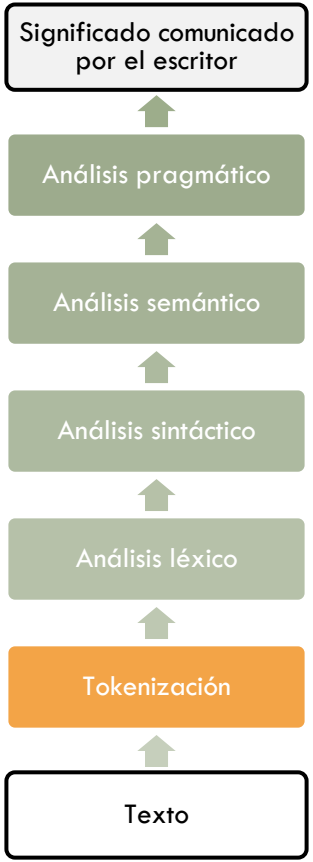
Análisis de textos:
Análisis de tokens

Fuente diapositivas: https://github.com/albarji/curso-analisis-textos

Caracteres y tokens

- Todo lenguaje escrito se conforma en base a unidades básicas
  - Caracteres: unidad mínima del lenguaje
    - A, i, ʊ, k, ñ, á, ö, α, ᘐ, あ, 国
  - Fonemas: articulación mínima de un sonido vocálico y consonántico
    - En el español: /a/, /k/, /č/ (ch), /ñ/
    - Al trabajar con texto escrito no se suelen utilizar
  - Palabras o tokens: agrupaciones de uno o varios caracteres
    - España, England, 日本
  - Frases: agrupaciones de palabras
- El primer paso de todo análisis del lenguaje será identificar en el texto estas unidades básicas: tokenización



Ejemplo de tokenización

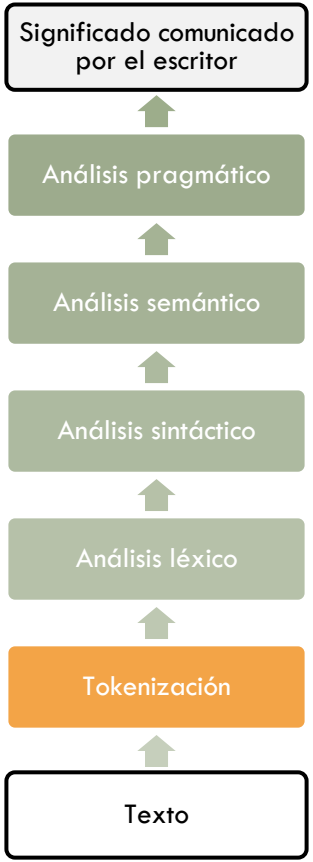
- Entrada
  - El veloz murciélago hindú comía feliz cardillo y kiwi. La cigüeña tocaba el saxofón detrás del palenque de paja.
- Salida
  - <SENTENCE>El<SEP>veloz<SEP>murciélago<SEP>hindú<SEP>comía<SEP>feliz<SEP>cardillo<SEP>y<SEP>kiwi<SEP>.<SENTENCE>La<SEP>cigüeña<SEP>tocaba<SEP>el<SEP>saxofón<SEP>detrás<SEP>del<SEP>palenque<SEP>de<SEP>paja<SEP>.<SENTENCE>

Tokenizando textos

Proceso de tokenización

- Entradas: texto como una sucesión de símbolos o caracteres, en formato digital
- Salidas
  - Agrupaciones de uno o más símbolos en palabras
  - Agrupaciones de una o más palabras en frases
- Ejemplo

  El veloz murciélago hindú comía feliz cardillo y kiwi. La cigüeña tocaba el saxofón detrás del palenque de paja
- ¡Fácil! Separar por espacios y puntos ☺



Tokenización: más difícil de lo que parece

- Abreviaturas, cantidades, signos de puntuación, …

El Sr. Gutierrez declaraba que el vehículo solo transportaba 20.5 kg entre equipajes, herramientas de trabajo, utensilios varios… el inspector del C.N.I. se mostraba receloso del asunto; no era la primera vez que veía un caso así.

- Lenguaje coloquial (y mal estructurado)

K bien lo pasamos dnd el david menudo colgao dtio sus colegas mas al próximo finde otra k no pare!!

- Idiomas sin marcadores de separación de palabras

昨日、吉森くんはクラスでいませんでした。病気になったかもしれません。。。ちょっと心配ですよ。

Tokenización: sistemas de escritura según caracteres

- Alfabéticos: cada símbolo representa (aproximadamente) un sonido
  - El veloz murciélago hindú comía cardillo y kiwi (Español)
  - The quick brown fox jumps over the lazy dog (Inglés)
  - Franz jagt im komplett verwahrlosten Taxi quer durch Bayern (Alemán)
  - דג סקרן שט בים מאוכזב ולפתע מצא לו חברה (Hebreo)
- Silábicos: cada símbolo representa una sílaba
  - ᏒᎠᎳᎰ*ᏉᏯᏁᏁᏧᎩᏌᏏᏢᏬᎷ (Cherokee)
  - □□□ □ □ □ □ □□□ □ □ □□□ □ (Vai)
- Logográficos: cada símbolo suele representar una idea o palabra
  - 敏捷的棕色狐狸 (Chino)
- En general los idiomas emplean símbolos de las tres clases
  - 13, %, ?, ¿, !, ¡, *, #, … (Símbolos logográficos en el español)

Tokenización: sistemas de escritura según separaciones

- Delimitados por espacios: cada palabra suele estar separada de otras por espacios (pero no siempre)
  - El veloz murciélago hindú comía cardillo y kiwi (Español)
  - The quick brown fox jumps over the lazy dog (Inglés)
- No segmentados: no existe ningún caracter separador entre palabras
  - 敏捷的棕色狐狸 (Chino)
- Las aproximaciones a seguir son diferentes en cada caso

- A continuación nos centramos en idiomas con escritura alfabética delimitada por espacios

Tokenización en idiomas delimitados por espacios

Tokenización base
Dividir el texto en tokens usando los espacios + reglas básicas de puntuación

Desarrollo de abreviaturas
Convertir abreviaturas en sus equivalentes desarrollados

Desarrollo de contracciones y multipart-words
Separar contracciones del idioma y palabras aglutinadas en varias palabras

Detección de multiwords
Unir palabras que conforman expresiones hechas, o que se convierten en entidades diferentes si se separan

Splitting
Agrupación de los tokens en frases

(1) Tokenizador base: ejemplo basado en reglas


```
<Macros>
ALPHA           [[:alpha:]]
ALPHANUM    [[:alnum:]°°]
SYMNUM      [\.:,_/\\=\*\+\-°°&$€£¢¥#%]
OTHERS      .
</Macros>
<Abbreviations>
aa.rr.
abr.
abrev.
a.c.
adj.
adm.
</Abbreviations>
```

```
<RegExps>
INDEX_SEQUENCE  0  (\.{4,}|-{2,}|\*{2,}|_{2,}|/{2,})
INITIALS1       1  ([A-Z](\.[A-Z])+)(\.\.\.)
INITIALS2          0  ([A-Z]\.)+
TIMES           0  (([01]?[0-9]|2[0-4]):[0-5][0-9])
NAMES_CODES   0  ({ALPHA}|{SYMNUM})*[0-9]({ALPHA}|[0-9]|{SYMNUM}+{ALPHANUM})*
THREE_DOTS      0  (\.\.\.)
QUOTES                0  (``|<<|>>|'')
APOSTR_CAT      1  ([dD]')({ALPHA})
MAILS                 0
{ALPHANUM}+([\._]{ALPHANUM}+)*@{ALPHANUM}+([\._]{ALPHANUM}+)*
URLS1                 0  ((mailto:|(news|http|https|ftp|ftps)://)[\w\.\-]+|^(www(\.[\w\-
]+)+))
URLS2           1  ([\w\.\-]+\.(com|org|net))[\s]
KEEP_COMPOUNDS  0  {ALPHA}+(['_\-\+]{ALPHA}+)+
*ABREVIATIONS1  0  (({ALPHA}+\.)+)(?!\.\.)
WORD            0  {ALPHANUM}+[\+]*
OTHERS_C        0  {OTHERS}
</RegExps>
```

Padró, L., & Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality.

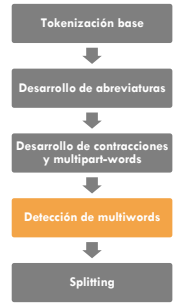
(2) Desarrollo de abreviaturas

- Objetivo: convertir abreviaturas en sus equivalentes desarrollados
  - Los desarrollos deben ser palabras reconocidas por el diccionario o "estándar", de forma que los análisis posteriores puedan procesarlas
  - Desarrollar abreviaturas ayuda al proceso de splitting posterior, porque abreviaturas como "Sr." pasan a ser "Señor" y se evita cometer un error al separar frases por ese punto
- Procedimiento
  - Búsqueda y reemplazo mediante listas de abreviaturas conocidas y sus desarrollos
    - Ej: c\ → calle, Av. → Avenida, Sr. → Señor, UAM → Universidad Autónoma de Madrid
- Ambigüedad
  - Dependiendo de la temática los desarrollos pueden cambiar
    - (IP → Internet Protocol) VS (IP → Intellectual Property)
    - Es necesario tener listados de abreviaturas especializados según la aplicación

(3) Desarrollo de contracciones y multipart-words

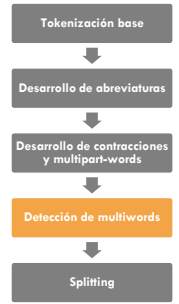
- Objetivo: expandir contracciones del idioma y palabras compuestas
  - Contracciones del idioma: it's → it is, del → de el
  - Multipart-words: bias-variance → bias variance, Boston-based → Boston based
- Procedimiento
  - Búsqueda y reemplazo mediante listas de contracciones y multipart-words conocidas y sus desarrollos
  - Cuidado especial con textos donde el carácter '-' se usa para continuar una palabra al final de la línea, y no debe confundirse con una multipart-word

(4) Detección de Multiwords



- Se considera Multi-Word Expression todo aquel conjunto de 2 o más palabras que, como grupo, tienen características que sus componentes individuales no tienen, pudiendo ser alguna de las siguientes:
  - Léxicas: la multiword se forma de algunos lexemas que no pueden aparecer en solitario.
    - Ej: ad hoc, de facto
  - Sintácticas: el PoS de la multiword no es el que tendría normalmente de hacer el análisis sintáctico de sus componentes
    - Ej: take a walk (VERBO, aunque debería ser SV)
  - Semánticas: el significado de la multiword no puede derivarse de los significados de sus componentes
    - Ej: ser la leche, pasar de castaño oscuro, liarla parda, romper el hielo
  - Pragmáticas: la multiword cobra un sentido especial bajo un contexto determinado, cosa que sus componentes individuales no
    - Ej: buenos días (dicho por la mañana implica un saludo)
  - Estadísticas: los componentes de la multiword aparecen con mucha más frecuencia en grupo que por separado, o con más frecuencia que otras expresiones con igual significado
    - Ej: toma y daca, Castilla y León, miedo cerval, sal y pimienta (vs pimienta y sal)

(4) Detección de Multiwords: métodos



- Método de diccionario
  - Mantener un diccionario de multiwords, creado por un experto, donde se indiquen los componentes de cada multiword y el PoS o semántica que adopta la multiword en su conjunto
  - Usar el diccionario para marcar multiwords en los textos
- Método de frecuencias (apoyo al diccionario)
  - Detectar grupos de palabras contiguas que aparecen con una frecuencia significativa en un corpus
  - Analizar manualmente los grupos detectados y añadir los que sean multiwords al diccionario
- Métodos basados en clasificadores
  - Usar corpus anotados con multiwords para entrenar un clasificador que indique si un grupo de palabras contiguas es una multiword, y si es así, cuál es su PoS o semántica
  - Como características suelen usarse las palabras del grupo siendo evaluado así como palabras vecinas

(5) Splitting: agrupación en frases

- Una vez se han obtenido los tokens (palabras) es necesario agrupar estos en frases
- Regla simple: separar por puntos
  - ¡Pero no funciona en todos los casos!

¿Os habéis preguntado por qué separar por puntos no funciona? Tal vez… ¡porque no siempre separamos frases por puntos! Las abreviaturas como 2.3 Kg tampoco separan frases (como Uds. sabrán), así que el tema es complejo ←

- Solución clásica el problema: reglas en forma de expresiones regulares, que para cada símbolo sospechoso de segmentación hagan uso de
  - Mayúsculas/minúsculas
  - Longitud de las palabras cercanas
  - Caracteres habituales de fin de palabras, prefijos, sufijos de palabras cercanas
  - Listas de abreviaciones

Ejemplo de splitter sencillo


```
<Markers>
" "
( )
{ }
/* */
</Markers>
```

```
<SentenceEnd>
. 0
? 0
! 0
</SentenceEnd>
<SentenceStart>
¿
¡
</SentenceStart>
```

Padró, L., & Stanilovsky, E. (2012). Freeling 3.0: Towards wider multilinguality.

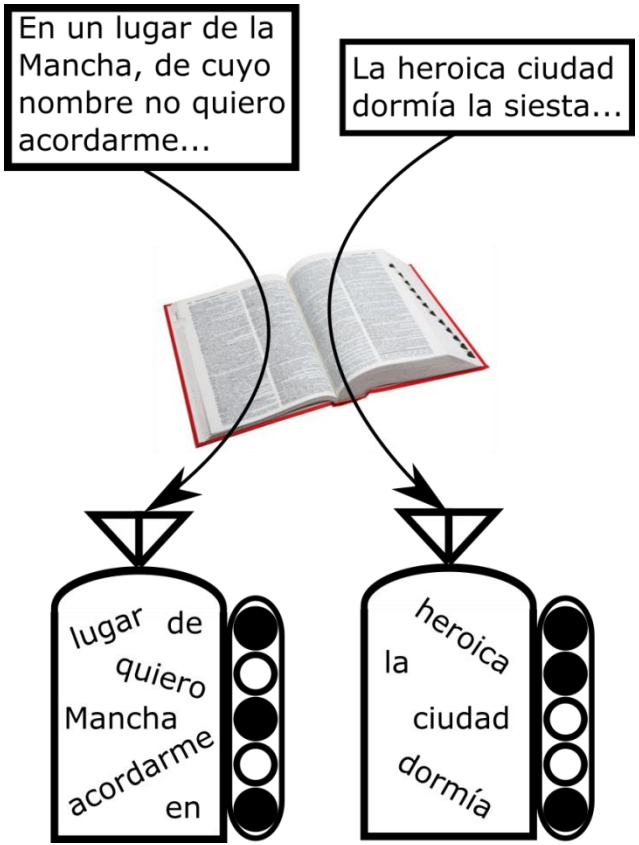
Tokenización basada en Machine Learning

- Alternativamente al proceso anterior, si se dispone de un corpus de suficiente texto en el idioma y dominio del problema a resolver, puede intentarse una aproximación basada en Machine Learning
  - El corpus debe estar etiquetado con caracteres especiales que indiquen separadores de palabra y separadores de frase
- Construir un modelo que dado un carácter y sus vecinos lo clasifique como carácter normal, separador de palabra o separador de frase
- El modelo puede enriquecerse con características como las empleadas en la aproximación clásica
  - Modelo híbrido

Extrayendo características del texto tokenizado

Bag of Words

- Las palabras que se utilizan en un texto dan mucha información
  - Temática del texto: vocabulario específico al tema, tecnicismos, …
  - Autor: nivel cultural, región, género, edad, …
  - Sentimientos del autor (levemente): emociones, opinión sobre el tema tratado…
- Bag of Words
  - Crear un diccionario con todas las palabras de un corpus referencia: N
  - Para cada texto a procesar, construir un vector binario de N entradas que refiere a cada palabra del diccionario, y toma el valor:
    - 1 si la palabra correspondiente del diccionario está presente en el texto
    - 0 si la palabra correspondiente no está presente en el texto



Ejemplo: grupos de palabras distintivas

1

- codec
- solenoid
- golem
- mach
- humvee
- claymore
- scimitar
- kevlar
- paladin
- bolshevism
- biped
- dreadnought

2

- taffeta
- tresses
- bottlebrush
- flouncy
- mascarpone
- decoupage
- progesterone
- wisteria
- taupe
- flouncing
- peony
- bodice

Guent University vocabulary test. http://vocabulary.ugent.be/wordtest/start

Ejemplo: palabras fuertemente indicativas del género

- codec (88%)
- solenoid (87%)
- golem (89%)
- mach (93%)
- humvee (88%)
- claymore (87%)
- scimitar (86%)
- kevlar (93%)
- paladin (93%)
- bolshevism (85%)
- biped (86%)
- dreadnought (90%)

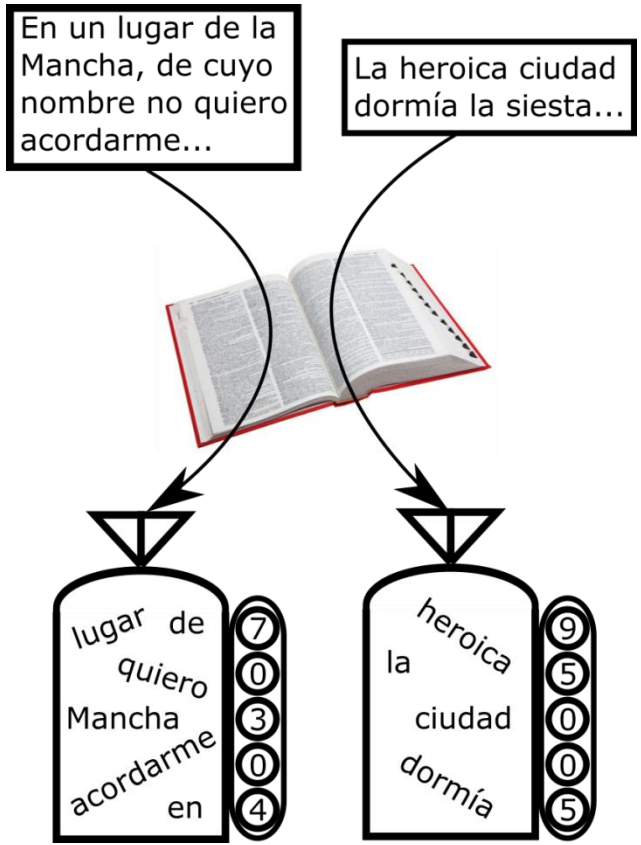
- taffeta (48%)
- tresses (61%)
- bottlebrush (58%)
- flouncy (55%)
- mascarpone (60%)
- decoupage (56%)
- progesterone (63%)
- wisteria (61%)
- taupe (66%)
- flouncing (67%)
- peony (70%)
- bodice (71%)

Guent University vocabulary test. http://vocabulary.ugent.be/wordtest/start

Vector de frecuencias

- Análogo al vector de frecuencias de caracteres, pero empleando tokens

- Rendimiento similar a usar Bag of Words
  - No es tan importante cuántas veces se usa una palabra, solo si se usa o no



Problemas de bag of words

	de	gallina	huevos	la	los
los huevos de la gallina	X	X	X	X	X
la gallina de los huevos	X	X	X	X	X

	baño	el	en	me	río
me baño en el río	X	X	X	X	X
me río en el baño	X	X	X	X	X

- Bag of Words desecha la estructura del texto original, perdiendo significado
  - Relaciones sustantivo – modificador
  - Relaciones de causalidad
  - Imposible realizar desambiguaciones de palabras con varios significados

n-gramas

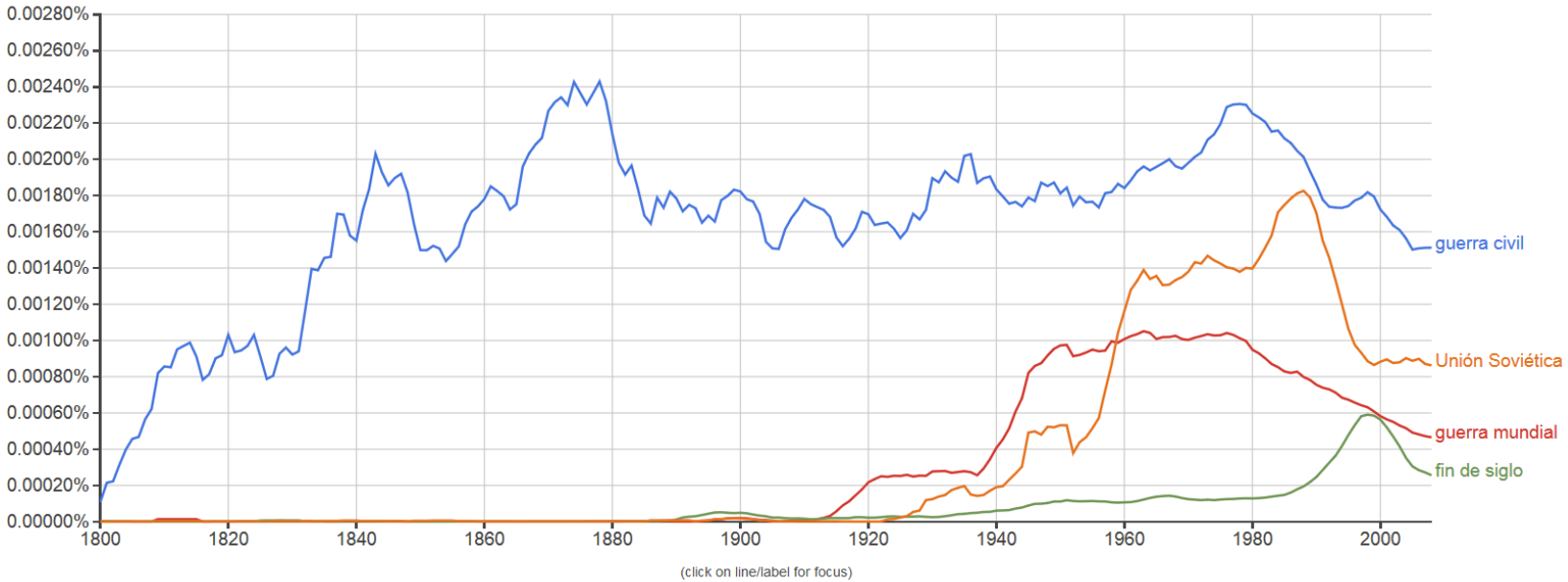
- Bag of Words o frecuencias de conjuntos de n tokens adyacentes
- Normalmente menos costoso que n-gramas de caracteres
  - En un diccionario de W palabras, la gran mayoría de las posibles $O(W^n)$ combinaciones de tokens nunca aparece en el corpus
- Permite mantener información parcial del contexto de cada token
  - Si esto es importante, puede mejorar a Bag of Words

	de la	de los	gallina de	huevos de	la gallina	los huevos
los huevos de la gallina	X			X	X	X
la gallina de los huevos		X	X		X	X

Google books Ngram Viewer

Graph these comma-separated phrases: guerra civil,guerra mundial,fin de siglo,Unión Soviética case-insensitive

between 1800 and 2008 from the corpus Spanish (2009) with smoothing of 3 . Search lots of books



- Google n-gram viewer: https://books.google.com/ngrams
- Corpus de n-gramas de Google: http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

skip-gramas

- Calcular n-gramas para n grande es muy costoso, debido a todas las combinaciones posibles que pueden existir en el texto
  - Para un idioma de W palabras posibles, existen $O(W^n)$ posibles n-gramas
  - Esto impide capturar con n-gramas las relaciones a larga distancia entre palabras
- k-skip-n-gramas: grupos de n palabras no consecutivas, en los que se saltan k palabras entre cada componente del n-grama
  - Representan relaciones a larga distancia de forma muy eficiente
  - Ej: 1-skip-2-gramas

	los de	huevos la	de gallina	la de	gallina los	de huevos
los huevos de la gallina	X	X	X			
la gallina de los huevos				X	X	X

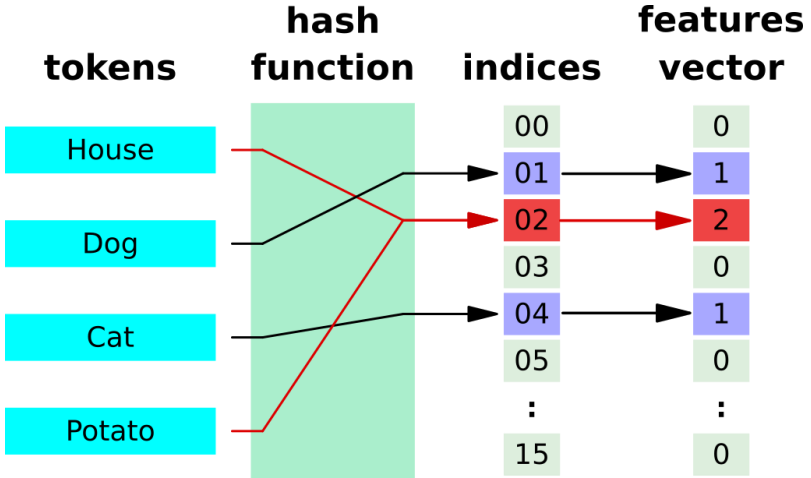
Un problema: palabras fuera del vocabulario

- Un problema importante de los métodos basados en diccionario (bag of words, n-gramas) las representaciones vectoriales se construyen en base a un diccionario fijo de palabras.
- Si más adelante nos encontramos con palabras antes no vistas, no tenemos forma de codificarlas correctamente como vectores, por lo que es necesario recalcular el diccionario.

	de la	de los	gallina de	huevos de	la gallina	los huevos
los huevos de la gallina	X			X	X	X
la gallina de los huevos		X	X		X	X
el zorro rojo						

Hashing Trick

- Una solución que evita rehacer el diccionario es el Hashing Trick
- Un texto se representa como un vector de características de longitud fija (ej. 1024) , en formato disperso (todo a 0s por defecto)
- Cada token del texto se pasa por una función de hash que devuelve un índice en el vector. Esa entrada del vector se incrementa.



✔ Esto permite obtener representaciones que aceptan palabras fuera del vocabulario, y que además son independientes del idioma.

✘ Existe el riesgo de que 2 tokens diferentes colisionen en el mismo índice. Para minimizar estas colisiones, se suelen emplear vectores de características muy grandes: $2 \cdot 10^6$

Filtrado de stop-words

- Las palabras más frecuentes en un idioma son palabras funcionales, que no dan significado por sí mismas
  - Ley de Zipf: la frecuencia de cualquier palabra es inversamente proporcional a su posición en una lista ordenada por frecuencia

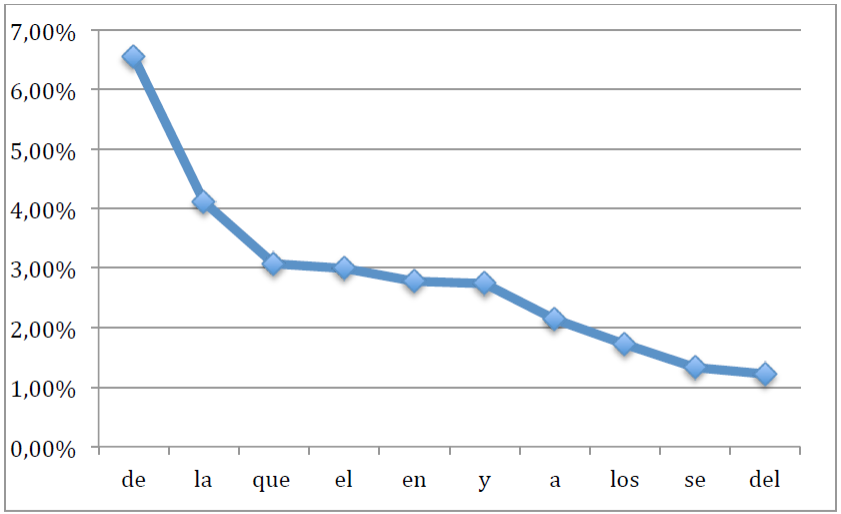


Figura 3.2. Curva gráfica de las 10 primeras palabras del CREA

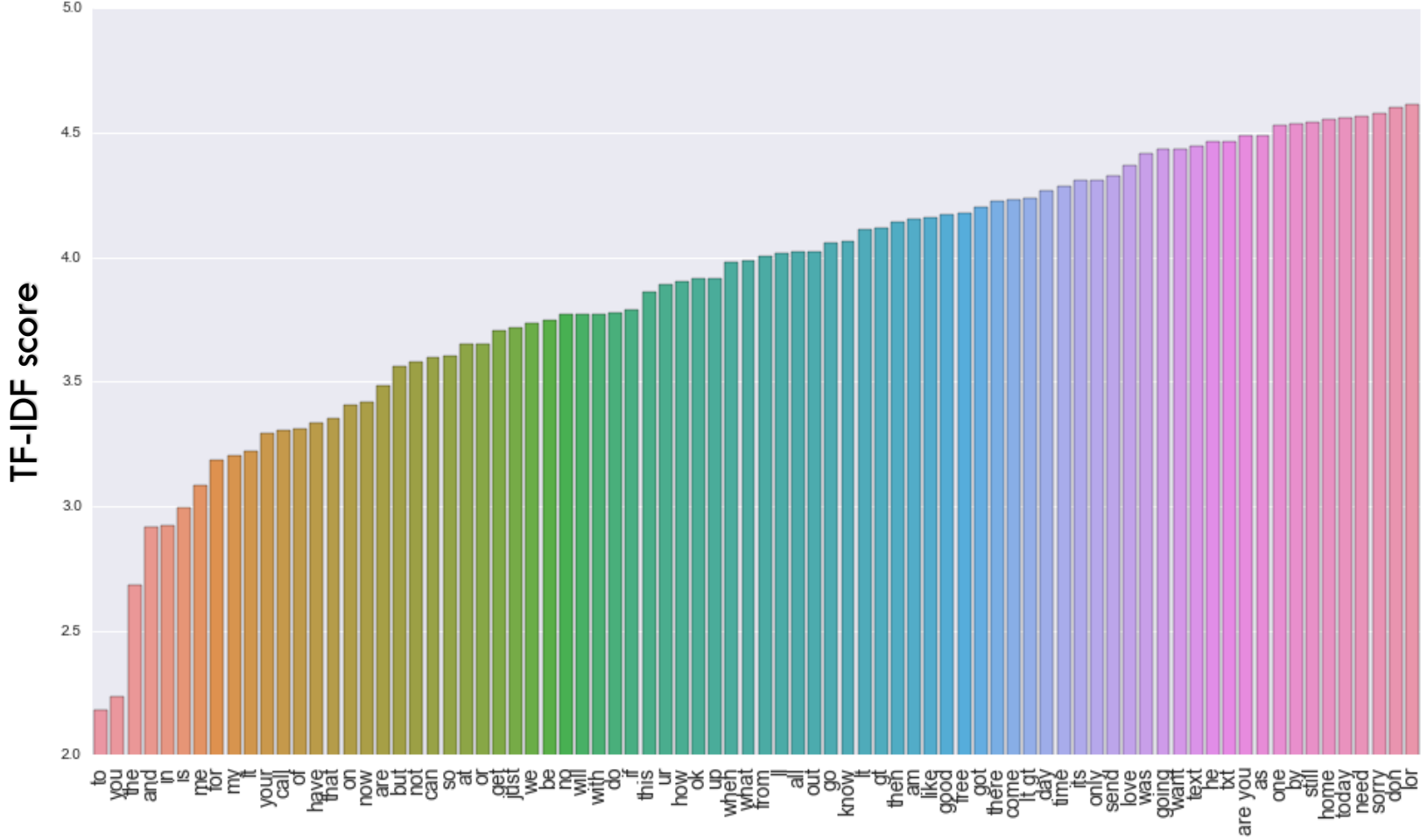
- Eliminar las stop-words (palabras funcionales) ayuda a quitar información superflua
  - La eliminación puede hacerse con un simple listado de las palabras más frecuentes del idioma
- En aplicaciones donde es importante no solo lo que se dice sino cómo se dice puede ser útil mantener las stop-words
  - Ej: identificación de autores

Moreno Sandoval, A (en prensa): Lenguas y Computación. Madrid, Síntesis.

TF-IDF

- Una forma estadística y automatizada de detectar palabras no relevantes es pesando el valor de cada palabra mediante Term Frequency – Inverse Document Frequency (TF-IDF)
  - Term Frequency: número de veces que aparece una palabra en un documento (normalizado por la longitud del documento)
  - Inverse Document Frequency: inversa a la proporción de documentos del corpus en los que aparece una palabra. Se escala con el logaritmo, conforme a la ley de Zipf.
- Matemáticamente, el TF-IDF de una palabra w en un documento d de un corpus D es:

$$TFIDF(w, d, D) = \frac{|w \in d|}{|d|} \cdot \log \frac{|D| + 1}{|\{d \in D : w \in d\}| + 1}$$


TF-IDF score
5.0
4.5
4.0
3.5
3.0
2.5
2.0
to
you
the
and
in
is
me
for
my
it
your
call
of
have
that
on
now
are
but
not
can
so
at
or
get
just
we
be
no
will
with
do
if
this
ur
how
ok
up
when
what
from
ll
all
out
go
know
it
gt
then
am
like
good
free
got
there
come
lt gt
day
time
its
only
send
love
was
going
want
text
he
txt
are you
as
one
by
still
home
today
need
sorry
don
lor

Riqueza de vocabulario (Type Token Ratio)

- Los métodos anteriores generan un gran número de características que intentan representar todo el texto
- También es posible extraer métricas globales: por ejemplo, medir el número de palabras diferentes que es capaz de usar el autor
  - Indicador de la riqueza de vocabulario
  - También aporta indicios sobre el nivel cultural y de la edad

$$TTR = \frac{\#\text{palabras diferentes del texto}}{\#\text{palabras del texto}}$$

Clasificación de documentos

- Objetivo: dado un documento, identificar qué tema trata, dentro de una lista de temas definidos
  - Problema de clasificación multiclase, y posiblemente multietiqueta
- Características útiles para el problema
  - Palabras: n-gramas de palabras
  - Morfología: bag-of-words de sustantivos
  - Semántica: bag-of-words de sustantivos según significado, relaciones de sinonimia, word2vec
- Clasificadores empleados
  - Naive Bayes
  - Support Vector Machines
  - Redes neuronales y LSTMs
  - Árboles de clasificación

Detección de spam

- Objetivo: dado un correo electrónico, determinar si es correo legítimo (ham) o fraudulento (spam)
  - Problema de clasificación binario, con texto como datos
- Características útiles para el problema
  - Palabras: bag-of-words ("viagra", "lottery"), n-gramas con n pequeño
  - Caracteres: n-gramas
- Clasificadores empleados
  - Naive bayes
- Recursos
  - Enron spam dataset: http://www.aueb.gr/users/ion/data/enron-spam/
  - Trec spam dataset: http://trec.nist.gov/data/spam.html

Detección de spam: ejemplo

llo rwxr scobox a4 shujaat satws

All of our products are exact gen/eric equivalants of the well known brand

name drugs, just like the lbuprofen you bvy at the grocery store is the

gen>eric version of Advil. Our gen+eric pha\rmaceuticals are made in a

state-of-the-art laboratory using the exact same top-grade ingredients as

the name b@rands.

V/ic0-din(named as Parac_odin), Su+perVia!gra Chea`p

Juf http://ewerthebestrx.info/track.asp?cg=great&c=tools

You should at least give it a try especially when we offer large disc#ounts

for bulk orders and make incr~edible deals for returning customers.

we had one common thought or fancy all that way, or whether our eyes, which

yet were formed upon the

Listed below are some guidelines for gathering the necessary information for NETCO contract setup so we can get as much done as possible prior to our start date. I suggest each manager start making the phone calls and if necessary get others at the desk to help out. Again, the goal is to get hard copies of the contracts in our hands for review so we can fill out all of the necessary information ahead of time. Take good notes on the questions that come up when speaking with the pipes. We can get legal involved as needed when major issues surface.

Please let me know if you have any further questions, tks - Bob