Due Date December 8, 2020
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.

Covid-19 Fact Checking

In this project, you will build a Naive Bayes bag-of-Word (NB-BOW) approach to determine if a tweet contains a verifiable factual claim. Then, you will compare the output of the NB model to a Word2Vec LSTM approach given to you (see Section 3).

1 The Dataset

Download the Assignment 3 dataset available on Moodle. This dataset, created by [Alam et al., 2020], contains a collection of tweets related to Covid-19 collected from March 9–10, 2020 and March 20–25, 2020. These tweets have been labeled for 7 different questions, such as:

Q1 Does the tweet contain a verifiable factual claim?
Q2 To what extent does the tweet appear to contain false information?
...
Q6 Is the tweet harmful for society and why?

For example, an instance of the dataset contains:

tweet_id	text	q1_label	q2_label	q3_label	q4_label	q5_label	q6_label	q7_label
1240716889162018816	Can y'all please just follow the government's instructions so we can knock this COVID-19 out and be done?! I feel like a kindergartner that keeps losing more recess time because one or two kids can't follow directions.	no	NA	NA	NA	NA	no_not_harmful	no_not_interesting

In this assignment, we will only use the classification of the 1st question (q1_label) whose labels are binary (yes/no):

1. `yes` – the tweet contains a verifiable factual claim
2. `no` – the tweet does not contain a verifiable factual claim

The dataset is already split into a training set of 400 instances and a test set of 55 instances.

2 The Naive Bayes Classifier (NB-BOW)

You will code a Multinomial Naive Bayes classifier yourself.

2.1 Parameters

Your Naive Bayes Classifier should use the following parameters:

Vocabulary: First fold the training set in lower case, then build a list of all words appearing in the training set. This list will constitute your vocabulary V which will be used as features. To identify the words, tokenise the tweets based on spaces only, use the words as features, and word frequencies as feature values.

Experiment with 2 versions of the model:

Original Vocabulary : one model where all words appearing in the training set are used as features. Let's call this model `NB-BOW-OV`.

Filtered Vocabulary : a second model where you filter out the words that appear only once in the training set, so V contains only the words that appear at least 2 times in the training set. Let's call this model `NB-BOW-FV`.

Smoothing: to smooth, use additive smoothing (add-δ) with $\delta = 0.01$.

Log: To avoid arithmetic underflow, work in log_{10} space.

2.2 Output

For both your models (NB-BOW-OV & NB-BOW-FV), your program should create 2 output files: a trace file (see Section 2.2.1) and one overall evaluation file (see Section 2.2.2).

2.2.1 Trace Files

Given a test set, your program should create trace files called `trace_NB-BOW-OV.txt` and `trace_NB-BOW-FV.txt`, 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 `yes`, `no`), 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_NB-BOW-0.txt` could contain:

```
1235714668833828864 ␣␣yes␣␣-1.23E-7␣␣no␣␣wrong
1235545254347984897 ␣␣no␣␣-3.21E-7␣␣no␣␣correct
```

2.2.2 Overall Evaluation Files

In addition to the trace file, create text files called `eval_NB-BOW-OV.txt` and `eval_NB-BOW-FV.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 (yes-P, no-P) separated by 2 spaces, then a carriage return
3. per-class recall (yes-R, no-R) separated by 2 spaces, then a carriage return,
4. per-class F1-measure (yes-F, no-F) separated by 2 spaces, then a carriage return,

For example the file `eval_NB-BOW-OV.txt` could contain:

```
.6666
.7777␣␣0.5555
.7777␣␣0.5555
.7777␣␣0.5555
```

2.3 Programming Environment

You must use Python 3.8. In addition, you must use GitHub (make sure your project is private while developing).

3 The LSTM Classifier (LSTM-W2V)

Your wonderful TAs have implemented a classifier for the same task using an LSTM and Word2Vec embeddings (LSTM-W2V). This code generates the output files (see Section 4.1) for you, so it can be used to compare the performance of your NB-BOW approach.

1. Download the embedding #6 from the http://vectors.nlpl.eu/repository. Note that the download is 606MB.
2. Download the code available at https://gitlab.com/Feasinde/lstm-for-covid-disinformation.
3. Place both downloads in the same folder.
4. Run the code and generate the corresponding output files. Note that the code may take a good 5 to 10 minutes to run.

4 Deliverables

The submission of the assignment will consist of 3 deliverables:

1. The code & output files
2. The demo (8 min presentation & Q/A)

4.1 The Code & Output files

Submit all files necessary to run your code in addition to a `readme.md` which will contain specific and complete instructions on how to run your experiments. You do not need to submit the datasets. If the instructions in your readme file do not work, are incomplete or a file is missing, you will not be given the benefit of the doubt.

Generate one output file for each model as indicated in Section 4.1.

4.2 The Demos

You will have to demo your assignment for $\approx$ 12 minutes. Regardless of the demo time, you will demo the program that was uploaded as the official submission. The schedule of the demos will be posted on Moodle. The demos will consist in 2 parts: a presentation $\approx$ 8 minutes and a Q/A part ($\approx$ 4 minutes). Note that the demos will be recorded.

4.2.1 The Presentation

Prepare an 8-minute presentation to analyse and compare the performance of your models. The intended audience of your presentation is your TAs. Hence there is no need to explain the theory behind the models. Your presentation should focus on **your** work and the comparison of the performance of 3 models.

Your presentation should contain at least the following:

- ☐ An analysis of the initial dataset given on Moodle. If there is anything particular about these datasets that might have an impact on the performance of some models, explain it.
- ☐ An analysis of the difference between the vocabulary of the NB-BOW-OV and NB-BOW-FV models. What is the size of V in each model? did the reduction in V lead to a significant difference in performance? Explain.
- ☐ An analysis of the results of all 3 models. In particular, compare and contrast the performance of each model with one another.
- ☐ In the case of team work, a description of the responsibilities and contributions of each team member.

Please note that your presentation must be analytical. This means that in addition to stating the facts (e.g. the 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. Tables, graphs and contingency tables to back up your claims would be very welcome here.

Any material used for the presentation (slides, ...) must be uploaded on EAS before the due date.

4.2.2 Q/A

After your presentation, your TA will proceed with a $\approx$ 4 minute question period. Each student will be asked questions on the code/assignment, and he/she will be required to answer the TA satisfactorily. In particular, each member should know what each parameter that you experimented with represent and their effect on the performance. Hence every member of team is expected to attend the demo.

In addition, your TA may give you a new dataset and ask you to train or run your models on this dataset. The output files generated by your program will have to be uploaded on EAS during your demo.

5 Evaluation Scheme

Students in teams can be assigned different grades based on their individual contribution to project.

Individual grades will be based on:

1. a peer-evaluation done after the submission.
2. the contribution of each student as indicated on GitHub.
3. the Q/A of each student during the demo.

The team grade will be based on:

Code	functionality, proper use of the datasets, design, programming style, ...	11
Output with initial datasets	correctness and format	1.5
Demo – Presentation	depth of the analysis, clarity and conciseness, presentation, time-management, ...	4
Demo – QA	correct and clear answers to questions, knowledge of the program, ...	2
Output with demo-dataset	correctness and format	1.5
Total		20

6 Submission

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.

6.1 Submission Schedule

Each deliverable is due on the date indicated below.

Deliverable	**Due Date**	**Upload as**
Submit your code, output files, presentation material	December 8, 2020, 2020, 11:59pm	Assignment 3
Submit the output files generated at demo time	during your demo	Assignment 6

6.2 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.

Code & Output files

- ☐ Create one zip file containing all your code, the output files for the initial test set on Moodle and the `README.md` file.
- ☐ Name your zip file: `472_Assignment3_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 `Assignment3`.

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

- ☐ Prior to your demo, make your GitHub repository public.
- ☐ Generate the output files for the test set that the TA will give you.
- ☐ Create a zip file called: `472_Demo1_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 `Assignment3`.

Have fun!

References

[Alam et al., 2020] Alam, F., Shaar, S., Dalvi, F., Sajjad, H., Nikolov, A., Mubarak, H., Martino, G. D. S., Abdelali, A., Durrani, N., Darwish, K., and Nakov, P. (2020). Fighting the covid-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. https://arxiv.org/abs/2005.00033.