Deekshant Mamodia MT19119

ASSIGNMENT 5

Preprocessing Steps:

- Remove the punctuations from the documents and queries(Qi+Oi).
- Convert all the data into lowercase.
- Remove all the digits using regex.

Observation

Doc2Vec :

- Doc2vec is an unsupervised algorithm to generate vectors for sentence/paragraphs/documents.

- The vectors generated by doc2vec can be used for tasks like finding similarity between sentences/paragraphs/documents.

Difference between Doc2vec and Word2Vec:

- In word2vec, you train to find word vectors and then run similarity queries between words. In doc2vec, you tag your text and you also get tag vectors.

- Word2Vec computes a feature vector for every word in the corpus, Doc2Vec computes a feature vector for every document in the corpus.

- Word2Vec works on the intuition that the word representation should be good enough to predict the surrounding words, the underlying intuition of Doc2Vec is that the document representation should be good enough to predict the words in the document.

Documents returned by 2 models:

- First Model returned the Termed Frequency -Inverse Document Frequency (Tf-idf) matrix between the vocab of the corpus and the documents.

- The second Model returned a Trained vector Model of Documents

Similarity Score:

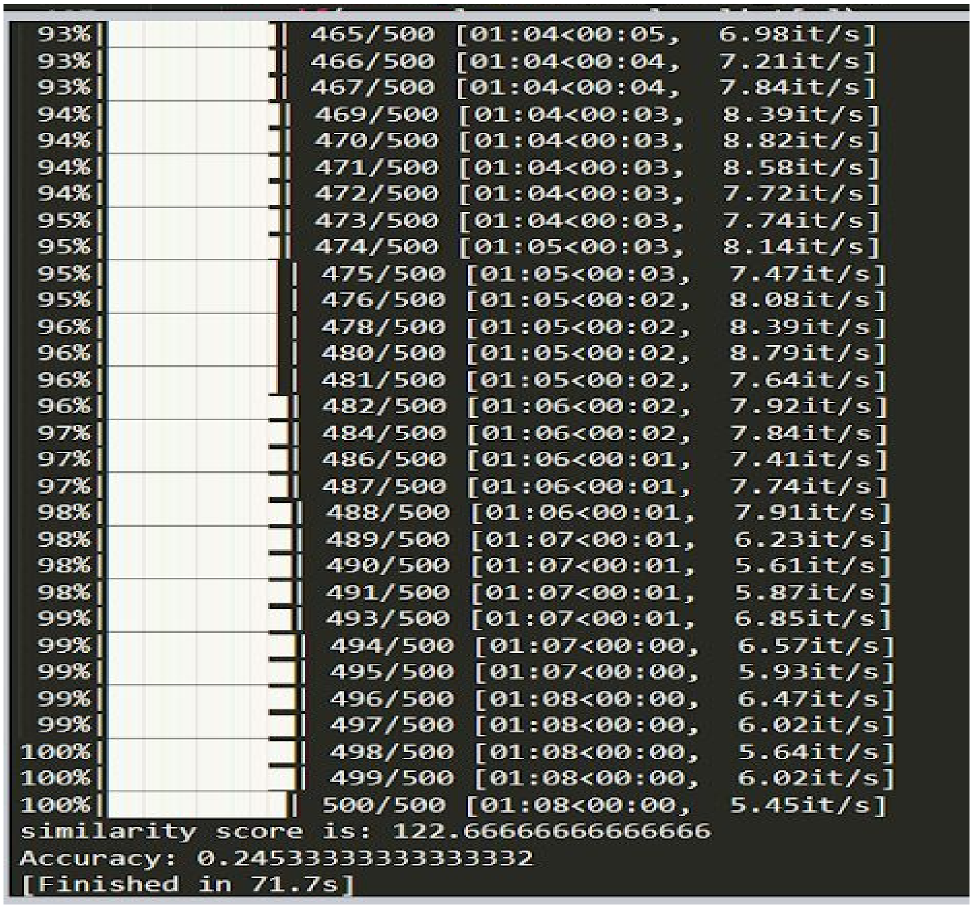
- The similarity score for Task 1 is: 122.666
- The similarity score for Task 2 is: 140.25 (ranges from 119 to 145)

Inferences:

- For both the model the testing document (Querry) having maximum word similarity to training document having high similarity score.

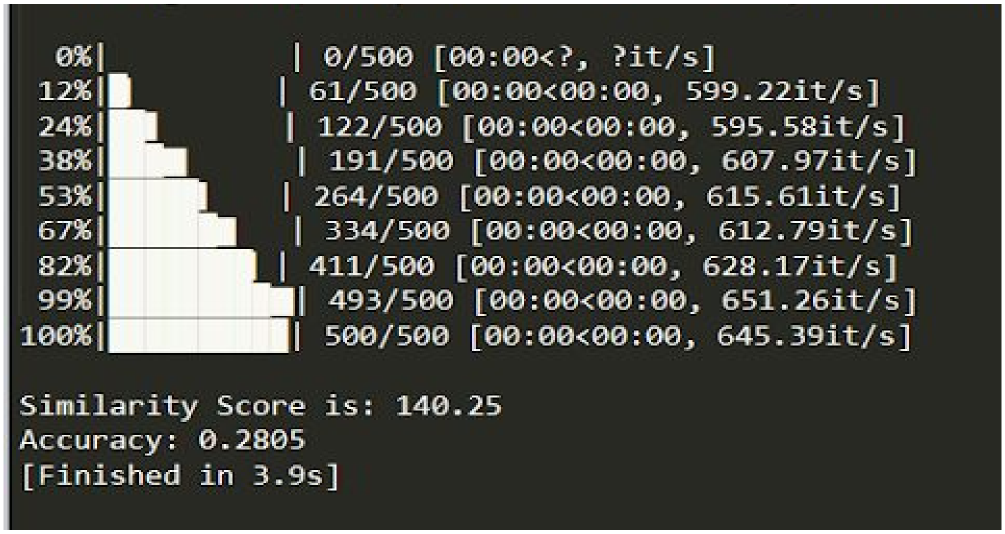
Accuracy Values

- **Task1:** 0.2453



```
 93%|█████████▎| 465/500 [01:04<00:05,  6.98it/s]
 93%|█████████▎| 466/500 [01:04<00:04,  7.21it/s]
 93%|█████████▎| 467/500 [01:04<00:04,  7.84it/s]
 94%|█████████▍| 469/500 [01:04<00:03,  8.39it/s]
 94%|█████████▍| 470/500 [01:04<00:03,  8.82it/s]
 94%|█████████▍| 471/500 [01:04<00:03,  8.58it/s]
 94%|█████████▍| 472/500 [01:04<00:03,  7.72it/s]
 95%|█████████▍| 473/500 [01:04<00:03,  7.74it/s]
 95%|█████████▍| 474/500 [01:05<00:03,  8.14it/s]
 95%|█████████▌| 475/500 [01:05<00:03,  7.47it/s]
 95%|█████████▌| 476/500 [01:05<00:02,  8.08it/s]
 96%|█████████▌| 478/500 [01:05<00:02,  8.39it/s]
 96%|█████████▌| 480/500 [01:05<00:02,  8.79it/s]
 96%|█████████▌| 481/500 [01:05<00:02,  7.64it/s]
 96%|█████████▋| 482/500 [01:06<00:02,  7.92it/s]
 97%|█████████▋| 484/500 [01:06<00:02,  7.84it/s]
 97%|█████████▋| 486/500 [01:06<00:01,  7.41it/s]
 97%|█████████▋| 487/500 [01:06<00:01,  7.74it/s]
 98%|█████████▊| 488/500 [01:06<00:01,  7.91it/s]
 98%|█████████▊| 489/500 [01:07<00:01,  6.23it/s]
 98%|█████████▊| 490/500 [01:07<00:01,  5.61it/s]
 98%|█████████▊| 491/500 [01:07<00:01,  5.87it/s]
 99%|█████████▊| 493/500 [01:07<00:01,  6.85it/s]
 99%|█████████▉| 494/500 [01:07<00:00,  6.57it/s]
 99%|█████████▉| 495/500 [01:07<00:00,  5.93it/s]
 99%|█████████▉| 496/500 [01:08<00:00,  6.47it/s]
 99%|█████████▉| 497/500 [01:08<00:00,  6.02it/s]
100%|█████████▉| 498/500 [01:08<00:00,  5.64it/s]
100%|█████████▉| 499/500 [01:08<00:00,  6.02it/s]
100%|██████████| 500/500 [01:08<00:00,  5.45it/s]
similarity score is: 122.66666666666666
Accuracy: 0.24533333333333332
[Finished in 71.7s]
```

- **Task2:** 0.23-.029



```
  0%|          | 0/500 [00:00<?, ?it/s]
 12%|█         | 61/500 [00:00<00:00, 599.22it/s]
 24%|██▍       | 122/500 [00:00<00:00, 595.58it/s]
 38%|███▊      | 191/500 [00:00<00:00, 607.97it/s]
 53%|█████▎    | 264/500 [00:00<00:00, 615.61it/s]
 67%|██████▋   | 334/500 [00:00<00:00, 612.79it/s]
 82%|████████▏ | 411/500 [00:00<00:00, 628.17it/s]
 99%|█████████▊| 493/500 [00:00<00:00, 651.26it/s]
100%|██████████| 500/500 [00:00<00:00, 645.39it/s]

Similarity Score is: 140.25
Accuracy: 0.2805
[Finished in 3.9s]
```