# Experiment No.6

**Title:** Applying similarity measures on the textual datasets

**Batch: B4** **Roll No.: 16010420133** **Experiment No.: 6**

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

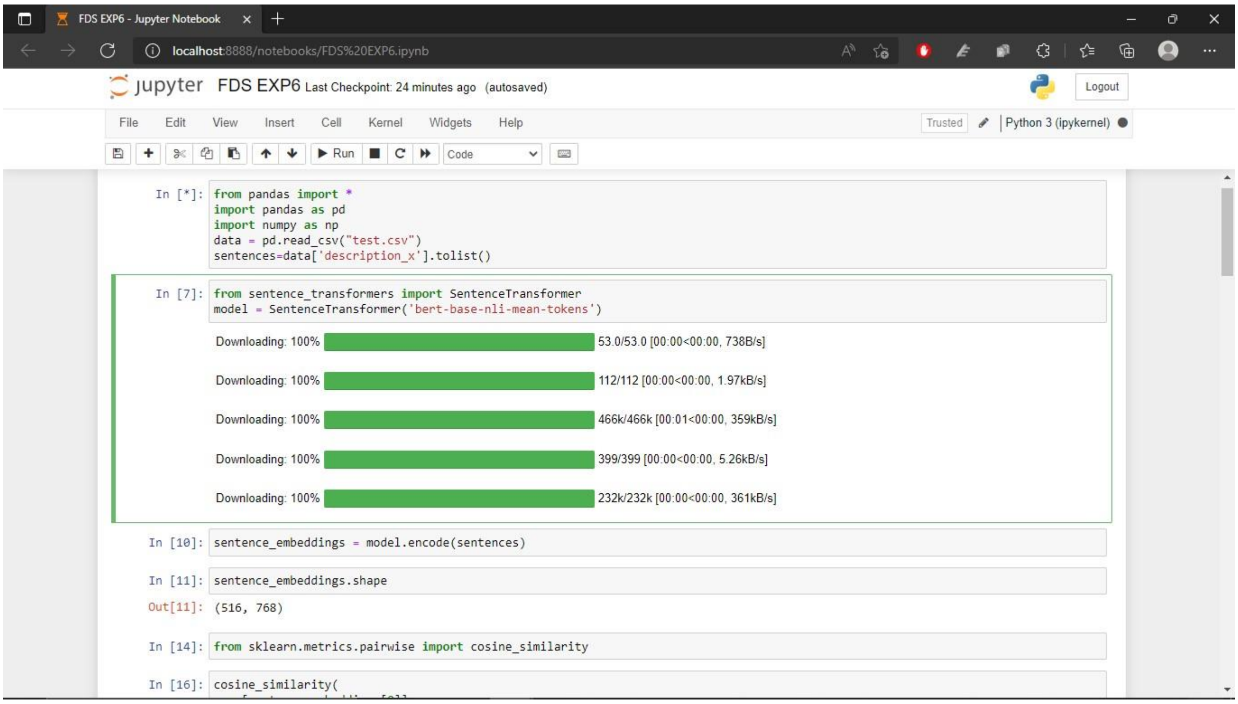
---

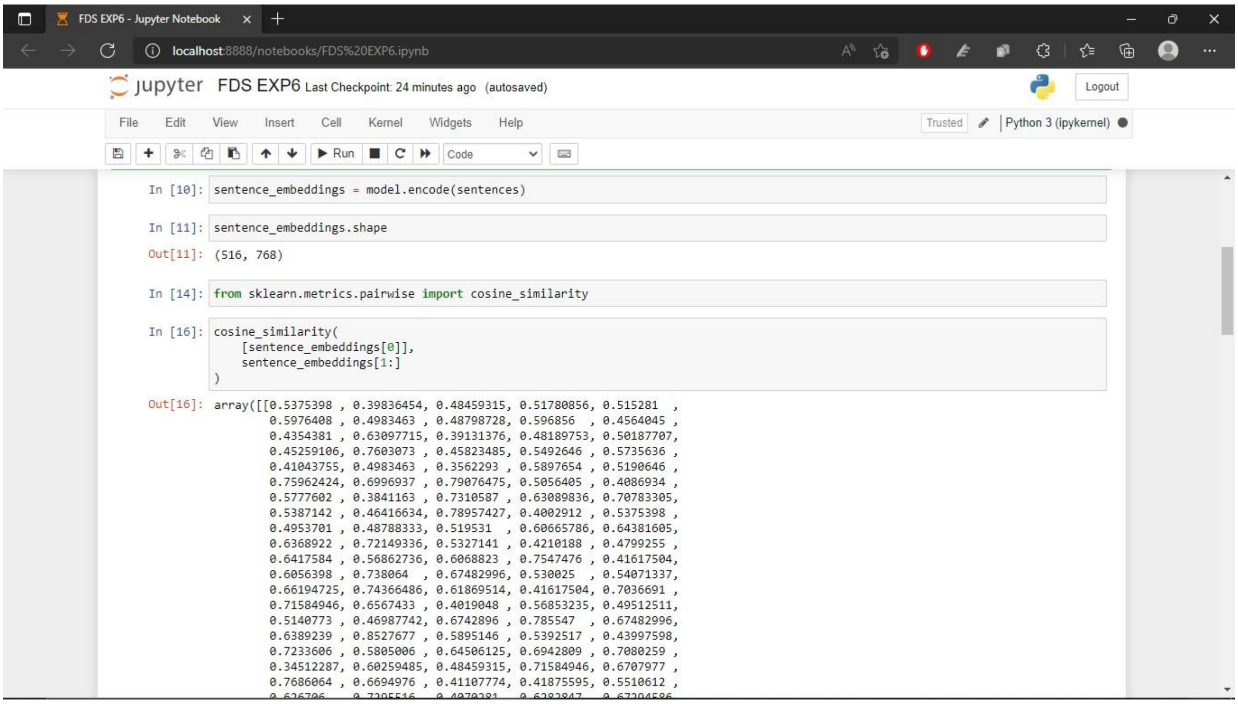
**Resources needed:** Python

---

**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.



```
In [16]: cosine_similarity(
             [sentence_embeddings[0]],
             sentence_embeddings[1:]
         )

Out[16]: array([[0.5375398 , 0.39836454, 0.48459315, 0.51780856, 0.515281  ,
                 0.5976408 , 0.4983463 , 0.48798728, 0.596856  , 0.4564045 ,
                 0.4354381 , 0.63097715, 0.39131376, 0.48189753, 0.50187707,
                 0.45259106, 0.7603073 , 0.45823485, 0.5492646 , 0.5735636 ,
                 0.41043755, 0.4983463 , 0.3562293 , 0.5897654 , 0.5190646 ,
                 0.75962424, 0.6996937 , 0.79076475, 0.5056405 , 0.4086934 ,
                 0.5777602 , 0.3841163 , 0.7310587 , 0.63089836, 0.70783305,
                 0.5387142 , 0.46416634, 0.78957427, 0.4002912 , 0.5375398 ,
                 0.4953701 , 0.48788333, 0.519531  , 0.60665786, 0.64381605,
                 0.6368922 , 0.72149336, 0.5327141 , 0.4210188 , 0.4799255 ,
                 0.6417584 , 0.56862736, 0.6068823 , 0.7547476 , 0.41617504,
                 0.6056398 , 0.738064  , 0.67482996, 0.530025  , 0.54071337,
                 0.66194725, 0.74366486, 0.61869514, 0.41617504, 0.7036691 ,
                 0.71584946, 0.6567433 , 0.4019048 , 0.56853235, 0.49512511,
                 0.5140773 , 0.46987742, 0.6742896 , 0.785547  , 0.67482996,
                 0.6389239 , 0.8527677 , 0.5895146 , 0.5392517 , 0.43997598,
                 0.7233606 , 0.5805006 , 0.64506125, 0.6942809 , 0.7080259 ,
                 0.34512287, 0.60259485, 0.48459315, 0.71584946, 0.6707977 ,
                 0.7686064 , 0.6694976 , 0.41107774, 0.41875595, 0.5510612 ,
                 0.626706  , 0.7295516 , 0.4070281 , 0.6282847 , 0.67294586,
                 0.721545  , 0.3230321 , 0.47742856, 0.60753655, 0.426233  ,
                 0.62356377, 0.70055604, 0.39307052, 0.47112888, 0.49258387,
                 0.5443704 , 0.4285535 , 0.6545238 , 0.6093745 , 0.47207683,
                 0.48654285, 0.5510993 , 0.38145378, 0.5266507 , 0.62236273,
                 0.5043524 , 0.4003529 , 0.70268047, 0.5691283 , 0.41011897,
                 0.5370861 , 0.39089447, 0.5189902 , 0.54176575, 0.56605315,
                 0.73110896, 0.37961233, 0.71584946, 0.56426513, 0.5912481 ,
                 0.5852668 , 0.63003564, 0.5758078 , 0.38299286, 0.48341605,
```

---

**Questions:**

1. Define cosine similarity with an example.

Cosine similarity is one of the metric to measure the text-similarity between two documents irrespective of their size in Natural language Processing. A word is represented into a vector form. The text documents are represented in n-dimensional vector space.

Mathematically, Cosine similarity metric measures the cosine of the angle between two n-dimensional vectors projected in a multi-dimensional space. The Cosine similarity of two documents will range from 0 to 1. If the Cosine similarity score is 1, it means two vectors have the same orientation. The value closer to 0 indicates that the two documents have less similarity. he mathematical equation of Cosine similarity between two non-zero vectors is:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Eg:

doc_1 = "Data is the oil of the digital economy"
doc_2 = "Data is a new oil"

# Vector representation of the document
doc_1_vector = [1, 1, 1, 1, 0, 1, 1, 2]
doc_2_vector = [1, 0, 0, 1, 1, 0, 1, 0]

| | data | digital | economy | is | new | of | oil | the |
|---|---|---|---|---|---|---|---|---|
| doc_1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 |
| doc_2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |

$$A \cdot B = \sum_{i=1}^{n} A_i B_i$$

$$= (1*1) + (1*0) + (1*0) + (1*1) + (0*1) + (1*0) + (1*1) + (2*0)$$

$$= 3$$

$$\sqrt{\sum_{i=1}^{n} A_i^2} = \sqrt{1+1+1+1+0+1+1+4} = \sqrt{10}$$

$$\sqrt{\sum_{i=1}^{n} B_i^2} = \sqrt{1+0+0+1+1+0+1+0} = \sqrt{4}$$

$$cosine\ similarity = cos\theta = \frac{A \cdot B}{|A||B|} = \frac{3}{\sqrt{10} * \sqrt{4}} = 0.4743$$

2. What are similarity measures applicable to textual texts other than cosine distance. List and explain at least 3 of them.

1) edit distance
2) cosine distance
3) Jaro distance
4) n-Gram distance
5) longest common subsequence

**Edit Distance:**

evenshtein distance is the most frequently used algorithm. It was founded by the Russian scientist, Vladimir Levenshtein to calculate the similarities between two strings. This is also known as the Edit distance-based algorithm as it computes the number of edits required to transform one string to another. The edits count the following as an operation:

- Insertion of a character
- Deletion of a character
- Substitution of a character

More the number of operations, less is the similarity between the two strings.

**Jaro Distance**

Jaro Similarity is the measure of similarity between two strings. The value of Jaro distance ranges from 0 to 1. where 1 means the strings are equal and 0 means no similarity between the two strings.

The Jaro Similarity is calculated using the following formula

$$Jaro\ similarity = \begin{cases} 0, \text{ if m=0} \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{m}\right), \text{ for m!=0} \end{cases}$$

where:

- m is the number of matching characters
- t is half the number of transpositions
- where |s1| and |s2| are the lengths of strings s1 and s2 respectively.

### n-Gram distance

The main idea behind n-gram similarity is generalizing the concept of the longest common subsequence to encompass n-grams, rather than just unigrams. We formulate n-gram similarity as a function sn, where n is a fixed parameter. s1 is equivalent to the unigram similarity function s .

**Outcomes:**

CO 2 Comprehend descriptive and proximity measures of data

**Conclusion:**

I have implemented the cosine distance on the text stored in the textual dataset using BERT in python.

**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 $3^{nd}$ Edition
2. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar. Introduction to data mining. Pearson Education India, 2016.