CS909: 2016-2017

<u>Exercise one:</u> Regular expressions, Segmentation & Edit distance, FSA & FST, N-grams and language models.

<u>Submission</u>: 12 pm Thursday March 2nd 2017

<u>Notes:</u>

a) **This exercise will contribute towards 15% of your overall mark.**
b) **Submission should include one or more iPython notebooks and a report, where requested, as a .zip.**

Preparation: Getting to know Python

Practice using the iPython notebooks from the module website.

<u>Part A: Regular expressions (20 marks)</u>

1. Using Python, write regular expressions to capture the following sets of strings and a function that will match a regex against a string to evaluate it. Sets of regular expressions to evaluate:
    a. the set of all alphabetic strings;
    b. the set of all lower case alphabetic strings ending in a s;
    c. the set of all strings with two consecutive repeated words (e.g., "Mirror mirror" and "the the" but not "the bang" or "the big bang");
    d. all strings that start at the beginning of the line with an integer and that end at the end of the line with a word;
    e. all strings that have both the word "like" and the word "duck" in them (but not, e.g., words such as "likes" that merely contain the word "like";
    f. write a pattern that places the first word of a sentence in the complete words of Shakespeare in a file. Deal with punctuation.

By "word", we mean an alphabetic string separated from other words by whitespace, any relevant punctuation, line breaks, and so forth. Show in your iPython notebook your work in debugging these regular expressions by illustrating examples that match or don't match the patterns. (10 marks)

2. Use the ART Corpus as introduced in the Seminars. Write a Python Program that will (a) recognize chemical compounds (e.g. [Fe(Rtpen)(η1-OOH)]2+, CH3CH2). Examples of chemical compounds can be found in paper b103844n_mode2.Andrew.xml, and you can also use these expressions to evaluate your chemical compound regular expressions; (b) identify which papers contain chemical compound expressions and how many sentences in each paper contain a chemical expression; (c) Identify which of the 11 CoreSC categories most chemical expressions appear in. (10 marks)

<u>Part B: Minimum Edit distance (25 marks)</u>

1. Compute the minimum edit distance (using insertion cost 1, deletion cost 1,

substitution cost 2) of "refa" to "fear". Show your work (using the edit distance grid). What are the possible alignments? (pen and paper exercise) (5 marks)

2. Figure out whether "drive" is closer to "brief" or to "divers" and what the edit distance is to each. (use the above assumption regarding cost, deletion, substitution. Pen and paper exercise) (5 marks)

3. Now implement a minimum edit distance algorithm in Python (making the same assumption about costs for the three operations) and use your hand-computed results to check your code. (10 marks)

4. Expand the minimum edit distance algorithm you have written to output an alignment. For this you will need to store pointers and add a stage to compute the backtrace. (5 marks)

Part C: Morphology & FSTs (10 marks)

Write a finite state transducer for the consonant doubling spelling rule for single syllable verbs in English. The rule should accept "stop", "stops", "stopped" and "stopping" but not "aimming" or "aimmed". (pen and paper exercise) (10 marks)

Below is a link explaining the consonant doubling rule:
http://speakspeak.com/resources/english-grammar-rules/english-spelling-rules/double-consonant-ed-ing

Note: If you enjoy this exercise and are interested in how you can write an FST that will automatically parse text visit the following link for a manual on using the Xerox FST tools:
http://web.stanford.edu/~laurik/fsmbook/home.html

Part D: N-grams and language models (30 marks)

1. We are given the following corpus, modified from the one in lecture 6:

<s> I am Sam </s>
<s> Sam I am </s>
<s> I am Sam </s>
<s> I do not like green eggs and Sam </s>

If we use linear interpolation smoothing between a maximum-likelihood bigram model and a maximum-likelihood unigram model with lambda1 = ½ and lambda2=½, what is P(Sam|am)? Include <s> and </s> in your counts just like any other token. (3 marks)

2. Write a program to compute unsmoothed unigrams and bigrams. (10 marks)
3. Run your N-gram program on the ART corpus. Don't consider any of the XML tags apart from <s></s> (sentence boundaries). What are the most common unigrams and bigrams? Give a list of the 50 most common in each case. (10marks)
4. Add an option to your program to generate random sentences. (7 marks)