacional sí lo son.

Ambos paradigmas están determinados por los métodos de procesamiento (análisis y generación), que se presentan a continuación.

Métodos de procesamiento

Estos dos paradigmas de representación corresponden a los dos métodos de procesamiento lingüístico principales: el racional (simbólico) y el neuronal (conexionista). A estos dos modelos se les une un tercero, que ha sido el modelo de procesamiento preponderante en los últimos veinte años: el modelo de procesamiento simbólico empírico.

El modelo simbólico racional se basa en la creación explícita de reglas de procesamiento lingüístico. Es el modelo más cercano a la Lingüística teórica,

Madrid	→	NOMBRE_PROPIO	→	Q2807
--------	---	---------------	---	-------

Cuadro 2.2: Análisis como transformación de símbolos

influido sobre todo por la teoría generativo-transformacional de N. Chomsky [6, 7]. Esta teoría formaliza los procesos de interpretación lingüística mediante reglas. Esta misma idea es la que se adaptó al PLN. Este tipo de modelos racionalistas se desarrollaron sobre todo en los años 80 del siglo XX.

Estas reglas son reglas de manipulación simbólica. Básicamente una regla transforma un símbolo en otro: un *token* en un texto (que como palabra es ya en sí un símbolo), por ejemplo, una regla lo transforma en su categoría gramatical (que también es un símbolo). A su vez, otra regla, a partir de la etiqueta categorial, puede transformar el *token* de entrada en un número que identifique su significado en una ontología, por ejemplo. Este identificador es de nuevo es un símbolo que se ha transformado en otro: una palabra en su significado. Ver Cuadro 2.2. Aplicado a todo el texto, se ha realizado la interpretación.

Un conjunto de reglas forman una gramática, y una gramática completa (en teoría) podría procesar cualquier texto.

La principal ventaja de estos métodos es que son claro y transparentes. El sistema realiza el análisis tal y como haya sido diseñado por el humano, que es quien crea las reglas. Si bien desde un punto de vista lingüístico esta capacidad explicativa es una ventaja, desde un punto de vista computacional con sistemas basados en reglas solo se pueden procesar un conjunto limitado de oraciones (“oraciones de laboratorio” llamadas). Son sistemas apropiados para tratar fenómenos lingüísticos muy concretos, pero estos sistemas de reglas puros no se pueden escalar para procesar grandes cantidades de texto.

La solución a este problema de escalabilidad vino con los métodos empíricos. Estos métodos también utilizan símbolos y reglas, pero esas reglas:

1. son deducidas directamente de los textos mediante técnicas de aprendizaje automático, y
2. incorporan pesos estadísticos de tal manera que se pueden aplicar o no según el contexto.

Como ya sabrás, hay dos tipos de métodos en aprendizaje automático: los métodos supervisados y los no supervisados, según partan o no de información previa.

En PLN, los métodos supervisados parten de corpus que han sido ya previamente analizados por expertos. Estos han marcado el corpus con un lenguaje formal (XML normalmente) y han incluido la etiqueta (el símbolo) a

cada fenómeno lingüístico que se quiere modelar. En el caso, por ejemplo, de crear un sistema de PLN que analice las categorías gramaticales de un texto, se parte de un conjunto de texto cuyas palabras tienen la etiqueta categorial correspondiente, que ha sido marcada por expertos. A partir de ahí, los algoritmos de aprendizaje aprenden cada palabra cuál es su categoría según el contexto donde aparece, y (si está bien hecho) harán el proceso de abstracción de tal manera que ante una palabra desconocida, pueda decidir por su contexto cuál es su categoría gramatical de manera correcta. Luego se verán ejemplos de esto.

Los procesos no supervisados realizan procesos de inferencia sobre los corpus, de tal manera que extraen la información sin necesidad de que haya sido previamente marcada por expertos. Normalmente hace agrupaciones de elementos lingüísticos según contextos similares. En el caso de las categorías gramaticales, a partir de las palabras no ambiguas (aquellas que solo pueden tener una categoría gramatical), el sistema podría aprender a categorizar palabras ambiguas (las que pueden tener dos o más categorías) según el contexto. Este es un caso de aprendizaje no supervisado, entre otros muchos.

Estas técnicas ya sí son capaces de procesar amplios corpus. Con ello, puede extraer regularidades y sus probabilidades. El análisis ya no es un análisis simbólico puro (transformación de símbolos) sino que a esa manipulación de símbolos se le une la información estadística que puede inferir del corpus y con ello la aplicación de probabilidades en el análisis. Este análisis basado en computación numérica acerca los sistemas simbólicos a los sistemas neuronales, pero en este caso se aplican técnicas de combinatoria y probabilidad.

Los métodos neuronales, finalmente, se basan en las RNA. En la entrada, el texto se transforma en vectores que capturan su información lingüística (cómo se realiza esa transformación se verá luego). Los nodos de la red operan con esos vectores hasta alcanzar la salida. En próximos temas se expondrá todo esto con detalle. Lo relevante aquí es que este procesamiento ya no es simbólico, pues no podemos relacionar el procesamiento de estos vectores a conceptos concretos: la representación es en este caso sub-simbólica. Sin embargo, hoy día son los métodos que mejor funcionan para prácticamente todas las tareas del PLN.

2.4. Conclusiones

En este capítulo se han expuesto los conceptos básicos del PLN. Más que soluciones, se han planteado los problemas generales que tiene procesar automáticamente un texto, así como los paradigmas de representación y procesamiento principales.

De estos métodos, la mayor parte de la asignatura está centrada en los modelos neuronales, que son hoy día la principal técnica de PLN. Antes de ello, se van a explicar los modelos de representación simbólica por dos razones: primero porque son modelos transparentes que permiten mostrar bien los problemas del procesamiento lingüístico (los neuronales son opacos), y segundo porque han sido los modelos preponderantes del inicio del PLN hasta la actualidad. Para calibrar bien el alcance de los modelos neuronales que se explicarán después es necesario conocer los modelos simbólicos previos.

Una reflexión final: ningún modelo es exactamente igual a la naturaleza humana. Los modelos inspirados en la lingüística teórica son claros, transparentes, explicativos, pero siempre parciales. Los modelos neuronales son eficaces, pero opacos: no es fácil extraer conocimiento sobre cómo el ser humano procesa un texto. Qué modelos habrá mañana no se sabe, pero los modelos del futuro son los que se están ideando ahora. No veáis esta asignatura como un conocimiento cerrado, sino como un área viva que está esperando vuestra aportación: nuevos modelos de representación, de procesamiento, nuevas soluciones a problemas conocidos, nuevos problemas por resolver, nuevas ideas... Sólo conociendo bien el pasado se puede crear un nuevo futuro.

2.5. Lecturas opcionales

Para completar este tema se pueden consultar cualquiera de los principales manual de Procesamiento del Lenguaje Natural. Entre otros:

1. [17]
2. [5]
3. [11]
4. ...

Análisis categorial basado en métodos simbólicos.

3.1. Unidades de comunicación básica. La palabra. *Type*, *token* y lema.

Si bien se suele utilizar como unidad mínima y básica de comunicación, la palabra es un concepto vago que no tiene una definición clara en Lingüística.

En lingüística de corpus se trabaja con dos conceptos relacionados: *type* y *token*.

- *Type* es la palabra entendida como clase. Una secuencia de caracteres que se diferencia de cualquier otra secuencia.¹
- *Token* es cada una de las instancias concretas de esas clase que se pueden hallar en un texto.

Por ejemplo, en esta oración:

Una rosa es una rosa es

encontramos tres *types*:

- una
- rosa
- es

¹Token se asimila en este caso a *occurrence*. Ver <https://plato.stanford.edu/entries/types-tokens/>

pero seis *tokens*. El cálculo de frecuencias más simple que se puede hacer es contar la cantidad de *tokens* de cada *type*:

- una: 2
- rosa: 2
- es: 2

El tamaño del corpus se suele indicar en número de *tokens*.

La tokenización más simple es separar las palabras por espacios en blanco. Pero hay algunos problemas que deben ser tenidos en cuenta como:

- signos de puntuación,
- unidades multipalabra (como formas complejas del verbo Ej. "he comido") o
- contracciones ("del", ".l") y en general formas aglutinantes ("dáselo").

Type y *token* se refieren siempre a formas flexionadas, es decir, a formas con variaciones morfológicas. Así, *çatamosz çantaré* son *tokens* distintos; al igual que *çasaz çasas*".

3.2. Lematización y *stemming*

Para agrupar todos los *tokens* relacionados con la misma palabra (es decir, la forma sin flexionar o la unidad léxica que podemos encontrar, por ejemplo, en los diccionarios) se realiza un proceso de lematización. La lematización es asignar a cada palabra su forma no marcada: infinitivo para verbos, forma masculino singular para nombres y verbos (es decir, la forma que aparece en el diccionario). El lema es una manera de nombrar la palabra en toda su diversidad flexiva.

La lematización es un fenómeno complejo porque es necesario analizar morfológicamente la palabra para determinar su lema. Ej. El lema del token "traje" puede ser "traer" (si es verbo) o "traje" (si es nombre).

Un proceso similar pero más sencillo es el *stemming*: reducir cada *token* a su raíz o lexema: la parte invariable que, en principio, asume el significado general de la palabra.

1	¿	¿	Fia
2	Usted	usted	PP2CS0P
3	no	no	RN
4	nada	nadar	VMIP3S0
5	nada	nada	PI0CS00
6	?	?	Fit
7	-	-	Fg
8	Es	ser	VSIP3S0
9	que	que	CS
10	no	no	RN
11	traje	traer	VMIS1S0
12	traje	traje	NCMS000
13	.	.	Fp

Cuadro 3.1: Análisis categorial a tres columnas: *token*, lema y etiqueta.

3.3. Análisis morfológico y categorial.

Esta es una tarea tradicional en PLN. Hoy día cualquier sistema básico de PLN incluye un analizador categorial como primera tipo de análisis, una vez separado el texto en palabras. Este análisis es, a su vez, la base del análisis sintáctico que se verá después. Se le suele llamar por su nombre en inglés: *Part of Speech tagger* o *PoS_tagger*.

El objetivo de este análisis es asignar a cada palabra de un texto (*token*) su categoría gramatical correspondiente. En concreto, por cada *token* del texto (incluidos signos de puntuación, etc.) se determina su lema o forma no marcada, su categoría gramatical y rasgos morfológicos.

Por ejemplo, la tabla 3.1 muestra el análisis categorial del siguiente texto:

"¿Usted no nada nada? -Es que no traje traje"

Como vemos este es un típico caso de representación simbólica de la información lingüística. Ésta aparece expresada de manera explícita y formal mediante etiquetas que aluden al concepto lingüístico.

A lo largo de la historia del PLN se han propuesto juegos diversos de etiquetas para representar la información categorial. Las principales (o al menos las más utilizadas) son las siguientes:

- **Penn Treebank tag set.** Se crearon para la anotación del corpus Penn Treebank, por lo que están pensadas solo para textos en inglés. Hoy día están en prácticamente cualquier herramienta de análisis categorial. Se

caracteriza porque utiliza entre dos o tres letras para representar la información: la primera indica la categoría gramatical general, y la segunda y tercera subcategorías o información morfológica. Las principales etiquetas son

- JJ: Adjective
- JJR: Adjective, comparative
- JJS: Adjective, superlative
- NN: Noun, singular
- NNP: Proper Noun, singular
- NNPS: Proper Noun, plural
- : NNS: Noun, plural
- VB: Verb, base form
- VBD: Verb, past tense
- etc.

La lista completa de etiquetas (no son muchas) se puede ver aquí: <https://www.cs.upc.edu/~nlp/SVMTool/PennTreebank.html>

- **EAGLES tag set:** a diferencia de las etiquetas PennTreebank, las etiquetas EAGLES surgen de un proyecto europeo para crear un juego de etiquetas que valiera para cualquier idioma. En este caso, las etiquetas son de tamaño variable. La primera posición indica la categoría gramatical con una letra. El resto de posiciones es la información morfológica. Si en alguna posición la información no es relevante, se marca con 0.

El nombre, por ejemplo, tiene seis posiciones que corresponden a: la categoría gramatical (nombre), el tipo (común o propio), el género (femenino, masculino o común), el número (singular, plural o invariable), la clase de entidad (persona, localización, organización y otros), la subclase de entidad y el grado. Así, se forman las siguientes etiquetas

- coche NCMS00: nombre (N) común (C) masculino (M) singular (S).
- coches NCMP00: nombre (N) común (C) masculino (M) plural (P).
- casa: NCFS00 nombre (N) común (C) femenino (F) singular (S).

Y así el resto de categorías. Este formalismo es el utilizado por el sistema de PLN [Freeling](#). Toda la información se encuentra aquí:

- Explicación general de las etiquetas: <https://freeling-user-manual.readthedocs.io/en/latest/tagsets/#freeling-tagset-description>
 - Explicación de las etiquetas para español: <https://freeling-user-manual.readthedocs.io/en/latest/tagsets/tagset-es/>
- **Universal tagset:** este juego de etiquetas forma parte del *Universal dependencies project*, un proyecto que busca un modelo de representación simbólica unificado de información categorial y sintáctica para cualquier idioma. Dado este carácter universal, se busca representar solo aquellas categorías comunes a prácticamente cualquier idioma. Esto se ha concretado en 17 etiquetas. Cada una está formada por tres letras, pero no incluyen información morfológica, solo la categoría gramatical. La información morfológica se codifica por otro lado.
- ADJ: adjetivo
 - ADV: adverbio
 - NOUN: nombre
 - DET: determinante
 - PROPN: pronombre personal
 - VERB: verbo
 - etc.

La lista completa de etiquetas se puede consultar aquí: <https://universaldependencies.org/u/pos/>.

Siguiendo la misma filosofía, para representar la información morfológica se ha creado una lista de rasgos morfológicos universales (presentes en la mayoría de los idiomas), como género gramatical, número, reflexivo, tiempo, voz, aspecto, etc. Cada uno se codifica con una etiqueta específica, que se puede consultar aquí: <https://universaldependencies.org/u/feat/index.html>. Cada idioma, según sus características morfológicas propias, utilizará unas etiquetas y otras para marcar explícitamente la morfología de las palabras de los textos procesados.

Al ser un modelo universalista, la información lingüística que queda explícitamente marcada es menor que con otros modelos, que son más específicos. La razón es que se busca marcar solo aquella información categorial y morfológica que sea común a la mayoría de los idiomas. La ventaja de este modelo es que permite realizar análisis multilingües. Los sistemas de PLN multilingües (es decir, capaces de analizar diferentes idiomas) suelen utilizar este modelos de codificación.

3.4. Ambigüedad categorial y proceso de análisis

El análisis computacional de categorías gramaticales no es en principio una tarea compleja. El problema principal es la ambigüedad categorial: aquellas palabras que pueden tener dos o más categorías gramaticales. Este es el caso de las palabras “nada” y “traje” del ejemplo anterior (Cuadro 3.1). La buena noticia es que las categorías gramaticales tienen una fuerte dependencia del contexto previo: sabiendo la categoría gramatical de una palabra, se puede deducir con cierta seguridad cuál será la categoría de la palabra siguiente. Por ejemplo, ante una palabra que puede ser nombre o verbo (ambigüedad ésta muy común en español, como ocurre con la palabra “traje”), si la palabra anterior es un artículo, la palabra ambigua será nombre con una precisión del 100 %, dado que en español la secuencia *artículo + verbo* no existe. Tras un artículo aparecerá un nombre, un adjetivo, una preposición u otro determinante, pero nunca un verbo.

Dada esta propiedad, un clasificador secuencia clásicos, como los Modelos Ocultos de Markov (HMM) o los Conditional Random Fields (CRF) realizan esta desambiguación de manera correcta. Así también los modelos neuronales.² Con los grandes modelos de lenguaje se puede hacer hoy día análisis categorial con técnicas sencillas, sin necesidad de afinar el modelo para la tarea análisis categorial.³ Los etiquetadores secuenciales neuronales (*Neural Sequence Labeling*) son la aproximación evidente, pero una simple consulta a ChatGPT 3.5 a través del *prompt* ya da buenos resultados.

No se va a profundizar aquí en estas técnicas. Para una explicación más detallada, véase el capítulo 8 “Sequence Labeling for Parts of Speech and Named Entities” de Jurafsky y Martin (2023) *Speech and Language Processing* (disponible aquí: <https://web.stanford.edu/~jurafsky/slp3/8.pdf>). Para un estado de la cuestión actualizado, véase Alebachew Chiche y Betselot Yitagesu (2022) “Part of speech tagging: a systematic review of deep learning and machine learning approaches” en *Journal of Big Data*, 9 (<https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00561-y>).

3.5. Recursos.

Cualquier sistema de PLN parte de un PoS tagger. Es el análisis básico.

- Freeling <http://nlp.lsi.upc.edu/freeling/index.php/>

²Por ejemplo <https://huggingface.co/PlanTL-GOB-ES/roberta-large-bne-capitel-pos>

³Por ejemplo, <https://huggingface.co/flair/pos-english>

- SpaCy: <https://spacy.io/>
- NLTK: <http://www.nltk.org/>
- Stanford CORE NLP: <https://stanfordnlp.github.io/CoreNLP/>
- Google CLOUD: <https://cloud.google.com/natural-language/>

3.6. Lecturas opcionales

Para profundizar en este tema, véase el capítulo 8 “Sequence Labeling for Parts of Speech and Named Entities” de [17]: <https://web.stanford.edu/~jurafsky/slp3/8.pdf>

Análisis sintáctico basado en métodos simbólicos.

El objeto de la sintaxis es detectar relaciones formales entre las palabras y agruparlas. Aunque la lingüística moderno no lo vea así, la principal razón de la sintaxis no es más que detectar relaciones entre palabras para poder determinar la semántica de la oración (principio de composicionalidad) y por extensión la del texto. En otras palabras, el objetivo final es siempre la interpretación del texto (la semántica, que se tratará luego).

Sin embargo, en el nivel sintáctico se buscan las relaciones *formales* entre las palabras: aquellas que se puedan establecer sin tener en cuenta (por ahora) la semántica. Esta separación teórico es solo por fines explicativos (tratar por separado rasgos formales y semánticos), pero un sistema de PLN no tiene por qué hacer esta separación. De hecho, los modelos neuronales actuales no la hacen: parten de la semántica y de ahí derivan relaciones sintácticas.

EL módulo de análisis sintáctico de un sistema de PLN se suele denominar simplemente *parser*.

En PLN hay dos modelos de representación simbólica de la información sintáctica: constituyentes y dependencias.

4.1. Análisis de constituyentes

El modelo de constituyentes agrupa las palabras de una oración según la relación sintáctica que tengan. Estas agrupaciones se denominan sintagmas. Según qué palabra actúa de núcleo, hay sintagmas nominales, adjetivos, preposicionales, adverbiales o verbales.

El resultado final de una análisis de constituyentes es un árbol como el mostrado en la figura [4.1](#).

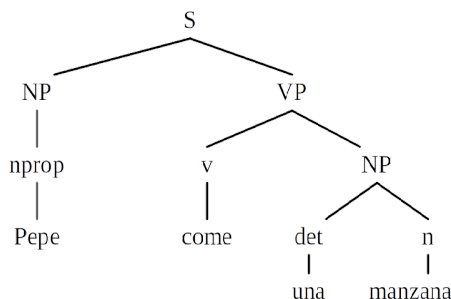


Figura 4.1: Ejemplo análisis sintáctico de constituyentes

$NT = \{S, NP, VP, nprop, n, v, det\},$
 $T = \{Pepe, manzana, come, una\},$
 P:

$S \rightarrow NP VP$
 $NP \rightarrow nprop$
 $NP \rightarrow det n$
 $VP \rightarrow v$
 $VP \rightarrow v NP$

Cuadro 4.1: Gramática independiente de contexto

De nuevo, aquí la representación es simbólica mediante etiquetas. Las etiquetas *NPROP*, *v*, *DET* y *N* son etiquetas categoriales. A partir de su combinación se llega a las etiquetas sintácticas:

- NP: sintagma nominal
- VP: sintagma verbal
- S: oración

Este análisis sintáctico es un ejemplo del clásico análisis simbólico basado en reglas. Para crear ese árbol sintáctico se creaban gramáticas (conjunto de reglas) que básicamente re-escriben un símbolo por otro. Por ejemplo, la oración de la Figura 4.1 podría ser analizada con la gramática del Cuadro 4.1.

Las dos primeras líneas especifican los símbolos a utilizar, terminales (*T*) y no terminales (*NT*). Las siguientes (el contenido de *p*) son las reglas. En cada una, el símbolo \rightarrow indica la transformación, indica un símbolo en qué otro símbolo se puede transformar. Si son dos símbolos que se transformación

en uno, entonces se habla de unificación. Así, si “una” es *det* y “manzana” es *n*, con la regla $NP \rightarrow det\ n$ esos símbolos se transforman en NP, y así sucesivamente.

Esta es una gramática independiente de contexto (*context free grammar*). Este sencillo formalismo fue con los años haciéndose más complejo con la introducción en las reglas de rasgos (de tal manera que, por ejemplo, se produjera unificación solo si los rasgos morfológicos eran compatibles) y con la introducción de pesos estadísticos aprendidos empíricamente en corpus anotados con sintaxis (los llamados *tree banks*), de tal manera que se aplicaran unas reglas u otras según probabilidades contextuales.

En lingüística se han desarrollado diferentes modelos basados en estas técnicas como las *Head-driven phrase structure grammar* o las *Lexical-Functional Grammar* (que sigue siendo un modelo válido. Ver <https://ling.sprachwiss.uni-konstanz.de/pages/home/lfg/>)

El formato de representación en el análisis de constituyentes suele ser el formato parentizado. Así, el árbol anterior se representa formalmente así:

S(SN(nprop(Pepe)) SV(v(come) SN(det(una) n(manzana))))

Para más información sobre el análisis de constituyentes, estrategias de análisis y modelos neuronales, véase el capítulo 17 “Context-Free Grammars and Constituency Parsing” de Jurafsky y Martin (2023) *Speech and Language Processing* (<https://web.stanford.edu/~jurafsky/slp3/17.pdf>)

4.2. Análisis de dependencias

La gramática de dependencias (*dependency grammar*) es en la actualidad el formalismo de representación sintáctica más utilizado, y con ello es el análisis de dependencias (*dependency parsing*) el tipo de análisis sintáctico automático más común.

Gramáticas de dependencias

Las gramáticas de dependencias representan de manera formal las dependencias sintácticas entre las palabras de una oración.

La información sintáctica que representan es complementaria a las gramáticas de constituyentes expuestas en anteriormente. En vez de representar las relaciones sintácticas mediante la agrupación de palabras en constituyentes o sintagmas (con forma de árbol) y su categorización en diferentes tipos (Figuras 4.1 y 4.2), las gramáticas de dependencias especifican directamente

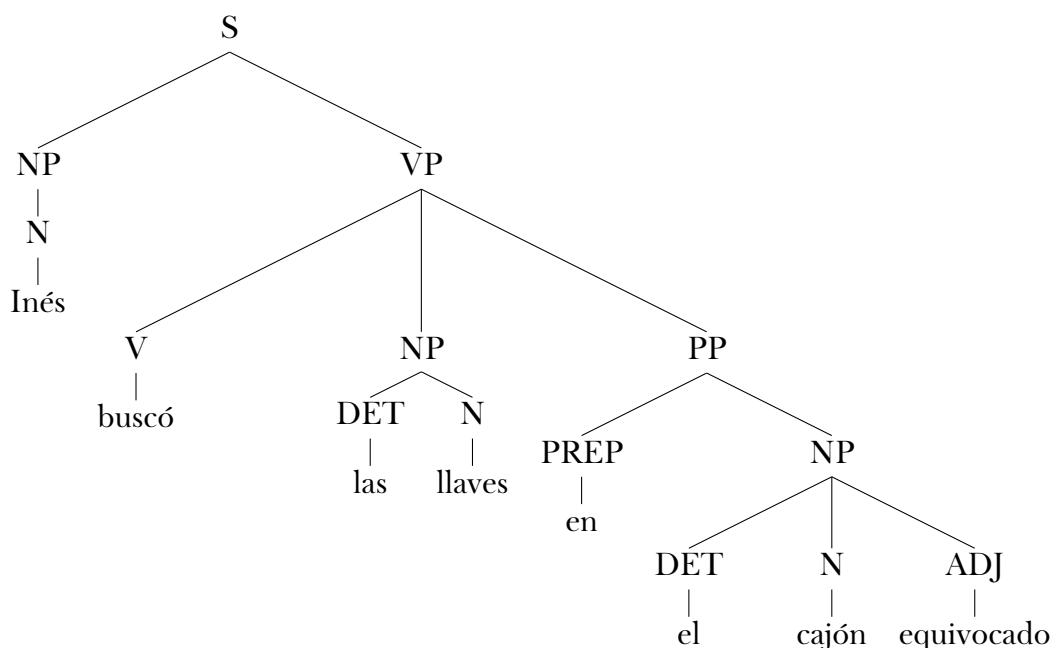


Figura 4.2: Ejemplo de análisis de constituyentes.

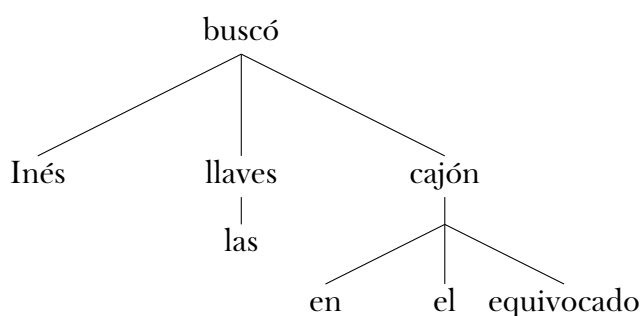


Figura 4.3: Ejemplo sencillo de análisis de dependencias.

qué palabras dependen (sintácticamente) de qué otras palabras y qué tipo de relación o dependencia mantienen entre sí (Figura 4.3).

La gramática de dependencias representa formalmente la relación sintáctica entre dos palabras mediante un arco binario directo. El elemento principal del arco es el **núcleo** (*head*) y el elemento relacionado es el **complemento** o palabra dependiente (*dependent*).

Cada arco está, además, categorizado con el tipo de dependencia sintáctica entre ambas palabras. A esto se le denomina **estructura de dependencia tipificada** (*typed dependency structure*), ya que las categorías de los arcos (los tipos de arcos) están predefinidas. En la gramática tradicional del español a estas

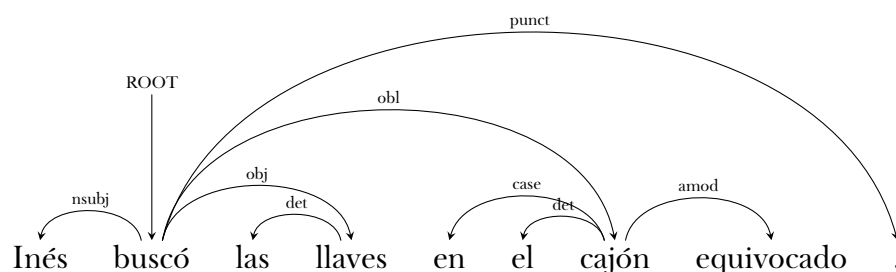


Figura 4.4: Ejemplo completo de análisis de dependencias.

Núcleo	Dependiente	Tipo	Explicación
<i>buscó</i>	<i>Inés</i>	NSUB	Sujeto
<i>buscó</i>	<i>llaves</i>	OBJ	C. directo
<i>buscó</i>	<i>cajón</i>	OBL	C. circunstancial (no argumental)
...

Cuadro 4.2: Relaciones de dependencias

categorías se las denomina **funciones sintácticas**.

Un análisis de dependencias completo se puede ver en la Fig. 4.4. Podemos observar primero que los arcos son dirigidos: van siempre de una palabra núcleo a una palabra objetivo (la palabra dependiente). No son, por tanto, relaciones simétricas.

El árbol se inicia con una palabra principal, categorizada como *root*. En este caso (como en toda oración enunciativa), la palabra principal es el verbo (“buscar”).

A partir de ahí, el resto de palabras quedan relacionadas dos a dos según su dependencia. En este caso, las dependencias están codificadas siguiendo el modelo de las Dependencias Universales (*Universal Dependencies*). La siguiente tabla 4.2 muestra las palabras relacionadas y el tipo de relación: sujeto, complemento directo, circunstancial, etc.

[17] indican que, frente al análisis de constituyentes, el análisis de dependencias presenta algunas ventajas:

1. Es más fácil extraer las relaciones entre las palabras en el modelo de dependencias, ya que están codificadas directamente. Por ejemplo, en la Fig. 4.2, para saber qué palabra actúa como SN_Sujeto del verbo *buscar* hay que recorrer el árbol cuatro pasos (subir a la raíz S y descender por

NP), mientras que en la representación por dependencias de la Fig. 4.4 es solo un paso.

2. El modelo de constituyentes, dado que hace agrupaciones, es bastante dependiente de la posición de las palabras en la oración, mientras que el de dependencias no. Esto lo hace un formalismo especialmente útil para lenguas con un orden de palabras relativamente libre y de morfología rica,¹ como el vasco. Por esto mismo, el modelo de dependencias es apropiado para analizar, precisamente, los cambios de posición (el hipérbaton), ya que permite codificar por separado la relación sintáctica y la posición de las palabras en la oración.²
3. Las relaciones de dependencias, con ser relaciones sintácticas, se acercan a las relaciones semánticas entre los constituyentes de la oración. Es, por tanto, un formalismo útil para una posterior análisis semántico como el análisis de roles semánticos o eventos. *Ver capítulo ??*.

Tipos de relaciones de dependencias. Las dependencias universales

Los modelos lingüísticos basados en dependencias tienen una larga tradición en Lingüística teórica, que se remonta a los clásicos greco-latinos o indios. En el siglo XX destacan sobre todo los trabajos de Tesnière o modelos formales como Meaning-Text Theory de Mel'cuk (1988), Word Grammar de Hudson (1984) o la Functional Generative Description (FDG) de Sgall et al. (1986). No se entrará en los detalles de estos modelos.

Desde la Lingüística computacional se ha desarrollado un modelo propio, que es hoy día en estándar en análisis de dependencias: las *Universal Dependencies* comentadas anteriormente.³

El objetivo de las UD es crear un modelo de representación gramatical formal común a gran cantidad de idiomas (para todos los idiomas, si es posible), que dé cuenta de las categorías gramaticales, análisis morfológico y análisis sintáctico de dependencias.

En la práctica, el modelo de las UD tiene dos partes. Una primera parte es la universal, la común a todos los idiomas. Ésta incluye por un lado un in-

¹Es precisamente la riqueza morfológica lo que permite a estas lenguas libertad en la posición de los constituyentes.

²Sobre codificación de información posicional en redes neuronales, véase el trabajo de Dufter et al "Position information in transformers: An Overview." <https://direct.mit.edu/coli/article/48/3/733/111478/Position-Information-in-Transformers-An-Overview>

³<https://universaldependencies.org/>

ventorio de categorías gramaticales, morfológicas y sintácticas presentes, en principio, en cualquier idioma; y por otro unas guías de anotación para asegurar la mayor consistencia posible en los procesos de anotación de corpus. Con ello se obtiene una simbología de representación sintáctica común a todos los idiomas, que permite el análisis comparativos entre idiomas y el desarrollo de sistemas de PLN multilingües.

Esta parte común se completa con una segunda parte específica de cada idioma. En esta se concretan aquellos aspectos (categoriales, morfológicos o sintácticos) propios de cada idioma que no tienen correlato en el resto de idiomas (o al menos no en todos). Son, por tanto, los rasgos sintácticos no universales.

Por lo que respecta al análisis de dependencias sintácticas, las categorías universales representan tipos de dependencias entre constituyentes⁴. Distingue primero tres tipos de dependencias: argumentales (*core arguments*), no argumentales (*Non-core dependents*) y nominales. A su vez, el constituyente dependiente puede ser un nominal (que puede ser tanto un sintagma nominal en sentido estricto como un preposicional), una cláusula, un modificador o una palabra funcional.

Las dependencias argumentales (entre verbo y complemento) de un nominal son tres:

1. *nsubj*: sujeto nominal.
2. *obj*: objeto (directo).
3. *iobj*: objeto indirecto.

Para una explicación completa de las etiquetas de representación sintáctica universal, ver <https://universaldependencies.org/u/dep/> y <https://universaldependencies.org/>.

El formato de representación para este tipo de análisis más común hoy día es el formato CONLL. Éste es un formato tabular en el que cada línea representa una palabra (*token*) del texto en orden secuencial, y cada columna información morfológica, categorial o sintáctica. El cuadro 4.3 muestra un ejemplo simplificado.

La información sintáctica de dependencias está codificada en las columnas 5 (Depend) y 6 (Tipo_depend). Cada palabra tiene un identificador único (primera columna). La columna 5 indica el identificador de la palabra con la que tienen algún tipo de dependencia, y la columna 6 el tipo de dependencia.

⁴Ver <https://universaldependencies.org/u/dep/>

ID	Token	Lema	Cat_gram	Depend	Tipo_depend.
1	Los	el	DET	2	det
2	hombres	hombre	NOUN	6	nsubj
3	que	que	PRON	4	nsubj
4	fuman	fumar	VERB	2	acl
5	puro	puro	ADJ	4	obj
6	tienen	tener	VERB	0	root
7	cara	cara	NOUN	6	obj
8	de	de	ADP	9	case
9	canguro	canguro	NOUN	7	nmod
10	.	.	PUNCT	6	punct

Cuadro 4.3: Análisis de dependencias con formato CONLL de la oración “Los hombres que fuman puro tienen cara de canguro”

Así, por ejemplo, la palabra “hombres” (fila 2) tiene una relación de dependencia con la palabra “tiene” (fila 6) de tipo `nsubj` (sujeto). Efectivamente, “hombres” es el sujeto del verbo “tener” en esta oración.

4.3. Estrategias de análisis

Ambos tipos de representación tiene dos estrategias de análisis básicas: ascendentes y descendentes. Las primeras parte de la información categorial de las palabras y van agrupando símbolos hasta llegar al nodo oración. La segunda parte del nodo principal de oración y, según las reglas, deriva la estructura hasta llegar a las categorías gramaticales y las palabras.

Con la herramienta *Natural Language Toolkit* (NLTK: <https://www.nltk.org/>) se puede ver una representación visual de ambas estrategias. Con Python, tras instalar la herramienta, ejecuta, para ver un análisis recursivo descendente:

```
import nltk
nltk.app.rdparser()
```

Y para un análisis “desplaza y reduce” (*shift reduce*) descendente:

```
import nltk
nltk.app.srparser()
```

El sistema “transition-based dependency parsing” de Nivre (2014) es el sistema de análisis de dependencias estándar con el algoritmo *shift-reduce*.

Situación actual

La aplicación de técnicas de aprendizaje supervisado al análisis sintáctico depende de la disponibilidad de corpus anotados con árboles sintácticos. Los llamados *tree banks*. Los principales, que han marcado el desarrollo de otros corpus, son los siguientes:

- Par inglés, el *Penn Treebank*:
 - <https://catalog.ldc.upenn.edu/LDC99T42>
 - <https://www.kaggle.com/nltkdata/penn-tree-bank>
- Para español y catalán, el corpus ANCORA:
 - <https://clic.ub.edu/corpus/es/ancora>
 - <http://clic.ub.edu/corpus/en/ancora-descarregues>

Como en todos los ámbitos del PLN, hoy los sistemas siguen el modelo neuronal. La aproximación neuronal estándar es el *Neural Network Dependency Parser* de la U. de Stanford. Ver <https://nlp.stanford.edu/software/nndep.shtml>. Otras aproximaciones pueden ser:

- Parsing as language modeling <https://aclanthology.org/D16-1257/>
- Recurrent Neural Network Grammars <https://arxiv.org/abs/1602.07776>
- Seq2seq Dependency Parsing <https://aclanthology.org/C18-1271/>
- Graph-based Dependency Parsing with Graph Neural Networks <https://aclanthology.org/P19-1237/>
- entre otros.

Dado que los modelos neuronales son opacos en el tratamiento de la información, hay varios trabajos que tratan de hacer explícito (mediante etiquetas) el análisis sintáctico de los modelos neuronales gracias, sobre todo, al mecanismo de atención (que se verá en próximos temas). Ver:

- What Do Recurrent Neural Network Grammars Learn About Syntax? <https://arxiv.org/abs/1611.05774>
- Rethinking Self-Attention: Towards Interpretability in Neural Parsing <https://arxiv.org/abs/1911.03875>

entre otros estudios.

4.4. Herramientas

- SpaCy: <https://spacy.io/> (<https://spacy.io/>)
- STANZA: <https://stanfordnlp.github.io/stanza/>
- Freeling: <https://nlp.lsi.upc.edu/freeling/node/1>
- UD-Pipe: <https://ufal.mff.cuni.cz/udpipe>
- Apache OpenNLP <https://opennlp.apache.org/>
- DKPro <https://dkpro.github.io/>

4.5. Lecturas opcionales

Para profundizar en el análisis sintáctico computacional, véanse los capítulos 17 (“Context-Free Grammars and Constituency Parsing”) y capítulo 18 “Dependency Parsing” de [17]:

- <https://web.stanford.edu/~jurafsky/slp3/17.pdf>
- <https://web.stanford.edu/~jurafsky/slp3/18.pdf>

Modelos simbólicos de representación semántica

Los modelos simbólicos anteriores se caracterizan por representar aspectos formales de las lenguas. Al ser aspectos formales, es posible representarlos y analizarlos con rasgos más o menos definidos. La semántica es totalmente diferente pues es muy difícil formalizarla.

La semántica estudia significado de los textos. El propio concepto de significado es, sin embargo, bastante vago pues se hace referencia a muchas cosas. Como se comentó antes, a un sistema de PLN le interesa hallar el significado de todo el texto, pero éste está formado a partir del significado de las palabras y las oraciones (principio de composicionalidad). Se habla, por tanto, de significado a diferentes niveles (léxico, oracional y textual). Además del significado denotativo (que se puede más o menos representar formalmente), hay también un significado connotativo, más relacionado con las emociones, sentimientos, evocaciones, recuerdos, etc. que nos puede producir un texto, todo ello de carácter muy subjetivo. Significado también es todo el conocimiento que se infiere durante la interpretación del texto: éste no contiene toda la información necesaria para entenderlo. Durante la interpretación de un texto se activa información de nuestro conocimiento de mundo necesaria para entender el texto (información sobre-entendida), y además aplicamos procesos lógicos al significado del texto para que éste tenga sentido y sea coherente con nuestra visión del mundo.

El caso es que no hay ningún sistema simbólico que pueda dar cuenta de todo lo que es el significado y en toda su complejidad. Los sistemas simbólicos de PLN son representaciones parciales de algún aspecto del significado. En este tema se van a exponer los principales: qué aspectos semántica representan formalmente y cómo lo hacen. En el siguiente tema se mostrarán los modelos semánticos conexionalistas.

5.1. Significado como representación lógica

Los antecedentes de la semántica computacional están en la lógica formal. Por influencia de la lingüística matemática, la representación semántica de los primeros sistemas de PLN era representaciones basadas en lógica de predicados o de primer orden. Con ello se consigue una representación no ambigua del texto que, además, permite hacer inferencias lógicas. La denotación de las palabras se representan mediante términos (constantes o variables) que, junto con los predicados (que relacionan términos), permite expresar el significado de oraciones. Véase Jurafsky y Martin (2023) “Logical Representations of Sentence Meaning”, *Speech and language processin*, cap. 19.

5.2. Semántica léxica y el *word sense disambiguation*

La semántica léxica se refiere al significado de las palabras. Las palabras son las unidades mínimas del idioma con significado pleno. Una sola palabra, emitida en una situación comunicativa concreta, puede ser un texto completo con sentido (el caso de “fuego” comentado en capítulos anteriores). Así, en el significado de las palabras se suele diferenciar por un lado el significado sistémico, es decir, todo el conjunto de significados que puede tener una palabra (como muestra, por ejemplo, un diccionario); y el significado contextual, que es aquél que se instancia en un contexto concreto.

Por otro lado, en PLN hay actualmente dos modelos para tratar la semántica léxica:

- La consideración del sentido de la palabra como una representación discreta, es decir, un conjunto de definiciones (una o más) en un diccionario.
- La consideración del sentido a partir de las relaciones contextuales (distribucionales) entre las palabras en su uso real (en un corpus, por ejemplo). Este modelo es la base de la semántica vectorial, de la cual surgen los *word embeddings* que se verán en el próximo tema.

Esta sección se centra en el primer modelo.

Significado léxico como unidad discreta

Según este modelo una palabra puede tener uno o más significados que además podemos especificar en un diccionario. Las palabras que, fuera de contexto, tienen dos o más significados son palabras ambiguas. Se calcula que más del 60 % de las palabras de un idioma son ambiguas: basta echar un vistazo a un diccionario para comprobarlo. En un texto concreta esa ambigüedad se reduce, de tal manera que un ser humano al interpretarlo es capaz de determinar, a partir del conjunto de posibles significados de esa palabra, el sentido apropiado para ese contexto.

Un ejemplo de miles que podríamos exponer es la palabra “ratón”, que entre otros puede tener dos significados dispares:

- “Mamífero roedor de pequeño tamaño, de hocico puntiagudo y cola larga, de pelaje corto”
- “Pequeño aparato manual conectado a una computadora u otro dispositivo electrónico, cuya función es mover el cursor en la pantalla para dar órdenes.”(RAE)

“El ratón muerde” o “El ratón no funciona” es suficiente contexto para instanciar un significado y otro. Este es el modelo de semántica léxica que conocemos desde el colegio, en el que nos pedían buscar palabras en un diccionario y determinar cuál era el significado apropiado según el texto.

Que una palabra tenga dos o más significados puede parecer en un principio ilógico. Este hecho se debe a dos fenómenos lingüísticos: la homonimia y la polisemia.

La **homonimia** se produce cuando dos palabras, en un principio diferentes en significante y significado, han evolucionado de tal manera que sus significantes (es decir, la forma de la palabra, cómo se pronuncia o escribe) se han hecho iguales. Así ocurre por ejemplo con palabras como “bota”, que puede ser el odre para beber (la bota de vino), procedente del latín “buttis”; o la bota de calzado, procedente del francés (“botte”).¹ Son por tanto dos palabras distintas que, por evolución, ahora se pronuncian igual. Las diferencias semánticas en los casos de homonimia son muy grandes por ser palabras diferentes.

La *polisemia* se produce por la evolución del propio significado de una palabra. Dada una palabra, el uso diario puede producir que genere un nuevo significado por procesos de metaforización (“ratón”), metonimia (“pluma”), especificación (“banco entidad” vs. “banco edificio”), etc. Desde un punto de

¹Ver <http://www.wikilengua.org/index.php/Homonimia>

vista computacional, la polisemia es más compleja de procesar que la homonimia, pues estos significados nuevos siempre están relacionados con el significado original y sus contextos de uso son parecidos.

Word Sense Disambiguation

El significado léxico como unidad discreta necesita, por tanto, de un diccionario donde estén recopilados todos los posibles significados de cada palabra. El proceso de análisis es similar al análisis categorial: seleccionar el significado apropiado para el contexto donde aparece la palabra a analizar.

En PLN, este análisis semántico consistente en seleccionar el significado apropiado para un contexto a partir de los significados establecidos en un diccionario se denomina *Word Sense Disambiguation* (WSD). Es una de las tareas del PLN con más tradición, junto al análisis categorial o el análisis sintáctico que vimos en temas anteriores.

Los sistemas de WSD están formados, por tanto, por dos componentes fundamentales: un diccionario en el que se representan todos los significados de las palabras y un algoritmo de desambiguación.

Representación del significado léxico: WordNet.

El principal diccionario electrónico utilizado en PLN para semántica léxica es WordNet² [22, 12]. En su origen fue un diccionario para el inglés, pero luego ha sido ampliado a lenguas europeas (EuroWordNet) y otras familias lingüísticas (balkanet, arabic wordnet, etc.) y más tarde a todas las lenguas del mundo con Global WordNet (<http://globalwordnet.org/>). Todos ellos se pueden consultar en el *Open Multilingual Wordnet*: <http://compling.hss.ntu.edu.sg/omw/>. WordNet en inglés se puede consultar desde su página oficial <http://wordnetweb.princeton.edu/perl/webwn>.

Las características principales de WordNet, que lo diferencian de otros diccionarios electrónicos, son:

- WordNet es una red de sentidos. Cada nodo de la red representa un posible sentido. La unidad estructural del diccionario no es la palabra, como en otros diccionarios electrónicos, sino el sentido.
- Cada sentido se representa mediante el conjunto de palabras sinónimas en un idioma. Esto forma el denominado *synset*. Un *synset* es un nodo de la red, pues representa un significado, y tiene asociado todas las palabras que pueden expresar ese significado. Una palabra con dos o más significados estará asociada a dos o más *synsets*.

²<https://wordnet.princeton.edu/>

- Cada *synset*, además del conjunto de sinónimos, dispone de información léxica como un ID único de sentido, ejemplos, glosas o definiciones, conceptos de dominio, etc. Excepto el ID, el resto de información es opcional.
- La red entre sentidos se forma a partir de relaciones léxicas. La principal relación léxica es la sinonimia a partir de la cual se forma el propio *synset*.
- Entre nombres, las relaciones principales son hiperonimia (relación “is_a”), la hiponimia (que es la relación inversa de la hiperonimia) y meronimia (relación “parte_de”).
- Para los adjetivos, las relaciones principales son la antonimia y la relación “similar a”.
- Para los verbos, las principales relaciones son la llamada “pseudo-hiperonimia” (se considera “pseudo” porque, en sentido estricto, la hiperonimia solo se produce entre nombres) y la troponimia: manera de realizar una acción. Por ejemplo, pasear es una manera de andar, por lo que entre los verbos “pasear” y “andar” hay una relación de troponimia.³

WordNet tiene unos nodos generales con sentidos muy abstractos (.^{en}tidad”) de las cuales van derivando por relación léxica el resto de sentidos hasta los más concretos. Así, si bien es una red, tiene en cierta manera una relación arbórea por las relaciones tipo “is a”.

En su concepción original, WordNet pretendía ser una representación computacional del lexicón humano: la organización del léxico en la mente humana. Finalmente se ha convertido en quizá el principal recurso para el análisis léxico-semántico. WordNet es el estándar *de facto* para la representación semántica léxica de un corpus.

Algoritmos de desambiguación léxico-semántica.

WSD asume que los significados de una palabra son unidades atómicas y discretas que están pre-establecidos en un diccionario (WordNet normalmente). No se plantean otras formas de significación como la metáfora y los sentidos figurados, en los que las palabras asumen un significado del que tiene pre-asignado en el diccionario, u otros aspectos como inferencias o conocimiento del mundo.

³En las últimas versiones se pueden encontrar otros tipos de relaciones que no vamos a tratar aquí.

La complejidad en WSD es determinar, de los posibles *synsets* asociados a una palabra (nombre, verbo o adjetivo), cuál es el apropiado en un contexto dado. La heurística básica es seleccionar siempre el sentido más frecuente. A partir de ahí, en los últimos 30 años se han propuesto diferentes algoritmos. En general, hay dos aproximaciones: algoritmos basados en conocimiento y algoritmos basados en aprendizaje supervisado.

Las estrategias basadas en conocimiento (knowledge-based) se caracterizan por explotar al máximo la información del recurso léxico (WordNet) comparando esa información con la que aporta el contexto donde aparece la palabra ambigua. El algoritmo de Lesk (Lesk 1986) es el método estándar de desambiguación basando en conocimiento. Para determinar el sentido apropiado de una palabra ambigua, compara las palabras del contexto donde aparece con la definición de cada sentido (en WordNet a la definición la denominan “glosa”). Finalmente selecciona como sentido apropiado aquél cuya definición tiene más coincidencias con el contexto.

Por ejemplo, según este uso de “banco”, quedaría claro que el sentido apropiado es el primero:

“Ingresé el *dinero* en el **banco** ayer tarde”

1. Entidad financiera que acepta *dinero* en depósito y ofrece préstamos con intereses.
2. Asiento largo y estrecho para varias personas.

Siempre y cuando se disponga de una buena definición, esta aproximación puede funcionar bien para casos de homonimia, pero no tanto para casos de polisemia.⁴

En esta línea, el algoritmo UKB [1, 25] es una aproximación mucho más avanzada. El algoritmo es una adaptación del algoritmo Page Rank de Google. La idea principal de éste es que no todos los nodos de un grafo son iguales, sino que unos tienen más importancia que otros. Un nodo es importante si otros nodos apuntan a él, y si un nodo importante apunta a otro, este también se considera relativamente importante. Sin entrar en detalles técnicos, este algoritmo determina la pertinencia de un sentido en un contexto mediante las relaciones léxicas de cada palabra dentro de WordNet. Dada una palabra am-

⁴Existen diferentes implementaciones de este algoritmo, como por ejemplo la disponible en el *Natural Language Toolkit* (NLTK) [5]: <https://www.nltk.org/howto/wordnet.html>, pero solo funciona para inglés.

bigua, aquel *synset* cuyas relaciones de hiperonimia, hiponimia, etc. mejor encaje con el resto de sentidos del contexto, será el apropiado [1].⁵

Las estrategias basadas en aprendizaje supervisado (*features-based algorithms*) se caracterizan por aprender diferentes rasgos del contexto de las palabras y utilizarlos para clasificar usos ambiguos. Por ejemplo, una estrategia óptima sería crear un clasificador basado en SVM con rasgos de aprendizaje como pudieran ser las categorías gramaticales de las tres palabras anteriores, n-gramas de las palabras alrededor de la palabra ambigua, o un vector contextual a partir de los vectores incrustados (*embeddings*) de cada palabra del contexto (Jurafsky y Martin, cap. 18, pág. 12).

Existen diferentes corpus anotados con sentidos desambiguados. El primero en ser desarrollado fue SemCor, con texto en inglés. Este corpus es el modelo a partir del cual se han desarrollado otros. El corpus se creó al mismo tiempo que WordNet y por las mismas personas. En SemCor, cada palabra tiene asignado el *synset* específico en WordNet. Y muchos sentidos de WordNet se han determinado a partir de los textos de SemCor. SemCor está disponible en diferentes páginas como éstas:

- http://www.gabormelli.com/RKB/SemCor_Corpus
- <http://web.eecs.umich.edu/~mihalcea/downloads.html#semcor>
- <https://www.kaggle.com/nltkdata/semcor-corpus>
- https://www.nltk.org/_modules/nltk/corpus/reader/semcor.html

A partir de SemCor se han creado corpus anotados con sentidos de WordNet para otros idiomas. Para español se creó el corpus Cast3LB, hoy enriquecido con más información y renombrado como Ancora Corpus: <http://clic.ub.edu/corpus/es/ancora>.

Otro recurso interesante para WSD es Nasari, que incluye representaciones vectoriales (ver próximo tema) de los *synsets* de WordNet y de la Wikipedia (ambos integrados en el recurso BabelNet): <http://lcl.uniroma1.it/nasari/>.

Junto a estas dos estrategias, hay una tercera basada en técnicas no supervisadas. En este caso, no hablamos de desambiguación de sentidos sin de inducción de sentidos. Al no haber un recurso léxico de referencia con los sentidos, los sistemas no determinan significados sino que agrupan oraciones. Dado un conjunto de oraciones con una palabra ambigua en común, agrupan las oraciones en las que esa palabra se utiliza con un sentido determinado.

⁵Este es el algoritmo implementado en *Freeling*: <http://nlp.lsi.upc.edu/freeling/>.

Lo que no pueden determinar es cuál es ese sentido. En esta tareas se suelen aplicar modelos de semántica vectorial que se verán en la próxima sesión.

5.3. Semántica oracional. Roles semánticos y semántica de eventos.

Los sistemas de WSD se centran únicamente en determinar el significado de las palabras. Sin embargo, el significado global de un texto no solo depende del significado de las palabras que lo forma, sino también de las relaciones que se establecen entre ellas tanto en la oración como en la globalidad del texto (principio de composicionalidad).

La semántica oracional se centra en estudiar el significado de la oración en su conjunto. Dentro del PLN hay diferentes aproximaciones a la semántica oracional, de las que destaca sobre todo el análisis de roles semánticos (Gildea y Jurafsky 2002).

Los roles semánticos se enmarcan dentro de la semántica eventiva o semántica de eventos. El objeto de esta aproximación semántica es determinar los eventos y estados expresados en un texto junto con sus participantes y las relaciones entre ellos. Dada, por ejemplo, una oración, el evento suele venir expresado por el verbo y los participantes por sus argumentos. Los roles semánticos representan la relación semántica de esos argumentos con el sentido verbal dentro del marco eventivo (Levin et al. 2005).

Por ejemplo, la oración

“Las fuerzas de seguridad persiguieron a los agresores”

expresa el evento “perseguir” que tiene una estructura argumental formada por la persona que persigue (“las fuerzas de seguridad”) y la persona perseguida (“los agresores”). El primer argumento se podría considerar como rol semántico “agente” y el segundo como rol “tema”.

Un evento puede estar expresado tanto por verbo (“luchar”) como por un nombre (“la guerra”). Los argumentos son los sintagmas que completan el significado del evento. La función semántica que pueden asumir los argumento es lo que se denomina “roles semánticos”.

Representación formal de los roles semánticos.

La teoría de los roles semánticos proviene de la Teoría de Casos de Ch. Fillmore y ha tenido diferentes desarrollos en lingüística teórica. El problema

principal que tienen estas teorías es que, por un lado, no se ha podido consensuar una única lista de roles semánticos y, por otro, no hay una clara distinción entre los roles. Así, al hablar de roles semánticos nos podemos encontrar roles como:

- AGENTE: el argumento que realiza la acción del evento;
- PACIENTE O TEMA: el argumento sobre el que actúa el evento);
- EXPERIMENTANTE: el argumento que experimenta la acción expresada por el verbo;
- INSTRUMENTO;
- DIRECCIÓN O META;
- LOCALIZACIÓN;
- PROTO-AGENTE y PROTO-PACIENTE: generalizaciones de agente y paciente;
- etc.⁶

Esta falta de acuerdo en los estudios lingüísticos ha propiciado el desarrollo de dos modelos de representación de roles semántico en PLN: el modelo de FrameNet y el modelo de PropBank.

FrameNet

FrameNet⁷ [3] propone una representación de roles semánticos muy fina: indica roles específicos para unidades léxicas concretas. Estas unidades pueden ser verbos, nombres o adjetivos. Cada uno de sus sentidos se agrupa en un marco semántico, entendido como un marco estructural conceptual (*frame*) que describe una situación, un objeto o un evento concreto más sus participantes: los roles semánticos asociados a ese marco (*frame elements*).

Por ejemplo, la unidad léxica “comer” pertenece al marco semántico INGESTION. En este marco semántico se han definido hasta siete elementos, entre los que se encuentran:

⁶Ver diferentes propuestas en el estándar EAGLES (1996): <https://www.ilc.cnr.it/EAGLES96/rep2/node8.html> o en la propuesta de T. Payne (2007): <https://pages.uoregon.edu/tpayne/EG595/H0-Srs-and-GRs.pdf>, entre otras muchas que se podrían plantear

⁷Ver Ruppenhofer, J., M. Ellsworth, M. R. L. Petruck, C. R. Johnson, C. F. Baker y J. Scheffczyk. 2016. *FrameNet II: Extended Theory and Practice*. <https://framenet2.icsi.berkeley.edu/docs/r1.7/book.pdf>.

- INGESTOR (“comensa”),
- INGESTIBLES (“comida” o “digeribles”),
- PLACE (“lugar” donde se come),
- MANNER (“manera” de comer)
- DEGREE (“cantidad”).

La representación de roles semánticos de la oración

“Alba aprendió a comer verduras hervidas y arroz quemado”

sería la siguiente:

(_INGESTOR Alba) aprendió a COMER _TARGET (_INGESTIBLES verduras hervidas y arroz quemado)

Del marco semántico general de “comer”, en esta oración se han instanciado, junto al propio evento (el *target*), dos roles: quién come (INGESTOR) y qué come (INGESTIBLES).

En general, hay tres tipos de *frame elements*:

1. *core*: aquellos que son específicos del evento y conceptualmente necesarios para que el marco tenga sentido completo;
2. *peripheral*: aquellos que aportan información importante para completar el marco semántico pero que no son centrales para que éste tenga sentido completo; y
3. *extra-thematic*: aquellos que amplían el contexto semántico del marco.

En el caso del marco INGESTION, los dos elementos *core* son INGESTOR e INGESTIBLES; elementos periféricos son INSTRUMENT o SOURCE, y el resto actuarían como extra-temáticos.

El resto de *frames*, así como información sobre el proyecto, se puede consultar en <https://framenet.icsi.berkeley.edu>

PropBank

La propuesta de PropBank (acrónimo de *Proposition Bank* [26]) es justo la contraria. En vez de definir roles semánticos muy específicos según el evento, PropBank determina pocos roles y muy generales, de tal manera que sean aplicable a cualquier evento. Además, en vez de nombrar los roles un nombre significativo, representa cada rol con un simple identificador. Así, de manera general PropBank establece que puede haber hasta cinco roles semánticos asociados a un evento:

Arg0 | Arg1 | Arg2 | Arg3 | Arg4

y además se establece un número indefinido de adjuntos:

ArgM

Cada rol se define por su relación con el verbo. Los dos argumentos que tienen una relación más estrecha con el sentido del verbo son ARG0 y ARG1. Para verbos transitivos, por ejemplo, el primero se suele identificar con el rol AGENTE y el segundo con el rol TEMA o PACIENTE, pero esta relación no siempre se cumple.

Lo importante es que la alternancia de diátesis no afecte a los roles. Así, independientemente de que la estructura verbal se exprese en activa o en pasiva, los roles ARG0 y ARG1 serán los mismos.

El siguiente ejemplo muestra la misma oración en activa y en pasiva. Al cambiar la voz verbal de una a otra cambian las relaciones sintácticas, pero no cambian las relaciones semánticas (los roles semánticos). Este fenómeno se denomina alternancia de diátesis:

1a. [ARG0 La policía militar] arrestó [ARG1 a tres personas]

1b. [ARG1 Tres personas] fueron arrestadas por [ARG0 la policía militar]

Este modelo ha sido adaptado al español en el corpus AnCora⁸ (Taulé et al. 2008), que también incluye anotación de textos en catalán (AnCora-Es y AnCora-Cat respectivamente).

De ambas propuestas de representación de roles semánticos, la más utilizada hoy día en PLN es la propuesta de PropBank.

⁸(<http://clic.ub.edu/corpus/es/ancora>)

Se pueden consultar los roles de PropBank de cada verbo en su base de datos unificada (*Unified Verb Index*): <https://verbs.colorado.edu/verb-index/vn3.3/>.

También se puede descargar desde su Github <https://github.com/propbank/propbank-frames/>. La web del proyecto: <https://propbank.github.io/>

Algoritmos de análisis de roles

Los sistemas de análisis de roles semánticos (*semantic role labeling*) clásicos toman como entrada un corpus anotado con categorías gramaticales y (en algunos casos, pero no siempre) con relaciones sintácticas. La salida es la especificación de qué elemento expresa el evento, qué palabras se agrupan en cada argumento y el tipo de argumento.

Los principales algoritmos de *semantic roles labeling* (SRL) suelen estar basados en técnicas de aprendizaje supervisado. A partir de corpus anotados con roles (como el propio corpus PropBank, se establecen una serie de rasgos de aprendizaje que se utilizan luego para clasificar por tipos de roles semánticos.

El algoritmo estándar de SRL es el de Gildea y Jurafsky (2002). Este sistema primero aprende de un corpus anotado qué elementos son los roles semánticos y de qué tipo son, junto a una serie de rasgos lingüísticos. Entre los rasgos utilizados está el verbo que rige la estructura argumental, los tipos de sintagma de los argumentos, la categoría gramatical de las palabras de cada argumento, los lemas de las palabras, etc. Es decir, tanto información categorial como sintáctica. Durante el proceso de análisis de un nuevo corpus, el algoritmo tratará de determinar los roles semánticos de una oración a partir de estos rasgos.

El modelo de Freeling, para español y otros idiomas, es similar. En el caso del español está entrenado con el corpus AnCora y entre los rasgos de aprendizaje utiliza, además de los establecidos en Gildea y Jurafsky (2002) otros como las relaciones de dependencia o la voz verbal (Lluís et al. 2013).

Los sistemas actuales, como el resto de tareas, están basados en modelos neuronales y *word embeddings*. También hay interés en desarrollar sistemas multi- y cross-lingües, como <https://www.aclweb.org/anthology/2020.acl-main.627/>

En este tema se han visto los dos tipos principales de representación semántica en PLN: la semántica léxica basada en WordNet y los roles semánticos. Hay otros tipos de representaciones semánticas en PLN, como *Abstract Meaning Representation*⁹ (que a los roles de PropBank une correferencia, tipos de entidades nombradas, modalidad, negación y algunas cuestiones más

⁹<https://amr.isi.edu/index.html>

en una representación mediante grafos), *Discourse Representation Theory* (que agrupa roles semánticos, *wsd*, correferencia y tipos de entidades), entre otras.

5.4. Lecturas opcionales

Para profundizar en estos temas, véanse los capítulos 32 (“Word Senses and WordNet”) y 24 (“Semantic Role Labeling”) de [17]:

Una visión sencilla, general y en español de estos y otros aspectos de semántica computacional, se pueden ver en [23].

Para otros modelos de análisis semántico, se pueden consultar, de [17], los capítulos 19 (“Logical Representations of Sentence Meaning”), 21 (“Relation and Event Extraction”), 22 (“Time and Temporal Reasoning”) y 25 (“Lexicons for Sentiment, Affect, and Connotation”).

Modelos vectoriales de representación semántica

En este tema veremos:

- los fundamentos de la semántica distribucional como modelo teórico,
- la representación vectorial de las relaciones semánticas distribucionales y los factores que lo determinan,
- los conceptos de distancia y similitud como métodos de interpretación, y las medidas básicas.

6.1. Lecturas

Para completar este tema, se recomiendan las siguientes lecturas (opcionales):

- El capítulo 6 “Vector Semantics and Embeddings” de Jurafsky and Martin (2023) *Speech and language processing* completa todos los aspectos aquí planteados. Sobre todo el punto 6.8, donde se explica cómo se calculan los *word embeddings*. Disponible aquí: <https://web.stanford.edu/~jurafsky/slp3/6.pdf>
- Peter D. Turney y Patrick Pantel (2010) “From Frequency to Meaning: Vector Space Models of Semantics” en *Journal of Artificial Intelligence Research*, 37, págs. 141-188. DOI: <https://doi.org/10.1613/jair.2934>. Disponible aquí <https://www.jair.org/index.php/jair/article/view/10640>.

Este artículo explica bien cómo un vector de frecuencias puede representar el significado de una palabra. El trabajo es ya antiguo, de 2010, anterior al desarrollo de los modelos neuronales y los *word embeddings*, por lo que hay aspectos hoy día ya desfasado. Son de interés, sobre todo, los primeros epígrafes.

6.2. Introducción

Los modelos semánticos vectoriales son una aproximación formal para la representación de la semántica de un texto totalmente diferente a los modelos simbólicos vistos anteriormente. La diferencia viene dada tanto por el modelo teórico subyacente como por el formalismo en sí mismo:

- El modelo teórico subyacente es la semántica distribucional, que asume que el significado de una palabra deriva de su uso concreto en un contexto determinado.
- El formalismo de representación son espacios vectoriales multidimensionales, que es una representación propia del modelo conexionista.

6.3. La semántica distribucional

El modelo semántico distribucional tiene su origen en la teoría distribucional del lenguaje, que se inició a mediados del s. XX y que se caracteriza, sobre todo, por la importancia que le da al contexto en todos los niveles lingüísticos.

En el capítulo anterior se vio que el significado se podía caracterizar de diferentes maneras. Así, el modelo de base lógico considera el significado de una palabra como una unidad atómica (el significado de “casa” sería CASA, o como variable $casa(x)$); el modelo lexicográfico considera que el significado de una palabra es también una unidad discreta y se representa mediante la definición de un diccionario; o el modelo léxico de WordNet, que considera que el significado de las palabras es un nodo finito (el *synset*) dentro de una red de relaciones léxicas tipo hiperonimia, hiponimia, etc.

El planteamiento semántico-vectorial es diferente. No se considera el significado como una unidad atómica y finita. Más bien se considera el significado a partir de las relaciones contextuales que una palabra tiene en los contextos donde aparece. Este modelo está basado en los siguientes planteamientos lingüísticos [9]:

1. La idea de Wittgenstein de “meaning just is use” [32], es decir, que más allá de diccionarios, estudios y academias; una palabra tendrá el significado que le den los hablantes cuando la usen en textos reales. El significado es simplemente el uso que se les dé a las palabras.
2. El concepto de *collocation* de Firth (1957) y su idea de que

“you shall know a word by the company it keeps”. [13]

Dicho con otras palabras, según esta idea es posible conocer cómo son las palabras, incluido su significado, estudiando con qué otras palabras puede aparecer, es decir, estudiando su contexto. En español podrías decir de las palabras aquello de “dime con quién andas, y te diré quién eres”.

3. La hipótesis distribucional de [16], que dice

“words will occur in similar contexts if and only if they have similar meanings”.

O dicho al contrario, un significado similar implica un contexto similar. De tal manera que podríamos determinar la similitud semántica entre dos palabras a partir de la similitud entre sus contextos.

En realidad, esto de determinar el significado de una palabra a partir de las palabras del contexto es algo que hacemos constantemente. Lee las siguientes oraciones, ¿qué significado tiene XXX en cada una?

- Mañana iré al XXX a firmar la hipoteca, y ya de paso sacaré dinero del cajero.
- He intentado ponerme los XXX de mi hermano pero me vienen pequeños: mis pies son muy grandes y necesito una talla más.

Esta idea de determinar palabras a partir de huecos será clave para la creación de los grandes modelos de lenguaje como BERT o GPT, como se verá en próximos temas.

De todo ello se deduce que una representación formal del contexto de una palabra es una representación computacional de su significado, pues ese contexto está en función de su uso real, especifica las palabras con las que suele aparecer la palabra objeto y permite determinar otras palabras similares a partir de la similitud entre contextos.

Esto es precisamente lo que hacen los modelos semánticos vectoriales. En este caso, para representar de manera formal el contexto, se utilizan vectores.

6.4. Espacio vectorial como modelo de representación formal

La semántica vectorial, por tanto, es una representación formal del significado de las palabras mediante vectores.

Este formalismo se basa, por tanto, en los modelos de espacio vectorial y permiten realizar operaciones sobre vectores y matrices propias del álgebra lineal. También está en la base de las aproximaciones cuánticas al lenguaje (*quantum semantics* en este caso) [10, 21] y el análisis semántico en términos geométricos (cálculo de similitud semántica basadas en *distancias*) [31].

Origen

El origen de la representación vectorial del significado en PLN está en la tarea de recuperación de información (*Information Retrieval*: IR) como son, por ejemplo, los buscadores de internet.

Como ya sabrás, estos sistemas, dada una consulta (formada por una o más palabras o términos), obtienen una lista de documentos ordenados de mayor a menor relevancia según la consulta.

Para ello, antes de la búsqueda en sí, la colección de documentos se transforma en una matriz término-documento. Esta matriz, en su forma más básica, considera el texto como una bolsa de palabras (*bag of words*), es decir, no toma en consideración las relaciones entre las palabras (ver temás anteriores): el texto es solo un conjunto no estructurado de palabras. En la matriz, cada documento es una columna y cada línea un término (palabra). Los valores de la matriz (las celdas) son la frecuencia de cada término en cada documento.

Así, por ejemplo, si asumimos los dos siguientes documentos:

- $\text{doc1} = \{\text{casa}, \text{madera}, \text{mesa}\}$
- $\text{doc2} = \{\text{papel}, \text{rama}, \text{madera}\}$

Se podría crear una matriz como la representada en el Cuadro 6.1.

Con matrices como ésta, para un sistema de RI es posible seleccionar la columna más relevante (texto) a partir de los valores de las filas (palabras) de la consulta. Pero esto ya no interesa aquí. Lo relevante para PLN es cómo esta matriz puede representar el significado de las palabras.

Representación vectorial del significado

Cada línea de este tipo de matrices es un vector que captura, dada una palabra, la relación contextual de ésta con el resto de palabras del corpus. Los

	Doc1	Doc2	...	Doc n
casa	1	0
madera	1	1
mesa	1	0
papel	0	1
rama	0	1

Cuadro 6.1: Matriz término-documento

	doc1	doc2
car	7	6
taxi	5	6
train	6	1

Cuadro 6.2: Matriz término documento (2)

valores del vector muestran la relación contextual (peso, relación, implicación, ...) de esa palabra con el resto: en qué medida dos palabras comparten contexto y con qué frecuencia (peso). En términos distribucionales, ese vector está capturando de manera formal el significado distribucional la palabra.

En definitiva, un vector de este tipo captura la semántica contextual/distribucional de la palabra (*token* o lema) al representar el número de veces (frecuencia) que la palabra aparece en cada contexto (en este caso, el contexto es el documento entero).

Así, de la matriz término - documento del Cuadro 6.2, el significado de cada palabra sería el vector contextual:

$$car = (7, 6)$$

$$taxi = (5, 6)$$

$$train = (6, 1)$$

por lo que realmente ahora el corpus es:

$$corpus = \begin{pmatrix} 7 & 6 \\ 5 & 6 \\ 6 & 1 \end{pmatrix}$$

Esto se puede representar en un espacio euclídeo (plano o lineal) mediante coordenadas cartesianas: los valores del vector se proyectan en los ejes de coordenadas, siendo la abscisa x el documento 1 (primera columna de la matriz) la ordenada y el documento 2 (segunda columna de la matriz). Así,

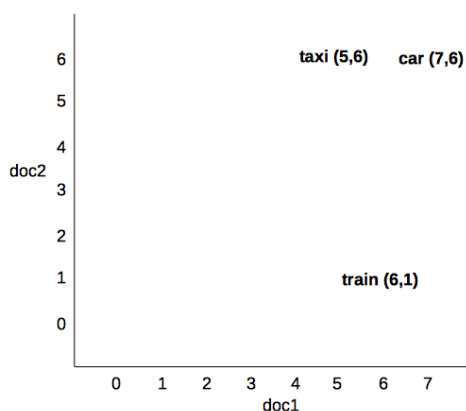


Figura 6.1: Representación vectorial del significado léxico

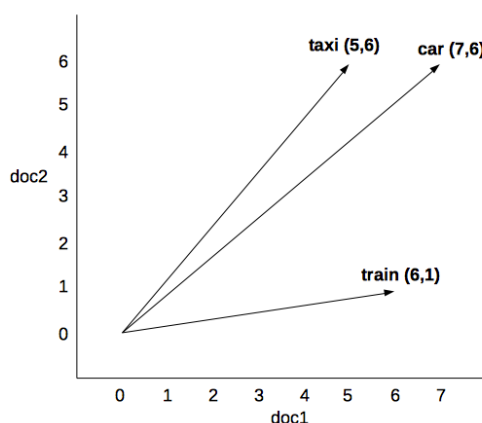


Figura 6.2: Representación vectorial del significado léxico

el significado de cada palabra es un punto en el espacio euclídeo (Figura 6.1). Si unimos ese punto con el origen $(0,0)$, se obtiene el vector en términos matemáticos (Figura 6.1).

En este caso, para poder visualizarlo, el corpus está formado por solo dos textos (documentos) y por tanto el espacio es bidimensional. En aplicaciones reales hay n documentos, por lo que el espacio es multidimensional.

Estos vectores n -dimensionales que representan el significado contextual de las palabras ya se podría denominar *word embeddings*, que son la representación semántica que utilizan los modernos modelos neuronales. Pero para ser operativos es necesario reducir la dimensionalidad de la matriz y su dispersión (cantidad de ceros). Estos dos problemas serán tratados en próximos

temas.

La representación del significado varía según se diseñe el espacio vectorial. Esta representación depende de tres factores:

1. Representación del contexto (dimensiones)
2. Representación de las palabras (filas)
3. Valores o pesos de cada palabra en cada contexto.

Representación del contexto (dimensiones)

Cada contexto de la palabra será una dimensión de la matriz. El problema es cómo delimitar este contexto: ¿cuántas palabras forman el contexto?, ¿dónde está el límite del contexto?

En el modelo de matriz término-documento que se utiliza en recuperación de información el contexto es todo el documento (todo el texto) porque son documentos lo que se quiere recuperar, pero se puede limitar a recuperación de pasajes, párrafos, etc.; es decir, se puede limitar el contexto según se quiera una representación u otra. Otras opciones de representación del contexto con motivación lingüística podrían ser:

- la oración,
- una ventana deslizante (un conjunto de palabras delante y detrás de la término),
- el párrafo o cualquier otra unidad textual,
- el capítulo,
- etc.

Por otro lado, además de la matriz término-documento que hemos visto (donde las columnas representan documentos y las filas palabras), se puede crear otro tipo de matriz: la llamada matriz de co-ocurrencias o matriz término-término. En estas matrices (normalmente cuadradas), tanto las columnas como las filas representan palabra, y los valores la relación entre esas dos palabras. Por ejemplo, esos valores pueden representar en cuántos contextos aparecen juntas esas dos palabras, como en el Cuadro 6.3:

Según esta matriz, la palabra “car” aparece el mismo contexto de la palabra “red” en cinco ocasiones (de un total de N contextos), no coincide en ningún contexto con la palabra “readable” y solo en uno con la palabra “blue”. “book”, por su parte, aparece tres veces en el mismo contexto de “red”, seis

	red	readable	blue
car	5	0	1
book	3	6	0

Cuadro 6.3: Matriz de coocurrencia

en el mismo contexto de “readable” y ninguna con “blue”. En ocasiones estas matrices son cuadradas porque tienen los mismo términos en las filas y en las columnas.

Se pueden plantear otros tipos de matrices. Turner y Pantel [30], por ejemplo, plantean una matriz *Pair-Pattern* donde las filas son parejas de palabras $X : Y$ (“carpenter:wood”) y las columnas son relaciones entre palabras co-ocurrentes (“X cut Y”).

Sea como sea el tipo de matriz, es muy relevante dónde se sitúa el límite del contexto (el documento, el párrafo, la oración, etc.) pues de este límite se podrán obtener representaciones muy precisas en matrices muy dispersas (es decir, con muchas celdas con valor 0), como ocurre con contexto muy pequeños, o representaciones menos precisas pero con matrices (algo más) densas (con menos cantidad de celdas con valor 0). Luego se comentará más sobre el problemas de las matrices dispersas.

Representación de las palabras

Hasta ahora hemos estado hablando de “palabra” o “término”, pero como ya se vio en temas anteriores el concepto de “palabra” es muy vago. Una matriz será más o menos representativa según se defina la palabra. Algunas opciones son (según vimos) pueden ser:

- el *token*,
- la raíz o *stem*,
- el lema,
- el lema más la categoría gramatical,
- los *tokens* pero eliminando *stopwords*,
- solo *tokens* de determinadas categorías gramaticales (solo nombres, o solo verbos, o solo adjetivos, etc.)
- el lema más su dependencia sintáctica,
- u otros casos.

Algunos de estos casos, como imagino ya sabrás, requieren procesar el corpus previamente con técnicas específicas de PLN, como lo que se vio en temas anteriores. Esto es complejo si la colección es muy amplia. En estos casos se utiliza el *token* o el carácter.

Cálculo de los valores o pesos

Finalmente, el modelo semántico vectorial puede ser más o menos representativo según se mida la relevancia (o peso) de la palabra en cada contexto.

El caso más simple para medir la relevancia de una palabra en un contexto es calcular la frecuencia ponderada: número de veces que la palabra aparece en el contexto, normalizado por el tamaño del contexto. Este modelo tiene, sin embargo, diversos problemas:

- Es muy dependiente del tamaño del contexto, que como hemos visto antes no está claro cómo limitarlo. En contexto pequeños se trabajaría con valores muy bajos (ceros y unos prácticamente).
- No discrimina la importancia real de cada palabra en el contexto, dado que hay palabra que siempre tienen frecuencias muy alta (como palabras de categorías gramaticales cerradas -artículos, preposiciones, conjunciones, etc.-, o nombres de uso muy común) frente a otras que siempre tienen frecuencias bajas.
- Sobre estas últimas, el caso extremos es el fenómeno del *hapax legomenon*:¹ las palabras que solo aparecen una vez en todo el corpus. El problema es que suelen ser la mayoría de las palabras: solo aparecen una vez, o con una frecuencia muy baja.

Una solución elegante para determinar la relevancia de una palabra por su frecuencia sin caer en estos problemas es el famoso valor TF/IDF que pasamos a explicar a continuación.

TF/IDF: term frequency / inverse document frequency (Sparck Jones, 1972)

La idea intuitiva que subyace a este valor es que las palabras de uso muy común (aquellas que aparecen con alta frecuencia en prácticamente todos los documentos) no son discriminativas para determinar la importancia del documento. Tienen por tanto poca relevancia en su documento y por tanto su

¹Ver https://en.wikipedia.org/wiki/Hapax_legomenon

valor debe ser bajo. Las palabras que realmente son relevantes en un documento, las que lo caracterizan, son aquellas que tiene una frecuencia relativamente alta en un documento pero, al mismo tiempo, tiene una frecuencia relativamente baja o nula en el resto de documentos. Esto es lo que intenta modelar TF/IDF: dar más peso a las palabras con frecuencia relevante en unos documentos pero no en la totalidad de la colección de textos.

TF/IDF son las siglas de “frecuencia del término por la frecuencia inversa del documento”. Así, en la fórmula nos encontramos con:

- *Term frequency* ($tf(w, d)$): frecuencia relativa de una palabra w en un documento d
- *Document frequency* ($df(t)$): cantidad de documentos donde aparece una determinada palabra w .
- *Inverse document frequency* ($idf(d, D)$): el valor determinante para saber la relevancia del documento es la inversa de la frecuencia de documentos donde aparece. Por tanto, se divide la cantidad toda de documentos N en la colección D entre la frecuencia del documento $df(t)$. Hay varias formas de obtener este valor. La más sencilla es logarítmica, tal que $idf(d, D) = \log \frac{N}{df}$

Así, el valor tf-idf de la palabra w en un documento d en una colección de documentos D es:

$$tfidf(w, d, D) = tf(t, d) \cdot idf(t, D)$$

PPMI: Point Wise Mutual Information (Church y Hanks 1990)

Otra alternativa clásica para medir el peso contextual de una palabra que ha sido muy utilizada en PLN es el *Point Wise Mutual Information* o PPMI. Así como tf-idf es la medida estándar para medir la relevancia de las palabras en matrices término-documento, PMMI es la medida que se suele utilizar en matrices de coocurrencia término-término.

La intuición detrás de PPMI es que la coocurrencia de dos palabras en el mismo contexto es relevante en la medida que podamos saber la posibilidad de que ambas palabras coocurran por casualidad. Si no coocurren en el mismo contexto por casualidad, es que esa coocurrencia es motivada y por tanto relevante.

Así, PMMI mide la probabilidad de que dos palabras aparezcan en el mismo contexto, y lo divide por la probabilidad de aparición de cada palabra por separado:

$$PPMI(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

Hoy día tf-idf y PPMI son medidas clásicas para la representación vectorial del significado. Luego se verán otras propuestas para determinar la relevancia de cada palabra en el contexto donde aparece; pero antes hay que tratar el gran problema de los modelos semánticos vectoriales: la matriz dispersa.

Matriz dispersa y matriz densa

Dada las características de los idiomas, este tipo de matrices de coocurrencias (bien sean término-término o término-documento) que miden las relaciones contextuales entre palabras son siempre matrices muy dispersas (*sparse matrix*), es decir, son matrices en las que la mayoría de los valores son cero (o uno). Lo normal en un idioma es que dos palabras no compartan contexto. Las palabras que comparten contexto entre ellas son pocas, por lo que lo normal es que el valor entre dos palabras sea cero. Esto es un problema tanto desde punto de vista matemático como computacional: se generan estructuras muy grandes pero muy poco informativas.

La solución a este problema es transformar la matriz dispersa en una matriz densa (*dense matrix*), es decir, una matriz sin ceros donde todas las relaciones entre palabras tengan valor superior a 0. Este problema ha sido el principal interés en la investigación en los últimos treinta años. Vamos a comentar tres soluciones que han tenido especial relevancia.

Una primera solución fue *Latent semantic analysis*² o LSA (Landauer y Dumais 1997). Esta aproximación consigue reducir una matriz dispersa en una matriz densa de 300 dimensiones mediante su descomposición en valores singulares (*singular value decomposition*).³ Lo interesante de la matriz resultante no es solo que sea una matriz densa; sino que esa matriz densa, además de mantener las relaciones contextuales entre palabras, muestra relaciones semánticas “latentes”: relaciones semánticas entre palabras que a simple vista no se detectan. LSA, así, supuso un avance en semántica vectorial en las tres áreas de conocimiento implicadas: matemática, computación y lingüística.

Años más tarde se propuso *Latent Dirichlet Allocation*⁴ o LDA, mediante el cual se pueden detectar temas o *topic* en amplios corpus de manera no supervisada.

Finalmente, la búsqueda de matrices densas y la optimización de la representación contextual mediante vectores ha llevado a los *skip grams*, que es la

²https://en.wikipedia.org/wiki/Latent_semantic_analysis

³https://en.wikipedia.org/wiki/Singular_value_decomposition

⁴https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

base de *Word2Vec* y de donde derivan los *word embeddings* y los modelos neuronales actuales. Para determinar la relevancia contextual entre dos palabras, la idea principal de los *skip gramms* es entrenar un clasificador con regresión logística que aprenda si una palabra formar parte o no del contexto de otra palabra. Como solo aprende si dos palabras comparten o no contexto, ese entrenamiento no necesita un corpus anotado a mano: basta con una amplia colección de documentos (cuanto más grande mejor). Y al final lo de menos es el clasificador: lo importante son los pesos que ha aprendido para cada palabra. Ese es su vector contextual o *word embedding*, que ha demostrado tener gran capacidad de representación semántica. Esto es el inicio del *deep learning* y del PLN moderno basado en redes neuronales, que se verá en próximos temas.

6.5. Interpretación semántica: distancia y similitud.

El vector de una palabra en sí mismo no tiene un significado como podría tenerlo, por ejemplo, una definición. Es una representación semántica más de tipo conexional que simbólico. Decir que el significado de la palabra “casa” es un vector tal que $\{1, 4, 0, 0, 1\}$ es como no decir nada: ese vector no tiene relación con un concepto cognitivo, sólo indica la relevancia de la palabra en cada contexto (el uso de la palabra).

¿Cómo se realiza, entonces, la interpretación de una palabra y oración en el modelo semántico vectorial? La interpretación en esta aproximación vectorial a la semántica distribucional se realiza por relaciones de *similitud* entre palabras, oraciones, fragmentos o documentos. Dos palabras con vectores contextuales similares implica que ambas palabras tienden a aparecer en los mismos contextos, y por tanto su significado está relacionado. Dos palabras cuyos vectores contextual sean muy diferentes implica que son palabras con significado dispar. Cualquier aplicación de semántica vectorial debe pensarse en términos de similitud entre palabras, grupos de palabras, textos, etc. y no tanto como una interpretación sustancial.

La similitud se calcula según la distancia entre los vectores en el espacio vectorial: a menor distancia entre vectores, mayor similitud semántica. Si bien hay diferentes medidas para calcular la distancia entre vectores, la más utilizada es la distancia del coseno, que mide el ángulo entre dos vectores ambos con origen en $0, 0$:

$$\cos(a, b) = \frac{ab}{||a|| ||b||}$$

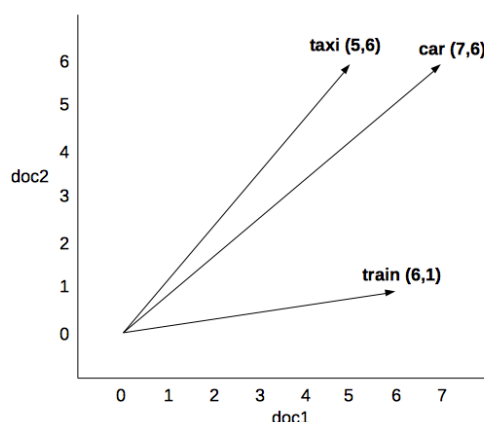


Figura 6.3: Representación vectorial del significado léxico

6.6. Conclusiones

En este capítulo se ha visto la idea general de la semántica vectorial, según la cual el significado de una palabra depende de su uso y de los contextos donde se usa. Se ha visto cómo se representa formalmente ese significado distribucional mediante vectores, cuyos valores miden el peso de la palabra en cada uno de los contextos (bien sean textos, oraciones, etc.). Finalmente se ha comentado brevemente el proceso de interpretación distribucional, que se basa en determinar la similitud entre vectores a partir de su distancia en el espacio multidimensional.

6.7. Situación actual

De aquí derivan los *word embeddings* que, junto con las redes neuronales, forman el PLN moderno. De todo esto se hablará en las próximas sesiones.

6.8. Herramientas y recursos

Para crear espacios vectoriales y calcular similitudes (sin entrar por ahora en redes neuronales):

- GENSIM⁵

⁵<https://radimrehurek.com/gensim/>

- NLTK⁶
- Pattern⁷
- SpaCy⁸

⁶<http://www.nltk.org/>

⁷<http://www.clips.ua.ac.be/pattern>

⁸<https://spacy.io/>

Temas abiertos en PLN

7.1. Introducción

Si bien los modelos neuronales han conseguido logros bastante impresionantes en el procesamiento automático del texto, las lenguas naturales son bastante más complejas de lo que la tecnología actual puede procesar. Hay, así, diferentes cuestiones que aún no tienen una respuesta válida por parte del PLN. En esta sección se exponen algunas de ellas

7.2. Más allá de la alquimia

Una de las grandes críticas a los modelos neuronales es su falta de transparencia. Son cajas negras: resultan muy complejo explicar qué hacen, cómo procesan la información. En tareas concretas (*end-to-end*) funcionan bien, pero no se sabe cómo, ni cuándo fallarán. Esto los convierte en una especie de alquimia computacional, alejada de la claridad y “explicabilidad” necesaria en la vertiente más científica de la inteligencia artificial.¹

Este concepto de “alquimia” lo plantean Church y Liberman en este artículo [8], y proponen líneas de trabajo futuras:

K. Church and M. Liberman (2021) “The Future of Computational Linguistics: On Beyond Alchemy”, *Front. Artif. Intell.*, vol. 4, 2021 <https://www.frontiersin.org/articles/10.3389/frai.2021.625341/full>.

¹Me refiero aquí con “científica” a que, más que soluciones a problemas, buscar explicaciones a fenómenos de la naturaleza (humana en este caso). No tiene ningún tipo de connotación ni positiva ni negativa.

7.3. Composicionalidad

Como se vio en temas anteriores, uno de los grandes principios de la aproximación simbólica a la interpretación semántica de un texto es el principio de composicionalidad. Este principio, sin embargo, solo es aplicable a los modelos simbólicos (de base lógica). No casa bien con la representación geométrica (vectorial, distribucional) del significado en la que se basan los modelos neuronales modernos. Kornai [18] considera ésta como “non-compositional semantics”.

El problema se puede formular del siguiente modo. Un *word embedding* representa el significado de una palabra mediante un vector. Una oración estará formada por tantos vectores como palabras tenga. El principio de composicionalidad dice que el significado de un texto está en función del significado de las palabras y sus relaciones. ¿Cómo se podría representar el significado de la oración con un único vector? ¿Suma ponderada de vectores? ¿Y cómo se representaría con vectores unidades textuales?

Modelos como SentenceBERT² toman esta idea de suma de vectores para representar con un solo vector el significado oracional a partir de *word embeddings*. Una especie de *sentence embedding*. Ver <https://arxiv.org/abs/1908.10084>

En 2023 se ha presentado un modelo (en realidad es un meta-modelo) capaz de hacer generalizaciones vía composicionalidad [19]. Lectura ésta necesaria para profundizar en este tema:

<https://www.nature.com/articles/s41586-023-06668-3>

Otra alternativa es la representación composicional propuesta por los modelos cuánticos. Ver [10]

7.4. Metáforas

La metáfora no es (solo) un fenómeno literario, sino un mecanismo cognitivo para nombrar la realidad [20]. Utilizamos metáforas, entre otras cosas, para tratar con conceptos muy abstractos. Siempre que hablamos del tiempo (como entidad abstracta), por ejemplo, lo hacemos con términos propios del campo del dinero (que es algo concreto), con expresiones como “ahorrar tiempo”, “perder el tiempo”, o la expresión “el tiempo es oro”. Este tipo de metáfora, denominada metáfora cognitiva, es ubicua, está presente en cualquier tipo de texto.

La metáfora, por definición, se utiliza un término en un contexto que no le corresponde. Se produce un conflicto entre el significado del término de manera aislada y el significado que asume en el contexto donde aparece. Es

²<https://www.sbert.net/>

así un fenómeno semántico muy difícil de tratar tanto con los modelos distribucionales como con los modelos simbólicos clásicos.

Sobre los problemas actuales para procesar metáforas, véase [15, 24]

7.5. Neural-Symbolic AI

Como no podía ser de otra manera, hay propuestas híbridas para combinar los modelos simbólicos y los modelos conexistas. Una de ellas es la llamada Neural-Symbolic AI en [4]. Según este trabajo:

“The goals of neural-symbolic computation are to provide a coherent, unifying view for logic and connectionism, to contribute to the modelling and understanding of cognition and, thereby, behaviour, and to produce better computational tools for integrated machine learning and reasoning. To this end, logic and network models are studied together as integrated models of computation. Typically, translation algorithms from a symbolic to a connectionist representation and vice-versa are employed to provide either (i) a neural implementation of a logic, (ii) a logical characterisation of a neural system, or (iii) a hybrid learning system that brings together features from connectionism and symbolic artificial intelligence.”

Ver <https://arxiv.org/abs/1711.03902>

7.6. Texto narrativos

Con el texto narrativo pasa algo similar a lo que se comentó sobre las metáforas: es un tipo de texto muy común (por ejemplo, la noticias periodísticas suelen tener estructura narrativa) y especialmente complejo de procesar. Entender un texto narrativo implica detectar los eventos expresados y las relaciones tanto temporales como causales entre ellos.

Desde el punto de vista de la generación automática del lenguaje, es también un tipo de texto difícil. Los grandes modelos de lenguaje actuales como GPT pueden generar texto narrativo, pero éste presenta varios problemas. No son capaces, por ejemplo, de establecer relaciones causales entre eventos, no hacen un plan narrativo general, no hay suspense ni sorpresa (suelen ser narraciones aburridas por evidentes), etc.

El tratamiento del texto narrativo, tanto en su interpretación como, sobre todo, en su generación, es un tema aún por explorar a fondo. Una presentación de los principales problemas puede verse en este reciente trabajo de A. Pipper: [27]

Bibliografía

- [1] E. Agirre and A. Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL)*. Association for Computational Linguistics, 2009.
- [2] John L. Austin. *Cómo hacer cosas con palabras: palabras y acciones*. Paidós, Barcelona, 2016 (1962).
- [3] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. Association for Computational Linguistics, 1998.
- [4] Tarek R. Besold, Artur d’Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation, 2017.
- [5] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 2019.
- [6] Noam Chomsky. *Aspects of the theory of syntax*. MIT Press, Cambridge, 1965.
- [7] Noam Chomsky. *The Minimalist Program*. MIT Press, Cambridge, 1995.
- [8] K. Church and M. Liberman. The Future of Computational Linguistics: On Beyond Alchemy. *Frontiers in Artificial Intelligence*, (4), 2021.

- [9] Daoud Clarke. A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics*, 38(1):41—71, 2012.
- [10] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark†. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis*, (36), 2010.
- [11] Jacob Eisestein. *Introduction to Natural Language Processing*. MIT Press, Cambridge, 2019.
- [12] C. Fellbau. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, 1998.
- [13] John R. Firth. *Papers in Linguistics. 1934-1951*. Oxford University Press, 1957.
- [14] J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:2–71, 1988.
- [15] Mengshi Ge, Rui Mao, and Erik Cambria. A survey on computational metaphor processing techniques: from identification, interpretation, generation to application. *Artificial Intelligence Review*, 56:1829—1895, 2023.
- [16] Zellig Harris. *Structural Linguistics*. University of Chicago Press, Chicago, 1951.
- [17] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 2023.
- [18] András Kornai. *Vector semantics*. Springer, 2023.
- [19] B. M. Lake and M. Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623:115—121, 2023.
- [20] George Lakoff and Mark Johnson. *Metáforas de la vida cotidiana*. Cátedra, 2017 (1980).
- [21] Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni De Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. Quantum natural language processing on near-term quantum computers, 2020.
- [22] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–40, 1995.

- [23] Borja Navarro Colorado. Sistemas de anotación semántica para corpus de español. In Giovanni Parodi, Pascual Cantos, and Lewis Howe, editors, *The Routledge Handbook of Spanish Corpus Linguistics*. Routledge, 2021.
- [24] Arthur Neidlein, Philipp Wiesenbach, and Katja Markert. An analysis of language models for metaphor recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2020.
- [25] L. Padró and E. Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association, 2012.
- [26] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1), 2005.
- [27] Andrew Piper. Computational narrative understanding: A big picture analysis. In Yanai Elazar, Allyson Ettinger, Nora Kassner, Sebastian Ruder, and Noah A. Smith, editors, *Proceedings of the Big Picture Workshop*, pages 28–39. Association for Computational Linguistics, 2023.
- [28] John R. Searle. *Actos de habla*. Cátedra, Madrid, 2017 (1965).
- [29] P. Smolensky. The constituent structure of connectionist mental states: A reply to fodor and pylyshyn. In T. Horgan and J. Tienson, editors, *Connectionism and the Philosophy of Mind*. Springer, 1991.
- [30] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010.
- [31] Dominic Widdows. *Geometry and Meaning*. CSLI Pub., 2004.
- [32] Ludwig Wittgenstein. *Investigaciones filosóficas*. Crítica, 2008 (1953).