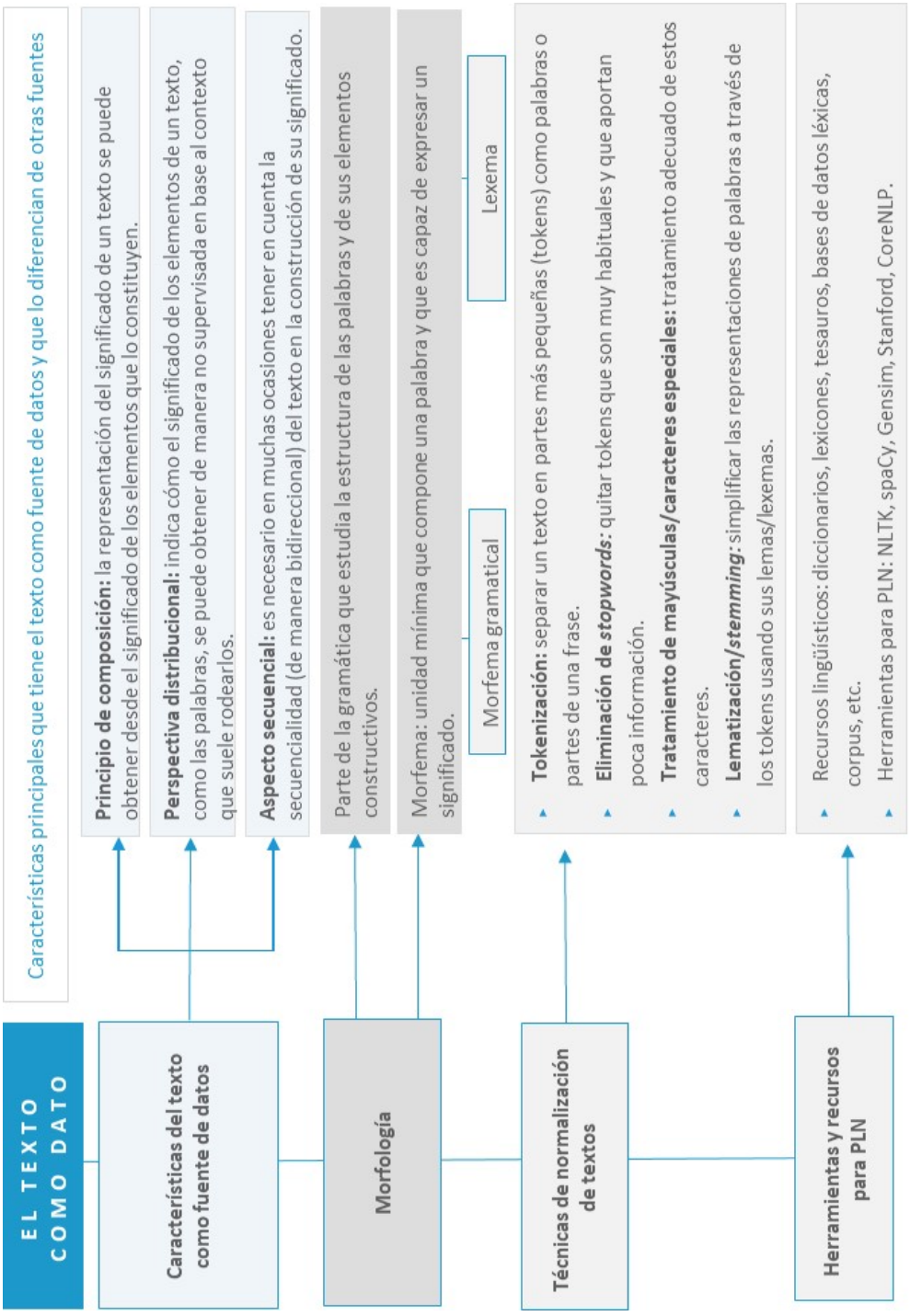
Procesamiento del Lenguaje Natural

Tema 2. El texto como dato

Índice

Esquema



2.1. Introducción y objetivos

A continuación, se presentarán las características principales que tiene el texto como fuente de datos, y qué lo diferencia de otras fuentes. Por otro lado, se define el concepto de morfología y qué aspectos incluye. Esto servirá como base para explicar distintas técnicas de normalización de textos, muy usadas en aplicaciones de PLN, y cómo se relacionan dentro de lo que se conoce como flujo (*pipeline*) de normalización. Finalmente, se explicarán los distintos recursos lingüísticos, se darán referencias a distintos corpus en español que se pueden usar para tareas de PLN, junto con referencias a algunas de las herramientas de software más usadas hoy en día para ello.

Objetivos

- Entender las características concretas que tienen los textos como fuentes de datos, y qué aspectos hay que tener en cuenta para poder trabajar con ellos.
- Describir los conceptos fundamentales de la morfología en español.
- Conocer algunas de las técnicas más habituales de normalización de textos, y cómo poder incluirlas en un flujo de normalización.
- Describir los diferentes tipos de recursos lingüísticos necesarios para implementar tareas de procesamiento del lenguaje natural.
- Conocer referencias a corpus en español, así como a herramientas para para poder trabajar en PLN

2.2. Características del texto como fuente de datos

Actualmente, en la era del Big Data, se generan grandes volúmenes de datos muy **heterogéneos** (imágenes, textos, datos de dispositivos IoT, etc.). Por este motivo, para poder trabajar adecuadamente con una fuente específica de datos, es necesario conocer bien **qué la caracteriza.** Esto ocurre con los textos en el ámbito del PLN. Los textos son una fuente más de datos disponible que contiene, como ya se vio en el tema previo, información muy relevante para **distintos casos de uso.**

Una de las primeras características asociadas al texto como dato es que el lenguaje es **composicional** y se puede tratar de manera discreta. Esto hace referencia a que el significado general de un texto (ej., una frase) se puede componer desde el significado de sus elementos (ej., palabras). Por ejemplo, la frase:

- He aprobado el examen.

El significado general de esta frase se constituye en base a las palabras individuales que lo componen, donde «He aprobado» hace referencia al sujeto (el que ha aprobado), mientras que «el examen» hace referencia al predicado. Combinando estos dos elementos se construye el significado de la frase original.

Análogamente, se pueden combinar textos para componer el significado de un texto más largo. Por ejemplo, la frase:

- He aprobado el examen tras mucho estudio.

Parte del significado de dicha frase se reúne en «He aprobado el examen», como hemos visto antes. Para obtener el significado de la frase completa, se construye con base en sus constituyentes que son «He aprobado el examen» junto con «tras mucho estudio», de manera que el significado de la primera parte de la oración se combina con el de la segunda parte. El principio de composición se puede llevar a

textos más amplios, como frases completas o incluso a nivel de documentos completos.

El principio de composición indica que la representación del significado de un texto se puede obtener desde el significado de los elementos que lo constituyen.

Este principio también aparece a nivel de palabras. Por ejemplo, la palabra *médicos* se compone del elemento raíz *médico* combinado con la letra *s* para especificar el significado final de la palabra, que en este caso es el significado de esa palabra raíz, pero en plural. El aspecto composicional a nivel de palabras está relacionado con el ámbito de la **morfología**, que se estudiará en el punto siguiente.

La aproximación de utilizar el principio de composición para construir el significado de un texto con base en sus componentes no es suficiente en algunas ocasiones. Esto ocurre en casos como cuando hay **ambigüedad léxica** (donde una palabra puede tener varios significados) o cuando se tienen **frases hechas** (donde el significado de la frase no se interpreta directamente desde sus palabras aisladas). Por ejemplo, la frase:

- Me he sentado en el banco del parque a ver el lago.

En ella aparece la palabra *banco*, que puede referirse tanto a una entidad financiera, como a un conjunto de peces o a un asiento. En este caso, el sentido es el de asiento y la forma por la que esto se sabe es por las palabras que aparecen en el contexto de la frase.

Esto está relacionado con la perspectiva distribucional de los textos, con la que se puede construir el significado de los elementos de un texto en base a las palabras que suelen aparecer en su contexto en otros textos.

Esta aproximación es no supervisada, ya que no es necesario que un anotador haga un trabajo previo para que se pueda deducir el significado de una palabra concreta. Sólo es necesario otros textos para que se pueda deducir ese significado gracias a esta **aproximación distribucional.**

A modo de ejemplo, textos como los siguientes servirían para generar el contexto necesario para esta perspectiva de la palabra *banco*:

- … fue al banco a contratar una hipoteca.
- … en el mar pude ver un banco de peces rojos.
- … ellos se sentaron en un banco de madera.

Esto no sirve para explicar por qué esas palabras aparecen en un contexto determinado de la palabra *banco*, pero sí para identificar el significado de esta palabra en un texto concreto.

La perspectiva distribucional de los textos indica cómo el significado de sus elementos, como las palabras, se puede obtener de manera no supervisada con base al contexto que suele rodearlos.

Ahora bien, aunque los textos se pueden discretizar y analizar desde el punto de vista del principio de composición, complementando el análisis con la perspectiva distribucional, se deben tener en cuenta aspectos como la creación de palabras nuevas en una lengua (ej., neologismos), o que algunas palabras son mucho más frecuentes en textos que otras (por ejemplo, las palabras que son determinantes o preposiciones frente a verbos y sustantivos) (Zipf, 1949). Como consecuencia de esto, los algoritmos de PLN deben tener en cuenta estos aspectos para poder trabajar adecuadamente con los textos.

Ideas clave

Además de la perspectiva distribucional para poder construir el significado de los elementos de un texto de manera no supervisada, existen otras maneras de ver las **relaciones** entre las palabras de manera supervisada. Este es el caso de las ontologías semánticas como WORDNET (Fellbaum, 2010), donde se recogen las relaciones que existen entre palabras y otras unidades de texto. Por ejemplo, WORDNET serviría para saber que una *rueda* es una parte de un *coche*.

Los textos tienen otra característica que los diferencian de otros conjuntos de datos: e l **aspecto secuencial**. En muchas ocasiones, el principio de composición y la perspectiva distribucional no son suficientes para construir el significado de una frase, como ocurre en el caso siguiente:

- Quiero información de hipotecas, pero no seguros.
- Quiero información de seguros, pero no hipotecas.

En ambas frases las palabras son las mismas, pero el significado es distinto. Tratar de construir el significado de la frase desde las palabras individuales con lo visto hasta ahora llevaría en ambos casos al mismo resultado final. Es por esto por lo que en PLN es necesario también atender al **aspecto secuencial de los textos,** de manera que viendo las palabras previas (ej., el «*no*» delante de «*seguros*» o «*hipotecas*») se puede construir mejor el significado global desde los elementos individuales. Ocurre también que en otras ocasiones se debe tener en cuenta no sólo las palabras previas, sino las posteriores, como en:

- Al campo no he ido, he ido a la playa.
- A la playa no he ido, he ido al campo.

En estos casos la información para saber que no ha ido al «*campo*» o a la «*playa*» aparece después de estas palabras, no antes, con lo que el aspecto secuencial de los textos es bidireccional.

El aspecto secuencial de los textos pone de manifiesto cómo es necesario en muchas ocasiones tener en cuenta la secuencialidad (de manera bidireccional) del texto en la construcción de su significado.

En algunos casos la construcción del significado puede ser aún más compleja, como en el caso siguiente:

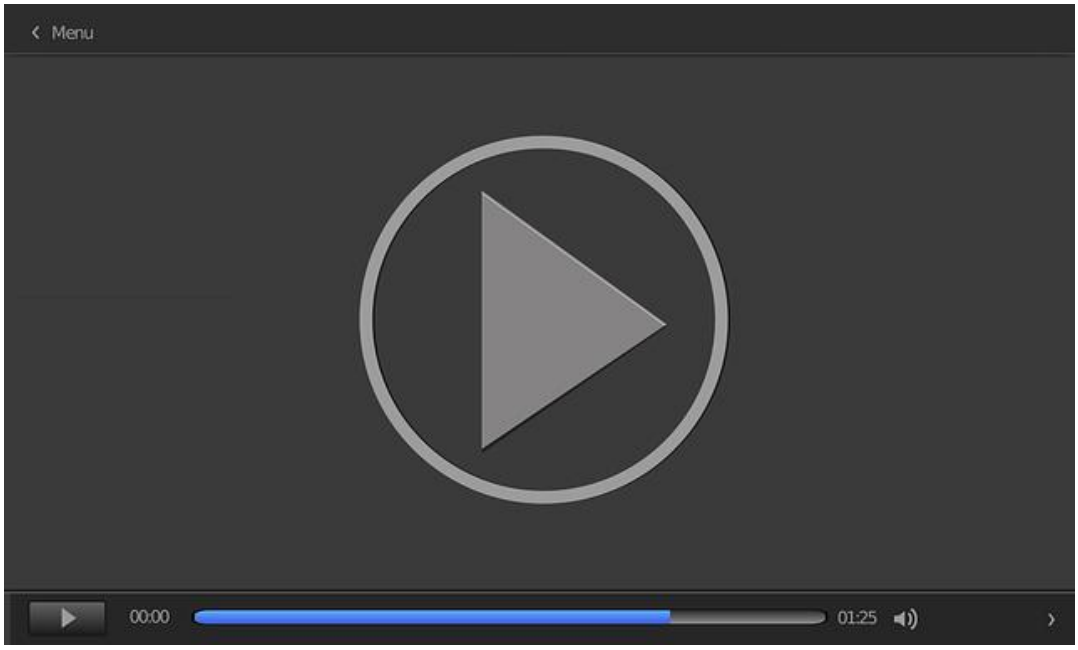
- Pedro ve un cuadro de su madre.

Esta frase se podría interpretar de dos maneras: el cuadro pertenece a la madre, o en el cuadro aparece la madre. La construcción del significado en estos casos debería tener en cuenta, además del aspecto composicional, distribucional y secuencial, la **relación sintáctica** entre las palabras dentro de la oración.

Aquí, por tanto, se ve otro tipo de ambigüedad que se puede encontrar en los textos: la ambigüedad sintáctica, donde una misma frase puede tener distintas interpretaciones.

Ideas clave

En el vídeo *El texto como dato* se presentará qué caracteriza a los textos como dato y cómo poder trabajar con ellos a nivel computacional.



02.01. El texto como dato

Accede al vídeo:

https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=1676efde-dc53-4b37-89ba-af5b0149cd3a

2.3. Morfología

La Real Academia Española (RAE) en su diccionario de la lengua española define la palabra **morfología:** «De *morfo-* y *-logía*. 2. f. Gram. Parte de la gramática que estudia la estructura de las palabras y de sus elementos constitutivos».

Por tanto, la morfología estudia la forma en que las palabras se descomponen en partes indivisibles, las cuales tienen un significado.

A estas unidades mínimas se les llama **morfemas** y, por ejemplo, la palabra «mujeres» contiene dos morfemas: el primero es *mujer* y el segundo es *es.*

Un morfema es la unidad mínima que compone una palabra y que es capaz de expresar un significado.

De hecho, no se debe confundir esta definición de morfema con la de morfema gramatical, al que en algunos textos se le llama de la misma manera. Se entiende como **morfema gramatical** al que tiene un significado gramatical, es decir, significado sobre:

- El género (masculino o femenino).
- Número (singular o plural).
- Persona (p. ej. tercera persona del singular).
- Modo y tiempo (p. ej. modo indicativo y tiempo futuro).

Ideas clave

En contraposición al morfema gramatical está el **lexema.**

Un lexema sería el morfema que conforma la raíz o parte invariante de la palabra y que es la mínima unidad con significado léxico, por lo que proporciona el significado principal de la palabra.

Para el ejemplo de la palabra «mujeres», el morfema *mujer* sería el lexema o raíz de la palabra con significado léxico y el morfema *es* sería el morfema gramatical que añade significado gramatical y expresa el número plural. El singular, que también sería en este caso «mujer», contiene a su vez dos morfemas: el lexema *mujer* y un morfema cero de número singular.

Un morfema cero es un morfema gramatical sin realización fonética, pero que tiene significado gramatical.

Una palabra como «cantábamos» contiene tres morfemas:

- El primero es el lexema *cant,* que indica el significado léxico de la palabra, es decir, el acto de que una persona produzca con la voz sonidos melodiosos.
- El segundo es el morfema *aba,* que proporciona el significado gramatical de modo indicativo y tiempo pasado.
- El tercero es el morfema *mos,* que indica el significado gramatical de primera persona del plural.

En el procesamiento del lenguaje natural, el ámbito de la morfología computacional trata de reconocer de forma automática los morfemas que contiene una palabra.

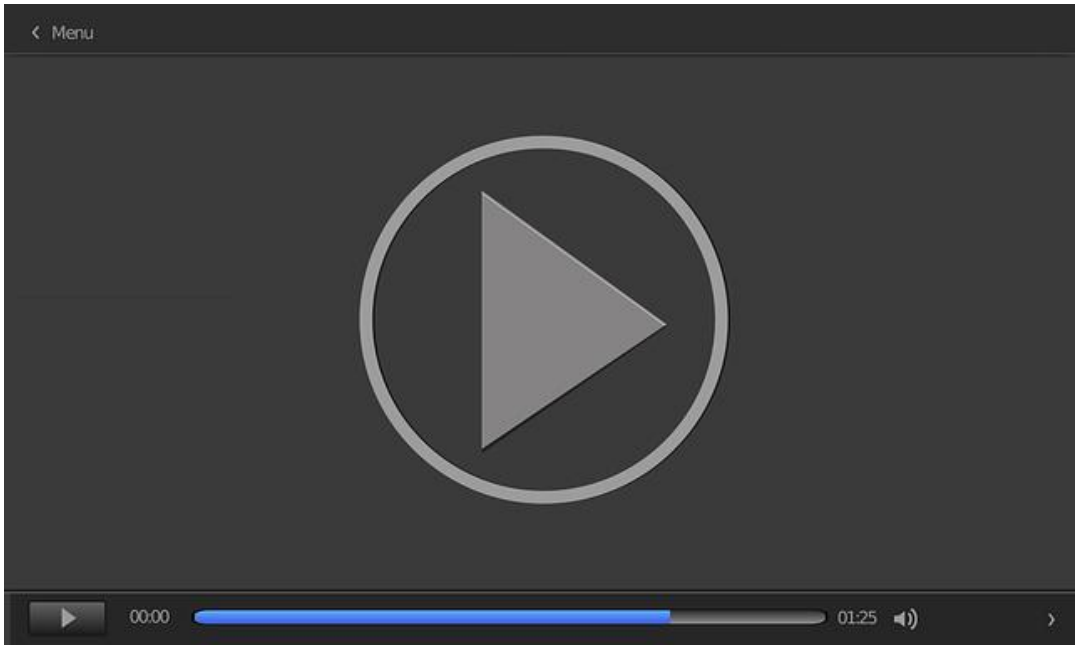
Conocer la morfología de las palabras es importante para, si se está buscando en un texto el verbo «pensar», que se reconozca también la fórmula *piénsalo,* aunque

ambas no se compongan de la misma cadena de caracteres en la raíz. Además, a la hora de generar de forma automática frases o expresiones de texto se debe conocer la morfología de las palabras a utilizar para encajar, por ejemplo, el género y el número de un nombre con el adjetivo que la acompaña o la persona del verbo con el sujeto de la frase.

El análisis morfológico en inglés, lengua en la que se realiza la mayoría de la investigación en el ámbito de la morfología computacional, es más simple que el análisis morfológico en español. Esto se debe principalmente a que en el español se dan muchas variedades diferentes de las formas en las terminaciones de las palabras y también se dan más alteraciones en la raíz de estas.

Ideas clave

En el vídeo *Morfología* se verá qué es y las bases para entenderlo a nivel lingüístico y podamos aplicarlo en el procesamiento del lenguaje natural a nivel computacional.



02.02. Morfología

Accede al vídeo:

https://unir.cloud.panopto.eu/Panopto/Pages/Embed.aspx?id=c1cb53b4-3d2e-41e2-865e-ac6b00c3def7

2.4. Técnicas de normalización de textos

El principio de composición visto antes, aunque no siempre es suficiente para construir el significado de determinadas frases, es una aproximación suficiente para cubrir muchos tipos de textos y casos de uso. Dado que este sirve para obtener la construcción del significado de un texto desde sus componentes, sirve también de cara a transformar un texto en distintas **variables de entrada** para poder entrenar un **modelo de aprendizaje automático.**

Anteriormente, se vio como un modelo de aprendizaje automático puede ser usado en tareas como el análisis del sentimiento de un texto y saber si este es positivo, negativo o neutro. Para poder llevar a cabo esta tarea con, por ejemplo, un modelo supervisado, se necesitaría tener un conjunto de textos anotados con esas tres categorías de sentimientos, y, como entrada, el modelo recibiría una serie de variables extraídas desde el texto siguiendo el principio de composición (por ejemplo, qué palabras aparecen en él).

> La transformación de un texto en los elementos que lo componen para poder trabajar con ellos posteriormente es una de las tareas habituales de PLN, y sigue un flujo (pipeline) de normalización en distintos pasos.

Primer paso

Se denomina **tokenización**. Consiste en partir de una cadena de texto (como una frase) para descomponerla en sus términos o componentes (ej., las palabras que la forman). Por ejemplo, para la siguiente frase:

- Después de estar estudiando 2 horas, he decidido estudiar 2 horas más.

Su descomposición en *tokens* sería la siguiente:

«Después», «de», «estar», «estudiando», «2», «horas», «he», «decidido», «estudiar», «2», «horas», «más».

Este proceso de tokenización ha seguido en este caso una de las aproximaciones más habituales, en las que se separa el texto en palabras en base a los **espacios en blanco** que aparecen en la secuencia de texto. Previamente a esto, se han eliminado signos de puntuación, como la coma o el punto final. Así, como resultado, tenemos una lista ordenada con las palabras que aparecen en el texto.

No obstante, si se analiza la lista final tras el proceso de tokenización, se pueden ver algunos aspectos a considerar. Por ejemplo, hay palabras que aparecen varias veces, como «*horas*». También, ocurre que hay palabras con mayúsculas, como «*Después*» por ser la primera palabra de la frase. También se ve que no todo son palabras, sino que aparecen números, como el *2*». Adicionalmente, se ve que algunas palabras similares aparecen expresadas de manera distinta, como es el caso de «*estudiando*» y «*estudiar*» por sus conjugaciones verbales. Estos aspectos también se tienen en cuenta en el proceso del flujo de normalización.

El proceso de tokenización busca separar un texto en partes más pequeñas (tokens) como palabras o partes de una frase.

Ideas clave

Un primer aspecto para comentar es que el proceso de tokenización que se ha seguido de separar palabras tras eliminar signos de puntuación, no es correcto para todos los casos. Por ejemplo, si se tienen palabras en inglés como *Dr.*, o'clock, *doctor's* (el apóstrofe *'s* en inglés), o formatos de fechas como *2022-06-01* o de horas como *08:30*, no sería correcto eliminar sin más estos signos y luego separar por espacios en blanco. Es necesario por tanto en el flujo de normalización diferenciar entre comas y puntos respecto a otro tipo de signos. Esto se puede realizar de distintas maneras, como por ejemplo mediante el uso de reglas o heurísticas, o usando bases de datos que sirvan para estandarizar algunos de estos términos.

Otra de las limitaciones en la aproximación de la tokenización que se ha visto es **cómo tratar los nombres propios.** Por ejemplo, si en un texto aparece *Nueva York*, no tendría demasiado sentido separarlo en dos tokens distintos, uno para *Nueva* y otro para *York*. Esto también ocurre con palabras compuestas, como *Estado del Arte* o *a priori*. En muchos casos se plantea en estos escenarios hacer una **tokenización en dos niveles**, de manera que primero se lleve a cabo una tokenización a bajo nivel, seguida de una a alto nivel que sirva para generar tokens que tengan más sentido desde un punto de vista semántico. Con ello, *Nueva York* pasaría a ser un único token expresado como *Nueva York*. Esto se puede llevar a cabo de distintas maneras, como por ejemplo mediante aproximaciones estadísticas que tienen en cuenta palabras que suelen ser coocurrentes.

Desde la perspectiva del principio de composición, según la tarea para la que se quiera usar el texto, se puede ver que no todos los tokens son igual de importantes. Si, por ejemplo, queremos identificar el sentimiento positivo, negativo o neutro de la frase

- Estoy contento de haber comprado este libro.

Las palabras como «de» y «este» no aportan mucha información para la tarea que se quiere llevar a cabo. Por este motivo, se presenta el siguiente paso.

Segundo paso

Es la **eliminación de *stopwords***, lo que suele incluir palabras como artículos, conjunciones o preposiciones, e incluso pronombres en algunas ocasiones. Una manera de **detectar estas palabras** es mediante el **uso de diccionarios específicos** para cada lengua, que contengan la lista de palabras a eliminar. La eliminación de *stopwords* es una técnica que no siempre se va a emplear, sino que **dependerá de cada caso de uso concreto.** Para algunas tareas de PLN puede ser necesaria la información de esas palabras.

La eliminación de *stopwords* se usa para quitar tokens que son muy habituales y que aportan poca información, como es el caso de artículos, conjunciones o preposiciones cuando esos tokens representan palabras.

Tercer paso

Es el **tratamiento de las mayúsculas**. En una frase como:

- Días y días pasaron sin noticias nuevas.

Aparece la palabra «*días*» tanto en mayúscula como en minúscula. Si no se tratan las mayúsculas, «*Días* y *días*» serían dos tokens distintos a todos los efectos. Para estos casos, se puede llevar a cabo un paso de **normalización que elimine las mayúsculas de la frase,** de manera que ambas palabras queden representadas por el mismo token. El tratamiento de las mayúsculas no siempre pasa por eliminarlas. Por ejemplo, en la frase:

- Su amiga Estrella estaba de vacaciones, y vio una estrella en el cielo.

En este caso, «Estrella» hace referencia a un nombre de persona (nombre propio), mientras que «estrella» hace referencia a un nombre común. En este caso, si se

quitase la mayúscula de «Estrella», se representarían ambas palabras con el mismo token, lo que no sería correcto. El objetivo es por tanto eliminar las mayúsculas, pero **solo cuando sea necesario.** Esto se puede hacer mediante el uso de reglas (por ejemplo, eliminar las mayúsculas sólo de la primera palabra de la frase), o mediante soluciones más sofisticadas, como los algoritmos de **reconocimiento de entidades nombradas** (***NER*, Named Entity Recognition)** que identificarían que tokens hacen referencia a personas y organizaciones de cara a dejar las mayúsculas en esos casos.

El tratamiento de las mayúsculas busca tratar adecuadamente las mayúsculas en los tokens, eliminándolas cuando tanto un token con mayúsculas como otro con minúsculas se refieren a la misma palabra.

Existen otro tipo de tratamientos que se pueden considerar para homogeneizar las representaciones de los tokens. Por ejemplo, en el primer texto se veía que «dos» se representaba por su número, «2». Puede ocurrir que en un documento coexistan ambas representaciones, de manera que sería necesario representar ambas de la misma manera.

Cuarto paso

También ocurre en algunos tipos de textos (por ejemplo, *tweets*), que aparecen caracteres no alfanuméricos (*hashtags*, *@*, etc.), de manera que en el flujo de normalización también se debe tener en cuenta qué hacer con estos caracteres a la hora de representar un texto por sus tokens normalizados. En algunas ocasiones puede servir eliminarlos (por ejemplo, eliminar algunos signos de exclamación que tenga un texto), pero dependerá de cada caso de uso y de si esto supone una pérdida relevante de información o no. Este paso se suele denominar **tratamiento de caracteres especiales**.

El orden de los pasos del proceso de normalización no siempre tiene porque ser el mismo. El proceso de tokenización es el primer paso, pero, según el caso concreto, el tratamiento de mayúsculas y de caracteres especiales puede ir antes de la eliminación de *stopwords*.

Por ejemplo, «Principado de Asturias» se detectaría como una entidad nombrada de un lugar concreto, y se podría expresar con un token como «Principado» «de» «Asturias» antes de eliminar la *stopword* «de».

Tras estos pasos, **una fase que incluye el proceso de normalización** para algunos casos de uso es la **obtención del lema o de la raíz** de las palabras. Para un texto como:

- Ayer jugaron al mismo deporte que han jugado hoy.

Se tienen dos verbos, «jugaron» y «jugado» en dos conjugaciones verbales distintas. Con la obtención del lema o de la raíz de estas palabras se busca representar ambas con un mismo token. Para el caso de la obtención del lema se lleva a cabo el proceso de **lematización**, con el que estas palabras quedan representadas por su lema, que sería *jugar*. Para esto, se pueden usar diccionarios donde se tengan las equivalencias entre distintas palabras y lemas, entre otras técnicas.

El proceso de obtención de la raíz, conocido como ***stemming,*** es distinto. Con él se busca **obtener el lexema eliminando el resto de los morfemas.** Por ejemplo, para estas dos mismas palabras sería *jug*.

Existen distintos algoritmos para llevar a cabo esta tarea, como por ejemplo el algoritmo Snowball (Porter, 2001).

Ideas clave

Uno de los problemas de la aproximación de *stemming* frente a la de lematizar es que hay ocasiones en las que dos palabras que deberían tener una misma representación no la tienen.

Por ejemplo, para casos como «jugaron» y «yo juego», ambas palabras se representarían como *jugar* en el caso de la lematización, pero se representarían como *jug* y *jueg* respectivamente con el *stemming*. Otro aspecto importante que remarcar es que no en todos los casos de uso es aconsejable llevar a cabo un proceso de lematización o *stemming*, ya que con este se puede perder información relevante de un texto. Para un texto como:

- El número de ingenieras egresadas ha aumentado en los últimos años.

Si se llevase a cabo un proceso de lematización, «ingenieras» y «egresadas» pasaría a ser «ingeniero» y «egresado», de manera que se estaría perdiendo una parte importante de la información que está expresando el texto.

En los procesos de lematización y de stemming se busca simplificar las representaciones de las palabras a través de los tokens usando sus lemas o sus lexemas respectivamente.

Como aspecto final, estos flujos de normalización son **útiles** para llevar a cabo la fase de **preprocesado de muchos textos** para luego utilizarlos en distintas tareas de PLN, principalmente cuando la fuente de origen es un texto en crudo. Existen otros formatos de datos que incluyen, junto con los textos, información relevante con la que no sería necesario en ocasiones llevar a cabo algunas de estas fases de normalización, como ocurre en muchos casos donde el texto está en **formato XML** (Extensible Markup Language).

2.5. Recursos lingüísticos

La implementación de diferentes tareas de procesamiento del lenguaje natural requiere de una serie de recursos lingüísticos. Entre las bases de conocimiento lingüístico destacan:

- Los diccionarios.
- Los lexicones.
- Los tesauros.
- Las bases de datos de relaciones léxicas.

Además, de estas bases de conocimiento lingüístico también cabe destacar las colecciones de textos o de recursos del habla llamados **corpus.**

A continuación, se presentan las definiciones de los diferentes tipos de recursos lingüísticos y, en las siguientes secciones, se presentan la base de datos de relaciones léxicas WordNet y algunos corpus en español.

Diccionario

Es el repertorio donde se agrupa, según un orden determinado, las palabras de una lengua acompañadas de su definición o explicación. Los diccionarios incluyen una descripción detallada y comprensible para un humano de los diferentes sentidos de las palabras.

Lexicón

Un diccionario que contiene información morfológica. Este diccionario morfológico es un repertorio donde se recogen una lista de morfemas y la información básica sobre estos.

Tesauro

También llamado *thesaurus*, *thesauri* o tesoro. Se refiere al diccionario que contiene una lista de significados de las palabras y las relaciones entre estos significados.

- En la literatura, un tesauro contiene una lista de palabras con sus sinónimos y sus antónimos.
- En lingüística, un tesauro reúne el conocimiento sobre las relaciones de hiperonimia/hiponimia y meronimia/holonimia representadas en la propia estructura jerárquica del tesauro. Es decir que, la jerarquía del tesauro modela la relación *is-a* y la relación parte todo de los sentidos de las palabras. Además, un tesauro también puede agrupar conocimiento sobre otras relaciones semánticas como la sinonimia o antonimia.

Bases de datos de relaciones léxicas

Una generalización de los tesauros, es decir, un tipo de diccionario que recoge conocimiento sobre cualquier relación semántica entre sentidos de las palabras. Estas bases contienen un conjunto de lemas, cada uno de los cuales está anotado con el posible conjunto de sentidos de la palabra y las relaciones entre esos sentidos.

Corpus lingüístico

Es una **colección de textos representativos de una lengua** que se utilizan para el análisis lingüístico. Se puede distinguir entre:

- Los **corpus textuales**, que recogen extractos de libros, revistas, periódicos o cualquier otra fuente escrita.
- Los **corpus orales** que son colecciones de habla, es decir extractos de audios provenientes de fuentes como puede ser la radio o la televisión.

Ideas clave

Los corpus pueden estar anotados o etiquetados de forma que las palabras que lo conforman presentan, además, algún tipo de información lingüística, por ejemplo, información sintáctica, semántica o pragmática, son los conocidos como **corpus etiquetados**.

Los primeros corpus disponibles *online* aparecieron en la década de 1960. Uno de los pioneros fue el **Brown Corpus**, un corpus del inglés americano desarrollado en 1963 por la Brown University y que contenía una colección de un millón de palabras extraídas de 500 textos de diferentes géneros: periódicos, novelas, no ficción, académico, etc. (Kucera y Francis, 1967). Al principio el corpus no estaba etiquetado, pero con el paso de los años se etiquetó con información sobre las categorías gramaticales (POS).

Hoy en día los corpus tienden a ser mucho más extensos que el Brown Corpus. Por ejemplo, el corpus etiquetado con información morfosintáctica **WSJ**, se trata de la colección del millón de palabras que se publicaron en los artículos del Wall Street Journal (WSJ) en el año 1989.

Para ampliar la primera versión del *Wall Street Journal (WSJ)* ingresa al siguiente enlace web: https://catalog.ldc.upenn.edu/LDC2000T43

2.6. Corpus en español

En las secciones anteriores se ha comentado sobre algunos corpus, esas colecciones de textos para el análisis lingüístico, utilizados en el procesamiento del lenguaje natural en inglés. En el ámbito hispánico son instituciones como la Real Academia Española, incluida en la Asociación de Academias de la Lengua Española, o el Instituto Cervantes las que se han ocupado de producir recursos lingüísticos que puedan ser utilizados en el procesamiento del lenguaje natural en español.

La Biblioteca Virtual Miguel de Cervantes reúne algunos corpus en español, por ejemplo:

- Corpus de sonetos del siglo de oro: http://goldenage.cervantesvirtual.com/
- Corpus diacrónico del español histórico IMPACT-es (documentación): http://data.cervantesvirtual.com/blog/documentacion-corpus-diacronico/

El Corpus diacrónico del español histórico IMPACT-es (Sánchez-Martínez, Martínez-Sempere, Ivars-Ribes y Carrasco, 2013) contiene 86 obras pertenecientes a la Biblioteca Virtual Miguel de Cervantes e impresas entre los años 1482 y 1647. Este corpus viene acompañado por un lexicón que enlaza más de 10 000 lemas con las diferentes variantes de las palabras encontradas en los documentos. Además, las palabras más frecuentes del corpus están anotadas con su lema, categoría gramatical y su forma moderna equivalente. Esto permite efectuar búsquedas de forma muy eficiente (Carrasco et al., 2015) a través del interfaz web.

- Accede al IMPACT-es: http://data.cervantesvirtual.com/blog/diasearch/
- Disponible para descarga: https://www.digitisation.eu/tools-resources/language-resources/impact-es/

Un ejemplo de búsqueda se presenta en la Figura 1 donde vemos el resultado de la búsqueda en el corpus diacrónico del español histórico (IMPACT-es) de obras literarias en las que aparece una expresión del tipo «hacer algo de», donde el verbo *hacer* puede aparecer en cualquiera de sus formas, como en español antiguo, y la palabra «algo» puede ser cualquier nombre, como por ejemplo *haz examen de* o *haciendo burla de.*

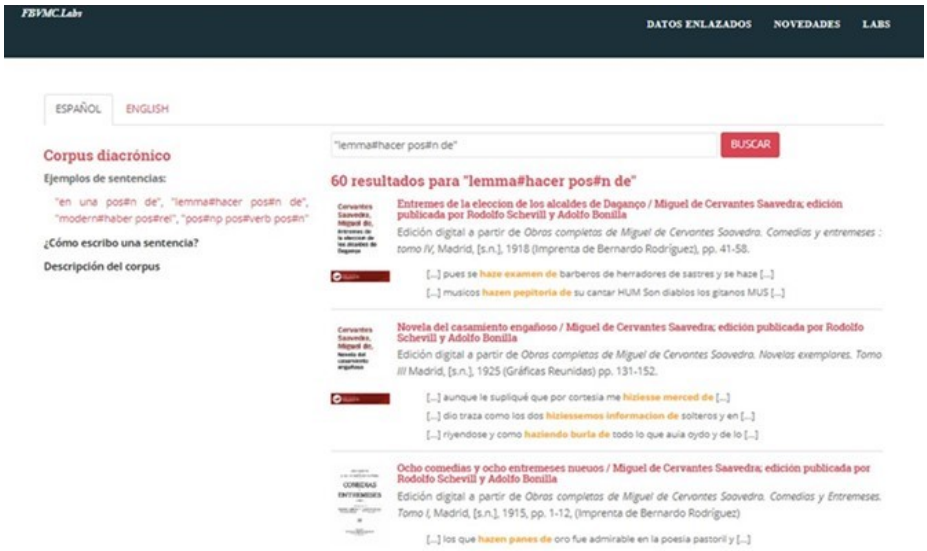


Figura 1. Resultado de la búsqueda en el corpus diacrónico del español histórico. Fuente: BVMC.Labs, s. f.

La Real Academia Española se fijó a finales del siglo pasado el objetivo de generar recursos lingüísticos para mostrar la evolución del español en distintas épocas y su funcionamiento actual en los diferentes países que conforman la comunidad lingüística hispánica. Es por eso por lo que, a partir de 1996, la RAE en colaboración con las otras academias de la Asociación de Academias de la Lengua Española crearon el **Corpus de Referencia del Español Actual (CREA)** y el **Corpus del Español del Siglo XXI (CORPES XXI)**.

Ideas clave

Para completar el estudio de esta sección puedes leer el capítulo 7.1 *CREA y CORPES, corpus de referencia del español* del siguiente libro: Sánchez, M. (2016). CREA y CORPES, corpus de referencia del español. *Tecnologías del lenguaje en España: comunicación inteligente entre personas y máquinas* (119-124). Fundación Telefónica y Ariel. https://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/itempubli/565/

2.7. Herramientas y librerías para el PLN

Existen numerosas herramientas y librerías que facilitan el desarrollo e implementación de diferentes tareas relacionadas con el procesamiento del lenguaje natural. Una de las herramientas más utilizadas para la implementación en Python de aplicaciones para el procesamiento del lenguaje natural es Natural Language Toolkit (NLTK). **NLTK** está disponible para Windows, Mac OS X y Linux, es *open source* y su código se distribuye bajo la licencia Apache License Version 2.0.

NLTK es una plataforma que proporciona interfaces a más de cincuenta corpus y recursos léxicos, como por ejemplo WordNet, además de una serie de librerías para realizar tareas de clasificación, obtención de tokens, derivación, etiquetado morfosintáctico, análisis sintáctico y análisis semántico. NLTK ha sido desarrollado principalmente para el procesamiento del lenguaje natural en inglés. Sin embargo, esta plataforma tiene la flexibilidad necesaria para poder realizar procesamiento del lenguaje natural en otros idiomas como el español.

NLTK se puede utilizar para entrenar un etiquetador morfosintáctico en español a partir de un corpus etiquetado y también permite incorporar un etiquetador externo que sea capaz de analizar el español.

Accede a NLTK a través del aula virtual o desde la siguiente dirección web: http://www.nltk.org/

Otra librería de Python para PLN es **spaCy,** muy usada tanto en el ámbito académico como en el industrial. SpaCy incorpora ya distintos flujos de normalización de textos (como los que se han visto previamente) de manera que se agilice todo el preprocesado de los textos para poder trabajar con ellos en las distintas tareas de PLN. Tiene soporte para más de sesenta idiomas, e incluye distintos modelos

neuronales recientes para distintas tareas de PLN como, por ejemplo, el reconocimiento de entidades nombradas que se comentó previamente. Incluye también modelos preentrenados como BERT, que se verán con más detalle en otros capítulos de la asignatura. SpaCy es un software *open source* para uso comercial, con licencia MIT.

Accede a spaCy a través del aula virtual o desde la siguiente dirección web: https://spacy.io/

También existe la librería **Gensim** *(Generate Similar)* de Python para trabajar con PLN. Con Gensim se pueden llevar a cabo tareas como *topic modelling*, indexado de documentos o búsquedas de documentos mediante el uso de técnicas como vector *embeddings*, los cuales se verán en detalle más adelante en la asignatura.

Accede a Gensim a través de la siguiente dirección web: https://radimrehurek.com/gensim/

Además de estos recursos en Python, cabe destacar algunas librerías para el desarrollo en Java de aplicaciones en el ámbito del procesamiento del lenguaje natural:

- LingPipe: http://alias-i.com/lingpipe/index.html
- Apache OpenNLP: https://opennlp.apache.org/docs/
- Stanford NLP: http://nlp.stanford.edu/software/
- Stanford CoreNLP: https://stanfordnlp.github.io/CoreNLP/

Ideas clave

Stanford CoreNLP incorpora modelos para realizar tareas de procesamiento del lenguaje natural en español. Por lo tanto, permite gestionar algunas características típicas de este idioma, como por ejemplo la separación de contracciones de palabras (del=de + el) durante el proceso de obtención de tokens o la identificación de los tiempos y modos de los verbos (presente de indicativo) en el etiquetado morfosintáctico. Stanford CoreNLP está implementado en Java, pero proporciona acceso por la línea de comandos lo que permite su integración en Python. Esta característica es muy útil si se quiere utilizar el etiquetado morfosintáctico para el español del Stanford CoreNLP en un código desarrollado en Python utilizando la plataforma NLTK.

En el caso de querer realizar una implementación de tareas de procesamiento del lenguaje natural en node.js se puede utilizar la librería Natural.

Accede a la guía de la librería Natural: https://github.com/NaturalNode/natural/blob/master/README.md

---

2.8. Referencias bibliográficas

Aggarwal, C. C. (2018). Text Preparation and Similarity Computation. *Machine Learning for Text* (17-25). Springer International Publishing.

Carrasco, R. C., Martínez-Sempere, I., Mollá-Gandía, E., Sánchez-Martínez, F., Candela-Romero, G. y Escobar-Esteban M. P. (2015). Linguistically-Enhanced Search over an Open Diachronic Corpus. En A. Hanbury, G. Kazai, A. Rauber y N. Fuhr (Eds.), *Advances in Information Retrieval. Lecture Notes in Computer Science, 9022.* https://doi.org/10.1007/978-3-319-16354-3_89

Eisenstein, J. (2019). *Introduction to Natural Language Processing.* MIT press.

Poli, R., Healy, M. y Kameas, A. (2010). WordNet. Fellbaum, C. (A.), *In Theory and applications of ontology. Computer applications* (231-243). Springer.

Jurafsky, D. y Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition and Computational Linguistics.* Prentice-Hall.

Kucera, H. y Francis, W. N. (1967). *Computational analysis of present-day American English.* Brown University Press.

Sánchez-Martínez, F., Martínez-Sempere, I., Ivars-Ribes, X. y Carrasco, R.C. (2013). An open diachronic corpus of historical Spanish. *Language Resources and Evaluation, 47*(4), 1327-1342.

Schütze, H., Manning, C. D., & Raghavan, P. (2008). *Introduction to information retrieval* (39, 234-65). Cambridge University Press.

Porter, M. F. (2001). *Snowball: A language for stemming algorithms.* Computer Science.

Zipf, G. K. (1949). *Human Behavior and The Principle of Least Effort.* Addison-Wesley Press.

Machine Learning for Text

Aggarwal, C. C. (2018). Text Preparation and Similarity Computation. *Machine Learning for Text* (17-25). Springer International Publishing.

Se recomienda la lectura del segundo capítulo, donde se realiza una explicación detallada de qué caracteriza al texto como dato, así como de las fases que se siguen para normalizarlo y poder trabajar con él para llevar a cabo las distintas tareas de PLN.

Natural Language Processing with Python

Bird, S., Klein, E. y Loper, E. (2014). *Natural Language Processing with Python.* Sebastopol. O'Reilly. https://www.nltk.org/book/

Este libro proporciona una introducción práctica a la programación para el procesamiento del lenguaje natural. El libro está escrito por los creadores de NLTK y guía al lector a través de los fundamentos de escribir programas en Python, trabajar con corpus, categorizar texto, analizar estructuras lingüísticas. La versión en línea del libro se ha actualizado para Python 3 y NLTK 3. La versión del libro editada por O'Reilly Media en 2009 presentaba la versión original de NLTK basada en Python 2.

WordNet: An Electronic Lexical Database

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge. MIT Press. http://mitpress.mit.edu/books/wordnet

Este libro tiene dos propósitos. Primero, comentar el diseño de WordNet y las motivaciones teóricas detrás de este. En segundo lugar, proporcionar un análisis de las aplicaciones de WordNet, incluida la identificación del sentido de las palabras, la recuperación de información, la selección de preferencias de los verbos y las cadenas léxicas.

1. El principio de composición:

   A. Indica que la representación del significado de una frase se puede obtener desde la composición del significado de las palabras que la constituyen.

   B. Indica que la representación del significado de un documento se puede obtener desde la composición de las frases que lo forman.

   C. Indica que la representación del significado de una palabra está relacionada con su morfología.

   D. Ninguna de las anteriores.

2. Indica cuál de estas afirmaciones es correcta:

   A. El principio de composición es suficiente para representar el significado de cualquier texto.

   B. La perspectiva distribucional indica que todas las palabras tienen el mismo peso en la representación del significado de un texto.

   C. El aspecto secuencial muestra cómo es irrelevante el orden de las palabras para la construcción del significado de un texto.

   D. Ninguna de las anteriores

3. Indica cuál de estas afirmaciones es cierta respecto a la normalización de textos:

   A. El proceso de tokenización suele ser la última fase del pipeline de normalización de los textos.

   B. Siempre es aconsejable usar técnicas de *stemming* en el pipeline de normalización.

   C. El proceso de lematización puede hacer que se pierda información importante para la construcción del significado de una frase.

   D. La eliminación de stopwords se hace para quitar palabras poco habituales en una lengua.

4. El proceso de *stemming* de la palabra «saltábamos» da como resultado:

   A. «Saltar».

   B. «Salto».

   C. «Salt».

   D. «Saltaba».

5. El proceso de lematización de la palabra «saltábamos» da como resultado:

   A. «Saltar».

   B. «Salto».

   C. «Salta».

   D. «Saltaba».

6. La mínima unidad que forma una palabra y tiene un significado es:

   A. El morfema gramatical.

   B. El lexema.

   C. El morfema.

   D. El lema.

7. Indica las afirmaciones correctas sobre el lexema:

   A. Tienen un significado gramatical.

   B. Conforma la raíz de una palabra.

   C. Tienen un significado léxico.

   D. Proporciona significado a la palabra.

8. Indica las afirmaciones correctas sobre los recursos lingüísticos:

A. Un lexicón es un tipo de diccionario que contiene una lista de lemas.

B. El diccionario es el repertorio donde se recogen las palabras de una lengua acompañadas de su definición o explicación.

C. El tesauro es un tipo de diccionario que contiene información sobre el léxico. aspecto secuencial muestra cómo es irrelevante el orden de las palabras para la construcción del significado de un texto.

D. Las bases de datos de relaciones léxicas contienen un conjunto de lemas anotados con sus posibles sentidos.

9. Indica las afirmaciones correctas sobre un tesauro:

A. Debe contener información sobre las relaciones de sinonimia.

B. Los términos que conforman un tesauro se relacionan entre sí para mostrar las relaciones entre significados.

C. Si la jerarquía del tesauro se basa en las relaciones de hiperonimia/hiponimia, cualquier término del tesauro es un descendiente del concepto de raíz.

D. Debe contener información sobre las relaciones de meronimia.

10. Indica las afirmaciones correctas sobre Natural Language Toolkit (NLTK):

A. Proporciona herramientas para analizar textos en español.

B. Proporciona interfaces a diferentes corpus y otros recursos léxicos como por ejemplo WordNet.

C. Permite integrar software de terceros por ejemplo la API de Stanford CoreNLP.

D. Es la plataforma más utilizada para desarrollar aplicaciones de procesamiento del lenguaje natural en Python.