

Cálamo Educación S.L.  
María Teresa Tijeras Pascual

## **Reconocimiento de entidades**

Cálamo Educación S.L.  
María Teresa Tijeras Pascual

Cálamo Educación S.L.  
María Teresa Tijeras Pascual

# Indice

<b>Reconocimiento de entidades</b>	<b>3</b>
0. Objetivos de la unidad	3
1. Introducción	3
1.1. Fundamentos y conceptos básicos	3
1.2. Proceso de reconocimiento de entidades	4
1.3. Representación del texto como secuencias	6
1.4. Entity linking	8
1.5. Aplicaciones	9
2. Métricas de evaluación	11
3. Técnicas de reconocimiento de entidades	13
3.1. Modelos clásicos de procesamiento del lenguaje natural	14
3.1.1. Características para reconocer/clasificar entidades	15
3.1.2. Tarea del lingüista	17
3.2. Técnicas de aprendizaje automático	17
3.2.1. Algoritmos de aprendizaje automático	18
3.2.1.1. Etiquetado de secuencias	18
3.2.1.2. Inferencia de reglas	23
3.2.2. Deep learning	24
3.2.3. Tarea del lingüista	26
<b>Ejercicios</b>	<b>29</b>
Ejercicio 1: Reconocimiento de entidades con un sistema real	29
1. Introducción	29
2. IBM Watson Natural Language Understanding	29
3. Google Cloud Natural Language	34
Ejercicio 2: Funcionalidades para la tarea de reconocimiento de entidades	36
1. Introducción	36
2. Google Colaboratory	36
3. Creación de un cuaderno nuevo	38
4. Reconocimiento de entidades con NLTK	40
5. Reconocimiento de entidades con spaCy	44
<b>Recursos</b>	<b>47</b>
Enlaces de Interés	47

# Reconocimiento de entidades



El objetivo de esta unidad es estudiar en detalle la tarea del procesamiento del lenguaje natural dedicada al reconocimiento de entidades.

## 0. Objetivos de la unidad

El objetivo de esta unidad es estudiar en detalle la tarea del procesamiento del lenguaje natural dedicada al reconocimiento de entidades. Para ello se van a presentar los fundamentos y conceptos básicos del reconocimiento de entidades, los campos de aplicación más importantes, los modelos de representación de la información y las técnicas utilizadas más importantes, así como las métricas empleadas para evaluar este tipo de sistemas.

En el apartado 1 se presentan los fundamentos, conceptos básicos y ejemplos de aplicaciones de la tarea de reconocimiento de entidades, así como el proceso general de reconocimiento y la tarea particular de *entity linking* (desambiguación y normalización de las entidades detectadas enlazándolas a una referencia externa, por ejemplo, una página de Wikipedia).

En el apartado 2 se describen las métricas de evaluación específicas que existen para esta tarea, como casos particulares o modificaciones de las métricas de evaluación generales presentadas en la unidad 1.

En el apartado 3 se describirán con cierto detalle las diferentes técnicas de reconocimiento de entidades, en particular, los modelos clásicos de procesamiento del lenguaje natural y las técnicas de aprendizaje automático, incluidos los modelos de *deep learning*. En ambos casos se describe en qué consiste la tarea del lingüista en los proyectos de reconocimiento de entidades.

## 1. Introducción

### 1.1. Fundamentos y conceptos básicos

La tarea de reconocimiento de entidades (en inglés, *Entity Recognition*), también denominado extracción o identificación de entidades, tiene como objetivo, como su nombre indica, la identificación y clasificación de las entidades de un texto.

Una entidad es cualquier palabra o secuencia de palabras que se utilice sistemáticamente en el texto para referirse al mismo objeto del mundo real (persona, organización, lugar, producto, etc.), aunque sea con diferentes variantes (nombre completo, solo apellido, pseudónimo, nombre oficial/popular, etc.).

Cada entidad detectada se clasifica en una categoría semántica predeterminada (si es una persona, un lugar, una empresa, un producto, una referencia de tiempo...), según la taxonomía empleada por el sistema.

Angiotensin-converting enzyme 2 **GENE\_OR\_GENOME** ( **ACE2 GENE\_OR\_GENOME** ) as a SARS-CoV-2 **CORONAVIRUS** receptor: molecular mechanisms and potential therapeutic target. SARS-CoV-2 **CORONAVIRUS** has been sequenced [ 3 **CARDINAL** ] . A **phylogenetic EVOLUTION** analysis [ 3 **CARDINAL** , 4 **CARDINAL** ] found a bat **WILDLIFE** origin for the SARS-CoV-2 **CORONAVIRUS** . There is a diversity of possible intermediate hosts for SARS-CoV-2 **CORONAVIRUS** , including pangolins **WILDLIFE** , but not mice **EUKARYOTE** and rats **EUKARYOTE** [ 5 **CARDINAL** ] . There are many similarities of SARS-CoV-2 **CORONAVIRUS** with the original SARS-CoV **CORONAVIRUS** . Using computer modeling , Xu et al . [ 6 **CARDINAL** ] found that the spike proteins **GENE\_OR\_GENOME** of SARS-CoV-2 **CORONAVIRUS** and SARS-CoV **CORONAVIRUS** have almost identical 3-D structures in the receptor binding domain that maintains Van der Waals forces **PHYSICAL\_SCIENCE** . SARS-CoV spike proteins **GENE\_OR\_GENOME** has a strong binding affinity to human **ACE2 GENE\_OR\_GENOME** , based on biochemical interaction studies and crystal structure analysis [ 7 **CARDINAL** ] . SARS-CoV-2 **CORONAVIRUS** and SARS-CoV spike proteins **GENE\_OR\_GENOME** share identity in amino acid sequences and .....

#### Ejemplo de reconocimiento de entidades en el ámbito de la salud

El término entidad es una simplificación del término **entidad con nombre** (en inglés, *named entity*), acuñado en la 6.ª conferencia MUC (*Message Understanding Conference*, MUC-6) [R. Grishman, B. Sundheim (1996). *Message Understanding Conference-6: a brief history*. COLING '96]. En esa 6.ª edición, el objetivo se centró en tareas de extracción de información estructurada de actividades de empresas y actividades relacionadas con la defensa a partir de texto no estructurado, como artículos de periódicos.

Cuando los organizadores estaban definiendo la tarea, observaron que para comprender el texto es esencial reconocer unidades de información relevante como nombres (incluidos los de personas, organizaciones y lugares) y expresiones numéricas como expresiones de tiempo (hora y fecha), de unidades monetarias (dinero) y expresiones porcentuales. La identificación de las referencias a estas entidades en el texto se reconoció como una de las subtareas importantes de la extracción de información, denominándose: **reconocimiento y clasificación de entidades con nombre** (en inglés, *Named-Entity Recognition and Categorization*, NERC).

Como salida, el sistema de reconocimiento de entidades genera una lista de las entidades reconocidas en el texto junto con su tipo semántico y, habitualmente, un valor de relevancia o de importancia dentro de ese texto, casi siempre de forma relativa (de 0 a 100%).

Además, algunos sistemas también devuelven la posición de la entidad en el texto, para poder identificarla en contexto. Para ello la salida incluye un campo para cada entidad que indica la posición del carácter inicial de dicha entidad respecto al inicio del texto ("offset" inicial) y bien la posición del carácter final ("offset" final) o la longitud de la entidad.

Las entidades no incluyen únicamente nombres propios de objetos como "Pablo Picasso", "Gran Bretaña" o "Médicos Sin Fronteras" (lo que serían individuos, objetos o *instances* en una ontología), sino que también incluyen nombres comunes que hacen referencia a una clase de objetos, como "pintor", "país" u "organización no gubernamental" (lo que serían las clases en una ontología).

Sin embargo, algunos sistemas de NER denominan **entidades** (*entities*) únicamente a los individuos y **conceptos** (*concepts*) a las clases, con lo que "Pablo Picasso" sería una entidad y "pintor" sería un concepto.

## 1.2. Proceso de reconocimiento de entidades

El proceso de NER consiste en dos pasos:

### Identificación de las entidades

Identificar, entre todas las palabras de un texto, qué palabras o secuencias de palabras constituyen una entidad con nombre.

Existen diferentes enfoques, por una parte, métodos clásicos basados en recursos lingüísticos y, por otra parte, enfoques basados en aprendizaje automático (que se estudiarán más adelante en esta unidad), pero, en definitiva, todos buscan detectar qué secuencias de palabras (una o más palabras consecutivas del texto, también llamadas *tokens*) representan una entidad con nombre, en las diferentes variantes utilizadas en el texto (por ejemplo “Sr. José Pérez González” o bien “JPG”, “Sr. Pérez” o incluso “Pepe”).

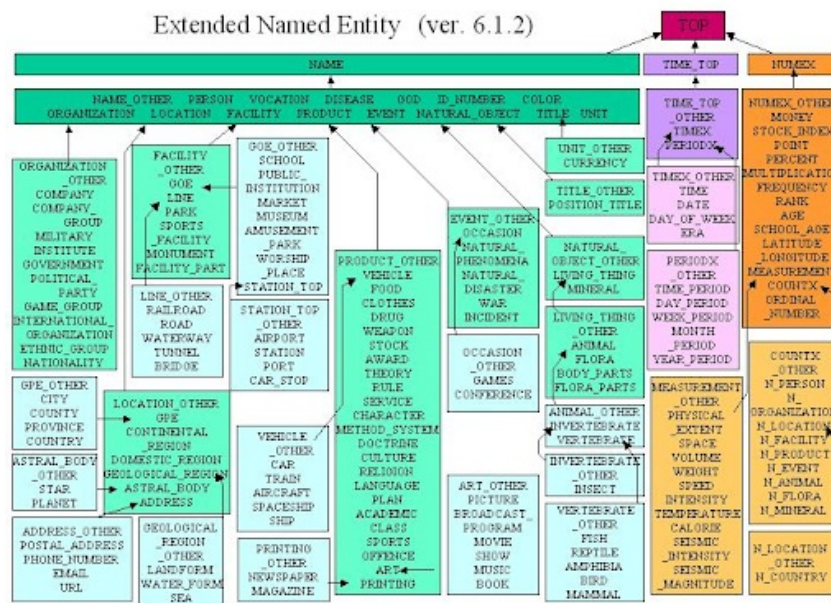
Una vez identificada la entidad, algunos sistemas modifican la segmentación del texto en palabras (*tokens*) agrupando todas las palabras de la entidad en una única unidad multipalabra (un único *token*), mientras que otros sistemas utilizan un etiquetado marcando el primer *token* de la entidad y los *tokens* sucesivos (notación *inside-outside-beginning*, IOB, ver apartado siguiente sobre representación del texto como secuencias).

### Clasificación de las entidades

Consiste en clasificar la entidad detectada en una o varias categorías semánticas según el modelo utilizado por el sistema (persona, lugar, organización, etc.).

Por ejemplo, “Sr. José Pérez González” y cualquiera de sus variantes se etiquetaría, por ejemplo, como PERSON o, en modelos más específicos, se podrían añadir rasgos: PERSON>FULLNAME (nombre completo), PERSON>LAST\_NAME (para “Sr. Pérez”), etc.

Existen muchas taxonomías de tipos de entidades con nombre, por ejemplo, fue muy popular la jerarquía ampliada de Sekine, propuesta en 2002, con más de 200 subtipos de entidad.



*Sekine's Extended Named Entity*

Fuente de la imagen: [New York University](#)

Típicamente, todos los modelos generales contienen las categorías:

- **PERSON.** Nombres de persona: "Manolo Escobar", "Sergio Ramos"...
- **ORGANIZATION.** Nombres de organizaciones o empresas: "Mercadona", "Tribunal Constitucional", "Universidad Carlos III de Madrid"...
- **LOCATION.** Lugares: "España", "Castilla y León", "Buenos Aires", "Museo del Prado"...
- **TIME.** Expresiones de tiempo: "2006", "21:39", "4 p.m."...

Para determinar qué es y qué no es una entidad relevante y cómo categorizarlas, un modelo requiere bien reglas que le permitan tomar decisiones (por ejemplo: [si "X" es va seguida de "S.A." → "X" es ORGANIZATION]) o bien un conjunto de entrenamiento con datos etiquetados para aprender un modelo de detección (un gran volumen de textos indicando qué secuencias son entidades con nombre y de qué tipo es cada una).

Cuanto más relevantes sean las reglas o los datos de entrenamiento para el escenario de aplicación real, más preciso será el sistema para ejecutar la tarea. Por ejemplo, si se entrena el modelo con mensajes de Twitter etiquetados, es probable que luego los resultados de detectar entidades en noticias de prensa sean peores, porque el lenguaje y las expresiones utilizadas en Twitter son diferentes al de los textos de noticias.

### 1.3. Representación del texto como secuencias

Históricamente, una representación habitual del texto para este tipo de sistemas es el formato **IOB** (abreviatura de *inside, outside, beginning*), un formato de etiquetado común para etiquetar *tokens* del texto en lingüística computacional, presentado por Ramshaw y Marcus en su artículo "Text Chunking using Transformation-Based Learning", 1995. En esta nomenclatura, se llama *chunk* ("trozo") a la secuencia de *tokens* que conforman una entidad.

El texto se divide en *tokens* y a cada *token* se le asigna una etiqueta. Hay variantes de IOB, pero normalmente se usa IOB2 donde:

#### Beginning

El prefijo **B-** (*beginning*) en una etiqueta indica que el *token* es el comienzo de una entidad (o *chunk*).

#### Inside

El prefijo **I-** (*inside*) en una etiqueta indica que el *token* "está dentro" (forma parte) de una entidad.

#### Outside

La etiqueta **O** (*outside*) indica que el *token* no pertenece a ninguna entidad.

Por ejemplo, para el texto:



El Museo del Prado está en Madrid, que es la capital de España.

Su segmentación como secuencias en notación IOB sería la siguiente:



El O

Museo B-ORGANIZATION

del I-ORGANIZATION

Prado I-ORGANIZATION

está O

en O

Madrid I-LOCATION

, O

que O

es O

la O

capital O

de O

España I-LOCATION

. O

Las marcas del tipo semántico de entidad varían entre los diferentes sistemas. En el ejemplo se ve que se utilizan los tipos *ORGANIZATION* (organización) y *LOCATION* (lugar).

Una variante de IOB llamada BIOES utiliza también marcas de final y de *chunks* de un solo *token*, consiste en las etiquetas (más bien prefijos de etiquetas) B (*begin*), I (*intermediate*), O (*other*), E (*end*) y S (*single token*). Las entidades con longitud mayor o igual a dos siempre comienzan con la etiqueta B y terminan con la etiqueta E.

Una representación más potente en la actualidad es el formato XML o JSON, que soportan anotaciones mucho más elaboradas y, a menudo, más cortas y legibles. Por ejemplo:





El <ORGANIZATION>Museo del Prado</ORGANIZATION> está en  
<LOCATION>Madrid</LOCATION>, que es la capital de  
<LOCATION>España</LOCATION>.



Para saber más sobre *inside-outside-beginning*, puedes pinchar en este [enlace](#).

## 1.4. Entity linking

La tarea de **entity linking** (enlace o vinculación de entidades), también denominada como *named entity linking* (NEL), *named entity disambiguation* (NED), *named entity recognition and disambiguation* (NERD) o *named entity normalization* (NEN), es una tarea derivada o relacionada con el reconocimiento de entidades que consiste en asignar una identidad única desambiguada a las entidades (como personas famosas, lugares o empresas) mencionadas en el texto.

Por ejemplo, en la frase "París es la capital de Francia", el objetivo es determinar que "París" se refiere a la ciudad de París y no a la popular Paris Hilton o a cualquier otra entidad a la que se pueda referir con ese nombre.



*Entity linking*

Fuente de la imagen: [Wikipedia.org](https://www.wikipedia.org)

El enlace de entidades se diferencia del reconocimiento de entidades en que el NER identifica la aparición de una entidad con nombre en el texto, pero no identifica de qué entidad concreta se trata.

En *entity linking*, las entidades que aparecen en el texto de entrada se asignan (enlazan) a las entidades únicas que corresponden en una base de conocimientos de destino. Esta base de conocimiento de destino depende del escenario de aplicación, pero, en los últimos tiempos, para escenarios abiertos en dominios genéricos se ha popularizado utilizar bases de conocimiento derivadas de Wikipedia (como Wikidata o DBpedia).

Así, el caso particular de *entity linking* que consiste en identificar palabras importantes en un texto y enlazarlas a páginas de Wikipedia (cada página puede considerarse una entidad única, identificada de forma no ambigua por su URL) se suele denominar **wikification**.

Por ejemplo, esta sería la salida de un sistema de que enlaza entidades a páginas de Wikipedia:





<ENTITY url="https://es.wikipedia.org/wiki/Médicos\_Sin\_Fronteras">  
Médicos Sin Fronteras</ENTITY> es una <ENTITY  
url="https://es.wikipedia.org/wiki/Organización\_no\_gubernamental">organización no gubernamental</ENTITY> médica y humanitaria  
internacional que ayuda a las víctimas de<ENTITY  
url="https://es.wikipedia.org/wiki/Desastre">  
desastres</ENTITY> naturales o humanos y de <ENTITY  
url="https://es.wikipedia.org/wiki/Guerra">  
conflictos armados</ENTITY>.

## 1.5. Aplicaciones

El reconocimiento de entidades es adecuado para cualquier situación en la que resulte útil una visión general de alto nivel de un texto, ya que ayuda a identificar fácilmente los elementos clave de un texto, como nombres de personas, lugares, marcas, valores monetarios, etc. La extracción de las entidades principales de un texto ayuda a clasificar los datos no estructurados y a detectar la información importante, lo que resulta crucial si se tienen que manejar grandes conjuntos de datos.

A continuación, se describen algunos casos de uso interesantes del reconocimiento de entidades.

### Atención al cliente

Si tiene que lidiar con un número creciente de tickets de atención al cliente, se pueden utilizar técnicas de reconocimiento de entidades con nombre para gestionar las solicitudes de los clientes con mayor rapidez. Es posible automatizar las tareas repetitivas de atención al cliente, como la categorización de los problemas y las consultas de los clientes, y ahorrar un tiempo valioso que ayuda a mejorar los índices de resolución y a aumentar la satisfacción de los clientes. También puede utilizar la extracción de entidades para extraer datos relevantes, como los nombres de los productos o los números de serie, lo que facilita el enrutamiento de los tickets al agente o equipo más adecuado para gestionar el problema.

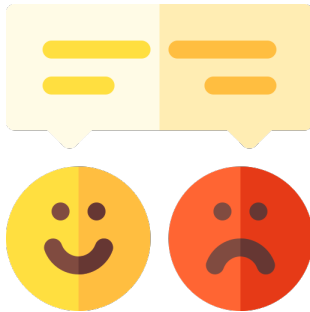


Atención al cliente

### Reseñas de clientes

Las reseñas en línea son una gran fuente de comentarios de los clientes: pueden proporcionar información muy valiosa sobre lo que les gusta y lo que no les gusta de sus productos, y sobre los aspectos de su empresa que deben mejorarse. Los sistemas de extracción de entidades pueden utilizarse para organizar todos estos comentarios de los clientes y detectar los problemas recurrentes. Por ejemplo, se puede utilizar NER para detectar los lugares que se mencionan con más frecuencia en los comentarios negativos de los clientes, lo que podría llevarle a centrarse en una sucursal concreta.

Esto formaría parte de lo que se conoce como análisis de sentimiento orientado a aspectos, es decir, la polaridad de sentimientos de una determinada entidad o concepto ("A es buena" mientras que "B es mala"). Habría que detectar primero gracias a NER la lista de entidades para luego poder realizar el análisis de sentimiento focalizado en ellos.



*Reseñas de clientes*

### Recomendación de contenidos

Muchas aplicaciones modernas (como Netflix y YouTube) se basan en sistemas de recomendación para crear experiencias óptimas para los clientes. Muchos de estos sistemas se basan en el reconocimiento de entidades con nombre, que es capaz de hacer sugerencias basadas en el historial de búsqueda del usuario. Por ejemplo, si un usuario ve muchas comedias en Netflix, recibirá más recomendaciones que hayan sido clasificadas como la entidad Comedia.



*Recomendación de contenidos*

### Recursos humanos

Los encargados de los procesos de selección del personal pasan muchas horas de su día revisando currículos, buscando el candidato adecuado. Cada currículo contiene el mismo tipo de información, pero a menudo están organizados y formateados de manera diferente: un ejemplo clásico de datos no estructurados. Mediante el uso de un extractor de entidades, los equipos de selección de personal pueden extraer al instante la información más relevante sobre los candidatos, desde la información personal (como el nombre, la dirección, el número de teléfono, la fecha de nacimiento y el correo electrónico), hasta los datos relacionados con su formación y experiencia (como las certificaciones, el título, los nombres de las empresas, las habilidades, etc.).



Recursos humanos

## 2. Métricas de evaluación

Para evaluar la calidad de los resultados de un sistema de reconocimiento de entidades, se emplean las medidas habituales en procesamiento del lenguaje natural de precisión (en inglés, *precision*), la cobertura (en inglés, *recall*) y la medida F1 (en inglés, *F1-score*).

Estas medidas estadísticas funcionan bien para los casos obvios de encontrar o no una entidad exactamente igual a la esperada. Es decir, una entidad se considera correcta solo si coincide exactamente con la entidad que se espera en el conjunto de evaluación, tanto en su forma (como "cadena de texto") como en su tipo semántico (tipo de entidad).

Sin embargo, el reconocimiento de entidades puede fallar si señala entidades que son "parcialmente correctas" (por ejemplo, encontrando solo parte del nombre de una entidad multipalabra), por lo que no deberían contarse como aciertos o fallos completos.



Por ejemplo, fallos típicos de los sistemas son identificar una entidad, pero:

- Incluir menos palabras de las esperadas (por ejemplo, en "Telefónica de España" falta el último *token* de "Telefónica de España, S.A.U.").
- Incluir más palabras de las esperadas (por ejemplo, incluyendo el artículo en "El Museo del Prado").
- Dividir incorrectamente una entidad (por ejemplo, tratar "Jordi Sevilla" como dos entidades: "Jordi" y "Sevilla").
- Asignar un tipo semántico incorrecto a una entidad (por ejemplo, identificar una persona como un lugar, o al revés).
- Asignar un tipo semántico relacionado, pero inexacto (por ejemplo, "organización" en vez de su subtipo "universidad").
- Identificar una entidad parcial que es parte de una entidad de mayor alcance (por ejemplo, identificar la persona "Carlos III" y el lugar "Madrid" cuando es parte de "Universidad Carlos III de Madrid").

Un método sencillo para medir la precisión podría ser simplemente contar qué porcentaje de todas las palabras del texto se identificaron correcta o incorrectamente como parte de entidades (o como entidades del tipo correcto). Esto tiene al menos dos problemas: primero, la gran mayoría de las palabras no forman parte de nombres de entidades, por lo que un sistema que devolviera siempre "no es una entidad" tendría de partida una precisión demasiado alta (normalmente >90%); y segundo, no se penaliza adecuadamente no encontrar la extensión completa de un nombre de entidad (por ejemplo, encontrar solo el nombre y no el apellido de una persona se penalizaría como precisión 0.5, que no es correcto porque el nombre por sí solo no sirve como entidad).

De la definición anterior se deduce que cualquier predicción que falle en un solo *token*, que incluya un *token* adicional o que identifique un tipo incorrecto, es un error "estricto" y afecta negativamente a la precisión y cobertura del sistema. Por lo tanto, se puede decir que este criterio es pesimista: puede darse el caso de que muchos "errores" estén cerca de ser correctos y sean adecuados para un propósito determinado. Por ejemplo, un sistema de reconocimiento de entidades que omita por diseño los títulos como "Sr." o "Don", tendría un bajo rendimiento al evaluarlo con datos que sí incluyan los títulos como parte del nombre de las entidades, aunque podría ser que este hecho concreto fuera irrelevante para el objetivo de la organización. Debido a este tipo de problemas, es importante examinar los tipos de errores y decidir su importancia en función de los objetivos y requisitos.

En las conferencias MUC se introdujeron métricas de evaluación considerando diferentes categorías de errores, definidas en términos de comparar la respuesta de un sistema contra las entidades esperadas:

#### Correcto (COR)

Ambas entidades son iguales.

#### Incorrecto (INC)

La salida del sistema y la anotación esperada no coinciden.

**Parcial (PAR)**

El sistema y la anotación esperada son similares, pero no son iguales.

**Ausente (MIS)**

El sistema no ha encontrado una de las entidades esperadas.

**Espurio (SPU)**

El sistema produce una respuesta que no existe en la anotación esperada.

Otra de las conferencias de evaluación, SemEval ([International Workshop on Semantic Evaluation](#)), en su edición de 2013, introdujo cuatro formas simplificadas de medir los resultados de precisión, cobertura y F1 basados en las métricas definidas por MUC:

**Estricta**

Coincidencia exacta del nombre completo y del tipo de entidad.

**Exacta**

Coincidencia exacta del nombre completo de la entidad, independientemente del tipo.

**Parcial**

Coincidencia parcial del nombre de la entidad, independientemente del tipo.

**Tipo**

Coincidencia parcial del nombre de la entidad y coincidencia del tipo.

### 3. Técnicas de reconocimiento de entidades

Para llevar a cabo el reconocimiento de entidades se aplican básicamente dos tipos de técnicas: las técnicas basadas en **métodos clásicos de procesamiento del lenguaje natural**, que hacen uso de recursos terminológicos (diccionarios o léxicos) y/o sistemas de reglas para detectar y clasificar las entidades del texto; y, por otro lado, las técnicas basadas en **métodos de aprendizaje automático**, con clasificadores entrenados específicamente para etiquetar las diferentes unidades de un texto considerado como una secuencia de palabras. Como caso específico de las técnicas de aprendizaje automático están los métodos más modernos de aprendizaje automático con *deep learning*.

En los siguientes apartados describiremos estos tipos de técnicas.

### 3.1. Modelos clásicos de procesamiento del lenguaje natural

La primera de las técnicas para llevar a cabo el reconocimiento de entidades se basa en métodos clásicos del procesamiento del lenguaje natural. Básicamente, estos métodos consisten en la **utilización de recursos lingüísticos** (diccionarios y modelos de reglas) que representan el conocimiento del sistema para llevar a cabo la tarea de reconocimiento.


Los **diccionarios** (o recursos léxicos) contienen listas de entidades, con sus diferentes variantes (alias, sinónimos, etc.) y su tipo semántico. Estos diccionarios se utilizan para reconocer las entidades en el texto y asignarles su tipo.

Según el sistema de reconocimiento de entidades que se utilice, la sintaxis de estos recursos léxicos es diferente, pero básicamente consisten en una lista de entradas donde cada entrada contiene la forma principal de la entidad, opcionalmente, una serie de variantes o formas alternativas, y el tipo semántico de la entidad.

En el caso de que haya diferentes variantes etiquetadas en el diccionario, todas ellas se refieren a una misma forma canónica o fundamental, que también está indicada en ese recurso léxico. Estas variantes se utilizan para indicar al sistema formas alternativas de mencionar esa entidad en el texto.

Según el escenario, los diccionarios de entidades pueden contener desde unos centenares de entradas hasta varias decenas de miles.


Un ejemplo de diccionario de entidades podría ser el siguiente:



Entidad: Real Madrid C.F.  
 Variantes: Real Madrid, El Madrid, el equipo blanco  
 Tipo: EQUIPO

Entidad: Manchester United Football Club  
 Variantes: Manchester United F.C., Manchester United, United  
 Tipo: EQUIPO

De esta forma, dada la siguiente frase de ejemplo:



El Real Madrid jugará el próximo mes la final de la Champions contra el Manchester United.

El sistema de reconocimiento de entidades será capaz de reconocer "Real Madrid" y "Manchester United" como entidades y asignarles el tipo semántico de "equipo de fútbol".

Adicionalmente, además del diccionario de entidades, el sistema puede utilizar un **modelo de reglas** que indique la forma de componer nombres de entidades y que se emplee para no tener que incluir en el diccionario todas las variantes posibles.

Por ejemplo, un sistema podría tener una regla de reconocimiento como "las palabras en mayúsculas son entidades candidatas" y otra regla "las entidades que empiecen por "D." o "Dña." seguido de una o varias palabras en mayúscula, son nombres de persona".

Otro ejemplo de regla podría ser:



EQUIPO C.F.|CF|F.C.|FC|Football Club|Club de Fútbol → EQUIPO



“Un nombre de equipo seguido de “C.F.” o “CF” o “Club de Fútbol” es también un equipo”.

Con esta regla el diccionario podría simplificarse eliminando variantes que se puedan generar automáticamente, y el sistema seguiría siendo capaz de detectarlas en el texto:



Entidad: Real Madrid  
Variantes: El Madrid, equipo blanco  
Tipo: EQUIPO

Entidad: Manchester United  
Variantes: United  
Tipo: EQUIPO

De forma alternativa, la regla podría indicar una forma de componer variantes eliminando fragmentos del nombre de la entidad, por ejemplo, eliminando “C.F.”, “Football Club”, etc.

Otro uso de este modelo de reglas, aparte de detectar variantes de una entrada en el diccionario, podría ser sugerir nuevas entradas para el diccionario, que posiblemente deben ser supervisadas a posteriori por el lingüista, para así aumentar la cobertura de detección.

Estos recursos del sistema (diccionario y modelo de reglas de generación) pueden ser para un único idioma, o bien ser multidioma. En este caso, el diccionario de entradas tendría las variantes en los diferentes idiomas soportados por el sistema.

En los idiomas flexivos, como el español, donde existe un proceso de flexión para la generación de las formas (en nuestro caso en general para nombres comunes, adjetivos, determinantes, pronombres y verbos), para no tener que incluir reglas de flexión para detectar conceptos (entidades de tipo clase), es necesario completar el sistema con un módulo encargado de realizar la lematización o determinar el análisis morfosintáctico de las palabras.

### 3.1.1. Características para reconocer/clasificar entidades

En esta sección se describen las características más utilizadas para el reconocimiento y la clasificación de entidades con nombre. Se pueden organizar en tres tipos diferentes: características a nivel de palabra, características de búsqueda de listas y características de documentos y corpus.

#### Características a nivel de palabra

Las características a nivel de palabra están relacionadas con la composición de caracteres (letras) de las



palabras. En concreto, se refieren al uso de mayúsculas y minúsculas en las palabras, la puntuación, valores numéricos y caracteres especiales.

Por ejemplo:

- **Capitalización:** si la palabra comienza con una letra mayúscula, si la palabra está toda en mayúsculas o la palabra tiene capitalización mixta (por ejemplo, "eBay").
- **Uso de signos de puntuación:** si la palabra termina con punto o tiene un punto (por ejemplo, "Sr."), un apóstrofo (por ejemplo, "O'Connor"), un guion o un carácter *ampersand* (por ejemplo, "M&A").
- **Dígitos:** los dígitos pueden expresar una amplia gama de información útil, como fechas, porcentajes, intervalos, identificadores, etc. Hay que prestar especial atención a algunos patrones particulares de dígitos. Por ejemplo, los números de dos y cuatro dígitos pueden representar años y, cuando van seguidos de una "s", pueden representar una década; uno y dos dígitos pueden representar un día o un mes. Otros ejemplos son el uso de símbolo cardinal y ordinal (en español, º, ª), el empleo de números romanos o una palabra con dígitos (por ejemplo, "W3C" o "3M").
- **Rasgos morfológicos:** relacionados esencialmente con los afijos y las raíces de las palabras (prefijos, sufijos, raíces o lemas de palabras). Por ejemplo, en inglés la terminación "ist" indica una profesión (journalist).
- **Análisis morfosintáctico (en inglés, Part-Of-Speech, POS):** desambiguación como nombre propio, nombre común o verbo, o, por ejemplo, uso de palabras extranjeras.
- **Patrones de palabras:** entidades generadas como patrones de letras alfanuméricas, no alfanuméricas, con una cierta longitud de *token* o de frase. Por ejemplo, el código ISBN o el número de pasaporte de un determinado país.

### Listas (o tablas de lookup)

Las listas son los recursos más importantes en el reconocimiento de entidades. Los términos "diccionario", "léxico" o a veces "nomenclátor", suelen utilizarse indistintamente con el término "lista".

La inclusión en una lista es una forma de expresar la relación "es un" (por ejemplo, "París es una ciudad"). Puede parecer obvio que si una palabra ("París") es un elemento de una lista de ciudades, entonces la probabilidad de que esta palabra sea ciudad en un texto determinado es alta. Sin embargo, debido a la polisemia de las palabras, la probabilidad casi nunca es del 100% (por ejemplo, la probabilidad de que "Julio" represente a una persona es baja, debido a que el sustantivo común "julio", como nombre del mes, es mucho más frecuente).

Existen diferentes tipos de listas:

- **Listas de propósito general:** diccionarios generales, palabras de parada, nombres comunes capitalizados (por ejemplo, en inglés, los nombres de los meses o los días de la semana), listas de abreviaturas de uso frecuente...
- **Listas de entidades:** diccionarios de organizaciones, empresas, organismos del gobierno, aerolíneas, universidades...; nombres y apellidos (por separado) de personas, nombres de celebridades, listas de cuerpos astrales (planetas, constelaciones, asteroides), continentes, países, estados/comunidades autónomas, provincias, ciudades, barrios...
- **Listas de indicadores de entidad:** palabras típicas de nombres de organización (por ejemplo, "S.A."), título de la persona ("doctor"), prefijo del nombre ("D." o "Sr."), indicador de lugares (ejemplo, "calle ABC", "museo de XYZ"), etc.

### Características del texto y/o del corpus

Las características del texto se definen tanto sobre su contenido como sobre su estructura. Las grandes colecciones de documentos (corpus) son también excelentes fuentes de características. Estas características van más allá de una única palabra o varias palabras consecutivas, e incluyen la metainformación sobre los textos y las estadísticas del corpus:

Por ejemplo:

- **Ocurrencias múltiples:** análisis de otras entidades en el contexto, ocurrencias en mayúsculas y minúsculas, uso de anáfora o correferencia (las correferencias son las apariciones de una palabra o secuencia de palabras que se refieren a una entidad determinada dentro de un texto).
- **Sintaxis local:** presencia en un contexto de enumeración o de aposición, o su posición en la frase, en el párrafo y en el texto.
- **Metainformación:** por ejemplo, si la palabra es parte de una URL, cabecera de correo electrónico, sección XML; contexto de listas con viñetas/números, tablas, figuras, etc.
- **Frecuencia en el corpus:** desambiguación o decisión sobre la entidad en función de su frecuencia de palabras y frases, la presencia de coocurrencias o con diferentes capitalizaciones.

### 3.1.2. Tarea del lingüista

La principal tarea del lingüista en este tipo de sistemas es incorporar al sistema el conocimiento necesario para identificar las entidades con sus tipos semánticos que sean relevantes en un determinado escenario de aplicación.

La sintaxis de los diccionarios de entidades las reglas de composición o generación de variantes varía según la tecnología del sistema de reconocimiento utilizado.

En general, los diccionarios de entidades son listas de entradas con diferentes campos, y son más sencillas de generar. Muchos sistemas permiten incorporar las entidades al diccionario utilizando un formato CSV o un fichero Excel.

Respecto a las reglas de generación de variantes, su variabilidad es mucho mayor. Algunos sistemas tienen reglas de generación de entidades directamente integradas en el código fuente del sistema. En este caso, si no se puede acceder al código fuente del sistema no sería posible introducir reglas adicionales. Otros sistemas sí soportan reglas escritas en un fichero externo, con una sintaxis específica, que se lee en tiempo de ejecución.

Otra de las tareas que suele estar a cargo del lingüista es la evaluación de los resultados del sistema, utilizando las métricas definidas para el proyecto. Con las conclusiones obtenidas habitualmente se realiza una nueva fase de edición de los modelos para corregir los problemas encontrados.

## 3.2. Técnicas de aprendizaje automático

Como se ha visto anteriormente, las técnicas iniciales para llevar a cabo reconocimiento de entidades se basaban sobre todo en métodos clásicos del procesamiento del lenguaje natural como diccionarios y reglas elaboradas a mano. Sin embargo, los enfoques más modernos utilizan el aprendizaje automático supervisado, que aprenden a reconocer y clasificar las entidades de un texto entrenando sobre ejemplos etiquetados.



Para saber más sobre este aspecto, ver apartado “Aprendizaje automático” en la unidad 1 del curso.

Básicamente, los métodos basados en aprendizaje automático consisten en etiquetar manualmente un gran volumen de textos indicando las entidades contenidas y su tipo semántico. Habitualmente se utiliza una representación donde el texto es una secuencia de palabras (*tokens*) y lo que se etiqueta manualmente es cada palabra indicando si es o no una entidad y, si lo es, el tipo de entidad.



Para saber más sobre este aspecto, ver apartado “Representación del texto como secuencias” de esta unidad.

Ese conjunto de entrenamiento se utiliza para ejecutar un algoritmo de aprendizaje que sea capaz de aprender a distinguir entidades de un texto y saber clasificarlas según se le ha enseñado. En definitiva, los algoritmos lo que aprenden es a adivinar qué palabras de la secuencia son entidades.

Los sistemas basados en diccionarios y reglas, elaborados a mano, obtienen generalmente una mejor precisión, pero a expensas de una menor cobertura y un gran esfuerzo de trabajo de lingüistas computacionales experimentados.

El principal inconveniente del aprendizaje automático es que es necesario disponer de un gran volumen de textos etiquetados en el dominio de aplicación para realizar el entrenamiento. Este conjunto de textos lo construyen también expertos lingüistas, con un esfuerzo considerable, aunque menor que elaborar diccionarios y reglas. Cuando no se dispone de ejemplos de entrenamiento, los diccionarios y las reglas elaboradas a mano siguen siendo la técnica preferida (en realidad, la única técnica disponible).

Como alternativa, se han propuesto **métodos semisupervisados** para evitar parte del esfuerzo de anotación del corpus. Un método sería la técnica de *bootstrapping*, que consiste en generar como punto de partida de forma manual un diccionario o sistema de reglas reducidos, que se utiliza para etiquetar un conjunto de textos que sirven como entrenamiento para construir un modelo de aprendizaje supervisado utilizando alguno de los algoritmos disponibles. Tras ello, se ejecuta el modelo con más textos, se corrige la salida manualmente, y los resultados se emplean para actualizar el diccionario o sistema de reglas, ampliándolos con el nuevo conocimiento obtenido. Repitiendo este proceso en varias iteraciones es posible generar un conjunto de recursos suficientemente rico y amplio como para llevar a cabo la tarea de reconocimiento de entidades.

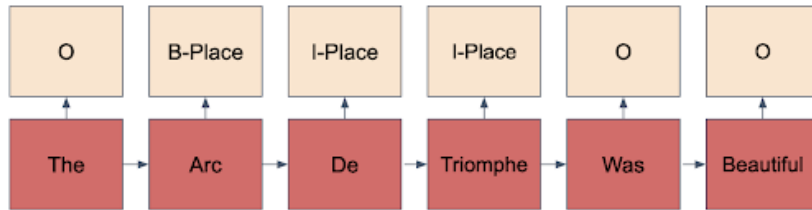
Por ejemplo, para desarrollar un sistema de reconocimiento de “nombres de enfermedades”, puede pedirse al usuario que proporcione un pequeño número de nombres de ejemplo. A continuación, el sistema busca frases que contengan esos nombres e intenta identificar algunas pistas contextuales comunes a los ejemplos proporcionados. A continuación, el sistema trata de encontrar otros casos de nombres de enfermedades que aparezcan en contextos similares. El proceso de aprendizaje se vuelve a aplicar a los nuevos ejemplos encontrados para descubrir nuevos contextos relevantes. Repitiendo este proceso, se acaba reuniendo un gran número de nombres de enfermedades y un gran número de contextos.

### 3.2.1. Algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático “clásico” para reconocimiento de entidades se aplican principalmente para etiquetado de secuencias e inferencia de sistemas de reglas.

#### 3.2.1.1. Etiquetado de secuencias

Estas técnicas consisten en entrenar un modelo que sea capaz de predecir qué etiqueta va a tener cada *token* que forma el texto (secuencia).



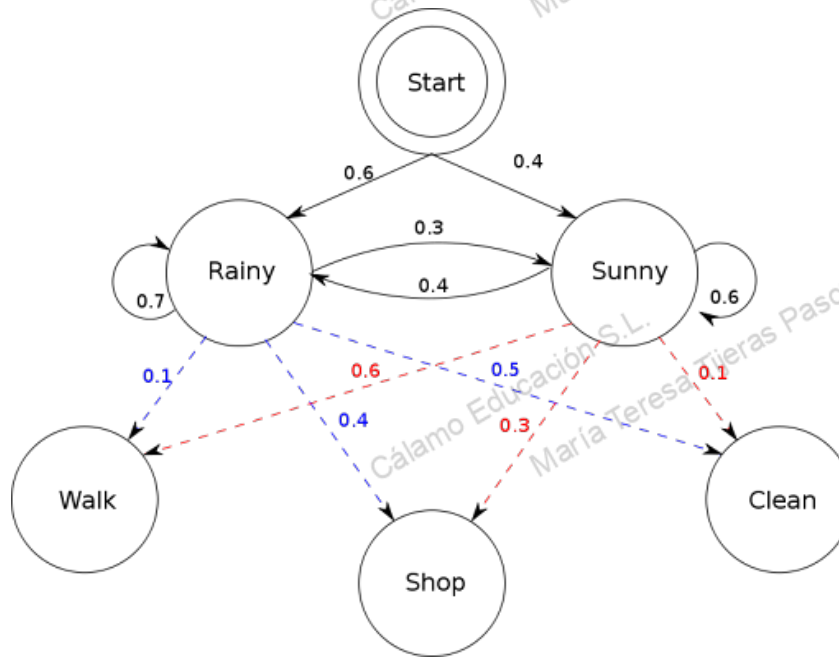
*Etiquetado de la secuencia "The Arc De Triomphe was beautiful" con notación IOB*

Dado un conjunto de secuencias de *tokens* etiquetados, por ejemplo, con la notación IOB, los modelos intentan aprender los mecanismos por los que un determinado *token* de la secuencia tiene una determinada etiqueta y no otra. Los modelos utilizan los *tokens* del contexto (*tokens* anteriores y posteriores de la secuencia) y/o las marcas B o I para considerar el inicio o final de las entidades.

Entre los algoritmos más utilizados están los siguientes:

### Modelos Ocultos de Markov

Los Modelos Ocultos de Markov (*Hidden Markov Models*, HMM) son un modelo estadístico/probabilístico, basado en sus raíces en la probabilidad bayesiana, que intenta modelar los datos como un proceso de Markov de parámetros desconocidos; en otras palabras, un tipo de autómata de estados, que represente una gramática dependiente del contexto que sirva para reconocer y clasificar las entidades.



Ejemplo de Modelo Oculto de Markov

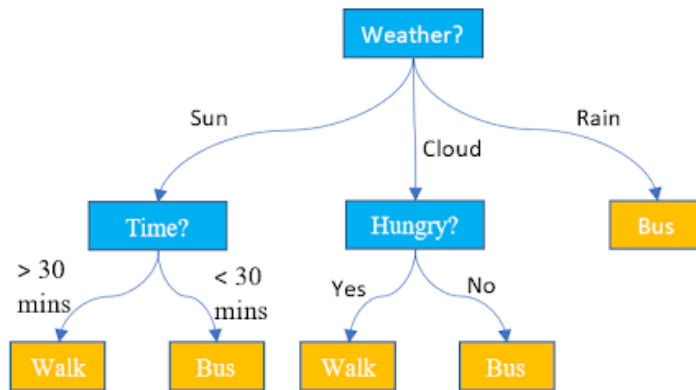
Fuente de la imagen: [Medium.com](https://medium.com)



Para saber más sobre el modelo oculto de Márkov, puedes pinchar en este [enlace](#).

### Árboles de decisión

Un árbol de decisión es un algoritmo que modela los datos de ejemplo mediante un árbol en el cual los nodos intermedios representan decisiones sobre las variables de entrada y los nodos finales (hoja) son los resultados, es decir, la categoría o resultado predicho para el ejemplo.

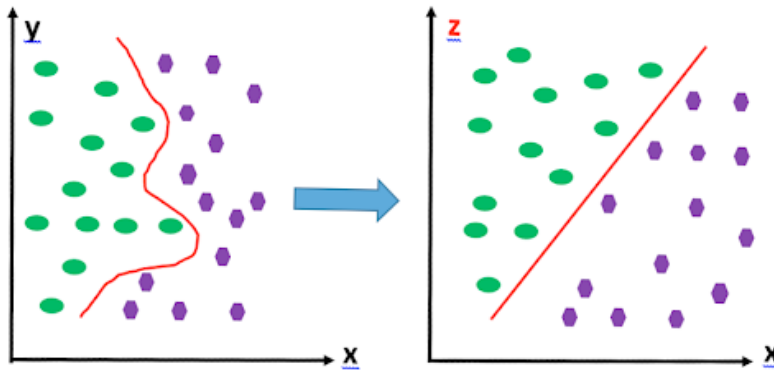


Ejemplo de árbol de decisión

Fuente de la imagen: [Displayr.com](http://Displayr.com)

### Máquinas de vectores soporte

Las máquinas de vectores soporte (*Support Vector Machines*, SVM) son algoritmos de clasificación binaria, que transportan los datos de entrada a un hiperespacio de mayor dimensionalidad mediante una función llamada *kernel*, en el que las diferentes clases a predecir se pueden separar fácilmente mediante una recta (representada por el llamado *vector soporte*), como se ilustra en la siguiente figura.



Ejemplo de proyección en SVM

Fuente de la imagen: [Software.intel.com](http://Software.intel.com)



Para saber más sobre las máquinas de vectores de soporte, puedes pinchar en este [enlace](#).

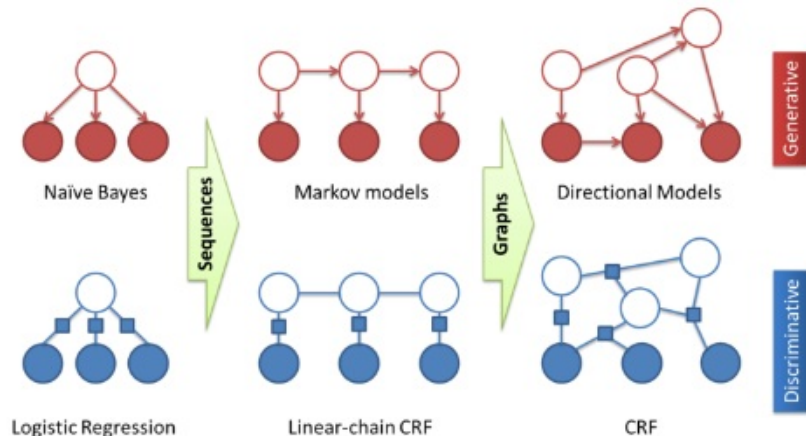
Cálamo Educación S.L.  
María Teresa Tijeras Pascual

Cálamo Educación S.L.  
María Teresa Tijeras Pascual



### Campos aleatorios condicionales

Los campos aleatorios condicionales (*Conditional Random Fields*, CRF) son un modelo probabilístico utilizado específicamente para etiquetar y segmentar secuencias de datos o extraer información de documentos. Se inspiran también en los modelos de Markov, pero en este caso representan el conocimiento mediante grafos no dirigidos entre variables aleatorias, que se construye a partir de los ejemplos de entrenamiento, extrayendo un conjunto de características que representan las dependencias existentes entre diferentes estados y entre estos y la secuencia de observaciones (ejemplos). En algunos contextos también se les denomina campo aleatorio de Markov (del inglés *Markov Random Fields*, MRF).



Adapted from C. Sutton, A. McCallum, "An Introduction to Conditional Random Fields", ArXiv, November 2010

*Conditional Random Fields*

Fuente de la imagen: [Datasciencecentral.com](http://Datasciencecentral.com)



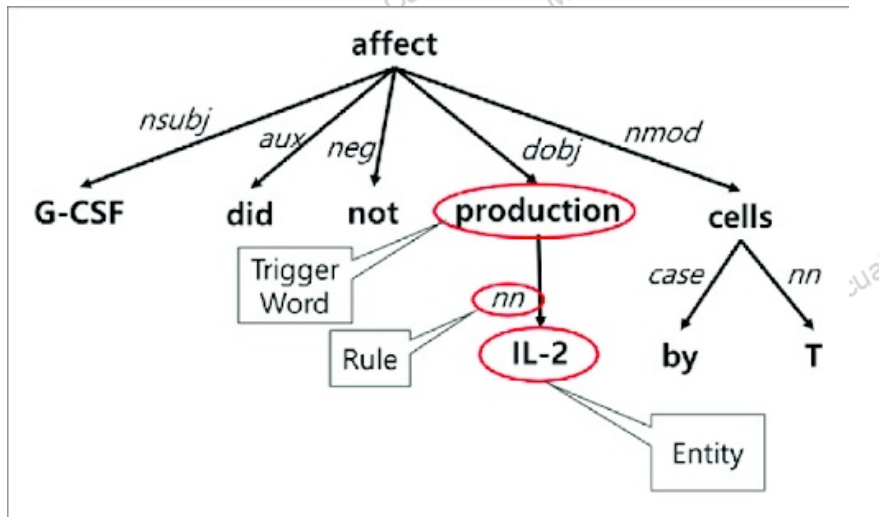
Para saber más sobre el campo aleatorio condicional, puedes pinchar en este [enlace](#).

#### 3.2.1.2. Inferencia de reglas

El objetivo de estas técnicas es la Inferencia (o generación) automática de sistemas de reglas a partir de secuencias etiquetadas. El modelo memoriza listas de entidades y crea reglas de detección y/o desambiguación basadas en características discriminativas (aquellas que permiten diferenciar entre contextos).

El método más básico consiste en "aprender" las etiquetas de los *tokens* del corpus de entrenamiento y aplicarlas para anotar los *tokens* de un texto nuevo. En este caso, el rendimiento (la precisión y cobertura del sistema) depende de la transferencia de vocabulario, es decir, la proporción de palabras, sin repeticiones, que aparecen tanto en el conjunto de datos de entrenamiento como en el texto que se va a etiquetar.

En modelos más avanzados se es capaz de identificar automáticamente reglas como las descritas en la sección de técnicas clásicas para el reconocimiento de entidades, usando las diferentes características a nivel de palabra, léxico o del corpus de entrenamiento completo.



Ejemplo de generación de regla de reconocimiento de entidades

Fuente de la imagen: [Researchgate.net](https://www.researchgate.net)

### 3.2.2. Deep learning



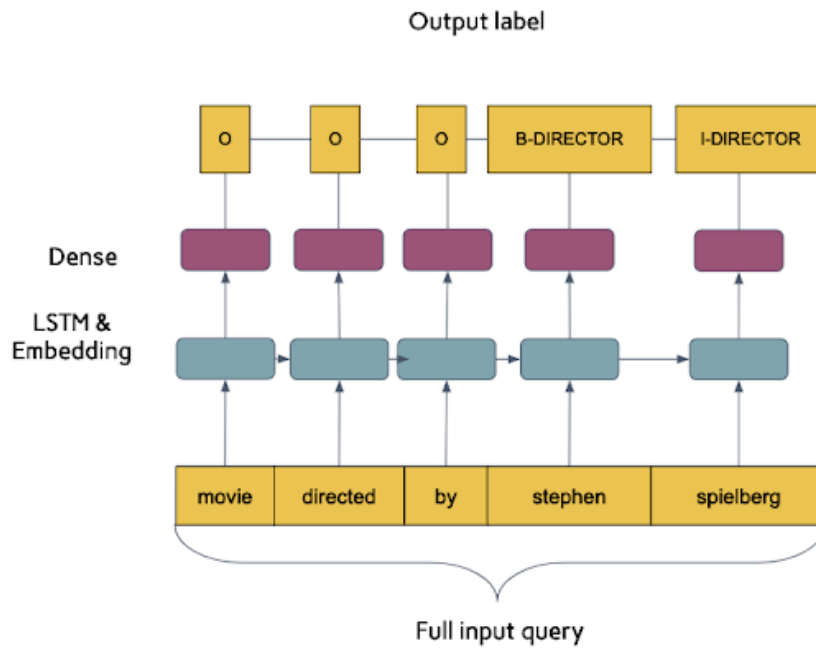
Para saber más sobre este aspecto, ver apartado "Deep learning" en la unidad 1 del curso.

Dado que *deep learning* no es más que un tipo especial de aprendizaje automático, las técnicas empleadas son similares a las explicadas en el apartado anterior, aunque lógicamente utilizando algoritmos basados en redes neuronales profundas.

En este caso se parte igualmente de un gran volumen de textos etiquetados con las entidades contenidas y su tipo semántico. De igual forma, habitualmente se utiliza una representación donde el texto es una secuencia de *tokens* y lo que se etiqueta manualmente es cada *token* indicando si es o no una entidad y, en caso afirmativo, su tipo.

De esta forma, para abordar el problema del reconocimiento de entidades se puede utilizar cualquier arquitectura de red neuronal profunda que tenga en cuenta que el texto es una secuencia de *tokens*.

Por ejemplo, se utilizan mucho las redes con LSTM (*Long-Short Term Memory*), es decir, memoria a largo y corto plazo, que incorporan información del contexto y además mecanismos de atención para centrarse en las palabras más importantes. Para mejorar los resultados se utilizan LSTM bidireccionales, una combinación de dos LSTM donde una va hacia delante de "derecha a izquierda" y otra va hacia atrás de "izquierda a derecha". Esto permite que para decidir sobre un *token* se pueda utilizar la información completa del contexto: la información de las etiquetas de los tokens anteriores, la información del *token* actual, y también las etiquetas de los tokens posteriores. Se utiliza habitualmente una capa inicial de *embeddings* para convertir los *tokens* en su representación como vector, aumentando la potencia de análisis semántico del modelo. Para decidir sobre la etiqueta del *token* se utiliza en la salida un perceptrón sencillo (una capa densa), como se muestra en la figura.



#### Ejemplo de reconocimiento de entidades con LSTM

Fuente de la imagen: [Medium.com](https://medium.com)

La siguiente figura ilustra un proceso completo de extracción de entidades utilizando LSTM en el ámbito de textos científicos:

1

Se procesan los resúmenes (*abstracts*) de artículos para convertirlos en secuencias de *tokens* (*tokenization*), que los lingüistas etiquetan como entidades con su tipo semántico correspondiente (*labeling*).

2

Todo el conjunto de datos se divide en dos partes: el conjunto de datos de entrenamiento (*training*) (que se utiliza para entrenar un modelo de aprendizaje automático) y el conjunto de datos de evaluación (*test*, que se utiliza para evaluar el sistema completo).

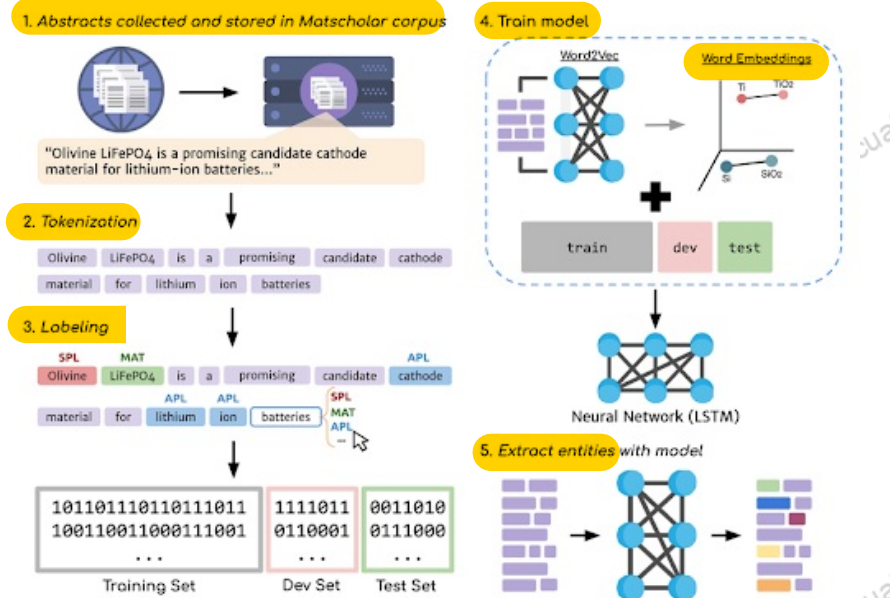
Además, en ciertos modelos, como es el caso de las redes neuronales (sean o no *deep learning*), el conjunto de datos de entrenamiento se divide a su vez en dos partes: el conjunto de datos para entrenamiento (*training*) propiamente dicho, y el conjunto de datos de validación (*development*, que se emplea para evaluación interna del algoritmo durante el proceso de aprendizaje).

3

Usando el conjunto de datos de entrenamiento, se entrena un modelo basado en LSTM, con una capa inicial de *embeddings* que convierte los *tokens* en vectores para aumentar el conocimiento semántico.

4

El modelo entrenado se evalúa con el conjunto de datos de test para obtener las métricas de rendimiento del sistema.



*Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature*

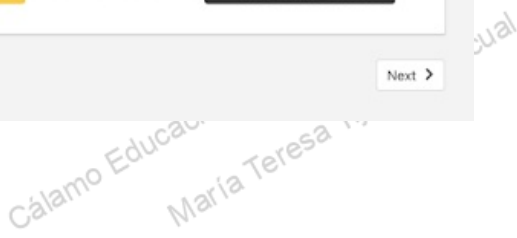
Fuente de la imagen: [Researchgate.net](https://www.researchgate.net)

### 3.2.3. Tarea del lingüista

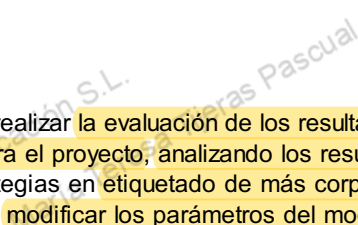
La tarea de índole lingüística en este tipo de sistemas consiste principalmente en el desarrollo del conjunto de entrenamiento adaptado a la tarea que se necesite y, por otro lado, la validación de los resultados en términos de precisión y cobertura de las diferentes entidades y tipos semánticos que genera el sistema.

Por una parte, el lingüista colabora en la definición de los tipos semánticos de las entidades que se tienen que reconocer, y realiza el etiquetado de las secuencias de acuerdo con esa clasificación. Aunque el lingüista puede trabajar con la notación IOB, es habitual realizar un etiquetado utilizando XML o directamente un programa de anotación como Doccano o Brat. Estas aplicaciones (*opensource* en ambos casos) permiten la anotación rápida de entidades con sus tipos semánticos e indicar las relaciones entre ellas, además los resultados pueden exportarse a diferentes formatos estándar (como XML o IOB) para su procesamiento por el sistema de reconocimiento de entidades.

cau.  
María Teresa



Fuente de la imagen: [Github.com](https://github.com)



Fuente de la imagen: [Brat.nlplab.org](http://Brat.nlplab.org)

val



## RESUMEN

Esta unidad se ha centrado en la tarea del procesamiento del lenguaje natural dedicada al **reconocimiento de entidades**. Una **entidad** es cualquier palabra o secuencia de palabras que se utilice sistemáticamente en el texto para referirse al mismo objeto del mundo real (persona, organización, lugar, producto, etc.). Algunos sistemas de NER denominan **entidades** (*entities*) a los individuos y **conceptos** (*concepts*) a las clases ("Pablo Picasso" sería una entidad y "pintor" sería un concepto).

El proceso de reconocimiento de entidades tiene lugar en dos fases. Primero se realiza la **identificación de las entidades**, que consiste en determinar, entre todas las palabras de un texto, qué palabra o secuencias de palabras constituyen una entidad. A continuación, se realiza la **clasificación de las entidades**, que consiste en categorizar cada entidad detectada en una o varias categorías semánticas de las incluidas en el modelo utilizado por el sistema (por ejemplo, persona, lugar, organización, etc.).

Para llevar a cabo este proceso habitualmente se utiliza una representación del texto como una secuencia de palabras o *tokens*.

Existen dos tipos de técnicas de reconocimiento de entidades: 1) los **modelos clásicos** de procesamiento del lenguaje natural, que se basan en recursos lingüísticos (diccionarios y modelos de reglas) que representan el conocimiento del sistema para extraer y clasificar las entidades; y 2) modelos basados en **aprendizaje automático**, ya sea aprendizaje automático "clásico" o basado en las modernas técnicas de *deep learning*, que consisten en cualquier algoritmo que pueda tomar como entrada la representación del texto como secuencias y bien generar un clasificador para el etiquetado de secuencias, o bien llevar a cabo la inferencia de un modelo de reglas que luego puede ser editado por los lingüistas.

En el apartado de **métricas de evaluación** se han descrito las métricas específicas que se utilizan en el caso del reconocimiento de entidades, que varían respecto a las métricas tradicionales de precisión y cobertura para adaptarlas en los casos de no coincidencia exacta en la forma completa de la entidad o el tipo semántico identificado.

## Ejercicios

### Ejercicio 1: Reconocimiento de entidades con un sistema real

Duración estimada del ejercicio



**30**  
minutos

#### 1. Introducción



El objetivo de este ejercicio es familiarizarse con la tarea de reconocimiento de entidades empleando un sistema real.

Existen muchas soluciones comerciales de analítica de texto en Internet, de numerosos fabricantes, incluidos los grandes proveedores de servicios en la nube como son Google, Microsoft e IBM.



En este ejercicio se van a utilizar las páginas de demostración de las plataformas de analítica de texto de IBM, llamada IBM Watson Natural Language Understanding, y de Google, llamada Google Cloud Natural Language.

#### 2. IBM Watson Natural Language Understanding



El sitio web de la plataforma es: <https://www.ibm.com/cloud/watson-natural-language-understanding>

Según la página, IBM Watson Natural Language Understanding utiliza *deep learning* para extraer el significado y los metadatos de los datos de texto no estructurados. Ofrece funcionalidad de analítica de texto para extraer categorías, clasificación, entidades, palabras clave, sentimiento, emoción, relaciones y obtener el análisis sintáctico.



## Reconocimiento de entidades

IBM Cloud Products Solutions Pricing Docs Support Explore more

Watson Natural Language Understanding

Features Pricing FAQ Resources

The natural language processing (NLP) service for advanced text analytics

Get started free View demo

Overview

Watson Natural Language Understanding

IBM Watson® Natural Language Understanding uses deep learning to extract meaning and metadata from unstructured text data. Get underneath your data using text analytics to extract categories, classification, entities, keywords, sentiment, emotion, relations, and syntax.

Keyword Sentiment Scores

Keyword	Sentiment	Score
Watson	Positive	0.75
IBM	Positive	0.65
Natural Language Understanding	Positive	0.65
Watson	Positive	0.65
IBM	Positive	0.65
Natural Language Understanding	Positive	0.65
Watson	Positive	0.65
IBM	Positive	0.65
Natural Language Understanding	Positive	0.65

Benefits Cost savings ROI Save time

6.1 383% 50%

### Watson Natural Language Understanding



Pulsa en el enlace que pone “View demo” para ir al demostrador, que corresponde a la siguiente página: <https://www.ibm.com/demos/live/natural-language-understanding/self-service/home>

El demostrador permite escoger el dominio del texto entre legal, finanzas y medios de comunicación (noticias), y procesar un texto de ejemplo o un texto propio.

SAMPLE INDUSTRY DOMAINS TRY YOUR OWN

Legal Financial Media

Input Text URL

Under the IBM Board Corporate Governance Guidelines, the Directors and Corporate Governance Committee and the full Board annually review the financial and other relationships between the independent directors and IBM as part of the assessment of director independence. The Directors and Corporate Governance Committee makes recommendations to the Board about the independence of non-management directors, and the Board determines whether those directors are

Analyze Text

NLU Text Analysis Demo

Natural Language Understanding includes a set of text analytics features that you can use to extract meaning from unstructured data. Choose from a domain sample to apply the knowledge of unique entities and relations in a particular industry. Try your own text to analyze using the base model.

Learn More

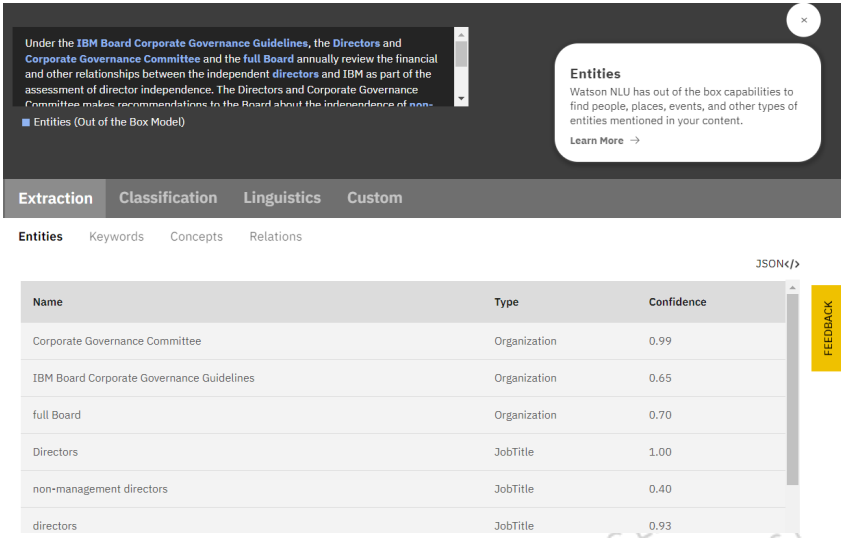
### Demostrador de Watson Natural Language Understanding

En primer lugar, deja las opciones por defecto y pulsa el botón “Analyze Text” para obtener los resultados.

La pestaña de **Extraction** → **Entities** devuelve los resultados de la extracción de entidades. Los resultados del análisis se muestran en la figura siguiente: el nombre de la entidad, su tipo semántico y la confianza del sistema en la extracción. Además, se marcan las entidades detectadas en el texto en color azul (ya que se detectan a partir del modelo estándar de detección de entidades “Out of the box Model”).

## Reconocimiento de entidades

Por ejemplo, “Corporate Governance Committee” se ha detectado como “Organization” con una confianza muy alta (de 0.99), mientras que “non-management directors” es un “JobTitle”, con una confianza más baja (de solo 0.40).



Under the IBM Board Corporate Governance Guidelines, the Directors and Corporate Governance Committee and the full Board annually review the financial and other relationships between the independent directors and IBM as part of the assessment of director independence. The Directors and Corporate Governance Committee makes recommendations to the Board about the independence of non-management directors.

Entities (Out of the Box Model)

Entities

Watson NLU has out of the box capabilities to find people, places, events, and other types of entities mentioned in your content.

Learn More →

Extraction Classification Linguistics Custom

Entities Keywords Concepts Relations

JSON</>

Name	Type	Confidence
Corporate Governance Committee	Organization	0.99
IBM Board Corporate Governance Guidelines	Organization	0.65
full Board	Organization	0.70
Directors	JobTitle	1.00
non-management directors	JobTitle	0.40
directors	JobTitle	0.93

FEEDBACK

### Lista de entidades (entities)



Pulsando en el enlace “Learn More” del cuadro en color blanco que describe las entidades, se llega a la página de documentación de la API relativa a entidades: <https://cloud.ibm.com/apidocs/natural-language-understanding#entities>

La definición de concepto que se ha empleado en este curso, a saber: una entidad que representa un tipo o una clase de individuos, es lo que denomina IBM Watson como **keyword** (palabra clave). En este caso no se muestra el tipo semántico. Por ejemplo, “ongoing basis” o “annual assessment of director independence”.

Keyword	Relevance
Independent directors	0.944799
Corporate Governance Committee	0.852644
part of the assessment of director independence	0.814854
IBM Board Corporate Governance Guidelines	0.80829
full Board	0.695587
ongoing basis	0.572888

### Lista de conceptos (keywords)

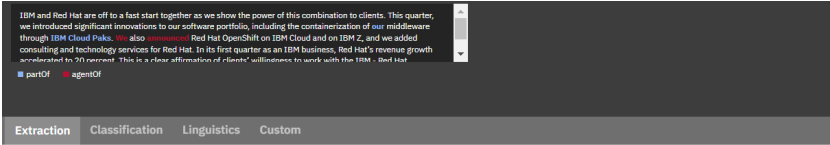
La pestaña **Concepts** muestra unos resultados que corresponden a los temas de los que trata el texto, como “Management” (gestión/dirección) o “Executive compensation” (remuneración de directivos). No corresponde a la definición de concepto que se ha utilizado en este curso, ya que esa forma no aparece como tal en el texto, sino que se extrae con técnicas más parecidas a la clasificación automática que se estudiarán en el tema siguiente.

## Reconocimiento de entidades

Concept	Score
Management	0.6681
Board of directors	0.539967
Non-executive director	0.493232
Say on pay	0.461305
Axiom of choice	0.45213
Executive compensation	0.445678

### Lista de temas (concepts)

Por último, la pestaña **Relations** muestra las relaciones extraídas en el texto. Para el contenido legal el sistema no es capaz de obtener ninguna relación. Prueba con el ejemplo en el dominio financiero, pulsando en dominio “Financial” y luego en el botón de “Analyze Text”, para el que sí se obtienen relaciones. Por ejemplo, se muestra que “we” se identifica como el agente de “announced” con una confianza del 0.97.




Entity 1	Relation	Entity 2	Score
our	partOf	IBM Cloud Paks	0.322122
We	agentOf	announced	0.976031

### Lista de relaciones (relations)

La pestaña **Custom** es similar a **Extraction** pero muestra los resultados de la extracción empleando un modelo personalizado de extracción de entidades, funcionalidad incluida en Watson Knowledge Studio, que permite ampliar las entidades mencionadas en el texto para que se adapten a cualquier dominio. Por ejemplo, para el texto legal, detecta una relación entre “directors” e “IBM” como un “JobTitle” relacionado con una “Company”.

La pestaña **Linguistics** muestra información adicional de tipo lingüístico, en concreto, analiza las frases del texto en forma de sujeto-acción-objeto e identifica las entidades y palabras clave que son sujetos u objetos de una acción. Por ejemplo, se encuentra que “the Board” (sujeto) es quien “determines” (verbo o acción) “whether those directors are independent” (objeto).



Subject	Action	Object Form
the Directors and Corporate Governance Committee and the full Board	review	the financial and other relationships between the independent directors and IBM
The Directors and Corporate Governance Committee	makes	recommendations
the Board	determines	whether those directors are independent

### Lista de triplas sujeto-acción-objeto (“semantic roles”)

Adicionalmente, la pestaña **Linguistics** en su segunda opción **Syntax** muestra el análisis morfológico (Part-Of-Speech) y el lema de cada palabra (*token*) del texto.

Token	Part of Speech	Lemma
Under	adposition	under
the	determiner	the
IBM	proper noun	
Board	proper noun	Board
Corporate	proper noun	Corporate
Governance	proper noun	Governance

### Análisis morfológico

En todos los casos, pulsando sobre el botón **JSON</>** se obtiene la salida de la API en **formato JSON**, que sería la forma más adecuada para integrar los resultados en un sistema externo. Por ejemplo, para los resultados de la pestaña de extracción, la salida se muestra en la figura siguiente.

```
{
  "entities": [
    {
      "type": "Organization",
      "text": "Corporate Governance Committee",
      "sentiment": {
        "score": 0.735252,
        "label": "positive"
      },
      "relevance": 0.959533,
      "emotion": {
        "sadness": 0.017299,
        "joy": 0.324057,
        "fear": 0.037414,
        "disgust": 0.043656,
        "anger": 0.10529
      },
      "count": 3,
      "confidence": 0.986549
    },
    {
      "type": "Organization",
      "text": "IBM Board Corporate Governance Guidelines",
      "sentiment": {
        "score": 0.660389,
        "label": "positive"
      },
      "relevance": 0.668984,
      "emotion": {
        "sadness": 0.053519,
        "joy": 0.157634,
        "fear": 0.024356,
        "disgust": 0.055615,
        "anger": 0.065261
      },
      "count": 1,
      "confidence": 0.650536
    }
  ]
}
```

### Salida en JSON

La pestaña de **Classification** muestra los resultados de la clasificación automática y el análisis de sentimiento, que obviaremos por el momento ya que se estudiarán en futuras unidades del curso.



Prueba con otros textos, por ejemplo, copiados de una noticia en un medio de comunicación, y analiza los resultados obtenidos.

### 3. Google Cloud Natural Language



La página de la plataforma es: <https://cloud.google.com/natural-language>

Cloud Natural Language

Contactar Empezar gratis

**NATURAL LANGUAGE**

- Información general
- Demostración de la API de Natural Language
- AutoML Natural Language
- Ventajas
- Características
- Cientes
- Precios
- Recursos
- Primeros pasos

**Natural Language**

Extrae información valiosa de textos sin estructurar con el aprendizaje automático de Google

Probar gratis

**Análisis de textos para extraer información valiosa**

Natural Language utiliza el aprendizaje automático para mostrar la estructura y el significado de los textos. Puedes extraer información sobre personas, lugares o eventos, así como comprender mejor las opiniones en las redes sociales y las conversaciones de los clientes. Esta herramienta te permite analizar textos e integrarlos en tu almacenamiento de documentos de Cloud Storage.

**AutoML Natural Language**

Gracias a la tecnología de AutoML, puedes entrenar tus propios modelos de aprendizaje automático

**API de Natural Language**

Los potentes modelos entrenados previamente de la API de Natural Language permiten a los

**API Natural Language de Healthcare**

Realiza análisis en tiempo real de los datos valiosos almacenados en textos

#### Google Cloud Natural Language

Más abajo en la página aparece la funcionalidad de demostración, según se ve en la figura siguiente. El demostrador reconoce automáticamente un gran número de idiomas (incluido el español), por lo que simplemente hay que escribir el texto y pulsar el botón “Analyze”.

### Demostración de la API de Natural Language

Try the API

Google, headquartered in Mountain View (1600 Amphitheatre Pkwy, Mountain View, CA 940430), unveiled the new Android phone for \$799 at the Consumer Electronic Show. Sundar Pichai said in his keynote that users love their new Android phones.

See supported languages

ANALYZE

#### Demostrador de Natural Language

Los resultados con el texto de ejemplo se muestran en la figura siguiente. La pestaña **Entities** muestra en concreto los resultados de la extracción de entidades y conceptos. El sistema es capaz de identificar la forma (por ejemplo, “Android”), su tipo semántico (“Consumer Good”), su relevancia en el texto (*salience*, 0.14) y, en algunos casos como este, lo acompaña de su enlace a la página de Wikipedia, lo que denominamos *entity linking* y, en concreto, *wikification*. Observa que el sistema es capaz de detectar un gran número de tipos semánticos.

2000

ual

**tax** muestra el análisis morfológico y el análisis sintáctico de dependencias.

análisis morfosintáctico completo  
co de dependencias.



### Análisis morfosintáctico

Las pestañas de **Sentiment** y **Categories** muestran los resultados de la clasificación automática y el análisis de sentimiento, que estudiaremos en la siguiente unidad del curso.



Prueba con otros textos y analiza los resultados obtenidos.

## Ejercicio 2: Funcionalidades para la tarea de reconocimiento de entidades

Duración estimada del ejercicio



**40**  
minutos

### 1. Introducción



El objetivo de este ejercicio es evaluar diversas funcionalidades para la tarea de reconocimiento de entidades proporcionadas por diferentes paquetes software en lenguaje de programación Python.



No hay que preocuparse, el ejercicio va a ser totalmente guiado y no es necesario saber Python. El propósito del ejercicio es hacerse una idea de cómo se abordan estos problemas desde el punto de vista de la solución técnica, conocer el entorno de trabajo e identificar las tareas donde el lingüista puede desempeñar un papel determinante.

### 2. Google Colaboratory

No es necesario instalar nada. Como entorno de trabajo, vamos a emplear una solución gratuita llamada Google Colaboratory (o "Colab" a secas). Lo único que necesitas es una cuenta de correo de Gmail o de Google.

El entorno de Colab permite ejecutar y programar en Python directamente en tu navegador, de forma totalmente gratuita, sin requerir ningún tipo de instalación ni de configuración.

El trabajo en Colab se basa en el desarrollo de "notebooks" (el nombre completo es Jupyter Notebooks, antes iPython Notebooks), que son un tipo de documento especial que mezcla secciones de código en Python (que se pueden ejecutar de forma interactiva) con secciones de descripciones de texto (que se usan para escribir documentación sobre el código) y secciones de resultados de la ejecución del código.

Los *notebooks* se utilizan masivamente en tareas de análisis de datos porque unifican la documentación, el código de análisis de datos y los resultados de su ejecución en un mismo documento, permitiendo la reutilización y difusión del conocimiento.





Para entrar en Colab solo hay que abrir la siguiente página con un navegador: <https://colab.research.google.com/>

Verás la figura mostrada en la página siguiente. Para poder interactuar con la página es necesario identificarse con una cuenta de Gmail o Google.



### Google Colaboratory

Fuente de la imagen: [Colab.research.google.com](https://colab.research.google.com)

Puedes observar diferentes secciones (llamadas “celdas”) con texto formateado con títulos, negritas, enumeraciones, etc., que se utilizan para describir el código o indicar instrucciones de ejecución. Haciendo clic con el ratón sobre el texto de cualquiera de ellas, aparece un menú con botones que permiten editar el contenido, moverlo arriba o abajo en el documento, y copiar o borrar la celda.

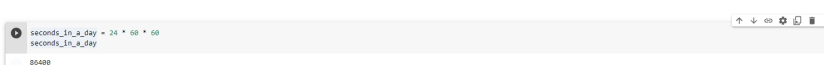
Prueba a modificar el texto que contiene pulsando en el icono del lápiz, o mover la celda arriba o abajo usando los iconos de las flechas.



### Celda de texto

Por otra parte, hay celdas de código Python. Tienen iconos similares a las celdas de texto, se puede escribir en lenguaje Python directamente en la celda, y además tiene un icono especial en la parte izquierda (como el icono “Reproducir” de reproductor de vídeo) para ejecutar el código contenido en la celda. Como se utiliza continuamente, además de pulsar en el icono es posible usar también la combinación de teclas Control+Enter. Los resultados de la ejecución del código se muestran en la parte inferior.

Prueba a ejecutar la celda donde se calcula el número de segundos que tiene un día, obteniendo como resultado el valor de 86.400 segundos.



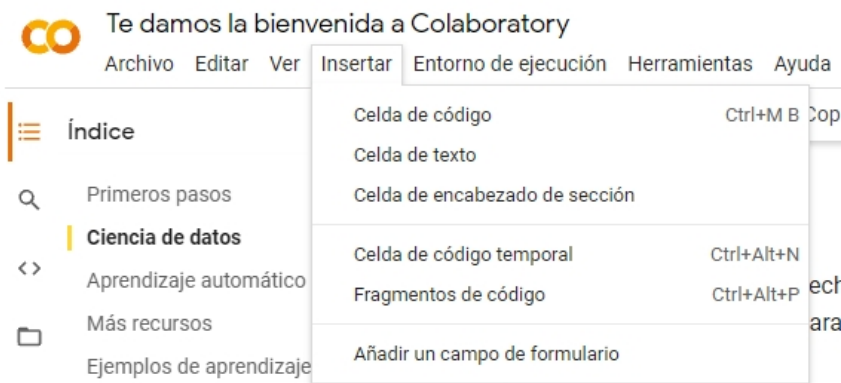
### Celda de código

El resultado de ejecución puede ser texto, como en este caso, o puede mostrar tablas o elementos gráficos como se muestra en la figura siguiente:



*Resultado de ejecución con un gráfico*

En la barra de menú superior, la opción “Insertar” permite añadir nuevas celdas de código o texto. Prueba a añadir alguna celda más.



*Insertar nuevas celdas*

Otra manera cómoda de añadir código o texto es colocar el cursor del ratón en la parte inferior de la celda.



*Insertar nuevas celdas (alternativa)*

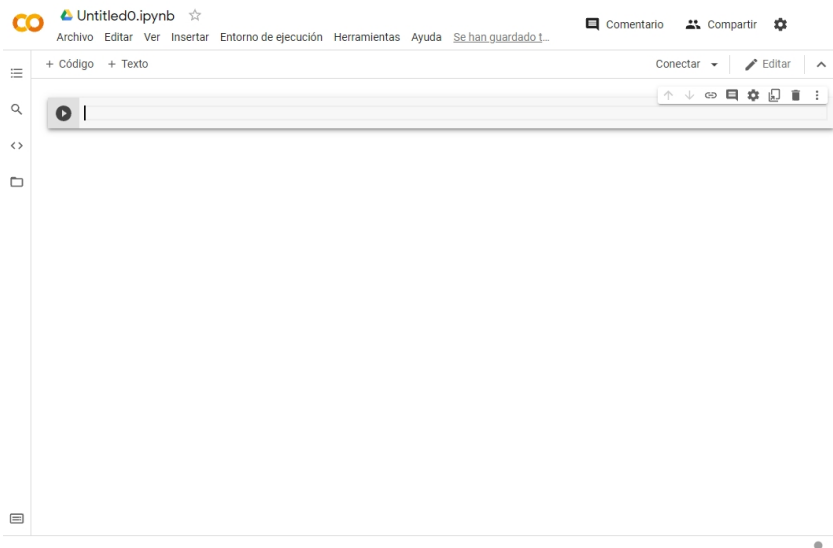
En el resto del ejercicio, lo que vas a hacer paso a paso es abrir un “notebook” nuevo, añadir celdas de código, donde irás copiando el código Python indicado, y ejecutarlas para analizar los resultados.

### 3. Creación de un cuaderno nuevo

El primer paso es crear un cuaderno nuevo para trabajar en este ejercicio. Para ello, en la página de Colab, selecciona en el menú “Archivo” la opción “Nuevo cuaderno”.

Si no te has identificado con una cuenta de Google, te pedirá que lo hagas en este momento.

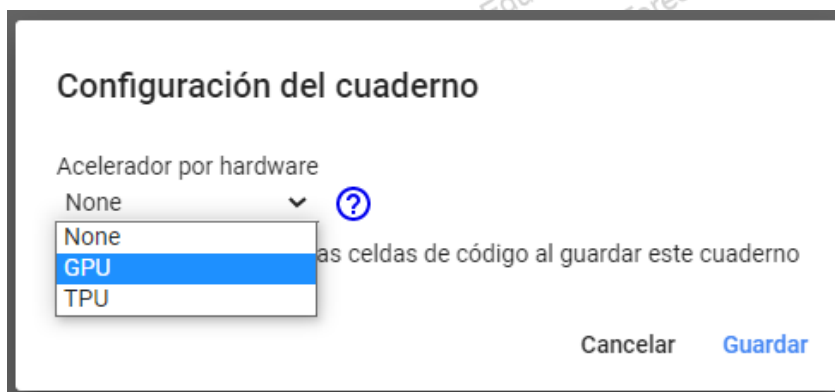
Se abrirá una pestaña nueva del navegador, como la siguiente figura, mostrando un cuaderno vacío. Su título por defecto es "Untitled0.ipynb". Pulsando sobre el nombre, cámbialo por ejemplo a "Ejercicio 2.ipynb" (mantén mejor la extensión "ipynb" para indicar que es un "iPython Notebook").



*Nuevo cuaderno (vacío)*

En la unidad 1, ya hablamos de las GPU, procesadores de alta potencia de cálculo, que resultan esenciales para realizar las operaciones matemáticas necesarias para los algoritmos de *deep learning* a una velocidad razonable. Colab ofrece la posibilidad de disponer de un entorno de ejecución basado en CPU (el procesador habitual) o bien un entorno con GPU o con TPU (una versión todavía más mejorada).

Como vamos a emplear técnicas de *deep learning*, necesitamos seleccionar el entorno acelerado con GPU. Para ello, en la opción "Entorno de ejecución" del menú superior, selecciona "Cambiar tipo de entorno de ejecución" y luego, en el desplegable, selecciona "GPU", como se muestra en la figura siguiente, y pulsa "Guardar".



*Configuración del tipo de entorno con GPU*

Con esto ya tienes preparado tu entorno de trabajo en Colab.



Estos son los pasos habituales para empezar a trabajar en un nuevo desarrollo con Colab, que emplearás en otros ejercicios de otras unidades de este curso.

## 4. Reconocimiento de entidades con NLTK

NLTK (Natural Language Toolkit) es una de las plataformas de código abierto más populares de programación en código Python para llevar a cabo tareas de procesamiento del lenguaje natural, ampliamente utilizada tanto en proyectos de investigación como comerciales. NLTK ha sido calificado como "una herramienta fantástica para enseñar y trabajar en lingüística computacional con Python" y "una biblioteca increíble para jugar con el lenguaje natural".



Para saber más sobre NLTK, puedes pinchar en este [enlace](#).

El primer paso es indicar a Python que añada todas las dependencias necesarias de NLTK. Crea una nueva celda de código, pega el siguiente fragmento en Python y ejecútalo. La función "pprint" (pretty print) nos permitirá imprimir por pantalla conjuntos de datos de manera más legible.



```
import nltk

nltk.download('punkt')

nltk.download('averaged_perceptron_tagger')

nltk.download('maxent_ne_chunker')

nltk.download('words')

from pprint import pprint
```

En primer lugar, hay que preprocesar el texto. NLTK incluye funciones para segmentar una frase en palabras y luego obtener el análisis morfológico (Part-Of-Speech, POS). Vamos a definir una función llamada "get\_pos" a la que se le pase la frase (en la variable "text") y devuelva este análisis (en la variable "pos"). Copia y ejecuta el código siguiente en una celda nueva:



```
def get_pos(text):

    tokens = nltk.tokenize.word_tokenize(text)

    pos = nltk.tag.pos_tag(tokens)

    return pos
```



En Python, la indentación es obligatoria para definir los bloques de sentencias. El número de espacios debe ser uniforme en un bloque de código (típicamente se usan 2 o 4 espacios). También se recomienda usar espacios frente a tabulaciones.

En el caso del ejemplo anterior, se define una función. Para ello, en Python se utiliza la instrucción “def” junto a un nombre descriptivo y los dos puntos. A continuación, el bloque de contenido debe estar correctamente indentado para delimitar qué sentencias son aquellas que forman parte de la función.

Ahora solo queda llamar a esta función con una frase. NLTK incluye modelos por defecto para inglés, por lo que vamos a emplear un ejemplo en este idioma, por ejemplo, un texto sobre Google. Copia y ejecuta el código siguiente:



**`pprint(get_pos('Google is an American multinational technology company founded in 1998 that specializes in Internet-related services and products.'))`**

El resultado es el que se muestra en la figura siguiente. Se puede observar cada palabra (*token*) con su análisis morfológico, usando como representación las etiquetas del proyecto Penn TreeBank (por ejemplo, N indica nombre, J es un adjetivo, V es un verbo, etc.).

```
[('Google', 'NNP'),
 ('is', 'VBZ'),
 ('an', 'DT'),
 ('American', 'JJ'),
 ('multinational', 'NN'),
 ('technology', 'NN'),
 ('company', 'NN'),
 ('founded', 'VBD'),
 ('in', 'IN'),
 ('1998', 'CD'),
 ('that', 'WDT'),
 ('specializes', 'VBZ'),
 ('in', 'IN'),
 ('Internet-related', 'NNP'),
 ('services', 'NNS'),
 ('and', 'CC'),
 ('products', 'NNS'),
 ('.', '.')]

```

#### Resultados del etiquetado morfológico



Para saber más sobre Penn TreeBank, puedes pinchar en este [enlace](#).

NLTK también incluye funcionalidad para la detección de entidades accesible a través de la función “ne\_chunk”. Copia y ejecuta el código siguiente en una celda nueva:



```
print(nltk.ne_chunk(get_pos('Google is an American multinational technology company
founded in 1998 that specializes in Internet-related services and products.')))
```

La salida se muestra en la figura siguiente. Se han detectado dos GPE ("geopolitical entities"), "Google" (incorrectamente) y "American" (como gentilicio de EE. UU.).

```
(S
  (GPE Google/NNP)
  is/VBZ
  an/DT
  (GPE American/JJ)
  multinational/NN
  technology/NN
  company/NN
  founded/VBD
  in/IN
  1998/CD
  that/WD
  specializes/VBZ
  in/IN
  Internet-related/NNP
  services/NNS
  and/CC
  products/NNS
  ./.)
```

#### Resultado del reconocimiento de entidades

NLTK ofrece también capacidades de análisis sintáctico, entre otras, con reglas basadas en patrones de expresiones regulares. Por ejemplo, el siguiente código procesa el árbol morfológico para detectar grupos nominales ("NP") definidos por grupos de *tokens* que contienen nombres, adjetivos, determinantes o conjunciones y acaban en nombre. Esta regla es lógicamente muy sencilla, pero ilustra un mecanismo simple para realizar un análisis sintáctico superficial.



```
def parse_groups(pos):
    pattern = 'NP: {(<N.+>|<DT>|<JJ>|<CC>)*<N.+>}'
    return nltk.RegexpParser(pattern).parse(pos)

print(parse_groups(get_pos('Google is an American multinational
technology company founded in 1998 that specializes in
Internet-related services and products.')))
```

Pincha aquí para acceder al código

```
def parse_groups(pos):
```

```
    pattern = 'NP: {(N.+>|<DT>|<JJ>|<CC>)*<N.+>}'
    return nltk.RegexpParser(pattern).parse(pos)
```

```
print(parse_groups(get_pos('Google is an American multinational technology company founded in
1998 that specializes in Internet-related services and products.')))
```

La salida se muestra en la figura siguiente, donde se ven que se detectan tres grupos nominales: “Google”, “an American multinational technology company” y “Internet-related services and products”.

```
(S
  (NP Google/NNP)
  is/VBZ
  (NP an/DT American/JJ multinational/NN technology/NN company/NN)
  founded/VBD
  in/IN
  1998/CD
  that/WD
  specializes/VBZ
  in/IN
  (NP Internet-related/NNP services/NNS and/CC products/NNS)
  ./.)
```

#### Resultado del análisis sintáctico superficial

En los materiales de esta unidad se ha descrito la notación IOB para representar la información semántica de un texto. El siguiente código genera la salida en notación IOB correspondiente del árbol sintáctico anterior:



```
pprint(nltk.chunk.tree2conlltags(parse_groups(get_pos('Google is an American
multinational technology company founded in 1998 that specializes in Internet-related
services and products.'))))
```

```
[('Google', 'NNP', 'B-NP'),
 ('is', 'VBZ', 'O'),
 ('an', 'DT', 'B-NP'),
 ('American', 'JJ', 'I-NP'),
 ('multinational', 'NN', 'I-NP'),
 ('technology', 'NN', 'I-NP'),
 ('company', 'NN', 'I-NP'),
 ('founded', 'VBD', 'O'),
 ('in', 'IN', 'O'),
 ('1998', 'CD', 'O'),
 ('that', 'WDT', 'O'),
 ('specializes', 'VBZ', 'O'),
 ('in', 'IN', 'O'),
 ('Internet-related', 'NNP', 'B-NP'),
 ('services', 'NNS', 'I-NP'),
 ('and', 'CC', 'I-NP'),
 ('products', 'NNS', 'I-NP'),
 ('.', '.', 'O')]
```

#### Etiquetado en notación IOB

Como se ve en la figura, los *tokens* iniciales de cada grupo se marcan con el prefijo “B-” y el resto de *tokens* con el prefijo “I-”. Los *tokens* que no forman parte de un grupo, reciben la etiqueta “O”.

Una única frase no sirve de mucho, pero etiquetando un corpus completo con esta salida, se tendría una primera versión de corpus anotado, que probablemente los lingüistas tendrían que revisar manualmente para corregir los errores, para entrenar un sistema de reconocimiento, en este caso, de grupos nominales.

## 5. Reconocimiento de entidades con spaCy

spaCy es otra biblioteca gratuita de código abierto para el procesamiento del lenguaje natural en Python, enormemente popular, que ofrece funcionalidad para multitud de tareas de procesamiento del lenguaje natural.



Para saber más sobre spaCy, puedes pinchar en este [enlace](#).

Primero es necesario instalar las dependencias en nuestro sistema.



```
import spacy
```

```
import en_core_web_sm
```

```
# Cargar segmentador, analizador y NER en inglés
```

```
nlp = en_core_web_sm.load()
```

Para ejecutar la tarea de reconocimiento de entidades, es tan sencillo como ejecutar el siguiente código:





```
doc = nlp('Google is an American multinational technology company founded in 1998
that specializes in Internet-related services and products.')
```

```
pprint([(X, X.ent_iob_, X.ent_type_) for X in doc])
```

```
[(Google, 'B', 'ORG'),
 (is, 'O', ''),
 (an, 'O', ''),
 (American, 'B', 'NORP'),
 (multinational, 'O', ''),
 (technology, 'O', ''),
 (company, 'O', ''),
 (founded, 'O', ''),
 (in, 'O', ''),
 (1998, 'B', 'DATE'),
 (that, 'O', ''),
 (specializes, 'O', ''),
 (in, 'O', ''),
 (Internet, 'O', ''),
 (-, 'O', ''),
 (related, 'O', ''),
 (services, 'O', ''),
 (and, 'O', ''),
 (products, 'O', ''),
 (., 'O', '')]

```

#### Resultado del reconocimiento de entidades

La figura muestra los resultados del proceso, donde se reconocen correctamente “Google” como ORG (“organization”), “American” como NORP (“nationalities or religious or political groups”) y “1998” como DATE (fecha). En este caso, como todos son unidades de un único *token*, todos ellos reciben la marca “B” (begin) en notación IOB.

Por ejemplo, prueba a ejecutar cambiando el texto por “Google LLC”.

Para obtener una representación visual del texto, se puede utilizar el siguiente código:



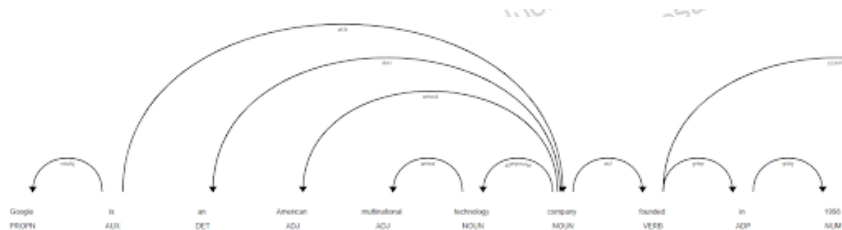
```
spacy.displacy.render(nlp('Google is an American multinational technology company
founded in 1998 that specializes in Internet-related services and products.'),
jupyter=True, style='ent')
```

Google ORG is an American NORP multinational technology company founded in 1998 DATE  
that specializes in Internet-related services and products.

#### Representación visual de las entidades detectadas

También es posible obtener el árbol de dependencias sintácticas, cambiando en el código anterior el valor de “style”: si en vez de “ent”, se pone “dep”, se obtiene como resultado la figura siguiente:

## Reconocimiento de entidades



### Representación visual de las dependencias sintácticas

Puedes probar con otros textos para comprobar si los resultados son correctos.

Por último, spaCy también incluye modelos para español. Primero, ejecuta el siguiente código en una celda de código:



```
!python -m spacy download es_core_news_sm
```

Y luego, en una celda diferente, copia y ejecuta el siguiente código:



```
import es_core_news_sm  
nlp = es_core_news_sm.load()
```

```
spacy.displacy.render(nlp('Google LLC es una compañía principal subsidiaria de la  
multinacional estadounidense Alphabet Inc., cuya especialización son los productos y  
servicios relacionados con Internet, software, dispositivos electrónicos y otras  
tecnologías.'), jupyter=True, style='ent')
```



Prueba con diferentes textos y analiza los resultados.

# Recursos

## Enlaces de Interés



### **Sekine's Extended Named Entity Hierarchy**

<https://nlp.cs.nyu.edu/ene/>



### **Inside-outside-beginning**

[https://en.wikipedia.org/wiki/Inside\\_Outside\\_Beginning](https://en.wikipedia.org/wiki/Inside_Outside_Beginning)



### **Entity linking**

[https://en.wikipedia.org/wiki/Entity\\_linking](https://en.wikipedia.org/wiki/Entity_linking)



### **SemEval (International Workshop on Semantic Evaluation)**

<https://www.cs.york.ac.uk/semeval-2013/>



### **Hidden Markov Model**

<https://medium.com/@kangeugine/hidden-markov-model-7681c22f5b9>



### **Modelo oculto de Márkov**

[https://es.wikipedia.org/wiki/Modelo\\_oculto\\_de\\_M%C3%A1rkov](https://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov)



### **What is a Decision Tree?**

<https://www.displayr.com/what-is-a-decision-tree/>



### **Improving Support Vector Machine with Intel® Data Analytics Acceleration Library**

<https://software.intel.com/content/www/us/en/develop/articles/improving-svm-performance-with-intel-daal.html>



### **Máquinas de vectores de soporte**

[https://es.wikipedia.org/wiki/M%C3%A1quinas\\_de\\_vectores\\_de\\_soporte](https://es.wikipedia.org/wiki/M%C3%A1quinas_de_vectores_de_soporte)



### **Conditional Random Fields (CRF): Short Survey**

<https://www.datasciencecentral.com/profiles/blogs/conditional-random-fields-crf-short-survey>



### **Campo aleatorio condicional**

[https://es.wikipedia.org/wiki/Campo\\_aleatorio\\_condicional](https://es.wikipedia.org/wiki/Campo_aleatorio_condicional)



### **Generación de regla de reconocimiento de entidades**

[https://www.researchgate.net/figure/An-example-of-rule-generation\\_fig3\\_323409576](https://www.researchgate.net/figure/An-example-of-rule-generation_fig3_323409576)



### **Applying Unsupervised Machine Learning to Sequence Labeling**

<https://medium.com/mosaix/deep-text-representation-for-sequence-labeling-2f2e605ed9d>



### **Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature**

[https://www.researchgate.net/publication/333638498\\_Named\\_Entity\\_Recognition\\_and\\_Normalization\\_Applied\\_to\\_Large-Scale\\_Information\\_Extraction\\_from\\_the\\_Materials\\_Science\\_Literature](https://www.researchgate.net/publication/333638498_Named_Entity_Recognition_and_Normalization_Applied_to_Large-Scale_Information_Extraction_from_the_Materials_Science_Literature)



### **Doccano**

<https://github.com/doccano/doccano>



### **Brat**

<https://brat.nlplab.org/>



### **Cuaderno de Colab**

<https://colab.research.google.com/>



**Colab.research.google.com**  
<https://colab.research.google.com/>



**NLTK**  
<https://www.nltk.org/>



**Penn TreeBank**  
[https://repository.upenn.edu/cgi/viewcontent.cgi?article=1603&context=cis\\_reports](https://repository.upenn.edu/cgi/viewcontent.cgi?article=1603&context=cis_reports)



**spaCy**  
<https://spacy.io/>



**IBM Watson Natural Language Understanding**  
<https://www.ibm.com/cloud/watson-natural-language-understanding>



**Demostrador IBM Watson Natural Language Understanding**  
<https://www.ibm.com/demos/live/natural-language-understanding/self-service/home>



**Documentación de la API IBM Watson Natural Language Understanding**  
<https://cloud.ibm.com/apidocs/natural-language-understanding#entities>



**Google Cloud Natural Language**  
<https://cloud.google.com/natural-language>