Assignment 4: Experimenting BERT on AG news

COMP 551 Winter 2025, McGill University
Contact TAs: Charlotte Volk and Yutang Song

Released on March 27
Due on April 12 midnight

Preamble

- This assignment is **due on April 12 at 11:59pm (EST, Montreal Time).**
- For late submission, 2^k percent will be deducted per k days of the delay.
- For late submission, 2^k percent will be deducted per k days of the delay. Late submission requests are handled case by case by the lead TA, Huiliang Zhang. You may post private message on Ed with title "Late submission request for Assignment 1" and select the "Assignment" tag.
- This assignment is to be completed in groups of three. All members of a group will receive the same grade except when a group member is not responding or contributing to the assignment. If this is the case and there are major conflicts, please reach out to the contact TA or instructor for help and flag this in the submitted report. Please note that it is not expected that all team members will contribute equally. However every team member should make integral contributions to the assignment, be aware of the content of the submission and learn the full solution submitted. See more solutions on Frequently Asked Questions on Groups and Assignments
- You will submit your assignment on MyCourses as a group. You must register your group on MyCourses and any group member can submit. See MyCourses or here for details.
- We recommend to use **Overleaf** for writing your report and **Google colab** for coding and running the experiments. The latter also gives access to the required computational resources. Both platforms enable remote collaborations. If you require additional computing resources for pre-training or fine-tuning the LLM, you may use the GPU clusters. A guide to accessing the GPU clusters is available in the "GPU Tutorial" on MyCourses lecture recordings.

- You should use Python for this and all assignments. You are free to use libraries with general utilities, such as matplotlib, numpy and scipy for Python, unless stated otherwise in the description of the task. In particular, in most cases you should implement the models and evaluation functions yourself, which means you should not use pre-existing implementations of the algorithms or functions as found in SciKit learn, and other packages. The description will specify this in a per case basis.

Synopsis

In this assignment, you will implement a Large Language Model (LLM), namely Bidirectional Encoder Representations from Transformers (BERT), using the existing PyTorch libraries. You will evaluate your LLM on the Academic Group (AG) Corpus of New Article (AG news) dataset.

1 Task 1: AG News data

The AG News data are described here:

- `https://huggingface.co/datasets/fancyzhx/AG_News`

The data can be easliy obtained via Python as follows:

```
from datasets import load_dataset
train_datasets = load_dataset('ag_news', split='train')
test_dataset = load_dataset('ag_news', split='test')
```

The dataset consists of four news categories: World, Sports, Business, and Sci/Tech. Each example includes a news headline and a short description. You should use the default ‘train‘ (120,000 news articles) and ‘test‘ (7600 news articles) splits provided by the dataset.

For BERT, you will work directly with raw text input. No manual feature extraction is necessary, as the tokenizer and embedding layers of the model will handle text preprocessing internally. You may find this tutorial helpful:

- `https://www.kaggle.com/code/wesleyacheng/news-topic-classification-with-bert`

2 Task 2: Loading the pre-trained BERT

Install the `transformers` library:

```
pip install transformers
```

Load `bert-based-uncased` model (110M params):

```python
from transformers import BertTokenizer, BertModel

# Load tokenizer and model
tokenizer = BertTokenizer.from_pretrained("google-bert/bert-base-uncased")
model = BertModel.from_pretrained("google-bert/bert-base-uncased")
```

All text were lowercased before tokenization for pretraining the uncased BERT model. Therefore, the BERT was trained entirely on lowercased version of the corpus (e.g., Wikipedia). This is the most commonly used version of BERT. You may read more about the model from here: `https://huggingface.co/google-bert/bert-base-uncased`.

3 Task 3: Run experiments

3.1 Probing

Begin by passing each news document from the training set through the frozen BERT model and extracting sentence-level representations. You may experiment with different strategies for this, such as using the `[CLS]` token from the final hidden layer, taking the first or last token embedding, or computing the mean over all token embeddings (excluding padding tokens). Once these representations are extracted, you will use them as input features for K nearest neighbour (KNN) and to train a multi-class logistic regression classifier on the training documents.

For KNN, you may split the training into training and validation and choose the best K based on validation accuracy. Because KNN does not require training, this can be considered as a *zero-shot* application of the pre-trained BERT. You may use KNN and multi-class logistic regression from SciKit-learn for this task or those ones you implemented for assignments 1 and 2. Use the validation set to choose the best embedding strategies and report your findings.

3.2 Fine-tuning

In this task, you will fine-tune all parameters of a pretrained BERT model on the AG News training data for the classification task by allowing gradients to update all the internal weights of the transformer via stochastic gradient descent.

You do not need to fine-tune on the entire dataset or for a large number of epochs. Training for 2–3 epochs on a subset of the training data (e.g., 10k–20k examples) is sufficient, especially if you are using limited computational resources.

You may use a subset of the training data as validation set to stop the training when the validation accuracies start to decrease.

3.3 Reporting classification performances

Report the multiclass classification accuracies for probing using KNN and logistc regression and end-to-end fine-tuning on the AG News test data. Analyze the performance differences and reflect on how much improvement is gained through end-to-end training, and why this might be the case.

3.4 Attention matrix visualization

Examine the attention matrix between the words and the class tokens for some of the correctly and incorrectly predicted documents. You will need to choose one of transformer blocks and use a specific attention head for the multilayer multiheaded transformer architecture in your LLM. This package may be helpful for visualization of the attention matrices: `https://github.com/jessevig/bertviz`

You are welcome to perform any experiments and analyses that you see fit, but at a minimum, you must complete the above experiments.

Deliverables

You must submit two separate files to MyCourses (**using the exact filenames and file types outlined below**):

1. `assignment4_group-k.ipynb`: Your data processing, classification and evaluation code should be all in one single Jupyter Notebook. Your notebook should reproduce all the results in your reports. The TAs may run your notebook to confirm your reported findings.
2. `assignment4_group-k.pdf`: Your (max **8-page**) assignment write-up as a pdf (details below). Compared to the past 3 assignments (5-page each), we provide 3 extra pages for this assignment to provide you more space to explore.

where k is your group number.

Project write-up

Your team must submit a project write-up that is a maximum of eight pages (single-spaced, 11 pt font or larger; minimum 0.5 inch margins, an extra page for references/bibliographical content can be used). We highly recommend that students use LaTeX to complete their write-ups. You have some flexibility in how you report your results, but you must adhere to the following structure and minimum requirements:

Abstract (100-250 words) Summarize the project task and your most important findings. For example, include sentences like “In this project we investigated the performance of BERT base

model on AG News dataset.", "We found that BERT achieved worse/better accuracy than the more traditional ML methods and was significantly faster/slower to train. We achieved test accuracy of ??.???% using probing and ??.???% via end-to-end fine-tuning."

Introduction (5+ sentences) Summarize the project task, the AG News datasest, and your most important findings. This should be similar to the abstract but more detailed. You should include background information and citations to relevant work (e.g., other papers analyzing these datasets).

Datasets (5+ sentences) Briefly describe the AG News dataset and how you processed them specifically for BERT. Present the exploratory analysis you have done to understand the data.

Benchmark (10+ sentences) Clearly describe the settings for BERT (architecture, number of attention heads, number of transformer layers, etc), probing strategies, K in KNN, multi-class logistic regression, and fine-tuning strategies.

Results (5+ sentences corresponding to 5 figures) Describe the results of all the experiments mentioned in Task 3 as well as any other interesting results you find. At a minimum you must have these results:

1. A table that summarizes the validation accuracies for probing strategies for KNN (with different K values you have experimented) and multi-class logistic regression
2. A barplot displaying test accuracies for the best probing strategy and fine-tuned BERT
3. From one of the attention head in one of the layers from the fine-tuned BERT, display the attention between the top words and the [CLS] token for a correctly predicted positive and negative AG News document with high probability.
4. Same as above but now show the two attention heatmaps for the incorrectly predicted positive and negative AG News examples.

Discussion and Conclusion (5+ sentences) Summarize the key takeaways from the project and possibly directions for future investigation.

Statement of Contributions (1-3 sentences) State the breakdown of the workload across the team members.

Evaluation

The assignment is out of 100 points, and the evaluation breakdown is as follows:

- Completeness (20 points)
  - Did you submit all the materials?
  - Did you run all the required experiments?
  - Did you follow the guidelines for the project write-up?
- Correctness (40 points)
  - Are your models implemented correctly?
  - Are your reported accuracies close to the reference solutions?
  - Does your the RF important features make sense?
  - Do the attention heatmaps you choose to display make sense?
- Writing quality (25 points)
  - Is your report clear and free of grammatical errors and typos?
  - Did you go beyond the bare minimum requirements for the write-up (e.g., by including a discussion of related work in the introduction)?
  - Do you effectively present numerical results (e.g., via tables or figures)?
- Originality / creativity (15 points)
  - Did you go beyond the bare minimum requirements for the experiments?
  - **Note:** Simply adding in a random new experiment will not guarantee a high grade on this section! You should be thoughtful and organized in your report.

Final remarks

You are expected to display initiative, creativity, scientific rigour, critical thinking, and good communication skills. You don't need to restrict yourself to the requirements listed above - feel free to go beyond, and explore further. You can discuss methods and technical issues with members of other teams, but **you cannot share any code or data with other teams.**