M21AdvNLP: Advanced NLP **Monsoon 2021**

Assignment 2

Strict Deadline : 7-11-2021, 23:55 Hrs

Instructor: Dr. Manish Shrivastava *TA: Ananya Mukherjee, Manikanta Sai Nuthi*

1 General Instructions

1. The assignment can be implemented in Python.
2. Ensure that the submitted assignment is your original work. Please do not copy any part from any source including your friends, seniors, and/or the internet. If any such attempt is caught then serious actions including an F grade in the course is possible.
3. A single .zip file needs to be uploaded to the Moodle Course Portal.
4. Your grade will depend on the correctness of answers and output. In addition, due consideration will be given to the clarity and details of your answers and the legibility and structure of your code.
5. Late submissions will not be evaluated.

2 Problem Statement

Embeddings from Language Model (ELMo) is a powerful contextual embedding method that finds application in a wide range of Natural Language Processing tasks. ELMo, along with others, started the trend of pretraining contextual word embeddings in NLP. The technique remains simple and intuitive, allowing itself to be easily added to existing models.

The ultimate goal is to learn meaningful vectors (embeddings) forwards in natural language, such that semantically similar words have mathematically similar vectors. The current assignment aims to make you familiar with the above algorithms. In this assignment, you need to implement the following tasks :

1. Implement a Embedding from Language Model (ELMO) on the monolingual dataset provided. [10 marks]

   For the above implementation, report the following for the model:

   - Calculate euclidean distance for any 5 different pair of words. [5 marks]
   - Calculate cosine distance for any 5 different pair of words. [5 marks]

3 Training corpus

Please train your model on the following corpus:
https://www.isip.piconepress.com/projects/switchboard/releases/switchboard_word_alignments.tar.gz

4 Submission Format

Zip the following into one file and submit in the Moodle course portal:

1. Source Code
2. Pre-trained model and Embeddings generated
3. Report answering all the questions raised above.
4. Readme File :on how to execute the code, how to restore the pre-trained model. Any other information.

If the pre-trained models and embeddings cross the file size limits: upload them to a OneDrive/GDrive and share the links in the Readme File.

5 Grading

1. Evaluation will be individual and will be based on your viva, report, submitted code review.
2. In the slot you are expected to walk us through your code, explain your experiments, and report.

6 Queries

1. **Vocab size: Do we train for all the unique words that we see in the corpus?**
   You can place a limit: words with frequency less than 5 can be ignored. If you have enough compute train for as many words as possible.
2. **Do I need to ignore the stop words?**
   Usually stop words occur with very high frequency. You don't have to ignore the stop words. To avoid the effect of the words(stop words) which occur very frequently you can use sub sampling. This usually speeds up training.
3. **Do I need to use Lemmatisation before feeding the samples to the model?**
   No. For the sake of this assignment do no worry about Lemmatisation.

4. **I do not have enough compute. Can I reduce the vocab size ?**
   For reference, you can easily implement a model with vocab size of 4,00,000 words on a 8 GB RAM, decently sized CPU, home laptop. You can make use of Google Collab for implementing this, which offers greater ram and better CPUs.

5. **Can I submit my code in jupyter notebooks?**
   Yes , you can submit your code base in jupyter notebooks.

7 Reference Material

1. https://arxiv.org/abs/1802.05365

2. https://towardsdatascience.com/elmo-helps-to-further-improve-your-word-embeddings-c