Samuel Amoabeng (EID: sba798) Jacob B Wendling (EID: jbw2976)

PROJECT 4 REPORT

PART 1

INTRODUCTION

In this project, we will dive into the realm of healthcare where diagnosing patients accurately with rapid response is of prime importance to the healthcare system. In this project, we are developing a machine learning model and language processing to improve diagnosis, based on a patient's symptoms description. This system will use text classification, with a pre-trained language model. By understanding the subtleties of how patients describe their symptoms, the system can suggest possible diagnoses. The dataset has a “natural language description of symptoms labeled with 22 corresponding diagnoses". The fusion of human-like language comprehension and machine learning has the potential to transform healthcare practices, paving the way for enhanced patient care and improved medical outcomes in less time.

PROJECT STATEMENT

The primary objective of our is to develop a system that helps physicians diagnose disease faster and more accurately. We will use a type of machine learning model called a text classifier, starting from a pre-trained text model. The pre-trained model will be further trained on a dataset of patients' symptoms and diagnoses. This will allow the model to learn from some examples and make predictions for new symptoms it hasn't seen before. By carefully testing and improving the system, we hope to show that it can be a valuable tool for doctors, leading to better care for patients. By fine-tuning the model on this specialized dataset, we endeavor to equip it with the ability to generalize and accurately predict diagnoses for unseen symptom descriptions. Through meticulous experimentation and validation, we seek to demonstrate the efficacy and reliability of our approach in assisting healthcare professionals with timely and accurate diagnoses, thereby facilitating more efficient patient care.

DATA SOURCES AND TECHNOLOGIES USED

The dataset contains natural language descriptions of symptoms labeled with 22 corresponding diagnoses. We used the “gretelai/symptom_to_diagnosis” dataset, available on Hugging Face. This dataset contains 1065 data points of text input and output. The input text is a few sentences long description of a patient’s symptoms, while the output text is a term describing a diagnosis, each row contains a data field as stated in the example.

For example:

Input Text:

```
I've been having a lot of pain in my neck and back. I've also been
having trouble with my balance and coordination. I've been coughing
a lot and my limbs feel weak.
```

Output Text:

```
cervical spondylosis
```

We used the transformer architecture and recurrent neural networks (RNNs) to deal with the natural language processing portion of this task.

The transformer architecture converts words into tokens (numbers) that can be input into neural networks. They do this along with attention vectors that represent how different words are related to each other. The attention vectors also get trained with the neural network, which is the main innovation that makes this process possible. An RNN acts like a regular neural network, except that it takes the previous output as an input to help find patterns in ordered information. This allows us to process things like sequences and, in this case, sentences. The model uses both of these ideas to contrast similar symptoms by their specific ordering with other words. The

pre-trained model is trained using this technology on tens of thousands of different input words and is therefore great at understanding this dataset. By further training the model, it can focus on the most important parts of how patients describe their symptoms and can pick out the key terms. This allows our model to understand our dataset and make a better diagnosis of symptoms.
We combined RNNs with a Transformer model in our system to have a refined model system that will fit our dataset. The transformer understands the overall meaning of the dataset sentences, and the RNNs add extra detail on how the words are ordered. This helps to create a complete picture of what the diagnosis will be prescribed for a particular symptom.

METHODS EMPLOYED

We begin with preprocessing the data by converting the output text into 22 categories, then splitting the data into an 80-20 train-test split. Then as mentioned, we used a pre-trained transformer model called "distilbert-base-uncased" which is intended for predicting the next word in a sentence. We feed the preprocessed dataset into a pre-trained RNN and train the entire model further so that it can predict one of the 22 diagnoses on the input text.

This is all done using the HuggingFace Python libraries (“datasets,” “Transformers,” and “evaluate”). The models are evaluated using the “accuracy” metric, which is a simple true or false based on the correctness of the predictions.

We also aim to examine the viability of few-shot learning (FSL) in this context. FSL is the concept of giving the model a small number of data points to train with. When data is scarce, this is an important idea to think about. This tries to emulate the way humans learn in the real world. We will train the model using a decreasing amount of each diagnosis and compare the results.

RUNNING THE MODEL

The model can be found at the github repository found in the references. Running the model for yourself is simple.

1. Navigate to the ProjectFinal folder
2. Run the flask server using: `./run_flask.sh`
3. In a new terminal, run the following to submit symptoms:

```
./submit_text.py "<symptoms>”
```

For example:

```
./submit_text.py "I have itchy red spots on my skin”
{'diagnosis': 'chicken pox',
 'score': 0.9349539875984192}
```

In this case, the score is its confidence in its diagnosis out of 1.0.

RESULTS

The dataset begins with about 40 of each of the diagnoses. Below are the results of each case trained for 20 epochs. The number of data points per category can be referred to as the number of “shots” in few-shot learning. The full results can be found in the jupyter notebook.

Shots	Accuracy
5	0.358491
10	0.693396
20	0.910377
40 (Full Dataset)	0.948113

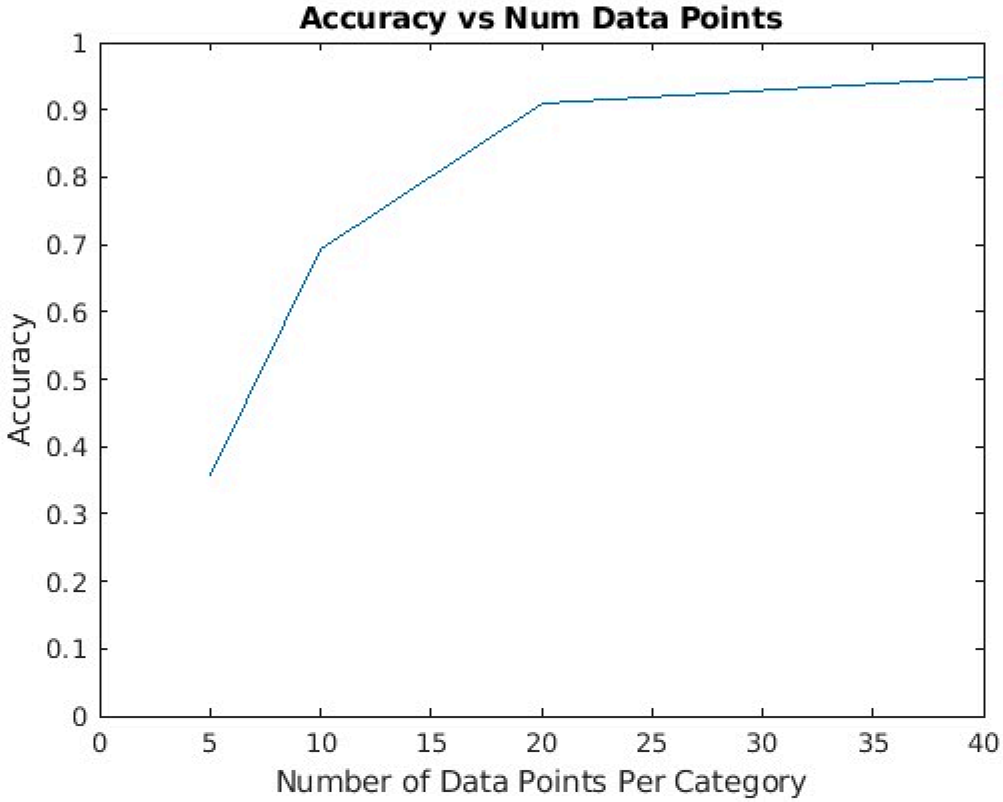

Figure 1: Graph of the Accuracy vs the number of shots

The model trained on the entire training dataset for 20 epochs reached an accuracy of 94.8%. When considering the number of shots the model was given, the model has a reasonable accuracy of 69% or greater with 10 shots or more. With less shots however, this model is unreliable. Using a more complex or general-purpose pre-trained language model, such as Chat-GPT, might have better success with this context.

This model should not be used in place of medical advice, as it will always diagnose symptoms with a diagnosis, even if the patient is healthy. Additionally, it does not consider all possible conditions a patient may have.

REFERENCES

Few shot learning: https://www.ibm.com/topics/few-shot-learning
BERT Pre-trained model: https://huggingface.co/distilbert/distilbert-base-uncased
Dataset used: https://huggingface.co/datasets/gretelai/symptom_to_diagnosis
Github repo for this project: https://github.com/Kwabena86/SW-DSIGN-FOR-RESP-COE379L