Universidad
Carlos III de Madrid

Módulo VIII
Técnicas de Procesamiento de Lenguaje Natural (PLN)

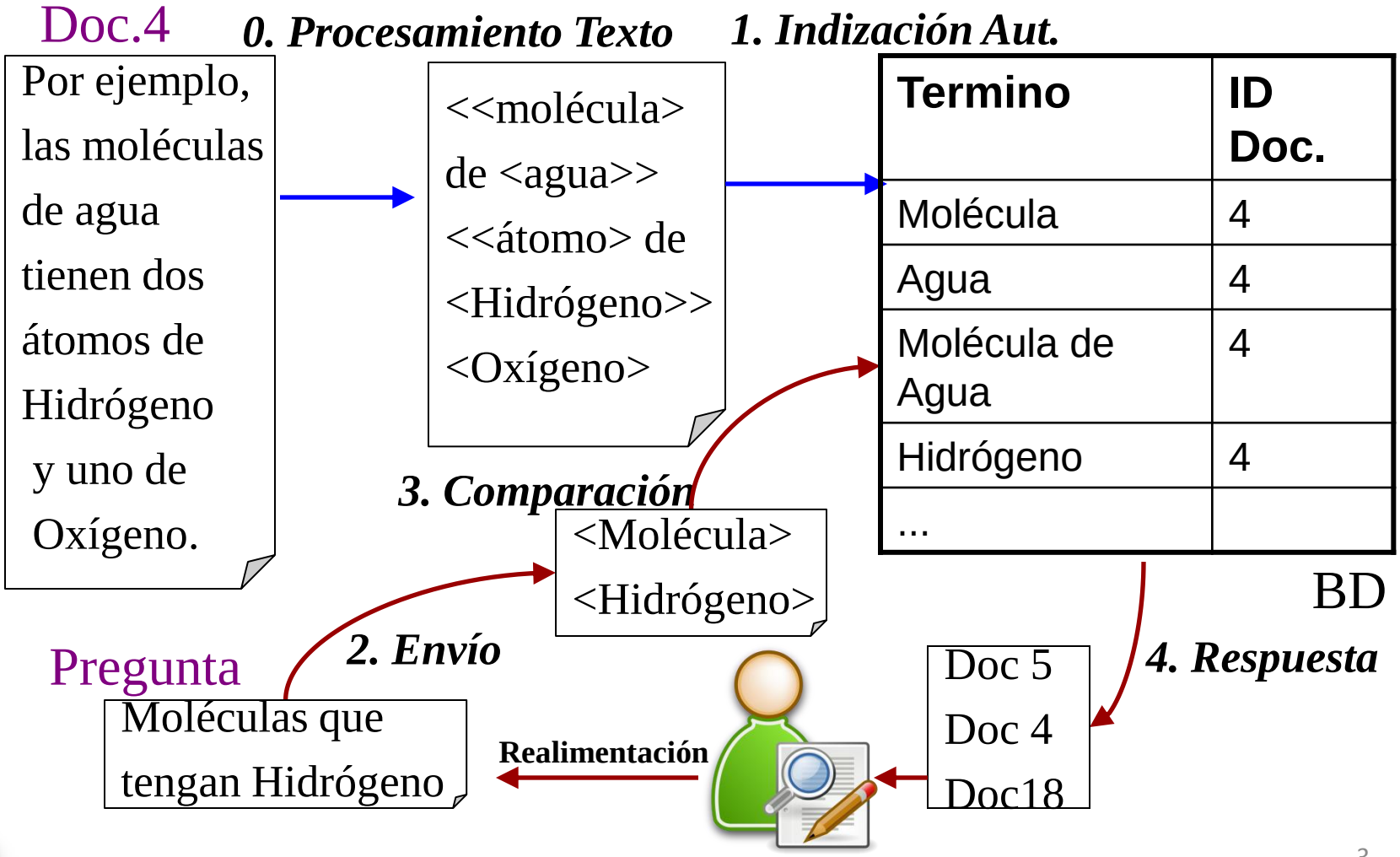
OpenCourseWare

Recuperación y Acceso a la Información

Contenidos

- Proceso de recuperación
  - Búsqueda por campos
  - Búsqueda por frase
- Preprocesamiento de documentos y consultas
  - Tokenización
  - Filtrados
    - Palabras vacías
    - Ley de Zipf
  - Normalización
    - Stemming y lematización
  - Análisis morfo-sintáctico
- Software para PLN
- Peso de los términos

Proceso de recuperación



Unidades de recuperación

- Unidades de indización:
  - Colecciones documentales
  - Documentos
  - Frases
- Unidades grandes (ej. recuperación por libros). Problemas:
  - Demasiado texto para el usuario
  - Ruido en los resultados (términos no representativos en el documento)
- Unidades pequeñas(ej. recuperación por frases). Problemas:
  - No relevante por sí sola. Requiere contexto
  - Silencio en los resultados (frases útiles que no contienen los términos de búsqueda, pero sí el documento en el que se encuentran)
- Adaptar unidades a necesidades:
  - Elementos potencialmente recuperables. Ej. libros en una biblioteca
  - Elementos que constituyan una unidad conceptual

Búsqueda por campos

- Búsqueda por campos
  - Puede interesarnos localizar información que pertenezca a determinada zona del documento (ej. título o resumen) o a determinados metadatos (ej. fecha) para permitir búsquedas por campos.
  - Esto es más importante si cabe en información con cierta estructura, como la etiquetada en XML.
  - También nos permite asociar diferentes pesos a cada documento según dónde se encuentren los términos de la consulta
- Métodos
  - Creación de índices independientes por campo
    - Problema: espacio
    - Cálculo de peso asociado a un campo (aprendizaje, expertos, …)
  - Entrada indicando el campo en el índice
- **¿Qué peso tiene cada campo?**

Búsqueda por frase

- **Biword indexes**
  - Almacenamiento de pares de términos en el índice
  - Ej. Búsqueda de “Universidad Carlos III de Madrid” generaría la siguiente consulta:

    “Universidad Carlos” AND “Carlos III” AND “III de” AND “de Madrid”

  - Problemas:
    - Falsos positivos
    - Aumento espacio almacenamiento en índice
    - El algoritmo no sirve para realizar búsqueda por proximidad
  - Ventaja: rapidez

Búsqueda por frase

- **Positional indexes**
  - Almacenamiento de la posición de cada ocurrencia de un término
  - De los documentos que contienen los términos de consulta, se analiza la posición de esos términos a ver si es correcta (si forman la frase)
  - Problema: tiempo de procesamiento
  - Ventaja: válido para búsquedas por proximidad

- **Enfoque combinado**
  - Podemos utilizar la estrategia de *positional indexes* pero mantener en el índice los pares más frecuentemente buscados
  - Ej. Una búsqueda muy frecuente en la Web es *Britney Spears*

Representación del documento en la base de datos

- Indización automática y manual
- Procesamiento del lenguaje para indización automática

Indización Automática

"[...] un ordenador reconoce los términos que figuran dentro del título, del resumen, del texto completo [...] empleando estos términos tal cual, o bien después de transformarlos en otros términos, equivalentes o conceptualmente próximos, con el fin de convertirlos en elementos que se incorporan al fichero de búsqueda y quedan disponibles para recuperar el documento"

(Van Slype, 1991)

Tratamiento textual. Indización

Variantes de Bagle en la 'sopa de virus' de Internet

EFE SAN FRANCISCO (EEUU).-

La aparición de cuatro nuevas variantes del **virus Bagle** tiene en jaque a los expertos en seguridad. Las variantes de este gusano amenazan con acabar con las letras del abecedario: se trata del *Bagle.Q* (al parecer, la más extendida), *Bagle.R*, *Bagle.S* y *Bagle.T*, hermanos menores del original, que hizo su aparición en enero.

La peligrosidad del virus, que afecta sólo a los sistemas que utilizan el sistema operativo *Windows*, de *Microsoft* (y no el *Macintosh*, de *Apple*) radica en que no necesita que el usuario abra el fichero incluido en el correo para infectar su computadora, informa la corresponsal de *EFE* en EEUU *Natalia Martín Cantero* tras entrevistar a *Graham Cluley*, consultor de la *compañía Sophos*. Sin embargo, algunos expertos creen que los parches que lanzó *Microsoft* para tapar ese agujero podrían ser insuficientes, ya que aún así podrían infectarse.

Los comentarios escondidos en el código de programación hicieron pensar a los investigadores que piratas informáticos podrían estar compitiendo entre ellos. Así, *Bagle.J* contenía una línea en su código que decía: "*Hey, Netsky... no arruines nuestro negocio, *quieres entrar en guerra?*"

Tratamiento textual

Vamos a extraer las palabras fundamentales del texto

Variantes de Bagle en la 'sopa de virus' de Internet

EFE SAN FRANCISCO (EEUU).-

La aparición de cuatro nuevas variantes del **virus Bagle** tiene en jaque a los expertos en seguridad. Las variantes de este gusano amenazan con acabar con las letras del abecedario: se trata del *Bagle.Q* (al parecer, la más extendida), *Bagle.R*, *Bagle.S* y *Bagle.T*, hermanos menores del original, que hizo su aparición en enero.

La peligrosidad del virus, que afecta sólo a los sistemas que utilizan el sistema operativo *Windows*, de *Microsoft* (y no el *Macintosh*, de *Apple*) radica en que no necesita que el usuario abra el fichero incluido en el correo para infectar su computadora, informa la corresponsal de *EFE* en EEUU *Natalia Martín Cantero* tras entrevistar a *Graham Cluley*, consultor de la *compañía Sophos*. Sin embargo, algunos expertos creen que los parches que lanzó *Microsoft* para tapar ese agujero podrían ser insuficientes, ya que aún así podrían infectarse.

Los comentarios escondidos en el código de programación hicieron pensar a los investigadores que piratas informáticos podrían estar compitiendo entre ellos. Así, *Bagle.J* contenía una línea en su código que decía: "*Hey, Netsky... no arruines nuestro negocio, *quieres entrar en guerra?*"

Indización Manual

Seguridad en Internet

Virus Informático

Bagle

S.O. Windows

Antivirus

Correo electrónico

- La indización manual es por asignación intelectual de conceptos, la automática por extracción de palabras
- La automática suele ser más exhaustiva (muchos descriptores) pero también da más ruido
- Consistencia o coincidencia entre ambas baja

- Netsky
- EEUU
- compañía Sophos
- Macintosh
- microorganismos
- EFE

- Gusanos
- Jaque
- Guerras
- Piratas
- Natalia Martín Cantero

Uso de recursos lingüísticos

- Preproceso (similar a cleansing)
- Tokenización = Trocear el texto
- Listado de palabras vacías (*stopwords*)
  - Filtrado de palabras: Zipf, ngrams
  - Asignación de pesos: tf-idf

Contar el número de veces que aparece un término en el texto

- Algoritmos para terminaciones (*stemming*)
- Diccionarios/listados de palabras
- Lematización

Preproceso

- Tareas a realizar antes de realizar un procesamiento del lenguaje. No siempre se pueden aplicar en un primer momento
  - Identificación de idioma
  - Corrección tipográfica y ortográfica:
    - Levehenstein, ngrams, fingerprint
    - Agrupamientos fonéticos

Tratamiento textual. Tokenización

- ¿Cómo dividimos las frases?
  - Por puntos ...?
- ¿Cómo identificamos las palabras?
  - Por espacios...?

- F-15
- MS-DOS
- ß-caroteno
- ¿Rosa y rosa diferentes?
- Bajo 0
- Archivo.exe
- www.uc3m.es
- Vitamina C, rayos X
- 125.564,8
- Blanco, veloz
- 91 644 1528
- 1994
- EE.UU.
- Carlos, Pepe
- Teclear

Tokenizer

- Es el programa encargado de segmentar la frase en palabras simples o compuestas. Pueden dividirla en sintagmas, grupos nominales, etc
- Tokens: unidades en el texto
- Términos: unidades en el índice del sistema de recuperación
- Ej. *el coche de caballos corría veloz -> 6* tokens, 4 términos
- Problemas:
  - Delimitadores de tokens.
    - Ej. O'Neill son dos términos o uno? Si es un término se mantiene o no el apóstrofe? Qué ocurre con el ejemplo *Tom's car*?
    - Ej. archivo.exe, 91 234 567, 18 de febrero de 2010
    - Ej. O.N.G, ONG
    - Ej. Ibex-35, Ibex 35
  - Términos compuestos. Ej. Vitamina C, bajo 0, coche de caballos

Demo	**http://text-processing.com/demo/tokenize/**

Tokenizer. Ejemplo Solr

Función	**Solr**
Fragmentar la frases en tokens	N/A, solr.KeywordTokenizerFactory, solr.WhitespaceTokenizerFactory
Identificar tokens con puntos (p.e. email o URL)	solr.StandardTokenizerFactory, solr.PatternTokenizerFactory solr.KeywordTokenizerFactory solr.UAX29URLEmailTokenizerFactory
Identificar abreviaturas	standard filter
Cantidades	solr.StandardTokenizerFactory (ver. 3.1, solo números, "5" pero no "five")
Expresiones regulares	solr.PatternTokenizerFactory

Tokenizer. Ejemplo Solr

Identificar mayúsculas	solr.PatternTokenizerFactory
Símbolos especiales	solr.PatternTokenizerFactory
Locuciones	N/A, se puede simular con solr.StopFilterFactory (palabras sencillas) or solr.CommonGramsFilterFactory (palabras compuestas)

Ejemplos de locuciones: “Por ejemplo,”, “En primer lugar, “, “sano y salvo”, “echar de menos”, “de veras”, “de ninguna manera”, “puesto que”, “a pesar de”, “la carne de gallina” (frecuentemente las palabras de estas frases tienen una semántica diferente al conjunto, es decir TIENE SENTIDO NO TOKENIZAR cada una de sus palabras)

Tratamiento textual. Tokenización

En conclusión …

- Definición de palabra, frase, párrafo
- Fórmulas, secuencias alfanuméricas
- Guiones
- Tamaño palabra mínimo
- Acrónimos
- Errores

Filtrado. Palabras vacías (stop words)

- Son palabras funcionales o de bajo contenido semántico
  - Las 10 palabras más frecuentes en inglés representan el 25% de palabras en el texto, la BD reduce el tamaño proporcionalmente
- Existen listas de palabras vacías centradas en dominios concretos: Bos taurus, Botrytis cinerea, C. elegans, Spinach, cDNA, DNA clone, BAC, PAC, cosmid, clone, etc
- Métodos
  - **Mediante listas ya existentes** (suelen recoger artículos, preposiciones, etc.)
  - Las palabras que con mayor **frecuencia** aparecen en una colección (**IDF** ¿dónde está el valor de corte?). Elimina términos con semántica pero comunes
  - **Categorías gramaticales** que consideremos que no aportan semántica (ej. artículos, preposiciones, adjetivos, adverbios, etc.). Requiere análisis gramatical.

Filtrado. Palabras vacías (stop words) (II)

- Tres listados: en rojo ORBIT, en mayúsculas WSJ

A	HAVE	out	who
all	he	she	WILL
AN	her	so	WITH
AND	him	THAT	would
ARE	his	THE	you
AS	i	their	
at	if	there	
BE	IN	they	
BEEN	IS	THIS	
BUT	IT	to	
BY	more	WAS	
FOR	not	we	
from	OF	were	
had	ON	when	
has	OR	WHICH	

Filtrado. Palabras vacías (stop words) (III)

- Problema: si eliminamos estos términos de ciertas consultas pierden el sentido
  - Ej. flights to London, Let it be (song)

- Según mejoran los Sistemas de Recuperación de Información se van utilizando menos palabras vacías
  - SMART entre 518-418 palabras
  - Brown corpus entre 425-300 palabras
  - WSJ 27 palabras
  - ORBIT 8 palabras

- Actualmente la tendencia es tratarlas adecuadamente, no eliminarlas

Normalización

- Proceso por el que los tokens se convierten a una forma canónica, de modo que pueda existir matching entre dos tokens a pesar de pequeñas variaciones superficiales en ellos.
- Ejemplos:
  - Tildes y otros símbolos de puntuación. Suelen eliminarse
  - Mayúsculas. Suelen convertirse a minúsculas
  - Acrónimos (e.g. C.A.T.->CAT->¿cat?)
  - Cuestiones idiomáticas (e.g. color<->colour)
- Modos de implementarla:
  - Clases equivalentes de términos: reglas según las cuales se sustituye un término por el transformado. Problema: bi-direccional por pares (e.g. Bush->¿bush?)
  - Expansión de términos a otros: se añaden términos a la consulta a modo de diccionario de sinónimos
    - Al indizar. Problema: espacio almacenamiento
    - Al recuperar. Problema: tiempo procesamiento durante consulta

Stemming

- Normalizan a la raíz teniendo sólo en cuenta la terminación de la palabra (sufijos) o el inicio.
- Sencillos de programar. Disminuyen la BD entre un 10-50%
- Son frecuentemente derivativos
- Normalmente hay un tamaño mínimo de palabra sobre el que hacer el stemming
- Tienen reglas muy simples del tipo sustituir final "-ando" por "-ar", así pasa "caminando" a "caminar"
- Las reglas se aplican iterativamente: "-al" por "-o" y "-mente" por " nada " así en "proverbialmente"→"proverbial"→"proverbio"
- En ocasiones se acaba en una raíz en vez de una palabra. P.e. "proverb"
- Dan errores de infra o sobre radicación ("comunismo", "comunal", "comunes" a "común") [under and overstemming]
- Los más conocidos son Lovins, Porter y eliminación del plural -s

Stemming

- Dependientes de idioma
- Stemmers más conocidos en inglés:
  - Lovins (Lovins, 1968) : 260 terminaciones
    http://www.cs.waikato.ac.nz/~eibe/stemmers/
  - Porter (Porter, 1980) : 60 terminaciones, 5 iteracciones
    http://tartarus.org/~martin/PorterStemmer/
  - Lancaster (Paice, 1990)

    http://textanalysisonline.com/nltk-lancaster-stemmer

- Snowball: librería en C y Java con Lucene para crear y utilizar stemmers en múltiples idiomas.
  - http://snowballstem.org/demo.html

Stemmers

Stemmers	On-line	Explicación
Lovins	http://snowballstem.org/demo.html	http://snowball.tartarus.org/algorithms/lovins/stemmer.html Snowball es un lenguaje para crear stemmers
Porter	http://9ol.es/porter_js_demo.html http://text-processing.com/demo/stem/	https://tartarus.org/martin/PorterStemmer/
WordNet	http://text-processing.com/demo/stem/	
N-gram	http://guidetodatamining.com/ngramAnalyzer/	

http://www.kenbenoit.net/courses/tcd2014qta/readings/Jivani_ijcta2011020632.pdf

Stemming

- **Sample text:** Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation
- **Lovins stemmer:** such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres
- **Porter stemmer**: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret
- **Paice stemmer:** such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret.

Manning, Raghavan and Schütze. Introduction to Information Retrieval. Cambridge University Press, 2008. página 32

Lucene stemmer snowball

all:web all:develop all:is all:a all:broad all:term all:for all:the all:work all:involv all:in all:develop all:a all:web all:site all:for all:the all:internet (all:world all:wide all:web all:or all:www) all:or all:an all:intranet (all:a all:privat all:network) all:this all:can all:includ all:web all:design all:web all:content all:develop all:"client side server side" all:script all:web all:server all:and all:network all:secur all:configur all:and all:"e commerc" all:develop all:howev all:among all:web all:profession all:"web develop" all:usual all:refer all:to all:the all:main all:"non design" all:aspect all:of all:build all:web sites:write all:markup all:code all:etc all:sinc all:the all:mid-1990 all:web all:develop all:has all:been all:one all:of all:the all:fastest all:grow all:industri all:in all:the all:world all:in all:1995 all:there all:were all:fewer all:than all:1,000 all:web all:develop all:compani all:in all:the all:unit all:state all:but all:by all:2005 all:there all:were all:over all:30,000 all:such all:compani all:in all:the all:us all:alon all:presid all:john all:f all:kennedi all:creat all:usaid all:in all:1961 all:by all:execut all:order all:to all:implement all:develop all:assist all:program all:in all:the all:area all:author all:by all:the all:congress all:in all:the all:foreign all:assist all:act all:microsoft all:co all:is all:an all:american all:multin all:corpor all:headquart all:in all:redmond all:washington all:establish all:on all:april all:4 all:1975 all:to all:develop all:and all:sell all:basic all:interpret all:for all:the all:altair all:8800 all:microsoft all:rose all:to all:domin all:the all:home all:comput all:oper all:system all:market all:with all:"ms dos" all:in all:the all:mid-1980 all:retriev all:from http:en.wikipedia.org all:fed all:rais all:interest all:rate all:0.5 all:percent all:the all:book all:were all:book all:by all:somebodi all:els

Tratamiento textual. Lematizador

- Existen dos tipos de normalizaciones lingüísticas en términos
  - Flexiones
    - Nominales: género, número E.g. gato-gata-gatos-gatas
    - Verbales: tiempo, modo, etc. E.g. leer-leyendo
  - Derivaciones: adición de prefijos o sufijos. E.g. fruta-frutero
- Los lematizadores unifican solamente a formas flexionadas (sin cambio categoría gramatical)
  - Para ello tienen en cuenta información lingüística (análisis morfológicos, sintácticos) y estadística (p.e. Algoritmos de Markov)
  - Ejemplo:

    "Los sobres están sobre el armarito bajo que está bajo los bajos de la escalera" →

    El sobre estar sobre el armario bajo que estar bajo el bajo de el escalera

Tratamiento textual. Lematización

- Unificar formas a una misma raíz léxica
  - Por derivación: cambio categoría (p.e. Camino (n) y caminar (v) a camino)
  - Por flexión: sin cambio (p.e. Caminando (v), caminaba (v) a caminar)
- Mejoran la eficacia de los algoritmos de posicionamiento y recuperación:
  - Estima mejor las frecuencias terminológicas: p.e "niño", "niñitos", "niños" como una única palabra. Resolver anáforas, como pronombres
  - Procesar la pregunta como el documento: si se pregunta por "niños" me puede devolver documentos que contengan la palabra "niño"
- Si se aplica antes de indizar, disminuye el espacio ocupado en la base de datos aumentando la velocidad

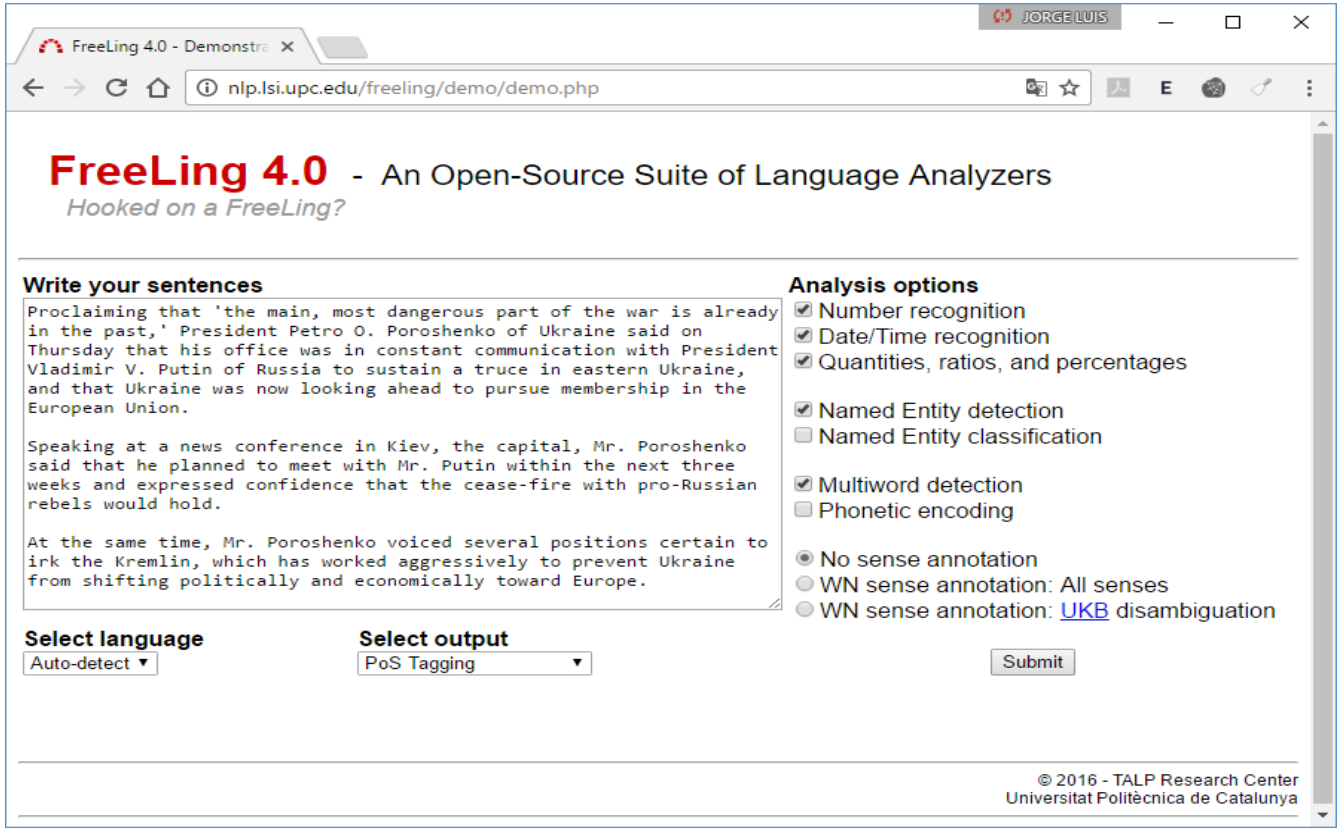
Lematización vs Stemming

- El **stemming** aplica heurísticas para eliminar flexiones y afijos
  - Ejemplo de reglas: sses->ss (caresses->caress), s->Ø (cats->cat), ies->y
  - Consume poco tiempo, pero es poco preciso
  - Más errores de infra o sobre radicación (“comunismo”, “comunal”, “comunes” a “común”) [under and overstemming]
- La **lematización** hace lo mismo, pero utiliza análisis morfológico para determinar el tipo de palabra y aplica reglas en consecuencia para obtener la base de la palabra (*lema).*
  - Consume más tiempo, pero es más preciso
  - Mejores que los stemmer si se pretende “extraer información” más que “recuperar documentos”
  - En español y lenguas con más derivaciones y flexiones funciona mejor que el stemmer

Análisis morfo-sintáctico

- Part Of Speech (POS) tagging: asignar a cada token su categoría gramatical correcta
  - Ej. Etiquetar la siguiente frase: “El petróleo es un combustible”
  - El (artículo) petróleo (sustantivo) es (verbo) un (determinante) combustible (sustantivo)
- Una misma palabra puede tener diferentes categorías gramaticales en distintas frases: desambiguación
  - Ej. “El **sobre** que **sobre** lo dejas **sobre** la mesa”
- Ayuda en la recuperación:
  - Identificando términos simples y compuestos
  - Identificando términos con mayor semántica
  - Identificando sintagmas nominales y relaciones
- Suelen basarse en aprendizaje a partir de corpus anotados (aprendizaje supervisado).

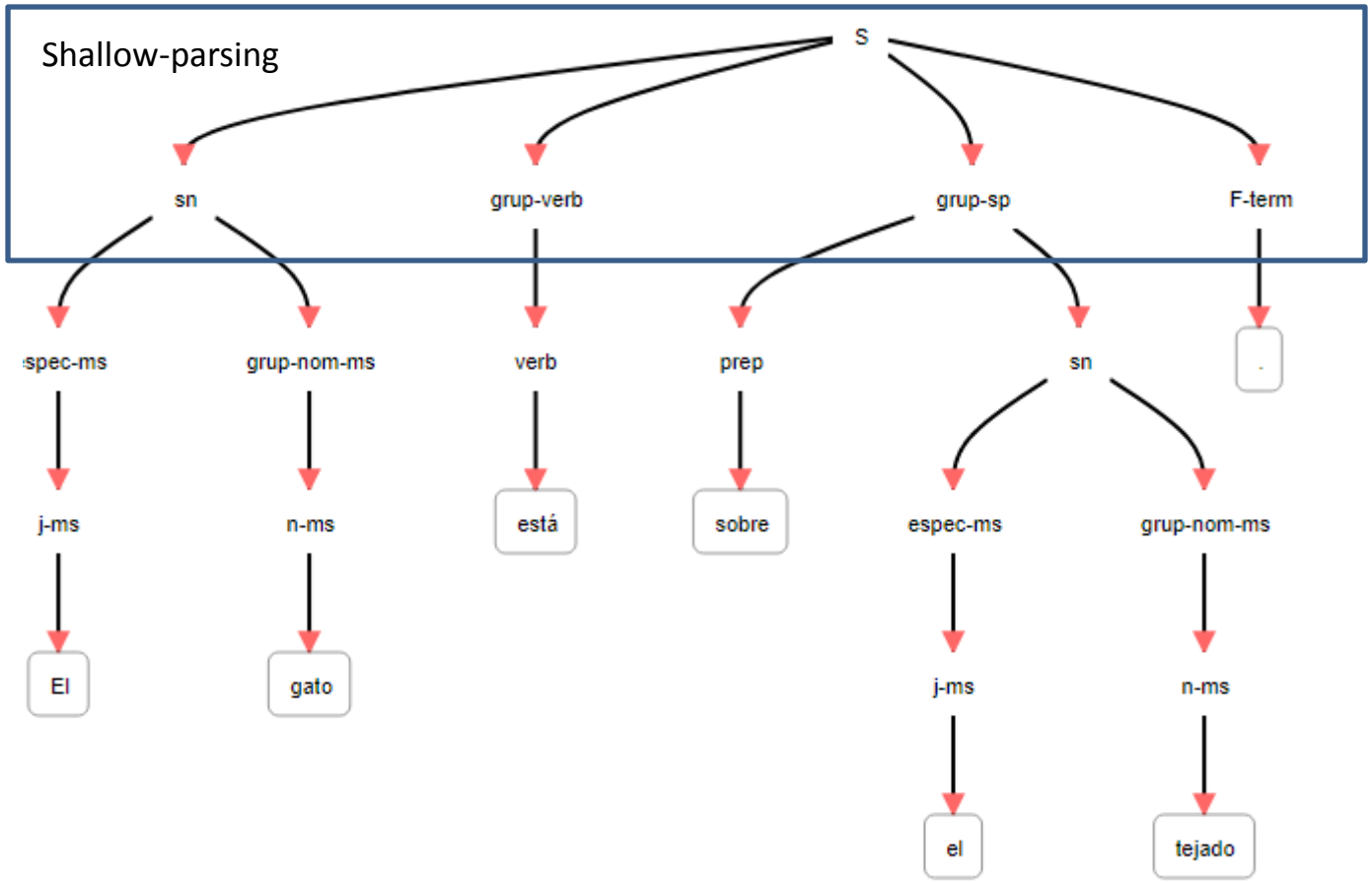
Demo. Freeling



http://nlp.lsi.upc.edu/freeling/demo/demo.php

Se puede descargar e instalar, en varios idiomas

Análisis morfo-sintáctico (I)



Análisis morfo-sintáctico (III)

- Ejemplos de etiquetadores morfo-sintácticos:
  - Brill Tagger
    http://cst.dk/online/pos_tagger/uk/index.html
  - Tree Tagger
    http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
  - TnT
    http://www.coli.uni-saarland.de/~thorsten/tnt/
  - Standford Log-linear POS tagger (en java)
    http://nlp.stanford.edu/software/tagger.shtml

- Otros:
  - Grupo Estructuras de datos y lingüística computacional de la Universidad de Las Palmas http://www.gedlc.ulpgc.es/
  - Corpus etiquetado ANCORA http://clic.ub.edu/corpus/es/ancora-descarregues

Etiquetas

Name	Web page reference
Penn Treebank	https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html, hay versiones con más etiquetas
EAGLES	http://www.ilc.cnr.it/EAGLES/annotate/annotate.html
Bnc2	http://ucrel.lancs.ac.uk/bnc2/bnc2guide.htm
brown	http://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html

Distintas herramientas de PLN utilizan diferentes etiquetas:

GATE	NLTK	C&C tools	OpenNLP	LingPipe	Freeling	Freeling	LUCENE	Stanford NLP
EN	EN		EN	EN	EN	ES		
PENN TREEBANK	Brown Corpus		PENN TREEBANK	Brown Corpus	PENN TREEBANK	EAGLES		

Herramientas POS

	Demo	Etiquetas
POS (tag y chunk)	http://text-processing.com/demo/tag/	https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html
POS (tag y chunk)	POS (tag y chunk)	Etiquetas Eagles http://www.lsi.upc.edu/~nlp/tools/parole-sp.html
POS	http://bionlp-www.utu.fi/parser_demo	http://www2.lingsoft.fi/doc/engcg/intro/mtags.html

Freeling

<WORD form="." lemma="." pos="Fp"> <ANALYSIS lemma="." pos="Fp" prob="1"/> </WORD> </SENT> <SENT> <WORD form="Fed" lemma="fed" pos="NP00000"> <ANALYSIS lemma="fed" pos="NP00000" prob="1"/> </WORD> <WORD form="raises" lemma="rais" pos="NCMP000"> <ANALYSIS lemma="rais" pos="NCMP000" prob="1"/> </WORD> <WORD form="interest" lemma="interest" pos="NCMS000"> <ANALYSIS lemma="interest" pos="NCMS000" prob="1"/> </WORD> <WORD form="rates" lemma="rates" pos="NCMP000"> <ANALYSIS lemma="rates" pos="NCMP000" prob="1"/> </WORD> <WORD form="0.5" lemma="0.5" pos="Z"> <ANALYSIS lemma="0.5" pos="Z" prob="1"/> </WORD> <WORD form="percent" lemma="percent" pos="NCMS000"> <ANALYSIS lemma="percent" pos="NCCS000" prob="0.285567"/> <ANALYSIS lemma="percent" pos="NCFN000" prob="0.0713896"/> <ANALYSIS lemma="percent" pos="NCMN000" prob="0.0357009"/> <ANALYSIS lemma="percent" pos="NCMS000" prob="0.607342"/> </WORD>

Software.
Herramientas (la mayoría libres)

LinguaStream	MIT	JAVA	Libre para investigación	
NLTK	aprendizaje	Python	Reunión de herramientas	http://www.nltk.org/
Learning Based Java	Univ. Illinois	JAVA	BSD	http://cogcomp.cs.illinois.edu/page/tutorial.201310/
LingPipe	No open source	JAVA	Comercial	http://alias-i.com/lingpipe/
MII NLP	UCLA	JAVA	LGPL	http://www.mii.ucla.edu/research/areas/natural-language-processing/
RASP	U. Sussex	C++	LGPL	http://www.sussex.ac.uk/Users/johnca/rasp/
GATE	framework	JAVA	LGPL	http://gate.ac.uk/sale/tao
UIMA (Julie)	Community	JAVA	APACHE/ECLIPSE	https://uima.apache.org/
OPENNLP	Community	JAVA	APACHE/ECLIPSE	http://incubator.apache.org/opennlp
Standford NLP	Univ. Stanford	JAVA	GPL	http://nlp.stanford.edu/
Freeling	Univ. UPC (Lluis Padró)	C++	GPL	http://nlp.lsi.upc.edu/freeling/
Lucene	Comunidad Apache			http://lucene.apache.org/core/

NLP tareas y herramientas

NLP Proceso	NLTK	OpenNLP	LingPipe	Freeling	LUCENE (SOLR)	Stanford NLP
IDENTIFICA IDIOMA	(P)	✓	✓	✓	✓	(P)
PREPROCESO	(P)			✓		
POSTPROCESO						
TOKENIZACION	✓	✓	✓	✓	✓	✓
STEMMER	✓	✓	✓	✓	✓	✓
NER	(P)	✓	✓	✓		✓
LEMATIZACION	✓					
POS	✓	✓	(P)	✓	(P)	✓
CHUNKING (Shallow Parsing)	✓	✓	✓		(P)	✓
CORREFERENCIA		✓		✓		

(P) Parcialmente (p.e. programándolo o con cierta complicación)

La plataforma Gate permite integrar muchas de estas herramientas, particularmente interesantes para idioma no inglés, POS y NER

Framework GATE

- GATE: http://gate.ac.uk/sale/tao/
- Pipelines (Annie, OpenNLP, etc)
- Plugins para spanish, chino, hindi, …
- Anotación de expresiones regulares
- NER
- OpenNLP, Annie, Lingpipe, Lucene
- JAVA

OPEN NLP, LINGPIPE, GATE/ANNIE

Frase de ejemplo “Fed raises interest **rates** 0.5 percent”

OPEN NLP	LINGPIPE	ANNIE
<Node id="1244" />rates<Node id="1249" /> <Annotation Id="1595" Type="Token" StartNode="1244" EndNode="1249"> <Feature> <Name lassName="java.lang.String">orth</Name> <Value className="java.lang.String">lowercase</Value></Feature><Feature> <Name className="java.lang.String">category</Name> <Value className="java.lang.String">NNS</Value></Feature><Feature> <Name className="java.lang.String">chunk</Name> <Value className="java.lang.String">I-NP</Value></Feature><Feature> <Name className="java.lang.String">string</Name> <Value className="java.lang.String">rates</Value></Feature><Feature> <Name className="java.lang.String">length</Name> <Value className="java.lang.String">5</Value> </Feature> <Feature> <Name className="java.lang.String">stem</Name> <Value className="java.lang.String">rate</Value> </Feature> <Feature> <Name className="java.lang.String">kind</Name> <Value className="java.lang.String">word</Value> </Feature> </Annotation>	<Node id="1244" />rates<Node id="1249" /> <Annotation Id="2490" Type="Token" StartNode="1244" EndNode="1249"> <Feature> <Name className="java.lang.String">length</Name> <Value className="java.lang.String">5</Value> </Feature><Feature> <Name ClassName="java.lang.String">category</Name> <Value className="java.lang.String">nns</Value></Feature></Annotation>	<Node id="1244" />rates<Node id="1249" /> <Annotation Id="1595" Type="Token" StartNode="1244" EndNode="1249"> <Feature> <Name className="java.lang.String">length</Name> <Value className="java.lang.String">5</Value> </Feature><Feature> <Name ClassName="java.lang.String">category</Name> <Value className="java.lang.String">NNS</Value></Feature>

Peso de los términos

- Problemas
  - Habitualmente las palabras vacías son las más frecuentes. Hay que tener cuidado si se realizan filtrados por palabras vacías, ya que los resultados pueden variar mucho según las palabras que se filtren (¿qué ocurre si un documento tiene una palabra vacía sin filtrar mientras que los demás no?)
  - Pueden existir términos que aparezcan a lo largo de un documento y no sean representativos. Este hecho afecta a la recuperación por ese término en concreto, pero si se aplica esta medida, afecta a todos los términos. Ej. “página”

Text mining

Proceso de derivación de nueva información mediante técnicas estadísticas, ontologías y recursos linguísticos.

Permite

- Clasificación de Documentos: se utilizan naive Bayes, support vector machines (SVM) y máxima entropía. Todos ellos implementados en Weka o NLTK, por ejemplo para saber si:
  - Si es o no spam
  - Tema principal de un documento: Saber de que trata el documento: deportes, política, etc. Norm
- WSD (word sense disambiguation): reconocer el significado en un texto de una palabra polisémica.
- Clusters de términos en una colección, se utiliza LSA (Latent Semantic Analysis) y LDA (Latent Dirichlet Allocation).

Minería de textos. Preprocesamiento

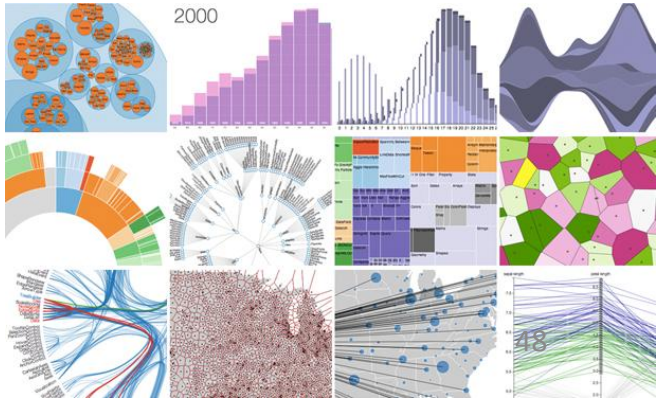
- Si se quiere incluir en un modelo dos atributos diferentes cuyos valores difieren se debe normalizar o estandarizar para que una variable no anule el valor de la otra.
  - Imagina un modelo en el que se tenga ejemplos de persona rica o pobre, y como atributos edad de la persona y dinero que tiene. La edad es un valor tan pequeño comparado con el dinero, que la edad no parecerá tener impacto.
- Normalizar
  - Transformar los valores de una variable para que oscilen dentro de un rango (normalmente entre cero y uno). También puede ser nombrar sus valores de una forma normalizada.
- Estandarizar
  - Trasformar los valores de una variable para que su media sea cero y sus diferencias sean distancias con respecto a la media, medido en desviaciones típicas

Minería de textos

- Análisis de regresión
  - Proceso para modelar la relación entre variables, creando una función que representa el comportamiento de las variables dependientes frente a una independiente.
  - {Ej. Salario=x*nivel_estudios+beta}
- Clasificación
  - Se trata de clasificar automáticamente datos a partir de un conjunto de atributos. Precisa ejemplos ya clasificados para aprender.
  - {Ej. Nacionalidad =“escocesa” si lleva_falda=“si” y sexo=“hombre”}
- Clustering
  - Descubrir conjuntos de datos (clusters) que se comportan de forma similar

Minería de textos

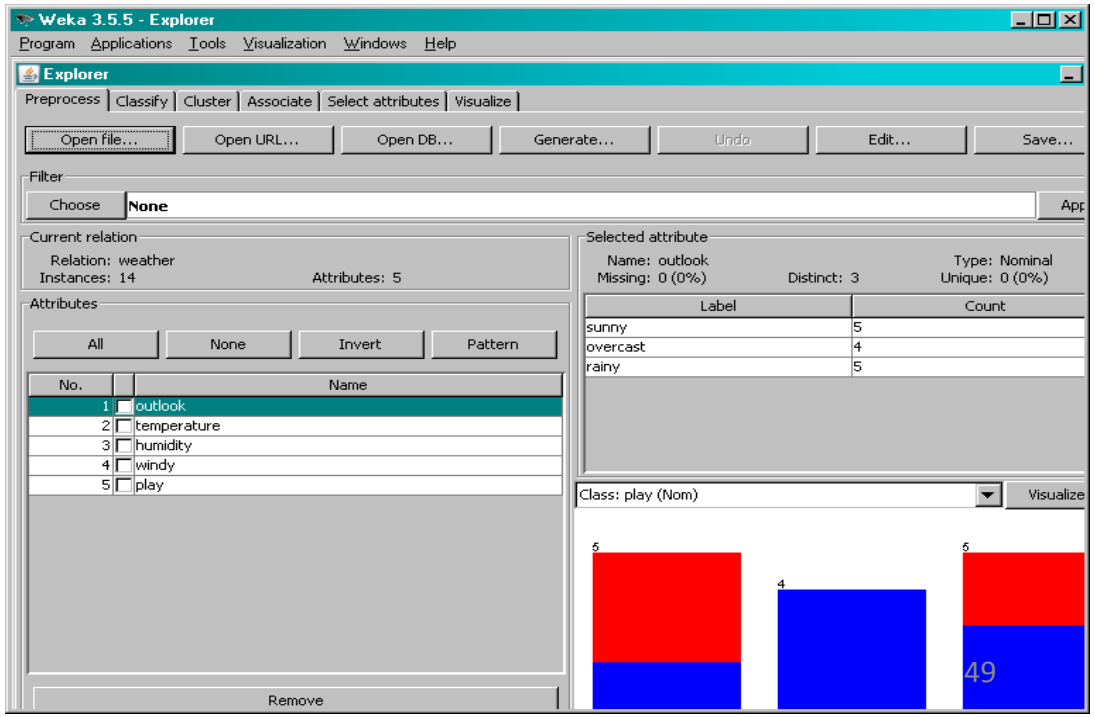
- Detección de anomalías
  - Consiste en detectar datos que se desvían de la tendencia marcada por los demás datos (outliers). Pueden ser errores o datos significativos. {p.e. edades >115 años}
- Reglas de asociación (dependencias),
  - sirven para encontrar correlaciones relevantes entre variables {expectativa de vida – cigarros al día}
- Resumen
  - Representación compacta de un conjunto de datos, incluye la visualización mediante diagramas



Weka Explorer

El explorador permite tareas de:

1. Preprocesado de los datos y aplicación de filtros.
2. Clasificación.
3. *Clustering.*
4. Búsqueda de Asociaciones.
5. Selección de atributos.
6. Visualización de datos.



Universidad
Carlos III de Madrid

Módulo VIII
Técnicas de Procesamiento de Lenguaje Natural (PLN)

Colaboradores

J.Morato, V.Palacios

J.Urbano, S.Sánchez-Cuadrado, M.Marrero