CSC 495/693: Natural Language Processing, Spring 2024 Assignment 3

Assigned Tuesday, Apr 9, due Wednesday Apr 17. Max points: 100.

In this assignment, you will practice various NLP tasks with deep learning methods.

Programming Requirements

- Dataset1: You have been provided with 2,000 movie comments as the dataset "comments2k.zip" including 1,000 positive comments and 1,000 negative comments.
- Dataset2: You have been provided a translation dataset with 139,705 *English-Spanish* pairs.
- Programming Language: Please use Python3, not Python2.
- Coding Style: define a function for each question.

1. Sentiment Analysis (60 points)

Sentiment analysis is the process of analyzing text to determine the emotional tone. Use **all the 2,000 comments in Dataset1** as the corpus, and choose related python libraries to finish the following tasks.

1) In Assignment2, you developed a neural network classifier C1 with the pre-trained word embedding model, now develop a new neural network classifier C2 without loading pre-trained word embeddings. Compare C2 and C1, which one has better performance? Try to discuss it. (Requirements: you developed three classifiers in Assignment2, pick the best one as C1. Make sure C1 and C2 are trained and evaluated with the same training, validation and testing sets)
2) Train three classifiers with RNN, LSTM and GRU respectively under the same settings, and output the classification reports (tip: you can use *sklearn.metrics.classification_report*). Which classifier has better performance? Compare their time cost, which one is faster? (Requirement: Use ***Pytorch*** for this problem. Clearly specify the dataset division, hyperparameters and other required settings in your report)
3) Build a bidirectional 3-layer stacked LSTM model, compared to your LSTM model in the last question, which one has better classification results? Why?

2. Text Translation (40 points)

Machine translation is the process of automatically translating content from one language to another without any human input. Given the ***English-Spanish*** **dataset2**, build a machine translator using ***TensorFlow***. (The format of the dataset is English + TAB + Spanish + TAB + CC-BY License + Attribution. You can basically ignore the license and attribution, only focus on the columns of English and Spanish.)

1) Prepossess the dataset and split the dataset into training, validation and testing subsets by the ratio 70/15/15.
2) Train the translator with Transformer method, report the classification results. What does the hyperparameter "num_heads" mean? Why do we need this mechanism?
3) What is the translation result for the sentence "Deep Learning is widely used in Natural Language Processing, as Dr. Sun said in CSC 495/693." with your translator? Plot the heatmap of the encoder-decoder attention scores.

Writeup

Prepare a writeup on your experiments by using any of the following template:

- ACM (https://www.acm.org/publications/proceedings-template/)
- IEEE (https://www.ieee.org/conferences/publishing/templates.html)

Write down any further insights or observations you made while implementing and running the program. Especially interesting insights may be awarded extra points. You may also receive extra points for well-written code with clear comments and runs efficiently. Conversely, poorly written, or not following the ACM/IEEE format, or hard to understand and inefficient code will lose points.

What to turn in

You will turn in:

1. Your writeup, and
2. Your source code. You may include a readme if needed (e.g. if you wish to bring anything to my attention). Please ensure your code is well documented. **I will not be able to spend a lot of time debugging your code if it crashes during our testing.**

To turn in your code and writeup, use Canvas. Prepare a zip file with all your files and name it <yourname>_assign3.zip. **This zip file should only contain your writeup, source code and readme (if needed) and not executables/object files/data files/unmodified code/anything else, and must be timestamped by the due date to avoid a late penalty.**