

23 al 27 de Septiembre de 2019

- Representación de bolsa de palabras (BoW)
- Representación distribucional de términos (BoC)
 - Document occurrence representation (DOR)
 - Term co-occurrence representation (TCOR)
 - Concise semantic analysis (CSA)

Definición

Un **documento** es una unidad de datos textual que usualmente, aunque no necesariamente, se corresponde con algún documento del mundo real (reporte de negocios, memorandum, artículo científico, e-mail, etc).

Cada tipo de documento, contiene información textual genérica y también puede haber específica del dominio considerado:

- 1 e-mails
- 2 artículos científicos
- 3 artículos en Wikipedia
- 4 conversaciones en chats
- 5 “tweets”

Las colecciones de documentos

- 1 Pueden variar significativamente en tamaño (de cientos a decenas de millones de documentos)
- 2 Pueden ser **estáticas** o **dinámicas**.
- 3 Pueden ser para uso general (ej. Reuters-21578) o específico (ej. Ling-spam)
- 4 Colecciones extremadamente grandes o (muy cambiantes) plantean importantes desafíos para sistemas de TM.
- 5 Ej. Medline (Pubmed). Repositorio on-line de información sobre artículos científicos bio-médicos.
 - 12 millones de abstracts científicos (1966- a la fecha)
 - 40000 nuevos abstracts por mes.
 - Accesible en: <http://www.ncbi.nlm.nih.gov/pubmed>

Algunas colecciones “clásicas”

La colección 20-Newsgroups

- 19997 mensajes con noticias de 20 grupos de discusión de Usenet enviados durante 1993.
- 20 grupos de noticias diferentes. 1000 docs. × grupo, menos 1 (997)
- Algunos grupos muy relacionados (*comp.sys.ibm.pc.hardware* y *comp.sys.mac.hardware*).
- Otros no tienen ninguna relación (*soc.religion.christian* and *misc.forsale*)

Algunas colecciones “clásicas” II

La colección Reuters-21578

- 21578 cables de noticias de la agencia Reuters.
- Distribuida en 22 archivos en formato SGML. 1000 docs. × archivo, menos 1 (578)
- Cada documento puede pertenecer a varias categorías.
- Se han generado sub-colecciones “single label” (SL)

Subcolección R52

- Subcolección SL de R90.
- R90: documentos de las 90 clases con al menos 1 ejemplo + de training y de testing.

Subcolección R8

- Subcolección SL de R10.
- R10: documentos de las 10 clases con el nro. más alto de ejemplos +.

Algunas colecciones “clásicas” III

La colección WebKB

- Conjunto de páginas Web de los departamentos de CS de distintas universidades: Cornell, Texas, Washington, Wisconsin, etc.
- 7 categorías: *student*, *faculty*, *course*, *project*, *department*, *staff*, and *other*.
- Usualmente sólo se usan las 4 primeras categorías (4199 páginas)

Interesante: En <http://web.ist.utl.pt/~acardoso/datasets/>, Ana Cardoso-Cachopo hace disponible estas colecciones con distintos grado de procesamiento para problemas de categorización SL.

Tipos de características (features)

Distintos autores consideran distintas **dimensiones** de clasificación:

- Layton: estáticas vs dinámicas (variables)
- Lex: léxicas vs estilométricas
- Koppel: contenido vs estilométricas
- Anderka: contenido vs estructura vs red vs historia de edición (específicas de Wikipedia)

Características estáticas

- el modelo Bag of Words (BOW)
- n -gramas de palabras
- n -gramas de caracteres

<i>FEATURE</i>	<i>Description/Example</i>
SIMLEYS	A list of emoticons compiled from the Wikipedia.
OMG	Abbreviation for ‘Oh My God’
ELLIPSES	‘....’
POSSESSIVE BIGRAMS	E.g. my_XXX, our_XXX
REPATED ALPHABETS	E.g. niceeeeeee, noooo waaaay
SELF	E.g., Lxxx, Im_xxx
LAUGH	E.g. LOL, ROTFL, LMFAO, haha, hehe
SHOUT	Text in ALLCAPS
EXASPERATION	E.g. Ugh, mmmm, hmmm, ahh, grrr
AGREEMENT	E.g. yea, yeah, ohya
HONORIFICS	E.g. dude, man, bro, sir
AFFECTION	E.g. xoxo
EXCITEMENT	A string of exclamation symbols (!!!!!)
SINGLE EXCLAIM	A single exclamation at the end of the tweet
PUZZLED PUNCT	A combination of any number of ? and ! (!?!!??!)

Ejemplo de características: Usuarios de Twitter (III)

Dinámicas:

Derivadas del **contenido** (textos) de los **tweets**.

Ejemplos:

- n -gramas de palabras.
- n -gramas de caracteres.
- ...

Explicamos luego.

Ejemplo de características: Artículos de Wikipedia

- **Objetos/instancias**: información de cada **artículo**.
- **Atributos**: datos de su **historia de edición**, propiedades del artículo, *n*-gramas (de palabras y caracteres), etc.
- **Valores de los atributos**: en general, de tipo **numéricos**.

Algunas características de su historia de edición y propiedades.

[illegible]

Ejemplo de características: Artículos de Wikipedia

- **NE**: número total de ediciones.
- **EA**: edad del artículo (en días).
- **VA**: número de vueltas al estado anterior.
- **LA**: longitud del artículo (en nro. de caracteres).
- **NLI**: número de links internos.
- **NI**: número de imágenes
- **IF**: índice de Flesch (facilidad de lectura) .

Algunas características de su historia de edición y propiedades.

[illegible]

Ejemplo de características: Artículos de Wikipedia

Dinámicas:

Derivadas de las palabras contenidas en los **artículos**.

Ejemplos:

- n -gramas de palabras.
- n -gramas de caracteres.
- ...

Explicamos luego.

Ejemplo: Textos arbitrarios - características estáticas

Documentos

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

Número de palabras (NP), longitud de palabra más larga (LPL), longitud promedio de palabras (LPP), verbos en pasado (VP)

ID	NP	LPL	LPP	VP
d1	6	8	4	1
d2	8	9	4,5	0
d3	9	7	4	1

Características estáticas: ejemplo en Wikipedia

Feature	Description
<i>Content-based</i>	
Character count	Number of characters in the text (no spaces).
Word count	Number of words in the plain text.
Sentence count	Number of sentences in the plain text.
Word length	Average word length in characters.
Sentence length	Average sentence length in words.
Paragraph count	Number of paragraphs.
Paragraph length	Average paragraph length in sentences.
Longest word length	Length in characters of the longest word.
Longest sentence length	Number of words in the longest sentence.
Shortest sentence length	Number of words in the shortest sentence.
Long sentence rate	Percentage of long sentences. A long sentence is defined as containing at least 30 words.
Short sentence rate	Percentage of short sentences. A short sentence is defined as containing at most 15 words.
Longest subsection length	Length in words of the longest subsection.
Shortest subsection length	Length in words of the shortest subsection.
Subsections length	Total number of words in the article's subsections.
Average subsection length	Average number of words per subsection.
Longest subsection	Length in words of the longest subsection.
Shortest subsection	Length in words of the shortest subsection.
Subsubsections length	Total number of words in the article's subsubsections.
Average subsubsections	Average number of words per subsubsection.
<i>Structure-based</i>	
Section count	Number of sections.
Subsection count	Number of subsections.
Subsubsection count	Number of subsubsections.
Heading count	Number of sections, subsections and subsubsections.
Section nesting	Average number of subsections per section.
Subsection nesting	Average number of subsubsections per subsection.
Reference Sections Count	Number of reference sections, e.g. "References", "Footnotes", "Sources", "Bibliography".
Mandatory Sections Count	Number of mandatory sections, e.g. "See also".
Related page count	Number of related pages, e.g. "Further reading", "See also", etc.
Lead length	Number of words in the lead section (text before the first heading).
Lead rate	Percentage of words in the lead section.
Image count	Number of images.
Image rate	Ratio of image count to section count.
Link count	Every occurrence of a link (introduced with two open square brackets) in the unfiltered text.
Link rate	Percentage of links.
Table count	Number of tables.
Reference count	Number of all references using the <code><ref>...</ref></code> syntax.
Reference section rate	Ratio of reference count to the accumulated section, subsection and subsubsection count.
Reference word rate	Ratio of reference count to word count.
Unique reference count	Number of unique references using the <code><ref>...</ref></code> syntax.
Reference ratio	Ratio between the reference word rate of the article and the maximum reference word rate found in the dataset.
Templates count	Number of (different) Wikipedia templates.

Características basadas en caracter

Considera al documento como una **serie de caracteres**. Incluye una cuenta de cada caracter individual, y proporciones de clases de caracteres.

Ejemplo:

- Número total de caracteres
- Proporción de caracteres **alfabéticos** en el documento
- Proporción de letras **mayúsculas**.
- Proporción de caracteres de **dígitos**.
- Frecuencia de los distintos caracteres.
- otros ...

Características basadas en palabras

Considera al documento como una **serie de palabras** en sentencias. Incluye estadísticas del tamaño de las palabras, riqueza del vocabulario, etc.

Ejemplo:

- Número total de palabras
- Proporción de palabras cortas (menos de 4 caracteres)
- Longitud promedio de palabras
- Proporción $|set(words)|/|words|$
- número de “hapax legomena” y “dislegomena”
- Medidas de riqueza de vocabulario: K de Yule, D de Simpson, S de Sichel, W de Brunet, R de Honore.
- proporción de palabras de cada longitud (de 1 a 19)
- proporción de palabras de longitud ≥ 20

Características sintácticas

Capturan aspectos del estilo de escritura del autor a nivel de la **sentencia**. Esta información sintáctica es capturada de manera directa o indirecta con características de muy variada complejidad. Ejemplos:

- Frecuencia de determinados **signos de puntuación** (, . ? ! : ; " ')
- Frecuencia de determinadas **palabras de paro** (which, that, among)
- Cuenta del uso del **pasivo**
- Cuenta de distintas categorías sintácticas (POS tags), como sustantivos, pronombres, adjetivos, adverbios, etc.

Características estructurales

Se derivan de cómo se estructura el texto. Refleja los hábitos del autor a la hora de organizar una pieza de escritura.

Ejemplo:

- Longitud promedio del párrafo (en sentencias, palabras o caracteres)
- Uso de indentado
- Número total de sentencias, líneas y párrafos
- otros ...

Características específicas del contenido

- Capturan la aparición de elementos del lenguaje típicos del problema, como por ejemplo la frecuencia de palabras claves específicas del problema asociadas a un dominio particular.
- Estas características presentan una selección de características dirigida por el conocimiento del experto.

Tipos de características dinámicas

Se derivan automáticamente del procesamiento de los documentos, por lo que no se puede definir a priori cuales serán exactamente estas características, ya que variarán de acuerdo a la colección de documentos considerados. Ejemplo: representar los documentos con la frecuencia de ocurrencia de las 50 palabras más frecuentes.

Características dinámicas más comunes:

- 1 **palabras** que ocurren en la colección de documentos (necesidad de filtros básicos)
- 2 **términos**
- 3 ***n*-gramas** de **caracteres** ($n = 1, 2, 3, 4 \dots$)
- 4 ***n*-gramas** de **palabras** ($n = 1, 2, 3, 4 \dots$)

Ejemplo de características dinámicas: Textos arbitrarios (Representación BOW)

- **Objetos/instancias**: información de cada **texto**.
- **Atributos**: las distintas **palabras** que aparecen en los documentos.
- **Valores de los atributos**: de tipo **numéricos** o **booleanos**.

texto1.txt

"pintaron el banco de la plaza"

texto2.txt

"te paso el programa, ejecútalo paso por paso"

texto3.txt

"sentado en el banco, miraba si el banco
abría"

Ejemplo de características dinámicas: Textos arbitrarios (Representación BOW)

Documentos

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

Pesos binarios

ID	abría	banco	de	ejecúto	el	en	la	miraba	paso	pintaron	plaza	por	programa	sentado	si	te
d1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
d2	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1
d3	1	1	0	0	1	1	0	1	0	0	0	0	0	1	1	0

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

Pesos *TF* (Frecuencia del término)

ID	abría	banco	de	ejecutó	el	en	la	miraba	paso	pintaron	plaza	por	programa	sentado	si	te
d1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
d2	0	0	0	1	1	0	0	0	3	0	0	1	1	0	0	1
d3	1	2	0	0	2	1	0	1	0	0	0	0	0	1	1	0

Ejemplo de características dinámicas: Textos arbitrarios - n -gramas (palabras)

Documentos

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

bi-gramas de palabras - Pesos binarios

ID	banco abría	banco de	banco miraba	de la	ejecútalo paso	el banco	el programa	en el	la plaza	...
d1	0	1	0	1	0	1	0	0	1	...
d2	0	0	0	0	1	0	1	0	0	...
d3	1	0	1	0	0	2	0	1	0	...

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

	ID	_ab	_ba	_de	_ej	_el	...	aso	aza	ba_	ban	brí	...	tad	tal	tar	te_	úta
	d1	0	1	1	0	1	...	0	1	0	1	0	...	0	0	1	0	0
	d2	0	0	0	1	1	...	1	0	0	0	0	...	0	1	0	1	1
	d3	1	1	0	0	1	...	0	0	1	1	1	...	1	0	0	0	0

Características aprendidas

- 1 **Idea:** extender el **aprendizaje automático**, usualmente usado para generar el **modelo de clasificación**, a la **representación de los documentos**.
- 2 Componente principal en el **aprendizaje de representaciones** para el PLN: **word embedding (WE)** (incrustación de palabra)
- 3 Los **WEs** son **representaciones distribuidas** de **palabras** basadas en **vectores densos**, de **longitud fija** construidas mediante estadísticas de co-ocurrencia de palabras según la **hipótesis distribucional**

Características aprendidas (II)

- 1 Los WEs pueden ser derivados mediante enfoques **basados en conteo**:
 - LSA
 - HAL
 - COALS
 - ... y Glove (entre otros)
- 2 Sin embargo, se han popularizado últimamente los enfoques **basados en predicción** utilizando métodos de **aprendizaje neuronal**: **word2vec**, **fasttext**, **Elmo** y **Bert**
- 3 Los WEs, además de capturar relaciones **sintácticas** y **semánticas** muy interesantes de las palabras, soportan el funcionamiento de modelos neuronales más generales para documentos como las arquitecturas **LSTM** y **CNN**.

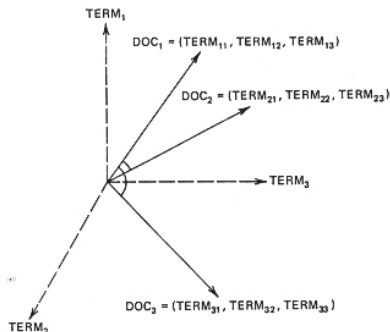
El Modelo de Espacio Vectorial

El **Modelo de Espacio Vectorial** (**VSM** por las siglas en inglés de **Vector Space Model**) es una de las variantes surgidas de la **recuperación de información** (IR) para la recuperación de documentos:

- 1 El Modelo de Espacio Vectorial
- 2 El Modelo Probabilístico
- 3 El modelo Lógico

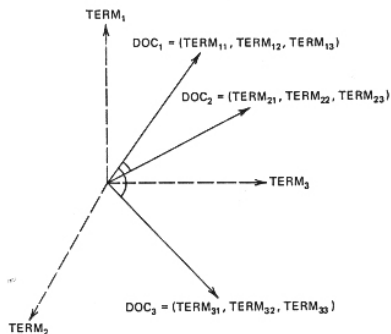
El Modelo de Espacio Vectorial

- Desarrollado por **Gerard Salton** y colegas para el sistema **SMART** de IR, fue pionero en muchos de los conceptos usados en los motores de búsqueda modernos.
- VSM representa **cada documento** en una colección como **un punto** en un **espacio** (un **vector** en un **espacio vectorial**)



El Modelo de Espacio Vectorial

- En VSM, la idea fue interpretar **distancia espacial** como **distancia semántica**
- Esto es, puntos **cercanos** en este espacio son **semánticamente similares** y los que están **muy separados** están **semánticamente distantes**

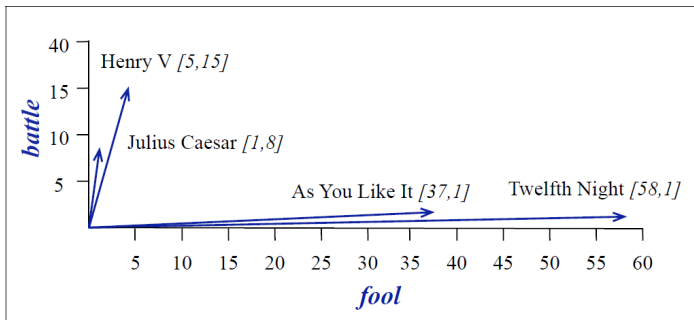


El Modelo de Espacio Vectorial

- El VSM, propuesto originalmente para representar **documentos** (como **bolsas de palabras**) fue posteriormente aplicado en **palabras** y **sentencias**.
- Existen variantes de VSM que difieren en el **tipo de matriz** utilizado y las **hipótesis** semánticas estadísticas subyacente:
 - 1 matriz **documento-término** (hipótesis de la **bolsa de palabras**)
 - 2 matriz **palabra-contexto** (hipótesis **distribucional**)
 - 3 matriz **par-patrón** (hipótesis de la **relación latente**)

El Modelo de Espacio Vectorial - ejemplo (Jurasky et. al)

	battle	soldier	fool	clown
As You Like it	1	2	37	5
Twelfth Night	1	2	58	117
Julius Caesar	8	12	1	0
Henry V	15	36	5	0



Representación de bolsa de palabras (BoW)

Representación de *bolsa de palabras* ("Bag of Words" (BoW))

Documentos

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

Representación BoW

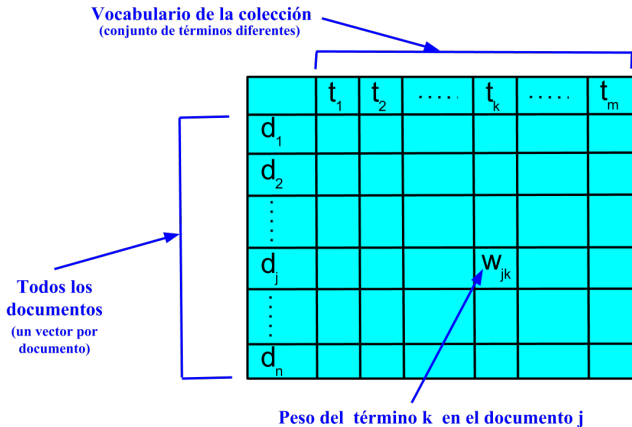
D/T	abría	banco	de	ejecútalo	el	en	la	miraba	paso	pintaron	plaza	por	programa	sentado	si	te
d1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
d2	0	0	0	1	1	0	0	0	3	0	0	1	1	0	0	1
d3	1	2	0	0	2	1	0	1	0	0	0	0	0	1	1	0

Representación BoW

- En BoW cada **documento** es representado como una **bolsa de palabras**
- Cada bolsa (multiset) es implementada como un **vector** con las **frecuencias** de ocurrencia en el documento de **cada palabra/término** del **vocabulario** de la colección de documentos.
- Así, las bolsas de todos los documentos de la colección constituyen una **matriz “documento-término”** X
- Con n **documentos** en la colección y m **términos** ocurriendo **en alguno** de esos documentos, tendremos una matriz X de dimensión $n \times m$

Representación BoW

- En IR, la **hipótesis de la bolsa de palabras** es que se puede estimar la relevancia de un documento a una consulta representando tanto los documentos como las consultas como **bolsa de palabras**.
- Esta hipótesis expresa la creencia que un vector fila en la matriz documento-término X captura (en alguna medida) un aspecto del **significado** del documento correspondiente: aquello de **lo que trata** el documento.
- Así, planteándolo en términos de la matriz documento-término X , si dos documentos tienen vectores **filas similares** en X , éstos tenderán a tener **significados similares**.



Consideraciones sobre el enfoque BoW

- Cada documento d_j es representado como un vector de **pesos de términos** $\vec{d}_j = \langle w_{j1}, \dots, w_{jm} \rangle$ donde $m = |\mathcal{T}|$ es la cardinalidad del **vocabulario** \mathcal{T} y $0 \leq w_{jk} \leq 1$ es la contribución del término t_k a la semántica del documento d_j
- Elecciones a realizar:
 - 1 ¿Qué tipo de **términos** considerar?
 - 2 ¿Cómo **ponderar** esos términos?

Términos versus palabras

- En muchos casos las **palabras** no son usadas directamente para la representación de documentos sino los **“términos”** que se obtienen a partir de ellas.
- Llamaremos **término** a toda entidad que constituye la **unidad atómica de significado** en un texto
- Dependiendo de la elección del diseñador, un término podrá ser:
 - 1 las palabras
 - 2 las **raíces morfológicas** (stems) de las palabras
 - 3 los **lemas** de las palabras
 - 4 **n-gramas** (de **caracteres** o de **palabras**)
 - 5 **frases**
 - 6 otros

Ponderación de términos: ideas principales

- La **importancia** (**peso**) de un término se **incrementa** proporcionalmente al número de veces que aparece en el documento (supuesto de la **frecuencia del término**)
 - Ayuda a **describir** el contenido del documento.
- La **importancia** general de un término se **decrementa** proporcionalmente a sus ocurrencias en la colección entera (supuesto de la **frecuencia de documento inversa**)
 - Términos comunes no son buenos para **discriminar** entre clases diferentes.
- Pesados tipo **tf – idf** favorecen **documentos largos** y deberían ser **normalizados**

Ponderación de términos: enfoques principales

Pesos binarios

$w_{ji} = 1 \Leftrightarrow$ el documento d_j contiene el término t_i , 0 en otro caso.

Frecuencia del término (tf)

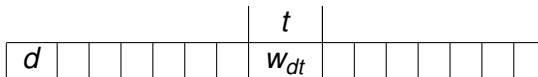
$w_{ji} = tf(t_i, d_j)$ (número de ocurrencias del término t_i en el documento d_j)

Esquema de ponderación $tf \times idf$

$$w_{ji} = tf(t_i, d_j) \times idf(t_i)$$

- n : número de documentos en la colección
- $idf(t_i) = \log[n/df(t_i)]$, donde $df(t_i)$ es el número de documentos que contienen al término t_i

Notación de ponderación genérica: codificaciones SMART



$$w_{dt} = TF'_{t,d} \cdot IDF'_t \cdot NORM$$

Frecuencia del término

$$n = TF_{t,d}$$

$$b = 1$$

$$m = \frac{TF_{t,d}}{\max_t(TF_{t,d})}$$

$$a =$$

$$0,5 + 0,5 \frac{TF_{t,d}}{\max_t(TF_{t,d})}$$

$$l = 1 + \log(TF_{t,d})$$

Frecuencia de Documento Inversa

$$n = 1$$

$$t = \log\left(\frac{n}{DF_t}\right)$$

NORM

$$n = 1$$

$$C = \frac{1}{\sqrt{\sum_t (TF'_{t,d} IDF'_t)^2}}$$

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

Pesos binarios (SMART bnn)

ID	abría	banco	de	ejecúto	el	en	la	miraba	paso	pintaron	plaza	por	programa	sentado	si	te
d1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
d2	0	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1
d3	1	1	0	0	1	1	0	1	0	0	0	0	0	1	1	0

Representación BOW - ponderación *TF*

Documentos

- 1 "pintaron el banco de la plaza"
- 2 "te paso el programa, ejecútalo paso por paso"
- 3 "sentado en el banco, miraba si el banco abría"

Pesos *TF* (Frecuencia del término - **SMART** nnn)

ID	abría	banco	de	ejecútalo	el	en	la	miraba	paso	pintaron	plaza	por	programa	sentado	si	te
d1	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
d2	0	0	0	1	1	0	0	0	3	0	0	1	1	0	0	1
d3	1	2	0	0	2	1	0	1	0	0	0	0	0	1	1	0

Representación BOW - ponderación *tf* – *idf* (normalizada)

Documentos

- ❶ "pintaron el banco de la plaza"
- ❷ "te paso el programa, ejecútalo paso por paso"
- ❸ "sentado en el banco, miraba si el banco abría"

Pesos *tf* – *idf* normalizada (**SMART ntc**)

ID	abría	banco	de	ejecútalo	el	en	la	miraba	paso	pintaron	plaza	por	programa	sentado	si	te
d1	0.	0.34	0.45	0.	0.26	0.	0.45	0.	0.	0.45	0.45	0.	0.	0.	0.	0.
d2	0.	0.	0.	0.27	0.16	0.	0.	0.	0.82	0.	0.	0.27	0.27	0.	0.	0.27
d3	0.33	0.51	0.	0.	0.40	0.33	0.	0.33	0.	0.	0.	0.	0.	0.33	0.33	0.

Representación BOW - implementación en Python

En la **notebook asociada**: <https://colab.research.google.com/drive/1UW0tYD8TjFvXGQdR9BzEgZlHwLkx6e7n>

```
//github.com/merrecalde/curso_la_plata_2019/  
blob/master/representacion_documentos.ipynb
```

se muestran, para este ejemplo, diversas vectorizaciones de los documentos utilizando distintos tipos de términos y distintas formas de pesados de los términos.

Consideraciones finales sobre BoW

Ideas subyacentes

- No se captura ningún tipo de información sobre el **orden** en que aparecen los términos/palabras
- BoW sólo mira a la **forma superficial** de las palabras, ignorando toda **información semántica** de las mismas

Ventajas

- Simplicidad.
- Eficiencia.

Desventajas

- BoW tiende a producir **representaciones muy dispersas** (“sparse”)
- Problemas “semánticos” con la **polisemia** y la **sinonimia**

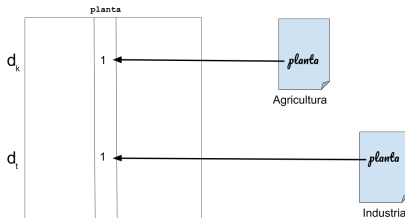
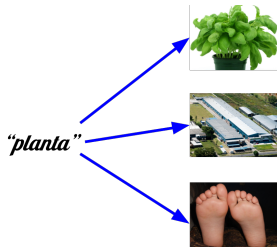
Polisemia (y homonimia)

“banco”



BoW y la polisemia

La **polisemia** introduce **ruido** en la **representación BoW**



Sinonimia

“automotor”

“auto”

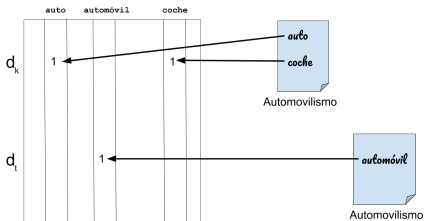
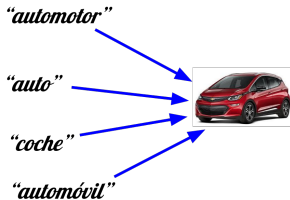
“coche”

“automóvil”



BoW y la sinonimia

La **sinonimia divide** la evidencia en la **representación BoW**

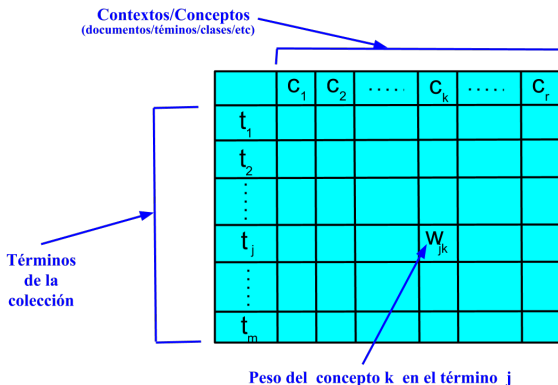


Representación distribucional de términos

- Las ideas surgen de las **dificultades** que se presentan para definir el **significado** de las palabras.
 - Circularidad** de las definiciones de diccionario.
 - Dificultad para capturar otros tipos de **relaciones** entre palabras (más allá de las relaciones semánticas clásicas de WN) (**asociación** de palabras, **campos semánticos**, **significados afectivos/connotaciones**).
- Identificadas por filósofos como **Ludwig Wittgenstein**:
*... the **meaning** of a word is its **use** in the language*
- ... y lingüistas como **John R. Firth**:
*You shall **know** a word by the **company** it keeps!*

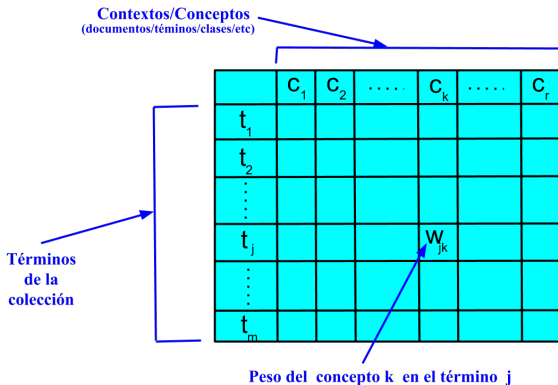
Representación distribucional de términos

- Propuesto para abordar las **deficiencias** del modelo BoW
- El foco se pone en las **palabras** y los **contextos** en que éstas ocurren



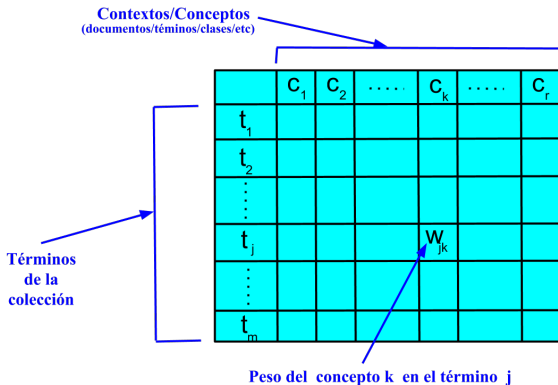
Representación distribucional de términos

- Se cambia el foco en medir **similitud de palabras**
- **Hipótesis distribucional**: palabras que ocurren en **contextos similares** tienden a tener **significados similares**.



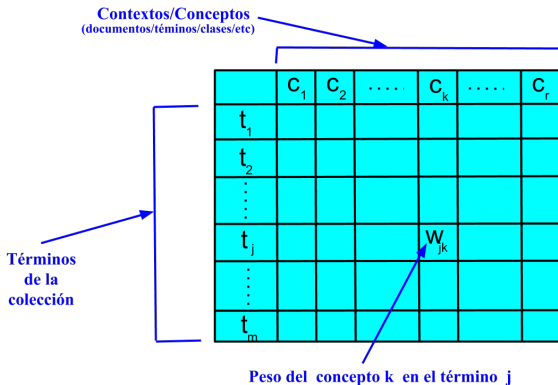
Representación distribucional de términos

- Cada palabra es representada por un **vector**
- Cada elemento del vector se deriva de la **ocurrencia de la palabra** en distintos **contextos**: otras **palabras**, **frases**, **sentencias**, **párrafos**, **capítulos**, **documentos**, etc.



Representación distribucional de términos

- Cada **contexto** puede ser visualizado como un **concepto**
- Cada **fila** de la matriz palabra-concepto puede ser visualizada entonces como una **bolsa de conceptos (BOC)**



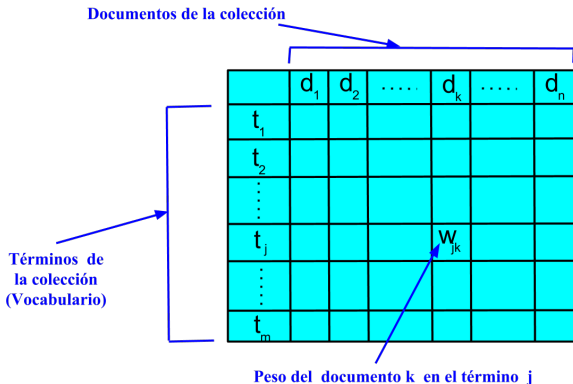
Algunas que veremos a continuación:

- Document occurrence representation (DOR)
- Term co-occurrence representation (TCOR)
- Concise semantic analysis (CSA)

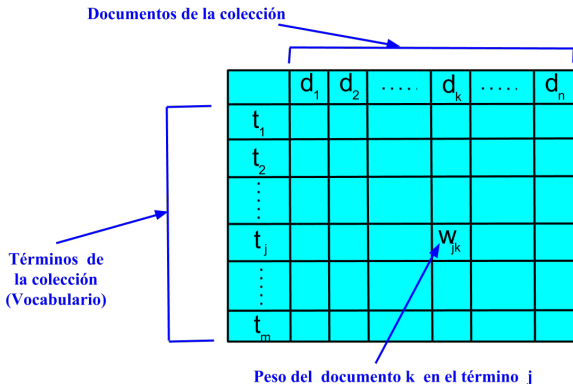
Usos de las representaciones distribucionales

- Tareas de **administración de términos**
 - **Categorización** de términos
 - **Clustering** de términos
 - **Generación** automática de tesauros
 - **Desambiguación** del sentido de las palabras (**WSD**)
- Tareas “clásicas” con **documentos**
 - El **significado de un documento** es considerado como la **unión** de los **significados de los términos**
 - Requiere una forma de **agregación** de los **vectores de los términos** para formar el **vector documento**

- Enfoque surgido de la IR, donde la **semántica de un término** es visualizada en función de la **bolsa de documentos** en la que el término ocurre
- Cada documento es una **feature** independiente



- Los términos son representados como **vectores** en el **espacio de documentos**.
- Dos **términos** están **relacionados** si muestran **distribuciones similares** a lo largo de los documentos



Intuiciones acerca de los pesos

Para entender cómo se derivan estos pesos, recordemos antes cómo se obtenían en BoW para la codificación SMART **Itc**:

- Cada documento d_j es representado como un vector de **pesos de términos** $\vec{d}_j = \langle w_{j1}, \dots, w_{jm} \rangle$ ($m = |\mathcal{T}|$ es la cardinalidad del **vocabulario** \mathcal{T}), $0 \leq w_{jk} \leq 1$.

$$tfidf(t_k, d_j) = tf'(t_k, d_j) \cdot \log(\frac{n}{DF_{t_k}})$$

$$tf'(t_k, d_j) = \begin{cases} 1 + \log(tf(t_k, d_j)) & \text{if } tf(t_k, d_j) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$w_{jk} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^m tfidf(t_s, d_j)^2}}$$

	d_1	d_2	d_k	d_n
t_1						
t_2						
\vdots						
t_i				w_{jk}		
\vdots						
t_m						

$$df_itf(d_k, t_j) = df'(d_k, t_j) \cdot \log(\frac{m}{\#\mathcal{T}_{d_k}})$$

$$df'(d_k, t_j) = \begin{cases} 1 + \log(df(d_k, t_j)) & \text{if } df(d_k, t_j) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$w_{jk} = \frac{df_itf(d_k, t_j)}{\sqrt{\sum_{s=1}^n df_itf(d_s, t_j)^2}}$$

- $df(d_k, t_j)$ denota el número de veces que t_j ocurre en d_k .
- $\#T_{d_k}$ denota el número de **términos distintos** en el vocabulario T que ocurren al menos una vez en d_k

	d_1	d_2	d_k	d_n
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

$$df_itf(d_k, t_j) = df'(d_k, t_j) \cdot \log(\frac{m}{\#\mathcal{T}_{d_k}})$$

$$df'(d_k, t_j) = \begin{cases} 1 + \log(df(d_k, t_j)) & \text{if } df(d_k, t_j) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$w_{jk} = \frac{df_itf(d_k, t_j)}{\sqrt{\sum_{s=1}^n df_itf(d_s, t_j)^2}}$$

DOR es la **versión dual** de la representación **BoW**

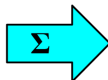
- Cuanto más frecuentemente t_j ocurre en d_k , más importante es d_k para caracterizar las semánticas de t_j
- Cuanto más términos distintos tiene d_k , menor es su contribución para caracterizar la semántica de t_i

- DOR es una representación de palabras, no de documentos.
- La representación de documentos es la suma ponderada de los vectores de sus términos

$$\vec{d}_j = \sum_{t_i \in d_j} (\alpha_{ij} \times \vec{t}_i) \quad (1)$$

Representación de documentos

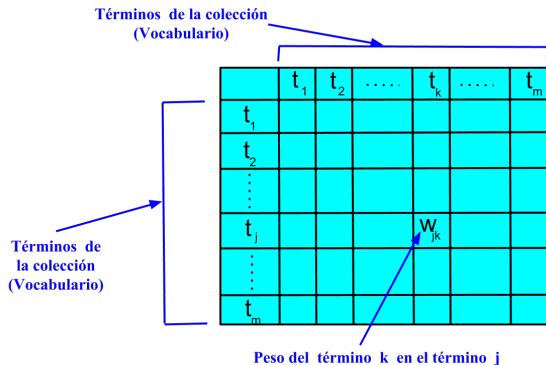
	d_1	d_2	d_k	d_n
t_1						
t_2						
\vdots						
t_j				w_k		
\vdots						
t_m						



	d_1	d_2	d_q	d_n
d_1						
d_2						
\vdots						
d_z				w_{zq}		
\vdots						
d_n						

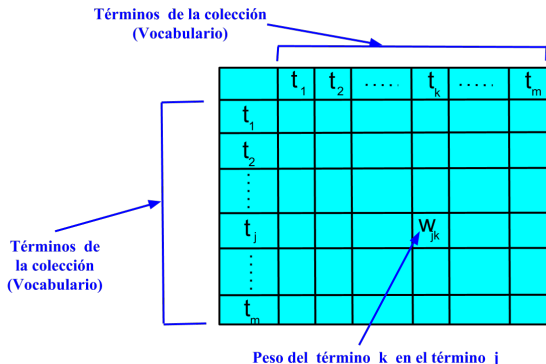
occurrence representation (TCOR)

- Enfoque surgido de la CL (Lingüística computacional), donde la **semántica de un término** es visualizada en función de la **bolsa de términos** que **co-ocurren** con el mismo en los documentos de la colección.
- Así, cada término es una **feature** independiente



Term co-occurrence representation (TCOR)

- Al igual que DOR es una representación de términos **distribucional** y **vectorial**.
- Dos **términos** están **relacionados** si muestran **distribuciones similares** de co-ocurrencia con el resto de los términos.



Intuiciones acerca de los pesos

Representación TCOR

	t_1	t_2	t_k	t_m
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

Cálculo de los pesos

$$tf_itf(t_k, t_j) = tf'(t_k, t_j) \cdot \log(\frac{m}{\#\mathcal{T}_{t_k}})$$

$$tf'(t_k, t_j) = \begin{cases} 1 + \log(tf(t_k, t_j)) & \text{if } tf(t_k, t_j) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$w_{jk} = \frac{tf_itf(t_k, t_j)}{\sqrt{\sum_{s=1}^m tf_itf(t_s, t_j)^2}}$$

- $tf(t_k, t_j)$ denota el número de documentos en que t_k y t_j co-ocurren.
- $\#T_{t_k}$ denota el número de **términos distintos** en el vocabulario T que co-ocurren con t_k en al menos un documento.

	t_1	t_2	t_k	t_m
t_1						
t_2						
\vdots						
t_i				w_{jk}		
\vdots						
t_m						

$$tf_idf(t_k, t_j) = tf'(t_k, t_j) \cdot \log(\frac{m}{\#\mathcal{T}_{t_k}})$$

$$tf'(t_k, t_j) = \begin{cases} 1 + \log(tf(t_k, t_j)) & \text{if } tf(t_k, t_j) > 0 \\ 0 & \text{en otro caso} \end{cases}$$

$$w_{jk} = \frac{tf_itf(t_k, t_j)}{\sqrt{\sum_{s=1}^m tf_itf(t_s, t_j)^2}}$$

TCOR es la típica representación usada en **WSD**

- Cuanto más documentos t_k y t_j co-ocurren, más importante es t_k para caracterizar las semánticas de t_j
- Cuanto más **términos distintos** t_k co-ocurre, menor es su contribución para caracterizar la semántica de t_j

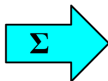
- Al igual que DOR, **TCOR** es una representación de **palabras/términos**.
- Para representar los documentos, se realiza la **suma ponderada** de los **vectores** de sus **términos**

$$\vec{d}_j = \sum_{t_j \in d_j} (\alpha_{ij} \times \vec{t}_i) \quad (2)$$

Representación de documentos

Matriz documento-término

	t_1	t_2	t_k	t_m
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						



	t_1	t_2	t_q	t_m
d_1						
d_2						
\vdots						
d_z				w_{zc}		
\vdots						
d_n						

Concise semantic analysis (CSA)

- Las representaciones de documentos analizadas hasta el momento son **no supervisadas**
 - Están pensadas para **tareas generales** de análisis de texto (**supervisadas** y **no supervisadas**)
 - No toman en cuenta información de la **clase/categoría** a la que pertenece el documento

Veremos ahora un **enfoque distribucional** que toma información de esas **clases**

- Este enfoque, denominado **concise semantic analysis (CSA)** puede considerarse una representación “**bag of concepts**” (BoC) supervisada.

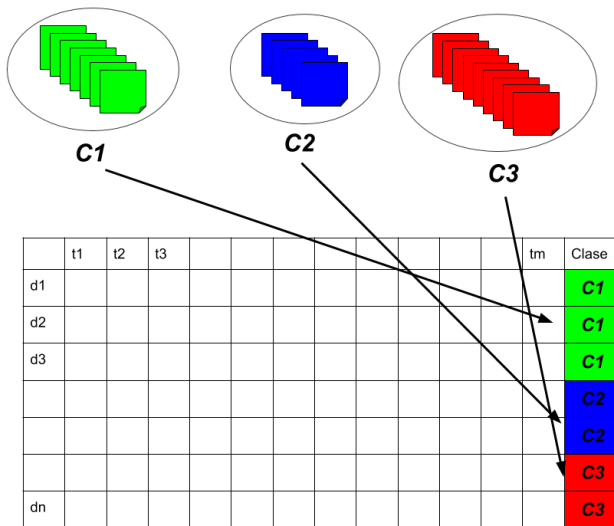
Bolsa de Palabras (BoW)



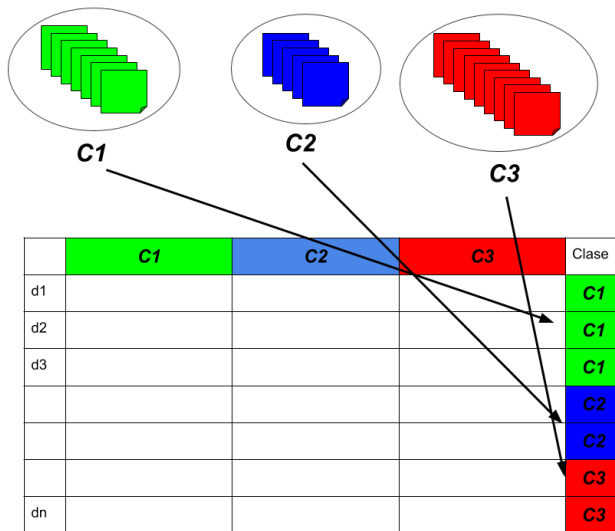
Bolsa de Palabras (BoW)

[illegible]

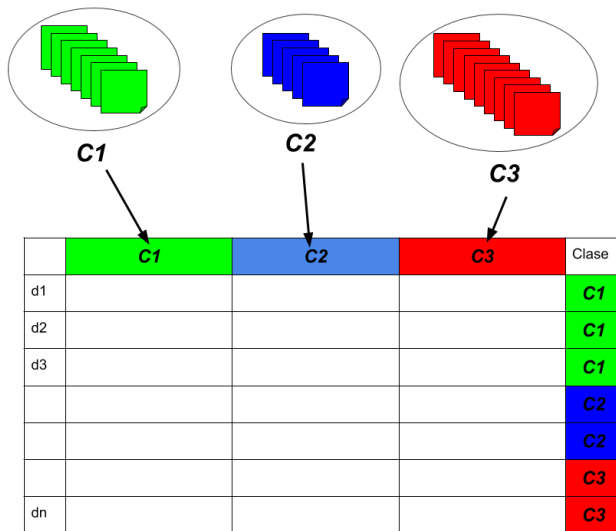
Bolsa de Palabras (BoW)



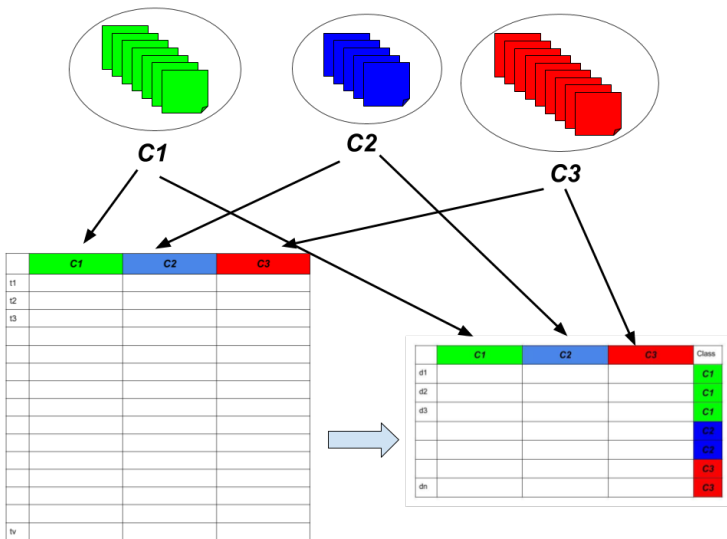
Concise Semantic Analysis (CSA)



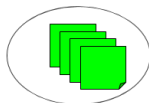
Concise Semantic Analysis (CSA)



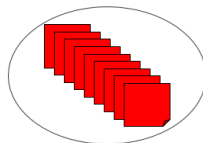
Concise Semantic Analysis (CSA)



Ejemplo de aplicación de CSA- predicción de depresión



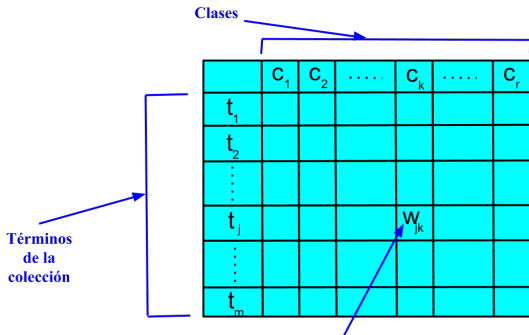
deprimido (+)



no-deprimido (-)

	<i>D</i>	<i>ND</i>	Clase
d1			<i>D</i>
d2			<i>D</i>
			<i>ND</i>
			<i>ND</i>
			<i>ND</i>
			<i>ND</i>
dn			<i>ND</i>

- La representación **distribucional** de términos en este caso toma como **conceptos** a las **clases** a las que pertenecen los documentos.
- Dos **términos** están **relacionados** si sus **distribuciones** de frecuencia relativa en (los documentos de) las distintas clases son similares.



Intuiciones acerca de los pesos

Representación CSA

	c_1	c_2	c_k	c_r
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

Cálculo de los pesos

$$cf(c_k, t_j) = \sum_{d_h \in \mathcal{D}_{c_k}} \log_2 \left(1 + \frac{tf(t_j, d_h)}{len(d_h)} \right)$$

$$w'_{jk} = \frac{cf(c_k, t_j)}{\sum_{s=1}^m cf(c_k, t_s)}$$

$$w_{jk} = \frac{w'_{jk}}{\sum_{s=1}^r cf(c_s, t_j)}$$

- $cf(c_k, t_j)$ es la **frecuencia de la clase**: cuantas veces aparece el término t_j en documentos de la clase c_k
- \mathcal{D}_{c_k} es el conjunto de documentos **etiquetados** con la clase c_k

Intuiciones acerca de los pesos

Representación CSA

	C_1	C_2	C_k	C_r
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

Cálculo de los pesos

$$cf(c_k, t_j) = \sum_{d_h \in \mathcal{D}_{c_k}} \log_2 \left(1 + \frac{tf(t_j, d_h)}{len(d_h)} \right)$$

$$w'_{jk} = \frac{cf(c_k, t_j)}{\sum_{s=1}^m cf(c_k, t_s)}$$

$$w_{jk} = \frac{w'_{jk}}{\sum_{s=1}^r cf(c_s, t_j)}$$

- $tf(t_j, d_h)$ es la **número de ocurrencias** del término t_j en el documento d_h
- $len(d_h)$ es el número de términos en el documento d_h

	C_1	C_2	C_k	C_r
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

$$cf(c_k, t_j) = \sum_{d_h \in \mathcal{D}_{c_k}} \log_2 \left(1 + \frac{tf(t_j, d_h)}{len(d_h)} \right)$$

$$w'_{jk} = \frac{cf(c_k, t_j)}{\sum_{s=1}^m cf(c_k, t_s)}$$

$$w_{jk} = \frac{w'_{jk}}{\sum_{s=1}^r cf(c_s, t_j)}$$

- cuanto **más frecuentemente** un término aparece en los documentos que pertenecen a un concepto (clase), mayor es su membresía a dicho concepto.

Cálculo de los pesos

	C_1	C_2	C_k	C_r
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

$$cf(c_k, t_j) = \sum_{d_h \in \mathcal{D}_{c_k}} \log_2 \left(1 + \frac{tf(t_j, d_h)}{len(d_h)} \right)$$

$$w'_{jk} = \frac{cf(c_k, t_j)}{\sum_{s=1}^m cf(c_k, t_s)}$$

$$w_{jk} = \frac{w'_{jk}}{\sum_{s=1}^r cf(c_s, t_j)}$$

- los términos en un documento **más corto** son más **cercanos al concepto** que uno **más largo**.

Intuiciones acerca de los pesos

Representación CSA

	c_1	c_2	c_k	c_r
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						

Cálculo de los pesos

$$cf(c_k, t_j) = \sum_{d_h \in \mathcal{D}_{c_k}} \log_2 \left(1 + \frac{tf(t_j, d_h)}{len(d_h)} \right)$$

$$w'_{jk} = \frac{cf(c_k, t_j)}{\sum_{s=1}^m cf(c_k, t_s)}$$

$$w_{jk} = \frac{w'_{jk}}{\sum_{s=1}^r cf(c_s, t_j)}$$

- w'_{jk} son los pesos **normalizados** en relación a los **otros términos** (por columna)
- w_{jk} son los pesos **normalizados** en relación a los **otros conceptos** (por fila)

Representación de documentos con CSA

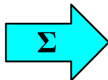
- Existen distintas alternativas para ponderar los términos al generar la representación de los documentos.
- Una que ha resultado efectiva es:

$$\vec{d}_j = \sum_{t_i \in d_j} \left(\frac{tf_{ji}}{len(d_j)} \times \vec{t}_i \right) \quad (3)$$

Representación de términos

Matriz término-clase

	c_1	c_2	c_k	c_r
t_1						
t_2						
\vdots						
t_j				w_{jk}		
\vdots						
t_m						



Representación de documentos

Matriz documento-clase

	c_1	c_2	c_q	c_r
d_1						
d_2						
\vdots						
d_z				w_{zq}		
\vdots						
d_n						