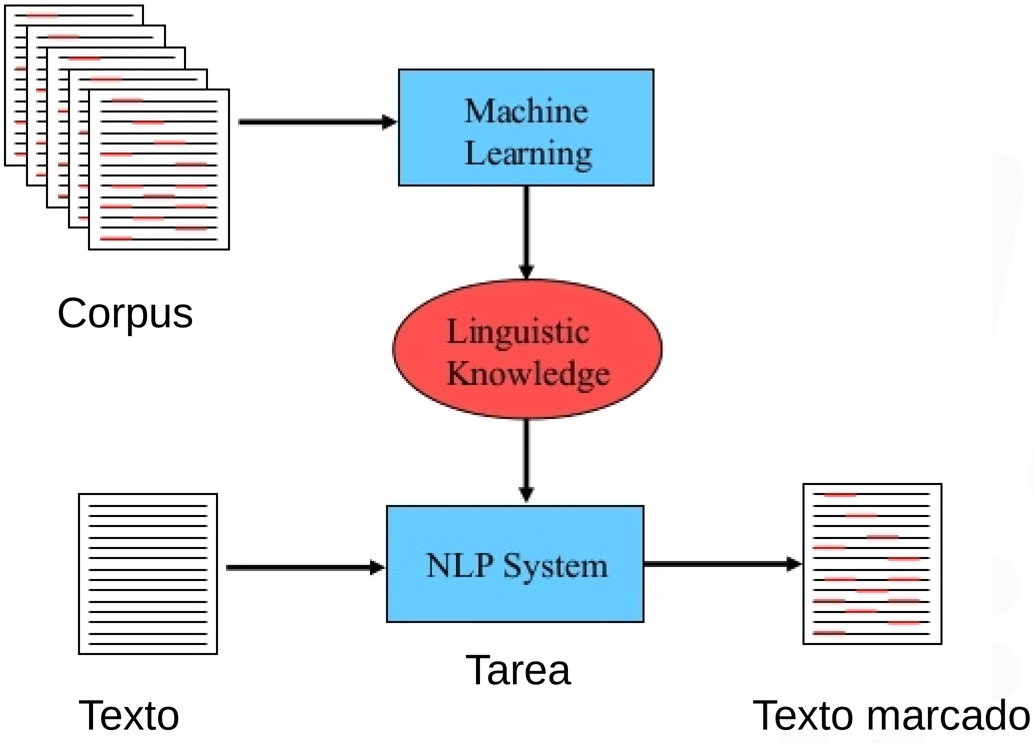
Text Mining

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

Vectorización de Texto (handcrafted)

Síntesis. El enfoque de NLP (clásico)



Matriz términos-documentos

	Antonio y Cleopatra	Julio Cesar	La Tempestad	Hamlet	Otelo	Macbeth	...
Antonio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
Cesar	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleopatra	57	0	0	0	0	0	
...							

Matriz términos-documentos

documentos →

	Antonio y Cleopatra	Julio Cesar	La Tempestad	Hamlet	Otelo	Macbeth	...
Antonio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
Cesar	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleopatra	57	0	0	0	0	0	
...							

↑

términos

Matriz términos-documentos

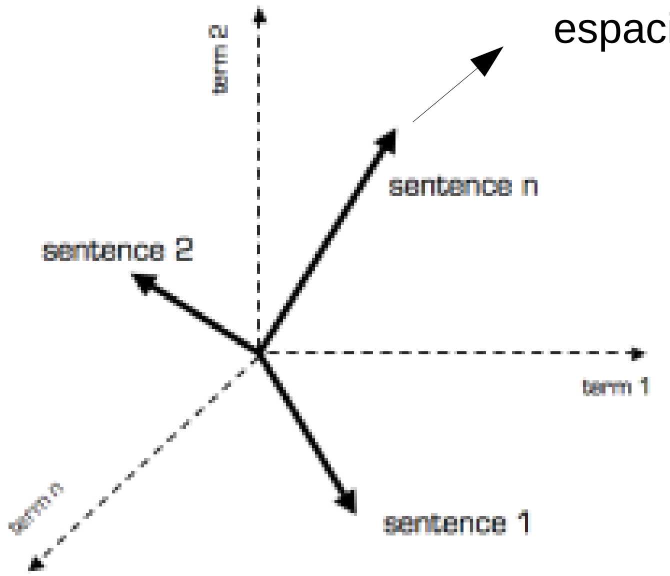
documentos

	Antonio y Cleopatra	Julio Cesar	La Tempestad	Hamlet	Otelo	Macbeth	...
Antonio	157	73	0	0	0	1	
Brutus	4	157	0	2	0	0	
Cesar	232	227	0	2	1	0	
Calpurnia	0	10	0	0	0	0	
Cleopatra	57	0	0	0	0	0	
...		...					

vectorización

términos

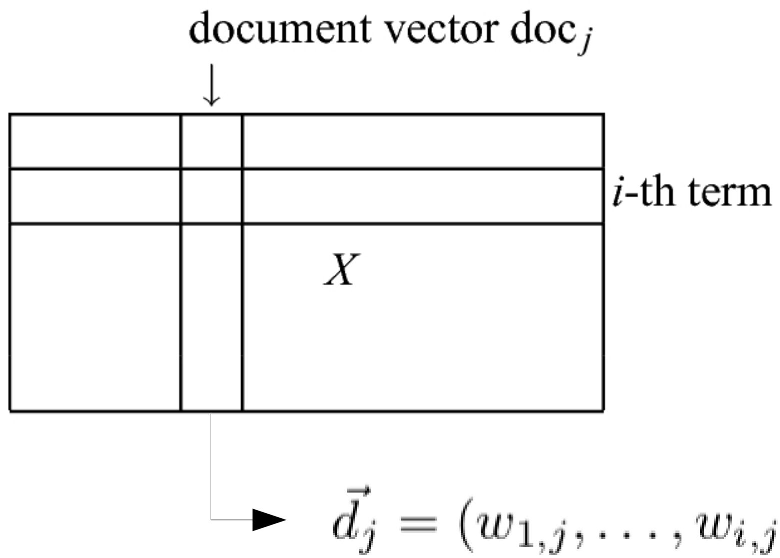
Vector-space model



Cada documento es un vector en el espacio vectorial de términos

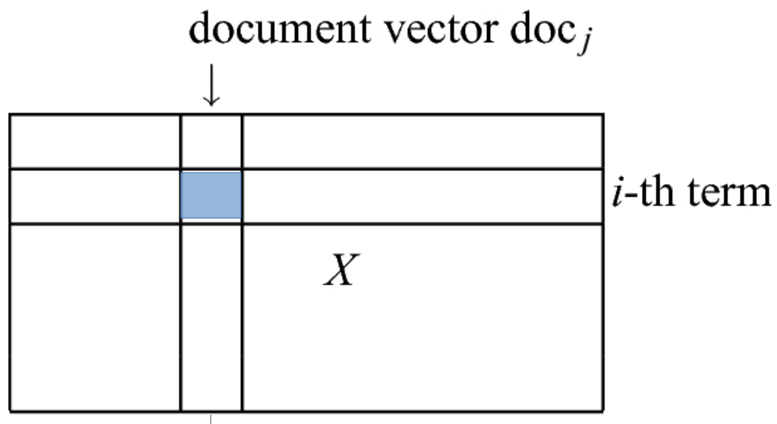
Cada término es una dimensión en el espacio vectorial de términos

Vector-space model



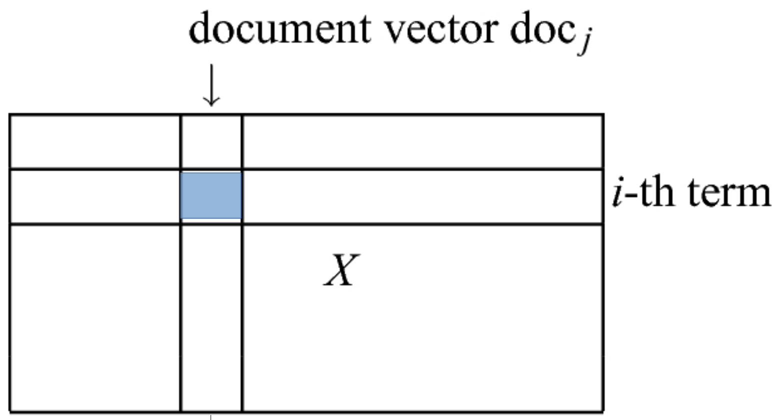
$$\vec{d}_j = (w_{1,j}, \ldots, w_{i,j}, \ldots, w_{V,j})$$

Vector-space model



$$\vec{d}_j = (w_{1,j}, \ldots, \boxed{w_{i,j}}, \ldots, w_{V,j})$$

Vector-space model



$$\vec{d}_j = (w_{1,j}, \ldots, \boxed{w_{i,j}}, \ldots, w_{V,j})$$

Term scoring function:

$$f(t_i, d_j) = w_{i,j}$$

Vector-space model

$f_{i,j}$: # occs. de ti en dj

$\max f_{l,j}$: # occs. del t más frecuente en dj

N : # docs

n_i : # docs donde ti ocurre

Term scoring functions:

- Tf: $$Tf_{i,j} = \frac{f_{i,j}}{\max f_{l,j}}$$

- Tf corregido: $$w_{i,j} = \begin{cases} 1 + \log_{10} f_{i,j} & \text{if } f_{i,j} > 0 \\ 0 & \text{e.t.o.c.} \end{cases}$$

- Idf: $$\mathrm{idf}_{t\,i} = \log_{10} \frac{N}{n_i}$$

- Tf-Idf (Salton): $$w_{i,j} = (1 + \log f_{l,j}) \cdot \log \frac{N}{n_i}$$

- Tf-Idf: $$w_{i,j} = \frac{f_{i,j}}{\max f_{l,j}} \cdot \log \frac{N}{n_i}$$

Vector-space model

$l(d_j)$: # tokens en dj

l_{avg} : largo promedio

Term scoring functions:

- Tf-Idf suavizado: $$w_{i,j} = \left(0.5 + \frac{0.5 f_{i,j}}{\max f_{l,j}}\right) \times \log \frac{N}{n_i}$$

- BM-25: $$w_{i,j} = \frac{f_{i,j} \cdot (k_1 + 1)}{k_1 \cdot \left[(1-b) + b \cdot \frac{l(d_j)}{l_{avg}}\right] + f_{i,j}} \cdot \log\left(\frac{N}{n_i}\right)$$

$$b \in [0,1] \,, \quad k_1 > 0$$

Empírico: $b \approx 0.75$

$k_1 \approx 1.2$

Vector-space model (referencias)

Tf-Idf

Salton, G. and Buckley, Ch. Term-weighting approaches in automatic text retrieval, Information Processing and Management, 24(5):513-523, 1988.

BM-25

Robertson, S. and Spärck Jones, K. Relevance weighting of search terms. Journal of the American Society for Information Science, 27(3):129-146, 1976.

Más variantes:

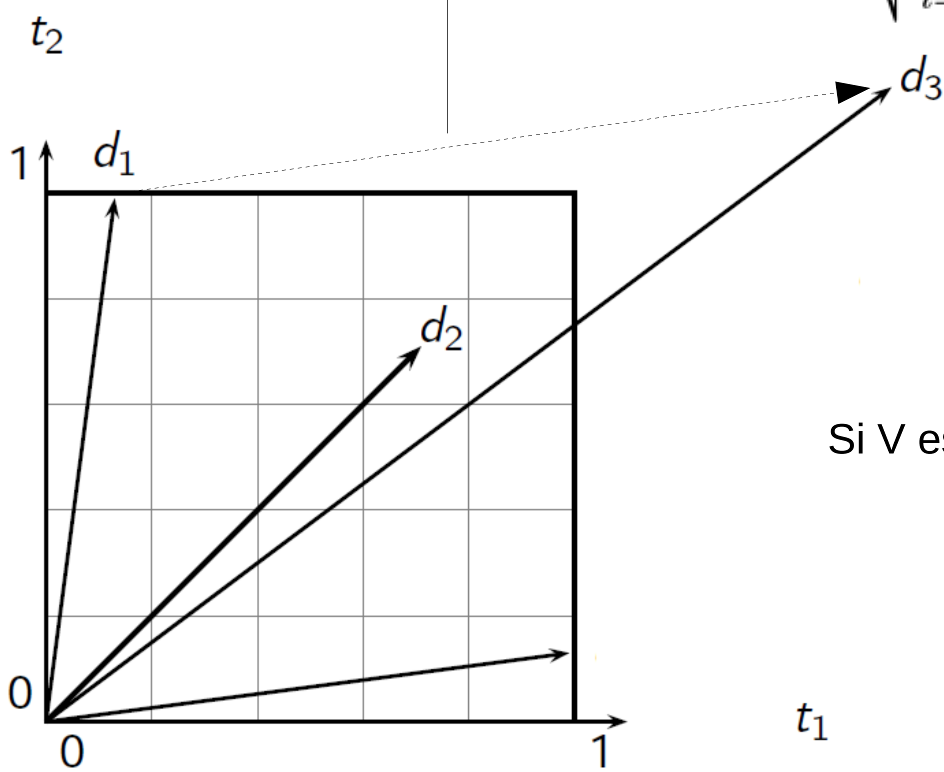
Method	Formula	Reference
Smoothing 1	$(1 + Tf_{i,c})/(n + Tf_c)$	Rennie *et al.* (2003)
Smoothing 2	$\log(1 + Tf_{i,d})$	Rennie *et al.* (2003)
Tf-Idf	$Tf_{i,d}\log(N/n_i)$	Salton and Buckley (1988)
Tf-L_Z	$Tf_{i,d}/\sqrt{\sum_{i=1}^{n} Tf_{i,d}^2}$	Salton and Buckley (1988)
Smoothing 3	$Min\{\log(1 + Tf_{i,d}),1\}$	Schneider (2005)
Tf-L_1	$Tf_{i,d}/\sum_{i=1}^{n} Tf_{i,d}$	Kolcz and Yih (2007)
Smoothing 4	$1 + \log Tf_{i,d}$	Qiang (2010)
Extended Idf	$\log((2N - n_i + 1)/n_i)$	
Extended Tf 1	$\log(1 + Tf_{i,d})/L_d$	
Extended Tf 2	$\log(1 + Tf_{i,d})/(L_d/L)$	
Extended Tf-Idf 1	$(\log(1 + Tf_{i,d})/L_d)(\log((2N - n_i + 1)/n_i)$	

Vector-space model

Distancia Euclideana

Funciones de proximidad entre vectores:

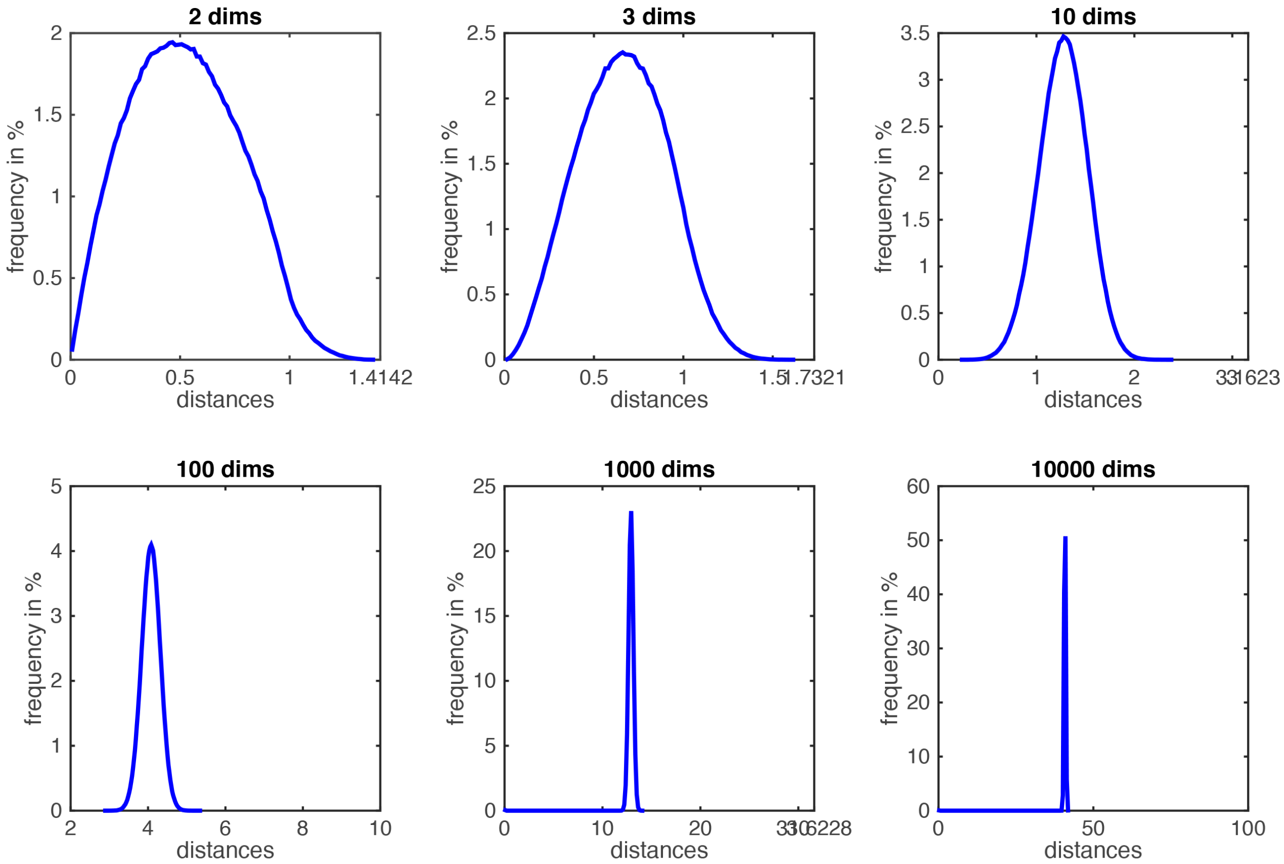
$$d(d_1, d_3) = \sqrt{\sum_{i=1}^{V} (w_{i,1} - w_{i,3})^2}$$



Si V es muy grande no funciona

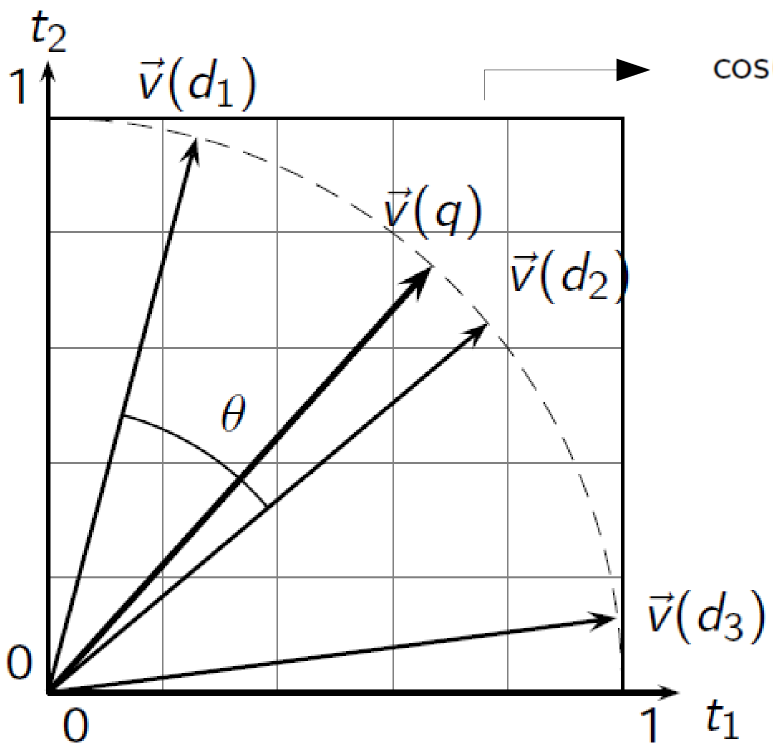
Vector-space model

Maldición de la dimensionalidad para distancia Euclideana:



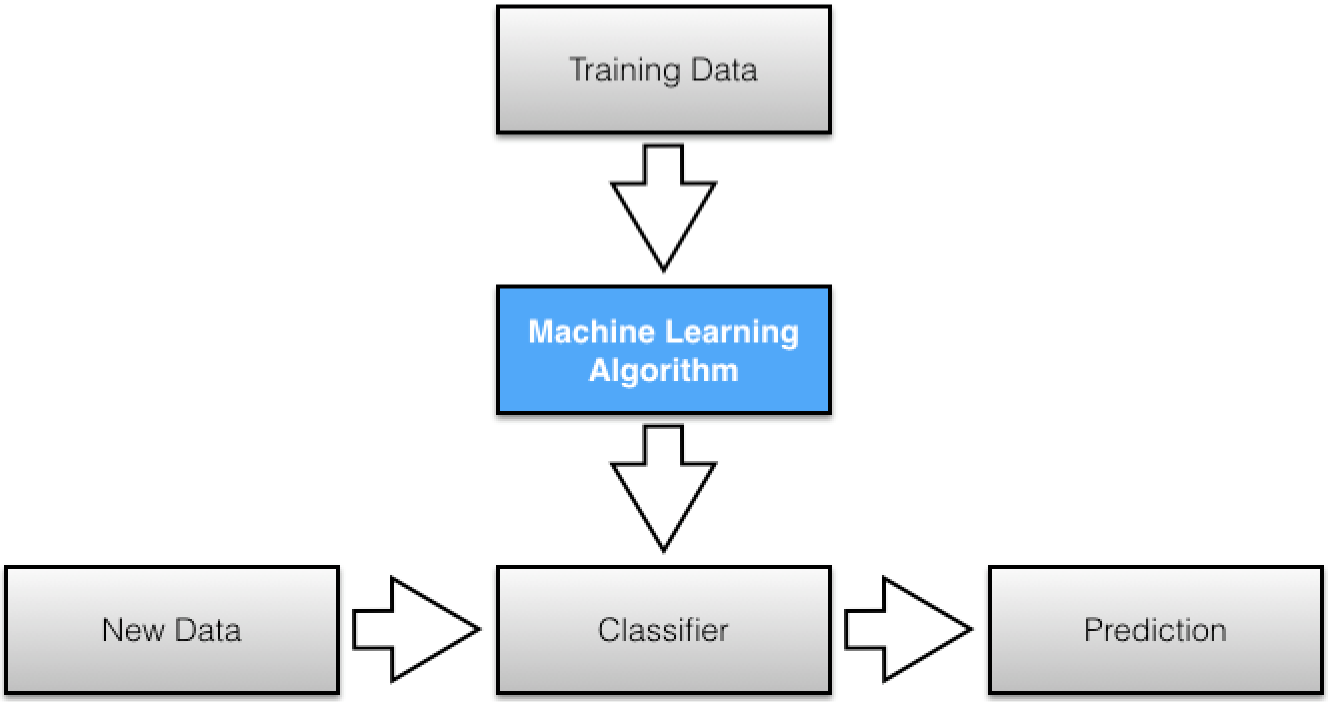
Vector-space model

Funciones de proximidad entre vectores:



$$\cos(\vec{q}, \vec{d}) = \mathrm{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

Vector-space model: clasificación de texto



Vector-space model: clasificación de texto

	20 Newsgroups			Web-KB		
	Accuracy	FP-rate	F	Accuracy	FP-rate	F
Smoothing 1	0.823	0.164	0.680	0.840	0.136	0.722
Smoothing 2	0.827	0.158	0.692	0.842	0.132	0.731
Tf-Idf	0.840	0.135	0.694	0.852	0.130	0.742
Tf-L2	0.820	0.152	0.686	0.836	0.138	0.732
Smoothing 3	0.838	0.174	0.692	0.846	0.135	0.743
Tf-L1	0.824	0.137	0.689	0.842	0.140	0.736
Smoothing 4	0.820	0.134	0.692	0.846	0.138	0.726
BM25	0.844	0.128	0.702	0.870	0.128	0.780
Extended Idf	0.832	0.117	0.685	0.849	0.130	0.755
Extended Tf 1	0.826	0.121	0.688	0.848	0.138	0.743
Extended Tf 2	0.830	0.128	0.690	0.851	0.135	0.742
Extended Tf-Idf 1	0.852	0.112	0.701	0.881	0.125	0.781

Vector-space model: clasificación de texto

System	Type	Results reported by	#1	#2	#3	#4	#5
		# of documents	21,450	14,347	13,272	12,902	12,902
		# of training documents	14,704	10,667	9,610	9,603	9,603
		# of test documents	6,746	3,680	3,662	3,299	3,299
		# of categories	135	93	92	90	10
System	**Type**	**Results reported by**					
Word	(non-learning)	Yang [1999]	.150	.310	.290		
	probabilistic	[Dumais et al. 1998]				.752	.815
	probabilistic	[Joachims 1998]				.720	
	probabilistic	[Lam et al. 1997]	.443 (MF_1)				
PropBayes	probabilistic	[Lewis 1992a]	.650				
Bim	probabilistic	[Li and Yamanishi 1999]				.747	
	probabilistic	[Li and Yamanishi 1999]				.773	
Nb	probabilistic	[Yang and Liu 1999]				.795	
	decision trees	[Dumais et al. 1998]					.884
C4.5	decision trees	[Joachims 1998]				.794	
Ind	decision trees	[Lewis and Ringuette 1994]	.670				
Swap-1	decision rules	[Apté et al. 1994]		.805			
Ripper	decision rules	[Cohen and Singer 1999]	.683	.811		.820	
SleepingExperts	decision rules	[Cohen and Singer 1999]	**.753**	.759		.827	
Dl-Esc	decision rules	[Li and Yamanishi 1999]				.820	
Charade	decision rules	[Moulinier and Ganascia 1996]		.738			
Charade	decision rules	[Moulinier et al. 1996]		.783 (F_1)			
Llsf	regression	[Yang 1999]		.855	.810		
Llsf	regression	[Yang and Liu 1999]				.849	
BalancedWinnow	on-line linear	[Dagan et al. 1997]	.747 (M)	.833 (M)			
Widrow-Hoff	on-line linear	[Lam and Ho 1998]				.822	
Rocchio	batch linear	[Cohen and Singer 1999]	.660	.748		.776	
FindSim	batch linear	[Dumais et al. 1998]				.617	.646
Rocchio	batch linear	[Joachims 1998]				.799	
Rocchio	batch linear	[Lam and Ho 1998]				.781	
Rocchio	batch linear	[Li and Yamanishi 1999]				.625	
Classi	neural network	[Ng et al. 1997]		.802			
Nnet	neural network	Yang and Liu 1999]				.838	
	neural network	[Wiener et al. 1995]			**.820**		
Gis-W	example-based	[Lam and Ho 1998]				.860	
k-NN	example-based	[Joachims 1998]				.823	
k-NN	example-based	[Lam and Ho 1998]				.820	
k-NN	example-based	[Yang 1999]	.690	.852	**.820**		
k-NN	example-based	[Yang and Liu 1999]				.856	
	SVM	[Dumais et al. 1998]				.870	**.920**
SvmLight	SVM	[Joachims 1998]				.864	
SvmLight	SVM	[Li Yamanishi 1999]				.841	
SvmLight	SVM	[Yang and Liu 1999]				.859	
AdaBoost.MH	committee	[Schapire and Singer 2000]		**.860**			
	committee	[Weiss et al. 1999]				**.878**	
	Bayesian net	[Dumais et al. 1998]				.800	.850
	Bayesian net	[Lam et al. 1997]	.542 (MF_1)				

F_1 en Reuters

Sebastiani, F. Machine learning in automated text categorization, ACM Computing Surveys 34(1):1-47, 2002

Otra estrategia de vectorización: PPMI

Positive pointwise mutual information (PPMI): Se usa en matrices término – término.

Otra estrategia de vectorización: PPMI

Positive pointwise mutual information (PPMI): Se usa en matrices término – término.

PPMI mide cuan a menudo dos eventos ocurren, comparados con el valor esperado de ocurrencias si ambos eventos fueran independientes:

Observaciones conjuntas de w y c (c es un documento o contexto)

$$\text{PMI}(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)}$$

Observaciones que asumen independencia entre ambas palabras

Otra estrategia de vectorización: PPMI

Positive pointwise mutual information (PPMI): Se usa en matrices término – término.

PPMI mide cuan a menudo dos eventos ocurren, comparados con el valor esperado de ocurrencias si ambos eventos fueran independientes:

Observaciones conjuntas de w y c (c es un documento o contexto)

$$\mathrm{PMI}(w,c) = \log_2 \frac{P(w,c)}{P(w)P(c)}$$

Observaciones que asumen independencia entre ambas palabras

Para restringir los valores de PMI a los reales positivos, se aplica la función piso:

$$\mathrm{PPMI}(w,c) = \max(\log_2 \frac{P(w,c)}{P(w)P(c)}, 0)$$

Reemplaza los valores negativos con 0

Otra estrategia de vectorización: PPMI

Ej.:

Contextos de Wikipedia (c)

w

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

Otra estrategia de vectorización: PPMI

Contextos de Wikipedia (c)

Ej.:

w

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

$$
\begin{aligned}
P(\text{w=information,c=data}) &= \frac{3982}{11716} = .3399 \\
P(\text{w=information}) &= \frac{7703}{11716} = .6575 \\
P(\text{c=data}) &= \frac{5673}{11716} = .4842 \\
\text{ppmi}(\text{information,data}) &= \log 2(.3399/(.6575 * .4842)) = .0944
\end{aligned}
$$

Otra estrategia de vectorización: PPMI

Ej.:

Contextos de Wikipedia (c)

w

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

PPMI

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

Otra estrategia de vectorización: PPMI

Ej.:

Contextos de Wikipedia (c)

w

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

PPMI

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

Vector de contexto

Otra estrategia de vectorización: PPMI

Contextos de Wikipedia (c)

Ej.:

w

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

PPMI

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

Vector de la palabra

Vector de contexto