Actividad 2: Representación de texto

Teoría

1. ¿Qué son las features en aprendizaje automático?
   1. Las features en la representación de texto son unidades de información extraídas de datos textuales, que se utilizan como entrada para modelos de aprendizaje automático. Estas características pueden variar desde simples representaciones basadas en palabras individuales hasta complejas representaciones semánticas y sintácticas.
2. ¿Para qué usamos representaciones de texto?
   1. El texto en lenguaje natural se transforma en un formato que pueda ser entendido y manipulado por algoritmos de aprendizaje automático y otros sistemas computacionales. Estas representaciones permiten a las máquinas interpretar, analizar, y generar texto de manera efectiva.
3. Explicar en qué consiste el proceso de vectorización, incluya los elementos que se consideran necesarios para llevarlo a cabo.
   1. Es el proceso de convertir palabras, frases o documentos completos en un formato numérico que una máquina puede entender y procesar. En esencia, se trata de tomar texto y representarlo como un vector de números, donde cada dimensión del vector captura cierta información sobre el texto.
   2. Tokens: Es el proceso de dividir el texto en unidades más pequeñas llamadas tokens. Los tokens pueden ser palabras individuales, subpalabras o caracteres, dependiendo de la granularidad deseada.
   3. Vocabulario: En muchos métodos de vectorización, es necesario construir un vocabulario, que es un conjunto de todas las palabras únicas (o tokens) presentes en el corpus de entrenamiento. Cada palabra del vocabulario se asigna a un índice único.
   4. Método de vectorización: One hot encoding, Bag of words, Bag of N-Grams, TF-IDF
4. Haga una tabla comparativa entre representaciones clásicas que incluya, Nombre, representación del texto, tamaño de vectores, sparsity, captura similaridad, OOV, Normalización, ventajas y desventajas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre** | One Hot Encoding | Bag of Words (BoW) | Bag of N-grams | Term Frequency-Inverse Document Frequency (TF-IDF) |
| **Representación del texto** | Cada palabra se representa como un vector binario, donde solo una dimensión es 1 (indicando la palabra) y todas las demás son 0. | Cada documento se representa como un vector basado en el recuento de palabras. Cada dimensión corresponde a una palabra en el vocabulario. | Similar a BoW, pero cada dimensión representa un n-grama (secuencia de N palabras). | Cada documento se representa como un vector, pero las palabras son ponderadas por su frecuencia inversa en el corpus, dándole mayor peso a las palabras raras. |
| **Tamaño de vectores** | Igual al tamaño del vocabulario (número de palabras únicas en el corpus). | Igual al tamaño del vocabulario (número de palabras únicas en el corpus). | Mayor que BoW, porque incluye combinaciones de palabras (n-gramas). | Igual al tamaño del vocabulario (número de palabras únicas en el corpus). |
| **Sparsity** (Dispersión) | Muy alto: la mayoría de los elementos del vector son 0. | Alto: la mayoría de los elementos son 0, especialmente para documentos cortos en un vocabulario grande. | Muy alto: debido a la inclusión de múltiples n-gramas, lo que aumenta la dimensionalidad y la dispersión. | Alto: similar a BoW, pero con menos dispersión debido a la ponderación de los términos. |
| **Captura similaridad** | No captura similitud entre palabras, ya que cada palabra tiene su propia dimensión sin interrelación. | No captura similitud entre palabras, pero palabras que ocurren juntas en documentos pueden estar correlacionadas. | Captura cierta similitud a nivel de frases cortas (n-gramas). | Captura la similitud en función de la frecuencia de términos y su relevancia en el corpus. |
| **OOV (Out-Of-Vocabulary)** | No maneja bien palabras fuera del vocabulario: cualquier palabra no vista previamente no tiene una representación adecuada. | Similar a One Hot Encoding, OOV no tiene una representación directa. | Similar a One Hot Encoding y BoW, OOV no tiene una representación directa. | Igual que en BoW, no maneja bien palabras fuera del vocabulario. |
| **Normalización** | No se requiere. | No se requiere, pero se puede aplicar para mejorar la interpretación (ej., L2 normalization). | No se requiere, pero es recomendable, especialmente con n-gramas largos. | La normalización puede mejorar el rendimiento, a menudo se aplica (ej., L2 normalization). |
| **Ventajas** | Simplicidad, fácil de implementar. | Simplicidad, buena para tareas básicas de NLP, no depende del contexto. | Captura más contexto que BoW debido a la inclusión de frases. | Captura la relevancia de las palabras, reduciendo el impacto de palabras comunes. |
| **Desventajas** | Ineficiente para grandes vocabularios, no captura similitudes, alta dispersión. | No considera el orden de las palabras, alta dispersión, no captura similitud semántica. | Aumenta la dimensionalidad, lo que puede llevar a problemas de dispersión y eficiencia. | Computacionalmente más costoso que BoW, no captura relaciones semánticas profundas. |

Práctica

En esta actividad práctica aplicar los conceptos vistos en la teoría. Para ello realizar una copia, ejecutar y analizar el colab 05 - Representación de texto: Enfoques básicos de representac… para responder las siguientes preguntas a modo de reflexión:

1. Indique qué diferencias encuentra entre representar un documento aplicando las siguientes técnicas de representación
   1. Bag of words (Binario y no binario).
      1. Binario: solo toma en cuenta que la palabra esté presente. Pone un 1 si está y 0 si no está.
      2. No binario: cuenta las veces que aparece una palabra en una oración. Pone 0 si no está y N si está, siendo N las ocurrencias.
   2. One hot encoding 🡪 Puede ser representado en un array de única dimensión o en una matriz de dos dimensiones (la dimensión de la matriz depende de la cantidad de documentos)
2. ¿Qué detalles captura Bag of N-grams? Indique cómo se aplicó para representar el texto.
   1. Bag of N-Grams es capaz de capturar algo de contexto analizando relaciones entre palabras del vocabulario. En el colab se dejó todo listo para que usara bigrams y trigrams.
3. ¿Qué detalles captura TF-IDF? qué diferencia notó con diferencia al resto de las representaciones.
   1. TF-IDF trata de capturar si alguna palabra en el corpus es más importante o no que otra. Utiliza dos métricas.
   2. TF: Term Frecuency. Mide cuantas veces aparece una palabra en un documento específico
   3. IDF: Inverse Document Frecuency. Mide qué tan común o rara es una palabra en el corpus completo. Si aparece en varios documentos, su IDF será bajo. Si aparece poco, será alto.
   4. TF-IDF: es el producto de TF y IDF, que evalúa la importancia de una palabra en un documento específico mientras ajusta esta importancia según la frecuencia de la palabra en todo el corpus.
4. Aplique una técnica vista en esta práctica a un corpus de texto pequeño y uno grande de su elección y observar cómo varía las dimensiones de vectores. ¿Qué problema puede encontrar al cambiar de un texto a otro? ¿Sirve la representación del corpus pequeño al mayor? ¿Por qué?

Elegí la técnica Bag of words (no binaria) y los siguientes textos.

Texto pequeño: “ayer fue viernes”

Texto grande: “el jueves pasado se ha disputado un partido de futbol entre el club atletico boca juniors y el cruzeiro esporte clube resultando en victoria por un solo gol para el equipo local cuyo tanto fue anotado por el delantero edinson cavani”

Al representar todo esto usando Bag of words, el vocabulario resultó en un vector muy grande. Por lo cual la representación del texto pequeño quedó como

[[0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]]

Mientras que la del texto grande quedó como

[[1 1 0 1 1 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 0]]

Me queda una dimensión muy grande para representar todo el texto pequeño.

Vocabulario: ['anotado' 'atletico' 'ayer' 'boca' 'cavani' 'club' 'clube' 'cruzeiro'

'cuyo' 'de' 'delantero' 'disputado' 'edinson' 'el' 'en' 'entre' 'equipo'

'esporte' 'fue' 'futbol' 'gol' 'ha' 'jueves' 'juniors' 'local' 'para'

'partido' 'pasado' 'por' 'resultando' 'se' 'solo' 'tanto' 'un' 'victoria'

'viernes']