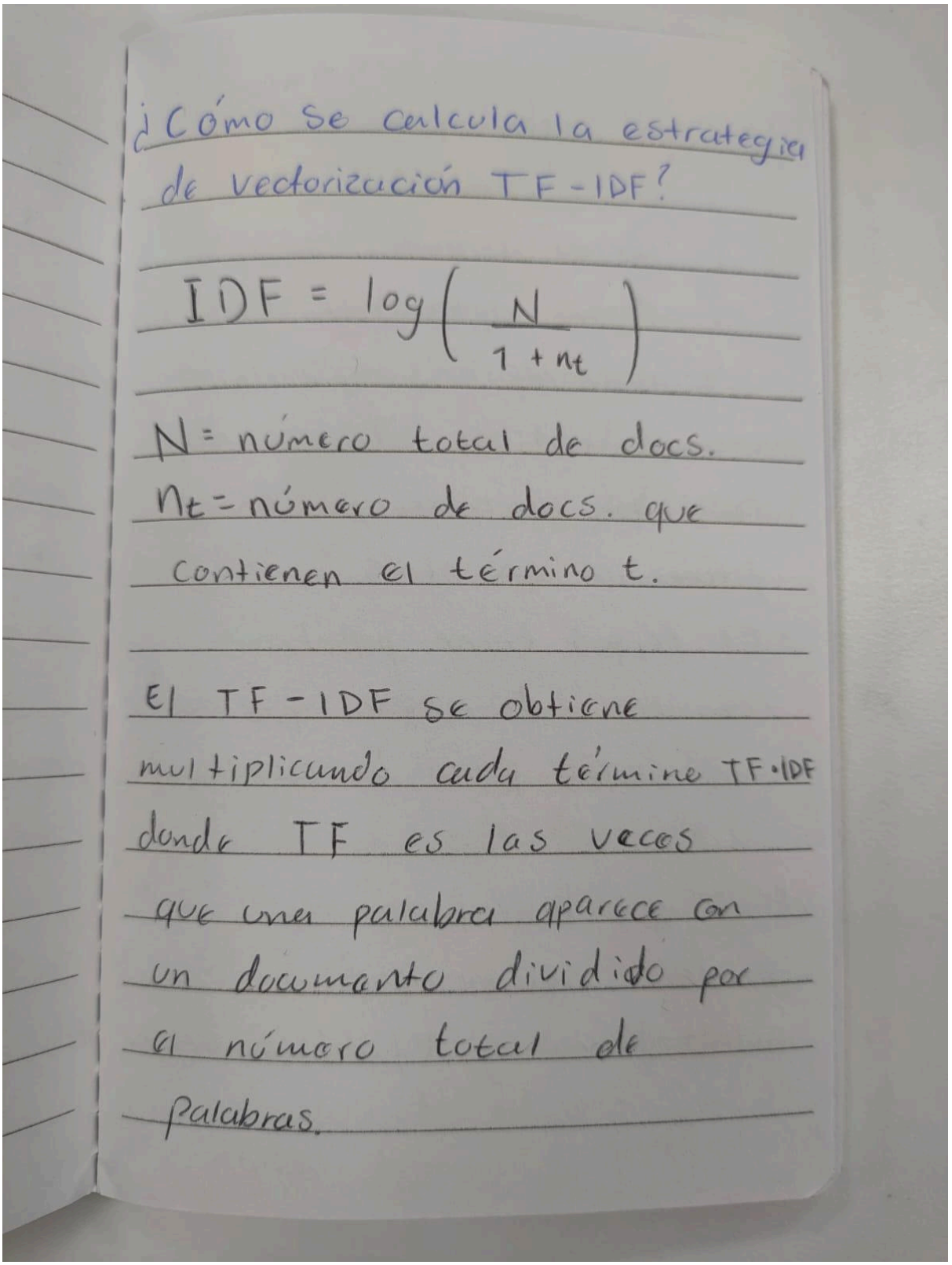
Tarea 2 - Investigación de TF-IDF y Laplace Smoothing



¿Cómo se calcula la estrategia de vectorización TF-IDF?

$$IDF = \log\left(\frac{N}{1 + n_t}\right)$$

N = número total de docs.
n_t = número de docs. que contienen el término t.

El TF-IDF se obtiene multiplicando cada término TF·IDF donde TF es las veces que una palabra aparece en un documento dividido por el número total de palabras.

¿En qué situaciones es más efectivo usar TF-IDF?

Es mejor cuando:

- Se desea optimizar términos y priorizar aquellos que son representativos de un doc. pero que no aparezcan con frecuencia en otros.
- El corpus tiene palabras comunes o de relleno que no son útiles para la clasificación.

¿con que bibiotecas se puede implementar?

scikit-learn y TfidfVectorizer.

¿Qué problemas dos los N-gram resuelve el "Laplace smoothing"?

Cuando algunas combinaciones de palabras no aparecen en el conjunto de entrenamiento resultado en una probabilidad de 0.

¿Cómo trabaja?

Añade un valor constante a todas los contadores de las N-gramas, lo que garantiza que incluso las que no aparecen en el conjunto de entrenamiento tengan un p no nula. $P(w_i|w_{i-1}) = \frac{count(w_{i-1}, w_i)+1}{count(w_{i-1})+V}$

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

- Evita que un modelo asigne una prob. de 0 a N-gramas no vistas.
- Puede diluir las prob. de N-gramas observadas, lo que afecta la precisión.

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

El modelo no puede asignarle una proba. Lo que puede llevar a asignar una proba. de 0.

¿Cómo se puede modelar la prob de palabras OOV?
Con laplace smoothing, tolaan especial OOV o tecnicas como BPE o Wordpiece.