

CS3216/AI5203
Assignment 3- NLP

Deadline: 7th april, 2024

Time: 11:59 PM

Total Marks: 100

Instructions: -

- Please don't copy from the internet or any other student.
- Plagiarism will thoroughly be checked for all submissions.
 - Refer to the policy on [Academic Unfair means](#).
- You may use any libraries **but it should be done in pytorch**. No other frameworks will be allowed
- Allowed Programming Language: Python
- It is recommended to use Jupyter Notebook/Google colab
- **Submission link:** <https://forms.gle/tNaCfcxoQALxxDpEA>
- The format of the submission file should be .zip format: **RollNo-A3.zip. No other format is accepted other than .zip**
 - The uploaded zip should contain:
 - The code in a step-by-step format with the results of each stage (**.ipynb file**)
 - Readable comments for each stage (**.ipynb file**)
 - A **.ReadMe file** having instructions on how to execute the code, implementation details, path of directories, and analysis/observations of the results
 - All trained models stored inside a folder called **"models"**
 - Dataset stored inside a folder called **"data"**

Problem Statement: Training a RNN model on the given dataset.

Dataset link: <https://ai.stanford.edu/~amaas/data/sentiment/>

Implementation:

[40 marks]

1. Implement the [Word2vec](#) model and train the word vectors using skip-gram model with negative sampling.
2. Implement the [FastText model](#) and train the word vectors [1].

Hint: Make use of only “train” folder for training your word vectors.

3. You can use “test” folder and sentiment labels, i.e., pos and neg for your sentiment classification task using RNN.
4. After creating word vectors using the methods provided above, train your RNN model on the sentiment classification task by making using of these word vectors.

Results and analysis

[25 marks]

Present the results of your experiments including performance metrics for each word vector technique used for sentiment classification task.

5. Make use of tables, graphs to compare results visually.
6. Discuss any findings and report all the hyperparameters for each technique used during experimentation.

Grading policy: Individual evaluations based on submission and viva along with code review. Expectations to explain the code, results, and analysis. Performance during viva carries 35 marks.

References:

[1] <https://github.com/facebookresearch/fastText>

Additional resources:

1. https://d2l.ai/chapter_recurrent-neural-networks/rnn-scratch.html#transforming-rnn-outputs
2. <https://colah.github.io/posts/2014-07-NLP-RNNs-Representations/>

-----All the best-----