Quiz 2 Report

Owen Yang

February 2022

1 Cosine Similarity

As discussed in class, the cosine similarity function is computed by the inner product of two vectors divided by product of vector norms. In the `cosine_similarity()` function, the variable `inner_product` is first computed by the sum of element wise product. Notice that we only take the product of the intersection of the dictionaries/sets, which is that we only consider similarity of co-occurred words in the two documents. This makes sense because, if under extreme scenario, no words are shared among two documents, then it would be natural (although very impractical because we forget synonyms, or replacement phrasing) to infer that they have low similarity metric. The variable `x11` and `x22` are vector L_2 norms which are computed by convention.

2 Vectorize & Similar Documents

In the `vectorize()` function, three vector representation methods are considered: "tf_idfs", "ntf", and "wf". They are calculated according to the formulas presented in class. Overall, "tf_idfs" results in the most sensible outcome because nearly 96% of the similar document pairs are consisted of the same stories but with versional variations. "ntf" also have strong output with only a few unexpected results. The abnormalities are reasonable because the formula does not consider the balancing impact of inverse document frequencies, which makes the algorithm neglect certain important terms while they have low document frequencies. "wf" have similar output as "ntf" with larger customizability because we can choose what α be. In my setting, I set $\alpha = 0.2$, but $\alpha \in \mathbb{R} - \{1\}$.

The `most_similar()` function nested inside the `similar_documents()` function is intended to compare the cosine similarity of the given input and return the maximum pair. Cosine similarity usually works well for text input because the scaling invariant nature ignores the impact of size difference among documents. This means that although one document is noticeably larger than the other, they could still share high similarity score if they convey similar semantic information. This is exemplified in this quiz because the different versions of the same story may vary a lot by size, but the cosine similarity is able to ignore the scalability factor and assign them with high similarity score.