Natural Language Processing – Assignment I

Reykjavik University – School of Computer Science

September 2009

1 Exercise I – 24%

Use *grep* or *egrep* to display information from the corpus *eng.sent*[1] that match certain patterns. This corpus contains one English sentence per line where each word is tagged with a part-of-speech (PoS) tag. The tagset used is the Penn Treebank tagset (see `http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html`).

For each item below, show the exact command that you use. Note: Chapter 2.3.4 in the text book might help as well as the *man* page for grep or egrep.

1. Display all sentences that include the exact word forms "German" and "car" (not necessarily adjacent, but in that order).
2. Display all sentences that include the multiword expression "in order to".
3. Display the number of sentences that include the exact word form "tennis".
4. Display the exact matched string (not the whole line) that include the word forms "tennis" and "at" (not necessarily adjacent, but in that order).
5. Display plural English nouns (not proper nouns) of length 12.
6. Display all words of length 9 that end in "ation".
7. Display all sentences that start with a Wh-pronoun.

[1] This is a Reuter's corpus, available at `http://www.ru.is/faculty/hrafn/Data/eng.zip`. Due to copyright reasons, please make sure you do not distribute this corpus.

8. Display all sentences that include the exact word form "offer", either as a verb or as a noun, and either preceeded by a word tagged as "DT" or a word tagged as "TO".

2 Exercise II – 24%

In this exercise, you are only allowed to use the following regular expression operators or meta-characters:

```
.  *  + ?  | ^ ( ) [ ]
```

Write regular expression patterns which match all strings ...

1. of length 4, starting with a or b, followed by any two letters in the middle, and ending with a or b.
2. of length 2, starting with a and followed by any other letter except x, y or z.
3. of length 3, containing only numerical letters.
4. from the input alphabet $\Sigma = \{a, b\}$, containing the substring *abba*.
5. from the input alphabet $\Sigma = \{a, b\}$, in which no two adjacent letters are the same (e.g., *abab* is in the language, but not *abba*). ϵ is in the language.
6. from the input alphabet $\Sigma = \{a, b\}$, starting and ending in the same letter. ϵ is not in the language.

3 Exercise III – 18%

1. Draw a finite-state automaton, a **DFA**, which corresponds to the regular expression $(ab)^*c$
2. Draw a finite-state automaton, a **DFA**, accepting all the strings (as well as ϵ) from the input alphabet $\Sigma = \{0, 1\}$, except those containing the substring *001*.
3. Draw a finite-state automaton, an **NFA**, with as few states as possible, which corresponds to the regular expression $a|b^*|(a|b)^*a$

4 Exercise IV – 34%

1. (14%) Write a Perl program, *findLongest.pl*, accepting a text file as an argument. The program finds the longest word, that only contains lower case Engish letters[2], in the file and prints it out along with its length. Note that the program should work for any text file, regardless of its format (i.e. whether the file has one word per line or multiple words per line). To invoke the program the user should type in:

   ```
   perl findLongest.pl <filename>
   ```

   - Test your program on the corpus *eng.sent* and return your program code along with the output of your program when running against this corpus.

2. (20%) Write a Perl program, *tagFrequency.pl*, accepting a tagged corpus as an argument (one word/tag per line) and a number t. The program prints out, in descending order, the t tags occuring most often in the corpus. Hint: Consult, for example, `http://www.perlfect.com/articles/sorting.shtml` for a discussion on Perl sorting.

   - Test your program on the corpus *eng.train* (which is included in the eng.zip file) and return your program code along with the output of your program when running against this corpus using the number *10*, i.e.

     ```
     perl tagFrequency.pl eng.train 10
     ```

5 Due date

This assignment is due on Monday, September 28th, at 23:59.

[2] Use a regular expression for enforcing this condition.