Name: Harshitha Bengaluru Raghuram

Student Id: 21235396

Solution 1:

The Classical Rocchio feedback algorithm uses the Rocchio formula to calculate the weight of the query terms when a new query is given by the user. The Classical Rocchio feedback method is an example of Local Query Refinement method. Some more examples of local query refinement methods are Implicit relevance feedback and Pseudo Relevance feedback techniques. Query refinement and Query modification is performed by changing the initial query submitted by the user. The query is modified in such a way that the context is not being changed in order to match the documents present in the collection.

The classical Rocchio relevance feedback method gives output as either positive or negative. Thus, it is called as "Binary Feedback Technique". The classical Rocchio relevance feedback method is usually denoted by the following formula.

$$\overrightarrow{Q_m} = (a \cdot \overrightarrow{Q_o}) + \left(b \cdot \frac{1}{|D_r|} \cdot \sum_{\overrightarrow{D_j} \in D_r} \overrightarrow{D_j}\right) - \left(c \cdot \frac{1}{|D_{nr}|} \cdot \sum_{\overrightarrow{D_k} \in D_{nr}} \overrightarrow{D_k}\right)$$

In the formula, Q0 indicates the query in the vector form. a, b and c are the weights that are used to shape the query vector towards or away from the original query Q0 and relevant documents.
a is the query weight, b is the weight of the relevant documents and c is the weight of the non-relevant documents.
Dr is a set of relevant documents and Dnr is a set of non-relevant documents.

According to the question, The classical Rocchio relevance feedback method has to be extended to a 5 star rating system. Here, the rating scale ranges from 1 to 5 where 1 indicates irrelevant documents whereas 5 indicates the highly relevant documents. In order to implement this, the non-relevant document set which provides negative outcome can be removed.
In the 5 star rating scheme, 3 can be considered as the mark for classification in to positive or negative. Suppose a document has been rated above 3 out of 5 then the output value will be positive, else if the document has been rated below 3 out of 5 then the output value will be negative. The magnitude of the positive output value will be slightly towards 5 when compared to 4. The magnitude of the negative output value will be slightly towards 1 when compared to 2. If a document has rating equal to three then the output value will be zero or null value.

$$\vec{Q}^{\rightarrow\rightarrow\rightarrow}m^{\rightarrow\rightarrow\rightarrow} = (a.\ \vec{Q}^{\rightarrow\rightarrow\rightarrow\rightarrow}0\ ) + b \sum_{dj \in D} (1 - 3/m)\ dj$$

$\vec{Q}^{\rightarrow\rightarrow\rightarrow}m^{\rightarrow\rightarrow\rightarrow}$: The modified Query vector.

$\vec{Q}^{\rightarrow\rightarrow\rightarrow}0$: The modified Query

vector. b: Related Document

Weight.

m: The rankings incurred by the document.

dj: Document which belongs to relevant Document Set D.

Solution 2:

Query Expansion can be defined as the process of adding extra related words to the query submitted by the user in the form of suggestions which would increment the quantity of documents returned. The keywords in the user submitted query is extracted and for each keyword detected, suggestions in the form of synonyms are added.

Query Expansion is of two types namely Local Query Expansion and Global Query Expansion.

Global Query Expansion is the process of reformulating the query independent of the query submitted by the user and the results. Global Query Expansion is carried out using WordNet or Thesaurus(manually created/ automatic) or using Spelling Correction.

Local Query expansion adds the terms to the query based on the documents that initially matched the query. Blind Relevance Feedback and Global Relevance Feedback are examples of Local Query Expansion. During the process of Query Expansion, the user is given an opportunity to provide feedback on the query terms selected.

After submitting the query, the documents related to the query are extracted. Furthermore, the documents returned are analysed to find the similarities. The terms that are present together usually have similar meaning or are related to each other.

A “Term Document Matrix” can be built to summarize the frequency of terms that appear in the document collection and Term -Term Co-Occurrence Analysis can also be performed.

The terms are fetched based on whether they are related to the context or depending on their frequency of occurrences in the documents returned based on which the suggestion terms are added. The synonyms or suggestion terms are added based on the context of the query.

To give an example, if the user is searching related to “Chess” then the terms like “ Pawns”, “Bishop”, “En Passant”, “Castling” etc will have their occurrences value closest to the query and can be suggested to the user as the consecutive term.

	D1	D2	D3	D4	D5	D6	D7
T1	46	99	66	35	36	89	90
T2	09	12	16	18	36	46	69
T3	45	98	68	36	35	86	92
T4	26	78	56	28	86	98	68

Solution 3:

According to the question, all the queries submitted to the users are recorded and record that consists the id of the user along with the terms in the query are also stored. But there is no timestamp or order in which the terms are stored. Thus, the only piece of information available to search the repository is the id of the user. The aim is to build a recommender system that can suggest terms to be added to the query while the user is searching the repository. Since all the queries made to the system are recorded along with the user Id, collaborative filtering can be utilised to provide suggestions. “Collaborative Filtering is the process of filtering for information patterns using techniques involving collaboration of multiple agents and data sources.” The most commonly used algorithm for Collaborative Filtering is the Nearest Neighbourhood algorithm. It can be further divided in to User Based Collaborative Filtering and Item Based Collaborative Filtering.

Knowledge Graph or Semantic Network is used to demonstrate a network of Real World entities and the relation among the entities. Implementation of Knowledge Graphs leads to improved outcomes. The knowledge graphs can be further utilised to develop vector space with low dimension. Two types of similarity called Term-Term Similarity and Semantic Similarity are computed. Semantic Similarity can be defined as the value which is used to measure the similarity between documents or terms. The most common approach to build a Semantic Similarity Matrix is to encode the statements to obtain their embeddings. Further, the cosine similarity is used to calculate the Similarity value. Term-Term Similarity Matrix is used to compute the similarity among the terms present in the query submitted by the user. The Term-Term Similarity Matrix is created by taking into account the User Term records. The nearest neighbors obtained using the Term Term Similarity Matrix and Semantic Similarity Matrix for the User Query is collected and the ratio among them is found. Using the computed ratio both the semantic and term-term neighbors are integrated with each other. The recommendations are created by using the integration of semantic and term-term neighbors.

Advantages:

1. Data Sparsity can be defined as a problem where some values in a dataset are missing in the large scale data. Using the above proposed approach can be used to overcome the data sparsity problem.
2. The knowledge related to the domain is not necessary because the embeddings automatically learned.
3. The model has the potential to assist users in discovering new hobbies. Even if the ML system does not know the user is interested in a certain item, it may nonetheless propose it since other similar users are interested in it.

Disadvantages:

1. In Collaborative Filtering, the recommendations cannot be made across different platforms mainly because the relationship can only be found between the user and item.
2. Collaborative Filtering cannot include side features. The features apart from the essential features such as Id are considered as the side features. Adding the side features can improve the performance of the model but including the side features is a challenging task in Collaborative Filtering.
3. Cold Start Problem: The dot product of the associated embeddings represents the model's forecast for a given (user, item) pair. As a result, if an item isn't encountered during training, the system won't be able to construct an embedding for it or query the model with it.
4. The dot product of the associated embeddings represents the model's forecast for a given (user, item) pair. As a result, if an item isn't encountered during training, the system won't be able to construct an embedding for it or query the model with it.

References:

https://nuigalway.blackboard.com/webapps/blackboard/content/listContent.jsp?course_id=_131974_1&content_id=_2609291_1

https://www.cl.cam.ac.uk/teaching/1718/InfoRtrv/slides/lecture7-relevance-feedback.pdf

https://www.sciencedirect.com/topics/computer-science/query-expansion

https://nlp.stanford.edu/IR-book/html/htmledition/relevance-feedback-and-query-expansion-1.html

https://www.ibm.com/cloud/learn/knowledge-graph#:~:text=A%20knowledge%20graph%2C%20also%20known,the%20term%20knowledge%20%E2%80%9Cgraph.%E2%80%9D

https://towardsdatascience.com/semantic-similarity-using-transformers-8f3cb5bf66d6

https://discovery.ucl.ac.uk/id/eprint/1474118/

https://developers.google.com/machine-learning/recommendation/collaborative/summary