Paula Ceprián (u198630), Judit Viladecans (198724), Berta Noguera (u199893)

Final Project - Part 3

Ranking & Filtering

GitHub: https://github.com/juditvribe/IRWA_2025_G14_FinalProject.git
TAG: IRWA-2025-part-3

PART 1: Ways of ranking

Before applying any ranking method, all approaches use **conjunctive filtering**, which means that a document is only considered if it contains every term in the query. This step ensures that all ranking algorithms start from the same candidate set and ensures fairness when comparing their behavior. It also removes clearly irrelevant items early on, so each ranking algorithm only needs to score documents that are already guaranteed to match the basic intent of the query.

1.1 TF-IDF + cosine similarity

TF-IDF combined with cosine similarity ranks documents based on the similarity between the query vector and each document's TF-IDF vector. Terms are weighted by their importance in the document (TF) and their rarity in the collection (IDF), and cosine similarity normalizes for document length. This approach works well for short, focused texts but does not account for repeated terms, semantic similarity, or vocabulary mismatch.

1.2 BM25

$$RSV_d = \sum_{t \in q} \log \left[\frac{N}{\mathrm{df}_t} \right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1 - b) + b \times (L_d / L_{\mathrm{ave}})) + \mathrm{tf}_{td}}$$

BM25 uses term frequency, document length, and inverse document frequency to rank documents. Term frequency is modeled with diminishing returns, and longer documents are penalized moderately. BM25 generally produces rankings that better reflect human intuition, especially for longer texts or documents with repeated query terms, while still ignoring semantic relationships.

1.3 Our Score

For the custom ranking, we designed a scoring function that combines **TF-IDF relevance** with **product-specific metadata**. First, the query and document vectors are computed in TF-IDF space, just like in the classical cosine similarity approach. Then, additional features are incorporated to reflect product attractiveness: the average rating, discount, actual price, stock availability, and the presence of query terms in the product title. Each of these features is normalized using min-max scaling to bring them into a comparable range.

Then a weighted sum combines the TF-IDF relevance with a "bonus" for favorable metadata. For example, products with higher ratings and discounts receive positive boosts, while higher prices slightly reduce the score. Additionally, items that are out of stock are penalized by omitting the bonus.

This approach allows the ranking to capture not only textual relevance but also practical aspects that affect customer preference.

By using this ranking, the system can promote items that are both relevant to the query and more likely to be desirable, providing a richer and more user-oriented ranking than purely text-based methods.

1.4 Comparison Between Scores

Using the example query *"Comfort Women Dark Blue T-Shirt"*, we can observe how the three ranking strategies differ.

```
Top 5 by TF-IDF + cosine:
 1. TSHFPNRFYBMKHSGK  score=0.3661  Solid Women Collared Neck Blue T-Shirt
 2. TSHFXZSHHJWFGWNU  score=0.3236  Solid Women Collared Neck Dark Blue, Yellow T-Shirt  (Pack of 2)
 3. TSHFGJBYWW425HFP  score=0.3225  Self Design Women Round Neck Dark Blue T-Shirt
 4. TSHFGJBYQWQAA5ZN  score=0.3210  Self Design Women Round Neck Dark Blue T-Shirt
 5. TSHFXZSHYWFBU2HM  score=0.3185  Solid Women Collared Neck Dark Blue, Maroon T-Shirt  (Pack of 2)
```

TF-IDF + cosine similarity produces results that are textually most similar to the query, such as “Solid Women Collared Neck Blue T-Shirt” and variants in dark blue. The ranking reflects how closely the terms in the product titles and descriptions match the query, without considering other factors such as emantic or contextual aspects.

```
Top 5 by BM25:
 1. TSHFXZSHHJWFGWNU  score=1.2909  Solid Women Collared Neck Dark Blue, Yellow T-Shirt  (Pack of 2)
 2. TSHFXZSHYWFBU2HM  score=1.2909  Solid Women Collared Neck Dark Blue, Maroon T-Shirt  (Pack of 2)
 3. TSHFGJB5BMWGRXXC  score=1.1406  Abstract Women Round Neck Dark Blue T-Shirt
 4. TSHF34PEWJWG5ZTN  score=1.1303  Solid Women Round Neck Dark Blue T-Shirt
 5. TSHFGJBYWW425HFP  score=1.1201  Self Design Women Round Neck Dark Blue T-Shirt
```

BM25 favors documents where the query terms appear more frequently and in moderately sized descriptions. For example, products with multiple query terms repeated across descriptions rise to the top, such as “Solid Women Collared Neck Dark Blue, Yellow T-Shirt (Pack of 2).” This shows BM25’s ability to reward term frequency while handling length variations better than TF-IDF.

```
Top 5 by Our Score:
 1. TSHFZWRT2KTZJ7BK  score=0.8608  Solid Women Round Neck Maroon, Blue, Light Blue, Dark Blue, Black T-Shirt  (Pack of 5)
 2. TSHEANKUTX7DQZJA  score=0.8554  Solid Women Polo Neck Dark Blue, Light Blue, Black T-Shirt  (Pack of 3)
 3. TSHFXZSHYWFBU2HM  score=0.8545  Solid Women Collared Neck Dark Blue, Maroon T-Shirt  (Pack of 2)
 4. TSHFXZSHHJWFGWNU  score=0.8545  Solid Women Collared Neck Dark Blue, Yellow T-Shirt  (Pack of 2)
 5. TSHFGBY8CEWZGAVH  score=0.8511  Self Design, Solid Women Polo Neck Dark Blue, Light Blue, Blue T-Shirt  (Pack of 3)
```

Our Score produces a slightly different ranking that balances textual relevance with product metadata. Highly rated items, products with larger discounts, and those with query terms prominently in the title receive higher scores. For instance, a “Pack of 5” T-shirt set appears first, even though it is not the top textual match, because it combines relevance with favorable attributes like rating, discount, and title match. This demonstrates how integrating metadata can influence the ranking to align more closely with what a user might prefer.

In summary, TF-IDF emphasizes strict textual matching and is easy to interpret, but it struggles with repeated-term saturation and does not handle document length well. BM25 improves retrieval by rewarding term frequency and applying length normalization, but it relies on hyperparameters and still depends entirely on lexical overlap. Our custom score combines textual relevance with metadata such as rating, discount, and title match, producing more user-centric results. However, it is domain-dependent and sensitive to the weight choices used in the formula.

PART 2: Word2vec + cosine ranking score

In this section, we implemented a semantic ranking method using Word2Vec. Each product is represented as a single text by concatenating its title, description, category, sub-category, brand, and seller. As in previous parts of the lab, we first apply conjunctive filtering so that only documents containing all query terms are considered. Then, within that filtered set, we re-rank the products using Word2Vec and cosine similarity.

We train a Word2Vec model on all concatenated product texts (vector size 100, window 5, minimum count 1). To convert a product or a query into a fixed-length vector, we take the average of the embeddings of all words that appear in the text. If the text contains words $w_1,\ldots,w_n$ with vectors $v_1,\ldots,v_n$, the text vector is simply $(v_1 + \ldots + v_n) / n$. If none of the words occur in the vocabulary, we return a zero vector. This averaging approach is simple and efficient, and it captures broad semantic similarity: documents whose words are meaningfully related to the query words will tend to have similar vectors, even when they do not share the exact same terms.

Because conjunctive filtering already limits the candidate set to relevant matches, the Word2Vec re-ranking stage focuses on refining the order based on semantic closeness rather than exact lexical overlap. This allows the system to retrieve items with synonyms or paraphrased descriptions that traditional TF-IDF or BM25 might rank lower.

However, averaging word vectors has important **limitations**. It ignores word order and context, gives equal weight to all words, and may dilute the influence of key terms. In addition, training Word2Vec on a relatively small domain-specific corpus may lead to weaker vectors for uncommon or brand-specific words.

A **better representation** than averaged Word2vec would be Doc2vec, which learns a dedicated vector for each document rather than averaging word embeddings. Doc2vec captures document-level semantics more effectively and preserves more contextual information. Even stronger alternatives include Sentence2vec, or other transformer-based encoders, which produce high-quality semantic embeddings, handle paraphrasing well, and generally outperform Word2vec in retrieval tasks. Their main drawbacks are higher computational cost and, in some cases, the need for fine-tuning on labeled data.