CS 590 NLP

HW4

Transformers, HuggingFace

Due 12/09 11:59 pm

Introduction

Your final homework is slightly open ended, but the goal is to get practice with some pretrained models readily available on Huggingface (https://huggingface.co/).

Hugging Face provides access to many pretrained transformer models which can be useful for baseline comparison or even finetuning for accomplishing many tasks.

Hugging Face also provides easy access to datasets as well.

Homework Goal

Your goal in this homework is to find several pretrained classification models on huggingface which all focus on a similar task (e.g. sentiment classification IMDB, sarcasm detection, AG News classification) and compare them.

The comparison should involve using a built in hugging face dataset and testing the accuracies on at least 100 examples of the test set portion of that dataset.
You need to have comparisons between at least 3 different models. Not all datasets have 3 models available so you may instead choose 2 different datasets to examine each with 2 classification models to compare.

It is up to you which classification task you choose, but it should have models and some dataset available so check for both of these.
Note that it is okay if the models are not trained on that dataset but are in the same task (e.g., trained on sentiment classification but not on IMDB sentiment dataset).

The major goal here is to give you practice with the pretrained models so going forward in your other classes and future jobs you know how simple it can be to bring in these models as a starting point.

Hugging Face and Python

Python has a nice built in module called “transformers” which easily allows you to load the pretrained models and have them give classification scores:
https://huggingface.co/transformers/v3.0.2/index.html

Before getting started, I would suggest you just practice loading in a model from the documentation above and giving it queries.
If you are using a BERT based model you can use the BERT under documentation which gives examples. (same with other model types).
Otherwise, I have found that AutoModel has also worked well for me in the past.

Note that the ...ForSequenceClassification are the models you will be wanting to use. (e.g. https://huggingface.co/transformers/v3.0.2/model_doc/bert.html#bertforsequenceclassification)

If you are struggling to get started make sure to let me know so I can help.
However, if you wait until the last minute to start, I can’t guarantee that I will have time to help you!

Specific Requirements

1. Choose pretrained models to evaluate
    a. Choose 3 models which are all trained in the same classification task

    OR

    b. Choose 4 models, 2 trained on same task A and 2 trained on same task B
2. Choose relevant dataset to compare pretrained models on (each dataset should use at least 100 examples for comparison)
3. Write a simple report:
    a. Summarize accuracy of all models for their corresponding datasets
    b. Write out explicit comparisons and discussion of results (were there any surprises? Any possible bugs that could be causing problems?)
    c. Small section on limitations of your comparisons
    d. Small section on how you could use these pretrained models in the future if you were to do research in these directions
4. Hand in report and code. Note that your code should be following the guidelines discussed in class and posted online. **Points will be deducted for failure to meet these guidelines.**

Explicit Note

This (and all homeworks) are meant to be worked on by yourself. Since there is so much room for variation in this project, I should see little overlap on chosen models.

Note that any copying of homework or collaboration still falls into the academic dishonesty category and measures will be taken against you.

Additional Notes

1. The accuracy you achieve will not determine your final grade for this homework. Rather your thoughtfulness in approaching and analyzing your models. This means that lack in analysis will receive lower scores as indicated in the grading scale.
2. Note that preprocessing the text is up to you. This assignment is not specifically evaluating preprocessing like the first assignment, but here you have a chance to practice any preprocessing for your final project and future assignments. (**You should be doing some preprocessing, or else you'll be purposely setting your ML model up to fail.**)
3. Get familiar with the transformer library first before trying to start any final portions of the homework.
4. DO NOT SHARE CODE. I have linked transformer libraries which will be useful for this assignment. You may also come to me to discuss/figure things out.
5. Start the assignment early. You will have 2 weeks to complete this assignment, which is plenty of time if you start early any familiarize yourself with the libraries. **IF YOU WAIT UNTIL THE LAST MINUTE TO START, YOU WILL BE LESS LIKELY TO DO WELL.** I won't be granting any additional extensions so each late assignment with follow the late grade policy.
6. Ask questions/approach like a researcher. Think like this is your chance to explore NLP models and analyze their effectiveness.

Description	**Points**
Transformer Implementation/Tasks	20
Report analysis and observations	20
Documentation (Comments, functions, etc)	10
Total:	**50**
Extra Credit (fine-tuning tasks)	**20**

Extra Credit (Fine-tuning a bert-base model)

For extra credit and additional transformer understanding, you may fine-tune the bert-base model (https://huggingface.co/bert-base-uncased) on the OLID dataset which have been working with.

Note that fine-tuning requires a bit more work (**AND TIME)** than the homework assignment so make sure you start early since no Extra Credit is granted to late assignments.

(You may find https://huggingface.co/docs/transformers/training useful for learning some steps for fine-tuning). (**There are others as well, but you need to be using transformers and pytorch (NOT KERAS or other**)).

Your tasks for the EC are:

1. Using transformers and pytorch, fine-tune a bert-based-uncased model on the subtask A of the OLID dataset (OFF or NOT).
2. Produce predictions for the test portion of OLID.
3. Calculate accuracy.
4. In a **SEPARATE** report, report the results and the previous “simpler” models’ results from previous homeworks.
5. Compare and contrast the results for the models and provide explicit discussion.
6. Hand in your code, report, and prediction files. (The prediction files may be in the same format that previous homeworks had.)