**CSE 515 Multimedia and Web Databases**

**Phase#1 Calculation of Similarity between Multimedia Objects**

**Riya Saxena**

**Team Members:**

1. **Riya Saxena 1215077747**
2. **Gaurav Mittal 1213185595**
3. **K Divya 1212953064**
4. **Harish Bhutra 1215149351**
5. **Sai Sudarshan Gollanpalli 1213383416**

**Abstract:**

The world is full of patterns and we can find some meaning out of it is a big achievement in itself. This project aims at getting similarity between the users, images and locations with the help of textual descriptors. Also, the project is about getting the similar images on the basis of given locations by the help of their visual descriptors.

**Keywords:**

* Similarity
* Metric
* Distance
* Term frequency
* Document frequency
* Images
* Data

**Introduction:**

This project is about getting the similarity between the users, images and locations on the basis of textual descriptors. There are basically five tasks. For the first three tasks, for the given textual descriptors we have to find similarity between users (task 1), images(task 2), and locations(task 3). The tasks 4 and 5 are basically based on finding the similarity between locations for the given image vectors on the basis of image visual descriptors.

* Terminology

Textual descriptors- TF(Term frequency), DF(document frequency), TF-IDF(inverse document frequency)

Goal Description

* Task 1 -

Aim: To find the most similar users for the given user id, model<tf,df,idf> and k. Also find out the terms contributing most to the similarity.

* Task 2-

Aim: To find the most similar images for the given imageid, model<tf,df,idf> and k. Also find out the terms contributing most to the similarity.

* Task 3-

Aim: To find the most similar locations for the given locationid, model<tf,df,idf> and k. Also find out the terms contributing most to the similarity.

* Task 4-

Aim: To find the most similar locations for the given locationid, model<(CM, CM3x3, CN, CN3x3,CSD,GLRLM, GLRLM3x3, HOG,LBP, LBP3x3),> and k. Also find out the image pairs contributing most to the similarity of the k locations.

* Task 5-

Aim: For given a location ID and value “k”, returns the most similar k locations based on the corresponding visual descriptors of the images

* **Assumptions**
* The distance metric used does not ensure the exact similarity between the locations for task4 and task5 as their are is dimensionality reduction involved to optimise the complexity of the solution.

**Description of the proposed solution/implementation**

**Task1** : For the given userID, file devset\_textTermsPerUser is read from the desctxt Folder and the data as a key value pair <locId, termList> is put into the dictionary. TermList has the terms mapped to the tf, df, tf-idf depending on the model selected by the user for the particular userID. For comparison between the model values of the given userID and other userIDs L2 distance as a similarity metric is used. The calculated distance is sorted in increasing order representing the users with top k scores are more similar. To calculate the terms contributing most to the similarity, for the given userId, for each term in it, the absolute difference between given model value is subtracted from all other terms model value for other userIDs and then multiplied by the given term value to encounter the high model value in case of TF. These values are sorted in increasing order and the top three terms are the ones which contribute most to the similarity for the given userID.

**Task2** : For the given imageID, file devset\_textTermsPerImage is read from the desctxt Folder and the data as a key value pair <imgId, termList> is put into the dictionary. TermList has the terms mapped to the tf, df, tf-idf depending on the model selected by the user for the particular imageID. For comparison between the model values of the given imageID and other imageID L2 distance as a similarity metric is used. The calculated distance is sorted in increasing order representing the images with top k scores are more similar. To calculate the terms contributing most to the similarity, for the given imageID, for each term in it, the absolute difference between given model value is subtracted from all other terms model value and then multiplied by the given term value to encounter the high model value in case of TF .These values are sorted in increasing order and the top three terms are the ones which contribute most to the similarity for the given imageID.

**Task3** : For the given locationID, file devset\_textTermsPerPOIwFolderNames is read from the desctxt Folder and mapped to devset\_topics.xml to get locID and location name pair. The data as a key value pair <locId, termList> is put into the dictionary. TermList has the terms mapped to the tf, df, tf-idf depending on the model selected by the user for the particular locID. For comparison between the model values of the given locID and other locIDs L2 distance as a similarity metric is used. The calculated distance is sorted in increasing order representing the users with top k scores are more similar. To calculate the terms contributing most to the similarity, for the given locID, for each term in it, the absolute difference between given model value is subtracted from all other terms model value for other locID and then multiplied by the given term value to encounter the high model value in case of TF. These values are sorted in increasing order and the top three terms are the ones which contribute most to the similarity for the given locID.

**Task4**: For the given locID and for the given model file csv file is picked from the image folder and then that file is assumed as a cluster of images. The vaerage of each field for the given and as well as for other locations is found. This is how we get the centroid of these location clusters. They are now compared using L2 distance. The distances are then sorted in increasing order. The k min distances are the locations most similar to the given location.

To compute the image pairs contributing most to the similarity for the given locations. The k locations are taken and now instead of centroid each of the image vectors in the given location is compared with the other image vectors of other k locations. The distance is sorted in increasing order and the top three image pairs are returned as the output.

**Task5:** File is read for each of the given location and each of the ten models. The image vectors in each of the location is reduced to one scalar using min-max scaling. This is done for each of the locations. After this the average is computed across all the models for each location. Thus each location is converted into a scalar. L1 distance(Manhattan distance) is used to compare the locations and k similar locations are given as output. To compute each model’s contribution, the computed min-max for each location is averaged out. Thus for each location we get how much the ten models are contributing to the similarity.

**System requirements/installation and execution instructions**

* Python3+ required
* import csv
* import datetime
* import math
* import pandas as pd
* from scipy.spatial.distance import cosine
* import numpy as np
* from scipy.spatial import distance
* from heapq import nsmallest
* import sys

Following are the libraries that need to be imported.

**Conclusions**

This project helped in how we can find similarity based on the given parameters using their features. Also, Task4, Task5 taught the power of dimensionality reduction. It can be a curse as well as a method to solve really large datasets.

**Bibliography**

1. **Evaluation of Similarity Measurement for image retrieval**

**Authors; Densheng zhang Lu.**

1. **Data Mangaement for Multimedia Retrieval, Candan**