

**Batch: A2 Roll No.: 16010421075 Experiment No.: 6**

**Aim:** Applying similarity measures on the textual datasets using cosine distance and BERT.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Resources needed:** Python

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Theory:**

**BERT:** High-performance semantic similarity with BERT

Bidirectional Encoder Representations from Transformers is a recent paper published by researchers at Google AI Language. It has caused a stir in the Machine Learning community by presenting state-of-the-art results in a wide variety of NLP tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others. BERT’s key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modelling. This is in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. The paper’s results show that a language model which is bidirectionally trained can have a deeper sense of language context and flow than single-direction language models.

Sentence similarity is one of the clearest examples of how powerful highly-dimensional magic can be. The logic is like this:

* Take a sentence, convert it into a vector.
* Take many other sentences, and convert them into vectors.
* Find sentences that have the smallest distance (Euclidean) or smallest angle

(cosine similarity) between them.

* We now have a measure of semantic similarity between sentences.

BERT, as mentioned — is the MVP of NLP. And a big part of this is down to BERTs ability to embed the meaning of words into densely packed vectors. We call them dense vectors because every value within the vector has a value and has a reason for being that value — this is in contrast to sparse vectors, such as one-hot encoded vectors where the majority of values are 0. BERT is great at creating these dense vectors, and each encoder layer (there are several) outputs a set of dense vectors.

Code: Please use the following link to find cosine distance between different statements using BERT in python:

https://www.youtube.com/watch?v=Ey81KfQ3PQU&t=102s

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Procedure / Approach /Algorithm / Activity Diagram:**

1. Convert some given set of statements into dense vectors and calculate the cosine distance between them using BERT model in python.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Results: (Program printout with output / Document printout as per the format)**

**Code:**

# hiral patel

with open ('txt1.txt', 'r') as t1:

    txt1\_st = t1.read()

with open ('txt2.txt', 'r') as t2:

    txt2\_st = t2.read()

f1 = set(txt1\_st.split(' '))

tempf1 = f1.copy()

f2 = set(txt2\_st.split(' '))

tempf2 = f2.copy()

common = f1.intersection(f2)

common = list(common)

x, y = [], []

f1 = list(map(str,txt1\_st.split(' ')))

f2 = list(map(str,txt2\_st.split(' ')))

for i in common:

      x += [f1.count(i)]

      y += [f2.count(i)]

      xy = 0

for i in range(len(x)):

      xy += x[i]\*y[i]

mod\_x, mod\_y = 0, 0

for i in range(len(x)):

      mod\_x += x[i]\*\*2

mod\_x = mod\_x\*\*0.5

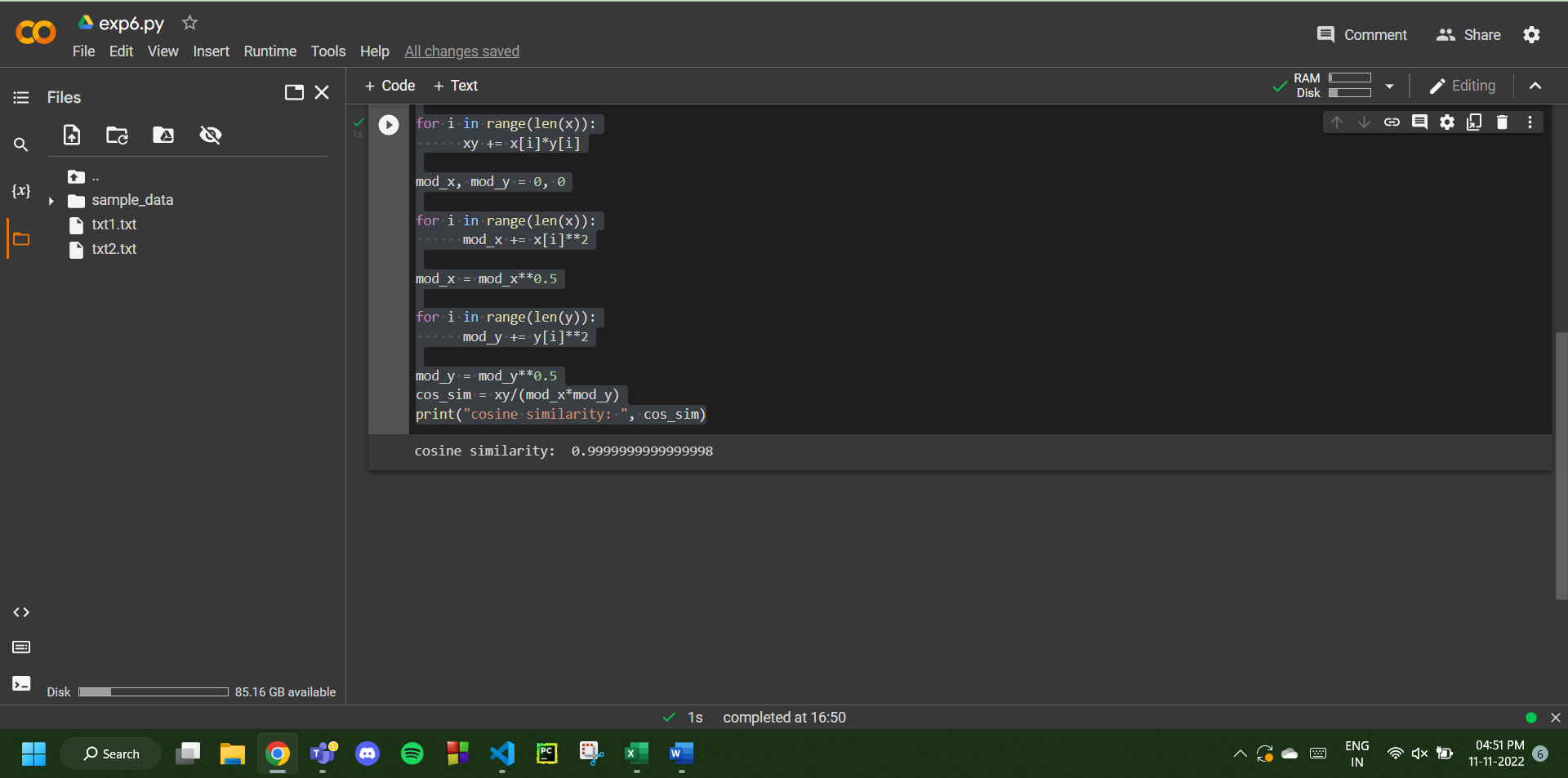
for i in range(len(y)):

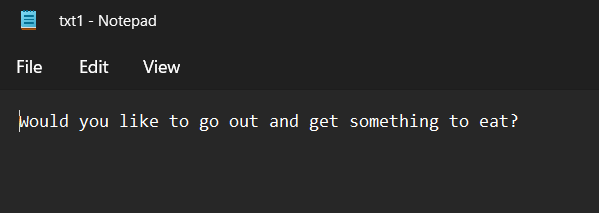
      mod\_y += y[i]\*\*2

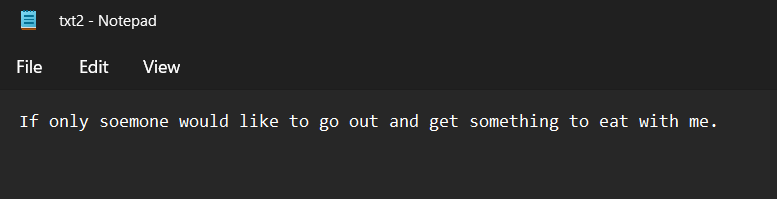
mod\_y = mod\_y\*\*0.5

cos\_sim = xy/(mod\_x\*mod\_y)

print("cosine similarity: ", cos\_sim)









**Questions:**

1. Define cosine similarity with an example.

Answer: In data analysis, cosine similarity is a measure of similarity between two sequences of numbers.

Example :  
Consider an example to find the similarity between two vectors – **‘x’** and**‘y’**, using Cosine Similarity.

The ‘x’ vector has values, x = { 3, 2, 0, 5 }  
The ‘y’ vector has values, y = { 1, 0, 0, 0 }

The formula for calculating the cosine similarity is : Cos(x, y) = x . y / ||x|| \* ||y||

x . y = 3\*1 + 2\*0 + 0\*0 + 5\*0 = 3

||x|| = √ (3)^2 + (2)^2 + (0)^2 + (5)^2 = 6.16

||y|| = √ (1)^2 + (0)^2 + (0)^2 + (0)^2 = 1

∴ Cos(x, y) = 3 / (6.16 \* 1) = 0.49

The dissimilarity between the two vectors ‘x’ and ‘y’ is given by –

∴ Dis(x, y) = 1 - Cos(x, y) = 1 - 0.49 = 0.51

* The cosine similarity between two vectors is measured in ‘θ’.
* If θ = 0°, the ‘x’ and ‘y’ vectors overlap, thus proving they are similar.
* If θ = 90°, the ‘x’ and ‘y’ vectors are dissimilar.

1. What are similarity measures applicable to textual texts other than cosine distance.

List and explain at least 3 of them.

Answer: Some of the popular similarity measures (other than cosine similarity) are –

1. Euclidean Distance: Euclidean distance is considered the traditional metric for problems with geometry. It can be simply explained as the ordinary distance between two points. It is one of the most used algorithms in the cluster analysis. One of the algorithms that use this formula would be K-mean. Mathematically it computes the root of squared differences between the coordinates between two objects.
2. Manhattan Distance: This determines the absolute difference among the pair of the coordinates. Suppose we have two points P and Q to determine the distance between these points we simply have to calculate the perpendicular distance of the points from X-Axis and Y-Axis.  
   In a plane with P at coordinate (x1, y1) and Q at (x2, y2). Manhattan distance between P and Q = |x1 – x2| + |y1 – y2|
3. Jaccard Similarity: The Jaccard distance measures the similarity of the two data set items as the intersection of those items divided by the union of the data items.

**Outcomes:** CO2: Comprehend descriptive and proximity measures of data.

**Conclusion:** We were successful in applying similarity measures on the textual datasets using cosine distance and BERT.



**Grade: AA / AB / BB / BC / CC / CD /DD**

Signature of faculty in-charge with date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **References:**

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd

Edition

1. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining.

Pearson Education India, 2016.