NLP and Deep Learning
MAT3399

Lecture 7:
Document Searching

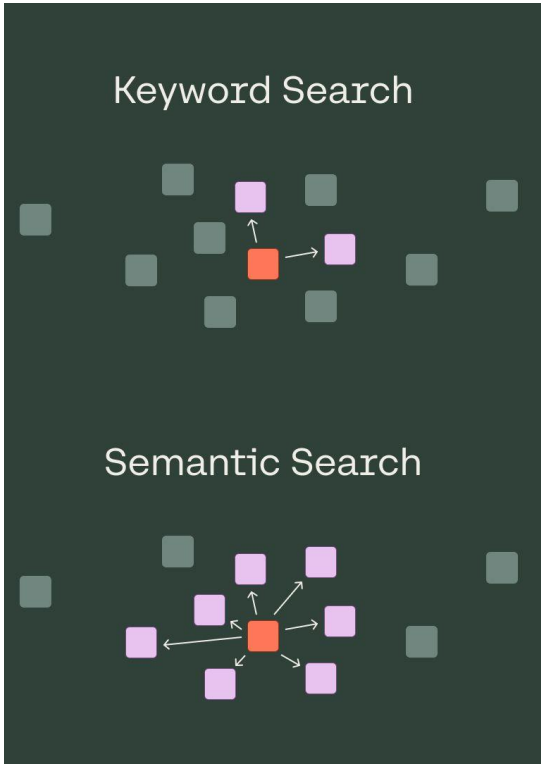
Tuan Anh Nguyen @ Aimesoft
ted.nguyen95@gmail.com

Document Search

Two methods:

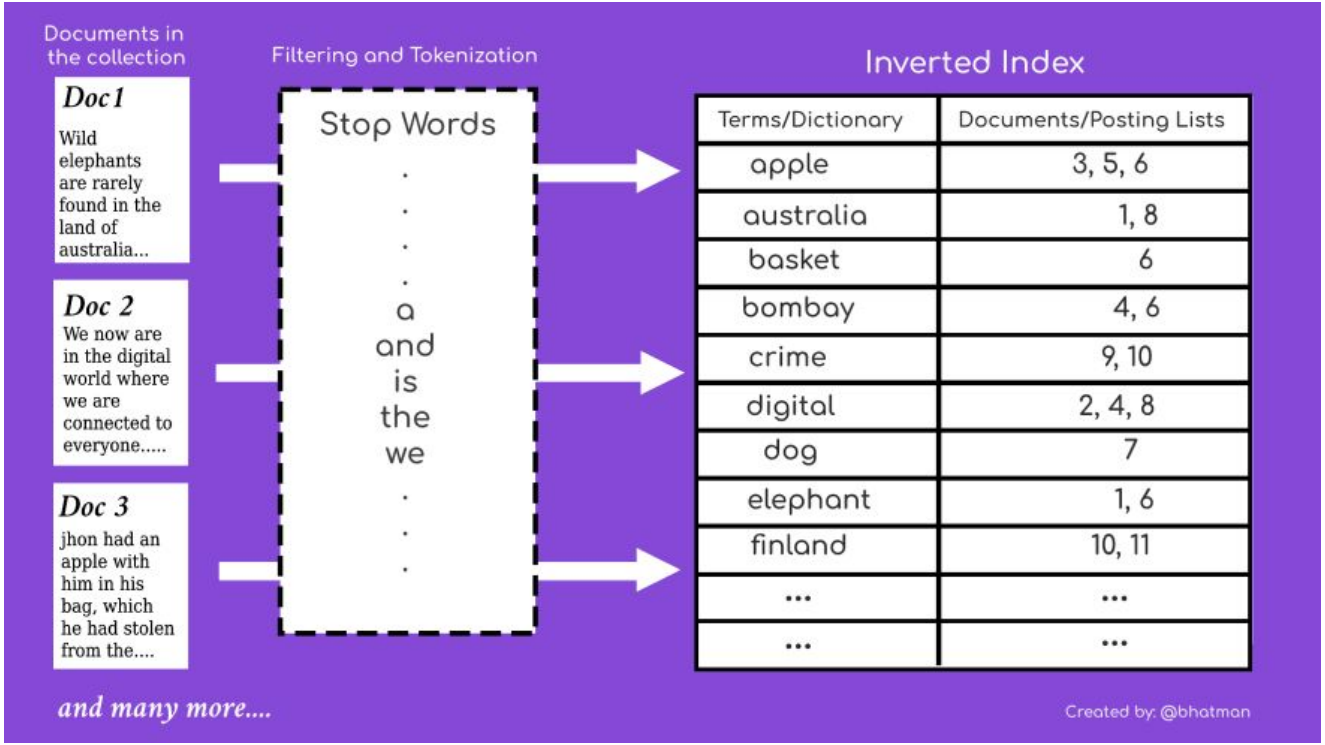
- **Keyword Search:** When a user performs a search in a keyword system, the search engine takes the words from their query and looks for an exact match in the database

- **Semantic Search:** Semantic search uses natural language processing and machine learning algorithms to analyze the context and relationships between words and concepts in a query, and to identify the most relevant results based on their semantic meaning



Simple Keyword Search using Inverted Index

An inverted index is an index data structure storing a mapping from content, such as words or numbers, to its locations in a document or a set of documents



Reminder: TF-IDF

Term Frequency (TF): Frequency of a **term t** in a **document d**

Inverse Document Frequency (IDF): Inversely proportional to the number of documents that contain the **term t**

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

raw count of a term t in a document d

total number of terms in document d

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

total number of documents in the corpus

number of documents where the term t appears (plus one to both the nominator and denominator to prevent division by zero)

Can TF-IDF helps in document searching?

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

Using TF-IDF to rank documents

Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document.

Query	**Documents**	**Score**
bún_chả hà_nội	thủ_đô của việt_nam là hà_nội	tf(hà_nội) * idf(hà_nội) + tf(bún_chả) * idf(bún_chả) = 1/5 * 4/3 + 0 * 4/2 = 0.267
	bún_chả là một món_ăn đặc_trưng ở hà_nội	tf(hà_nội) * idf(hà_nội) + tf(bún_chả) * idf(bún_chả) = 1/7 * 4/3 + 1/7 * 4/2 = **0.476**
	đà_nẵng là một điểm_đến du_lịch nổi_tiếng	tf(hà_nội) * idf(hà_nội) + tf(bún_chả) * idf(bún_chả) = 0 * 4/3 + 0 * 4/2 = 0

Better way to rank documents: BM25

Given a query Q, containing keywords q1, q2,...qn, the BM25 score of a document *D* is:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

f(qi, D): the number of times that qi occurs in the document D
|D|: the length of the document D
avgdl: the average document length in the text collection.
k1, b: free parameters. Usually, k1 = 1.2 and b = 0.75

Better way to rank documents: BM25 (2)

$$\mathrm{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

N: the total number of documents in the collection, and
n(qi): the number of documents containing qi

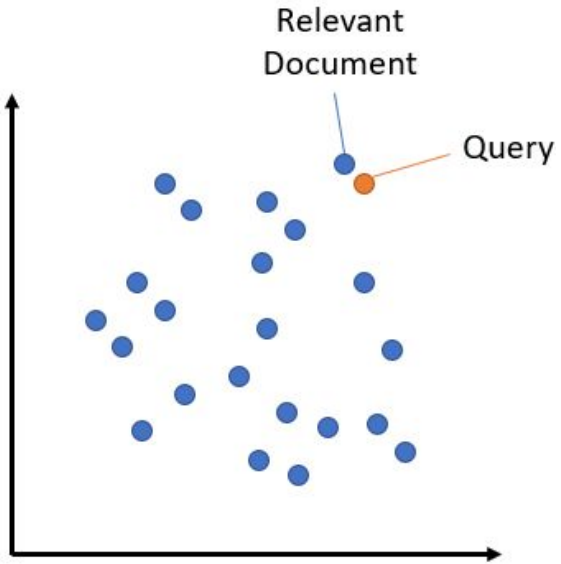
BM25 challenges

1. According to the formula, we need to calculate the score for every document to rank them. Can we speed up this process somehow?

2. What are the disadvantages of this method?

Semantic Search

Semantic search, also known as vector search, is a type of search method that goes beyond traditional keyword-based search and attempts to understand the intent and meaning behind the user's query.



General idea: Convert both query and documents to vectors then calculate similarity score

How to convert query and documents to vectors?

We can use:

- Average word2vec
- Sentence encodings from RNNs

Modern methods:

- Sentence encodings from transformers

Keyword Search vs Semantic Search

Keyword Search	Semantic Search
Cannot handle synonyms, exact matching	Can handle synonyms, no exact matching
Fast	Slow
Allow boolean search (show me this OR these AND those but NOT that)	No boolean search
Cannot handle queries in natural language	Can handle queries in natural language. Understand relationships between words

Depends on different use cases, we use different methods to search

Coding Exercise

Implement BM25 for document searching. You can use this library or any other library you want. Use the Vietnamese wikipedia data in previous lectures.

Advanced Exercise: Speed up BM25 by combining BM25 with inverted indexing