

Assignment 2: Sentence Representations

CSE 538 Fall 2019
Released: Oct 15
Due: Oct 25 11:59pm

This assignment is to learn (i) how to train sentence representations for a classification task and (ii) how to probe these representations for analysis. You will also explore (i) how the data size and training time affects your model (ii) how to perform error-analysis on your trained models.

You will implement two models for sentence representations for a task: (i) Deep Averaging Network and (ii) GRU based Model. You will also implement a linear probing model to understand what kind of information the learnt representations capture. Most of the code for this is provided with instructions (in Readme) and you only need to fill-up the placeholders. See README for overview of the code and more assignment details. There are many scripts available in the provided code to help you with the analysis.

1 Getting Started

As well as reading this document, which covers the general tasks being asked of you, you should follow the README in the code repository. The README outlines each part of the code, file by file, as well as explains what should be turned in on submission. This assignment pdf serves only to give you an introduction to the assignment and what to put in your report, the details are in the README!

2 Model Implementation [50 pts]

You will implement two models for text classification:

- Deep Averaging Network: Please look at the details in the paper. Figure 1 and equations 5-8 summarizes the model. [20 pts]
- GRU based model: A GRU is a recurrent neural network that gates sequential information over time-steps. You have to use the built-in GRU layers from tensorflow rather than building from scratch. For a n layered GRU based model, you will stack n layers of gru layers unlike n layers

of feedforward network in DAN. Make sure to use the provided masks appropriately in the model. [20 pts]

Additionally, You will implement a linear probing model that takes vector representations from n^{th} layer of the underlying model and learns a linear classifier on these representations. For DAN, the probing vector of n^{th} layer is simply the average that you computed for that layer. For GRU, the proving vector is the last state of GRU for that layer. [10 pts]

3 Analysis [50 pts]

3.1 Learning Curves:

The provided code has scripts to generate plots for the following. Run them and explain in brief, what changes as we:

- Increasing the training data [7 pts]
- Increase training time (number of epochs) [7 pts]

For first use `plot_performance_against_data_size.py` and for second use `train_dan_for_long.sh` scripts.

3.2 Error Analysis:

- Explain one advantage of DAN over GRU and one advantage of GRU over DAN.[7 pts]
- Use the above to show and explain failure cases of GRU that DAN could get right and vice-versa. Use one of your trained models to do this analysis. [7 pts]

4 Probing Tasks

4.1 Probing Sentence Representation for Sentiment Task [7 pts]

Here, we are interested in what each layer of our models are capturing with respect to their ability to predict sentiment. The probing model you trained will be used for this. You will take the (frozen) weights from each layer of your model and then the probing model on top of that. Using that final model you will apply it to the sentiment task. In your report add in your observations for each layer and its accuracy(generated on its associated plot). Use `plot_probing_performances_on_sentiment_task.py` for this analysis.

4.2 Probing Sentence Representations for Bigram Order Task [8 pts]

In this task we are looking at a direct comparison of DAN vs GRU in their ability to predict order of words. After training both your models you will test them on the reversed bi-gram dataset supplied in the assignment. Add your plot of accuracy (results) and observations to your report.

Use `plot_probing_performances_on_bigram_order_task.py` for this analysis.

4.3 Analysing Perturbation Response of Representations [7 pts]

Lastly, we want you to track what happens to a specific input when it is run through your network. Each time it goes through you will change 1 word (the example sentence is in the README). Again, generate a plot then add it and your observations to the report.

Use `plot_perturbation_analysis.py` for this analysis.

5 Report

The write up should be 3 to 4 pages of length. You should have the following 3 sections corresponding to section 2-4 in this assignment:

1. Model Implementation: dan, gru and probing model. Brief (1 page max) description of how you implemented the models.
2. Analysis: Learning curves and Error Analysis.
3. Probing Tasks: plots and observations/explanations.

6 Resources

Here is a paper covering a review of RNN and GRU (as well as LSTM). You'll be most interested in sections 2 and 3.2. <https://arxiv.org/abs/1412.3555>

As well as a paper covering the details of Deep Averaging Networks. https://people.cs.umass.edu/miyyer/pubs/2015_aclan.pdf