



Inteligencia Artificial

Vectorización de Textos

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingeniería Electrónica

Universidad Popular del Cesar

Aprendizaje Automático y PLN



El Aprendizaje Automático es esencial en el PLN para tareas como:

- Traducción automática
- Análisis de sentimientos
- Resumen de texto
- Q/A
- Extracción de entidades

Aprendizaje supervisado



El aprendizaje supervisado es un enfoque del aprendizaje automático en el que el modelo se entrena en un conjunto de datos etiquetados, es decir, datos que tienen una respuesta o salida conocida.



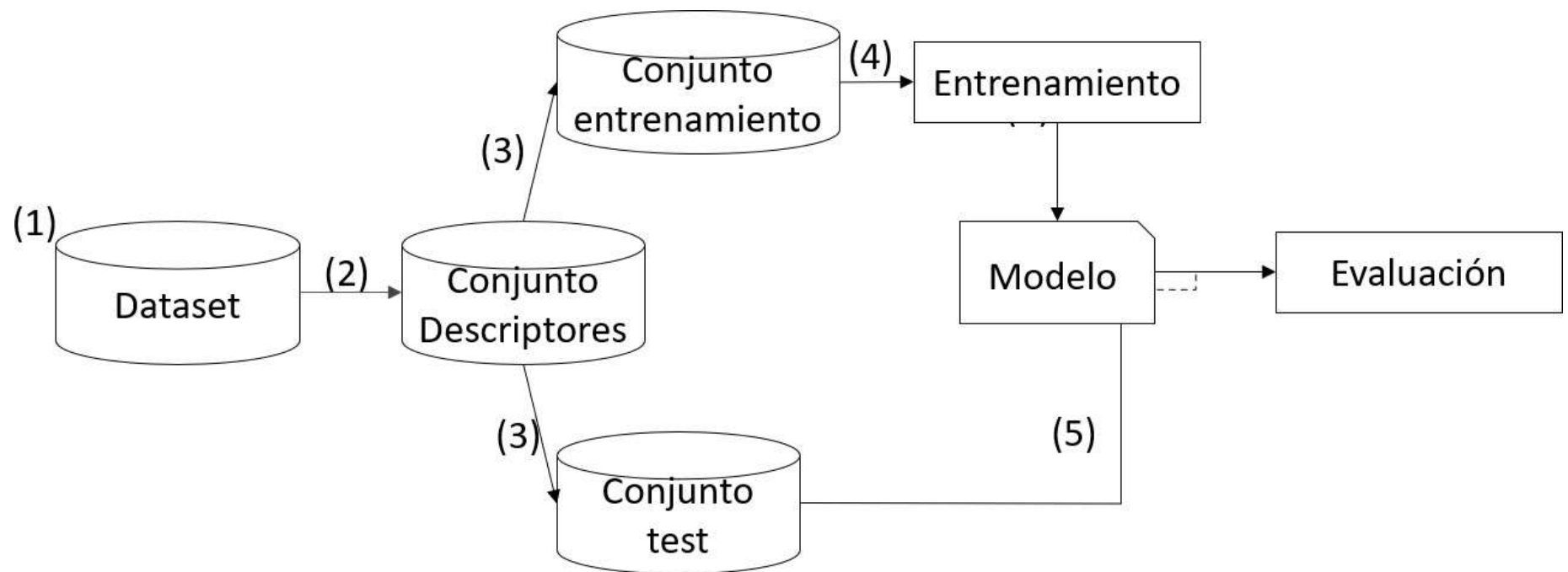
Ejemplo: Clasificación de correos electrónicos como spam o no spam en función de ejemplos previamente etiquetados

Flujo de trabajo

Proceso de aprendizaje supervisado:

1. Estructurar dataset inicial
2. **Extraer descriptores**
3. Partir dataset en dos (o tres) partes
4. Entrenar/construir un modelo de predicción
5. Evaluar el modelo

Flujo de trabajo





Vectorización de textos

Índice



REPRESENTACIÓN DE
TEXTOS

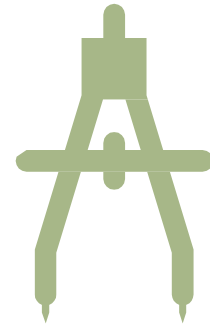


USO EN MODELOS DE
APRENDIZAJE
AUTOMÁTICO

Representación de texto



Modelos de aprendizaje trabajan
con vectores de números



Vectorización: proceso de
convertir texto a vectores
numéricos

Vectorización

Segmentar texto en palabras, y transformar cada palabra en un vector

Segmentar texto en caracteres, y transformar cada carácter en un vector

Extraer n-gramas de palabras o caracteres, y transformar cada n-grama en un vector

Vectorización

La inteligencia artificial está transformando muchas industrias.

Bi-gramas (2-gramas):

- "La inteligencia"
- "inteligencia artificial"
- "artificial está"
- "está transformando"
- "transformando muchas"
- "muchas industrias"

Tri-gramas (3-gramas):

- "La inteligencia artificial"
- "inteligencia artificial está"
- "artificial está transformando"
- "está transformando muchas"
- "transformando muchas industrias"

Vectorización

Inteligencia

Bi-gramas (2-gramas):

- "in"
- "nt"
- "te"
- "el"
- "li"
- "ig"
- "ge"
- "en"
- "nc"
- "ci"
- "ia"

Tri-gramas (3-gramas):

- "int"
- "nte"
- "tel"
- "eli"
- "lig"
- "ige"
- "gen"
- "enc"
- "nci"
- "cia"

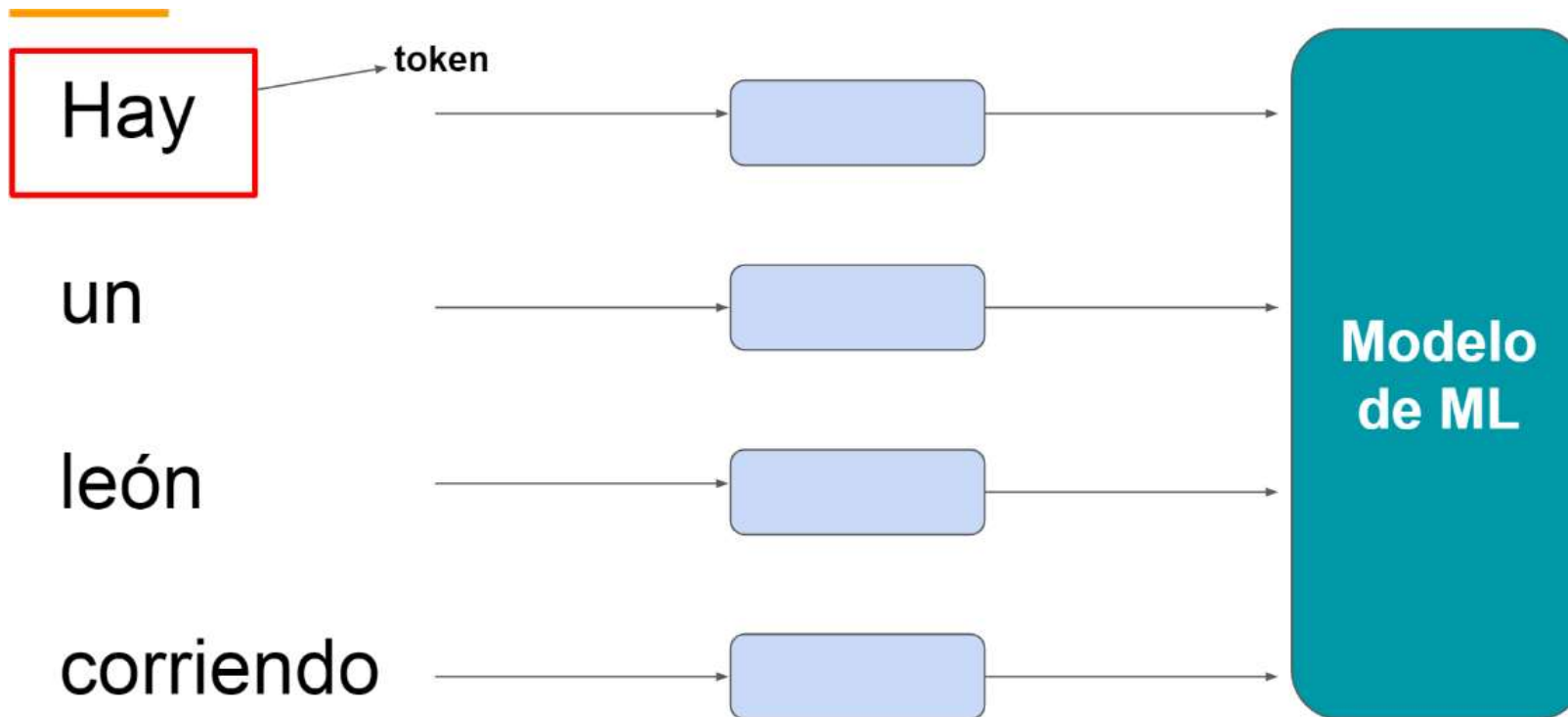
Tokenización

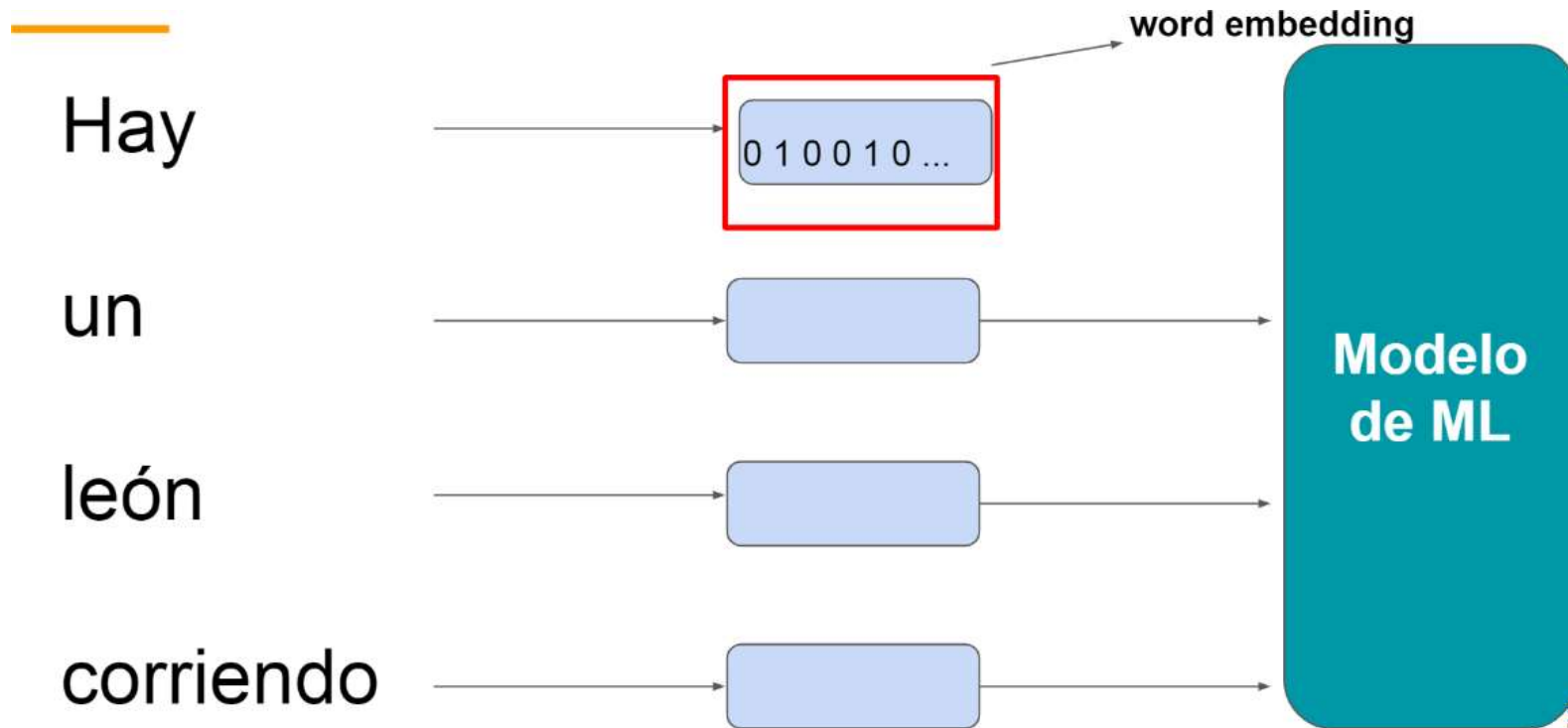
Proceso de partir texto en tokens (o elementos):

- Palabras
- Caracteres
- N grammas

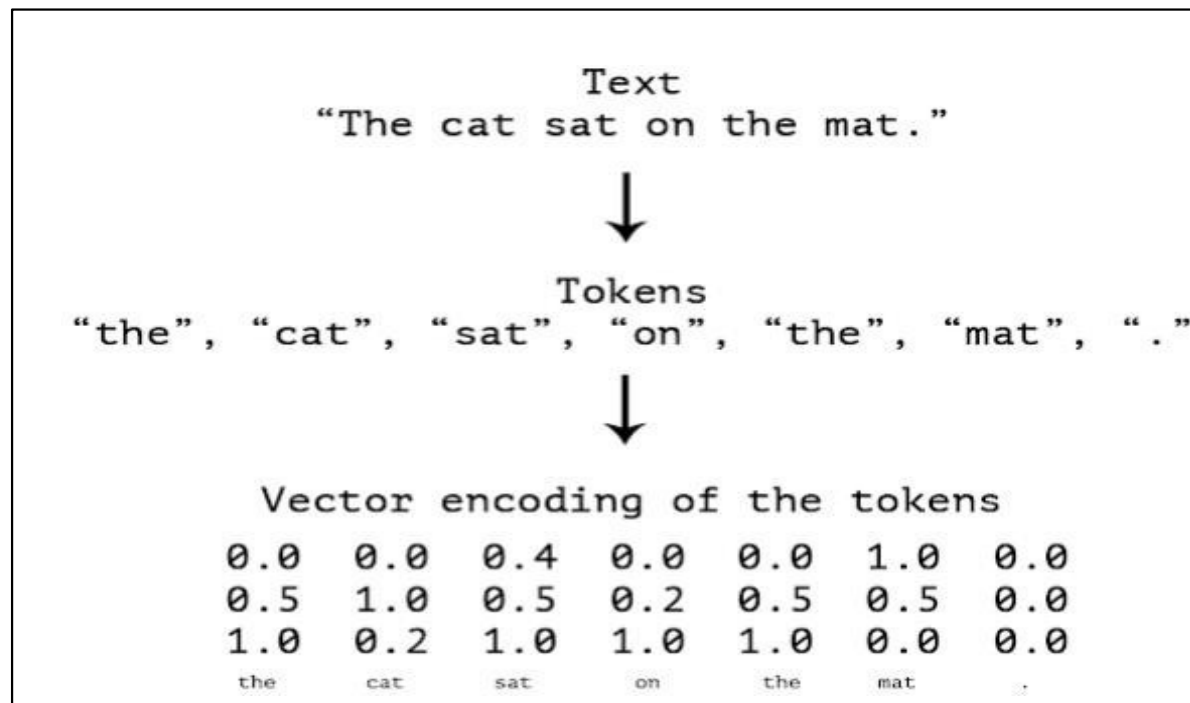
Tokenización

- Hay un león corriendo





Tokenización + Vectorización



Vectorización

¿Cómo asignamos un vector a un token?

- One-hot encoding
- Embedding

One-hot encoding

1. A cada token del vocabulario se le asigna un id (entero)
2. Se construye vector binario:
 - + Vector tiene tamaño N, donde N es el número de tokens del vocabulario
 - + Vector consta de todo 0s salvo un 1 en la posición id

Word	Number	
a	1	
able	2	
about	3	
...	...	
hand	615	
...	...	
happy	621	
...	...	
zebra	1000	

1	0	a
2	0	able
3	0	about
...	⋮	...
615	0	hand
...	⋮	...
621	1	happy
...	⋮	...
1000	0	zebra

Diagram illustrating the mapping of words to their IDs and the corresponding one-hot vector for the word "happy". The word "happy" has ID 621. The vector is a column of 1001 elements (indices 1 to 1000), where the element at index 621 is 1, and all other elements are 0.

Country				
India	1	0	0	0
Australia	0	1	0	0
Russia	0	0	1	0
America	0	0	0	1

Diagram illustrating the mapping of countries to their IDs and the corresponding one-hot vector for the word "happy". The word "happy" has ID 621. The vector is a column of 1001 elements (indices 1 to 1000), where the element at index 621 is 1, and all other elements are 0.

Sol [0, 1, 0, 0]

Playa [0, 0, 1, 0]

Hola [1, 0, 0, 0]

Feliz [0, 0, 0, 1]

Vocabulario: Hola, Sol, Playa

Word	Number				
a	1		1	0	a
able	2		2	0	able
about	3		3	0	about
...
hand	615		615	0	hand
...
happy	621	←→	621	1	happy
...
zebra	1000		1000	0	zebra

One-hot encoding

Problemas:

- Altas dimensiones para los vectores
- No hay relación entre las palabras

Representando palabras por su contexto

Hipótesis de distribución:

- *“You shall know by the company it keeps” [J. R. Firth (1957)]*
- *“Dime con quien andas y te dire quien eres”*
- Las palabras que ocurren en contextos similares tienden a tener significados similares

Distribución semántica

¿Qué significa gazpiña?



REAL ACADEMIA ESPAÑOLA



Diccionario de la lengua española Edición del Tricentenario Actualización 2022

Consulta posible gracias al compromiso con la cultura de la  Fundación "la Caixa"

por palabras ▼

Escriba aquí la palabra 

Consultar

Aviso: La palabra **gazpiña** no está en el Diccionario. Las entradas que se muestran a continuación podrían estar relacionadas:

Distribución semántica

Vamos a ver cómo se usa *gazpiña* en distintos contextos:

- Hay una jarra de gazpiña en el frigorífico
- A todo el mundo le gusta la gazpiña
- La gazpiña no me sentó bien y acabé borracho
- La gazpiña es muy azucarada

Distribución semántica

¿Qué puede significar *gazpiña*?

Distribución semántica

¿Qué puede significar *gazpiña*?

La *gazpiña* podría ser una bebida alcohólica y azucarada

¿Como llegamos a esa conclusión?

Distribución semántica

¿Qué otras palabras encajan?

- Hay una jarra de [REDACTED] en el frigorífico
- A todo el mundo le gusta la [REDACTED]
- La [REDACTED] no me sentó bien y acabé borracho
- La [REDACTED] es muy azucarada

Distribución semántica

¿Qué otras palabras encajan?

- Hay una jarra de [REDACTED] en el frigorífico
- A todo el mundo le gusta la [REDACTED]
- La [REDACTED] no me sentó bien y acabé borracho
- La [REDACTED] es muy azucarada

	(1)	(2)	(3)	(4)
Gazpiña	1	1	1	1
Música	0	1	0	1
Basura	0	0	0	0
Cerveza	1	1	1	0
Sangria	1	1	1	1

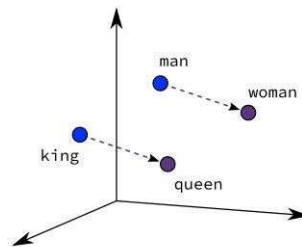
Distribución semántica

Las palabras que aparecen de manera frecuente en contextos similares tienen un significado similar

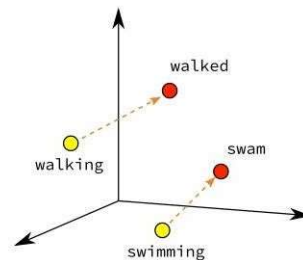
Palabras como vectores

Construimos un modelo de significado basado en la similitud:

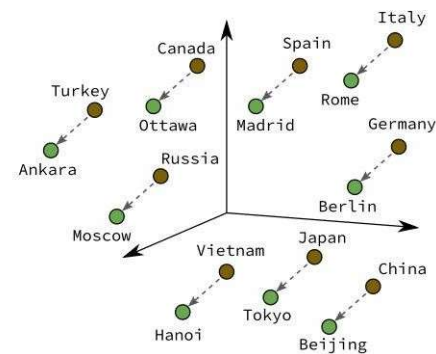
- Cada palabra es un vector
- Las palabras similares están cerca en el espacio



Male-Female



Verb Tense



Country-Capital

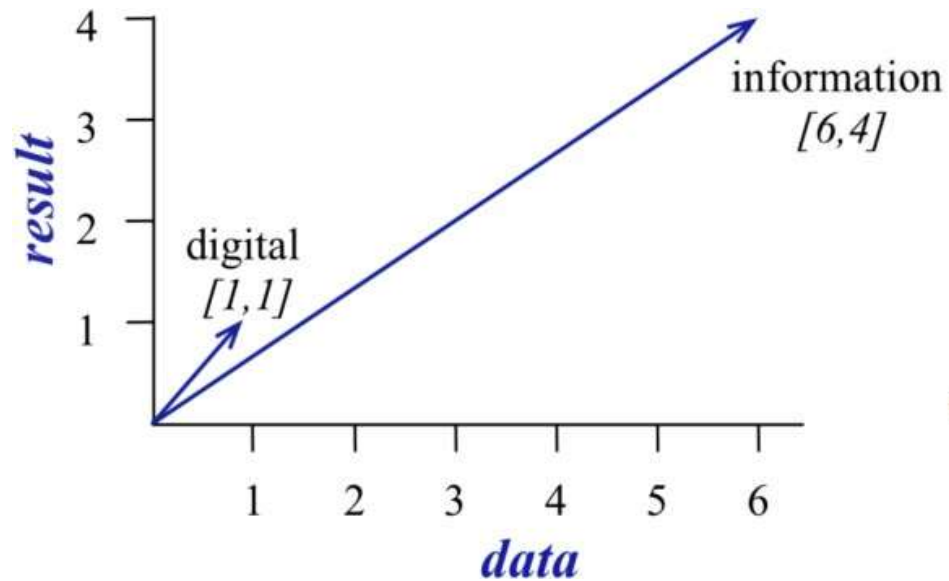
Una primera solución

Usar vectores de contexto para representar palabras

Matriz de co-ocurrencia palabra-a-palabra:

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

Similitud



$$\cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$$

$$\cos(\mathbf{u}, \mathbf{v}) = \frac{\sum_{i=1}^V u_i v_i}{\sqrt{\sum_{i=1}^V u_i^2} \sqrt{\sum_{i=1}^V v_i^2}}$$

Similitud (ejemplo)

	Vuelo	Animal	Casa
Perro	0	4	5
Gato	0	5	4
Avión	5	1	0

$$\cos(\text{perro}, \text{gato}) = \frac{0 * 0 + 4 * 5 + 5 * 4}{\sqrt{0^2 + 4^2 + 5^2} \sqrt{0^2 + 5^2 + 4^2}} = 0.97$$

$$\cos(\text{perro}, \text{avion}) = \frac{0 * 5 + 4 * 1 + 5 * 0}{\sqrt{0^2 + 4^2 + 5^2} \sqrt{5^2 + 1^2 + 0^2}} = 0.12$$

PPIM

Problema:

- No todos los conteos son iguales
- Palabras pueden ocurrir de forma aleatoria

Solución:

- PPIM (Positive Pointwise Mutual Information)

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

Que ocurre con la representacion basada en ocurrencias?

- A pesar de que las palabras tienen relacion
- Vectores son dispersos (mayoría 0s) y largos
- Nos gustaría vectores cortos (50-300) y densos.

Vectores densos

- Vectores cortos son más fáciles de usar en aprendizaje automático
- Generalizan mejor que simplemente contar
- Capturar mejor las relaciones
 - + w1 co-ocurre con coche, w1 co-ocurre con automóvil

Apreniendo embeddings

Buen embedding depende del problema

Mejor embedding para tarea de analizar valoraciones
será distinto al mejor embedding para clasificar textos
legales

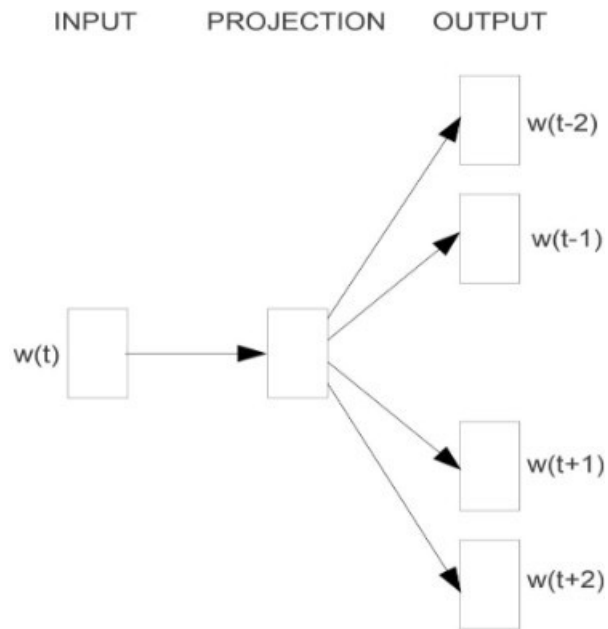
Embeddings pre-aprendidos

¿Qué ocurre si tenemos pocos datos?

- Utilizar embedding genérico
- [Word2Vec](#), [GloVe](#), [FastText](#)

Word2vec

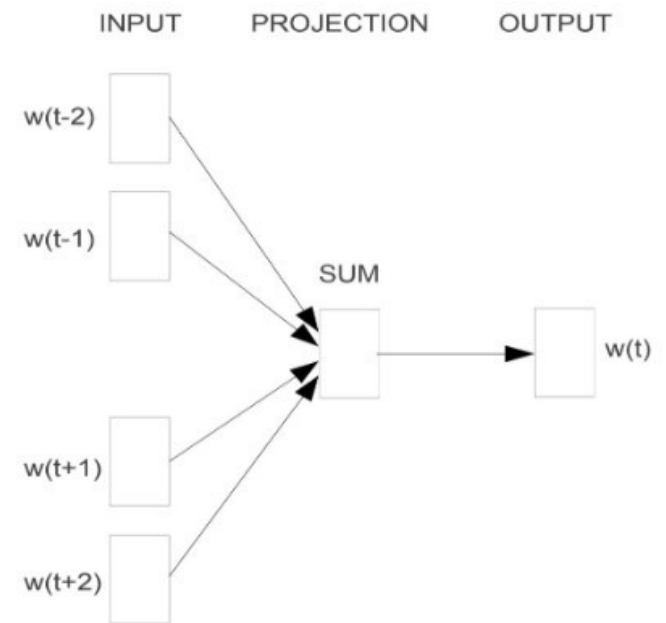
Skipgram



Skip-gram

predice el contexto dada una palabra central

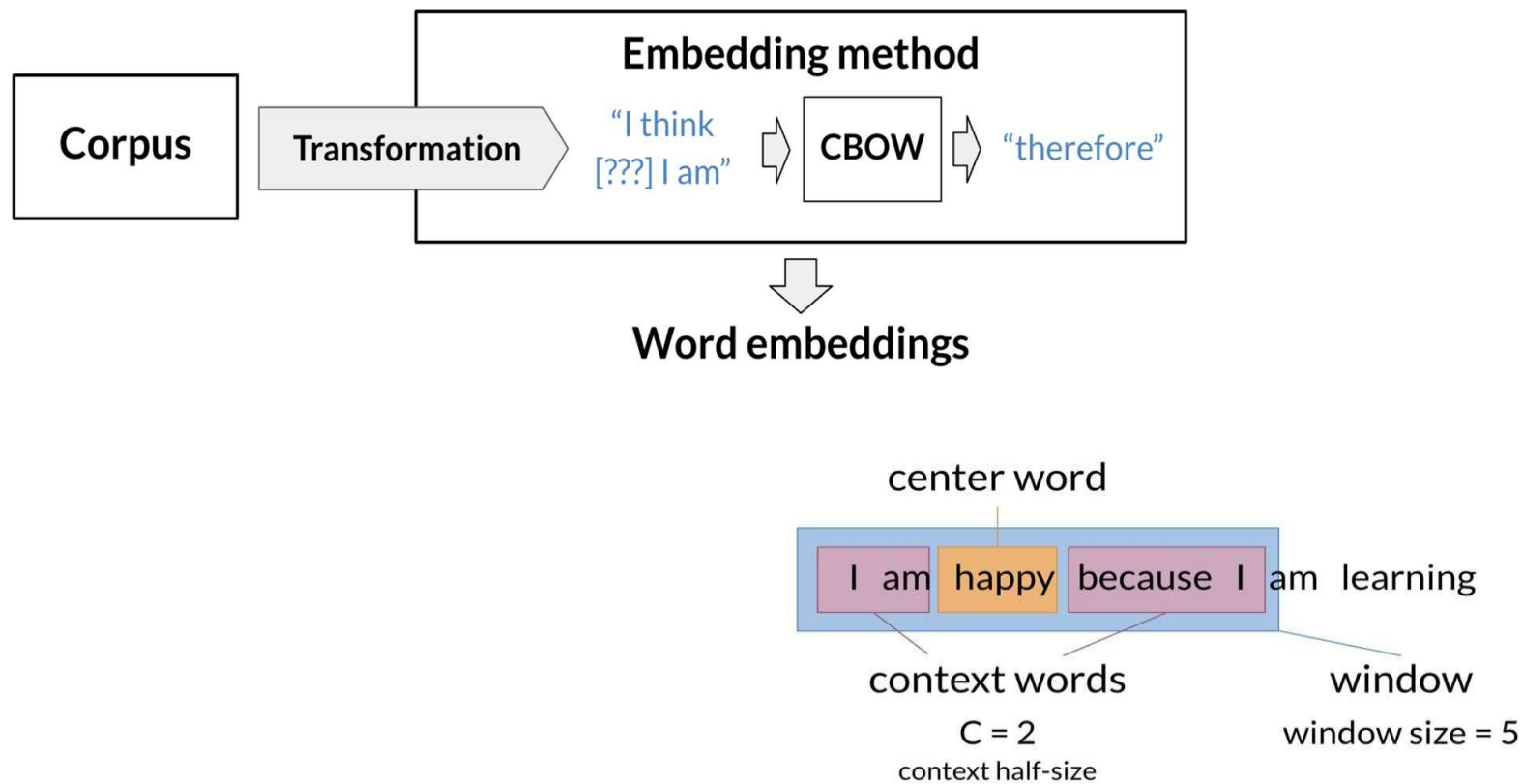
CBOW



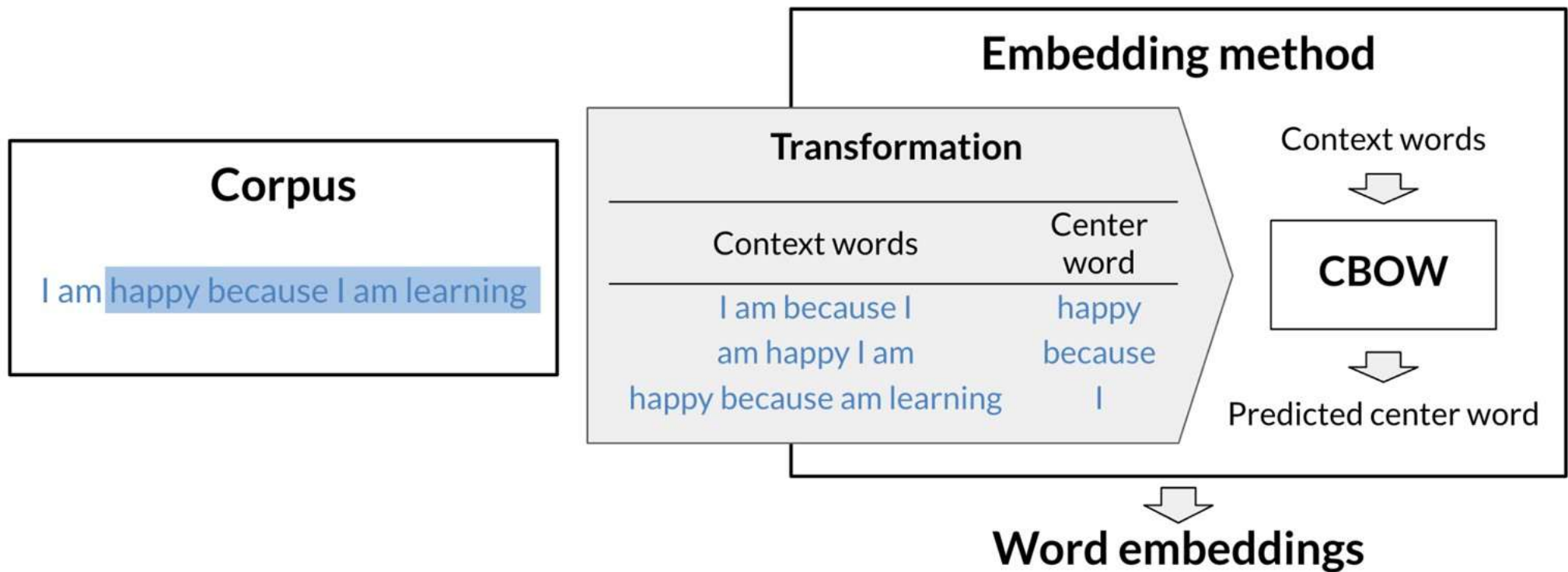
CBOW

predice la palabra central sumando los vectores de contexto

Modelo CBOW



Modelo CBOW



Glove

- Matriz de co-ocurrencias
 - ✕ Derivamos la relacion semantica entre las palabras
- Factorización de Matrices para reducir la dimensión de la matriz de co-ocurrencia
 - + Representaciones más compactas y significativas para cada palabra
- Vectores de Palabras

FastText

- Extensión de word2vec
- En lugar de usar palabras se usan n-gramas
- La palabra “artificial” con $n=3$ se divide en:
 - + <ar, art, rti, tif, ifi, fic, ici, cia, ial, al>

Espacio semántico

Conjunto de embeddings representados en el espacio forman un espacio semántico

https://lena-voita.github.io/nlp_course/word_embeddings.html#analysis_interpretability

Vecinos más cercanos

Closest to **frog**:

frogs

toad

litoria

leptodactylidae

rana

lizard

eleutherodactylus

litoria



leptodactylidae



rana



eleutherodactylus



Similitud de palabras

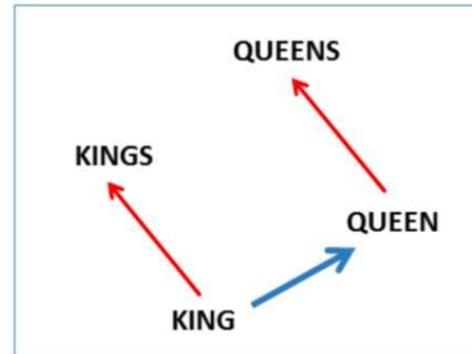
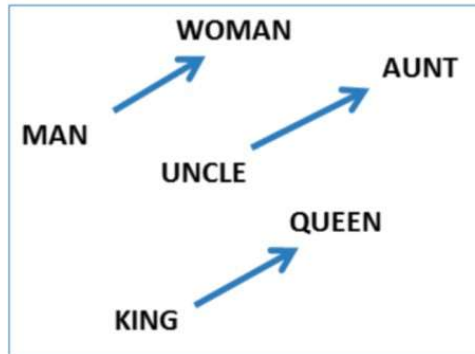
Rare words similarity benchmark

<u>word pair</u>		<u>score</u>
vulgarism	profanity	9.62
subdividing	separate	8.67
friendships	brotherhood	7.5
exceedance	probability	5.0
assigned	allow	3.5
marginalize	interact	2.5
misleading	beat	1.25
radiators	beginning	0

Relación lineal

semantic: $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$

syntactic: $v(\text{kings}) - v(\text{king}) + v(\text{queen}) \approx v(\text{queens})$



Analogías

Analogía: **a** es a **a*** como **b** es a _____

Tarea: $v(a^*) - v(a) + v(b) \approx ?$

<u>relation</u>	<u>word pair 1</u>	<u>word pair 2</u>
man-woman	brother sister	grandson granddaughter
currency	Angola kwanza	Iran rial
opposite	possibly impossible	ethical unethical
past tense	walking walked	swimming swam
superlative	easy esiest	lucky luckiest

Índice

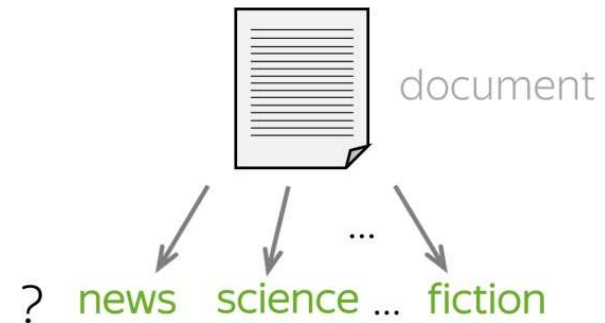
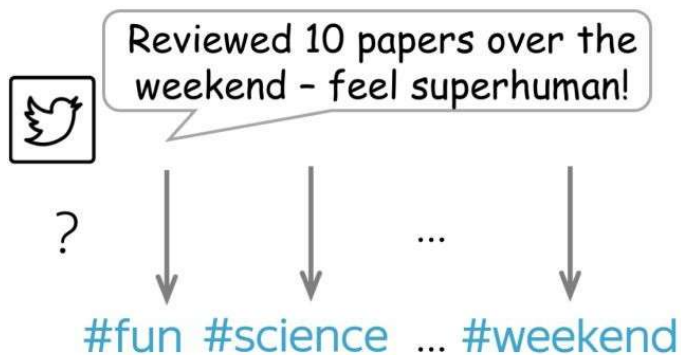
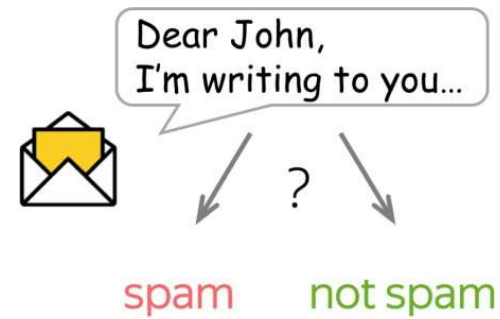


REPRESENTACIÓN DE
TEXTOS

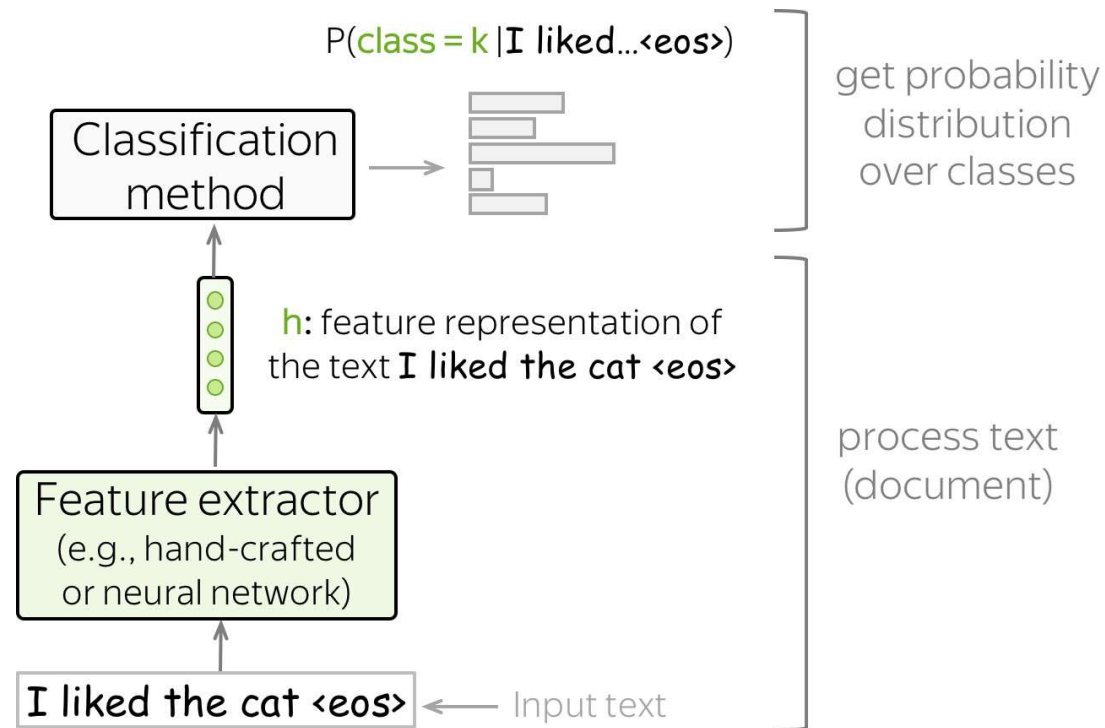


USO EN MODELOS DE
APRENDIZAJE
AUTOMÁTICO

Clasificación de texto



Clasificación de texto



Bolsa de palabras (bag of words)

Frecuencia de palabras en cada frase

```
messages = ["Hey hey hey lets go get lunch today :)",  
            "Did you go home?",  
            "Hey!!! I need a favor"]
```

	did	favor	get	go	hey	home	lets	lunch	need	today	you
0	0	0	1	1	3	0	1	1	0	1	0
1	1	0	0	1	0	1	0	0	0	0	1
2	0	1	0	0	1	0	0	0	1	0	0

Bolsa de palabras

Palabras muy comunes tienen mucha más importancia que palabras “raras”

TF-IDF

Frecuencia de términos (*term frequency, tf*)

Frecuencia inversa de documentos (*inverse document frequency, idf*)

TF-IDF:

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D)$$

TF-IDF

2 cálculos:

- A través de TF (Term Frequency / Frecuencia de término) se calcula la frecuencia relativa de una palabra en un documento entre el número total de palabras del documento.
- Con IDF se calcula la frecuencia inversa de documento (Inverse Document Frequency) dividiendo el número total de documentos entre el número de documentos que contienen el término.

De esta manera, mediante TF-IDF es posible el cálculo de la relevancia y de los pesos de los términos en los documentos

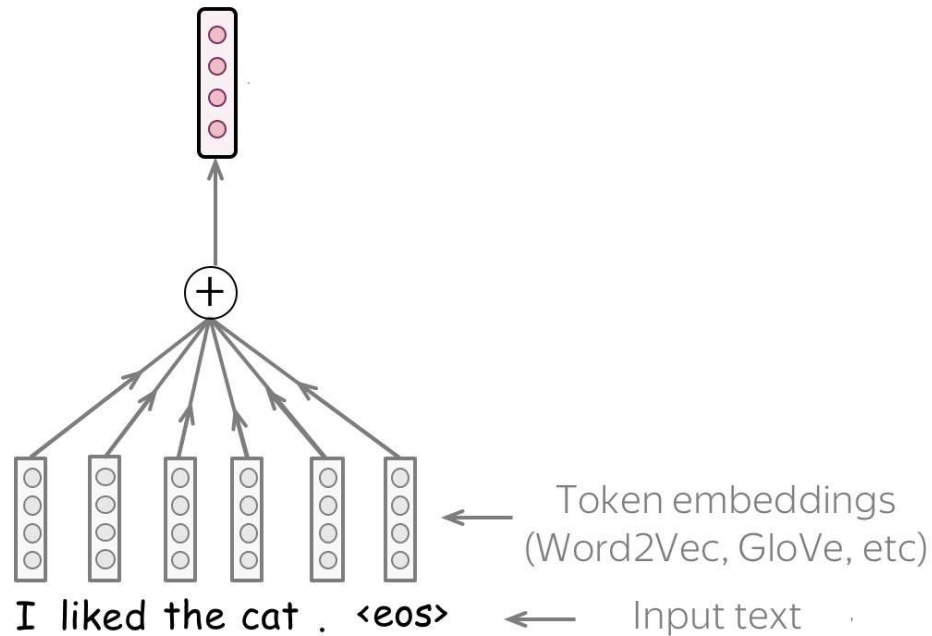
Limitaciones

- Dimensión muy grande
- No tienen en cuenta el orden
- Vectores sparse

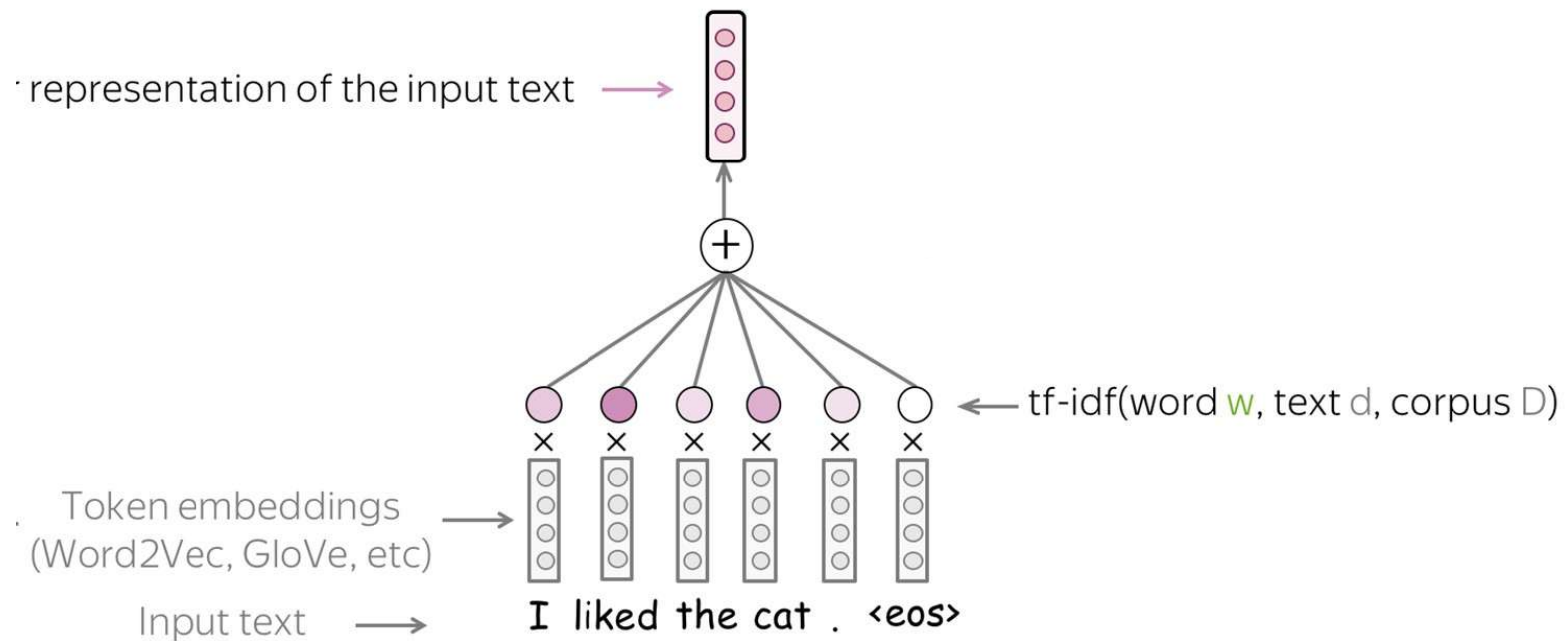
Embeddings

- Embeddings dan representación compacta
- Algoritmos de ML necesitan un vector, pero las frases tienen distinta longitud
- No sirve con concatenar los vectores

Bolsa de embeddings (bag of embeddings)



Bolsa de embeddings con peso



Métodos de clasificación

Los ya conocidos:

- SVM
- KNN
- Redes neuronales
- Árboles de decisión
- ...

Bolsa de embeddings con peso

