uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Procesamiento de Lenguaje Natural con Aprendizaje Profundo,

Máster en Ciencia y Tecnología Informática

Tema 1.2. Tareas básicas para representación de textos en aplicaciones de NLP

Isabel Segura Bedmar

Objetivos

- Comprender la necesidad de representar los textos para procesarlos.
- Conocer las principales técnicas para reducir la variabilidad de los textos.
- Estudiar los modelos tradicionales para la representación de textos, como el modelo bolsa de palabras y tf-idf.
- Conocer las ventajas y desventajas de estos modelos tradicionales.

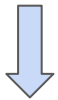
Índice

- Representación de textos
- Modelos tradicionales:
  - bolsa de palabras
  - tf-idf.

Representación de textos, ¿qué es?

- Transformar un texto en un vector de números.

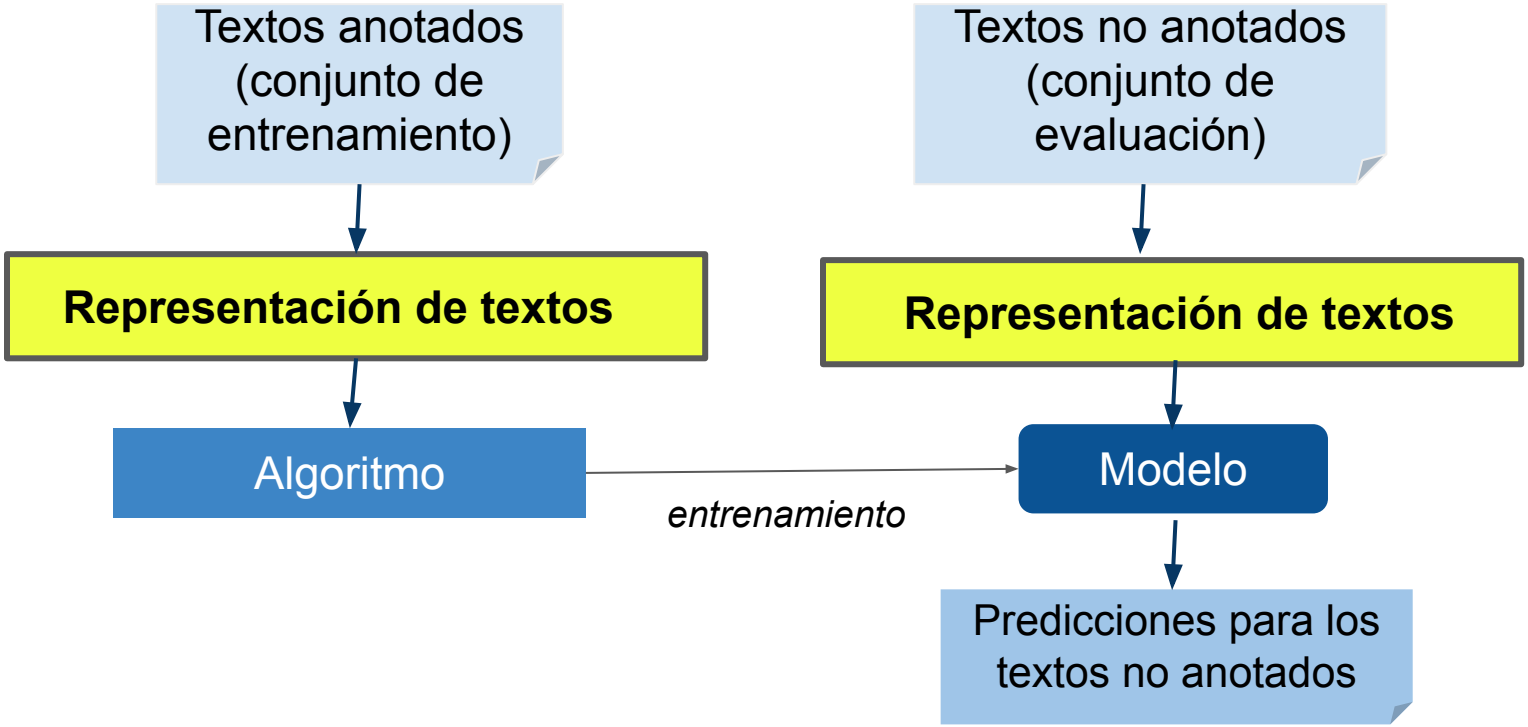
The big cat is on the table



ball	big	cat	moon	small	table	tree	…	zoo
0	1	1	0	0	1	0	…	0

- La dimensión del vector depende del tamaño del vocabulario (conjunto de palabras distintas en la colección de textos).

Arquitectura de un sistema PLN basado en de aprendizaje automático



Limpiar textos

- Antes de la transformación a vectores, es deseable realizar algunas **tareas** de limpieza que permitan **reducir la variabilidad del lenguaje**:
  - Transformar a minúsculas
  - Eliminar signos de puntuación, números, etc (puedes utilizar patrones).
  - Eliminar stopwords.
  - Lematización.
  - Stemming.

Stopwords

- Palabras más comunes en un idioma y que no añaden significado relevante al texto.
- Ejemplos de stopwords en inglés: *“the”, “is”, “in”, “for”, “where”, “when”, “to”, “at” etc.*
- Ejemplos de stopwords en español: *de, que, el, en y, a ,los, se, del, las, un, por, con, no una, etc.*
- Eliminar stopwords es recomendable en algunas tareas de PLN como la clasificación de textos o recuperación de información.
- En un dominio concreto (por ejemplo, el domino clínico) la lista de stopwords puede ser ampliada con palabras comunes en dicho dominio: paciente, médico, ambulancia, etc.

Stopwords

- Ventajas al eliminar stopwords:
  - Disminuye el tamaño del vocabulario.
  - Disminuye la dimensionalidad en modelos de representación como la bolsa de palabras o tf-idf.
  - Reduce ruido, el algoritmo se puede centrar en las palabras que sí aportan semántica al texto.
- Sin embargo, las stopwords no deben ser eliminadas en otras aplicaciones de PLN como la traducción automática o el reconocimiento de entidades.

Lematización

- Dada una palabra, la lematización consiste en devolver su lema o forma canónica (palabra que aparece en un diccionario).
- Ejemplos: *comió, comiendo, comeré, come -> comer.*
- Debe ser aplicada en aplicaciones de PLN como la clasificación de textos o recuperación de información:
- Ventajas:
  - Disminuye tamaño del vocabulario
  - Disminuye dimensionalidad en modelos de representación como la bolsa de palabras y tf-idf.
  - Reduce ruido.
- Varias librerías de PLN: ntlk, spacy.

Stemming

- Similar a la lematización, dada una palabra, devuelve su raíz.
- Basada en algoritmos como Porter o Lancaster.
- Algunos ejemplos de reglas:
  - *SSES -> SS (caresses -> caress)*
  - *S -> (cats -> cat)*
  - *EED -> EE (agreed -> agree, feed -> feed)*
  - *ATOR -> ATE (operator -> operate)*
  - *ER -> (airliner -> airlin)*
- Stemming es más eficiente que la lematización, pero menos robusta. Ejemplos de errores: stories -> stori, leaves -> leav, horses -> hors, better -> better (debería ser good).

Índice

- Representación de textos
- Modelos tradicionales:
  - bolsa de palabras
  - tf-idf.

Modelos de representación de textos

- Una vez que el texto ha sido limpiado, es posible aplicar distintas técnicas para transformar los textos a vectores.
- Algunos de los enfoques más populares son:
  - Modelo de bolsa de palabras
  - Modelo tf-idf
  - Modelos word embedding (entrenados con redes de neuronas).

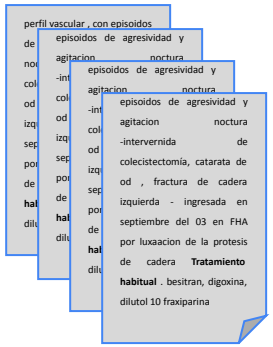
Índice

- Representación de textos
- Modelos tradicionales:
  - **bolsa de palabras**
  - tf-idf.

Modelo Bolsa de Palabras (Bag of Words, BoW)

- Dada una colección de textos (previamente limpiados), se obtiene su vocabulario.
- El **vocabulario** es el conjunto de todas las **palabras distintas** (sin incluir las repeticiones) de la colección de textos.
- En el vocabulario, las palabras están ordenadas alfabéticamente y **cada palabra** es **representada** con un número entero (**índice**), que indica su posición en el vocabulario.
- Cada **texto** puede ser **representado** como un **vector** cuya dimensión es el tamaño del vocabulario. Cada posición del vector representa una de las palabras del vocabulario.
- El valor asociado a cada posición del vector es el **número de veces que ocurre esa palabra en el texto**.

Modelo Bolsa de Palabras (Bag of Words, BoW)



Fuente: Smashicons Flaticon

0	ball
1	big
2	cat
3	moon
4	small
5	table
6	tree
7	window
8	zoo

Conjunto textos de entrenamiento

Bolsa de Palabras

Vocabulario

Modelo Bolsa de Palabras (Bag of Words, BoW)

Texto:

The big cat is on the table and the small cat in the window.

Texto limpio:

~~The~~ big cat ~~is on the~~ table ~~and the~~ small cat ~~in the~~ window~~.~~

ball	big	cat	moon	small	table	tree	window	zoo
0	1	2	0	1	1	0	1	0

Modelo Bolsa de Palabras (Bag of Words, BoW)

D1: ~~The~~ big cat ~~is on the~~ table ~~and the~~ small cat ~~in the~~ window
D2: ~~The~~ table ~~and the~~ window ~~are~~ small
D2: ~~The~~ moon ~~and the~~ small tree ~~are~~ big

	ball	big	cat	moon	small	table	tree	window	zoo
D1	0	1	2	0	1	1	0	1	0
D2	0	0	0	0	1	1	0	1	0
D3	0	1	0	1	1	0	1	0	0

Modelo TF-IDF

- Versión extendida del modelo bolsa de palabras.
- Cada texto es representado usando tf-idf de cada palabra en el vocabulario.
- Se utiliza **TF-IDF** porque consigue **disminuir** el **peso** de aquellas **palabras que son muy comunes** en la colección de textos.

TF-IDF

- Term frequency - inverse document frequency de la palabra w en el document d

TF-IDF(w,d) = TF(w,d) * IDF(w)

  - TF(w,d) = frecuencia de la palabra w en el documento d
  - IDF(w) = *inverse document frequency.* Logaritmo del cociente entre el número total de documentos (N) y el número de documentos que contienen a la palabra w

$$IDF(w) = log(\frac{N}{|d \in D : w \in d|})$$

Bolsa de Palabra

	ball	big	cat	moon	small	table	tree	window	zoo
D1	0	1	2	0	**1**	1	0	1	0
D2	0	0	0	0	**1**	1	0	1	0
D3	0	1	0	1	**1**	0	1	0	0

TF-IDF

	ball	big	cat	moon	small	table	tree	window	zoo
D1	0	0.17	0.95	0	**0**	0.17	0	0.17	0
D2	0	0	0	0	**0**	0.17	0	0.17	0
D3	0	0.17	0	0.47	**0**	0	0.47	0	0

Ventajas de los modelos BoW y TF-IDF

- Fáciles de implementar.
- Buenos resultados en tareas de clasificación de textos.

Desventajas de los modelos BoW y TF-IDF

- Los vectores tienen una gran dimensionalidad (tamaño del vocabulario) y la información es escasa y dispersa (muchos 0s).
- No pueden capturar información semántica. Por ejemplo, estas dos expresiones tienen un significado similar, pero tendrán vectores diferentes:
  - *Edema de glotis != hinchazón de la laringe*

Desventajas de los modelos BoW y TF-IDF

- No tienen en cuenta la posición de las palabras. De esta forma, las dos siguientes oraciones son representadas con el mismo vector, pero sin embargo tienen significados opuestos:
  - *The hotel was very good and not expensive !=*
  - *The hotel was very expensive and not good*

Cómo implementar los modelos BoW y TF-IDF

- Aunque sería posible desarrollar ambos modelos paso a paso, afortunadamente, la librería sklearn, ya hace este trabajo para nosotros y nos proporciona clases CountVectorizer para el modelo bolsa de palabras, y TfidfVectorizer para el modelo tf-idf.

- En el siguiente link puedes encontrar un ejemplo de cómo utilizar estas clases.

Resumen

- **PLN** basado en **aprendizaje automático** implica representación de **textos** (**transformación** a **vectores** de números).
- Aplicar técnicas para **reducir** la **variabilidad** del **lenguaje**.
- **Bolsa de palabras** y **tf-idf** basados en la **frecuencias** de las palabras en los textos.
- Son modelos **eficientes**.
- Sin embargo, presentan algunas limitaciones: **alta dimensionalidad** y su **imposibilidad** de capturar **información semántica**.

OpenCourseWare
Procesamiento de Lenguaje Natural con Aprendizaje Profundo,

Gracias!!!

https://github.com/isegura