Objectives

The objectives of this assignment are to:

* Write a program to find *n*-gram statistics
* Compute the probability of a sentence
* Write a short report of 1 to 2 pages on the assignment
* Know what a language model is
* Optionally read a short article on the importance of corpora

Organization and location

The first lab session will take place on

* Group 1: Wednesday, September 9 from 10:15 to 12:00 in the Alpha room
* Group 2: Wednesday, September 9 from 10:15 to 12:00 in the Beta room
* Group 3: Wednesday, September 9 from 13:15 to 15:00 in the Beta room

You can work alone or collaborate with another student:

* Each group will have to write Java programs to count unigrams, bigrams, and trigrams in a corpus of approximately one million words and to determine the probability of a sentence.
* You will probably need to read the [Regular expression tutorial](https://docs.oracle.com/javase/tutorial/essential/regex/) for Java
* You can test you regular expression using the [regex101.com](https://regex101.com/) site
* Each student will have to write a short report of one to two pages and comment briefly the results. In your report, you must produce the tabulated results of your analysis as described below.

Programming

Collecting a corpus

1. Collect a corpus of at least 750,000 words. You will check the number of words using the Unix command wc -w.
2. Alternatively, you can retrieve a corpus of novels by Selma Lagerlöf from this URL:<http://fileadmin.cs.lth.se/cs/Education/EDAN20/corpus/Selma.txt>.
3. Run the [concordance program](https://github.com/pnugues/ilppp/tree/master/programs/java/src/lppp/ch02) to print the lines containing a specific word, for instance *Nils*.
4. Run the [tokenization program](https://github.com/pnugues/ilppp/tree/master/programs/java/src/lppp/ch05) on your corpus and count the words using the Unix sort and uniq commands.

Normalizing a corpus

1. Write a program to insert <s> and </s> tags to delimit sentences. You can start from the tokenization and modify it. Use a simple heuristics such as: a sentence starts with a capital letter and ends with a period. Estimate roughly the accuracy of your program.
2. Modify your program to remove the punctuation signs and set all the text in lower case letters.
3. The result should be a normalized text without punctuation signs where all the sentences are delimited with <s> and </s>tags.
4. The five last lines of the text should look like this:
5. <s> hon hade fått större kärlek av sina föräldrar än någon annan han visste och sådan kärlek måste vändas i välsignelse </s>
6. <s> när prästen sa detta kom alla människor att se bort mot klara gulla och de förundrade sig över vad de såg </s>
7. <s> prästens ord tycktes redan ha gått i uppfyllelse </s>
8. <s> där stod klara fina gulleborg ifrån skrolycka hon som var uppkallad efter själva solen vid sina föräldrars grav och lyste som en förklarad </s>
9. <s> hon var likaså vacker som den söndagen då hon gick till kyrkan i den röda klänningen om inte vackrare </s>

Counting unigrams and bigrams

1. Read and try programs to compute the frequency of unigrams and bigrams of the training set: [[Program folder](https://github.com/pnugues/ilppp/tree/master/programs/java/src/lppp/ch05)].
2. What is the possible number of bigrams and their real number? Explain why such a difference. What would be the possible number of 4-grams.
3. Propose a solution to cope with bigrams unseen in the corpus. This topic will be discussed during the lab session.

Computing the likelihood of a sentence

1. Write a program to compute a sentence's probability using unigrams. You may find useful hash tables that we saw in the mutual information program: [[Program folder](https://github.com/pnugues/ilppp/tree/master/programs/java/src/lppp/ch05)].
2. Write a program to compute the sentence probability using bigrams.
3. Select five sentences in your test set and run your programs on them.
4. Tabulate your results as in the examples below with the sentence *Det var en gång en katt som hette Nils*:
5. Unigrams
6. =====================================================
7. wi C(wi) #words P(wi)
8. =====================================================
9. det 22086 1086845 0.0203212049556284
10. var 12852 1086845 0.0118250532504635
11. en 13921 1086845 0.0128086341658654
12. gång 1332 1086845 0.00122556574304524
13. en 13921 1086845 0.0128086341658654
14. katt 15 1086845 1.38014160252842e-05
15. som 16790 1086845 0.0154483850043014
16. hette 107 1086845 9.84501009803606e-05
17. nils 84 1086845 7.72879297415915e-05
18. </s> 62283 1086845 0.057306239620185
19. =====================================================
20. Prob. unigrams: 4.49191263644087e-27 Entropy rate: 8.75247286930358 Perplexity: 431.277568080082
21. Bigrams
22. =====================================================
23. wi wi+1 Ci,i+1 C(i) P(wi+1|wi)
24. =====================================================
25. <s> det 5913 62283 0.0949376234285439
26. det var 4023 22086 0.182151589242054
27. var en 753 12852 0.0585901027077498
28. en gång 695 13921 0.0499245743840241
29. gång en 23 1332 0.0172672672672673
30. en katt 5 13921 0.000359169599885066
31. katt som 2 15 0.133333333333333
32. som hette 50 16790 0.00297796307325789
33. hette nils 0 107 \*backoff: 7.72879297415915e-05
34. nils </s> 2 84 0.0238095238095238
35. =====================================================
36. Prob. bigrams: 2.29220556082587e-19 Entropy rate: 6.19198973746961 Perplexity: 73.1096397351286

Complement

As a complement, you can read a paper by [Church](http://researcher.watson.ibm.com/researcher/view.php?person=us-kwchurch) and Hanks, [Word Association Norms, Mutual Information, and Lexicography](http://www.aclweb.org/anthology/J/J90/J90-1003.pdf), Computational Linguistics, 16(1):22-29, 1990, as well as another one on backoff by Brants et al. (2007) [Large language models in machine translation](http://www.aclweb.org/anthology/D07-1090.pdf).