Name: Prachi Tiwari Student ID: 23102057

1. In class we discussed the document collection as term-document matrix, where each cell in the matrix indicates the usefulness of term i in describing document j. We also discussed how we could evaluate the similarity of a query and document.

Outline a suitable indexing structure to store the information in the matrix (note that matrix is sparse).

Outline at a high level, in pseudo-code, an algorithm to calculate the similarity of a document to a query.

ANSWER:

PART A ———————————————————————————————————————

A suitable indexing structure to store the information in the sparse matrix will be using Inverted index. A sparse matrix is a matrix containing majority of the elements with value 0.

Inverted index is a data structure which works by using terms i.e. words to structure the indexing, in order to quickly locate the documents or the web pages that include a term or group of related terms. In this, every term directs to a list of documents or web pages that use that term.

In layman's terms, inverted index stores a mapping between text and their positions in a document or set of documents.

Inverted index consists of two types-

1. Record-level
2. Word-level

Record-level contains a list of references to documents for each word. In word-level, along with a list of references to documents it also contains the location of each word within a document.

Example: Let's consider the below documents

1. This is Information Retrieval Assignment
2. Information Retrieval involves retrieval of documents
3. Inverted index is a data structure that directs word to a documents

The indexing of the text will be as follows:

This	(1,1)
Is	(1,2) ; (3,3)
Information	(1,3); (2,1)
Retrieval	(1,4); (2,2); (2,4)
Assignment	(1,5)

Involves	(2,3)
Of	(2,5)
Documents	(2,6); (3,12)
Inverted	(3,1)
Index	(3,2)
a	(3,4); (3,11)
Data	(3,5)
Structure	(3,6)
That	(3,7)
Directs	(3,8)
Word	(3,9)
To	(3,10)

Now, to generate the inverted index following steps needs to be followed:

STEP1: Obtain the document and remove the stop words such as “I”, “the”, “we”, “is”, “an”, as they are quite frequent and useless.

STEP2: Stemming of the root word that is, while searching for dog result document should contain the information of dog. However, term used in the document is referred to as “dogs” or “doggy”. For relating the words, we can remove some part of the word to identify the root word.

STEP3: For repeated words, we can add reference of the document to the index otherwise we can create a new entry.

Features of the inverted index are as follows:

Flexibility: Inverted indexes are flexible to diverse information retrieval system requirements.

Effective search: Inverted indexes make it possible to search through huge amounts of text-based data quickly and effectively. The index dramatically reduces search time by immediately identifying all documents because it indexes every term in document.

Support for stemming and synonym expansion can be enabled in inverted indexes, which can enhance the precision and relevancy of search results.

PART B ———————————————————————————————————————

We can use methods like Cosine Similarity, which are frequently employed in information retrieval and text analysis, to determine how similar a document is to a query. The commonly used algorithms in information retrieval are Cosine Similarity, Euclidean Distance, and Pearson's Correlation Coefficient.

Name: Prachi Tiwari Student ID: 23102057

Cosine Similarity

Cosine similarity is a calculation that expresses how similar two or more vectors are. The cosine of the angle between two vectors is the measure of similarity. Usually not zero, the vectors are located in an inner product space. The cosine of the angle ranges from -1 (opposite directions) to 1 (same direction), with 0 indicating orthogonality.

$$\text{cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

The pseudocode for the cosine similarity is as follows:

1. Create cosine_similarity function
2. Pass two arguments to function: document_vector and query_vector
3. Perform dot product between document_vector and query_vector
4. Save the above result in variable dot_product
5. Sum the square of document_vector and take the square root of it
6. Store the above result in variable document_sum
7. Sum the square of query_vector and take the square root of it
8. Store the above result in variable query_sum
9. Divide the dot_product by product of document_sum and query_sum
10. Store the above result in variable similarity
11. Return similarity

Cosine similarity has following benefits:

1. The cosine similarity, when plotted on a multi-dimensional space, captures the direction (the angle) rather than the magnitude of the data objects.
2. The cosine similarity is advantageous because it allows for the possibility that there may be a smaller angle between two similar data objects that are separated by a large Euclidean distance due to their size. The resemblance increases with decreasing angle.

---

2. With respect to D1 = {Shipment of gold damage in a fire} and a query = {gold silver truck}, consider how the similarity sim(Q, D1), should change for each of the following augmentations to D1.

(a) D1 = Shipment of gold damaged in a fire. Fire.
(b) D1 = Shipment of gold damaged in a fire. Fire. Fire.
(c) D1 = Shipment of gold damaged in a fire. Gold.
(d) D1 = Shipment of gold damaged in a fire. Gold. Gold.

Note, there is no need to show any calculations; the question pertains to how the similarity should change.

ANSWER:

Similarity is determined by considering how similar the document is to a given query.

The document we have given is,

D1 = {Shipment of gold damage in a fire}

And the query given is,

query = {gold silver truck}

Now, lets consider how similarity sim(Q, D1) will change for each of the following augmentations to D1.

(a) D1 = Shipment of gold damaged in a fire. Fire.
Because "Fire" is not present in the query, adding it to the document D1 has no meaningful impact on the similarity. The resemblance is largely unchanged.

(b) D1 = Shipment of gold damaged in a fire. Fire. Fire.
Adding several instances of the term “Fire” to the document has no effect on the similarity. The resemblance is largely unchanged.

(c) D1 = Shipment of gold damaged in a fire. Gold.
Because "Gold" is already included in the query, adding it to the document D1 has no meaningful impact on the similarity. The existence of "Gold" in both the query and the document contributes to the similarity, although only to some extent.

(d) D1 = Shipment of gold damaged in a fire. Gold. Gold.

The addition of many occurrences of the term "Gold" to the document, similar to scenario (c), has no meaningful effect on the similarity because "Gold" is already present in the query. The resemblance is largely unchanged.

---

3. In the term weighting schemes covered in class thus far, we have considered the tf factor, the idf factor and normalisation approaches.

Assuming that your document collection consists of all the scientific articles published in the Communications of the ACM (www.acm.org/dl), identify two other sources of evidence (features or sets of features) one could consider and suggest a weighting scheme that incorporates these features.

Your answer should define the evidence/feature, your reason for including it, and a means to include it in the weighting scheme.

ANSWER:

The two other sources of evidence other than the traditional approaches that can be consider for weighting scheme are Semantic Embeddings (Word Embeddings) and Citation and Reference Counts.

Semantic Embeddings (Word Embeddings)

Word embeddings are a method of numerically representing words and entire phrases. In a continuous vector space, word embeddings capture the semantic meaning of words. Rather than handling terms independently, we can include semantic information by comparing the word embeddings of terms in a document to the query.

Semantic Embeddings has several benefits that it offers-

1. The vector representation of a word might alter based on its context inside a sentence or document because word embeddings are trained on massive volumes of text data.

2. They are a type of feature learning and they develop relevant features for words or phrases, making them useful for machine learning jobs in the future.

3. They tend to show improved performance in some tasks when they are used as features in machine learning models.

Reason for including it: By taking into account the semantic relatedness of terms, including semantic embeddings can increase retrieval accuracy. Since the semantic similarity may be neglected by traditional phrase weighting systems.

Means to include it in the weighting scheme: Word embeddings can be generated using approaches such as Word2Vec, GloVe, or FastText. As a weighting factor, cosine similarity between the query and the document embeddings might be utilised. The similarity between the query and the document is then calculated by averaging or aggregating the word embeddings of phrases in both.

Citation and Reference Counts

The term "citation count" describes how many times a specific academic publication or document has been referenced in other scholarly works. The term "reference count" describes how many references or citations a specific academic publication or document has made. It denotes the breadth of the author's investigation and the information sources he or she employed. Citation and reference counts are useful characteristics in academic and research-focused document collections. In a

collection, papers with more citations or references may be more influential in the field. The amount of citations or references a work receives can be important proof of its value and relevance. If the number of reference counts are large then it indicates that the work was thoroughly studied and draws from a variety of sources, making it more strong.

Reason for including it: By including citation and reference counts in the weighting scheme, can aid in determining the importance of works that have made substantial contributions and are widely cited in the academic community.

Means to include it in the weighting scheme: Citation count of a document can be used as a characteristics in a weighting scheme for document retrieval. Reference count, like citation count, can be utilised as a characteristic in a weighting scheme.