Winter 2017 (Jan9-Apr10)
Faculty of Computer Science

CSCI 4152/6509 — Natural Language Processing

Assignment 1

Due: *by midnight, Monday, Jan 30, 2017*
Worth: 50 marks
Instructor: Vlado Keselj, CS bldg 432, 902.494.2893, vlado@dnlp.ca
TA: Magdalena Jankowska, jankowsk@cs.dal.ca

Assignment Instructions:

The submission process for Assignment 1 and later assignments will be based on the SVN system. We assume here that you finished Lab 1, so you are already familiar with SVN (i.e., the Subversion version control system).

Brief SVN instructions: To submit this assignment, you should start with having a new working copy of your SVN course repository in your course directory. It should be a different copy than the copies you used in the Lab 1. Provided that you finished Lab 1 properly, you could even delete the directories 'primary' and 'secondary' from the lab, with their contents.

You can create a new working copy of your SVN course repository using the following commands on bluenose. You can login to bluenose and we assume here that you have already created the directory '`csci6509`' or '`csci4152`' depending on your course number. You should go into that directory using the command:

```
cd csci6509
```

or

```
cd csci4152
```

using the version of the directory according to your course number.

You can create a fresh, checked-out copy of your SVN course repository using the following command:

```
svn co https://svn.cs.dal.ca/nlp-course/CSUSERID
```

where CSUSERID is replaced with your actual CS userid, a.k.a., CSID.

As a subdirectory of this directory, you must create and add a directory named `a1`, which will contain your answers to Assignment 1. You can do this using the following commands:

```
cd CSUSERID
```

```
svn mkdir a1
svn commit -m'a1 created'
cd a1
```

All files with answers must be added to SVN. Do not forget to commit them before the deadline. You can commit the files many times, since the newest files before the deadline will be used.

To clarify the terminology: when we say *submit*, it means that the file must be added to SVN (command `svn add`), and committed to the repository (command `svn commit`).

1) (10 marks) Complete the SVN lab tutorial as instructed in the Lab 1 and the lab notes.

For this question, you do not need to submit anything. Your work will be directly checked using the directory used for the lab.

2) (10 marks) By solving this question, you will make sure that you completed the instructions for preparing the 'a1' assignment directory for submission of the assignment solutions. You should first follow the steps given in the assignment instructions for creation of the SVN assignment directory:

1. Login to your bluenose account and go to the directory 'csci6509' or 'csci4152', depending on the course you are registered in.
2. Make sure that you have a checked out version of your SVN course repository with your CSUSERID as the directory name.
3. Go to your CSUSERID directory and make sure that you have the directory 'a1' for assignment solution files, and that it is added to the SVN.
4. Go to the directory 'a1'.

In addition to above instructions, you need to do the folowing: Create a file named '`a1q2.txt`' and add some simple content in this file; e.g., just the line '`a1q2.txt created!`'. Add this file to SVN and submit it to the SVN repository.

3) (10 marks) List the levels of NLP and briefly describe all of them (1–3 sentences for each). Submit your answer as a plain-text file called '`a1q3.txt`'.

4) (10 marks) Write and submit a Perl program named `a1q4.pl` which reads the standard input and prints out content of all header tags such as: `<H1>...</H1>`, `<h2>...</h2>`, and so on. In other words, it should print out the content of all HTML tags starting with the letter 'H' or 'h' and followed by one or more digits.

The tags should be matched in a case-insensitive way, however when printing, you should always print an uppercase ‘H’. You should match only the cases where opening and closing tags are on the same line. It is possible to have multiple tags in the same line, but you can assume that they do not overlap. You should also assume that an opening or closing tag may contain additional attributes such as `<h4 attr=1>` but those will not be printed in the output.

The program must not print any unspecified output; it should simply read the standard input (using the `<>` operator), and produce tags and their content in the standard output, one tag per line.

For example, for the following sample input file:

```
This is <h1>a test</h1>.
This line has no tags.

<H2>Multiple tags</h2> in a <h3>Line.</h3>
<h4 atr=1>Attributes included</h4>
and <h5a>not this one</h5a>
<h1>so</h1>
<h1 > on.... </H1>
```

the output should be:

```
<H1>a test</H1>
<H2>Multiple tags</H2>
<H3>Line.</H3>
<H4>Attributes included</H4>
<H1>so</H1>
<H1> on.... </H1>
```

Some sample input and output files are posted on the web in the assignment directory.

5) (10 marks) Consider a DFA (Deterministic Finite Automaton) represented by the following transition function:

δ	a	b	other letters
$\rightarrow q_0$	q_1	q_2	q_2
q_1	q_1	q_3	q_1
q_2	q_2	q_2	q_2
q_3	q_4	q_3	q_3
$F : q_4$	q_4	q_3	q_3

The alphabet are all lowercase letters $\Sigma = \{a, b, \ldots, z\}$ and the set of states is $Q = \{q_0, q_1, q_2, q_3, q_4\}$. The start state is q_0, which is indicated with an arrow in the above table, there is one accepting (or final) state q_4, which is indicated with the 'F :' in the above table, and the transition function is given in the above table.

a) (5 marks) Represent this DFA as a graph, as done in class. Remember that the states are represented as graph nodes, the transition function as graph edges, the initial state with an arrow, and the final states with double circles. The graph must be submitted as a PDF file, or an image file such as JPG, and it must be named `a1q5.pdf`, `a1q5.jpg`, or similar. You can also draw the graph on a paper, scan it or take a photo of it, and then upload the scanned image.

b) (5 marks) Briefly explain what words this DFA accepts. Submit your answer in a plain text file named `a1q5.txt`.