Information Retrieval and Web Analytics, 2025

Universitat Pompeu Fabra Barcelona

Project Part 2: Indexing and Evaluation

Team: G_004
Alex De La Haya Gutiérrez, 268169
Marc Guiu Armengol, 268920
Nil Tomàs Plans, 268384

Github URL: https://github.com/niiltomas/irwa-search-engine-G_004.git

Tag: IRWA-2025-part-2

Index of Contents

1. **Indexing**
   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.
   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).
   1.3. Rank your results: Implement the TF-IDF algorithm and provide ranking-based results.
2. **Evaluation**
   2.1. Implement the following evaluation metrics to assess the effectiveness of your retrieval solutions. These metrics will help you measure how well your system retrieves relevant documents for each query:
   i. Precision@K (P@K)

ii. Recall@K (R@K)

iii. Average Precision@K (P@K)

iv. F1-Score@K

v. Mean Average Precision (MAP)

vi. Mean Reciprocal Rank (MRR)

vii. Normalized Discounted Cumulative Gain (NDCG)

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.

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

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

Universitat Pompeu Fabra Barcelona

Report

1.1 Build inverted index:

After pre-processing the data we have created the inverted index. This is the structure we have followed:

Example term: 'solid' → postings: [[0, array('I', [0])], [1, array('I', [0])], [2, array('I', [0])]]

Where we indicate document_ID and position inside of document.

1.2 Propose test queries:

The queries that we have proposed are the following:

1. women track pant
2. men track pant
3. men pack
4. women formal shirt
5. men slim fit formal shirt

We have chosen queries that fit our information needs.

1.3 Rank your results:

We have implemented the standard TF-IDF algorithm (seen in lectures).

The results are the following:

1. women track pant
    1. Solid Women Multicolor Track Pants | TF-IDF: 2.2847
    2. Solid Women Multicolor Track Pants | TF-IDF: 2.2847
    3. Solid Women Multicolor Track Pants | TF-IDF: 2.2847
    4. Solid Women Multicolor Track Pants | TF-IDF: 2.2847
    5. Solid Women Black Track Pants | TF-IDF: 1.8378
2. men track pant
    1. Solid Men Multicolor Track Pants | TF-IDF: 2.2909
    2. Solid Men Multicolor Track Pants | TF-IDF: 2.2909
    3. Solid Men Blue Track Pants | TF-IDF: 1.8327
    4. Solid Men Green Track Pants | TF-IDF: 1.8327
    5. Solid Men Multicolor Track Pants | TF-IDF: 1.8327
3. men pack
    1. Men Brief (Pack of 12) | TF-IDF: 1.1269
    2. Men Brief (Pack of 2) | TF-IDF: 1.1269
    3. Men Brief (Pack of 5) | TF-IDF: 1.1269
    4. Men Brief (Pack of 6) | TF-IDF: 1.1269
    5. Men Brief (Pack of 4) | TF-IDF: 1.1269

4. women formal shirt
    1. Women Solid Formal Shirt | TF-IDF: 1.8863
    2. Women Regular Fit Checkered Formal Shirt | TF-IDF: 1.2575
    3. Women Regular Fit Solid Formal Shirt | TF-IDF: 1.2575
    4. Women Slim Fit Solid Formal Shirt | TF-IDF: 1.2575
    5. Women Slim Fit Solid Formal Shirt | TF-IDF: 1.2575
5. men slim fit formal shirt
    1. Men Slim Fit Solid Formal Shirt | TF-IDF: 2.1545
    2. Men Slim Fit Printed Formal Shirt | TF-IDF: 2.1545
    3. Men Slim Fit Checkered Formal Shirt | TF-IDF: 2.1545
    4. Men Slim Fit Printed Formal Shirt | TF-IDF: 2.1545
    5. Men Slim Fit Printed Formal Shirt | TF-IDF: 2.1545

2.1. Implement the evaluation metrics

i. Precision@K (P@K)

ii. Recall@K (R@K)

iii. Average Precision@K (P@K)

iv. F1-Score@K

v. Mean Average Precision (MAP)

vi. Mean Reciprocal Rank (MRR)

vii. Normalized Discounted Cumulative Gain (NDCG)

We have implemented them using the algorithms seen in the lecture classes.

Evaluation results:

Query	Precision	Recall	AP	F1	NDCG
1	0.400	1.000	0.700	0.571	1.808
2	0.400	1.000	0.417	0.571	0.571
3	0.000	0.000	0.000	0.000	0.000
4	0.400	1.000	0.750	0.571	1.114
5	0.400	1.000	1.000	0.571	1.501

MAP@5	MRR
0.573	0.667

2.2. Apply evaluation metrics:

We will apply the evaluation metrics to the following queries.

Query 1: women full sleeve sweatshirt cotton

Query 2: men slim jeans blue

Evaluation results:

Query	Precision	Recall	AP	F1	NDCG
1	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.000	0.000	0.000

MAP@5	MRR
0.198	1

2.3. Act as expert judges:

a.

We've created a copy of the dataset to create a new one with a new field which states the relevance of the document to the queries defined previously in the array *test_queries:*

```
test_queries = [
    "women track pant",
    "men track pant",
    "men pack",
    "women formal shirt",
    "men slim fit formal shirt"
]
```

b.

We've used and computed different metrics to evaluate the results of our algorithm for each query. Next, we citate each of them, explain how the different metrics differ between them and the results obtained and meaning.

Precision@K (P@K):

Is defined by the fraction of the first k retrieved documents that are relevant. It focuses on how accurate the top retrieved results are.

Recall@K (R@K)

Is defined as the fraction of all relevant documents that are retrieved within the top k results. It focuses on how complete the retrieval is.

That is, precision cares about the quality, while recall cares about coverage.

Average Precision@K (AP@K)

Is the average of Precision@K values at each point where a relevant document is found. It considers both relevance and ranking order, rewarding systems that show relevant documents earlier

F1-Score@K

Is the harmonic mean of Precision@K and Recall@K. It is important because it emphasizes the importance of small values.

Mean Average Precision (MAP)

Is the mean of the Average Precision values across all queries. It summarizes the overall ranking performance of the system.

Mean Reciprocal Rank (MRR)

Is the average of the reciprocal ranks of the first relevant document for each query. It measures how quickly the system returns the first relevant result.

That is, MRR focuses on how soon we achieve relevance, while MAP focuses on how well all relevant documents are ranked.

Normalized Discounted Cumulative Gain (NDCG)

Measures the quality of the ranking, giving higher weight to relevant documents that appear near the top of the results list. It focuses on ranking quality rather than just relevance. It aims to capture ranking quality and degrees of relevance, not just presence

if we analyze the results of the operations into the different queries we can relate that:

For Q1 (“women track pant”), the relevant items (“Solid Women Multicolor Track Pants” and “Solid Women Black Track Pants”) match the query closely, because most words appear in both. If we look into the results the Recall@5 = 1.000, meaning both relevant products were found, and Precision@5 = 0.400, showing that only 2

out of the top 5 retrieved items were relevant. The higher the values of AP@5 and NDCG@5 (closer to 1), the more relevant items are ranked at the top, indicating a very strong match.

In Q2 (“men track pant”), the relevant items (“Solid Men Blue Track Pants” and “Solid Men Green Track Pants”) also fit the query well. Recall@5 = 1.000 again shows all relevant products were retrieved, but AP@5 and NDCG@5 values indicate that they appeared lower in the ranking compared for example to Q1.

Q3 (“men pack”) represents a different case. Its relevant items (“Superhero Men Round Neck Multicolor T-Shirt (Pack of 2)” and “Superhero Women Round Neck Multicolor T-Shirt (Pack of 2)”) share almost no direct words with the query. As a result, all metrics are 0, showing that none of the relevant items were retrieved.

For Q4 (“women formal shirt”), the relevant items (“Women Solid Formal Shirt” and “Women Slim Fit Solid Formal Shirt”) have strong textual overlap with the query. Recall@5, AP@5, and NDCG@5 show that both relevant products were retrieved and ranked near the top, producing solid results.

Finally, Q5 (“men slim fit formal shirt”) has the closest match between query and relevant items (“Men Slim Fit Solid Formal Shirt” and “Men Slim Fit Printed Formal Shirt”). Every word from the query appears in the titles, which explains the perfect AP@5 = 1.000 and NDCG@5 = 1.000, indicating both items were retrieved and ranked ideally.

Overall, these queries and their relevant items were carefully chosen to represent different scenarios, observe the differences on the metrics results and highlight how relevance and ranking interact across queries.

.c One of its main problems is retrieval accuracy, as we use the AND query model.
We only retrieve documents that contain all query terms. It fails otherwise and also ignores synonyms e.g. trousers - pants. To improve that we could expand query terms using synonyms or use also OR queries.
Ranking quality could also be improved as it is based only on TF-IDF which assumes term independence and ignores word context. We could switch to BM25, which is a

more robust probabilistic model.

All text fields are treated equally. We should give more importance to fields like title or brand. We could give higher weights to those terms.

When you misspell a word our retrieval model will return 0 items. We could implement spelling correction for example.

We like our indexing strategy.

HEREBY WE STATE THAT WE'VE USED AI TOOLS TO DEBUGGING CODE SCRIPTS AND GRAMMAR CORRECTION AND COHESION

d.