Text Mining

Marcelo Mendoza
http://www.inf.utfsm.cl/~mmendoza
mmendoza@inf.utfsm.cl
A 131, Campus San Joaquín - UTFSM

¿De qué se trata la asignatura?

Descripción

Esta asignatura provee una introducción sistemática a una amplia gama de técnicas que permiten analizar texto. Estas técnicas permiten detectar patrones en el texto para ayudar a descubrir información útil para la toma de decisiones.

¿De qué se trata la asignatura?

Descripción

Esta asignatura provee una introducción sistemática a una amplia gama de técnicas que permiten analizar texto. Estas técnicas permiten detectar patrones en el texto para ayudar a descubrir información útil para la toma de decisiones.

Objetivos

Capacitar al estudiante en los temas fundamentales de text mining. Al aprobar la asignatura el estudiante será capaz de:

1. Comprender los fundamentos de text mining.
2. Aplicar técnicas de text mining para analizar colecciones de texto.
3. Diseñar técnicas de text mining para satisfacer nuevos requerimientos de análisis.

¿De qué se trata la asignatura?

Unidades temáticas

Fundamentos de text mining : modelos de palabras (unigram, bigram, n-grams), collocations, POS tags y NER, elementos de teoría de información para texto (entropía, entropía conjunta y condicional, divergencias Kullback-Leibler y Jensen-Shannon, perplejidad de modelos).

Modelos de tópicos : Latent Semantic Analysis, probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation (LDA), extensiones a LDA.

Representación de texto basado en aprendizaje automático: skip-grams, CBOW, GloVe, FastText, ElMo, GPT-1, GPT-2, BERT, extensiones a BERT.

Procesamiento de texto con redes neuronales: modelos de lenguaje, sequence labeling, seq2seq, context free parsing, dependency parsing.

Tareas NLP: Sematic role labeling, reference resolution, textual entailment, question-answering, machine translation, dialog systems, evaluación en NLP.

Temas avanzados: procesamiento de texto basado en grafos, meta-learning para NLP.

¿De qué se trata la asignatura?

Descripción

1. Clases expositivas con apoyo de medios audiovisuales.
2. Desarrollo de ejercicios en clases que permitirán ilustrar los conceptos del área.
3. Proyecto semestral.

¿De qué se trata la asignatura?

Descripción

1. Clases expositivas con apoyo de medios audiovisuales.
2. Desarrollo de ejercicios en clases que permitirán ilustrar los conceptos del área.
3. Proyecto semestral.

Evaluación

1. 2 certámenes (individual, tipo cuestionario aula).
2. 2 presentaciones de papers (individual, en clases).
3. 1 proyecto semestral (individual o en grupos de hasta dos personas).

Calificación

- Promedio certámenes: 30%
- Promedio papers: 30%
- Proyecto: 40%

Bibliografía

- Goldberg, Y. Neural Network Methods for Natural Language Processing, In Synthesis on Human Language Technologies, Morgan & Claypool, 2017.
- Eisenstein, J. Introduction to Natural Language Processing, MIT Press, 2019.

¿De qué se trata la asignatura?

Bibliografía

- Goldberg, Y. Neural Network Methods for Natural Language Processing, In Synthesis on Human Language Technologies, Morgan & Claypool, 2017.
- Eisenstein, J. Introduction to Natural Language Processing, MIT Press, 2019.

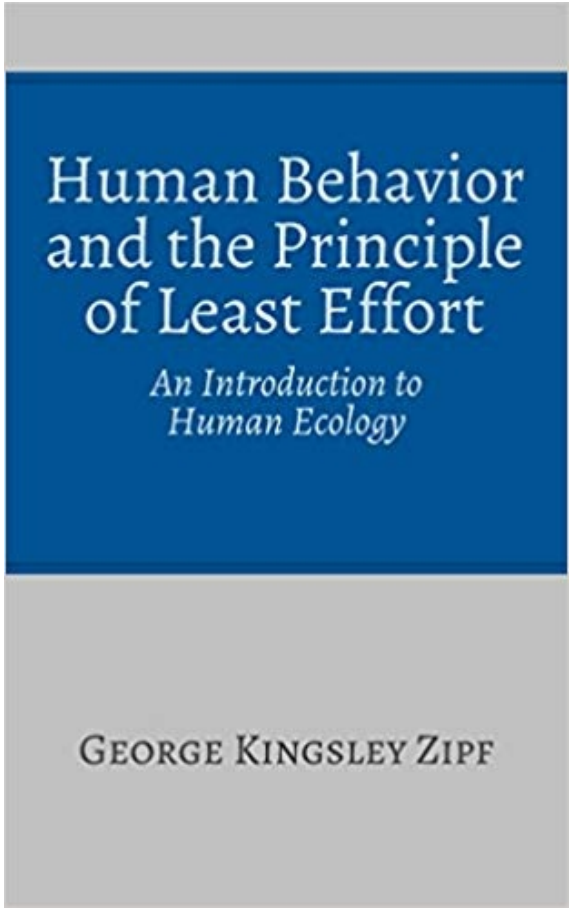
Planificación

- Inicio semestre: 30 de Agosto. Cierre de semestre: Jueves 30 de Diciembre.
- Días libres: 13 a 17 de Septiembre, 20 a 22 de Octubre, 23 a 27 de Diciembre.
- Fechas certámenes: C1 (19 de Octubre), C2 (15 de Diciembre).
- Presentaciones de papers, en clases: P1 (12-13 de Octubre), P2 (30 de Noviembre – 1 de Diciembre)
- Presentación de propuesta de proyecto semestral: 3 de Noviembre.
- Presentación de cierre de proyecto semestral: 22 de Diciembre.

Tema 1

Fundamentos de text mining

Leyes del texto



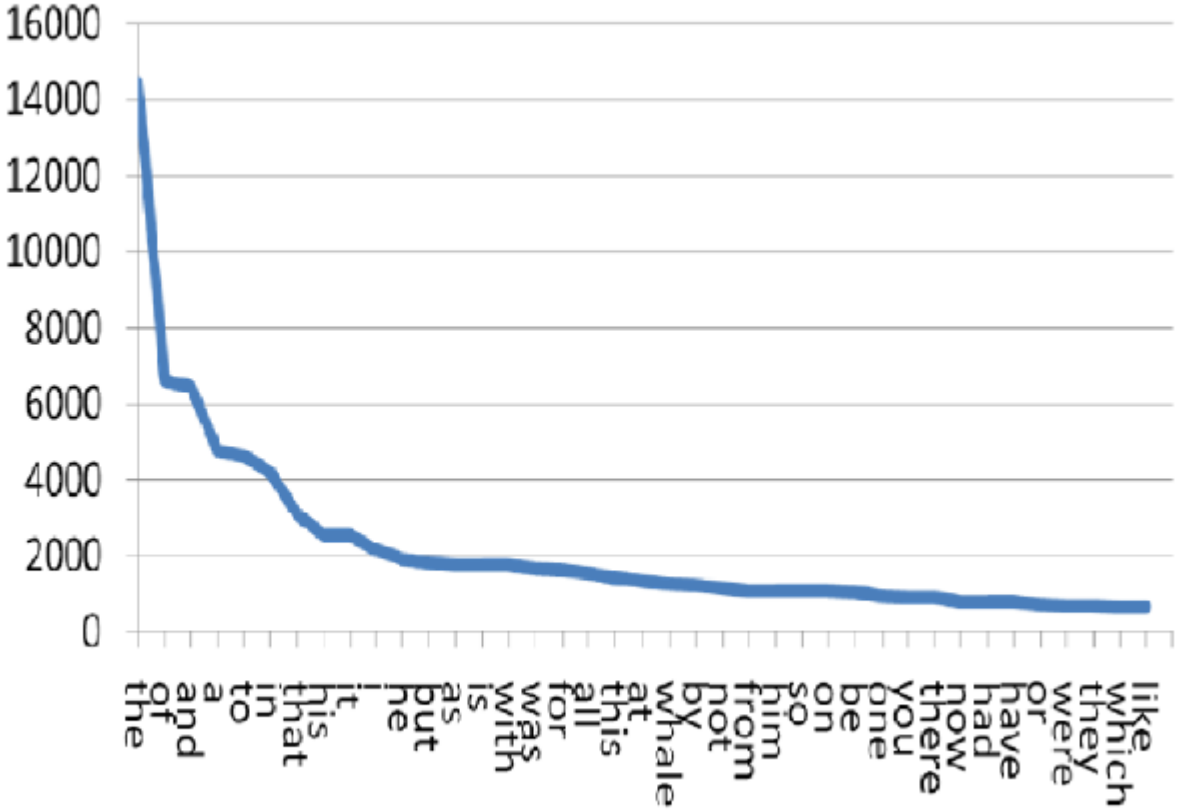
George Kingsley Zipf (1949), Human behavior and the principle of least effort, Addison-Wesley Press

Leyes del texto

[1] Agencia de noticias

Leyes del texto

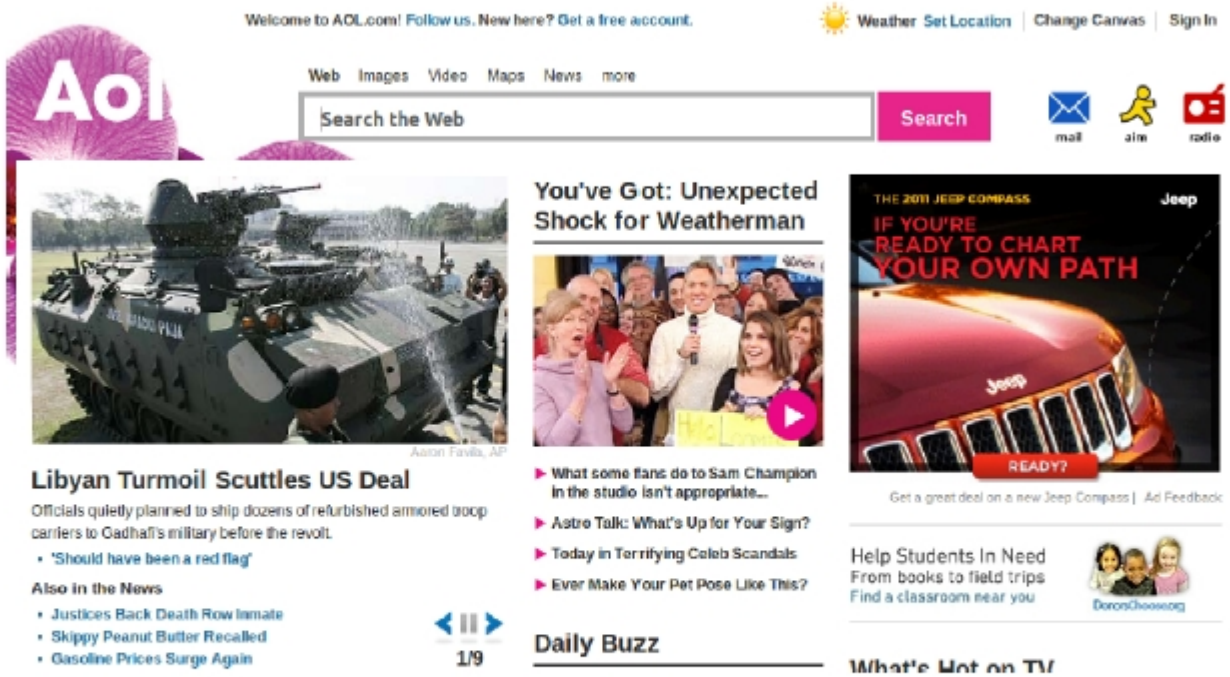
Zipf para Reuters[2]



[2] Dataset de noticias, disponible on-line

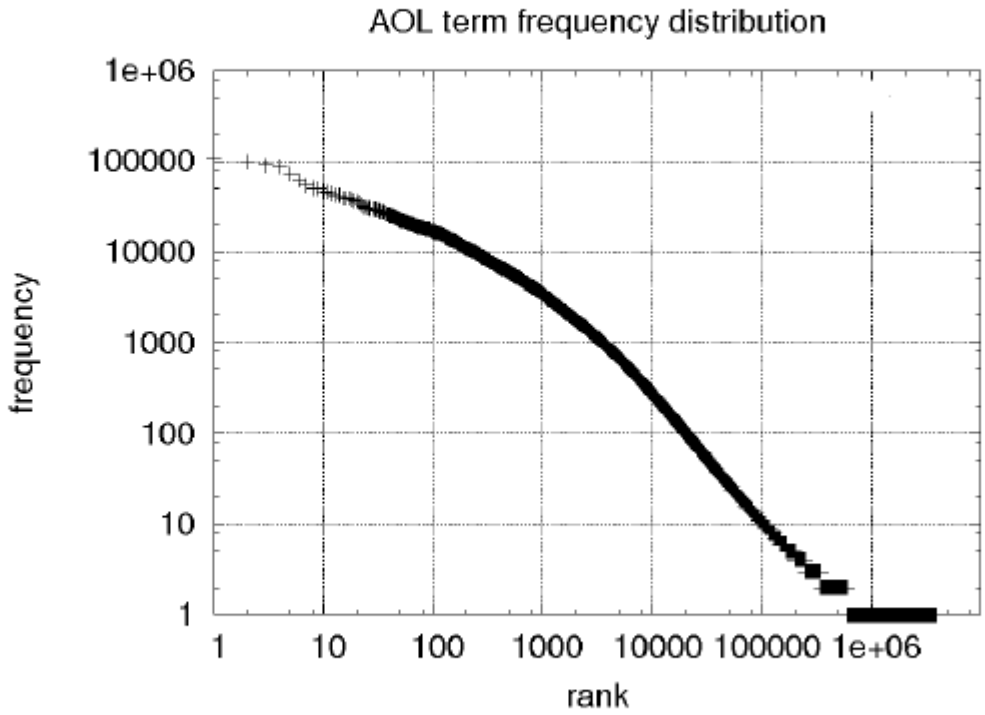
Leyes del texto

AOL[3]



[3] sitio con autenticacion, America On-Line

Zipf para AOL query log[4]



[4] Dataset de consultas formuladas a AOL, disponible on-line

Leyes del texto

Word	Freq. (f)	Rank (r)	$f \cdot r$	Word	Freq. (f)	Rank (r)	$f \cdot r$
the	3332	1	3332	turned	51	200	10200
and	2972	2	5944	you'll	30	300	9000
a	1775	3	5235	name	21	400	8400
he	877	10	8770	comes	16	500	8000
but	410	20	8400	group	13	600	7800
be	294	30	8820	lead	11	700	7700
there	222	40	8880	friends	10	800	8000
one	172	50	8600	begin	9	900	8100
about	158	60	9480	family	8	1000	8000
more	138	70	9660	brushed	4	2000	8000
never	124	80	9920	sins	2	3000	6000
Oh	116	90	10440	Could	2	4000	8000
two	104	100	10400	Applausive	1	8000	8000

Producto $f \cdot r$ en el libro *Tom Sawyer*, versión en inglés.

Leyes del texto

Ley de Zipf: $$f \sim \frac{1}{r}$$

θ : pendiente de la curva log-log

n : # tokens

r : ranking de la palabra

f : # ocurrencias de la palabra

Leyes del texto

Ley de Zipf:

$$f \sim \frac{1}{r}$$

$$\rightarrow \quad f \sim \frac{1}{r^{\theta}}$$

θ : pendiente de la curva log-log

n : # tokens

r : ranking de la palabra

f : # ocurrencias de la palabra

Leyes del texto

Ley de Zipf:

$$f \sim \frac{1}{r}$$

$$f \sim \frac{1}{r^{\theta}}$$

$$f_r = \frac{n}{r^{\theta} \cdot H_V(\theta)}$$

θ : pendiente de la curva log-log

n : # tokens

r : ranking de la palabra

f : # ocurrencias de la palabra

Leyes del texto

Ley de Zipf: $f \sim \frac{1}{r}$

$$f \sim \frac{1}{r^{\theta}}$$

$$f_r = \frac{n}{r^{\theta} \cdot H_V(\theta)}$$

θ : pendiente de la curva log-log

n : # tokens

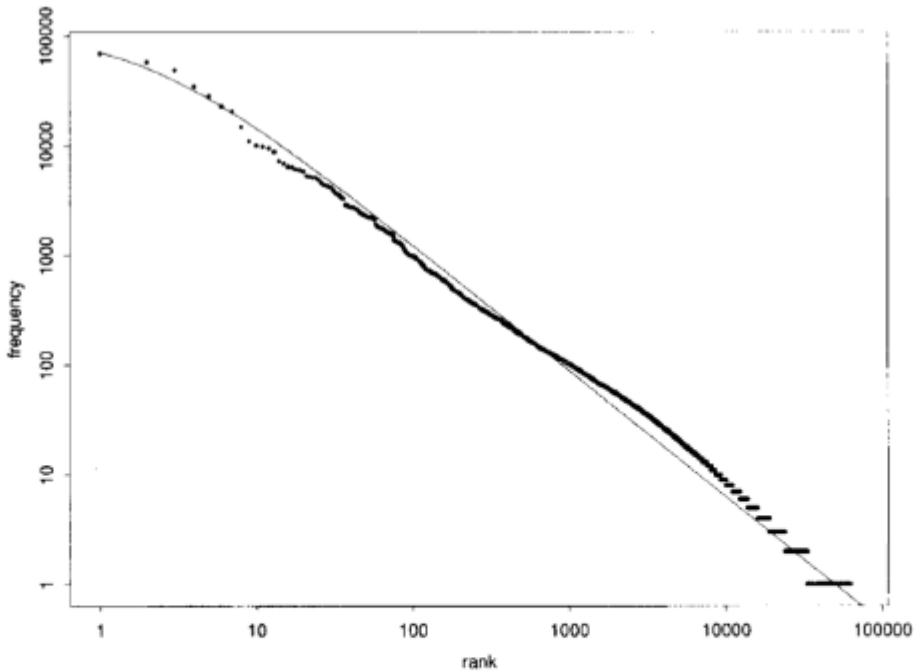
r : ranking de la palabra

f : # ocurrencias de la palabra

Si $\theta \approx 1 \rightarrow H_V(\theta) = \log(n)$

$$H_V(\theta) = \sum_{j=1}^{V} \frac{1}{j^{\theta}}$$

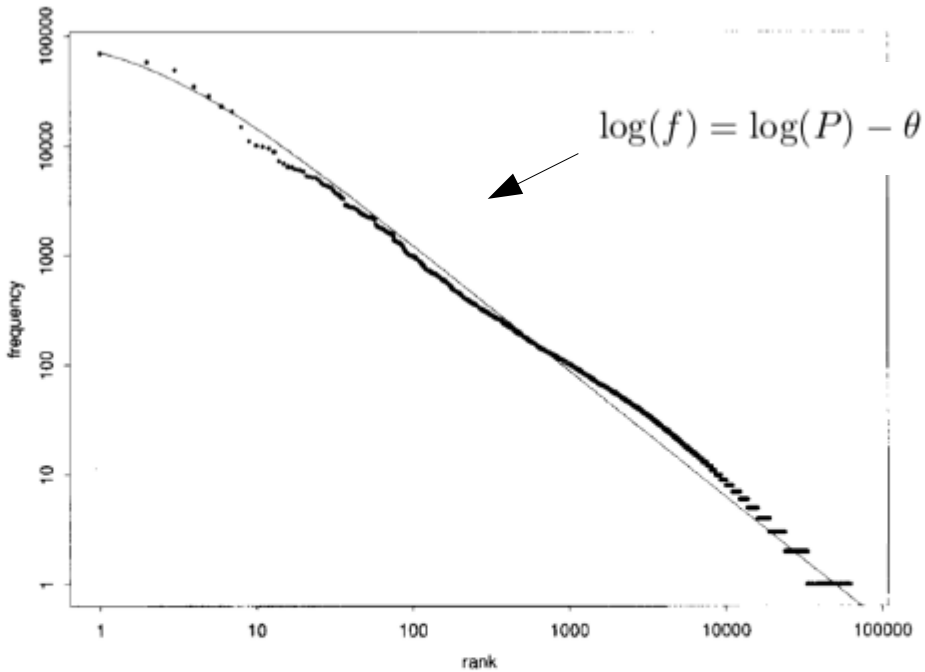
Leyes del texto



Ajuste Mandelbrot en el corpus *Brown* [5].

[5] The Brown Corpus was the first million-word electronic corpus of English, created in 1961 at Brown University. This corpus contains text from 500 sources, and the sources have been categorized by genre, such as news, editorial, and so on

Leyes del texto



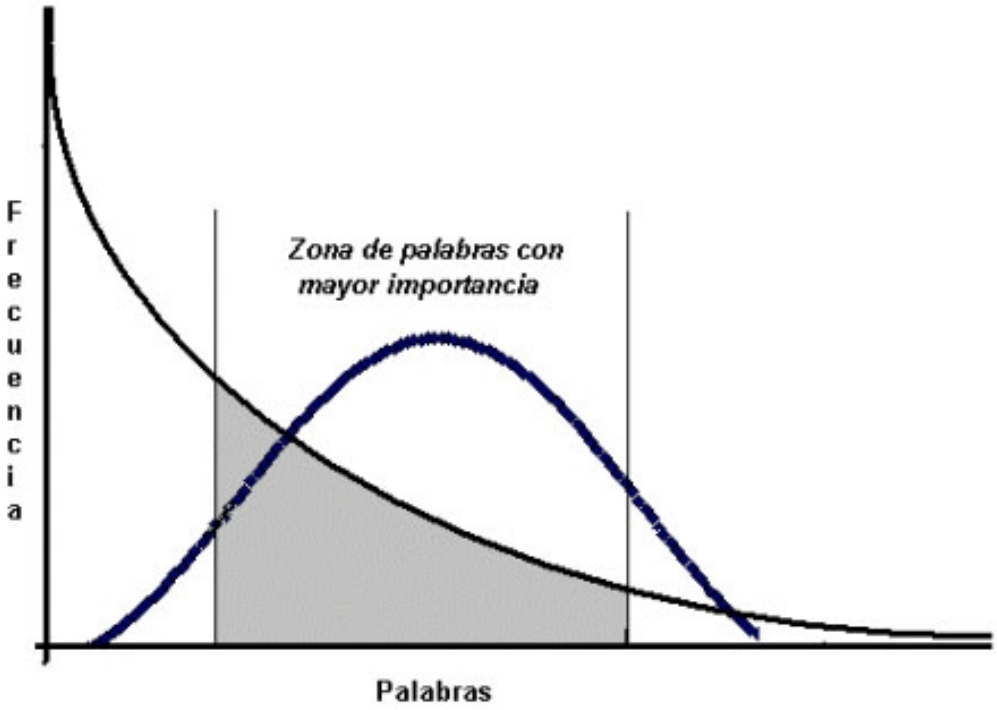
$$\log(f) = \log(P) - \theta \cdot \log(r + \rho)$$

$$P \approx 10^5 \quad \rho \approx 10^2$$

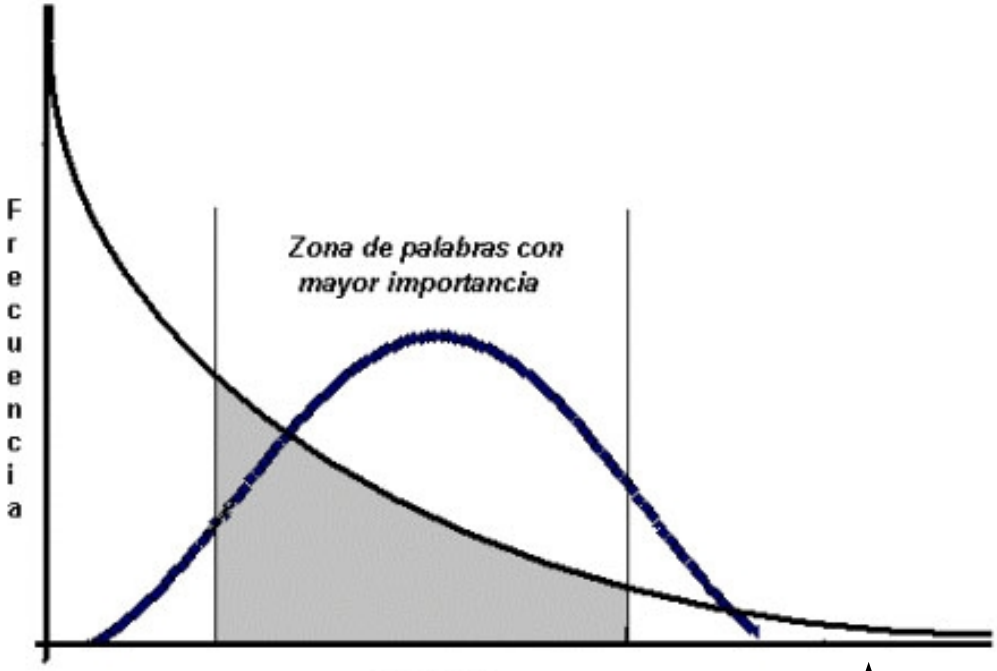
Ajuste Mandelbrot en el corpus *Brown* [5].

[5] The Brown Corpus was the first million-word electronic corpus of English, created in 1961 at Brown University. This corpus contains text from 500 sources, and the sources have been categorized by genre, such as news, editorial, and so on

Keywords versus stopwords



Keywords versus stopwords



stopwords

keywords

outliers (e.g. typos)

Leyes del texto

ACM DL DIGITAL LIBRARY

Latin American Consortium
Universidad Tecnica Federico Santa Maria

SIGN IN SIGN UP

SEARCH

H S Heaps

No contact information provided yet.

Authors:
Add personal information

Bibliometrics: publication history

Publication years	1978-1978
Publication count	1
Citation Count	71
Available for download	0
Downloads (6 Weeks)	0
Downloads (12 Months)	0

SEARCH

Search Author's Publications

ROLE

- **Author only**

FEEDBACK

AUTHOR PROFILE PAGES (BETA)
Project background

1 search result

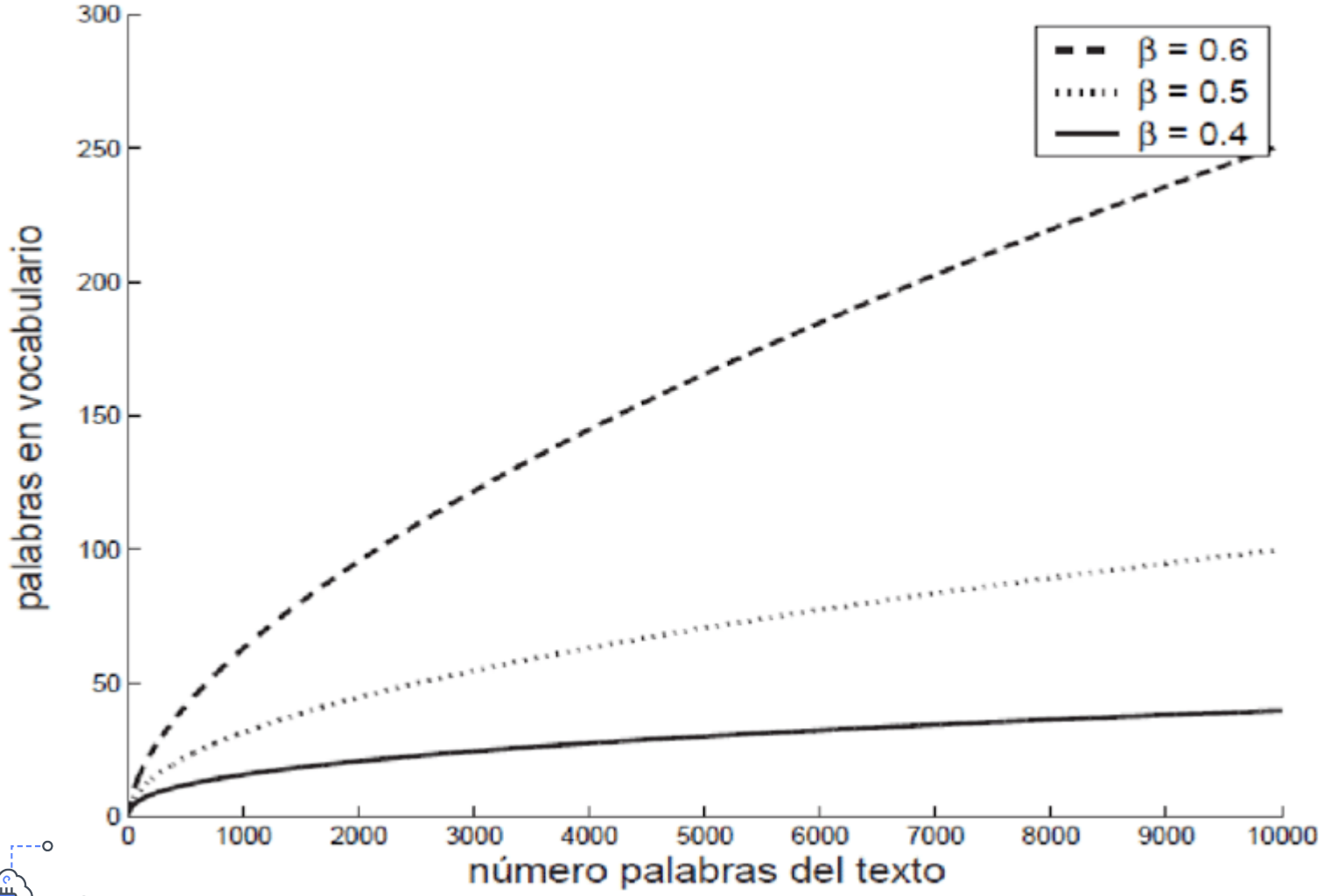
Export results as: BibTeX EndNotes ACM Ref

1978

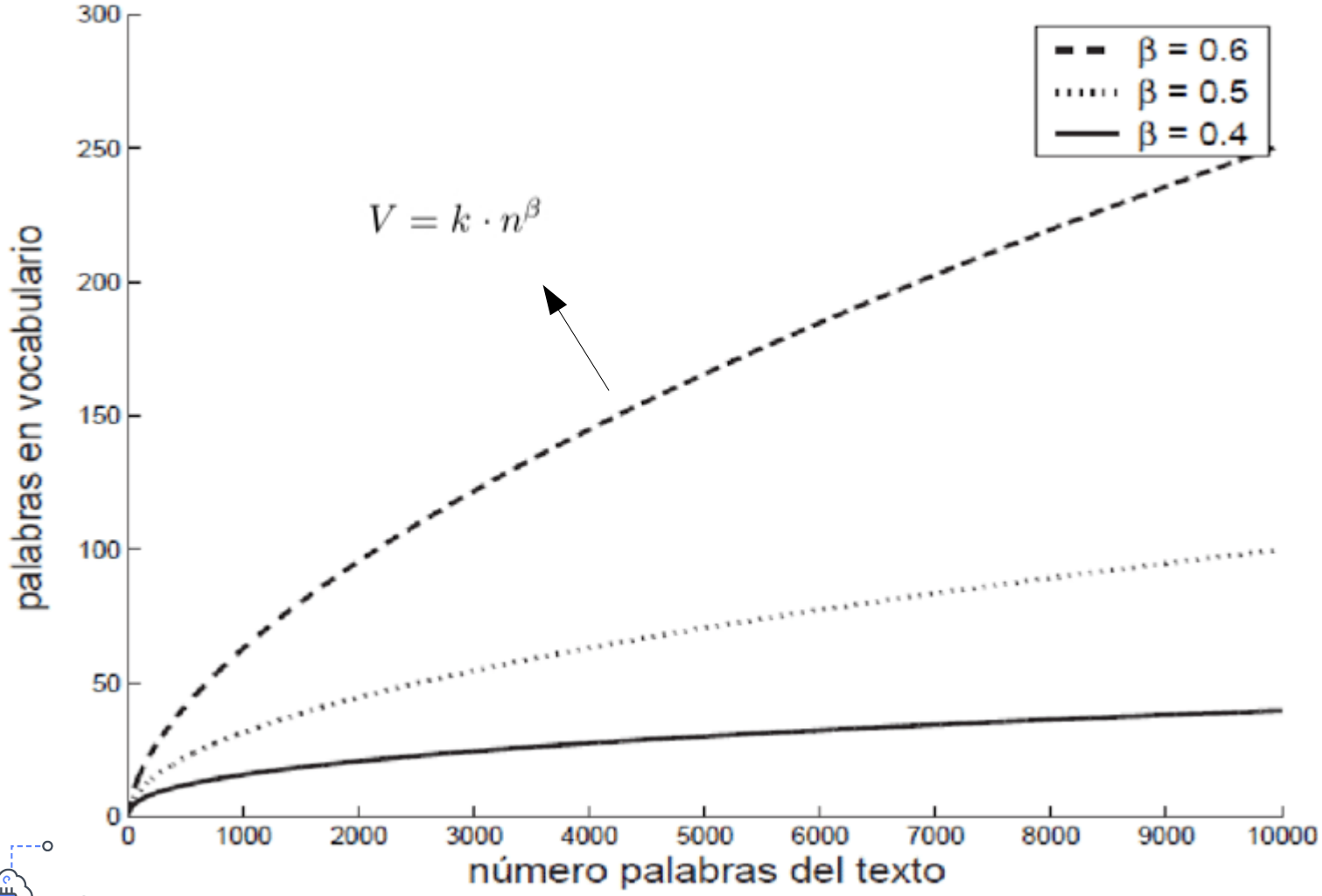
1 Information Retrieval: Computational and Theoretical Aspects
H. S. Heaps
November 1978 Information Retrieval: Computational and Theoretical Aspects
Publisher: Academic Press, Inc.
Additional Information: full citation, cited by

Bibliometrics: Downloads (6 Weeks): n/a, Downloads (12 Months): n/a, Citation Count: 71

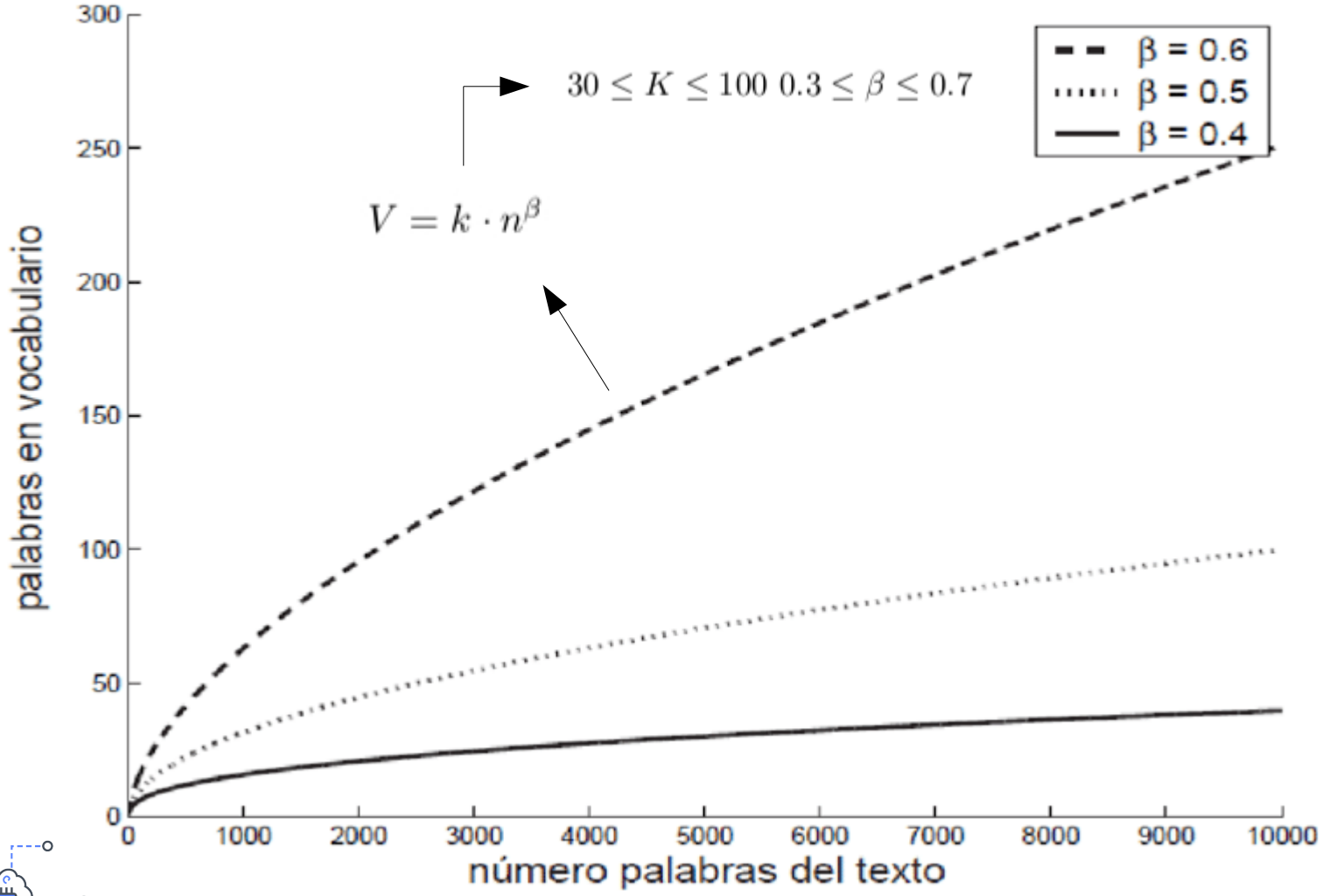
Leyes del texto



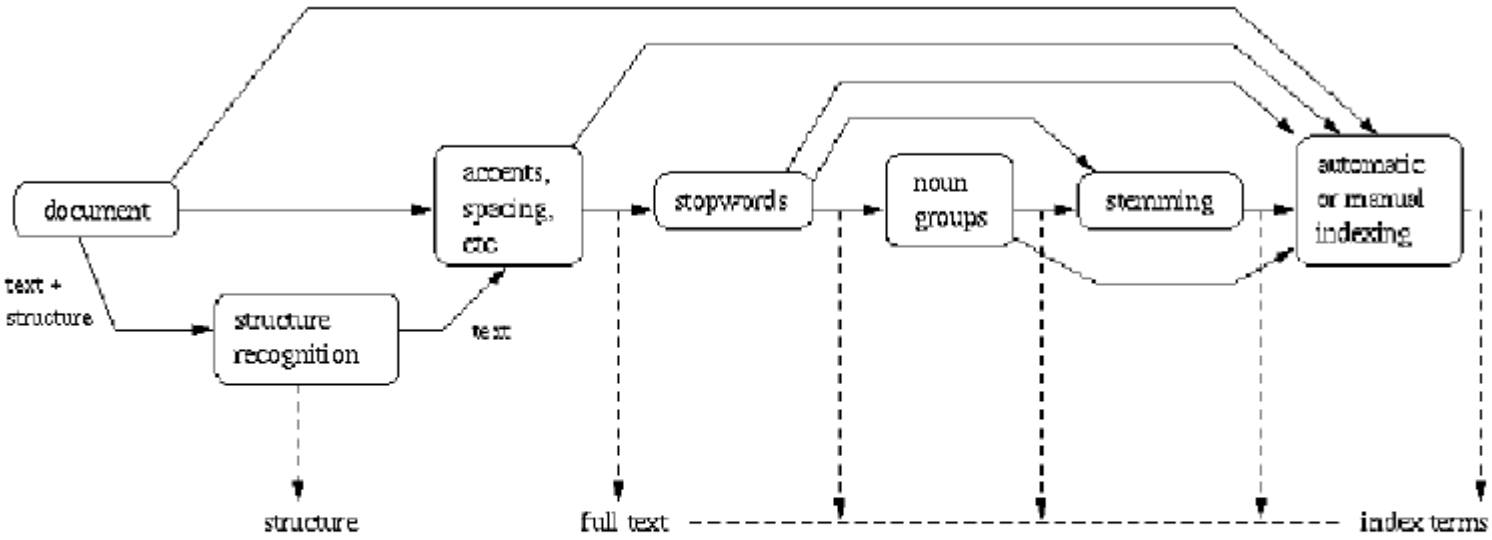
Leyes del texto



Leyes del texto



Procesamiento de texto



[6]Ref.: R. Baeza & B. Ribeiro, Modern Information Retrieval, 1999.

Índice invertido:

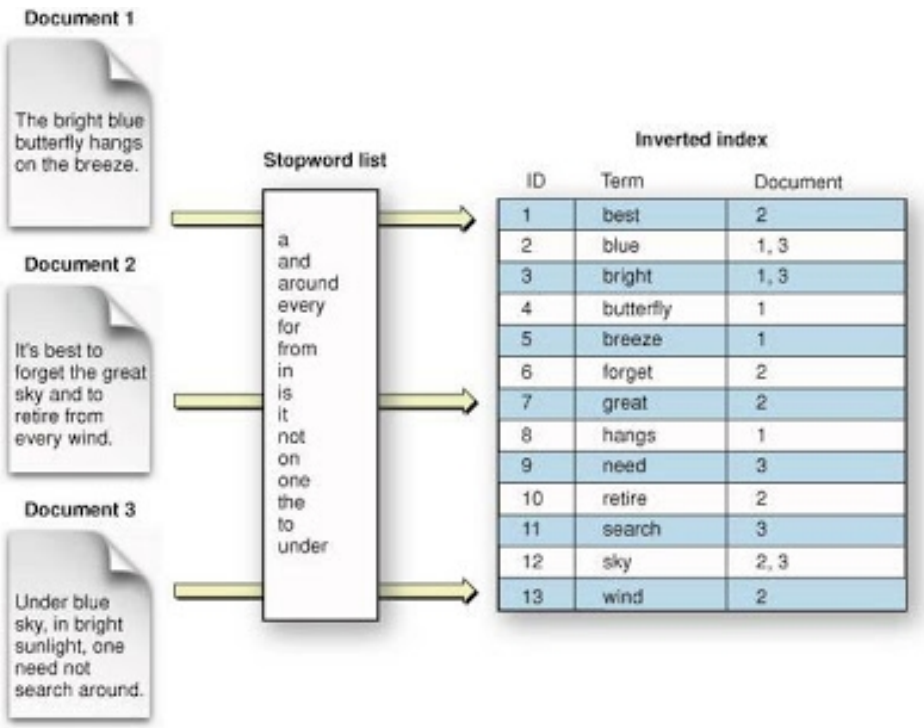
Brutus	$\longrightarrow$	1	2	4	11	31	45	173	174
Cesar	$\longrightarrow$	1	2	4	5	6	16	57	132	...
Calpurnia	$\longrightarrow$	2	31	54	101				

$\vdots$

vocabulario **posteo**

Procesamiento de texto

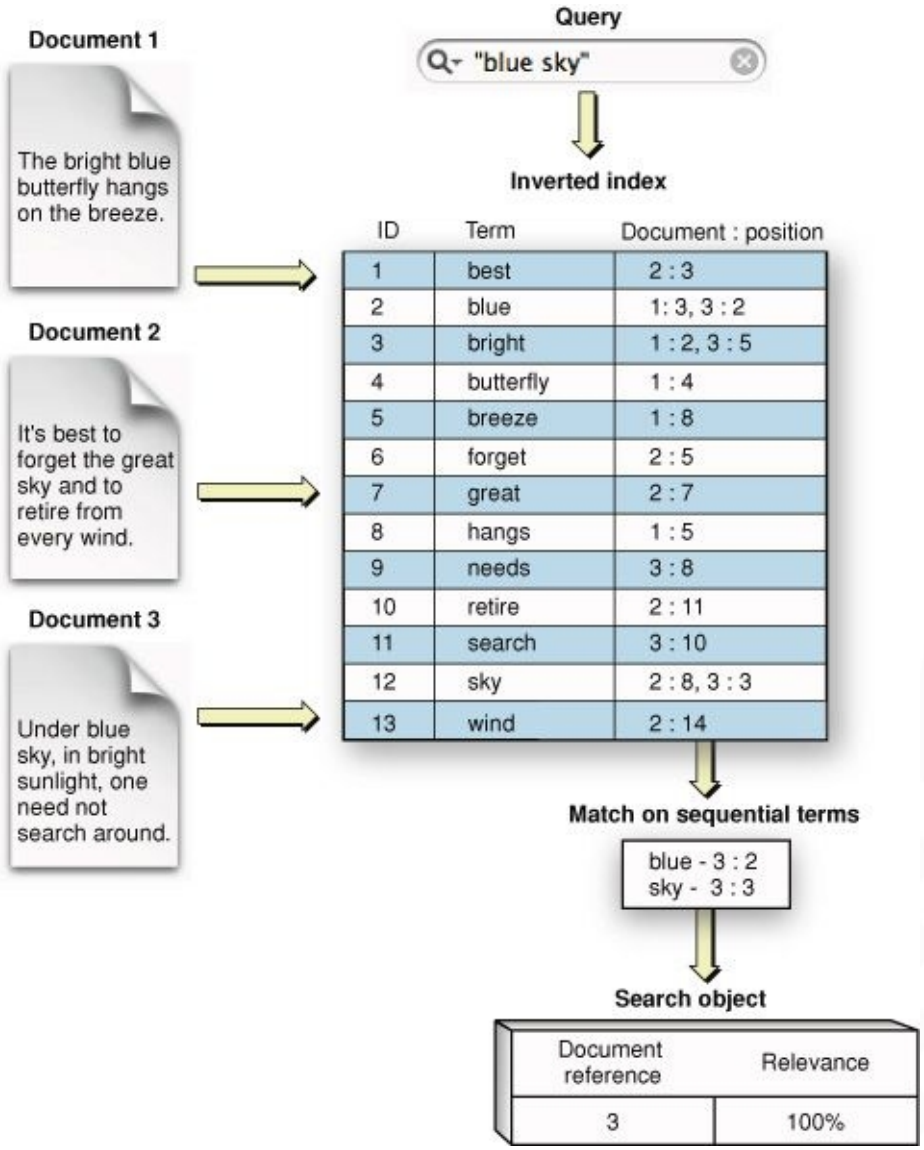
Índice invertido:



ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

Procesamiento de texto

Índice invertido:



ID	Term	Document : position
1	best	2 : 3
2	blue	1: 3, 3 : 2
3	bright	1 : 2, 3 : 5
4	butterfly	1 : 4
5	breeze	1 : 8
6	forget	2 : 5
7	great	2 : 7
8	hangs	1 : 5
9	needs	3 : 8
10	retire	2 : 11
11	search	3 : 10
12	sky	2 : 8, 3 : 3
13	wind	2 : 14

Procesamiento de texto, diferencias entre idiomas

和尚

Chino

ノーベル平和賞を受賞したワンガリ・マータイさんが名誉会長を務めるＭＯＴＴＡＩＮＡＩキャンペーンの一環として、毎日新聞社とマガジンハウスは「私の、もったいない」を募集します。皆様が日ごろ「もったいない」と感じて実践していることや、それにまつわるエピソードを８００字以内の文章にまとめ、簡単な写真、イラスト、図などを添えて１０月２０日までにお送りください。大賞受賞者には、５０万円相当の旅行券とエコ製品２点の副賞が贈られます。

Japonés

استقلت الجزائر في سنة 1962 بعد 132 عاما من الاحتلال الفرنسي.

← → ← → ← START

"Algeria achieved its independence in 1962 after 132 years of French occupation."

Árabe

Procesamiento de texto

	Stopwords
A	a, about, again, all, almost, also, although, always, among, an, and, another, any, are, as, at
B	be, because, been, before, being, between, both, but, by
C	can, could
D	did, do, does, done, due, during
E	each, either, enough, especially, etc
F	for, found, from, further
H	had, has, have, having, here, how, however
I	i, if, in, into, is, it, its, itself
J	just
K	kg, km
M	made, mainly, make, may, mg, might, ml, mm, most, mostly, must
N	nearly, neither, no, nor
O	obtained, of, often, on, our, overall
P	perhaps, pmid
Q	quite
R	rather, really, regarding
S	seem, seen, several, should, show, showed, shown, shows, significantly, since, so, some, such
T	than, that, the, their, theirs, them, then, there, therefore, these, they, this, those, through, thus, to
U	upon, use, used, using
V	various, very
W	was, we, were, what, when, which, while, with, within, without, would

[7] http://www.pubmed.gov

Procesamiento de texto

a, acá, ahí, ajena, ajenas, ajeno, ajenos, al, algo, alguna, algunas, alguno, algunos, algún, allá, allí, aquel, aquella, aquellas, aquello, aquellos, aquí, cada, cierta, ciertas, cierto, ciertos, como, cómo, con, conmigo, consigo, contigo, cualquier, cualquiera, cualquieras, cuan, cuanta, cuantas, cuánta, cuántas, cuanto, cuantos, cuán, cuánto, cuántos, de, dejar, del, demasiada, demasiadas, demasiado, demasiados, demás, el, ella, ellas, ellos, él, esa, esas, ese, esos, esta, estar, estas, este, estos, hacer, hasta, jamás, junto, juntos, la, las, lo, los, mas, más, me, menos, mía, mientras, mío, misma, mismas, mismo, mismos, mucha, muchas, muchísima, muchísimas, muchísimo, muchísimos, mucho, muchos, muy, nada, ni, ninguna, ningunas, ninguno, ningunos, no, nos, nosotras, nosotros, nuestra, nuestras, nuestro, nuestros, nunca, o, os, otra, otras, otro, otros, para, parecer, poca, pocas, poco, pocos, por, porque, que, qué, quien, quienes, quienesquiera, quienquiera, quién, si, siempre, sí, sín, Sr, Sra, Sres, Sta, suya, suyas, suyo, suyos, tal, tales, tan, tanta, tantas, tanto, tantos, te, tener, ti, toda, todas, todo, todos, tomar, tuya, tuyo, tú, un, una, unas, unos, usted, ustedes, varias, varios, vosotras, vosotros, vuestra, vuestras, vuestro, vuestros, y, yo.

Stemming:

Texto de ejemplo: 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

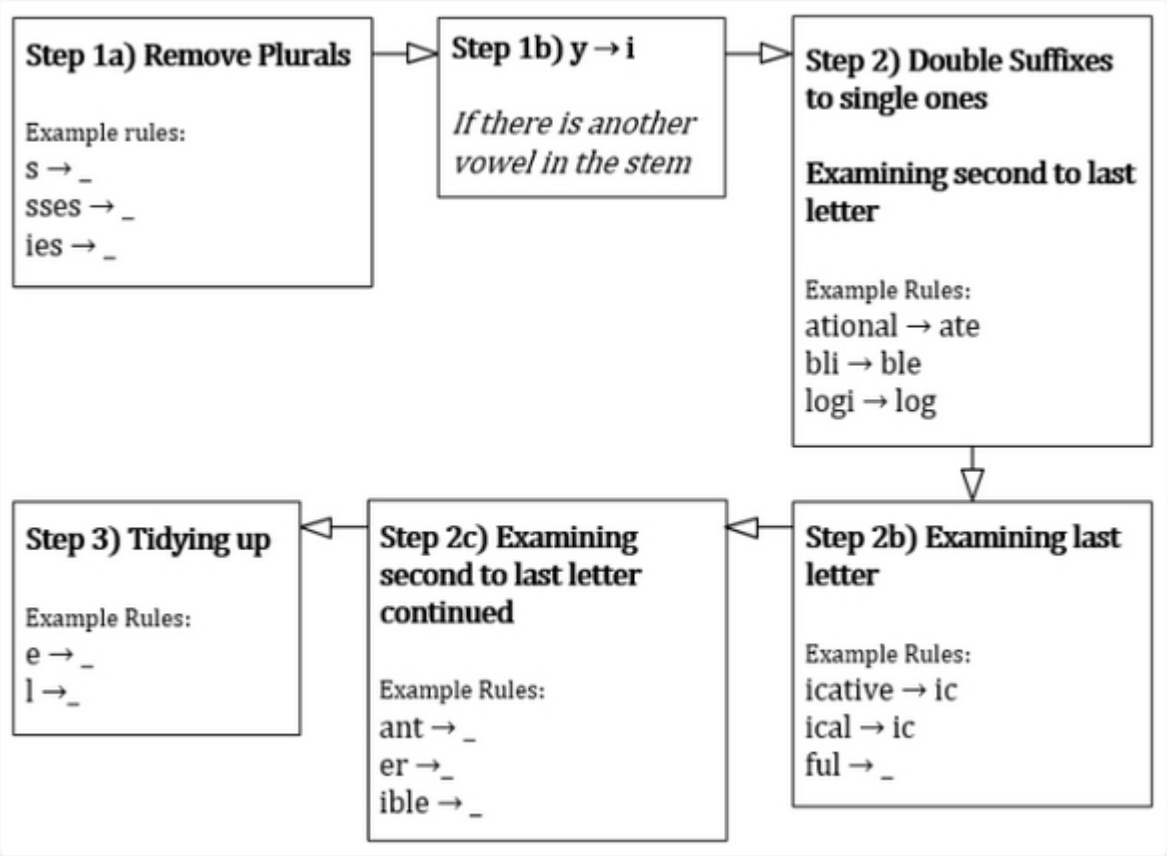
Porter: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Lovins: 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

Paice: 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

Algoritmo de Porter

Algoritmo basado en reglas de reducción de sufijos a nivel de palabras:



Stemming en NLTK

```python
# import these modules
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

# choose some words to be stemmed
words = ["program", "programs", "programmer", "programming", "programmers"]

for w in words:
    print(w, " : ", ps.stem(w))
```

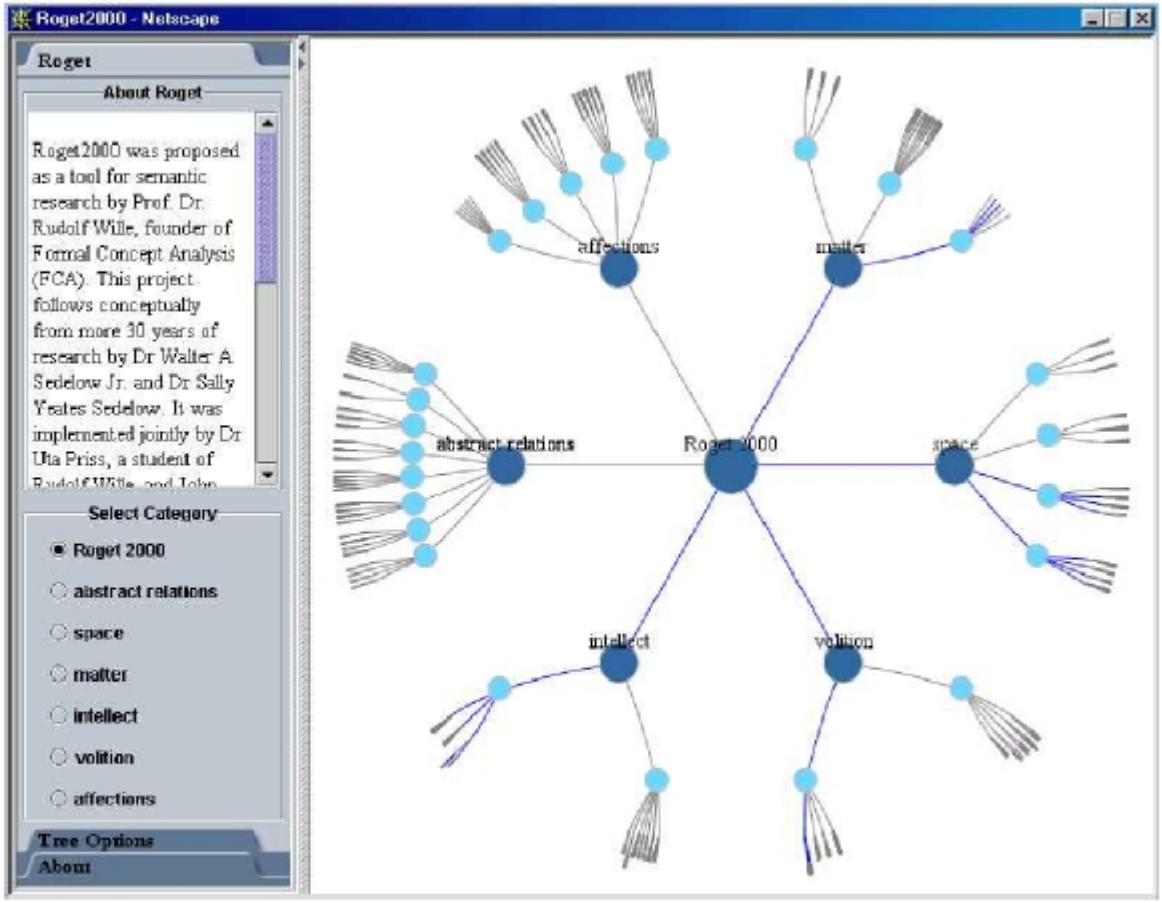
```
program  :  program
programs  :  program
programmer  :  program
programming  :  program
programmers  :  program
```

Stemming en NLTK

```python
# importing modules
from nltk.stem import PorterStemmer
from nltk.tokenize import word_tokenize

ps = PorterStemmer()

sentence = "Programmers program with programming languages"
words = word_tokenize(sentence)

for w in words:
    print(w, " : ", ps.stem(w))
```

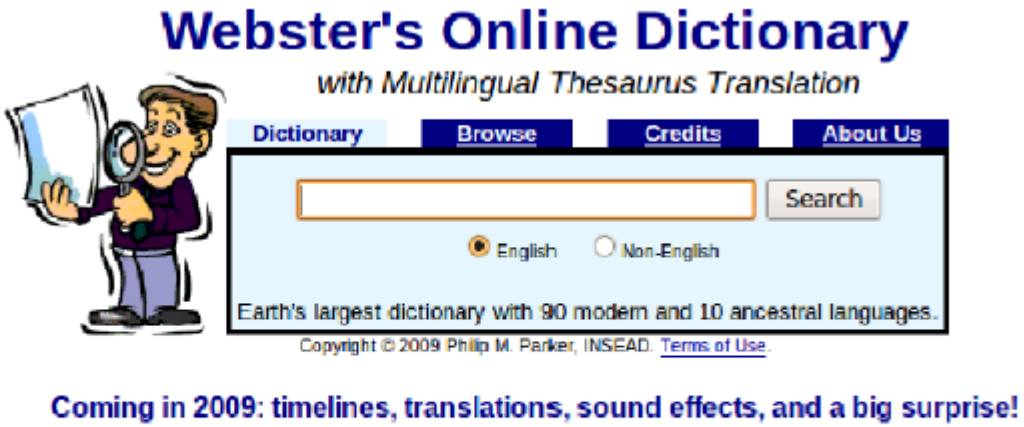
```
Programmers  :  program
program  :  program
with  :  with
programming  :  program
languages  :  languag
```

Corpus



Ver más en http://www.roget.org

Corpus



Ver más en http://www.websters-online.dictionary.org

Corpus

WordNet Search - 3.0 - WordNet home page - Glossary - Help

Word to search for: car [Search WordNet]

Display Options: (Select option to change) [Change]

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

Noun

- S: (n) **car**, auto, automobile, machine, motorcar (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*
- S: (n) **car**, railcar, railway car, railroad car (a wheeled vehicle adapted to the rails of railroad) *"three cars had jumped the rails"*
- S: (n) **car**, gondola (the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant)
- S: (n) **car**, elevator car (where passengers ride up and down) *"the car was on the top floor"*
- S: (n) cable car, **car** (a conveyance for passengers or freight on a cable railway) *"they took a cable car to the top of the mountain"*

Ver más en http://www.wordnet.princeton.edu

Corpus

```
WN(1WN)                    WordNet™ User Commands                    WN(1WN)

NAME
       wn - command line interface to WordNet lexical database

SYNOPSIS
       wn  [  searchstr  ]  [ -h] [ -g ] [ -a ] [ -l ] [ -o ] [ -s ] [ -n# ] [
       search_option... ]

DESCRIPTION
       wn() provides a command line interface to the WordNet database,  allow-
       ing  synsets and relations to be displayed as formatted text.  For each
       word, different searches are provided, based on syntactic category  and
       pointer types.  Although only base forms of words are usually stored in
       WordNet, users may search for inflected forms.  A morphological process
       is  applied  to the search string to generate a form that is present in
       WordNet.

       The command line interface is often  useful  when  writing  scripts  to
       extract  information from the WordNet database.  Post-processing of the
       output with  various  scripting  tools  can  reformat  the  results  as
       desired.

OPTIONS
       -h              Print help text before search results.

       -g              Display textual glosses associated with synsets.

       -a              Display lexicographer file information.

       -o              Display synset offset of each synset.

       -s              Display each word's sense numbers in synsets.

       -l              Display  the  WordNet  copyright notice, version number,
                       and license.
```

Corpus

wn car -hypon

Sense 1:

car, auto, automobile, machine, motorcar

- ambulance
- beach wagon, station wagon, wagon, beach waggon
- bus, jalopy, heap
- cab, hack, taxi, taxicab
- compact, compact car
- convertible
- coupe
- cruiser, police cruiser, patrol car, police car
- electric, electric automobile, electric car
- gas guzzler
- hardtop

...

Corpus

wn car -hypen

Sense 1

car, auto, automobile, machine, motorcar

- motor vehicle, automotive vehicle
- self-propelled vehicle
- wheeled vehicle
- vehicle
- conveyance, transport
- instrumentality, instrumentation
- artifact, artefact
- object, physical object
- entity
- whole, whole thing, unit
- object, physical object
- entity

...

Corpus

wn car -meron

Sense 1

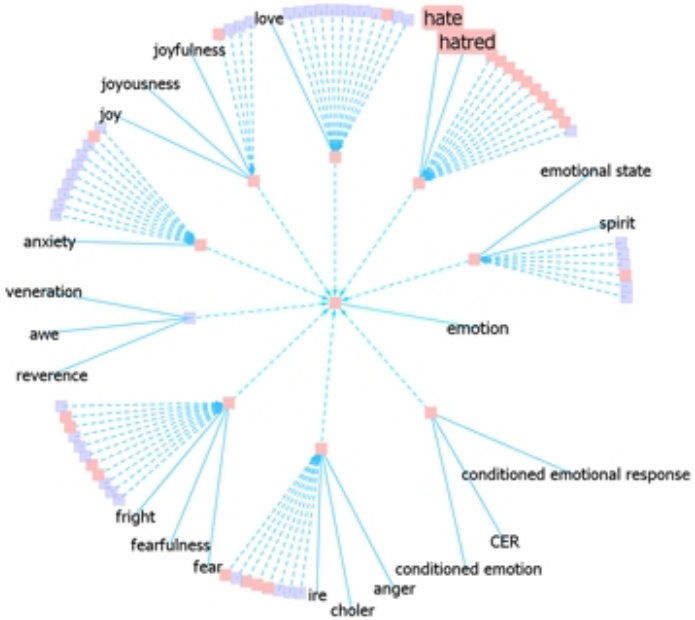
car, auto, automobile, machine, motorcar

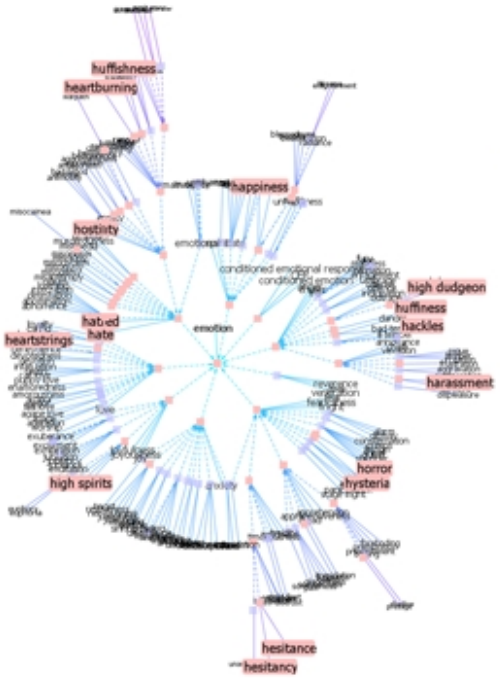
- HAS PART: accelerator, accelerator pedal, gas pedal
- HAS PART: air bag
- HAS PART: auto accessory
- HAS PART: automobile engine
- HAS PART: automobile horn, car horn, motor horn, horn, hooter
- HAS PART: buffer, fender
- HAS PART: bumper
- HAS PART: car door
- HAS PART: car mirror
- HAS PART: car seat
- HAS PART: car window

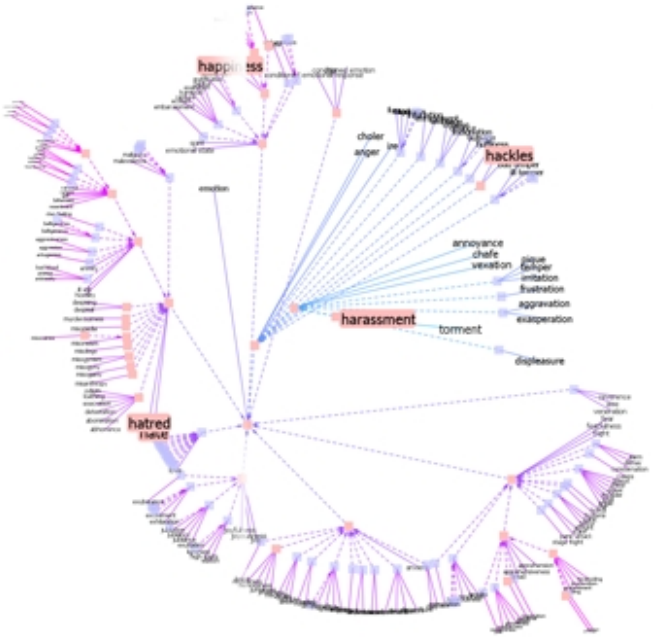
...

Corpus

WordNet es una enorme red de palabras







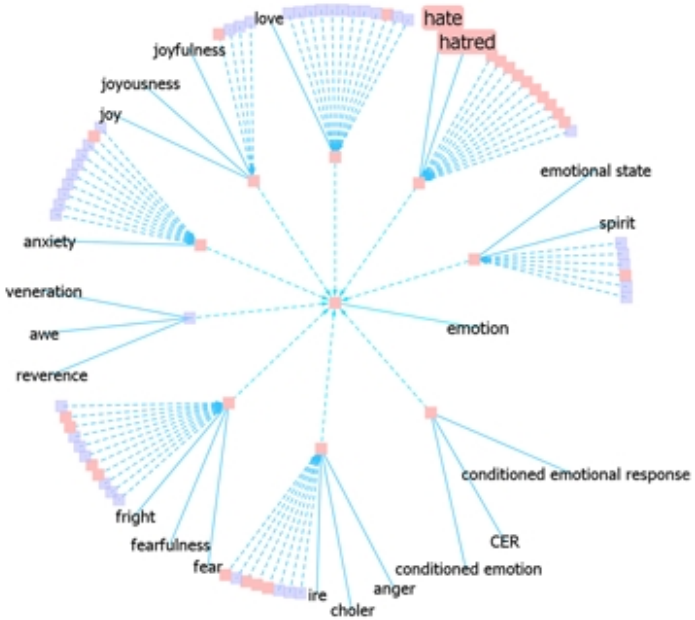
Zoom out

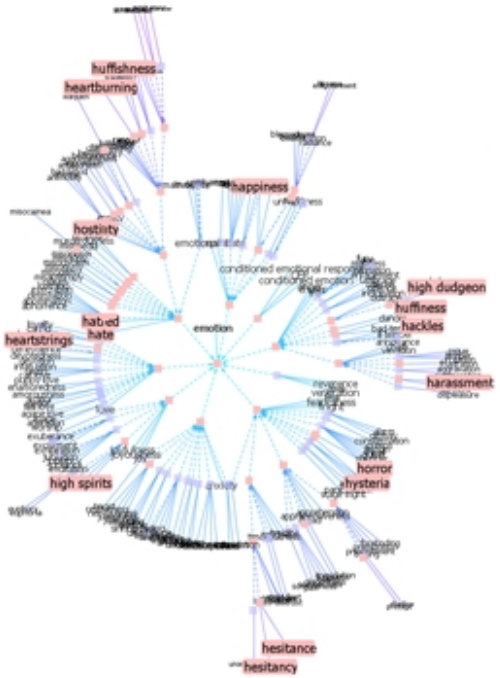
Zoom in

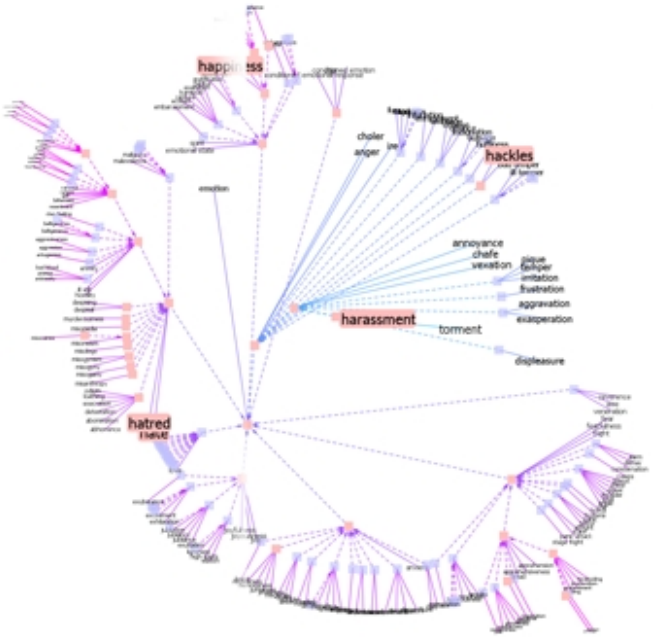
Corpus

WordNet es una enorme red de palabras

- 155287 palabras organizadas en 117659 synsets







Zoom out

Zoom in

Conceptos de NLP

- Token – String delimitado que aparece en el texto.
- Término – token con significado según un corpus (por ejemplo diccionario)

Conceptos de NLP

- Token – String delimitado que aparece en el texto.
- Término – token con significado según un corpus (por ejemplo diccionario)
- Input:
  amigos, Romans, habitantes. | habia una vez ... Cesar | ...
- Output:
  amigo | romano | habitante | cesar | ...
- Cada token es candidato a término.
- Cuáles elegimos? Depende del corpus.

Lematización

- Reducir formas infleccionales a su raíz
- Ejemplo: *am, are, is* $\rightarrow$ *be*
- Ejemplo: *autos, auto, automoviles* $\rightarrow$ *auto*
- Ejemplo: *Los autos de los jóvenes son de colores* $\rightarrow$ *auto joven es color*
- Lematización implica realizar una reducción hacia la raíz (lema). (*destruccion* $\rightarrow$ *destruir*)

Lematización

- Reducir formas inflecionales a su raíz ⟶ Raíz semántica
- Ejemplo: *am, are, is* → *be*
- Ejemplo: *autos, auto, automoviles* → *auto*
- Ejemplo: *Los autos de los jóvenes son de colores* → *auto joven es color*
- Lematización implica realizar una reducción hacia la raíz (lema). (*destruccion* → *destruir*)

WordNet lemmatizer

Lematización

- Reducir formas inflecionales a su raíz → Raíz semántica
- Ejemplo: *am, are, is* → *be*
- Ejemplo: *autos, auto, automoviles* → *auto*
- Ejemplo: *Los autos de los jóvenes son de colores* → *auto joven es color*
- Lematización implica realizar una reducción hacia la raíz (lema). (*destruccion* → *destruir*)

Wordnet Lemmatizer

Los lematizadores consideran el análisis morfológico de las palabras. Para ello usan recursos léxicos como diccionarios. Una red semántica como Wordnet reduce naturalmente las palabras a su lema, debido a que contiene la información morfológica de cada forma:

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study
niñas	Feminine gender, plural number of the noun niño	niño
niñez	Singular number of the noun niñez	niñez

Wordnet lemmatizer en NLTK

```
import nltk
nltk.download('wordnet')
from nltk.stem import WordNetLemmatizer

# Create WordNetLemmatizer object
wnl = WordNetLemmatizer()

# single word lemmatization examples
list1 = ['kites', 'babies', 'dogs', 'flying', 'smiling',
         'driving', 'died', 'tried', 'feet']
for words in list1:
    print(words + " ---> " + wnl.lemmatize(words))

#> kites ---> kite
#> babies ---> baby
#> dogs ---> dog
#> flying ---> flying
#> smiling ---> smiling
#> driving ---> driving
#> died ---> died
#> tried ---> tried
#> feet ---> foot
```

Wordnet lemmatizer en NLTK

```
# sentence lemmatization examples
string = 'the cat is sitting with the bats on the striped mat under many flying geese'

# Converting String into tokens
list2 = nltk.word_tokenize(string)
print(list2)
#> ['the', 'cat', 'is', 'sitting', 'with', 'the', 'bats', 'on',
#   'the', 'striped', 'mat', 'under', 'many', 'flying', 'geese']

lemmatized_string = ' '.join([wnl.lemmatize(words) for words in list2])

print(lemmatized_string)
#> the cat is sitting with the bat on the striped mat under many flying goose
```

NLP en Español

Recurso: spaCy (https://spacy.io/)

TRAINED PIPELINES

- Catalan
- Chinese
- Danish
- Dutch
- English
- French
- German
- Greek
- Italian
- Japanese
- Lithuanian
- Macedonian
- Multi-language
- Norwegian Bokmål
- Polish
- Portuguese
- Romanian
- Russian
- Spanish

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

text = """Soy un texto. Normalmente soy más largo y más grande. Que
no te engañe mi tamaño."""

doc = nlp(text)

lexical_tokens = [t.orth_ for t in doc if not t.is_punct | t.is_stop]
```

NLP en Español

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

def normalize(text):
    doc = nlp(text)
    words = [t.orth_ for t in doc if not t.is_punct | t.is_stop]
    lexical_tokens = [t.lower() for t in words if len(t) > 3 and
    t.isalpha()]

    return lexical_tokens

word_list = normalize("Soy un texto de prueba. ¿Cuántos tokens me
quedarán después de la normalización?")
```

NLP en Español

```
import nltk
from nltk import SnowballStemmer

spanishstemmer=SnowballStemmer('spanish')

text = """Soy un texto que pide a gritos que lo procesen. Por eso yo
canto, tú cantas, ella canta, nosotros cantamos, cantáis, cantan…"""

tokens = normalize(text) # crear una lista de tokens

stems = [spanishstemmer.stem(token) for token in tokens]
```

NLP en Español

Porter 2

```
import nltk
from nltk import SnowballStemmer

spanishstemmer=SnowballStemmer('spanish')

text = """Soy un texto que pide a gritos que lo procesen. Por eso yo
canto, tú cantas, ella canta, nosotros cantamos, cantáis, cantan…"""

tokens = normalize(text) # crear una lista de tokens

stems = [spanishstemmer.stem(token) for token in tokens]
```

['text', 'pid', 'grit', 'proces', 'cant', 'cant', 'cant', 'cant', 'cant', 'cant']

NLP en Español

OK por hoy

Porter 2

```
import nltk
from nltk import SnowballStemmer

spanishstemmer=SnowballStemmer('spanish')

text = """Soy un texto que pide a gritos que lo procesen. Por eso yo
canto, tú cantas, ella canta, nosotros cantamos, cantáis, cantan…"""

tokens = normalize(text) # crear una lista de tokens

stems = [spanishstemmer.stem(token) for token in tokens]
```

['text', 'pid', 'grit', 'proces', 'cant', 'cant', 'cant', 'cant', 'cant', 'cant']

… también tiene lematización y otros módulos del pipeline NLP.