SI 650/EECS 549 - Information Retrieval
Assignment 1 (Due Wednesday, Feb. 11th)

Please submit as pdf attachment via CTools.
General discussion encouraged, but everyone should come up with the solution independently.
If you received help from anyone, you should list the name(s) on the top of your submission.

1. **Edit Distance** [10 points]

   Find the edit distance of the following pairs of strings (No need to include the table in your submission):

   “heterogeneous” and “hetterogenous”:

   “wolverine tower” and “wolverrine taller”:

2. **Vector Representation and Similarity** [25 points]

   Consider the following query and documents:

   (1) q: computer information;
   (2) d1: school of information;
   (3) d2: school of informatics and computing;
   (4) d3: school of information and library science.

   A. [5 points] Represent the query and documents with vectors. Describe the dimensions of your vector space.
   B. [5 points] Compute the cosine similarity between the query and each of the three documents. Which document is closer to q? Rank the three documents by the cosine similarity.
   C. [5 points] Now remove the stopwords from the query and documents. Will the removal of stopwords affect the ranking of the documents?
   D. [5 points] Suppose you have a stemmer that can normalize “computer” and “computing”, and “information” and “informatics” into the same base form. Will the use of this stemmer change the ranking?
   E. [5 points] However, it looks like the query isn’t looking for an iSchool but is looking for information about computers. How would you define the dimensions of your vector space differently so that all the documents will have a smaller cosine similarity with the query?

3. **Probabilistic Reasoning and Bayes Rule** [20 points]

   Consider the problem of classifying whether a tweet is a spam. This problem can be modeled using probabilities by treating each tweet as an observation of the values of the following 4 random variables:

   (1) S: whether the tweet is a spam (1 for yes, 0 for no);
   (2) A: whether the author has an avatar (1 for yes, 0 for no);
   (3) L: whether the tweet has an URL link (1 for yes, 0 for no);
   (4) K: whether the author has more than K followers (1 for yes, 0 for no).

Given a message, we can observe the values of A, L, and K, and we want to infer its value of S. In terms of probabilistic reasoning, we are interested in evaluating the conditional probability $P(S|A, L, K)$, and we would say that the tweet is a spam if $P(S = 1|A, L, K) > P(S = 0|A, L, K)$. We make a further conditional independence assumption that $P(A, L, K|S) = P(A|S)P(L|S)P(K|S)$ for $S = 0$ and $S = 1$. In other words, we assume that if the status whether a tweet is a spam is known (i.e., value of S is known), the values of A, K, and L would be independent to each other. For a more detailed explanation of the concept "conditional independence", please read this page: (http://www.dcs.qmw.ac.uk/∼norman/BBNs/Independence_and_conditional_independence.htm).

A. [10 points] Use the Bayes formula and the following known conditional probabilities to compute the probability that a tweet with A=1, L=0, and K=0 is a spam. Show the computation $P(S = 1|A = 1, L = 0, K = 0)$ and $P(S = 0|A = 1, L = 0, K = 0)$. Would we conclude that the tweet is a spam? (*hint: You need to first compute $P(L = 0|S)$ and $P(K = 0|S)$, and then try to compute $P(S|A = 1, L = 0, K = 0)$ based on $P(A = 1, L = 0, K = 0|S)$*).

V	$P(A = 1\|S)$	$P(L = 1\|S)$	$P(K = 1\|S)$	prior $P(S)$
0	0.1	0.9	0.9	0.9
1	0.99	0.2	0.05	0.1

B. [5 points] In probability theory, can we change all the 8 probabilities in the table above to arbitrary values freely between 0 and 1? If not, are there any constraints on some values that we must follow?

C. [5 points] Explain how you can change our conclusion regarding whether the tweet is a spam by changing the value of *only one cell* in the table.

4. **Word Distribution** [45 points]

Install the NLTK package (http://www.nltk.org/). Download two text collections of similar number of words (congress_speech.txt and blog.txt) and a list of stopwords (stopword.txt). Tokenize the two collections using the NLTK package (use the **nltk.word_tokenize()** function) and then write a program that compute the frequency of words. You can use any programming language for computing the statistics. Now plot the frequency distribution of words in each collection after the removal of the stopwords: x-axis - word frequency (number of times a word appears in the collection); y-axis - proportion of words with this frequency. Plot the distributions on a log-log scale. Does each plot look like a power-law distribution? Are the two distributions similar or different? (20 points)

Now compare the two collections more rigorously. Report the following properties of each collection:

(a) vocabulary size (number of unique words);

(b) frequency of stopwords;

(c) number of capital letters;

(d) average number of characters per word;

(e) number of nouns, adjectives, verbs, adverbs, and pronouns;

(f) the top 10 nouns, top 10 verbs, and top 10 adjectives.

Can you explain these differences based on the nature of the two collections? (25 points) (For part of speech tagging, use the **nltk.pos_tag()** function of the NLTK package.)

(Hint: you can find a tutorial of nltk.word_tokenize() and nltk.pos_tag() in the NLTK book chapter 3 and 5: `http://www.nltk.org/book/ch03.html` and `http://www.nltk.org/book/ch05.html`. There is also a simple tutorial of NLTK at `http://www.slideshare.net/japerk/nltk-in-20-minutes`. Just use the simple, default settings.)