1. (10 points) Prove that for normalized document vectors, the Euclidean distance gives the same ranked ordering as the cosine measure.

2. (10 points) Given prior knowledge of which documents are relevant and which are not, we wish to construct a query that distinguishes the relevant from the irrelevant documents (using their vector-space representations). Explain how we can construct such a query. What assumptions are you making?

3. (12 points) Tinker with the Google search engine for some sample queries and answer the following questions:

   (a) Does it perform any form of stemming on your input? If so, what algorithm do you think it uses?

   (b) Does it utilize stopwords? *Hint:* Try a query that has only stop words, e.g., ‘when and where is it’ or ‘this and the that.’

   (c) Write a query whose first hit is your instructor’s webpage. Your query must not contain any reference to the instructor’s name. What do you learn?

4. (18 points) When we restrict our attention to binary vectors, there are many more similarity metrics available than the simple cosine measure. Two examples are the Dice coefficient and the Jaccard’s measure (sometimes called the Tanimoto coefficient). Research these measures with an eye toward: (i) how they are defined, (ii) what their ranges are, (iii) how do they compare with the cosine measure (e.g., are they more strict or more lenient), (iv) what are ways to get higher ranking with these measures, and (v) the applications they are suited for. Write a concise report summarizing your experiments and observations. Do not make rambling statements.

5. (10 points) In this question we will tackle the problem of cross-language IR. For the same collection of documents, we are given two term-document modelings, the first in English and the second in German. Notice that the number of documents is the same, but the number of terms will be different. Explain how the GVSM (generalized vector space model) can be extended so that queries can be supplied in either German or English term spaces.


   - Use the stopwords present in the collection and the Porter’s stemmer (http://www.tartarus.org/~martin/PorterStemmer/) to create a vector space model of the TIME collection. Explore (bfc), (tfc), and (nfc) encodings for both the term-document matrix and the queries. How many terms are there in your modeling?

   - Apply simple cosine similarity as well as GVSM for retrieval.

   - Compute precision at 11 fixed recall levels (0 to 1 in steps of 0.1) and also perform the mapping so that precision is a non-increasing function of recall. Average the curves across all queries.
• Interpret your results w.r.t. both the vector-space and GVSM models and w.r.t. the encodings.

For full credit, provide a URL where your code and data can be inspected and a report listing: (i) the design choices you made, (ii) some sample precision-recall curves for targeted queries (just a few) and (iii) interpretation of results.