CSCI 4820/5820 (Barbosa)

Project 5: A Named Entity Recognition Data Science Project Using BERT Fine tuning

Due: **See course calendar**. May not be turned in late.

Assignment ID: **proj5**
File(s) to be submitted: **proj5.ipynb, proj5doc.pdf proj5.pdf**

Automatically find names of people, places, products, and organizations in text

Note: Your code must run on the **TAMU FASTER** system – Ensure you test your code there before submission.

****** This project requires the **Spring 2024 SIF** on TAMU FASTER (see the video on D2L) ******

Objective(s): This is a project that is intended to resemble a task you may be assigned while working as an NLP data scientist. In this project you will augment a named entity recognition (NER) dataset with additional tags and then fine tune a BERT model for token classification using the resulting dataset using. You are provided starter code for this project and must modify that code to complete the task.

Project description: Your company is using an NER solution with a limited tag set (Person, Organization, Location, and Miscellaneous tags in addition to the No Tag label). Your manager has asked you to convert the Person tags so that male and female persons are identified separately. The manager is concerned about a potential drop in performance (since the number of Person samples would now be split into two classes). You are to make the necessary code changes to the dataset and to evaluate the performance of the final model and to report results, including any drop in performance as well as misclassifications.

Dataset description: The Conference on Natural Language Learning (CoNLL) 2003 dataset available on Hugging Face is used in this project, and is loaded and used to fine tune a BERT model in the starter code. It consists of approximately 20K samples in total, with around 3,500 each in the validation and test sets and the remainder as training samples. The dataset uses the BIO tag system: the **B** prefix (for *begin*) indicates the beginning of an entity of a specific type, the **I** prefix (for *inside*) is for the second and subsequent tags of entities composed of multiple tokens, and the **O** tag (for *outside*) is for tokens that are not of the covered entity types.

The tags in this dataset are:

- O is the outside tag
- B-PER and I-PER are tags applied to Person entities
- B-ORG and I-ORG are tags applied to Organization entities
- B-LOC and I-LOC are tags applied to Location entities
- B-MISC and I-MISC are tags applied to miscellaneous entities

As an example, the sentence *John Smith flew to New York City to tour the IBM facility* should be tagged as shown:

Tokens:	*John*	*Smith*	*flew*	*to*	*New*	*York*	*City*	*to*	*tour*	*the*	*IBM*	*facility*
Tags:	B-PER	I-PER	O	O	B-LOC	I-LOC	I-LOC	O	O	O	B-ORG	O

PER tags must be converted to **B-MPER/I-MPER and B-FPER/I-FPER for male and female persons respectively**.

Starter code description: The starter code pip installs required packages, loads the dataset from Hugging Face, loads a pretrained BERT model, and fine tunes the model for named entity token classification.

Constraints on starter code modifications: The starter code must be modified and cannot be completely replaced. This includes swapping the model for another model in the BERT family (like DistilBERT, RoBERTa, etc.). **The dataset for this project cannot be changed. It must be the one loaded by the starter code.**

Some ideas on coming up with true labels for gender specific tagging:

- Manual labeling – This provides the best quality but can be time consuming
  - Review the 6,600 *Person* labels and manually label then as male or female
  - Split the 6,600 *Person* labels among class members and each person labels part of the dataset
  - Use people outside of the class to help you label the data.
- Use a name to gender classifier – Many of these use a name's beginning and/or ending characters or character bigrams as features for classification. There are some good and some not-so-good classifiers for this task. Based on features used, this can be a good option that lowers accuracy but takes less time. **Note: If a classifier is used it must be installed/called/loaded in the submitted code and not in a separate code file**. The model's source must be documented/attributed.
- Use an algorithmic or rule-based approach to assign the new true labels and **include it in the submission**.

Hints on approaching the problem:

- To limit changes to tags in the original dataset (represented as an integer index) convert the B-PER/I-PER tags to B-MPER/I-MPER and add the new B-FPER/I-FPER label as new tags at the end of existing tags (the female person tags will get new, previously non-existent index values).
- Convert the dataset tags before tokenization – Recall that BERT utilizes subword tokenization.
- **The dataset must be preserved as a Hugging Face dataset**. This allows to code to be used as-is after tag conversion. Some hints on functions and methods to use in adding the new labels to the data (you will have to read the documentation on these to understand what they do):
  - *Sequence* from the *datasets* module
  - *ClassLabel* from the *datasets* module
  - **Note**: This dataset is not a gold standard dataset. It is labeled using rules, heuristics, and algorithms, and contains misclassified samples. Make a good effort at converting it, but do not try to fix every issue with it.

Requirements: Where two weights are shown for an item, the first is for CSCI 4820 and the second for CSCI 5820 credit.

1. (**85%/70%**) Your program must provide the functionality listed, in a single file named ***proj5.ipynb***:

   - Study the starter code and run it. Note: See troubleshooting steps in the last page if you run into problems.
   - **This is a research project - You may use resources at your disposal to make modifications but must document all sources. The quality of your code alterations and data presentation matter.**
   - Since this is a data science project, **extra credit can be earned** for good solutions that have good data exploration and results that use figures/plots, and clearly communicate the scope of the work done and the results obtained (consider also adding a colorful confusion matrix but ensure you verify its correctness).

2. (**10%**) Test your program changes as you make them **test your submission on the TAMU system** prior to turn-in.

3. (**5%**) Code comments - Add the following comments to your code:

   - Place a Markdown cell at the top of the source file(s) with the following identifying information:

     Your Name
     CSCI <course number>-<section number>
     Project #X
     Due: mm/dd/yy

   - Add a Markdown cell above each code cell that describes the processing done in the code cell.

4. (**CSCI 5820 Students only 15%**) Analysis – Provide a **detailed** analysis (in a Markdown cell) addressing the following:

   a) Discuss the benefits of using a pretrained model like BERT on this task. Focus on aspects relevant to this specific task rather than generalities: How does BERT pretraining contribute to this task.
   b) Provide a detailed description of an alternative way to solve this problem without fine tuning BERT.

5. Generate a pdf file of the notebook (from the terminal):

```
$ jupyter  nbconvert  --to  html  proj5.ipynb
$ wkhtmltopdf  proj5.html  proj5.pdf
```

6. Document all input and output from AI and other external tools and sources in the file ***proj5doc.pdf***

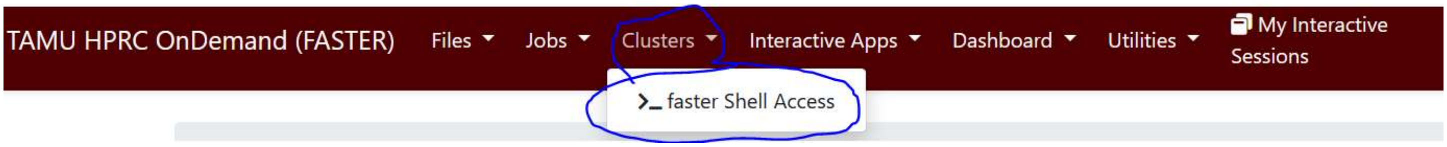
7. Submit required files via the D2L dropbox for this project.

TAMU FASTER Troubleshooting

****** This project requires the **Spring 2024 SIF** on TAMU FASTER ******

If you have installed packages on TAMU FASTER prior to this assignment you may encounter some errors regarding missing or incompatible packages. The notebook has been tested and works on the default container without problems.

To **revert to default container**, log on to TAMU and before starting a notebook instance, bring up a shell as shown below:



Once in the shell change to your home directory (using the ***cd*** command) and type the following command (note this command will remove all of the contents of the .local directory):

```
rm –fr .local/*
```

Then instantiate a notebook following the video instructions on D2L.

After running the pip install cell, an error may appear saying that the **datasets module** (just installed) was **not found**. Simply restart the notebook and re-run it.