COMP150AIH: Artificial Intelligence in Health Informatics **Fall 2011**

Problem Set 2

Handed Out: October 7, 2011 *Due: October 19/21, 2011*

- Homework is due either physically, in class, October 19 or, if you'd prefer, electronically (via e-mail) by midnight on 21.
- We encourage you to discuss the homework with other members of the class. The goal of the homework is for you to learn the course material. However, you should write your own solution.
- Please keep your solution brief, clear, and legible.
- We encourage you to ask questions before class, after class, or via email. Please don't start e-mailing us questions the night before the homework is due, though.

In this homework, we're going to explore the steps in the biomedical text classification pipeline. You will write code to: fetch citation records from PubMed[1] (a repository of biomedical literature), process/encode this text for input to classification algorithms and, finally, experiment with different ways of classifying the data. The dataset we will work with is a corpus of biomedical citations included in systematic review, provided by Aaron Cohen: `http://www.cs.tufts.edu/comp/150AIH/hw2/cohen09.txt`. For each review (topic), the *target concept* here is whether a given citation is *relevant* (1, by convention) or *irrelevant* (-1). Thus the aim is to induce a model to automatically discern relevant from irrelevant citations.

The file format is tab-delimited; each line corresponds to TOPIC (i.e., the systematic review), the PubMed identifier (with which the citation can be retrieved from PubMed) and the STATUS (this is the label). For the latter, "Included" implies a relevant study (1),"Excluded" an irrelevant study.

1. First we need to download the citations comprising the dataset(s). We'll work with the 'AtypicalAntipsychotics' review. As you might have guessed, this review concerned studies investigating antipsychotic drugs.[2]

   We will fetch the corresponding citations from PubMed, using e-utills. E-Utils is very easy to use, and there is a lot of documentation out there,[3] or just Google around for E-Utils and PubMed. For any Pythonistas among you, I particularly recommend using BioPython to interface with PubMed; there is a very nice tutorial on how to do this.[4] The EFetch utility[5] will be of particular interest for our purposes here. You'll want to: 1) grab all of the pubmed IDs corresponding to the 'AtypicalAntipsychotics' review and then 2) write out the titles ('TI'), abstracts ('AB') and MeSH keywords ('MH') of

[1] http://www.ncbi.nlm.nih.gov/pubmed/

[2] The 'typical' refers to known side-effects of earlier such drugs, not found in the atypical variety. Somewhat ironically, the 'atypical' drugs are now quite typical.

[3] e.g., http://www.ncbi.nlm.nih.gov/books/NBK25501/

[4] http://biopython.org/DIST/docs/tutorial/Tutorial.html

[5] http://eutils.ncbi.nlm.nih.gov/corehtml/query/static/efetch_help.html

these citations to your local machine. Note that the latter (MeSH) is an ontology that Medline uses. They actually pay people to manually tag citations with these things!

2. Now we're going to encode the data fetched above. You'll need to write code to perform Bag-of-Words. There are libraries to do this for you, but please code it yourself. Include the option to perform tf-idf[6] as well as binary Bag-of-Words. Either e-mail us your code, or attach a print-out when you turn in your homework. We don't care what language you use.[7]

3. Encode the 'AtypicalAntipsychotics' abstracts, titles and MeSH separately for consumption by the machine learning library/tool of your choice. WEKA[8] is probably a good choice and will be the easiest for this task, but there other options (e.g., scikit-learn). Whatever library you use, the file formats are likely similar.

    (a) Perform cross-fold validation (WEKA will do this for you) over the abstracts (i.e., the Boolean BOW and TFIDF BOW representations) using two learning algorithms of your choice (we suggest SVM and Naive Bayes). Report which algorithms (and corresponding parameters, if any) you used, and output standard metrics (accuracy, sensitivity, specificity) for each algorithm. Interpret them: which algorithm performs better, in your opinion? What assumptions are you making? **Bonus:** Do this for a third learning algorithm – preferably something which is not a linear function.

    (b) Re-run the experiments over the title and MeSH representations of the documents. Is performance better or worse than it was over the abstracts? Which of these three 'feature-spaces' performs best? Does the poorer algorithm (as assessed above) given the best reprsentation out-perform the better algorithm given the worse representation?

4. How might you combine all three spaces into one? Implement this idea, and create a single model that somehow integrates all three sets of attributes. Re-run the experiments. Does the combined space beat the three feature-spaces individually?

[6] http://en.wikipedia.org/wiki/Tf-idf

[7] Go for something esoteric!

[8] http://www.cs.waikato.ac.nz/ml/weka/