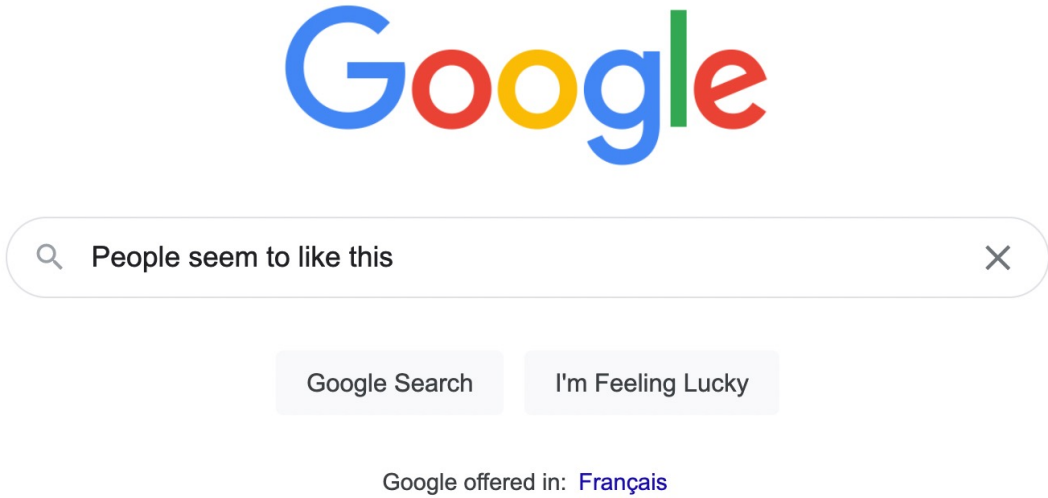
Information Retrieval

A Simple Search Project

Daniel Zhou



My Story

About Me:

- New Grad 🎓 Full Stack Dev. 💻 CMPUT 361 📗

Why This Project:

- IR is neat 👍
- Search is useful 😀
- Search can be complex ⚛️

Takeaways:

- Overview of Two IR methods ✅
- Thoughts On Side Projects ✅

My Project

What is it about:

- A simple search service on movie summaries.
- Hosts a toy movie summaries collection from ~1950 – 2010, ~1700 movies!.
- Exposes two search complementary search interfaces.

Demo

Demo

Demo

Demo

Demo

Demo

Demo

Demo

Demo

… Or Rather Text Retrieval

”Finding material that satisfies an informational need”

Why Use It:

- Direct ✅
- Tolerant 😀
- Easy To Use 👍

Why Implement It:

- ☝️ People Like It
- 🌳 Simple & Cheap (ish)

Two Information Retrieval Methods

Boolean Retrieval

- Queries as Boolean expressions
- Find documents that satisfy them …

Vector Space Retrieval

- Treat queries & documents as vectors in a "keyword" space
- Score & rank documents by similarity in this space

Boolean Retrieval

Unpack the query into a search expression tree, recursively compute the expression.

Parse operators as set operations.

Parse Symbols as getter calls to an index.

Index: Keyword → [Doc ID 1, Doc Id 2, ...]

```
>>> import boolean
>>> algebra = boolean.BooleanAlgebra()
>>> algebra.parse('x & y')
AND(Symbol('x'), Symbol('y'))

>>> parse('(apple or banana and (orange or pineapple and (lemon or cherry)))')
OR(
  Symbol('apple'),
  AND(Symbol('banana'),
      OR(Symbol('orange'),
         AND(Symbol('pineapple'),
             OR(Symbol('lemon'), Symbol('cherry'))))))
```

Vector Space Retrieval

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Documents exist across various keywords (dimensions).

Queries are just short documents.

Similarity Scores:

Cosine Similarity (Angle Similarity)

Term (Dimension) Scores:

TF*IDF=TERM FREQUENCY * INVERSE DOCUMENT FREQUENCY

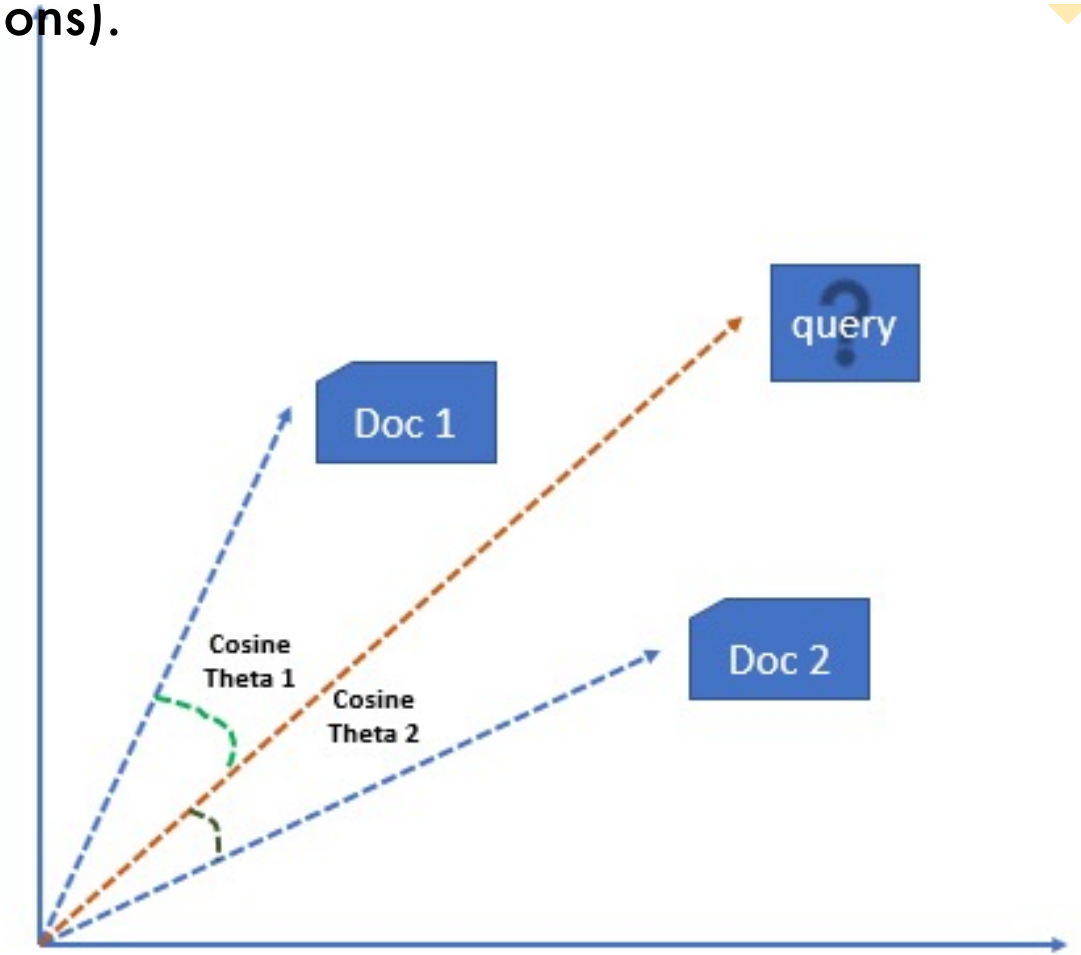
TERM FREQUENCY=

THE AMOUNT OF TIMES A TERM APPEARS IN A DOCUMENT

X

INVERSE DOCUMENT FREQUENCY=

A MEASURE OF WHETHER A TERM IS RARE OR COMMON IN A COLLECTION OF DOCUMENTS.



Considerations I've Skipped

Index Creation & Text Parsing

Optimizing Boolean Search

Vector Space Term Score Weighing & Normalization

Phrase Queries

Query Expansion

Implementation

NLTK – NLP / Text Parsing / Indexation

Boolean.Py - / Boolean Expressions

Flask – Web framework

Heroku – Hosting service

Tech Takeaways

Search is a useful and interesting feature

Boolean search systems - good for power users

Vector space search systems – good for generic search

Python specifics:

- Boolean.py
- NLTK

Soft Takeaways

CMPUT 361 Is Pretty Neat

Booking an hour to work means you’re more likely to do it.

GitHub Projects is a lightweight and great way of staying organized.