**CSCE 5200 Information Retrieval and Web Search**

**Assignment 6**

The following are the assignment 6 questions. Submit your assignment through Canvas.

1. Consider the table of term frequencies for 3 documents denoted Doc1, Doc2, Doc3 in Figure 6.9. Recall the tf-idf weights computed in Assignment 5 question 6. Compute the Euclidean normalized document vectors for each of the documents using the tf-idf weights for terms, where each vector has four components, one for each of the four terms.

*Figure 6.9: Table of tf values*

|  |  |  |  |
| --- | --- | --- | --- |
| Term Frequencies: | Doc1 | Doc2 | Doc3 |
| car | 27 | 4 | 24 |
| auto | 3 | 33 | 0 |
| insurance | 0 | 33 | 29 |
| best | 14 | 0 | 17 |

*Figure 6.8: Example of idf values. Here we give the idf’s of terms with various frequencies in the Reuters collection of 806,791 documents.*

|  |  |  |
| --- | --- | --- |
| TERM |  |  |
| car | 18,165 | 1.65 |
| auto | 6,723 | 2.08 |
| insurance | 19,241 | 1.62 |
| best | 25,235 | 1.5 |

1. With term weights as computed above (i.e., the tf-idf weights), rank the three documents by computed score for the query **car insurance**, for each of the following cases of term weighting in the query:
2. The weight of a term is 1 if present in the query, 0 otherwise.
3. Euclidean normalized idf.

3. If we were to stem jealous and jealousy to a common stem before setting up the vector space, detail how the definitions of tf and idf should be modified.

4. Compute the vector space similarity between the query “digital cameras” and the

document “digital cameras and video cameras” by filling out the empty columns in the following table. Assume *N* = 10,000,000, logarithmic term weighting (wf columns) for

query and document, idf weighting for the query only and cosine normalization for

the document only. Treat and as a stop word. Enter term counts in the tf columns.

What is the final similarity score?

|  |  |  |  |
| --- | --- | --- | --- |
| Word: | query  tf wf df idf | document  tf wf |  |
| digital | 10,000 |  |  |
| video | 100,000 |  |  |
| cameras | 50,000 |  |  |

5.Consider the case of a query term that is not in the set of *M* indexed terms; thus our standard construction of the query vector results in *V*(*q*) not being in the vector space

created from the collection. How would one adapt the vector space representation to

handle this case?