InformationRetrieval

Advanced Database Systems - Project 1

Team Information

- **Names:** Begum Cicekdag, Maxine Tamas
- **UNIs:** bc2975, mt3634

Submitted Files

- `query_expansion.py` - Main Python script implementing query expansion
- `rocchio.py` - Implementation of the Rocchio Algorithm
- `requirements.txt` - List of required Python libraries
- `README.md` - This documentation file
- `proj1-stop.txt` - List of all of our stop words

API Credentials

- **Google Custom Search API Key:** `AIzaSyDSfAcOzN3SyrUDN2fCe_QNIIx3sOLN5Rk`
- **Google Custom Search Engine ID:** `c66d0519df77f44f1`

Setup Instructions:

1. **Connect to Google Cloud VM:**

```
ssh your-username@your-vm-ip
```

2. **Install Required Software:**

```
sudo apt update
sudo apt install python3-pip
```

3. **Clone the Repository:**

```
git clone https://github.com/maxinetamas/InformationRetrieval.git
cd InformationRetrieval
```

4. **Install Dependencies:**

```
pip install -r requirements.txt
```

5. **Run the Program:**

To run the program, you need to use the Google Custom Search API key and Google Custom Search Engine ID above, as well as provide the desired precision, and the query you want to run.

The format would be something like:

```
python3 query_expansion.py YOUR_GOOGLE_API_KEY YOUR_GOOGLE_ENGINE_ID PRECISION "your search query"
```

And an example would look like:

```
python3 query_expansion.py AIzaSyDSfAcOzN3SyrUDN2fCe_QNIIx3sOLN5Rk c66d0519df77f44f1 0.8 "machine learning"
```

Internal Design

The project consists of the following main components:

- **Query Expansion Module:** Implements the Rocchio Algorithm for query refinement.
- **Google Custom Search API Handler:** Handles search queries and fetches results using the Google Custom Search API.
- **Relevance Feedback Processor:** Analyzes the search results and selects relevant keywords based on user feedback.
- **Command-Line Interface:** Provides an interface for running queries and obtaining results.

External Libraries Used

- `requests` - For making API calls to Google Custom Search (Note: While `requests` is commonly used for making API calls, this project uses `google-api-python-client` to interact with the Google Custom Search API.)
- `numpy` - For vector operations in the Rocchio Algorithm
- `json` - For handling API responses and configurations
- `argparse` - For parsing command-line arguments
- `google-api-python-client` - Interacting with the Google Custom Search API
- `scikit-learn` - Using `TfidfVectorizer` for calculating TF-IDF values.

Query Modification Method

The core component of this project is the query expansion process using the Rocchio Algorithm, which follows these steps:

1. **Initial Query Execution:**

   The original query is sent to the Google Custom Search API.

2. **Relevance Feedback Collection:**

   The user marks each of the top-10 results as relevant (Y) or non-relevant (N) via the command line.

3. **Vector-Based Query Adjustment:**

   The Rocchio Algorithm modifies the query vector by:

   - Increasing weights of terms from relevant results
   - Decreasing weights of terms from non-relevant results

4. **Term Reordering and Expansion:**

   Our enhanced implementation:

   - Reorders ALL terms (both original and new) based on their TF-IDF weights
   - Places the most discriminative terms first in the query
   - Adds up to 2 new high-weight terms not in the original query

5. **Iterative Refinement:**

   Steps 1-4 are repeated until the target precision is reached or the precision no longer improves.

Query Word Order Determination

Our implementation improves upon basic Rocchio by:

- Computing weights for all terms (original query + potential new terms)
- Sorting terms by their weight in descending order
- Creating a new query that maintains this weight-based ordering

This approach leverages the fact that search engines often give more weight to terms that appear earlier in the query.

Program Run Transcripts

Below is a sample transcript of the program's execution:

```
Parameters:
Client key = AIzaSyDSfAcOzN3SyrUDN2fCe_QNIIx3sOLN5Rk
Engine key = c66d0519df77f44f1
Query = machine learning
Precision = 0.8
Google Search Results:

Result 1
URL: https://example.com/1
Title: Introduction to Machine Learning
Summary: Machine learning is a subfield of artificial intelligence...
Relevant (Y/N)? Y
[results continued...]
FEEDBACK SUMMARY
Query: machine learning
Precision: 0.6
Still below the desired precision of 0.8
Indexing results...
Indexing results...
Augmenting by deep neural
Parameters:
Client key = YOUR_GOOGLE_API_KEY
Engine key = YOUR_GOOGLE_ENGINE_ID
Query = deep neural machine learning
Precision = 0.8
[second iteration continues...]
...
...
...
Desired precision reached.
```