Due date: Tuesday, December 4, 2018
Late submission: 25% per day.
Teams: COMP 472 students can do the assignment individually or in teams of 2.
COMP 6721 students must do the assignment individually.
Teams must submit only 1 copy of the assignment.
Purpose: The purpose of this assignment is to make you experiment with language models.

Automatic Language Identification

In this project you will build and experiment with a probabilistic language identification system to identify the language of a sentence.

Your Task

You will build a character-based n-gram model for 3 languages: English, French and a language of your choice that uses the same character set. As a basic setup, you must:

- Train a unigram and bigram character-based language model for each language.
- Use your language models to identify the most probable language of a sentence given as input.

Training Set: As training set, you will start with the corpora available on Moodle (2 books in English, and 2 books in French, where diacritics have been removed). For the 3rd language, you must find your own training corpora. The Web is a great place to find electronic texts. Look at the Web page of Project Gutenberg[1] or Internet Archive[2] for good starting points.

Character Set: Make sure that all 3 languages use the same character set. In particular, you should not take diacritics (accents, cedillas...) into account. The French corpora on Moodle have been cleaned of diacritics. For the basic setup, use only the 26 letters of the alphabet (i.e. ignore punctuations and other characters) and reduce all letters to their lower case versions.

Log Space: In order to avoid arithmetic underflow, remember to work in log space. Any logarithmic base would work; but for the basic setup, use base 10.

Smoothing: For the basic setup, both your unigram and your bigram models must be smoothed using add-delta, with delta=0.5.

Programming Environment:

You can use Java, C, C++ or Python. If you wish to use another language, please check with me first.

Input:

Your program will take as input:

- 3 file names (`trainFR.txt, trainEN.txt` and `trainOT.txt`) containing the training texts for each language, and
- a file containing the sentences to be classified.

[1] http://www.gutenberg.org/
[2] https://archive.org/

Output: Your program will output:

1. a dump of the language models in the following files:
`unigramFR.txt, bigramFR.txt, unigramEN.txt, bigramEN.txt, unigramOT.txt, bigramOT.txt`

Each file should contain a dump of the language model in list format. For example,

for the unigram models:

```
(a) = 7.9834e-005 // some arbitrary value used
(b) = 7.9834e-005 // everywhere in the handout
(c) = 7.9834e-005
..
(z) = 7.9834e-005
```

for the bigrams models:

```
(a|a) = 7.9834e-005
(b|a) = 7.9834e-005
(c|a) = 7.9834e-005
..
(z|z) = 7.9834e-005
```

2. For each input sentence in the input file:
   a. on the console, the most probable language of the sentence
   b. a dump of the trace of that sentence in a file name `out#.txt` where # is the number of the sentence in the input file.
   Each output file must contain the sentence itself, a trace and the result of its classification, following the format below.
   For example, if the input file contains 30 sentences to test, then you should generate 30 output files named `out1.txt` to `out30.txt`. If the first sentence is *What will the Japanese economy be like next year?* then `out1.txt` will contain:

```
What will the Japanese economy be like next year?

UNIGRAM MODEL:

UNIGRAM: w
FRENCH: P(w) = 7.9834e-005  ==> log prob of sentence so far: 7.9834e-005
ENGLISH: P(w) = 7.9834e-005  ==> log prob of sentence so far: 7.9834e-005
OTHER: P(w) = 7.9834e-005  ==> log prob of sentence so far: 7.9834e-005

UNIGRAM: h
FRENCH: P(h) = 7.9834e-005  ==> log prob of sentence so far: 7.9834e-005
ENGLISH: P(h) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005
OTHER: P(h) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005
…
FRENCH: P(r) = 7.9834e-005  ==> log prob of sentence so far: 7.9834e-005
ENGLISH: P(r) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005
OTHER: P(r) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005

According to the unigram model, the sentence is in English
----------------
BIGRAM MODEL:

BIGRAM: wh
FRENCH: P(h|w) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005
ENGLISH: P(h|w) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005
OTHER: P(h|w) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005

BIGRAM: ha
FRENCH: P(a|h) = 7.9834e-005 ==> log prob of sentence so far: 7.9834e-005
…
According to the unigram model, the sentence is in English
```

3. Submit the output of your program with 30 input sentences. Your 30 sentences **must include:**
   a. **the following 10 sentences**
      1) *What will the Japanese economy be like next year? (EN)*
      2) *She asked him if he was a student at this school. (EN)*
      3) *I'm OK. (EN)*
      4) *Birds build nests. (EN)*
      5) *I hate AI. (EN)*
      6) *L'oiseau vole. (FR)*
      7) *Woody Allen parle. (FR)*
      8) *Est-ce que l'arbitre est la? (FR)*
      9) *Cette phrase est en anglais. (FR)*
      10) *J'aime l'IA. (FR)*
   b. 10 sentences that your system classifies correctly
   c. 10 sentences that your system gets wrong

Experiments:
As with the previous mini-projects, you are expected to perform additional experimentations with your program beyond the basic setup specified above.
Examples of experiments include: experimenting with a variety of values for n, for delta or for the character set, a variety of languages (similar or different), etc. For each experiment, report and analyse your results in your report.

Deliverables:
The submission of the project will consist of 3 deliverables:
1. The source code and an executable that runs in the lab machines
2. The 30 output files when your language model is trained with the corpora given
3. A Report
4. A Demo

The Code:
Submit all files necessary to run your code in addition to a `README.txt` which will contain specific instructions how to run your program on the desktops in the computer labs.

The Report:
Your deliverable must be accompanied by a written report (~4-5pages) that:
- indicates which 3rd language you chose and why
- reports and analyses the results of the basic setup
- states, justifies and analyses the results of your experiments.

Please note that your report must be analytical. This means that in addition to stating the facts, you should also analyse them (e.g. explain why something behaves this or that way, in which case something would be better than another...).

The report must be:
- in PDF format and
- must be called:
  - `472_Report3_StudentID1_StudentID2.pdf` (for team work) or
  - `472_Report3_StudentID.pdf` (for individual work in COMP 472) or
  - `6721_Report3_StudentID.pdf` (in COMP 6721)

The Demo:
All submissions will be demoed during the lab time on the lab machines. You will not be able to demo on your laptop. Regardless of the demo time, you will demo the program that was uploaded as the official submission on or before the due date. The schedule of the demos will be posted on Moodle.

Evaluation Scheme:

	COMP 472	COMP 6721
Implementation (functionality, design, programming style, ...) & Demo (clear answers to questions, knowledge of the program, ...)	50%	45%
Report (clarity and conciseness, depth of the analysis, presentation, grammar,...) & Experimentations (thoroughness, originality,...)	50%	55%
Total	100%	100%

Submission:
The source code, an executable that runs in the lab machines, the output files and the report must be handed-in electronically by midnight on the due date.

1. Create one zip file, containing all necessary files for your program and your report. Remember that your report must be in PDF and must be named as indicated in the section "The Report" above
2. Name your zip file:
   - `472_Project3_StudentID1_StudentID2.pdf` (for team work) or
   - `472_Project3_StudentID.pdf` (for individual work in COMP 472) or
   - `6721_Project3_StudentID.pdf` (in COMP 6721)
3. Upload your zip file at: https://fis.encs.concordia.ca/eas/ as `project3`.

In addition to the electronic submission, please submit a paper version of your report.
If your report is ready by 11:45am on Monday Dec. 4, please bring it to class; otherwise, slip it under my office door (EV 3.117) before the deadline.

om pret te hê! *("Have fun!" in Afrikaans according to Google Translate)*