CS380 Information Retrieval and Web Search

Lab 2

Hand-built corpus-based stemming rules

Monday, Feb 3

The purpose of stemming is to eliminate semantically meaningless variations in texts, thereby making them easier to find. Specifically, stemmers remove the ends of tokens. For example, a stemmer might make the following changes: animals to animal, higher to higher or sinking to sink. Generic stemmers (e.g. Porter) attempt to do this for every word in the language, one word at a time. Hence, they make mistakes. Corpus-based stemmers attempt the same task, but they use the texts which they are stemming control the stemming process. Hence, a corpus based stemmer should make fewer mistakes than a general stemmer. However, a corpus-based stemmer may be limited by the diversity of the corpus on which it is based. In this lab you will hand-write a very simple form of a corpus-based stemmer and explore its limitations.

Specific Tasks:
In this lab you will hand build a set of rules for stemming your corpus. Then you will apply those rules to a second corpus; answering the following questions:

1. What stemming rules did you create for the first corpus (and how many)?
2. How many unique tokens are there in the first corpus before and after applying your stemming rules? (The after can be an estimate.)
3. How many of those stemming rules could be applied correctly (without modification) on the second corpus? (See below for "second corpus").
4. How many of the rules form the first corpus were used on the second corpus?

5. How many rules needed corrections to work on the second corpus? Keep in mind that any corrections must still work on the first corpus.
6. How many new rules did you need?

What to turn in:

A single sheet (or 2), (handwritten is OK) with the name of the people who worked together and the answers to the above questions.

Process:

1. Form groups of 2-3 people (2 is better).
2. Select one member of the group to have the "first corpus" and the other to have the "second corpus".
3. With the first corpus, write (if needed) some code to produce a sorted list of all the words in the cleaned (ie text-only) corpus. You might find this UNIX helpful. For the file "TEMP" it does exactly that

   ```
   grep -o -E '\w+' TEMP | tr A-Z a-z | sort -u -f
   ```

   Also useful, the following UNIX command will append files (with names that match *.txt) into a single file named TEMP

   ```
   cat *.txt > TEMP
   ```

   (if you are not familiar with these UNIX commands, look them up)
4. Go though the sorted list from the first corpus word by word finding stemming opportunities. Write stemming rules. However, this is a corpus-based stemmer you are writing rules for, not a general stemmer. So, for instance, a general stemming rule might be:

   if a token ends in "s" and not "ss" remove the the "s"

   The equivalent corpus-based rule would be:

   if a token (t) ends in "s" then remove the "s" only if there is another token (u), such that after removing the "s" from token t it is identical to token u.
5. Take up to 50 minutes for step 3.
6. In the final 30 minutes, work with the sorted list from the second corpus answer the questions posed above.