COMP 472 **Project 2 Adjusted for School Closing** **Winter 2020**

Due Dates Deliverable 1: March 29 + Deliverable 2: April 13
Late Submissions 30% per day per late deliverable
Teams You can do the project individually or in teams of 3.
Teams must submit only 1 copy of the project via the team leader's account.

In this project, you will build and experiment with Naive Bayes classification for natural language processing.

Download the Project 2 dataset available on Moodle. This dataset contains 2 files: a training set and a test set of tweets (in utf-8 character set) labelled with one of 6 languages:

1. Basque (eu)
2. Catalan (ca)
3. Galician (gl)
4. Spanish (es)
5. English (en)
6. Portuguese (pt)

In this project you will build and experiment with Naive Bayes classification to determine the most likely language of a tweet written in one of these languages. The project will be divided into 2 deliverables:

Deliverable 1: Build a variety of models to identify the language of tweets, and report their performance on the initial test set (the one available on Moodle).

Deliverable 2: Evaluate and analyze the results of the above models on a new test set that will be given to you at demo time. This involves analysis, report writing. ~~and an oral presentation.~~

The timeline will be the following:

1. Deliverable 1: Submit your code and the output files for the initial test set available on Moodle.
2. Do your demo and submit the output files for the new test set given at the demo time.
3. Deliverable 2: Submit the report & the slides of the presentation.
4. Do your oral presentation.

1 Deliverable 1: Building the Models

Write a Python program to build a variety of Naive Bayes classifiers using the training set available on Moodle. To avoid arithmetic underflow, you should work in log_{10} space.

1.1 The Input

Your models will all be based on the same Naive Bayes classifier, but will differ only on the hyperparameters used. Your program should therefore take as input the following hyperparameters:

1. an integer, V, indicating the vocabulary to use (see Section 1.2.1)
2. an integer, n, indicating the size of the n-grams to be used as features (see Section 1.2.2)
3. a real, δ, indicating the smoothing value to be used for additive smoothing (see Section 1.2.3)
4. the name of a file to use for training.
5. the name of a file to use for testing.

1.2 The Hyper-Parameters

1.2.1 Vocabulary (V)

Your program should work at the character level and should account for 3 types of vocabulary:

V	Significance
0	Fold the corpus to lowercase and use only the 26 letters of the alphabet [a-z]
1	Distinguish up and low cases and use only the 26 letters of the alphabet [a-z, A-Z]
2	Distinguish up and low cases and use all characters accepted by the built-in `isalpha()` method

1.2.2 Size of n-grams (n)

Your program should work with 3 types of character-based n-grams as features:

n	Significance
1	character unigrams
2	character bigrams
3	character trigrams

Note that any character not in the vocabulary should be ignored, but their existence should not. This means that when creating bigrams and trigrams, the out-of-vocabulary characters should not be part of the n-grams, but the preceding and following characters should not be considered consecutive. For example, given the string `abc*def`, where `*` is an out-of-vocabulary character, your program should create:

6 unigrams: `a`, `b`, `c`, `d`, `e`, `f` 4 bigrams: `ab`, `bc`, `de`, `ef` 2 trigrams: `abc`, `def`

1.2.3 Smoothing value (δ)

The 3^{rd} parameter read in input will indicate the δ value to be used for additive smoothing (add-δ). The value of δ will be a real in the interval $[0 \ldots 1]$.

1.3 BYOM: Build Your Own Model

Use the results of the previous models to inspire you to create a new model, that you feel will outperform all others. To do so, your program can take additional parameters as input. It is important to remember that you should never look at the test set to develop your model; hence optimising your model for the initial test set available on Moodle may not lead to good results with the test set given at demo time.

1.4 Programming Environment

To program the project, you must use Python 3.7 and run on the lab computers. In addition, you must use GitHub (make sure your project is private while developing).

1.5 The Output

The output of Deliverable 1 will consist of a trace file (see Section 1.5.1) and one overall evaluation file (see Section 1.5.2).

1.5.1 Trace File

Given a test set, your program should create a trace file called either `trace_myModel.txt` or `trace_V_n_d.txt`, where `V` is the value of V taken as input, `n` is the value of n and `d` is the δ value. The trace file should contain:

1. the tweet ID as indicated in the test file, followed by 2 spaces
2. the most likely class as determined by your model (i.e. the label `eu`, `ca`, `gl`, `es`, `en` or `pt`), followed by 2 spaces
3. the score of the most likely class (in scientific notation), followed by 2 spaces
4. the correct class as indicated in the test file, followed by 2 spaces
5. the label `correct` or `wrong` (depending on the case), followed by a carriage return.

For example the file `trace_0_1_0.1.txt` (V=0, n=1, δ = 0.1) could contain:

```
439376185941049344␣␣eu␣␣-1.23E-7␣␣ca␣␣wrong
439381208234196992␣␣es␣␣-3.21E-7␣␣es␣␣correct
```

1.5.2 Overall Evaluation File

In addition to the trace file, create a text file called either `eval_myModel.txt` or `eval_V_n_d.txt` summarising the performance of the model with the initial test set given on Moodle. The file should indicate the model's:

1. accuracy (Acc), carriage return
2. per-class precision (eu-P, ca-P, gl-P, es-P, en-P and pt-P) separated by 2 spaces, then a carriage return
3. per-class recall (eu-R, ca-R, gl-R, es-R, en-R and pt-R) separated by 2 spaces, then a carriage return,
4. per-class F1-measure (eu-F, ca-F, gl-F, es-F, en-F and pt-F) separated by 2 spaces, then a carriage return,
5. macro-F1 and weighed-average-F1 separated by 2 spaces

For example the file `trace_1_2_0.5.txt` could contain:

```
.6666
.7777␣␣0.5555␣␣0.4444␣␣0.3333␣␣0.2222␣␣0.1111
.7777␣␣0.5555␣␣0.4444␣␣0.3333␣␣0.2222␣␣0.1111
.7777␣␣0.5555␣␣0.4444␣␣0.3333␣␣0.2222␣␣0.1111
0.8888␣␣0.9999
```

1.6 The Demo

Your submission for deliverable 1 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.

No special preparation is necessary for the demo (no slides or prepared speech). Your program will be expected to read a new test file, and output its results (see Section 1.5). The results of your program will need to be uploaded on EAS during your demo. In addition, your TA will ask you questions on your code, and you will need to answer him/her satisfactorily.

2 Deliverable 2: Analysis

In order for your analysis to include the results of your models with the test set used at the demo, it will be due as Deliverable 2, after the demos.

2.1 Report

The report will be used to describe your own model and analyse the results of the models. The intended audience of your report is me (your prof) and your TAs. Hence there is no need to explain the theory behind the models. Your report should focus on **your** work and the comparison of the performance of the models when the hyper-parameters are modified. Your report should be 4-6 pages (without references and appendices) and use the template provided on Moodle. The report should contain at least the following:

- ☐ $\frac{1}{2}$ to 1 page: Introduction and technical details.
- ☐ $\frac{1}{2}$ to 1 page: An analysis of the initial dataset given on Moodle, and the one given at the demo time. If there is anything particular about these datasets that might have an impact on the performance of some models, explain it.
- ☐ $\frac{1}{2}$ to 1 page: A motivation and description of your model. Explain its hyper-parameters and why you chose them.
- ☐ 2 to 3 pages: An analysis of the results of all the models with the demo-test set and the initial test set. In particular, compare and contrast the performance of each model with one another, and with the initial and demo test sets. Please note that your report must be analytical. This means that in addition to stating the facts (e.g. the macro-F1 has this value), you should also analyse them (i.e. explain why some metric seems more appropriate than another, or why your model did not do as well as expected with the test set given at the demo ...). Tables and graphs would be very welcome here. A confusion matrix would be a great tool for the analysis.
- ☐ $\frac{1}{2}$ page: In the case of team work, a description of the responsibilities and contributions of each team member.
- ☐ Your report should have a reference section (not included in the page count) that properly cites all relevant resources that you have consulted (books, Web sites ...), even if it was just to inspire you. Failure to properly cite your references constitutes plagiarism and will be reported.
- ☐ Use appendices (not included in the page count), if you wish to show additional tables or graphs.

The report must:

- ☐ follow the Word or LaTeX template provided on Moodle.
- ☐ be submitted in PDF format
- ☐ be called `472_Project2_Report_ID1_ID2_ID3.pdf` where `ID1` is the ID of the team leader.

~~2.2 Oral Presentation~~

~~The results of your analysis will also be presented orally in a 3 minute talk plus 2 minute question period. The presentation will be done in front of the prof and TA - not in front of the whole class. All members of the team must be present and ready to answer questions.~~

~~You must prepare slides (which will essentially be a summary of your report) and upload them on EAS prior to your presentation (see Section 4). The slides must:~~

- ☐ ~~be submitted in PDF format~~
- ☐ ~~be called `472_Project2_Slides_ID1_ID2_ID3.pdf` where `ID1` is the ID of the team leader.~~

3 Evaluation Scheme

Students in teams can be assigned different grades based on their individual contribution to project. Individual grades will be based on a peer-evaluation done after the deadline.

The team grade will be based on:

Deliverable 1 - Code	functionality, design, programming style, ...		15%
Deliverable 1 - Output with initial dataset	correctness and format		15%
Deliverable 1 - Demo	clear answers to questions, knowledge of the program, ...		5%
Deliverable 1 - Output with demo-dataset	correctness and format		15%
Deliverable 2 - Report	depth of the analysis, originality, motivation, clarity and conciseness, presentation, grammar, ...	~~40%~~	50%
~~Deliverable 2 - Oral Presentation~~	~~clarity and conciseness, depth of the content, presentation, grammar, clear answers to question, knowledge of the work, presentation skills (time management, voice), ...~~		~~10%~~
Total			100%

4 Submission

4.1 Submission Schedule

Each deliverable is due on the date indicated below.

Deliverable	**Due Date**	**Upload as**
Deliverable 1 - submit your code and output files by	March 29, 2020, midnight	Project 10
Deliverable 1 - do your demo and submit the output files by	March 30 - April 3, during the lab	Project 11
Deliverable 2 - submit your report ~~and your slides~~ by	April 13, 2020, midnight	Project 12
~~Deliverable 2 - do your oral presentation by~~	~~April 14 - 15~~	

If you work in a team, identify one member as the team leader. The only additional responsibility of the team leader is to upload all required files (including the files at the demo) from her/his account and book the demo on the Moodle scheduler. If you work individually, by definition, you are the team leader of your one-person team.

4.2 Submission Checklist

In your GitHub project, include a `README.md` file that contains:

1. on its first line: the URL of your GitHub repository,
2. specific and complete instructions on how to run your program on the desktops in the computer labs.

Deliverable 1 - Code & Output files

- ☐ Create one zip file containing all your code, the output files for the initial test set on Moodle and the `README.me` file.
- ☐ Name your zip file: `472_Project2_D1_ID1_ID2_ID3.zip` where `ID1` is the ID of the team leader.
- ☐ Have the team leader upload the zip file at: https://fis.encs.concordia.ca/eas/ as `Project10`.

Deliverable 1 - Demo & Outut files During your actual demo with the TA:

- ☐ Generate the output files for the test set that the TA will give you.
- ☐ Create a zip file called: `472_Project2_Demo_ID1_ID2_ID3` where `ID1` is the ID of the team leader.
- ☐ Have the team leader upload the zip file at: https://fis.encs.concordia.ca/eas/ as `Project11`.
- ☐ In addition, a few days after the deadline, make your GitHub repository public.

Deliverable 2 - Report ~~and Oral Presentation~~

- ☐ **Name your report `472_Project2_Report_ID1_ID2_ID3.pdf` where `ID1` is the ID of the team leader.**
- ☐ ~~**Name your slides `472_Project2_Slides_ID1_ID2_ID3.pdf` where `ID1` is the ID of the team leader.**~~
- ☐ **Create a zip file called: `472_Project2_D2_ID1_ID2_ID3` where `ID1` is the ID of the team leader.**
- ☐ **Have the team leader upload the zip file at: https://fis.encs.concordia.ca/eas/ as `Project12`.**
- ☐ ~~**Print your report and your slides and submit a paper copy in the appropriate assignment box in EV 3.177.**~~

ondo pasa!