# TF-IDF & Laplace Smoothing

1. Investigar la estrategia de vectorización TF-IDF. ¿Cómo se calcula? ¿En qué situaciones es más efectivo usarlo para tareas de clasificación de texto? ¿Con qué bibliotecas se implementa?

TF-IDF mide la relevancia de una palabra en un documento respecto al mismo contexto de ese documento. se calcula multiplicando la frecuencia del término en un doc. por la frecuencia inversa del documento. Eso disminuye el peso de palabras comunes en todos los documentos. se usa en tareas de clasificación de texto cuando se quiere resaltar palabras importantes sin que se opaquen por palabras frecuentes (él o y). Hablando de Python, existen librerías que podemos usar como scikit-learn (TfidVectorizer) y NLTK.

Ejemplo:

| | Frecuencia de la palabra "gato" | Frec. inversa |
|---|---|---|
| · D1: el gato come pescado | · D1: 1 | |
| · D2: el perro come carne | · D2: 0 | IDF("gato") = ln(3/2) = 0.405 |
| · D3: el perro y el gato juegan | · D3: 1 | |

---

TF-IDF

- D1: $1 \times 0.405 = 0.405$
- D2: $0 \times 0.405 = 0$
- D3: $1 \times 0.405 = 0.405$

2. ¿Qué problema de los N-gram resuelve el "laplace smoothing"? ¿Cómo trabaja? ¿Qué pasa con un modelo de NLP cuando se emplea esa técnica?

los problemas que se resuelven son cuando hay términos con P=0, o sea cuando un N-gram no está en los datos. Lo que hace esta técnica es agregar un 1 a los conteos de los N-grams para evitar probabilidades de cero. En un modelo NLP esto sirve para garantizar que las combinaciones no antes vistas tengan alguna Probabilidad asignada, lo que permite que el modelo se desempeñe mejor, sobretodo en casos no comunes. La fórmula matemática es la siguiente:

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}, w_n) + 1}{C(w_{n-1}) + V}$$ ; donde:

- $C(w_{n-1}, w_n)$ : conteo de N-grams
- $C(w_{n-1})$ : # de veces que la 1ra palabra del biagrama (dos palabras) aparece.
- V : tamaño del vocab. total

3. ¿Qué pasa cuando una palabra en el test set no se encuentra en el vocabulario del modelo de los N-gram? ¿Cómo se puede modelar la prob. de palabras out-of-vocabulary (OOV)?

Cuando una palabra del test set no está en el vocab. de los N-grams su prob. de aparecer sería 0. Para manejar OOV se usan técnicas como probabilidad de respaldo o suavizados (como el que se mencionó anteriormente). Esto permitiría asignar una probabilidad muy bajita pero no nula a las palabras OOV, y así mejorar el modelo. Otra opción sería usar un token especial en las palabras desconocidas.