PROJECT DESCRIPTION

Problem Statement :
We took 30,000 documents of Google conceptual captions dataset. The aim of the project is to build a retrieval system that retrieves Top K documents that match to the given user query using IR Techniques. We built the retrieval systems using several methodologies.

Methodologies used:
By using pseudo-relevance feedback (Assuming Top K documents as relevant) IR model namely vector space model we had ranked the documents.
Similarity measures used are:
1)By computing Cosine Similarity
2)Euclidean Distance using tf-idf scores.

- ➢ **Procedure followed:**

  In Vector Space Model the index term weighting is defined as

  $[wi,j = tfi,j \cdot idfi]$

  - $tfi,j \rightarrow$ frequency of term ti in document dj (provides one measure of how well term ti describes the document contents)
  - $idfi \rightarrow$ inverse document frequency of term ti for the whole document collection (terms which appear in many documents are not very useful for distinguishing a relevant document from a non-relevant)

Three possible models for $tf_{i,j}$

- $tf_{i,j} = freq_{i,j}$ – simplest model
- $tf_{i,j} = freq_{i,j}$ max$_i$ $freq_{i,j}$ – normalized model
- $tf_{i,j} = 1 + \log_2(freq_{i,j})$ if $freq_{i,j} \geq 1$, otherwise 0

- $idf_i \rightarrow$ inverse document frequency of term t_i for the whole documents collection.

$$idf_i = \log_2 \frac{N}{n_i}$$

N = Number of documents in the collection
n_i = Number of documents containing term t

• Determining the ranking of the documents collection with respect to the given query with the following Similarity measures Euclidean Distance and Cosine Similarity.

For Euclidean Distance:

The euclidean distance of two vectors $\vec{x} = (x_1, \ldots, x_n)$ and $\vec{y} = (y_1, \ldots, y_n)$ is defined as

$$|\vec{x} - \vec{y}| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

For Cosine Similarity:

The cosine similarity between the same vectors is defined as

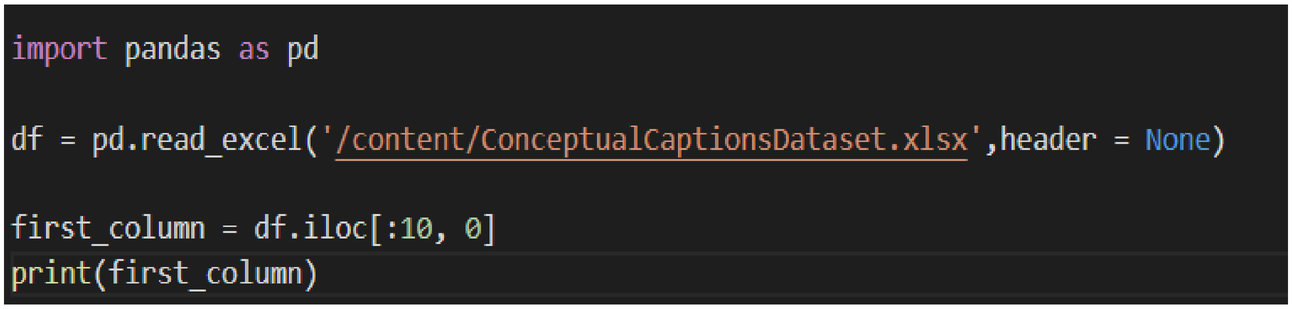
$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|} = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} \cdot \sqrt{\sum_{i=1}^{n} {y_i}^2}}$$

Basically, we implemented 4 IR Systems with two term weighting methods with 2 similarity Q, D_i measures for each. [2X2$\rightarrow$ 4]

- *Simple Term Weighting : Euclidean, Cosine Similarities.*
- *Log-Term Weighting : Euclidean, Cosine Similarities.*

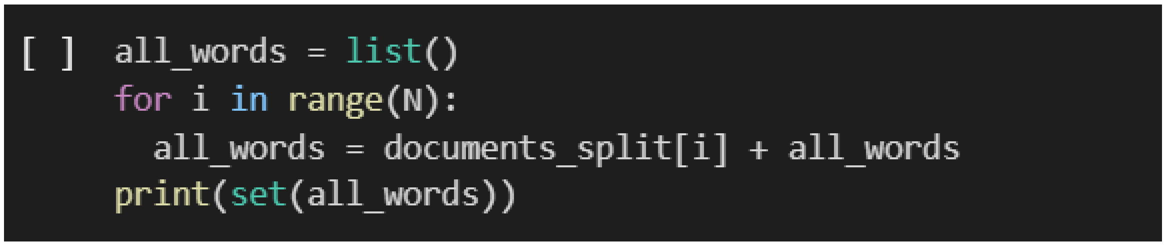
Implementation of the IR System:

❖ ***Importing Processed Dataset:***



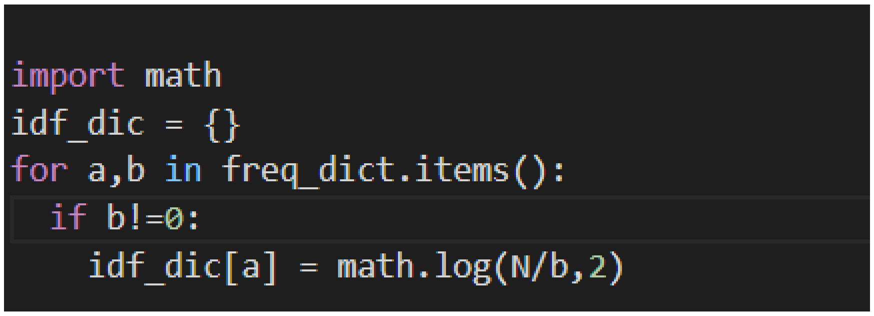
```
import pandas as pd

df = pd.read_excel('/content/ConceptualCaptionsDataset.xlsx',header = None)

first_column = df.iloc[:10, 0]
print(first_column)
```

- ## *Bag of words:*



```
[ ]  all_words = list()
     for i in range(N):
       all_words = documents_split[i] + all_words
     print(set(all_words))
```

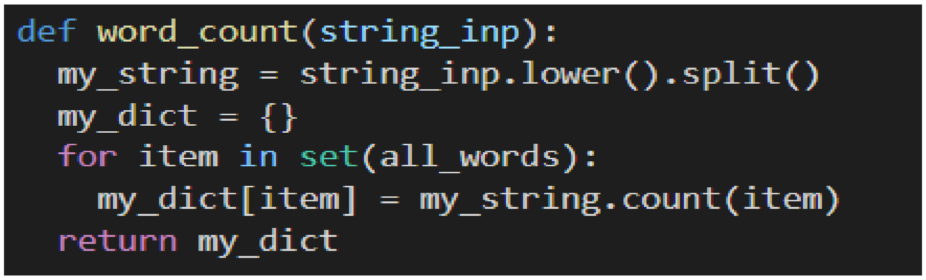
- ## *Calculating idf:*



```
import math
idf_dic = {}
for a,b in freq_dict.items():
  if b!=0:
    idf_dic[a] = math.log(N/b,2)
```

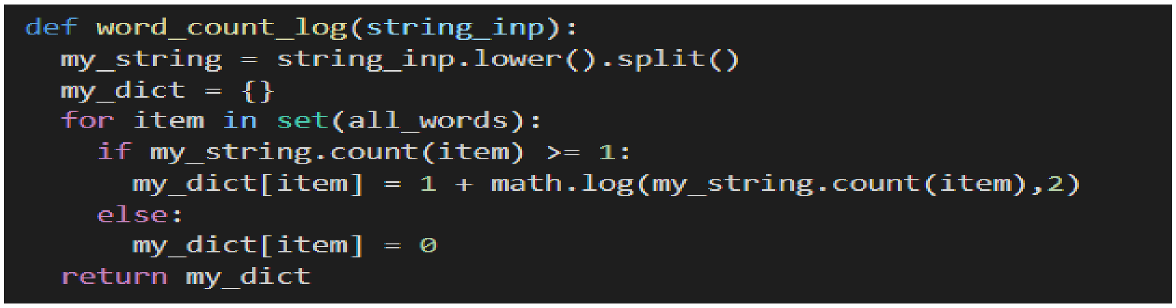
- ## *Term weighting:*

  - ### *Simple Term weighting:*



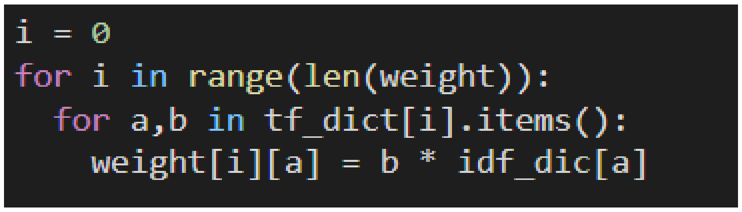
```
def word_count(string_inp):
  my_string = string_inp.lower().split()
  my_dict = {}
  for item in set(all_words):
    my_dict[item] = my_string.count(item)
  return my_dict
```

  - ### *Log Term weighting:*



```
def word_count_log(string_inp):
  my_string = string_inp.lower().split()
  my_dict = {}
  for item in set(all_words):
    if my_string.count(item) >= 1:
      my_dict[item] = 1 + math.log(my_string.count(item),2)
    else:
      my_dict[item] = 0
  return my_dict
```

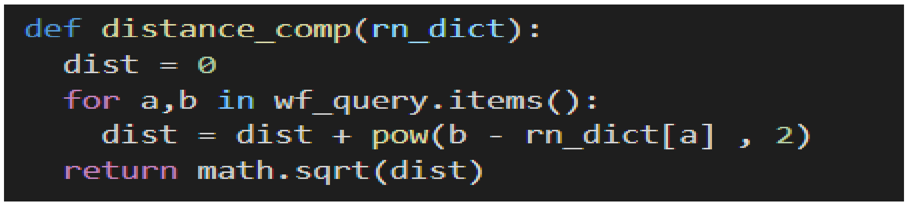
❖ *Calculating tf-idf:*



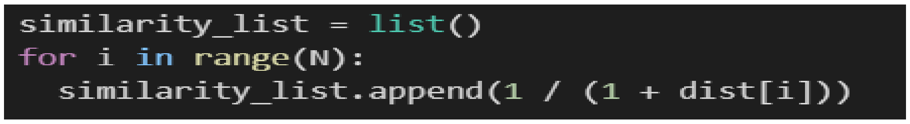
```python
i = 0
for i in range(len(weight)):
  for a,b in tf_dict[i].items():
    weight[i][a] = b * idf_dic[a]
```

❖ *Similarity Measures:*

➔ *Euclidean Distance*

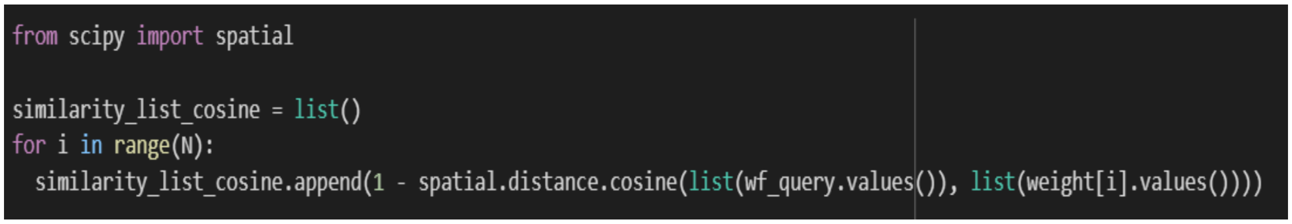


```python
def distance_comp(rn_dict):
  dist = 0
  for a,b in wf_query.items():
    dist = dist + pow(b - rn_dict[a] , 2)
  return math.sqrt(dist)
```



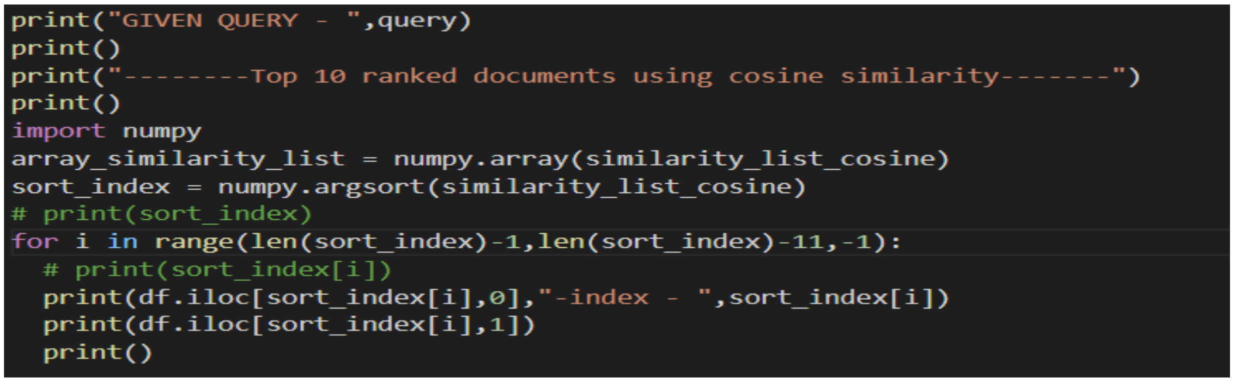
```python
similarity_list = list()
for i in range(N):
  similarity_list.append(1 / (1 + dist[i]))
```

➔ *Cosine Similarity*



```python
from scipy import spatial

similarity_list_cosine = list()
for i in range(N):
  similarity_list_cosine.append(1 - spatial.distance.cosine(list(wf_query.values()), list(weight[i].values())))
```

❖ *Ranked Retrieval of Top "K" Documents for a Query:*



```python
print("GIVEN QUERY - ",query)
print()
print("--------Top 10 ranked documents using cosine similarity-------")
print()
import numpy
array_similarity_list = numpy.array(similarity_list_cosine)
sort_index = numpy.argsort(similarity_list_cosine)
# print(sort_index)
for i in range(len(sort_index)-1,len(sort_index)-11,-1):
  # print(sort_index[i])
  print(df.iloc[sort_index[i],0],"-index - ",sort_index[i])
  print(df.iloc[sort_index[i],1])
  print()
```

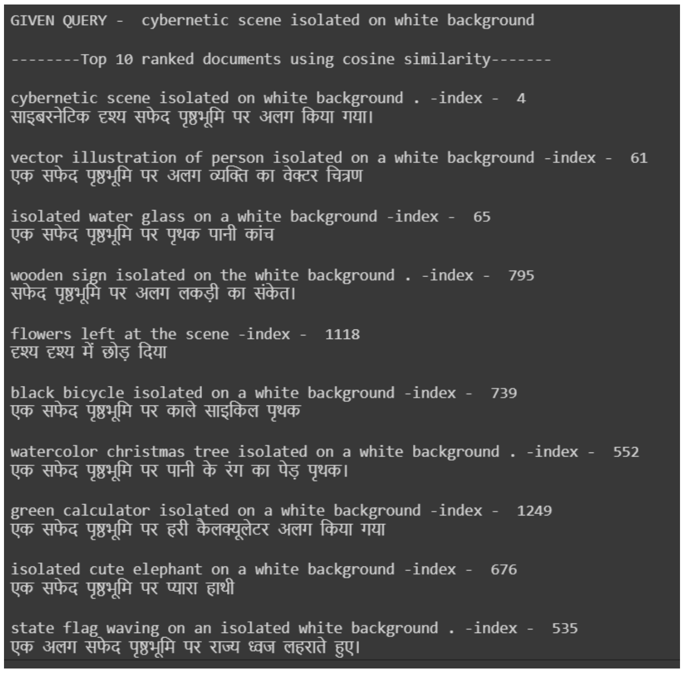
Conclusion:

By using the vector space model, we were successful to retrieve the top K documents.For Evaluation Purpose we took K=10. The retrieved documents are in pseudo-relevance manner with the most relevant document retrieved first.

We can evaluate the relevance by using similarity measures between the document and query.

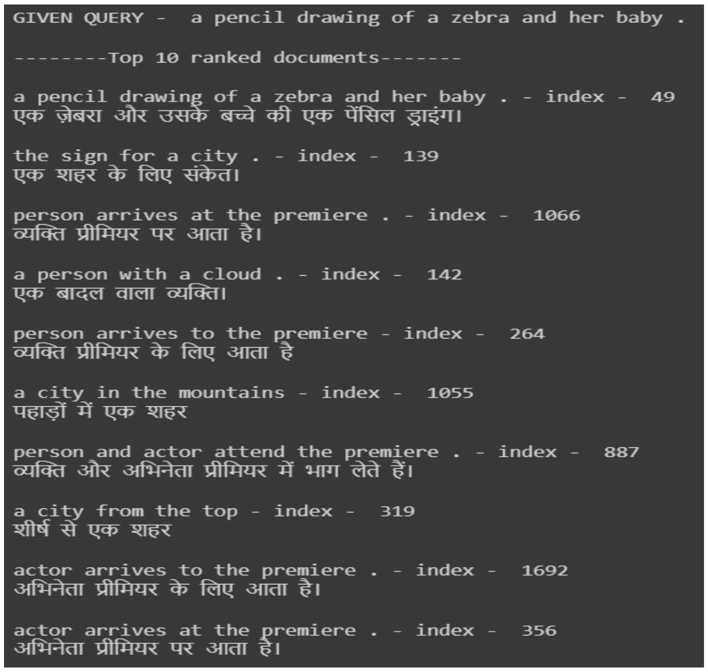
The retrieved documents are displayed along with their translated “Hindi” captions that are obtained using GTRANSLATE method in Google Sheets API.

1) Simple Term with Cosine

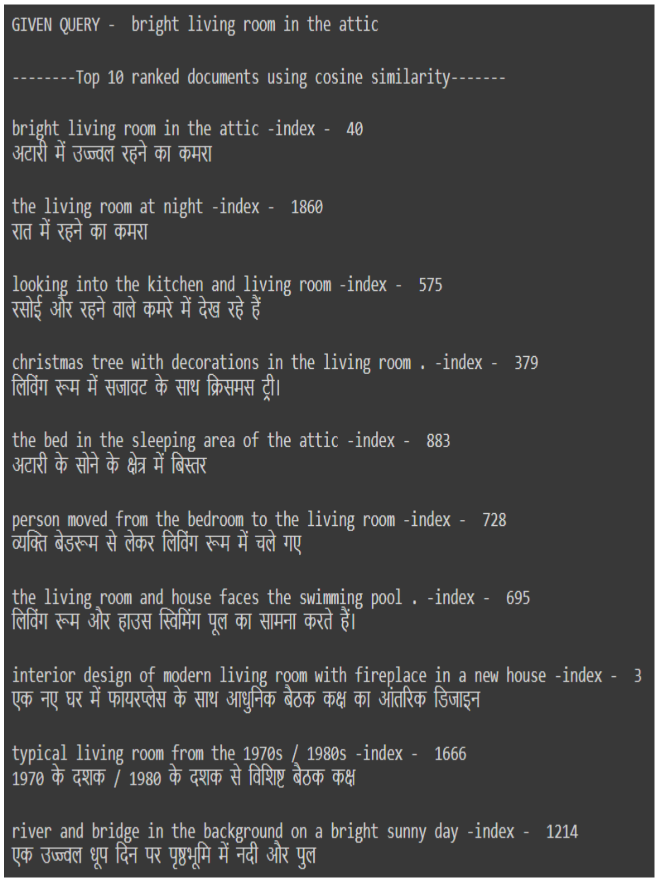


```
GIVEN QUERY -  cybernetic scene isolated on white background

--------Top 10 ranked documents using cosine similarity-------

cybernetic scene isolated on white background . -index -  4
साइबरनेटिक दृश्य सफेद पृष्ठभूमि पर अलग किया गया।

vector illustration of person isolated on a white background -index -  61
एक सफेद पृष्ठभूमि पर अलग व्यक्ति का वेक्टर चित्रण

isolated water glass on a white background -index -  65
एक सफेद पृष्ठभूमि पर पृथक पानी कांच

wooden sign isolated on the white background . -index -  795
सफेद पृष्ठभूमि पर अलग लकड़ी का संकेत।

flowers left at the scene -index -  1118
दृश्य दृश्य में छोड़ दिया

black bicycle isolated on a white background -index -  739
एक सफेद पृष्ठभूमि पर काले साइकिल पृथक

watercolor christmas tree isolated on a white background . -index -  552
एक सफेद पृष्ठभूमि पर पानी के रंग का पेड़ पृथक।

green calculator isolated on a white background -index -  1249
एक सफेद पृष्ठभूमि पर हरी कैलक्यूलेटर अलग किया गया

isolated cute elephant on a white background -index -  676
एक सफेद पृष्ठभूमि पर प्यारा हाथी

state flag waving on an isolated white background . -index -  535
एक अलग सफेद पृष्ठभूमि पर राज्य ध्वज लहराते हुए।
```

2) Simple Term with Euclidean

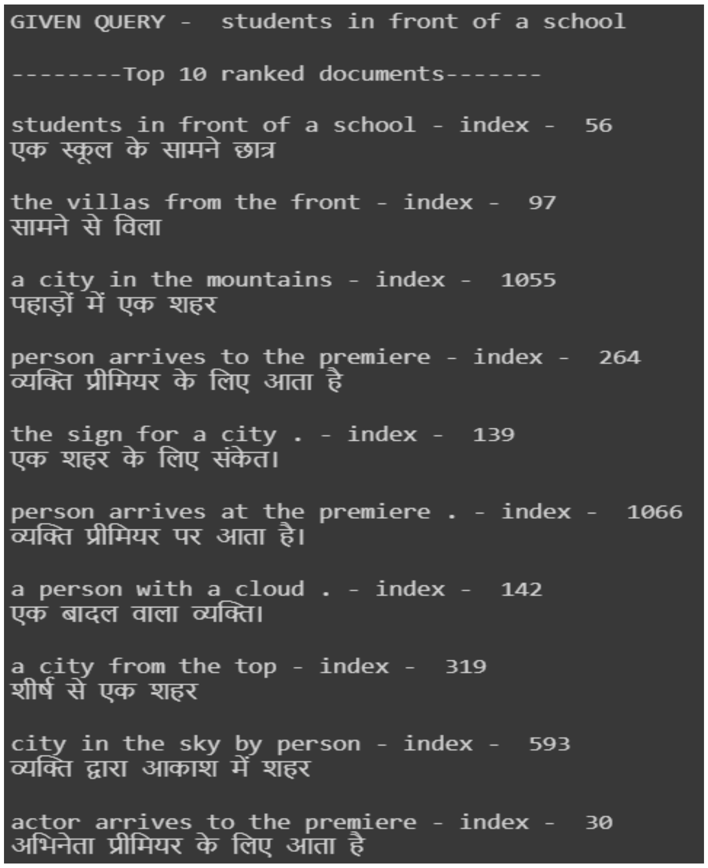


```
GIVEN QUERY -  a pencil drawing of a zebra and her baby .

--------Top 10 ranked documents-------

a pencil drawing of a zebra and her baby . - index -  49
एक ज़ेबरा और उसके बच्चे की एक पेंसिल ड्राइंग।

the sign for a city . - index -  139
एक शहर के लिए संकेत।

person arrives at the premiere . - index -  1066
व्यक्ति प्रीमियर पर आता है।

a person with a cloud . - index -  142
एक बादल वाला व्यक्ति।

person arrives to the premiere - index -  264
व्यक्ति प्रीमियर के लिए आता है

a city in the mountains - index -  1055
पहाड़ों में एक शहर

person and actor attend the premiere . - index -  887
व्यक्ति और अभिनेता प्रीमियर में भाग लेते हैं।

a city from the top - index -  319
शीर्ष से एक शहर

actor arrives to the premiere . - index -  1692
अभिनेता प्रीमियर के लिए आता है।

actor arrives at the premiere . - index -  356
अभिनेता प्रीमियर पर आता है।
```



```
GIVEN QUERY -  bright living room in the attic

--------Top 10 ranked documents using cosine similarity-------

bright living room in the attic -index -  40
अटारी में उज्ज्वल रहने का कमरा

the living room at night -index -  1860
रात में रहने का कमरा

looking into the kitchen and living room -index -  575
रसोई और रहने वाले कमरे में देख रहे हैं

christmas tree with decorations in the living room . -index -  379
लिविंग रूम में सजावट के साथ क्रिसमस ट्री।

the bed in the sleeping area of the attic -index -  883
अटारी के सोने के क्षेत्र में बिस्तर

person moved from the bedroom to the living room -index -  728
व्यक्ति बेडरूम से लेकर लिविंग रूम में चले गए

the living room and house faces the swimming pool . -index -  695
लिविंग रूम और हाउस स्विमिंग पूल का सामना करते हैं।

interior design of modern living room with fireplace in a new house -index -  3
एक नए घर में फायरप्लेस के साथ आधुनिक बैठक कक्ष का आंतरिक डिजाइन

typical living room from the 1970s / 1980s -index -  1666
1970 के दशक / 1980 के दशक से विशिष्ट बैठक कक्ष

river and bridge in the background on a bright sunny day -index -  1214
एक उज्ज्वल धूप दिन पर पृष्ठभूमि में नदी और पुल
```

3) Log-Term with Cosine Similarity measure



```
GIVEN QUERY -  students in front of a school

--------Top 10 ranked documents-------

students in front of a school - index -  56
एक स्कूल के सामने छात्र

the villas from the front - index -  97
सामने से विला

a city in the mountains - index -  1055
पहाड़ों में एक शहर

person arrives to the premiere - index -  264
व्यक्ति प्रीमियर के लिए आता है

the sign for a city . - index -  139
एक शहर के लिए संकेत।

person arrives at the premiere . - index -  1066
व्यक्ति प्रीमियर पर आता है।

a person with a cloud . - index -  142
एक बादल वाला व्यक्ति।

a city from the top - index -  319
शीर्ष से एक शहर

city in the sky by person - index -  593
व्यक्ति द्वारा आकाश में शहर

actor arrives to the premiere - index -  30
अभिनेता प्रीमियर के लिए आता है
```

4)Log-Term with Euclidean Similarity