Luis Maximiliano López Ramírez | 29/09/2024

A00833821

1. Estrategia de Vectorización TF-IDF

- ¿Cómo se calcula?

TF-IDF (Term Frequency - Inverse Document Frequency) es una técnica para convertir documentos de texto en vectores numéricos, utilizada frecuentemente en tareas de procesamiento de lenguaje natural (NLP). Se calcula a partir de dos componentes principales:

1. Frecuencia de Término (TF): mide la frecuencia de un término en un documento específico. Se define cómo:

$$TF(t,d) = \frac{\text{Número de veces que el término } t \text{ aparece en el documento } d}{\text{Número total de términos en el documento } d}$$

2. Frecuencia Inversa de Documento (IDF): mide la importancia del término en todo el corpus. Se calcula como:

$$IDF(t,D) = \log\left(\frac{N}{1 + \text{Número de documentos en las que aparece } t}\right)$$

Donde:

- N es el número total de documentos
- $1+$ en el denominador se usa para evitar división por cero.

El valor TF-IDF se calcula multiplicando ambos componentes:

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

¿En qué situaciones es más efectivo usar TF-IDF para tareas de clasificación de texto?

TF-IDF es más efectivo cuando se necesita:

1. Reducir el peso de las palabras comunes como "el", "la", "de", etc., que pueden aparecer en muchos documentos, pero no tienen mucha importancia para clasificar el contexto específico.

2. Destacar términos específicos que son relevantes para una clase o categoría particular.

3. Clasificar documentos: cuando se tiene un corpus de texto y se necesita distinguir entre distintas categorías con diferentes vocabularios.

Es menos adecuado para tareas donde se necesita capturar la secuencia o el contexto sintáctico de las palabras (por ejemplo, en modelos basados en redes neuronales recurrentes o transformadores.

¿Con qué bibliotecas se puede implementar?

- Scikit-learn (sklearn.feature_extraction.text.TfidfVectorizer)
- NLTK (aunque no tiene implementaciones directas con TF-IDF, permite calcularlo manualmente con nltk.FreqDist y fórmulas de IDF
- Gensim (gensim.models.TfidfModel)
- SpaCy (puede usarse en combinación con Scikit-learn para generar TF-IDF a partir de tokens preprocesados.)

2. Laplace Smoothing para N-gramas

¿Qué problema de los N-gramas resuelve "laplace smoothing"?

El suavizado de Laplace resuelve el problema de asignar probabilidad cero a las secuencias de palabras que no aparecen en el conjunto de entrenamiento. En un modelo de N-gramas, la probabilidad de un N-grama específico se calcula contando las frecuencias observadas en el corpus de entrenamiento. Si una combinación de palabras (N-gram) no se encuentra en el entrenamiento, se le asigna una probabilidad de cero, la cual afecta negativamente la estimación general.

¿Cómo trabaja?

El suavizado de Laplace añade un valor constante (generalmente 1) a todas las frecuencias de N-gramas, de modo que ninguna frecuencia es cero. La probabilidad de un N-gram se calcula como:

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + 1}{C(w_{i-1}) + |V|}$$

Donde:
- $C(w_{i-1}, w_i)$ es la frecuencia del N-grama en el corpus de entrenamiento
- $C(w_{i-1})$ es la frecuencia de la secuencia previa.
- $|V|$ es el tamaño del vocabulario (número total de palabras únicas)

¿Qué pasa con un modelo de NLP cuando se emplea esta técnica?

Cuando se aplica el suavizado de Laplace:

1. Todas las palabras (conocidas las no observadas) reciben una probabilidad mínima distinta de cero.
2. Se evita el problema de probabilidad cero, asegurando que cualquier nueva secuencia tenga una probabilidad positiva, lo que permite al modelo manejar vocabularios incompletos o secuencias inusuales.
3. Puede disminuir el sesgo, pero introduce un error sistemático, ya que aumenta artificialmente la probabilidad de palabras raras.

3. Modelado de Palabras OOV en N-gramas

¿Qué pasa cuando una palabra en el test set no se encuentra en el vocabulario del modelo de los N-gram?

→

Cuando una palabra en el conjunto de prueba (test-set) no se encuentra en el vocabulario del modelo (se le asigna una probabilidad de cero sino se implementa ninguna técnica de manejo de vocabulario desconocido. Esto causa que cualquier oración contenga esa palabra tenga una probabilidad total de cero, lo que afecta negativamente la predicción del modelo.

¿Cómo se puede modelar la probabilidad de palabras out-of-vocabulary (OOV)?

1. Usar suavizado de Laplace o suavizado de Lidstone para asignar una probabilidad pequeña a cada posible palabra que no aparezca en el corpus.
2. Agregar un token especial <OOV> en el vocabulario durante el entrenamiento. El token <OOV> actúa como un sustituto para cualquier palabra que no esté presente en el vocabulario del modelo. La probabilidad del token se calcula de manera similar a la de cualquier otra palabra.
3. Back-off y modelos interpolados:
   - Los modelos de back-off ajustan la probabilidad a un N-gram más corto (por ejemplo, de un modelo bigrama a un unigram) cuando el N-gram no está en el corpus.
   - Los modelos interpolados combinan N-gramas de diferentes longitudes, ponderando las probabilidades de N-gramas de orden inferior para manejar secuencias que no aparecen en el corpus.
4. Asignación basada en las frecuencias globales: Se asigna una probabilidad a las palabras OOV basada en sus frecuencias observadas en otros corpus o datos externos.