Information retrieval

Project

Fall’16

Dr. Nada J. Naji

Aly

Akhtar

Lalit

Pathak

Tushar

Gupta

Contents

[INTRODUCTION 3](#_Toc468892691)

[LITERATURE AND RESOURCES 4](#_Toc468892692)

[IMPLEMENTATION AND DISCUSSION 5](#_Toc468892693)

[Phase 1: Introduction and Initial Setup 5](#_Toc468892694)

[On Your Marks! 5](#_Toc468892695)

[Get Set Go! 5](#_Toc468892696)

[Phase 1: Task-1 6](#_Toc468892697)

[Vector Space-Cosine Similarity Model 6](#_Toc468892698)

# INTRODUCTION

# LITERATURE AND RESOURCES

# IMPLEMENTATION AND DISCUSSION

## Phase 1: Introduction and Initial Setup

### On Your Marks!

In the first phase, we perform indexing and retrieval. We use the CACM dataset. This phase is divided into three tasks-

1. Task 1: Build search engines using four different retrieval models.
2. Task 2: Perform query expansion using Rocchio’s Algorithm.
3. Task 3: Perform stopping and stemming.

### Get Set Go!

The following tasks were done in order to prepare the system, and get relevant initial files-

#### Extract queries to a simple readable format

* “extract\_queries.py” is a python program which takes the “cacm.queries” file as input and returns a file “queries.txt” which contains the queries given in the file, sans the decorations, so that it is easy to feed them to the retrieval system.
* “queries.txt” is the file that all our systems use for processing queries.

#### Tokenize the Corpus

* “task1\_tokenizer.py” is a python program which takes the CACM Corpus in html format as input and processes it to generate tokenized corpus in .txt format.
* We have used the NLTK library to perform tokenization.
* NLTK library leaves behind some punctuations, which have been taken care of, manually.

#### Index the Corpus

* “task1\_indexer.py” is a python program which takes the tokenized corpus as input and creates the inverted index for the corpus.
* It saves the index to the file “Inverted\_Index.txt” in alphabetically sorted order.

## Phase 1: Task-1

In this, we performed four base runs. One for each model-

* Vector-Space-Cosine-Similarity Model
* BM25 Model
* TF-IDF Model
* Lucene Model

### Vector Space-Cosine Similarity Model