Due Date Saturday Oct 11, 2024, 11:59pm
Late Submissions 30% per day per late deliverable
Teams Assignments can be done individually or in teams of 2.

1 Task

1.1 Base Implementation

In this assignment, you will learn how to implement your Word2Vec model using a CBOW method and use these embeddings for a sentiment analysis task. This notebook will guide you through the implementation process, from getting the data to creating your prediction algorithm.

1.2 Experiments

Once you have completed the notebook and have the base implementation running, follow the instructions indicated in the last section called *Let's Experiment*. In this section, 4 experiments are proposed. Pick 3 experiments of your choice and implement them in your colab code. Make sure that you can run your experiments during the demo, show their results and be ready to discuss them.

2 Evaluation

Your assignment will be evaluated in 2 parts: your code, and your demo.

2.1 Code

The base code will be worth 30% and each experiments 10%, for a total of 60%($30\% + 3 \times 10\%$). Your code does not need to contain excessive documentation, but should be easy to follow and efficient. Each experiment should be easy to understand and run and the results/output should be displayed directly in the notebook.

- ☐ Save your code in `.ipynb` format.
- ☐ If you work alone,
  - ☐ name the file: `COMP6781-A1-YourLastName-YourFirstName-YourID.ipynb`
  - ☐ upload the file on Moodle.
- ☐ If you work in a team,
  - ☐ name the file: `COMP6781-A1-LastName1-FirstName1-ID1-LastName2-FirstName2-ID2.ipynb`
  - ☐ upload the file on Moodle from `FirstName1`'s account.

2.2 Demo

When the schedule of the demos will on be available on Moodle, book a slot for your team. All members of the team must attend the demo, as each team member may be attributed a different grade. The demos will be in person, but will be recorded.

You will have to demo your assignment for ≈ 25minutes. Regardless of the demo time, you will demo the program that was uploaded as the official submission on the due date. The demos will consist of 2 parts: a presentation ≈ 5 minutes and a Q/A part (≈ 20 minutes).

2.2.1 Presentation

Prepare an 5-minute presentation to explain your base implementation and your 3 experiments. Analyse and compare the performance of your models. The intended audience is your marker (i.e. a person knowledgeable in NLP, but who does not know your specific code). Hence there is no need to explain the theory behind the models. Your presentation should focus on **your** work and the analysis of your experiments.

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 results of all 3 experiments. 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 analyze 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.

2.2.2 QA

After your presentation, your TA will proceed with a question period. Each student will be asked individual questions on the assignment. In particular, each member should be knowledgeable on all aspects of the assignment. An answer of the type *I didn't work on that part* will not be accepted. In addition, your TA may give you a new dataset and ask you to train or run your models on this dataset. The output file generated by your program will have to be uploaded on Moodle before leaving the demo room.

General Marking Scheme

Below are the general expectations of your work in order to receive a grade in a specific range:

Bugs in base code + incomplete experiments + vague presentation + unconvincing QA	0%...40%
Working base code + minimal experiments + adequate presentation + adequate QA	55%...70%
Working base code + interesting experiments + strong analysis + clear QA	75%...90%
Working and efficient base code + extensive experiments + very strong analysis + clear QA	90%...100%