Boolean Model

for an Information Retrieval System

Davide Basso

What is

- **Document representation**
  - As **set of terms**
- **Query formulation**
  - As **Boolean Formula**
- **Exact matching**
  - **Retrieved documents** are just the **relevant** ones
- **No ranking** is provided

What is implemented

- System is able to **answer**:
  - **Boolean Queries** (using **AND, OR, NOT** operators)
  - **Phrase Queries** (also **"Term1 /k Term2"** queries)
  - **Wildcard Queries** (**Leading, Trailing, General** and **Multiple** ones)
- Moreover, **Spelling Correction** is given as a feature
  - **Edit Distance** is used

The dataset

- It is made of the **posts** present in famous **subreddit** r/nba.
- Retrieved by using **Reddit's built in API .json** functionality to scrape post data.
- It counts ***2315*** posts in total.

Implementation steps

01

BASIC COMPONENTS

Posting, Posting List, Term definition

02

PRE-PROCESSING

Normalization, Tokenization

03

INDEX

Inverted, Positional and Permuterm Index

04

BOOLEAN IR SYSTEM & features

How to answer queries, Spelling Correction

05

QUERY DEFINITIONS

Execution of a query

06

TESTING

Test overall system correctness

Basic Components

01

- **Posting**, the atomic element of a Posting List:
  - contains **docID** and **Positions** list
- **Posting List**: list of Postings related to a specific Term.
- **Term**, an entry of the **Dictionary**:
  - contains **word** itself and associated **Posting List**

Pre-Processing

02

- **Normalization**:
  - **Removes punctuation**, **accents** and **capitalization**
- **Tokenization**:
  - Subdivides **text** into **tokens**

Index Object

03

Combines **multiple indices**

- **Inverted Index**: used to answer trivial Boolean queries
- **Positional Index**: used to answer Phrase queries
- **Permuterm Index**: used to answer Wildcard queries

*Deatils of the implementation will be shown inside the **fromCorpus** function.*

Saving and **loading** of the Index to/from **disk** through **Pickle**.

Boolean IR System

04

Made of a **Cropus** and an **Index**.

Answers:

- **AND, OR, NOT** queries
  - **Intersection**, **union** and **difference** of **Posting Lists**
- **Phrase queries** and queries that specify **maximum** number of **words between two terms**.
  - Performing a **Positional Search**
- **Trailing, Leading, General** and **Multiple Wildcard queries**.

Boolean IR System cont. *Spelling Correction*

04

- **Edit** (or **Levenshtein**) **Distance** measures the "distance" between two terms.
- **Counts** the **minimum number** of
  - **Insertion**
  - **Deletion**
  - **Update**

  of a character that is **needed** to **transform** one **word into another** one.

Query Definitions

05

- **Each function** defines the **approach** for executing a **specific query**:
  - E.g. for ***queryWithParentheses*** it has been implemented a way to answer a Boolean queries with multiple parentheses, even nested ones.

Test Procedure and Results

06

- **Procedure**:
  - Retrieve **answer using Boolean IR System defined function**
  - Retrieve **answer "by hand"**
  - **Compare** the results
- **Results**:
  - **All the answers matched!**