IR_Assignment -1 Vasu Bhatia(MT23104)

Ques 1. Image Feature Extraction

Soln. We are given the dataset having the image and review corresponding to the each image, in this ques we have to do the image preprocessing by downloading the image from the given url and then perform the various function such as :-

- Altering contrast
- Resizing,
- Geometrical orientation
- Random flips
- Brightness and exposure

Note – We have also used the pre-trained Convolutional Neural Network Architecture as VGG16.

After performing the above function, we have stored the normalized feature of each link with the corresponding index in the dictionary so that we can perform the further function to implement the multimodal information retrieval.

Ques 2.

Soln. Text Feature Extraction: -

a) In this ques we are given the dataset where we are given the review in the form of text and we have preprocess that review by performing various function so that it become meaningful and can be used for computation or making the Multimodal information retrieval system .

The function that we will perform on the text to make it meaningful and understandable are :-

- Lower-Casing,
- Tokenization,
- Removing punctuations
- Stop Word Removal
- Stemming
- Lemmatization
- Blank space Removal

After performing the above function, we will get stream of token corresponding to the respective review

b) In this part we will calculate the tf-idf factor of all the token corresponding to each review of each image and save then in the dictionary according to the index.

Tf -Idf stands for the Term Frequency - Inverse Document Frequency (IDF)

Term Frequency tells us how often a term appears in a document. Calculated as the ratio of the number of occurrences of a term to the total number of terms in the document. It tells us the importance of terms that appear frequently within a specific document.

Formula - TF(t,d)=

(Number of occurrences of term t in document (d)) / (Total number of terms in document (d))

Inverse Document Frequency tells the importance of a term across a collection of documents. It is Calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term. It is intended to emphasize terms that are rare across the entire document collection.

Formula – IDF (t, D)=

log (Total number of documents in the collection (D)) / (Number of documents containing term (t))

Note - The TF-IDF score is often normalized to prevent bias towards longer documents. A common normalization is to divide the TF-IDF score by the Euclidean norm of the TF-IDF vector for a document.

Ques 3.

Soln –

a) In this part we are given an input as an image and in this we must first preprocess that picture by performing the various feature that we perform in the ques 1 and after that we will get the preprocess image in the form of an np. array.

After that we must find the cosine similarity of the inputted image with all the pre-processed image of ques 1 and then save the cosine similarity result according to the index and then sort the dictionary with respect to the cosine similarity score in the descending order, so that we can match with whom the input image is matching in the dataset.

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. It is particularly used in natural language processing and information retrieval for comparing documents or terms. Cosine similarity measures the cosine of the angle between two vectors and ranges from -1 (completely dissimilar) to 1 (completely similar).

For two vectors cosine_similarity (A, B) is calculated using the dot product and vector norms as follows:

Cosine Similarity (A, B) = (A·B) / (||A||·||B||)

b) In this part we are given an input in the form of a text and in this we must first preprocess the text by performing the various feature that we perform in the ques2 and after that we will get the preprocess text in the form of an np. array.

After performing the preprocessing, the text , we will find the cosine similarity of the inputted text with the review that we pre-processed in ques2(b) and store them in the list and after storing we will sort the list and find the first three greatest cosine similarity.

Ques4.

Soln. In this we must find the combined cosine similarity by adding the similarity of text and image and then divide it by 2 .

Ques 5

Soln. In this ques user will input the image URL and the image review, and we have to then first preprocess the image and review then find the cosine similarity among the image and preprocess input image that we have find in the ques 1.

After finding the cosine similarity we have to save it in the dictionary corresponding to index and then sort the dictionary according to the cosine similarity in the descending order.

And then take the top – 3 Similarity of the image and corresponding to that image index we have to find the review and then find the cosine similarity with the inputted text and then print the cosine similarity of the image as well as the cosine similarity of the text and the combined cosine similarity.

We have to follow the above step 3 times for the top – 3 images with the greatest cosine similarity

Ques 6

Soln. This que is same as the ques 5. But in this ques, we have to do all steps that we have done in the above ques but in this ques we have to first find the cosine similarity of the text corresponding to the input review and then save it in the dictionary according to the index and then sort the dictionary according to the similarity score and then find the top – 3 similarity score and then find the image cosine similarity score according to the index and print them

We have to follow the above steps 3 times for the top – 3 review with the greatest cosine similarity of the text.

Challenges:

1. Understanding what users want is hard because sometimes they don't explain well or use unclear words.

2. Words can mean different things, like "Python" can refer to both a programming language and a snake.

3. There's so much information online that it's easy to get overwhelmed with too many search results.

4. Finding images, videos, and audio can be tough because there's so much of it, and it's not just text.

Potential Improvements:

1. Use natural language processing to better understand what users are asking for.

2. Keep updating search results in real-time so users get the latest information.

3. Use context like location or past searches to give more relevant results.

4. Ask users for feedback to improve search results and ranking.

Note – Not uploading the pickle file as it is not being uploaded in the GitHub due to large size file