Assignment 2: K-nearest neighbor for text classification.

The goal of text classification is to identify the topic for a piece of text (news article, web-blog, etc.). Text classification has obvious utility in the age of information overload, and it has become a popular turf for applying machine learning algorithms. In this project, you will have the opportunity to implement *k-nearest neighbor* and apply it to text classification on the well known Reuter news collection.

1. Download the dataset from my website, which is created from the original collection and contains a training file, a test file, the topics, and the format for train/test.
2. Implement the k-nearest neighbor algorithm for text classification. Your goal is to predict the topic for each news article in the test set. Try the following distance or similarity measures with their corresponding representations.
    a. Hamming distance: each document is represented as a boolean vector, where each bit represents whether the corresponding word appears in the document.
    b. Euclidean distance: each document is represented as a numeric vector, where each number represents how many times the corresponding word appears in the document (it could be zero).
    c. Cosine similarity with TF-IDF weights (a popular metric in information retrieval): each document is represented by a numeric vector as in (b). However, now each number is the TF-IDF weight for the corresponding word (as defined below). The similarity between two documents is the dot product of their corresponding vectors, divided by the product of their norms.
3. Let w be a word, d be a document, and N(d,w) be the number of occurrences of w in d (i.e., the number in the vector in (b)). TF stands for term frequency, and TF(d,w)=N(d,w)/W(d), where W(d) is the total number of words in d. IDF stands for inverted document frequency, and IDF(d,w)=log(D/C(w)), where D is the total number of documents, and C(w) is the total number of documents that contains the word w; the base for the logarithm is irrelevant, you can use e or 2. The TF-IDF weight for w in d is TF(d,w)*IDF(d,w); this is the number you should put in the vector in (c). TF-IDF is a clever heuristic to take into account of the "information content" that each word conveys, so that frequent words like "the" is discounted and document-specific ones are amplified. You can find more details about it online or in standard IR text.
4. You should try k = 1, k = 3 and k = 5 with each of the representations above. Notice that with a distance measure, the k-nearest neighborhoods are the ones with the smallest distance from the test point, whereas with a similarity measure, they are the ones with the highest similarity scores.