ELL786 Assignment-3

Ramneek Singh Gambhir
Mathematics Department
Indian Institute of Technology Delhi
2018MT60788

Bhupender Dhaka
Mathematics Department
Indian Institute of Technology Delhi
2018MT60779

Ravi Pushkar
Mathematics Department
Indian Institute of Technology Delhi
2018MT60790

Shaurya Goyal
Mathematics Department
Indian Institute of Technology Delhi
2018MT60244

Amokh Varma
Mathematics Department
Indian Institute of Technology Delhi
2018MT60527

Vishal Meena
Mathematics Department
Indian Institute of Technology Delhi
2018MT60797

***Abstract*—This document describes our submission for Assignment 3 for the ELL786 (Multimedia Systems) course. Various techniques and methods which were used to develop a text based search engine, image based search engine and multi-modal search engine have been described .**

***Index Terms*—TF-IDF, Word2Vec, K-means, SIFT, BoVW**

I. Text Based Search Engine

A. Introduction

The text based search engine takes input as a word and use *Algo 1: TF-IDF* and *Algo 2: Word2Vec* to output the articles which are closest to the input word.

We used **Wikipedia** package of python to get the wikipedia article (using *wikipedia.page(articleName)*) page from the article name.

B. Approach

1) **Preprocessing**
The data (words in the documents) were preprocessed before using the algorithms by the following ways:

a) Lowercase
Lowercase is used to describe the shorter, smaller versions of letters (like w), called lowercase letters, as opposed to the bigger, taller versions (like W). This was done through python in-built function, ”lower()”.

b) Removing Stop words
A stop word is a commonly used word (such as ”the”) that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. Stop words are any word in a stop list which are filtered out before or after processing of natural language data. Ex- ’this’, ’the’, ’and’, ’but’, etc. We made an array of all the stop words and removed it whenever we encountered any stop words in the documents.

c) Removing Contractions
A contraction is a shortened form of a word (or group of words) that omits certain letters or sounds. In most contractions, an apostrophe represents the missing letters. The most common contractions are made up of verbs, auxiliaries, or modals attached to other words. Ex- ”He’d”: He would, ”ain’t”: ”are not”, etc. We made a dictionary for the contractions corresponding to their original/full forms and updated the contracting words to their full original forms.

d) Stemming
Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes. It is just like cutting down the branches of a tree to its stems. For example, the stem of the words eating, eats, eaten is eat. Also, the stem of the word studies is studi. This was done using **Porter stemming algorithm** from python **NLTK** library.

e) Lemmatization
Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. For example, the root word of studies is study. This was done using **NLTK Lemmatization** with NLTK Tokenization.

2) Algorithm I: **TF-IDF**
TF-IDF stands for “Term Frequency — Inverse Document Frequency”. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

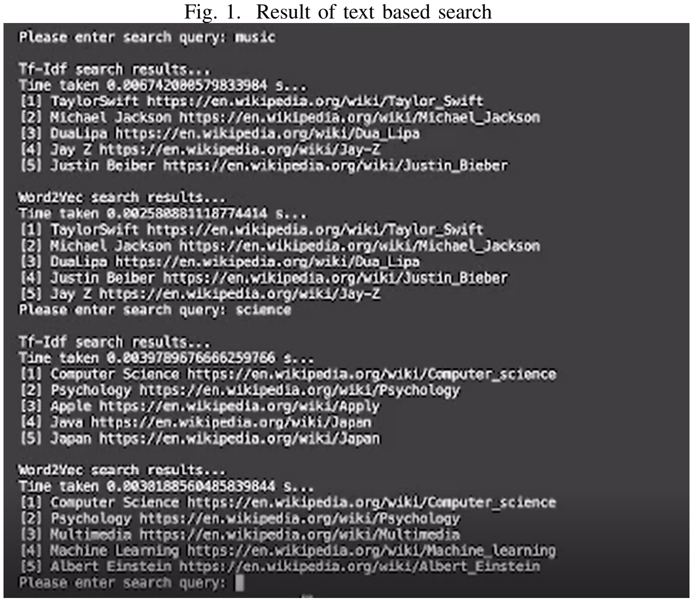
sklearn.feature_extraction.text.TfidfVectorizer.fit_transform was used to transform documents to document-term matrix. And then using *cosine similarity* between the input word and the term matrix we get the results.

3) Algorithm II: **Word2vec**
The *Word2vec* algorithm uses a neural network model to learn word associations from a large corpus of text. *Word2vec* takes as its input a large corpus of text and produces a vector space with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.

Word2vec was imported from **gensim.models** and using Word2vec() a model is trained (example: word2vec_model = Word2Vec(train_data, size=self.embeddingSize, window=5, sg=1). We can get the embedding using word2vec_model.wv, all the embedding were stored in a list (docEmbeddings). And then using *cosine similarity* between the input word and the docEmbeddings we get the results.

C. Results

Fig. 1. Result of text based search



```
Please enter search query: music

Tf-Idf search results...
Time taken 0.006742080579833984 s...
[1] TaylorSwift https://en.wikipedia.org/wiki/Taylor_Swift
[2] Michael Jackson https://en.wikipedia.org/wiki/Michael_Jackson
[3] DuaLipa https://en.wikipedia.org/wiki/Dua_Lipa
[4] Jay Z https://en.wikipedia.org/wiki/Jay-Z
[5] Justin Beiber https://en.wikipedia.org/wiki/Justin_Bieber

Word2Vec search results...
Time taken 0.002580881118774414 s...
[1] TaylorSwift https://en.wikipedia.org/wiki/Taylor_Swift
[2] Michael Jackson https://en.wikipedia.org/wiki/Michael_Jackson
[3] DuaLipa https://en.wikipedia.org/wiki/Dua_Lipa
[4] Justin Beiber https://en.wikipedia.org/wiki/Justin_Bieber
[5] Jay Z https://en.wikipedia.org/wiki/Jay-Z
Please enter search query: science

Tf-Idf search results...
Time taken 0.003978967666259766 s...
[1] Computer Science https://en.wikipedia.org/wiki/Computer_science
[2] Psychology https://en.wikipedia.org/wiki/Psychology
[3] Apple https://en.wikipedia.org/wiki/Apply
[4] Java https://en.wikipedia.org/wiki/Japan
[5] Japan https://en.wikipedia.org/wiki/Japan

Word2Vec search results...
Time taken 0.003018856048583984 s...
[1] Computer Science https://en.wikipedia.org/wiki/Computer_science
[2] Psychology https://en.wikipedia.org/wiki/Psychology
[3] Multimedia https://en.wikipedia.org/wiki/Multimedia
[4] Machine Learning https://en.wikipedia.org/wiki/Machine_learning
[5] Albert Einstein https://en.wikipedia.org/wiki/Albert_Einstein
Please enter search query: █
```

II. Image Based Search Engine

A. Introduction

The Image based search engine takes an image as input and show the images which are closest visually to the input image. The images corresponding to a word were downloaded using **icrawler**.

B. Approach

1) **Preprocessing**
We resized image to max $(1000 * 1000)$ pixels maintaining the aspect ratio. Firstly, we calculated the aspect ratio

$$\textbf{aspectratio} = \frac{\textbf{height}}{\textbf{width}}$$

from the original images. For images whose atleast one dimension (width or height) was greater than 1000 pixels. We make the larger dimension 1000 pixels while calculate the smaller dimension using the aspect ratio. However, images smaller than $(1000 * 1000)$ were left as it is.
Also, the images were changed to grayscale as grayscale image is a one layer image from 0-255 whereas the RGB have three different layer image, while the features remain intact.

2) **Method**
Then, using the module icrawler, we downloaded the images. Using SIFT algorithm in openCV library we extract the key-point descriptors of each image. We train a KNN on all these descriptors. To get vectors for an image(in our dataset as well as an query image) we create a histogram from the KNN predicted classes of each of the descriptors in the image. We also do TF-IDF re-weighting. Finally we return the closest images to a query image based on consine similarity.

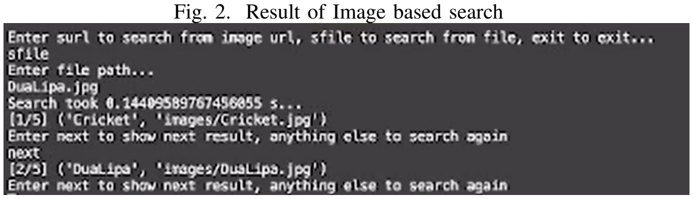
C. Results

On running the command

```
python3 Text_Search/main.py
```

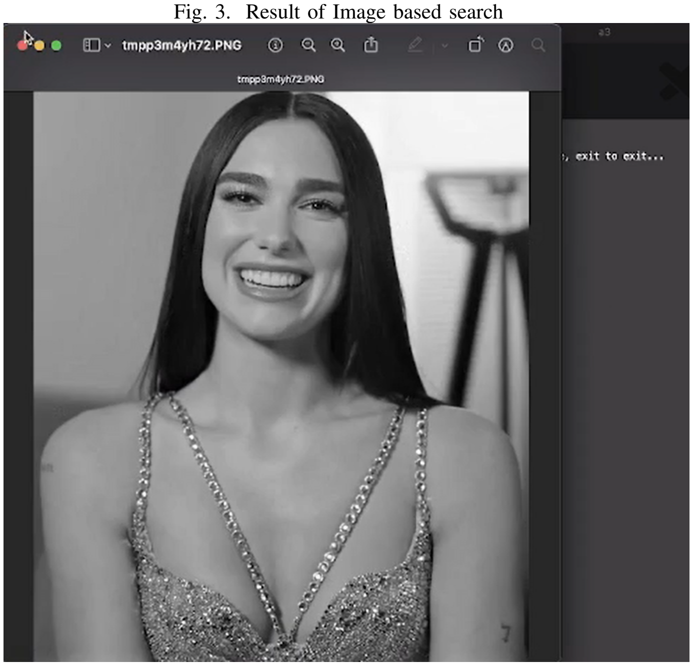
we get the following results:

Fig. 2. Result of Image based search



```
Enter surl to search from image url, sfile to search from file, exit to exit...
sfile
Enter file path...
DuaLipa.jpg
Search took 0.14409589767456055 s...
[1/5] ('Cricket', 'images/Cricket.jpg')
Enter next to show next result, anything else to search again
next
[2/5] ('DuaLipa', 'images/DuaLipa.jpg')
Enter next to show next result, anything else to search again
```

Fig. 3. Result of Image based search



```
, exit to exit...
```

III. Multi Modal Search Engine

A. Introduction

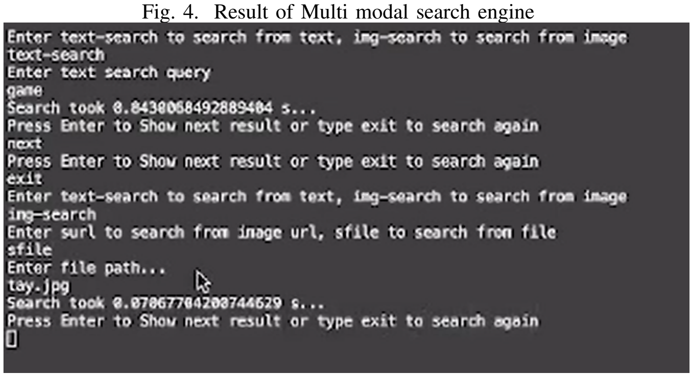
Given a word or image as input, Multimodal Search Engine return the closest words and images.

B. Approach

User can select whether he wants to run text-search or image-search, If the text-search is selected then closest words are returned using text based search engine, If image-search is selected then closest images are returned using image based search engine.

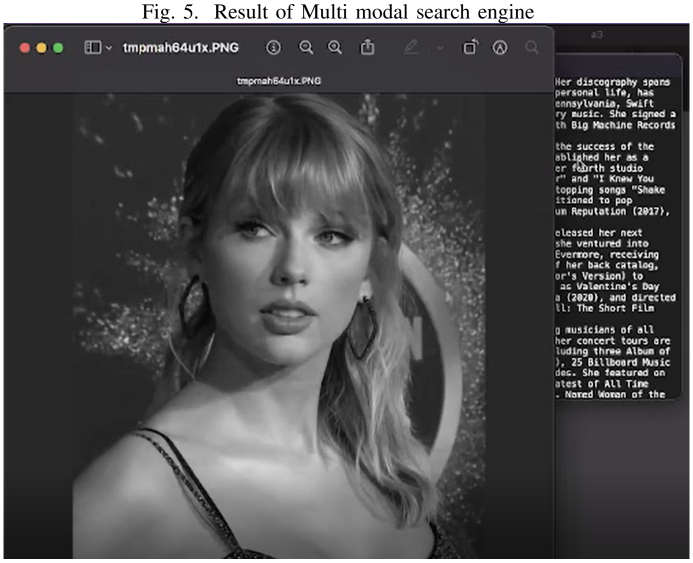
C. Results

Fig. 4. Result of Multi modal search engine



```
Enter text-search to search from text, img-search to search from image
text-search
Enter text search query
game
Search took 0.8430068492889404 s...
Press Enter to Show next result or type exit to search again
next
Press Enter to Show next result or type exit to search again
exit
Enter text-search to search from text, img-search to search from image
img-search
Enter surl to search from image url, sfile to search from file
sfile
Enter file path...
tay.jpg
Search took 0.07067704200744629 s...
Press Enter to Show next result or type exit to search again
```

Fig. 5. Result of Multi modal search engine



```
Her discography spans
personal life, has
ennsylvania, Swift
ry music. She signed a
th Big Machine Records

the success of the
ablished her as a
er fourth studio
r" and "I Knew You
topping songs "Shake
itioned to pop
um Reputation (2017),

eleased her next
she ventured into
Evermore, receiving
f her back catalog,
or's Version) to
 as Valentine's Day
a (2020), and directed
ll: The Short Film

g musicians of all
her concert tours are
luding three Album of
), 25 Billboard Music
des. She featured on
atest of All Time
. Named Woman of the
```

IV. Demo

We wrote and ran the code in VS Code on Macbook M1, which has 8-core CPU with 4 performance cores and 4 efficiency cores, 8-core GPU, 16-core Neural Engine along with 256GB SSD.
Click **here** to watch Demo videos of the working code.

References

[1] Wikipedia: Popular Pages

[2] Python: Icrawler

[3] OpenCV: SIFT

[4] Word2Vec: Gensim

[5] sklearn: TF-IDF