CSC869: Data Mining, San Francisco State University
Mini project #1: Association Pattern-based Family History Identification

1. **Due** by 11:55PM, Monday, March 9, 2015.

2. **Submission:** Submit your solution and documents through iLearn.

3. **Dataset:** The dataset for this project is a collection of transcribed medical reports available on the MTSample website (http://mtsamples.com/). I have uploaded a total of 1274 such reports to the iLearn site under "Mini-projects" as a zip file.

4. **Problems**

(i) Identify all the sentences that contain at least one of the following family members in each of the 1274 reports. We will term such sentences as "candidate family history sentences".

mother, father, brother(s), sister(s), aunt(s), grandfather(s), grandmother(s), uncle(s), son(s), daughter(s), cousin(s), mom, dad, nephew(s), niece(s)

Note that this task contains an implicit subtask: sentence boundary detection. This is a common preprocessing task for text analysis. You can find many ready-to-use source codes online for this task. For your reference, here are several sources that contain a module for this subtask: (1) the GATE toolkit (http://gate.ac.uk/), (2) the MXPOST software as listed on http://nlp.stanford.edu/links/statnlp.html ; (3) the NIST/DUC splitter as listed on http://courses.washington.edu/ling573/software.html; and (4) the sentence splitter posted on iLearn. This splitter is written in Python by Neal Lewis, one of my previous graduate research assistants. Instructions on running this program is included in the program itself.

(ii) Remove the *stop words* from each candidate family history sentence. For a list of commonly used stop words and the rationale behind this step, please refer to http://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html.

(iii) Treat each transformed family history sentence as a bag of words and identify all the word associations that occur in at least 5 medical reports. For this task, you are allowed to use an existing software such as Weka (http://www.cs.waikato.ac.nz/ml/weka/) or any of the association mining algorithms available at http://www.kdnuggets.com/software/associations.html. Should you decide to implement your own version, bonus points will be granted.

(iv) Post-process all the word associations, remove an association from further consideration if it has a span of over k words in more than 60% of the original sentences that contain this association, where k is a parameter. For example, the word association {mother, diabetes} has a span of 5 words in the sentence "Her mother was diagnosed with diabetes." In this assignment, please experiment the following values of k: 3, 5, and 10.

(v) For each of the remaining association patterns, further process it by integrating the occurring order of the involved words in the original sentences. Take the pattern {mother, diabetes} as an example again. Let us assume it occurs in 10 medical reports. Up to this point, we have not considered the relative arrangement of these two words in

these 10 reports. In this step, we will locate all these 10 reports that contain this pattern. We then identify the number of times that the word "mother" is positioned before the word "diabetes", and the number of times that "mother" is positioned after "diabetes". We will name these ordered patterns as **wordLists**.

(vi) Order these wordLists by their frequency and characterize the top ***N*** lists from the following aspects: (a) how many of them contain at least one family member? (b) how many of them contain at least one of the following diseases? (c) how many of them contain one family member but no disease? (e)how many of them contain both a family member and a disease? (f) how many of them contain neither a family nor a disease? and (g) with respect to each family member, what are the characteristics of the top 20 frequent patterns? Please set ***N*** to the following values in this project: 100, 200, 500, and 1000.

Disease list:

breast cancer	ADHD	HTN
Breast cancer	bipolar	hypertension
CA	bipolar disorder	brain aneurysm
cancer	depressed	cerebral aneurysm
colon cancer	depression	cerebrovascular accident
gastric carcinoma	mental illness	stroke
lung cancer	mood disorder	strokes
prostate cancer	mood disorder/bipolar	adult-onset diabetes
renal CA	nervous breakdowns	diabetes
throat cancer	Schizophrenia	diabetes mellitus
CHF	suicide	DM
CAD	coronary heart disease	type 2 diabetes
acute myocardial infarction	heart attack	alcohol abuse
congestive heart failure	heart disease	alcoholic
coronary artery disease	Heart disease	alcoholism
myocardial infarction	heart failure	alcohol to excess
valvular heart disease	MI	alcohol use
vascular strokes	drug addict	deceased from alcohol
substance abuse	using substance	

(vii) Optional: feel free to conduct additional analysis in addition to these listed above.

(viii) Based on the above analysis, suggest a potential solution to identify family history information of a patient in the format: (family member, disease). Discuss then pros and cons of your proposed solution.

5. **Guidelines and requirements**
   (i) Individual work only. We will use the anti-plagiarism feature on iLearn to detect potential plagiarism.
   (ii) Use one of the following programming languages: C, C++, Java, or Python.

6. **Submission instructions**
   (i) Pack the following items into one file with your name in the file name, e.g., JohnDavis-AssociationPatterns.zip:
      a. Source code with comments in the language of your choice.
      b. A brief description of the main steps that you have adopted in accomplishing this project.
      c. Instructions on compiling and running your program.
      d. Detailed discussion of the quality of the word association patterns, wordLists, the impact of the parameter *k,* and the recommended solution in step (viii).

   (ii) Submit the above file on iLearn. No late or e-mail submission will be accepted.