Programming Project II – Transformers for NLP

CAP 5619, Deep & Reinforcement Learning (Spring 2024), Department of Computer Science, Florida State University

Points: 100
Maximum Team Size: 2
Due: 11:59pm on Tuesday, April 23rd, 2024

Submission: You need to submit electronically via Canvas by uploading a) a pdf file (named "**lab2Lastnames.pdf**") as your report (including analysis and experimental results), and b) the program(s) you have created (named as "**lab2-prog-Lastnames.???**"); if there are multiple program files, please zip them as a single archive. Here replace "Lastnames" using your last names of the group members in the file names. Only one submission is required for each group.

The main purpose of this assignment is to use the token embeddings and sentence representations given by a transformer model for solving the analog problem and also the sentiment analysis problem.

For this assignment, you can choose to use the embeddings and sentence representations given by either BERT or GPT; for each of the models, there are multiple versions, and any version is acceptable. Note that both of them use tokens, rather than words directly; in other words, you need to tokenize the words and if a word breaks up into more than one token, you need to use the average embeddings of the tokens as the embedding for the word.

Task I – Word Analogy Prediction Task

The dataset (available from https://www.cs.fsu.edu/~liux/courses/deepRL/assignments/word-test.v1.txt) contains 14 groups of word analogy relationships. You can choose any of the three groups to work with. For a given group, each line contains four words in the form of <a b c d>. The task is to predict the last word given the first three. For a given k, the prediction is correct of the actual last word is among the first k closest words. To make the prediction task more meaningful, we only consider the second and fourth words from the lines that belong to the same group as the candidates. One possible way to solve the problem is to compute the embeddings of a, b, c, and d, then compute a-b. Then sort all the candidate words based on the cosine similarity (or L2 distance) between a-b and c-d', where d' is one of the candidates. For each of the groups you choose, complete the following table:

k	Accuracy Using Cosine Similarity (larger is closer)	Accuracy Using L2 Distance (smaller is closer)
1		
2		
5		
10		
20		

Since you need to choose three groups, you should have three tables. Note that cosine similarity is a similarity measure (the closer two vectors are, the larger the cosine similarity between them) while L2 distance (Euclidean distance) is a distance measure (the closer two vectors are, the smaller the L2 distance between them).

Task II - Sentiment Analysis By Fine-tune a Pretrained Transformer Model

First, we need to split the samples (the dataset is available from https://www.cs.fsu.edu/~liux/courses/deepRL/assignments/amazon_reviews.csv) into a training set and a test set. We will use 80% of the samples for training and the remaining ones for testing. Then you need to use BERT or GPT to create a representation for each of the reviews. After that, you need to train a classifier using the samples in the train set and then use the trained classifier to classify the samples in the test set. Optionally, you can finetune BERT or GPT for sentiment classification by adding a classifier to BERT or GPT and finetune the entire model using the training set and evaluate on the test set. Since the dataset is relatively small, you should not over finetune the model. Note that there are five different ratings, 1.0, 2.0, 3.0, 4.0, and 5.0.

Extra Credit Options

(1) **Aspect Based Sentiment Analysis.** In this case, your program needs to identify the aspects and classify each aspect accordingly using the SemEval-2014 Task 4 dataset from https://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools. You can choose to work on the Laptops or the Restaurants dataset using the given training and test set.

(2) **Learning Rewards from Linguistic Feedback by Finetuning BERT or GPT.** As autonomous agents are being adopted, how such machines can learn from natural interactions with humans becomes very important. For this task, you will finetune BERT or GPT to understand human feedback as another inference network. The dataset is available from https://github.com/tsumers/rewards and more details about the experiments can be found in the paper available from https://arxiv.org/pdf/2009.14715.pdf. You need to report your results in the interaction sampling setting only; in other words, your results should be another row in Table 2 under the interaction sampling.

Grading

- **Report and analysis – 30 points**
  - Note that the focus is on understanding, and you have to provide meaningful/insightful analyses for what you expect from your experiments before doing them and explain what you have observed in your experiments accordingly.
- **Correct implementation and experimental results – 70 points**
  - **Task I – 30 points**
    - **The correctness of the algorithms – 15 points**
    - **The results and the tables for three groups - 15 points**
      - **5 points for each group**
  - **Task II – 40 points**
    - **The correctness of the method – 20 points**
    - **The results - 20 points**
- **Aspect Based Sentiment Analysis – 10 points**
- **Learning Rewards from Linguistic Feedback by Finetuning BERT or GPT - 10 points**

Additional information:

You can find BERT models from huggingface (https://huggingface.co/google-bert). For GPT models, they are available from https://huggingface.co/openai-community. I would suggest using a small model as the primary goal of the project is to gain understanding and smaller models are easier to train and work with.

For Task I, you need to tokenize each word using the tokenizer corresponding to BERT or GPT and then find their embeddings.

For Task II, it is easier to compute the sentence embedding for each review text first. Then you can train a neural network or any other classifier (such as a supporting vector machine). Finetuning a BERT or GPT should work better; please make sure you do not over finetune it on the training set.

The vocabulary of a BERT model is available from https://www.cs.fsu.edu/~liux/courses/deepRL/assignments/bert_vocab.txt and the corresponding token embeddings are available from https://www.cs.fsu.edu/~liux/courses/deepRL/bert_lm_word_embeddings.txt