Astrid Alins, Miquel Bisbe, Cinta Carot

Part II: Indexing and Evaluation

Github link: https://github.com/astridalins/IRWA_final_project.git

Tag name: **IRWA-2025-part-2**

PART 1: Indexing

2.1.1 Build inverted index:

After having pre-processed the data, you can then create the inverted index.

HINT - you may use the vocabulary data structure, like the one seen during the Practical Labs:

{
Term_id_1: [document_1, document_2, document_4],
Term_id_2: [document_1, document_3, document_5, document_6],
etc…
}

Important: For this assignment, we will be using **conjunctive queries (AND)**. This means that every returned document must contain all the words from the query in order to be considered a match.

For indexing, we implemented an inverted index using preprocessed tokens from product titles and descriptions. The index maps terms to posting lists, where each list contains document IDs and term positions within those documents. This structure facilitates efficient retrieval by providing direct access to documents containing query terms.

We also maintain a mapping from internal document IDs to original product metadata for result presentation. This indexing approach supports rapid lookup for conjunctive queries.

2.1.2 Propose test queries:

Define five queries that will be used to evaluate your search engine. (Be creative)

HINT - How to choose the queries? The selection of the queries is up to you, but it's suggested to select terms based on the popularity (keywords ranked by term frequencies or by TF-IDF, etc).

The chosen 5 queries are:

- women black dress
- men casual shirt cotton
- solid color track pant
- print t shirt
- blue jeans slim fit

The five test queries were formulated based on insights from the Exploratory Data Analysis (EDA), particularly the word clouds generated from product titles and descriptions. By observing the most frequent and bigger terms in these visualizations, we identified common keywords and phrases that users are likely to search for. The selected queries combine these popular terms to represent realistic scenarios within the dataset.

2.1.3 Ranking of Query Results Using TF-IDF

Rank your results: Implement the TF-IDF algorithm and provide ranking-based results.

After building the inverted index (Section 2.1.1) and defining test queries (Section 2.1.2), we implemented a ranking method for query results using **TF-IDF**, ensuring compatibility with evaluation metrics in Part 2.

Indexing Recap:

- The inverted index maps each term to a **posting list** of document IDs and positions.
- Document metadata is stored separately to enable result presentation.
- This structure allows **efficient conjunctive queries** (AND), where only documents containing all query terms are retrieved.

TF-IDF Ranking Implementation:

1. **Query tokenization:** The query string is split into individual terms.
2. **Document frequency (DF):** Count of documents containing each term.
3. **Inverse document frequency (IDF):** Computed as

$$\mathrm{IDF}(t) = \log \frac{N}{DF_t}$$

   where NN is the total number of documents. Terms appearing in fewer documents get higher weight.

4. **Term frequency (TF):** Number of times the term appears in a document.
5. **TF-IDF score per document:**
   The sum of TF-IDF scores across all query terms gives a final relevance score.

$$\text{Score}(d) = \sum_{t \in \text{query}} TF(t,d) \cdot IDF(t)$$

6. **Ranking:** Documents are sorted in descending order by their score.

Evaluation Integration:

- The ranking output is compatible with the evaluation metrics in Part 2: **P@K, R@K, MAP, NDCG, MRR**, etc.
- For each query, we can compute these metrics using the ranked list and binary relevance labels (1 for relevant, 0 for non-relevant) defined in `validation_labels.csv`.

Query Examples:

- women black dress
- men casual shirt cotton
- solid color track pant
- print t shirt
- blue jeans slim fit

These queries were selected based on **term frequency and keyword prominence** from the dataset. They simulate realistic search behavior and provide a basis for measuring retrieval effectiveness.

Results Interpretation:

- For most queries, documents containing **all relevant keywords** receive the highest scores. For example, in the query `'women black dress'`, the top documents correctly prioritize products with both `'women'` and `'black'` in the title.
- Some documents with **partial keyword matches** also appear in the top 5 but have lower TF-IDF scores, highlighting the discriminative power of the TF-IDF weighting.
- Score distribution typically decreases sharply after the top 2–3 documents, indicating that the most relevant documents dominate the ranking.
- This qualitative analysis provides an initial check on the system's performance **before formal evaluation** with Part 2 metrics.

Conclusion:

- Using TF-IDF ranking ensures that documents containing the **most important and discriminative query terms**appear first.
- The ranked output is directly usable for **evaluation metrics**, allowing assessment of retrieval quality and identification of potential system improvements.
- Preliminary observations from the score distribution and top-ranked documents can guide **future refinements**, such as query expansion, improved tokenization, or weighting adjustments, to enhance retrieval accuracy.

PART 2: Evaluation

2.2.1 Evaluation Metrics Implementation

To assess the effectiveness of the retrieval system and the quality of the TF-IDF–based rankings, we implemented a set of standard **Information Retrieval (IR)** evaluation metrics. These metrics allow us to quantitatively measure how well the system retrieves relevant documents for each query.

The implemented metrics are the following:

i. Precision@K (P@K)

Precision@K measures the proportion of retrieved documents within the top *K* results that are relevant. It provides insight into how precise the system is when showing its highest-ranked results.

$$P@K = \frac{\text{Number of relevant documents in top K}}{K}$$

ii. Recall@K (R@K)

Recall@K evaluates the system's ability to retrieve all relevant documents. It measures how many of the total relevant documents are found within the top *K* results.

$$R@K = \frac{\text{Number of relevant documents in top K}}{\text{Total number of relevant documents}}$$

iii. Average Precision@K (AP@K)

Average Precision considers the order of retrieved documents by averaging the precision at every rank position where a relevant document is found. It provides a single-number summary of ranking quality for a query.

$$AP@K = \frac{1}{R} \sum_{i=1}^{K} P@i \text{ for relevant documents}$$

iv. F1-Score@K

The F1-Score is the harmonic mean of Precision and Recall, combining both measures into a single value to balance precision and coverage.

$$F1@K = \frac{2 \times P@K \times R@K}{P@K + R@K}$$

v. Mean Average Precision (MAP)

MAP provides a global performance measure across multiple queries. It is the mean of the Average Precision values for all queries, thus rewarding systems that rank relevant documents consistently high across queries.

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} AP_q$$

vi. Mean Reciprocal Rank (MRR)

MRR evaluates how early the first relevant document appears in the ranking. For each query, the reciprocal rank is the inverse of the position of the first relevant document.

$$MAP = \frac{1}{Q} \sum_{q=1}^{Q} AP_q$$

vii. Normalized Discounted Cumulative Gain (NDCG)

NDCG accounts for graded relevance and ranking position, giving higher scores to relevant documents appearing near the top of the ranking. It normalizes the Discounted Cumulative Gain (DCG) by the ideal ranking (IDCG).

$$NDCG@K = \frac{DCG@K}{IDCG@K} \qquad DCG@K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

Conclusion

These evaluation metrics provide a comprehensive framework for assessing the retrieval performance of the implemented search engine.

- **Precision@K** and **Recall@K** measure relevance quality and coverage.
- **MAP** and **MRR** evaluate overall ranking effectiveness.

- **NDCG** adds a position-sensitive measure that rewards well-ordered results.

Together, they allow an in-depth quantitative analysis of the system's strengths and weaknesses in document retrieval.

2.2.2 Apply the evaluation metrics you have implemented to the search results and relevance judgments provided in validation_labels.csv for the predefined queries.

When reporting evaluation results, provide **only numeric values**, rounded to **three decimal places**. Do not include textual explanations or additional statistics in this section.

a. Query 1: `women full sleeve sweatshirt cotton`
b. Query 2: `men slim jeans blue`

Our results:

```
Query: women full sleeve sweatshirt cotton
  Precision@10: 0.100
  Recall@10: 0.077
  Average Precision@10: 0.015
  F1-Score@10: 0.087
Query: men slim jeans blue
  Precision@10: 0.000
  Recall@10: 0.000
  Average Precision@10: 0.000
  F1-Score@10: 0.000

Mean Average Precision (MAP)@10: 0.008
Mean Reciprocal Rank (MRR): 0.102
Mean Normalized Discounted Cumulative Gain (NDCG)@10: 0.043
```

The results show very low performance across all metrics, with the best query achieving a Precision@10 of 0.100 and a Recall@10 of only 0.077. The second query obtained zero scores for all measures, indicating that no relevant documents were retrieved in its top results. The overall MAP (0.008), MRR (0.102), and NDCG (0.043) confirm that the system struggles to rank relevant documents effectively.

These poor scores suggest that the current TF-IDF–based retrieval method **fails to capture the semantic meaning of the queries**.

2.2.3 You will act as expert judges by establishing the ground truth for each document and query.

a. For the test queries you defined in Part 1, Step 2 during indexing, assign a binary relevance label to each document: 1 if the document is relevant to the query, or 0 if it is not.

The code that we did for this section makes the user manually assign relevance labels to the top 10 retrieved documents for every query. For every test query, the system retrieves the top 10 ranked documents using the TF-IDF model and displays their title, score, and a short snippet of the description. In our case, we implemented the line:
print(f"Description: {description[:120]}...") , which shows us the first 120 characters of the document. Then, the user indicates whether each document is relevant (1) or not (0) to the query by introducing '1' or '0'. We managed this information by creating a dictionary ('manual_relevance').

b. Comment on each of the evaluation metrics, stating how they differ, and which information gives each of them. Analyze your results.

Our results:

```
GLOBAL REVIEW:
  • women black dress: 0 relevant / 10 documents
  • men casual shirt cotton: 7 relevant / 10 documents
  • solid color track pant: 10 relevant / 10 documents
  • print t shirt: 6 relevant / 10 documents
  • blue jeans slim fit: 10 relevant / 10 documents
```

```
--- Evaluation Results (Manual Labels) ---
```

	Query	Precision@10	Recall@10	F1@10	AP@10	NDCG@10
0	women black dress	0.0	0.0	0.0	0.000	0.000
1	men casual shirt cotton	0.7	1.0	0.824	0.844	0.945
2	solid color track pant	1.0	1.0	1.0	1.000	1.000
3	print t shirt	0.6	1.0	0.75	0.505	0.660
4	blue jeans slim fit	1.0	1.0	1.0	1.000	1.000
5	GLOBAL AVERAGE	-	-	-	0.670	0.667

```
MAP@10: 0.670
MRR: 0.667
```

- **Precision@K:** Precision measures how many of the retrieved documents in the top 10 are actually relevant.

In the results, three queries (“solid color track pant”, “blue jeans slim fit”, and “men casual shirt cotton”) showed very high precision, meaning most or all retrieved documents were relevant. The query “print t shirt” had a moderate precision (0.6), showing that some non-relevant documents appeared in the top results. “women black dress” had a precision of 0, indicating that no retrieved document was relevant. This highlights inconsistencies in keyword matching for certain product categories.

- **Recall@K:** Recall measures how many of all the relevant documents in the dataset were retrieved in the top 10.

Except for “women black dress”, all queries achieved perfect recall (1.0). This means that, for most cases, the system was able to find all relevant documents within the first ten results. The zero recall for “women black dress” indicates that the query failed to match any relevant documents, likely due to vocabulary mismatch or sparse data.

- **F1@K:** The F1-score combines precision and recall into a single value, balancing both aspects.

Queries with high precision and recall, such as “solid color track pant” and “blue jeans slim fit”, obtained perfect F1 = 1.0. The “men casual shirt cotton” query also performed well (F1 = 0.824), while “print t shirt” had a lower score (0.75), showing that its ranking could be improved. The “women black dress” query again scored 0, confirming poor retrieval for that case.

- **Average Precision@K**: Average Precision evaluates both the precision at different ranks and how early relevant documents appear in the list.

The results show strong ranking performance for most queries (AP above 0.8), with perfect ranking for “solid color track pant” and “blue jeans slim fit”. “print t shirt” achieved lower AP (0.505), meaning some relevant documents were ranked further down the list. “women black dress” had AP = 0.0, consistent with the absence of relevant results.

- **NDCG@10:** NDCG takes into account the ranking position of relevant documents, giving higher importance to those retrieved earlier.

The high NDCG scores (close to 1.0) for most queries indicate that relevant documents appeared near the top of the ranking. The “print t shirt” query had a moderate NDCG (0.66), showing that relevant results were mixed with less relevant ones. Again, “women black dress” scored 0, as no relevant documents were retrieved.

- **Global Metrics:** The Mean Average Precision (0.67) and Mean Reciprocal Rank (0.67) suggest that the system performs well overall but not perfectly. Most queries have relevant documents ranked high, but one or two poorly performing queries lower the overall score. The issue likely stems from vocabulary limitations and TF-IDF’s inability to capture semantic similarity between words.

c. Analyze the current search system and identify its main problems or limitations. For each issue you find, propose possible ways to resolve it. Consider aspects such as retrieval accuracy, ranking quality, handling of different field types, query formulation, and indexing strategies.

Our main issues:

- Limited retrieval accuracy

The current search system relies on a basic TF-IDF approach with exact keyword matching. This means it fails to recognize variations or synonyms of the same word. Inour case, we had many problems with terms like “t shirt”, “t-shirt”, and “tshirt”, which we believe are treated as completely different tokens, leading to missed relevant results.

→ Possible improvement: Applying a Word2Vec approach could help capture meaning beyond surface forms.

- Rigid query formulation

The current system only supports conjunctive (AND) queries, meaning that all query terms must appear in a document to be considered a match. This approach is too strict and can reduce recall.

→ Possible improvement: Introduce disjunctive (OR) or phrase queries, query expansion using relevance feedback, or fuzzy matching to better handle partial or misspelled inputs.