Machine Learning Tutorials

Learn Machine Learning and Artificial Intelligence

NLP Tutorial

Tensorflow : BERT Fine-tuning with GPU

NATURAL LANGUAGE PROCESSING

Jaccard Similarity – Text Similarity Metric in NLP

By Bhavika Kanani on Friday, April 24, 2020

Jaccard Similarity is also known as the **Jaccard index** and **Intersection over Union** . **Jaccard Similarity** matric used to determine the similarity between two text document means how the two text documents close to each other in terms of their context that is how many common words are exist over total words.

In Natural Language Processing, we often need to estimate text similarity between text documents. There are many text similarity matric exist such as **Cosine similarity** , **Jaccard Similarity** and **Euclidean Distance** measurement. All these text similarity metrics have different behaviour.

In this tutorial, you will discover the **Jaccard Similarity** matric in details with example. You can also refer to this tutorial to explore the **Cosine similarity** metric.

Jaccard Similarity defined as an intersection of two documents divided by the union of that two documents that refer to the number of common words over a total number of words. Here, we will use the set of words to find the intersection and union of the document.

The mathematical representation of the **Jaccard Similarity** is:

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2}$$

The Jaccard Similarity score is in a range of **0 to 1** . If the two documents are identical, Jaccard Similarity is **1** . The Jaccard similarity score is **0** if there are no common words between two documents.

Let's see the example about how to **Jaccard Similarity** work?

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

Let's get the set of unique words for each document.

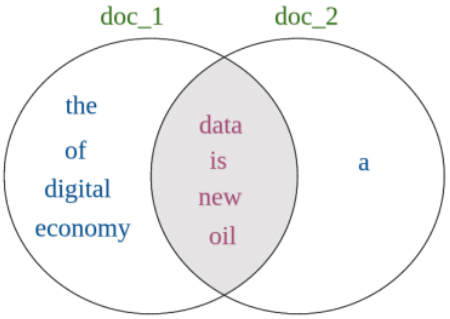
```
words_doc1 = {'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'}
words_doc2 = {'data', 'is', 'a', 'new', 'oil'}
```

Now, we will calculate the intersection and union of these two sets of words and measure the **Jaccard Similarity** between **doc_1** and **doc_2**.

$$J(doc_1, doc_2) = \frac{\{'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'\} \cap \{'data', 'is', 'a', 'new', 'oil'\}}{\{'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'\} \cup \{'data', 'is', 'a', 'new', 'oil'\}}$$

$$= \frac{\{'data', 'is', 'new', 'oil'\}}{\{'data', 'a', 'of', 'is', 'economy', 'the', 'new', 'digital', 'oil'\}}$$

$$= \frac{4}{9} = 0.444$$



Python Code to Find Jaccard Similarity

Let's write the Python code for Jaccard Similarity.

```
def Jaccard_Similarity(doc1, doc2):

    # List the unique words in a document
    words_doc1 = set(doc1.lower().split())
    words_doc2 = set(doc2.lower().split())

    # Find the intersection of words list of doc1 & doc2
    intersection = words_doc1.intersection(words_doc2)

    # Find the union of words list of doc1 & doc2
    union = words_doc1.union(words_doc2)

    # Calculate Jaccard similarity score
    # using length of intersection set divided by length of union set
    return float(len(intersection)) / len(union)



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

Jaccard_Similarity(doc_1,doc_2)



0.44444
```

The Jaccard similarity between **doc_1** and **doc_2** is **0.444**

. . .

NATURAL LANGUAGE PROCESSING

A complete introduction to GPT-3 with Use Case examples

NATURAL LANGUAGE PROCESSING

Deep Unveiling of the BERT Model

NATURAL LANGUAGE PROCESSING

Word Embedding

Leave a Reply

Comment

Name *

Email *

Submit