Máster Universitario en Tecnologías del Lenguaje

Minería de Textos

Conceptos preliminares

La **minería de textos** es el proceso de analizar colecciones de textos para descubrir información y patrones que no aparecen de forma explícita en los textos.

Engloba tareas como la **extracción de información**, la **recuperación de información**, la **categorización** y **agrupamiento de documentos**.

En minería de textos se utilizan principalmente técnicas de **procesamiento del lenguaje natural** y de **aprendizaje automático.**

Conceptos preliminares

La **minería de textos** se aplica a colecciones de documentos con información textual no estructurada y escrita en lenguaje natural.

La información textual normalmente conforma documentos que pueden agruparse en colecciones o corpus.

Las colecciones de documentos o corpora pueden contener anotaciones y metadatos

Conceptos preliminares

La **extracción de información** consiste en extraer automáticamente información estructurada a partir de textos. Abarca procesos como el reconocimiento de nombre de entidades (NER: Named Entity recognition),
la extracción de terminología y de relaciones.

Conceptos preliminares

La **recuperación de información** (Information Retrieval) consiste en buscar documentos, buscar información dentro de los documentos y en buscar metadatos que describan los documentos. Incluye la búsqueda en todo tipo de repositorios y bases de datos, tanto aisladas como conectadas en red y con hipertexto, como World Wide Web.

Conceptos preliminares

La **categorización de documentos** consiste en asignar a un documento una o más categorías en función de su contenido. Las categorías con las que se hace la clasificación están definidas previamente.

El **agrupamiento de documentos** (Document Clustering), es una forma de organización de documentos en grupos en la que ni la naturaleza de los grupos, ni en ocasiones su número están definidos de antemano.

Conceptos preliminares

El **procesamiento del lenguaje natural** (PLN) es la rama de la informática cuyo objetivo es el desarrollo de sistemas que permitan a los ordenadores comunicarse con personas utilizando el lenguaje humano.

Procesamiento del lenguaje natural

- PLN tiene muchas aplicaciones.
- PLN se utiliza para adquirir conocimientos a partir de cantidades masivas de datos textuales
  - Por ejemplo, comprobación de hipótesis a partir de informes médicos, generación de resúmenes, sistemas de diálogos, etc.
- PLN presenta diversas dificultades!

Procesamiento del lenguaje natural

Qué hace difícil el PLN?

- Alta ambigüedad a todos los niveles: Léxico, Sintáctico, Semántico, de discursos
  - Estuvo esperando en el banco durante dos horas
    - ¿banco de sentarse o banco de dinero?
  - No los aceptaron en el partido por sus prejuicios
    - ¿los prejuicios son de ellos o del partido?
  - Los niños eligieron juguetes alegres
    - ¿quien o qué era alegre?
  - Etc..

Procesamiento del lenguaje natural

Qué hace difícil el PLN?

- El lenguaje natural implica conocimiento del mundo:
  - Estuve esperando una hora en el banco para sacar dinero: el conocimiento del mundo nos permite saber que “banco” es una institución bancaria, y no otro tipo de banco

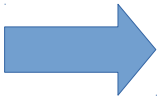
Procesamiento del lenguaje natural

Qué hace difícil el PLN?

- Hay muchos aspectos del lenguaje que intervienen en la interpretación correcta:
  - Negación: al extraer la información es importante saber si un dato, concepto, relación etc. está siendo negado.
  - Especulación: Muchas veces se dice que “puede” darse un hecho.
  - Correferencia: es necesario identificar expresiones que hacen referencia a la misma entidad:
    - Aunque todos vieron al *Presidente*, ninguno reconoció al *ganador de las últimas elecciones*

Evaluación de tareas de PLN

- Obtener datos de evaluación de los modelos y sistemas utilizados es fundamental para el avance del área.
- Es necesario comparar modelos y sistemas de formas justa:
  - Sobre los mismos datos: corpus o colecciones comunes
  - Usando las mismas medidas



Campañas de evaluación

Evaluación de tareas de PLN

- Un corpus es una compilación de textos en formato electrónico.
- Son un recurso fundamental en el desarrollo de aplicaciones basadas en PLN:
  - Permiten evaluar los sistemas
  - Proporcionan un marco común para comparar técnicas alternativas
  - Permiten entrenar los sistemas de aprendizaje automático supervisados: ajustan sus parámetros a partir de una parte del corpus usada para entrenamiento.

Evaluación de tareas de PLN

- Proporcionan una muestra amplia y natural del lenguaje.
- Son una fuente de datos estadísticos sobre las palabras, sus relaciones y sus construcciones.
- Algunas de sus aplicaciones son:
  - Asignación automática de categorías léxicas,
  - Desambiguación sintáctica
  - Extracción de gramáticas
  - Traducción automática
  - Extracción de entidades (nombres, medicamentos, etc.)
  - Extracción de relaciones

Evaluación de tareas de PLN

- Tipos generales de corpus:
  - Sin anotar texto puro
  - Anotados marcados con distintos tipos de información lingüística (se les denomina Gold standard y se usan de referencia)
- Monolingüe: textos pertenecientes a una única lengua
- Multilingüe: textos en diversas lenguas. Proporcionan anotaciones similares en distintas lenguas.
- Pueden ser
  - Corpus paralelo: contiene los mismos textos en más de una lengua,
  - Corpus comparable: Contiene un número equilibrado de textos en distintas lenguas. También están equilibrados respecto al tema y tipo de textos.

Evaluación de tareas de PLN

- Ejemplos de corpus:
  - Corpus de Brown (http://www.nltk.org/nltk_data)
    - Contiene más de un millón de palabras
    - Anotado con categorías léxicas
    - Consta de archivos de 15 temáticas
    - Considera un conjunto de 87 etiquetas:
      - AT artículo
      - BE verbo “to be” en infinitivo o imperativo
      - CC conjunción copulativa
      - DT determinante/pronombre, etc.

Evaluación de tareas de PLN

- Corpus de Brown: fragmento

<s> The/AT Fulton/NP County/NP Grand/NP Jury/NP said/VBD Friday/NR an/AT investigation/NN of/IN Atlanta/NP 's/\$ recent/JJ primary/NN election/ NN produced/VBD “/” no/AT evidence/NN "/" that/CS any/DTI irregularities/NNS took/VBD place/NN ./.
<s> The/AT jury/NN further/RBR said/VBD in/IN term-end/NN presentments/ NNS that/CS the/AT City/NP Executive/NP Committee/NP ,/, which/WDT had/HVD over-all/JJ charge/NN of/IN the/AT election/NN ,/, ”/” deserves/VBZ the/AT praise/NN and/CC thanks/NNS of/IN the/AT City/NP of/IN Atlanta/NP "/" for/IN the/AT manner/NN in/IN which/WDT the/AT election/NN was/BEDZ conducted/VBN ./.

Evaluación de tareas de PLN

- Corpus de Susanne:
- Contiene 130.000 palabras
- Formado por un subconjunto de 64 archivos del corpus de Brown
- Anotado con categorías léxicas y análisis sintáctico
- Disponible en la dirección http://www.grsampson.net/Resources.html

Evaluación de tareas de PLN

- Corpus de Susanne:
- Por cada palabra de texto hay una línea de anotaciones. Cada línea consta de:
  - referencia (nombre de archivo, línea, posición en la línea)
  - estado (indicativo de abreviatura o símbolo)
  - categoría gramatical
  - Palabra
  - Lema
  - análisis sintáctico

Evaluación de tareas de PLN

- Corpus de Susanne: fragmento

Ref.	*est.*	*cat.*	*palabra*	*lema*	*análisis*
N06:0180.12	-	NN1u	Baldness	baldness	[S[Ns:s.Ns:s]
N06:0180.15	-	VBDZ	was	be	[Vsu.
N06:0180.18	-	VVGt	attacking	attack	.Vsu]
N06:0180.21	-	APPGm	his	his	[Ns:o.
N06:0180.24	-	NN1c	pate	pate	.Ns:o]S]

Evaluación de tareas de PLN

- Ejemplos de corpus:
  - Corpus ADE (relaciones entre efectos adversos y medicamentos)

Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, Luca Toldo:

Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. Journal of Biomedical Informatics 45(5): 885-892 (2012)

https://github.com/trunghlt/AdverseDrugReaction

Evaluación de tareas de PLN

- Bioscope: textos biomédicos anotados con negación y especulación

The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik:,

BMC Bioinformatics 2008, 9.

http://rgai.inf.u-szeged.hu/index.php?lang=en&page=bioscope

Evaluación de tareas de PLN

- Corpus DDI: anotación de medicamentos e interacciones entre ellos.

The DDI corpus: an annotated corpus with pharmacological substances and drug-drug interactions.

Herrero-Zazo M, Segura-Bedmar I, Martínez P, Declerck T.

J Biomed Inform. 46(5):914-920 (2013)

https://omictools.com/ddi-corpus-tool

Evaluación de tareas de PLN

- Ejemplos de corpus:
  - Corpus Europarl (Actas del parlamento europeo en 21 lenguas de Europa: Español, Inglés, Francés, Griego, …, con alineamiento o correspondencia a nivel de oración)

Europarl: A Parallel Corpus for Statistical Machine Translation, Philipp Koehn, MT Summit 2005,

http://www.statmt.org/europarl/

Evaluación de tareas de PLN

- Ejemplos de corpus:
  - Corpus DIANN: discapacidades anotadas

Overview of the DIANN Task: Disability Annotation Task. Hermenegildo Fabregat, Juan Martinez-Romo, Lourdes Araujo. IberEval@SEPLN 2018: 1-14

https://github.com/gildofabregat/

DIANN-IBEREVAL-2018/tree/master/
DIANN_CORPUS

Evaluación de tareas de PLN

- Campañas de evaluación: proponen retos a los participantes, proporcionando datos y un marco de evaluación común y bien definido.
- TREC: Text Retrieval conference
- CLEF: Conference and Labs of the evaluation Forum
- Semeval: Semantic Evaluation
- Ibereval: Lenguas Ibéricas

Evaluación de tareas de PLN

- Medidas de evaluación más usuales:
  - Precisión: fracción de predicciones del modelo propuesto acertadas (coinciden con los datos de referencia):

$$p = \frac{tp}{tp + fp}$$

siendo tp: true prositive, fp: false positive

  - Cobertura o exhaustividad (recall): fracción de los datos de referencia que han sido propuestas por el modelo evaluado:

$$r = \frac{tp}{tp + fn}$$

siendo fn: false negative

Evaluación de tareas de PLN

Medida-F: media armónica de precisión y cobertura:

$$Medida - F = \frac{2.\ p\ .r}{p+r}$$

cuando p y r se ponderan igual.

- Medida F_β : forma general de la media armónica que permite dar más importancia a la precisión o a la cobertura en función del parámetro β:

$$Medida - F_\beta = (1+\beta^2)\frac{p\ .r}{\beta^2\ .\ p+r}$$

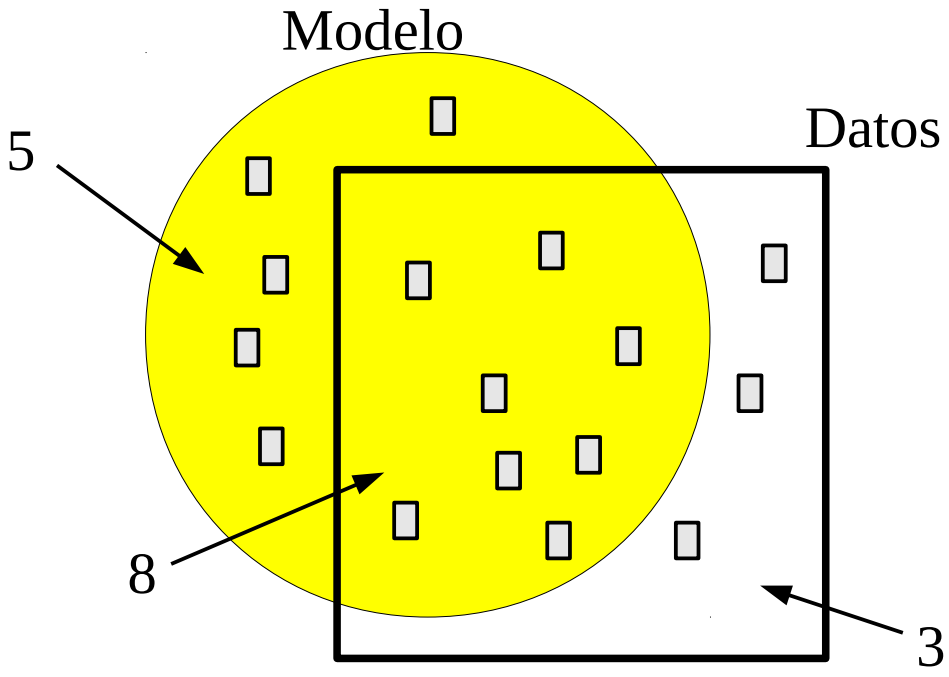
- También se usan otras medidas específicas de tareas de PLN concretas.

Medidas

Media-F: $F_1 = \frac{2\,p\,r}{p+r}$

$p \rightarrow$ precision

$r \rightarrow$ recall



$p = \frac{8}{13}$ $r = \frac{8}{11}$

Baseline

- En una tarea es importante tener un referencia de que valores de las medidas pueden considerarse aceptables.
- El rango de valores cambia mucho de una tarea a otra.
- Los resultados de otras propuestas son una referencia. Pero no siempre se dispone de ellos cuando aparece una nueva tarea.
- Baseline para una tarea: medida de los resultados de un modelo básico que resuelve la tarea.

Baseline

- Ejemplo:
  - Tarea: asignar el significado correcto a una palabra ambigua en el contexto de un documento.
  - Baseline: asignar el significado más frecuente de la palabra.
  - Se espera que cualquier otro modelo propuesto sea capaz de superar el baseline.

Análisis de errores

- Crucial de cualquier aplicación de PLN
- Puede ayudar a encontrar errores en los programas, problemas en los datos de entrenamiento y, también es fundamental para desarrollar nuevos modelos de conocimiento o algoritmos para usar en la resolución de problemas.
- Las matrices de confusión o tablas de contingencia se utilizan para analizar los errores de los clasificadores.

Análisis de errores

- Una matriz de confusión para una tarea de clasificación de N clases es una matriz de NxN donde la celda(x,y) contiene el número de veces que un elemento con la clasificación correcta x fue clasificado por el modelo como y.

Análisis de errores

Valores predichos

Valores correctos	Clase	1	2	3	4	Total
	1	70	10	15	5	100
	2	8	67	20	5	100
	3	0	11	88	1	100
	4	4	10	14	72	100

- En el ejemplo, la clase 1 se ha clasificado 70 veces de forma correcta, 10 veces de forma errónea, como clase 2, etc.
- A veces la matriz de confusión contiene porcentajes.

Herramientas

- Nltk: Natural Language Toolkit
  - Librería para desarrollar programas en python para procesar datos de lenguaje natural
  - https://www.nltk.org/
  - Disponible para linux, windows y Mac OS
  - Ejemplos de como iniciarlo, y algunas de las utilidades más básicas:
    - https://www.nltk.org/book/ch01.html

Herramientas

- Spacy
  - Librería para desarrollar programas en python para procesar datos de lenguaje natural
  - https://spacy.io/
  - Disponible para linux, windows y Mac OS
  - Información de instalación y uso:
    - https://spacy.io/usage/

Herramientas

- Spacy
  - Tiene modelos preentrenados
  - En inglés:
    - https://spacy.io/models/en
    - Es necesario cargarlos antes de usarlos. Por ejemplo en Linux: python -m spacy download en
  - Y en español:
    - https://spacy.io/models/es
    - Linux: python -m spacy download es

Herramientas

- Otras herramientas
  - Stanford
  - Treetagger
  - Freeling
  - Etc.
  -

Introducción al procesamiento del Lenguaje Natural

Algunos procesos importantes para la minería de textos:

- Preprocesador de textos: Normalización, Expresiones regulares, distancia de edición
- POS tagging
- Stemming
- Chunking
- Parsing
- Desambiguación del sentido de las palabras

Expresiones regulares

- Lenguaje formal para especificar patrones de texto.
- Herramienta fundamental en muchas tareas de PLN.
- Distinguen mayúsculas y minúsculas
- Ver descripción en Capítulo 2: Speech and Language Processing. Daniel Jurafsky & James H. Martin.

Normalización de textos

- Proceso de transformación del texto para obtener una forma canónica. El objetivo es uniformizar la forma del texto para facilitar posteriores procesos.
- Tareas que se aplican normalmente como parte del proceso de normalización:
  - Normalización de la forma de las palabras
  - Segmentación o tokenización de oraciones
  - Segmentación o tokenización de palabras

Normalización de palabras

- Objetivo: unificar palabras que se refieren de distinta forma a la misma entidad (forman una clase común)
- Ejemplos:
  - “U.S.A” y “USA”
  - “Manzana” y “manzana”
  - “helado de fresa” y “helado de fresa”
  - Erratas de escritura
- Técnicas comunes:
  - Eliminar acentos, quitar espacios repetidos, reducir todas las palabras a minúsculas, etc.

Distancia de edición

- Distancia de edición mínima entre dos cadenas: Número/coste de las operaciones de edición:
  - Inserción
  - Borrado
  - Sustitución

  necesarias para transformar una cadena en la otra.
- Cada operación puede tener un coste distinto.
- Es necesario explorar todas las combinaciones de operación para seleccionar la de menor coste

Distancia de edición

- Es importante para
  - medir la similitud entre dos cadenas
    - Si se ha tecleado “graffe”

      ¿qué cadena es más próxima?
      - graf
      - graff
      - giraffe
  - Se usa
    - en la corrección a errores ortográficos
    - En la extracción de información, la traducción automática, la recuperación de información, etc.

Distancia de edición

- Ejemplo

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
d s s   i s
```

- Si cada operación tiene coste 1: la distancia es 5.
- Si la sustitución tiene coste 2, la distancia es 8

Palabras vacías

- Palabras sin significado propio: artículos, preposiciones, etc.
- Suelen ser las más frecuentes en un idioma
- En muchos procesos de PLN se eliminan para centrar el análisis en las palabras con contenido semántico
- Ejemplo (inglés)
  - >>> from nltk.corpus import stopwords
  - >>> stopwords.words('english')
- Ejemplo (español)
  - >>> from nltk.corpus import stopwords
  - >>> stopwords.words('spanish')

Spacy: palabras vacías

- Aplicando el modelo cargado (inglés, español, etc) realiza diversas funciones de NLP
-

```
import spacy

nlp = spacy.load('es_core_news_sm')

doc = nlp(u'Iberdrola ha alcanzado un beneficio neto de 1.644,4 millones de euros en el primer semestre del año')

for token in doc:
    print(token.text, token.is_stop)
```

Segmentación de palabras

- Objetivo: Romper una cadena de caracteres (grafemas) en una secuencia de palabras.
- Ejemplos de urls:
  - ayudaenlaweb.com/internet-basico/partes-de-una-url/

    ⇒ ayuda en la web .com internet basico partes de una url
  - myspace.com/pluckerswingbar

    ⇒ myspace .com pluckers wing bar

    ⇒ myspace .com plucker swing bar

Segmentación (tokenización) de oraciones

- Las claves usuales para segmentar un texto en oraciones son la puntuación: puntos, los signos de interrogación y los signos de exclamación.
- Los signos de interrogación y los signos de exclamación son marcadores relativamente inequívocos de los límites de las oraciones.
- Los puntos pueden ser ambiguos: fin de oración o marcador de abreviatura: Mr. O Inc.
- Por ello a veces la segmentación (o tokenización) de las oraciones y de las palabras se aborda conjuntamente.

Segmentación de oraciones

- Métodos:
- Suelen decidir primero (en base de reglas o de aprendizaje automático) si un punto forma parte de la palabra o es un marcador de límite de la oración.
- Un diccionario de abreviaturas puede ayudar a determinar si el punto forma parte de una abreviatura de uso común.

Segmentación de palabras

- Ejemplo NLTK

```
>>> import nltk
>>> from urllib import urlopen
>>> https://www.gutenberg.org/files/2554/2554-0.txt
>>> raw = urlopen(url).read()
>>> raw
>>> len(raw)
>>> raw[:75]
>>> raw = raw.decode('utf-8')
>>> tokens = nltk.word_tokenize(raw)
>>> tokens
>>> len(tokens)
```

Análisis morfolológico

- La morfología es el campo de la lingüística que estudia la estructura de las palabras.
- Un morfema es la unidad lingüística más pequeña que tiene significado semántico.
- El análisis morfológico es la tarea de segmentar una palabra en sus morfemas:
  - carried ⇒ carry + ed (past tense)

Análisis morfolológico

- Se distinguen dos grandes grupos de morfemas: las raíces (o stems) y los afijos (que pueden ser prefijos o sufijos)
- Raíz: morfema principal que proporciona significado a la palabra
- Afijos: Añaden significado adicional de distintos tipos (número, género, tiempo del verbo, etc.)
- Ejemplos de análisis morfológico:

  morfemas

  - Inglés: fox → fox

    cats → cat + s
  - Español:

    perros → perr + o + s

    acrecentar → a + crec + entar

Análisis morfolológico

- Los lematizadores identifican el lema o forma canónica de una palabra: leíamos → leer
- El proceso de stemming se refiere al proceso de eliminar la parte final de las palabras, que en muchas ocasiones es una aproximación a la lematización.
- El proceso de stemming es muy frecuente en PLN:
  - Permite reconocer todas las palabras correspondientes a una misma raíz

Lematización y stemming

“Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. If confronted with the token saw, stemming might return just s, whereas lemmatization would attempt to return either see or saw depending on whether the use of the token was as a verb or a noun. The two may also differ in that stemming most commonly collapses derivationally related words, whereas lemmatization commonly only collapses the different inflectional forms of a lemma.”

Manning, C. D., Raghavan, P. y Schütze, H. (2008). Stemming and lemmatization. Introduction to information retrieval. Cambridge: Cambridge University Press. Consultado en https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html

Stemming

- Stemming con nltk:
  - NLTK incluye varios stemmer externos (Porter y Lancaster)
  - Son prefereribles a los que podemos construir manualmente con expresiones regulares → Son capaces de manejar casos irregulares.
  - El stemmer de Porter es una buena elección y probablemente el más utilizado.
  - Tiene versiones para distintos idiomas.

Stemming

- Stemming con nltk:
  - Inglés (Ejemplo):
  - >>> raw = """DENNIS: Listen, strange women lying in ponds distributing swords... is no basis for a system of government. Supreme executive power derives from ... a mandate from the masses, not from some farcical aquatic ceremony."""
  - >>> tokens = nltk.word_tokenize(raw)
  - >>> porter = nltk.PorterStemmer()
  - >>>[porter.stem(t) for t in tokens]
  - >>> ['den', ':', 'list', ',', 'strange', 'wom', 'lying', 'in', 'pond', 'distribut','sword', 'is', 'no', 'bas', 'for', 'a', 'system', 'of', 'govern', '.', 'suprem','execut', 'pow', 'der', 'from', 'a', 'mand', 'from', 'the', 'mass', ',', 'not','from', 'som', 'farc', 'aqu', 'ceremony', '.']

Stemming

- Stemming con nltk:
- Español
  - Lematizador Snowball
  - Basado en el algoritmo de Porter
  - Funciona con varias lenguas entre las que encontramos: inglés, frances, espanol, portugés, italiano, alemán, dutch, swedish, norwegian, danish, ruso y finés.

Stemming

- Español (ejemplo):
  - >>> from nltk.stem import SnowballStemmer
  - >>> spanish_stemmer = SnowballStemmer("spanish")
  - >>> opinion = "Una muy buena cámara de fotografía, muy orientada también a vídeo con su sistema dual pixel de enfoque automático y seguimiento en vídeo. Es una excelente cámara tanto para empezar en el mundo de la fotografía como para profundizar y mejorar como fotógrafo."
  - >>> text = unicode(opinion, 'utf-8')
  - >>> tokens = nltk.word_tokenize(text)
  - >>> [spanish_stemmer.stem(t) for t in tokens]

Lematizador

- Lematizador con nltk:

  Ejemplo (inglés):

  - >>>from nltk.stem import WordNetLemmatizer
  - >>> raw="DENNIS: Listen, strange women lying in ponds distributing swords... is no basis for a system of government. Supreme executive power derives from ... a mandate from the masses, not from some farcical aquatic ceremony."
  - >>> tokens = nltk.word_tokenize(raw)
  - >>>lemmatizer = WordNetLemmatizer()
  - text = unicode(opinion, 'utf-8')
  - tokens = nltk.word_tokenize(text)
  - lemmatizer = WordNetLemmatizer()
  - lemas = [lemmatizer.lemmatize(t) for t in tokens]
  - for lem in lemas:
  - print lem.encode("utf-8" + " ")

Spacy: lemas

- Aplicando el modelo cargado (inglés, español, etc) realiza diversas funciones de NLP
- spacy.load('es')

```
import spacy

nlp = spacy.load('es')

doc = nlp(u'Iberdrola ha alcanzado un beneficio neto de 1.644,4 millones de euros en el primer semestre del año')

for token in doc:
    print(token.text, token.lemma_)
```

POS tagging

- Cada palabra de una oración se puede clasificar en clases: verbos, adjetivos, nombres, etc.
- El etiquetado gramatical o Part-Of-Speech (POS) Tagging es el proceso de etiquetar las palabras de una oración con su categoría léxica, en base a:
  - Su definición
  - El contexto en la oración (las etiquetas y palabras que la rodean)
- Ejemplo (Corpus de Brown): The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

POS tagging

- POS tagging es útil en muchos problemas de PLN:
  - Identificación de entidades
  - Desambiguación del sentido de las palabras
  - Análisis sintáctico, etc.

POS tagging

- Principales técnicas:
  - Basados en reglas
  - Modelos estadísticos: Modelo de probabilidades de las secuencias de palabras y sus etiquetas en función de los datos de un corpus anotado.
  - Aprendizaje automático

POS tagging

- POS tagging con nltk
  - Ingles:

    >>> text = nltk.word_tokenize("And now for something completely different")

    >>> nltk.pos_tag(text)

    [('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'), ('completely', 'RB'), ('different', 'JJ')]

POS tagging

- POS tagging con nltk
  - Español
  - No tiene un modelo pre-entrenado
  - Requiere un corpus en español anotado para el entrenamiento
  - Por ejemplo CESS-ESP
  - El entrenamiento se puede basar en
    - Uni-gramas: se elige la etiq. en función de la palabra
    - Bi-gramas: se elige en función de la palabra y de la etiqueta de la palabra anterior

POS tagging

- POS tagging con nltk: Español (bi-gramas)

  >>> from nltk.corpus import cess_esp as cess

  >>> from nltk import BigramTagger as bt

  # Lee el corpus a una lista: elemento: oración,

  >>> cess_sents = cess.tagged_sents()

  >>> bi_tag = bt(cess_sents, backoff=None)

  >>> sentence = "Esto es una pequeña prueba del tagger en español ."

  >>> print bi_tag.tag(sentence.split(" "))

Spacy: POS tagging

- Aplicando el modelo cargado (inglés, eapañol, etc) realizada diversas funciones de NLP

```
import spacy

spacy.load('es')

nlp = spacy.load('es')

doc = nlp(u'Iberdrola ha alcanzado un beneficio neto de 1.644,4 millones de euros en el primer semestre del año')

for token in doc:
    print(token.text, token.lemma_, token.pos_, token.tag_)
```

  - token.pos_ → POS tag de grano grueso
  - token.tag_ → POS tag de grano fino

POS tagging

- Evaluación: se utilizan
  - la exactitud (accuracy): número de etiquetas acertadas sobre el número total de palabras etiquetadas
  - Precisión, cobertura y medida-F para cada tipo de etiqueta del conjunto de etiquetas considerado

POS tagging

- Ejemplo de evaluación:

  EU NNP NNP

  rejects VBZ VBZ

  German JJ JJ

  call NN VB

  to TO TO

  boycott VB NN

  British JJ JJ

  lamb NN NN

  . . .

Exactidud: 7/9 = 0.77
Medidas para la etiqueta NN
Precision = 0.5
Cobertura = 0.5
F1 = 0.5

Chunking

- Objetivo: parte y etiqueta secuencias multipalabra. Cada secuencia es un *chunk*.
- Uno de los más usados es el de las frases nominales: NP chunking.
- Ejemplo (Wall street journal):
- [The/DT market/NN] for/IN [system-management/NN software/NN] for/IN [Digital/NNP] [’s/POS hardware/NN] is/VBZ fragmented/JJ enough/RB that/IN [a/DT giant/NN] such/JJ as/IN [Computer/NNP Associates/NNPS] should/MD do/VB well/RB there/RB ./.

Chunking

- NP-chunks suelen ser menores que las frases nominales: NP-chunks no contienen otros NP-chunks
- Los sintagmas preposicionales o las cláusulas subordinadas que modifican un nominal no se suelen incluir.
- Se puede construir una gramática de detección de chunks usando expresiones regulares sobre las etiquetas POS. (pag. 265 libro nltk)
- También puede construirse con clasificadores entrenados.

Chunking con expresiones regulares

```
>>> sentence = [("the", "DT"), ("little", "JJ"), ("yellow", "JJ"), ("dog", "NN"), ("barked", "VBD"), ("at", "IN"),  ("the", "DT"), ("cat", "NN")]


>>> grammar = "NP: {<DT>?<JJ>*<NN>}"
>>> cp = nltk.RegexpParser(grammar)
>>> result = cp.parse(sentence)
>>> print result
```

Análisis sintáctico

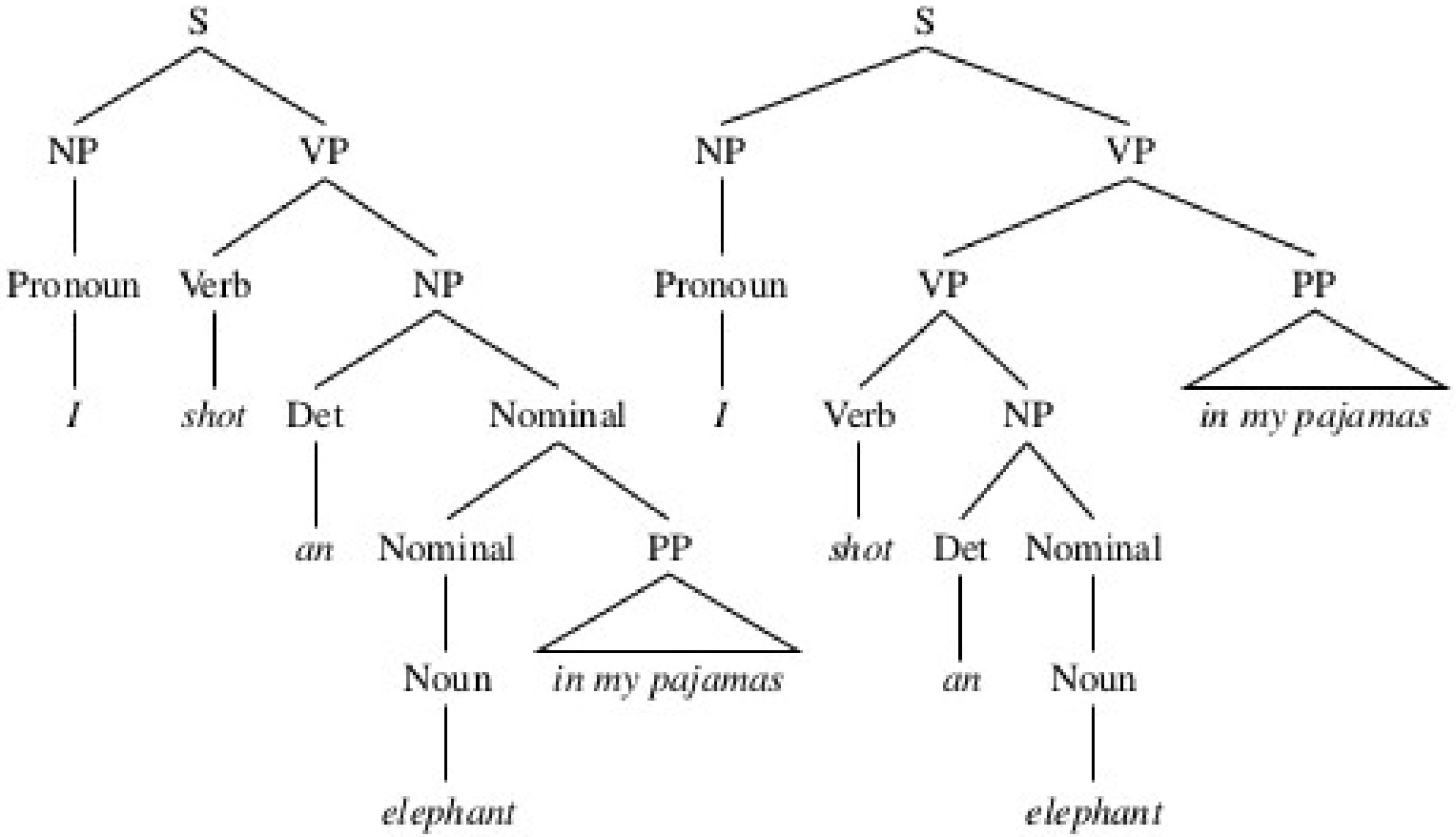
- Objetivo: Identificar la estructura sintáctica de una oración.
- La estructura asignada depende del tipo de gramática considerada.
- Numerosas aplicaciones:
  - Representación intermedia para el análisis semántico.
  - Sistemas de búsqueda de respuestas
  - Sistemas de extracción de información
  - Etc.

Análisis sintáctico

- Dificultades: ambigüedad estructural → una oración puede tener múltiples árboles de análisis respecto a una misma gramática

Análisis sintáctico

- Ejemplo:



Análisis de constituyentes

- Una frase bien formada se puede descomponer en constituyentes de acuerdo a unas reglas gramaticales.
- Un analizador sintáctico de constituyentes es un programa que obtiene la estructura asociada a una frase respecto a una gramática.
- Las gramáticas que utilizan los analizadores son Context Free Grammar (CFG): las producciones o reglas no dependen de las palabras anteriores o posteriores a las asignadas a la regla.

Análisis de constituyentes

- Una CFG consiste en:
  - Un conjunto de terminales (las palabras del lenguaje a representar) (∑)
  - Un conjunto de símbolos no terminales N(S, NP, etc.)
  - Un símbolo inicial de la gramática S, que pertenece a N
  - Un conjunto de producciones o reglas
  - R: $A \rightarrow \alpha$ donde $\alpha \in (\sum \cup N)^*$

Análisis sintáctico con nltk

```
# python 3
from nltk import CFG
groucho_grammar = CFG.fromstring("""
S -> NP VP
PP -> P NP
NP -> Det N | Det N PP | 'I'
VP -> V NP | VP PP
Det -> 'an' | 'my'
N -> 'elephant' | 'pajamas'
V -> 'shot'
P -> 'in'
""")
groucho_grammar.start()
groucho_grammar.productions()
sent = ['I', 'shot', 'an', 'elephant', 'in', 'my', 'pajamas']
parser = ChartParser(groucho_grammar)
trees = parser.parse(sent)
for tree in trees:
   print (tree)
```

Análisis de constituyentes:PCFGs

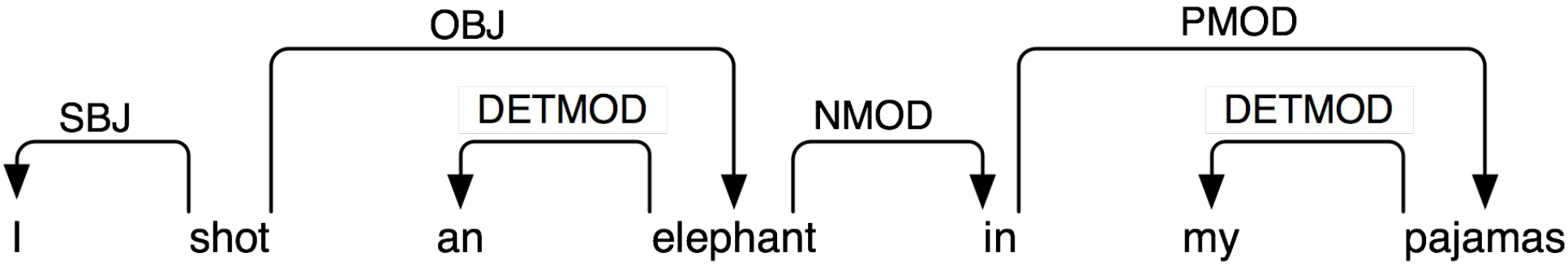
- Algunos análisis validos de las oraciones pueden ser muy infrecuentes (como ej Groucho Marx sobre pijamas).
- En las probabilistic CFG (PCFGs) las reglas gramaticales tienen asociada una probabilidad:
  - S → NP (0.3)
  - S → NP VP PP (0.7)
- El analizador busca la secuencia de producciones más probable.
- Permiten asociar una probabilidad a cada árbol de análisis alternativo.
- Frecuentemente interesa el más probable.

Análisis sintáctico con nltk

```
# python 3
from nltk import PCFG
groucho_grammar = PCFG.fromstring("""
S -> NP VP [1.0]
PP -> P NP [1.0]
NP -> Det N [0.5] | Det N PP [0.3] | 'I' [0.2]
VP -> V NP [0.6]| VP PP [0.4]
Det -> 'an' [0.5] | 'my' [0.5]
N -> 'elephant' [0.5] | 'pajamas' [0.5]
V -> 'shot' [1.0]
P -> 'in' [1.0]
""")
sent = ['I', 'shot', 'an', 'elephant', 'in', 'my', 'pajamas']
viterbi_parser = nltk.ViterbiParser(groucho_grammar)
trees = viterbi_parser.parse(sent)
for tree in trees:
   print (tree)
```

Análisis de dependencias

- Un análisis alternativo basado en las relaciones entre palabras de una oración.



- Las dependencias pueden estar anotadas (tener nombre)
- Aprenden de un corpus anotado con análisis de dependencias (Treebank de dependencias)
- Generalmente no hay una gramática (a diferencia del análisis de constituyentes)
- Las palabras están también en nodos internos (a diferencia del análisis de constituyentes que produce árboles con palabras sólo en los nodos hoja).

Análisis sintáctico con nltk

```
# python 3
from nltk import DependencyGrammar

groucho_dep_grammar = nltk.DependencyGrammar.fromstring("""
  'shot' -> 'I' | 'elephant' | 'in'
  'elephant' -> 'an' | 'in'
  'in' -> 'pajamas'
  'pajamas' -> 'my'
  """)
print(groucho_dep_grammar)

pdp = nltk.ProjectiveDependencyParser(groucho_dep_grammar)
sent = 'I shot an elephant in my pajamas'.split()
trees = pdp.parse(sent)
for tree in trees:
     print(tree)
```

Spacy: token, POS tag,análisis..

- Aplicando el modelo cargado (inglés, español, etc) realiza diversas funciones de NLP
- spacy.load('es')

```
import spacy

nlp = spacy.load('es')

doc = nlp(u'Iberdrola ha alcanzado un beneficio neto de 1.644,4 millones de euros en el primer semestre del año')

for token in doc:

    print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_, token.shape_, token.is_alpha, token.is_stop)
```

Spacy: token, POS tag,análisis..

Algunos atributos de un token en Spacy (https://spacy.io/api/token)

- token.text → Texto
- token.lemma_ → lema
- token.pos_ → POS tag de grano grueso
- token.tag_ → POS tag de grano fino
- token.dep_ → Relación sintáctica de dependencia
- token.is_stop → indicativo de si es una stopword

Análisis sintáctico: Técnicas frecuentes

- Programación dinámica:
  - Análisis descendente (partiendo del símbolo inicial de la gramática)
  - Análisis ascendente (partiendo de los símbolos terminales)
- Aprendizaje automático a partir de un corpus anotado con análisis (treebank)

Desambiguación del sentido de las palabras

- Muchas palabras tienen más de un significado o sentido dependiendo del contexto en que se usan:
  - Los pescadores encontraron un gran *banco* de sardinas
  - Tuve que ir al *banco* a sacar dinero
  - Nos sentamos en el *banco* hasta que llegó
- Muchas aplicaciones requieren seleccionar el sentido correcto: Word Sense Disambiguation (WSD).

Desambiguación del sentido de las palabras

- WSD: Consiste en seleccionar el sentido (interpretación o significado) correcto de una ocurrencia de una palabra en un documento (o conversación) de entre todos los posibles de la palabra.
- Aplicaciones:
  - traducción automática: WSD es necesaria las palabras que tienen diferentes traducciones para diferentes sentidos.
  - Recuperación de información: es necesario desambiguar las consultas: depresión puede hacer referencia una enfermedad, a una situación económica, etc.
  - Extracción de información: para la interpretación correcta de los textos. *Drug* puede hacer referencia a un medicamento o a una sustancia ilegal.
  - Y otras muchas.

Desambiguación del sentido de las palabras

Técnicas:

- Métodos basados en el diccionarios y bases de conocimiento léxico: por ejemplo en base a *distancias semánticas* entre las definiciones de las palabras.
- Métodos supervisados: entrenados sobre corpus anotados con los sentidos correctos de las palabras.
- Métodos semisupervisados o mínimamente supervisados: Usan una fuente secundaria de conocimiento como un pequeño corpus anotado como datos de semillas en un proceso de bootstrapping.
- Métodos no supervisados: Estos evitan (casi) completamente la información externa y trabajan directamente desde los corpus no anotados. La hipótesis subyacente es que sentidos similares aparecen en contextos similares y, por lo tanto, se pueden inducir sentidos a partir del texto agrupando ocurrencias de palabras utilizando alguna medida de similitud de contexto.

Desambiguación del sentido de las palabras

- Relaciones entre sentidos:
  - Sinónimos: distintas palabras con el mismo (o muy similar) significado
  - Hipónimo: un sentido es hipónimo de otro si el primero es más específico (una subclase).
  - Hiperónimo: un sentido es hiperónimo de otro si el primero es más general (superclase)

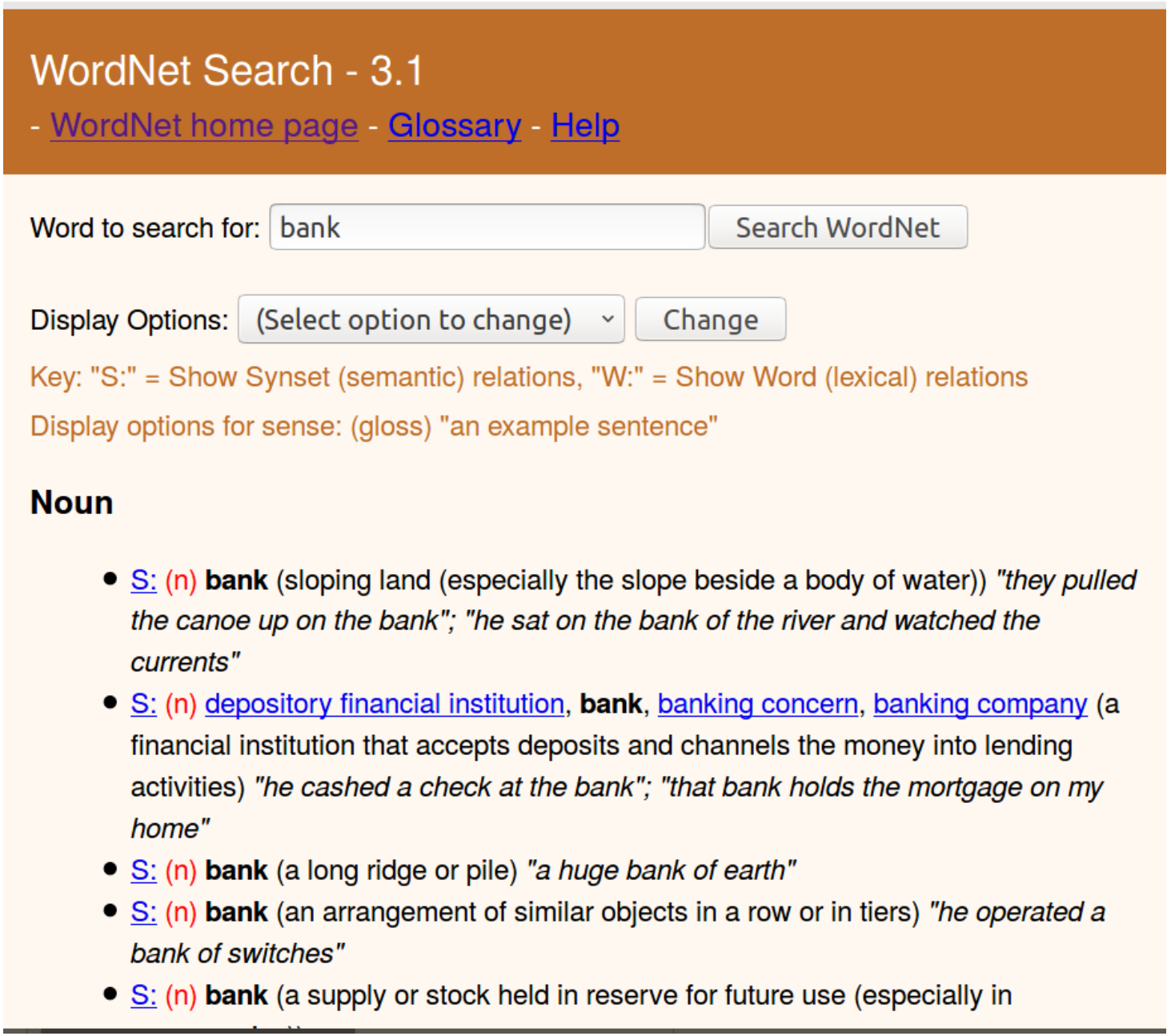
Desambiguación del sentido de las palabras

- Wordnet: una base de datos de relaciones léxicas.
- Este lexicón o taxonomía se divide en 5 categorías: nombres, verbos, adjetivos, adverbios y palabras de función (pronombres, etc).
- Representación del sentido de las palabras:
  - synsets (synonym sets) como unidades básicas
  - El significado de una palabra se representa listando las formas de la palabra que se pueden usar para expresarlo

Desambiguación del sentido de las palabras

- Los elementos del synset rara vez son verdaderos sinónimos:
  - Wordnet no intenta capturar distinciones sutiles
- Los synsets suelen ser suficiente para analizar diferencias y similitudes.
- Wordnet también proporciona glosses: definiciones y/o ejemplos

Desambiguación del sentido de las



WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: bank Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Noun

- S: (n) **bank** (sloping land (especially the slope beside a body of water)) *"they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents"*
- S: (n) depository financial institution, **bank**, banking concern, banking company (a financial institution that accepts deposits and channels the money into lending activities) *"he cashed a check at the bank"; "that bank holds the mortgage on my home"*
- S: (n) **bank** (a long ridge or pile) *"a huge bank of earth"*
- S: (n) **bank** (an arrangement of similar objects in a row or in tiers) *"he operated a bank of switches"*
- S: (n) **bank** (a supply or stock held in reserve for future use (especially in

WSD basada en diccionarios

- WSD a veces se trata con métodos supervisados.
- Pero frecuentemente se recurre a diccionarios y tesauros.
- Algoritmo de Lesk: elige el sentido cuya definición (glosa) comparte más palabras con el contexto de la palabra a desambiguar.

WSD en nltk

```
from nltk.wsd import lesk
sent = ['I', 'went', 'to', 'the', 'bank', 'to', 'deposit',
'money', '.']

ambiguous = 'bank'

print(lesk(sent, ambiguous))

print(lesk(sent, ambiguous).definition())
```

Referencias

- Speech and Language Processing (3rd ed. draft)

  Dan Jurafsky and James H. Martin (Draft chapters in progress, Sep 23, 2018)

  - Tema 2: Regular Expressions, Text Normalization, and Edit Distance
  - Tema 8: Part-of-Speech Tagging
  - Tema 11: Syntactic Parsing
  - Tema C: Computing with Word Senses: WSD and WordNet

Referencias

- Natural Language Processing with Python

  Steven Bird, Ewan Klein, and Edward Loper

  O'Reilly Media Inc. https://www.nltk.org/book/

  - 2. Accessing Text Corpora and Lexical Resources
  - 3. Processing Raw Text
  - 5. Categorizing and Tagging Words
  - 8. Analyzing Sentence Structure

Enlaces de interés

- The Stanford NLP Group

  Diversos recursos: POS tagging, parsing, etc

  https://nlp.stanford.edu/software/
- TreeTagger - a part-of-speech tagger for many languages

  http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/