Name: Sahil C | Ashutosh W
Class: MSc. Data Analytics
Student ID:
23100637 | 23100703

Information Retrieval
Assignment 2

1 Question 1

1.1 Given a query that a user submits to an IR system and the top N documents that are returned as relevant by the system, devise an approach (high level algorithmic steps will suffice)

- **To suggest query terms to add to the query. Typically, we wish to give a large range of suggestions to the users capturing potential intended query needs, i.e., high diversity of terms that may capture the intended query context/content.**

Ans.
The IR system returns the top N papers that may be relevant to the user's query once they submit it; however, we can take many ways, one of which is the User Feedback approach, to improve the vast variety of suggestions.

User Feedback Approach:

1. ***Presentation of Suggestions***:- After coming up with suggested terms for query growth, show them to the user in a way that is easy for them to understand.

2. ***Feedback Collection***:- Give the user choices about how to connect with the suggested terms. You can mark terms as "Relevant" or "Not Relevant," rate how important they are, or add notes.

3. ***Feedback Processing***:- Analyze the feedback provided by the user. Categorize the feedback into relevant and non-relevant terms based on user input.

4. ***Update Term Relevance Scores***:- Adjust the relevance scores of terms based on user feedback. Increase the scores of terms marked as relevant and decrease scores of terms marked as non-relevant.

5. ***Incorporate Feedback into Future Suggestions***:- Use the updated relevance scores to influence the selection and ranking of terms in future query expansion suggestions.

   Example:

   User is presented with suggested terms for query expansion related to "artificial intelligence":

   Suggested Terms: "machine learning", "neural networks", "natural language processing", "football", "breakfast recipes"

   User Feedback:

   Relevant: "machine learning", "neural networks", "natural language processing" **Not Relevant:** "basketball", "recipes"

   Updated Relevance Scores:

   "machine learning": High relevance

   "neural networks": High relevance

   "natural language processing": High relevance

   "basketball": Low relevance

   "recipes": Low relevance

   This score will be useful to us in the future, and we will continue to rate the terms in accordance with it in order to receive relevant user feedback.

   Note:- Usefulness feedback adds user opinions about document usefulness to the query suggestion process, making it better. It helps find terms that are especially important to the user's information needs, which leads to more accurate and tailored query suggestions. This process can be repeated over and over to keep improving the search results and making the question more specific.

2 Question 2

Consider the following scenario: a company search engine is employed to allow people to search a large repository. All queries submitted to the system are recorded. A record that contains the id of the user and the terms in the query is stored. The order of the terms is not stored and neither is any timestamp. Each entry in this record is effectively an id and a set of terms. Any duplicate terms in a query are ignored.

The designers of the search engine, decide to use this information to develop an approach to make query term suggestions for users, i.e., at runtime, once a user an entered their query terms, the system will suggest potential extra terms to add to the query.

Given the data available, outline an approach that could be adopted to generate these suggested terms. A brief outline is sufficient to capture the main ideas in your approach.

The designers of the system wish to take into evidence in previous queries and also any similarities between users.

Identify the advantages and disadvantages of your approach (briefly).

Ans. We can utilize a collaborative filtering-based strategy to produce query phrase suggestions based on user IDs and their submitted questions, taking into account similar inquiries from other users and past searches. Collaborative filtering generates recommendations by utilizing user behavior and preferences. Although there are several ways to tackle this, we'll use the Associate Rule Mining Approach.

Note:- We are not taking user group into consideration, here we are just considering that all the users are searching for the same example. For example, Java, here java=coffee and not other meaning(Programming languages).

Associate Rule Mining

Definition - Finding interesting correlations in big datasets can be accomplished through the use of association rule mining. Information retrieval, recommendation systems, and market basket analysis are just a few of the fields in which it is extensively used.

1. ***Generate Frequent Item sets***:- The words that a user types into a search engine are called "items" when we're suggesting question terms. The goal is to find groups of words that show up together a lot in searches. A frequent itemset is a group of terms that show up together in a lot of questions. The support of a set of things is the percentage of queries that have all the terms in the set.

   For example, consider the following queries:

   User 1 Query: ["machine learning", "AI", "data"]

   User 2 Query: ["AI", "deep learning"]

   User 3 Query: ["machine learning", "NLP", "AI"]

   Frequent item sets with a minimum support of 2:

   AI, machine learning

   AI, data

   machine learning, data

2. ***Extract Potential Query Terms***:-

   Once common item sets have been made, we can take individual terms from them. These terms are considered potential query terms that tend to occur together in queries.

   Such search terms in the given case would be "AI", "machine learning" and "data".

3. ***Rank Suggestions***:-

   Potential query phrases can then be extracted, and their support can be used to rank them. The percentage of searches that contain a phrase is its support.

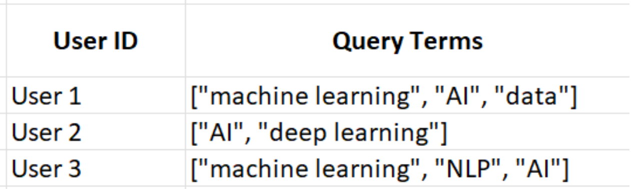
   Higher support terms are seen as more significant and are recommended as possible query terms. In the example, "AI" has 100 Percent support if it shows up in three of the three queries, indicating that it is a highly ranked proposal.

4. ***Present Suggestions to User***:-

   The user is provided with a ranked list of query term options for their consideration. These recommendations were derived from trends discovered in the historical data pertaining to user queries.

5. ***Lets Understand this with an example***:-

   Suppose we have the following user-query data:



User ID	Query Terms
User 1	["machine learning", "AI", "data"]
User 2	["AI", "deep learning"]
User 3	["machine learning", "NLP", "AI"]

Figure 1: User-query data

Potential query terms extracted from these item sets are "AI", "machine learning", and "data". These terms are ranked based on their support:

- AI: Appears in all queries (100% support)
- Machine Learning: Appears in 2 out of 3 queries (66.67% support)
- Data: Appears in 2 out of 3 queries (66.67% support)

As a result, the following search terms, in descending order of support, are suggested: "AI," "Machine Learning," and "Data." These ideas can be shown to the user so that they can decide whether or not to include them in their first search.

This strategy makes use of frequent item sets to detect terms that have a tendency to co-occur in queries. As a result, it offers helpful suggestions for improving the user's experience of searching.