## **1. Traditional Information Retrieval Approach** Don Krasniqi 121410189

### **Method Overview**

For the traditional Information Retrieval (IR) model, we implemented a retrieval pipeline based on **BM25**, a classic bag-of-words ranking function that scores documents based on the frequency and rarity of query terms, while accounting for term saturation and document length normalization. BM25 remains a strong baseline in many IR tasks.

We used the rank\_bm25 library, a pure Python implementation of the BM25Okapi algorithm. BM25 ranks documents higher if they contain more of the query terms (especially rare ones) and penalizes very long documents.

Our implementation performs the following steps:

**1. Document Preprocessing:**

* Concatenated the document **title** and **abstract** from the provided CORD-19 collection.
* Tokenized text using NLTK's word\_tokenize.
* Built a corpus of tokenized documents in memory.

**2. Index Building:**

* Constructed a BM25 index using the preprocessed corpus.

**3. Claim Processing and Retrieval:**

* Each tweet (claim) is tokenized similarly.
* The BM25 index is queried using the processed claim.
* The top-k relevant documents (default: **k=5**) are returned based on their BM25 scores.

The full implementation is located in our Jupyter notebook under the Task 1 section and supports integration with the full claim-source retrieval pipeline.

### **Why It Qualifies as a Traditional IR Approach**

The BM25 model is a canonical example of traditional information retrieval because:

* It does **not** rely on neural networks or learned representations.
* It uses **term frequency (TF)** and **inverse document frequency (IDF)** statistics computed directly from the corpus.
* The retrieval process is based entirely on **lexical overlap** between the query and the documents — no embeddings or semantic models are used.
* The approach is **unsupervised** and parameter-free, apart from optional hyperparameters like k1 and b.

This makes it a textbook example of a traditional IR system in the context of this project.

### **Evaluation Setup & Performance**

Since we missed the official CLEF deadline and could not submit results to Codalab, we implemented our **own local evaluation setup**:

* We **merged the official train and dev splits** from the dataset.
* To avoid data leakage and overfitting, we removed any **duplicate queries**.
* We performed a **random 80/20 split** to create new **train** and **test** sets.
* The BM25 model was evaluated **only on the new test set**.
* The main metric used for evaluation was **MRR@5**, consistent with the CLEF task guidelines.

### **Results**

On our local test split, the BM25-based system achieved an **MRR@5 of 0.5460**, demonstrating solid baseline performance. This provides a strong foundation for comparison against the team’s neural IR approaches, which aim to improve over this traditional method.

## **2. Neural IR Model**

[insert text here]

## **3. Neural Re-Ranker**

[insert text here]