**Assignment 4**

Hruday Vishal Kanna Anand

S.No- 1006874517

**Q1.1** Context: Unlike the Spanish milled dollar the U.S. dollar is based upon a decimal system of values. In addition to the dollar the coinage act officially established monetary units of mill or one-thousandth of a dollar, cent or one-hundredth of a dollar, dime or one-tenth of a dollar, and eagle or ten dollars, with prescribed weights and composition of gold, silver, or copper for each. It was proposed in the mid-1800s that one hundred dollars be known as a union, but no union coins were ever struck and only patterns for the $50 half union exist. Question: What is the US dollar based upon?

**A-** The US dollar is based on a decimal system of values.

**Q1.2** Context: Age, diameter, height, radial growth, geographical location, site and growing conditions, silvicultural treatment, and seed source, all to some degree influence wood density. Variation is to be expected. Within an individual tree, the variation in wood density is often as great as or even greater than that between different trees (Timell 1986). Variation of specific gravity within the bole of a tree can occur in either the horizontal or vertical direction. Question: Which part of a tree can have vertical or horizontal variation in its specific gravity?

**A-** The bole of a tree can have vertical or horizontal variation in its specific gravity.

**Q1.3** Context: Canonical jurisprudential theory generally follows the principles of AristotelianThomistic legal philosophy. While the term “law” is never explicitly defined in the Code, the Catechism of the Catholic Church cites Aquinas in defining law as “...an ordinance of reason for the common good, promulgated by the one who is in charge of the community” and reformulates it as “...a rule of conduct enacted by competent authority for the sake of the common good.” Question: What school of thought serves as a model for canon theory?

1. the principles of AristotelianThomistic legal philosophy is the school of thought that serves as a model for canon theory.

**Q2.1** For each of the three question answering problems in Section 1, copy the Question and the Context into the model interface and obtain the resulting generation. Report the results, and comment on how the model outputs compare to your answers from Section 1. Create your own context and question to query this same model, and report the answer. Is it correct?

**A-**

1. a decimal system of values
2. bole
3. AristotelianThomistic legal philosophy
4. Context: My name is Vishal and I live in Toronto.

Question: what is my name?

Answer: Vishal

From the examples above it can be seen the first 3 answers are accurate with the answers that I had inferred and the fourth new example also had the right answer.

**Q2.2** Read the pipeline tutorial from Huggingface to see the most straightforward way to use

models from the Huggingface model hub. Review all of the tasks that are supported by the

pipeline framework, and list those that take in text as input. For each of those, give a one

line description of what it does.

**A-**

The hugging face pipeline framework can support-

1. "conversational"- Used for a chatbot like use case where you have to generate conversations between 2 users
2. "question-answering"- Can give answers to questions given a context
3. Text2TextGeneration- given input text the model gives a text output based on the input
4. "text-classification"- can be used to classify input text
5. "table-question-answering"- given a input in the form of a table of strings and a question we will get the answer as the output.
6. "text-generation"- it is used for the autoregressive generation of text given a input.
7. "token-classification"- to classifiy the tokens in an input to their respective classes.
8. "translation"- can be used to translate from one to another(input to output)
9. "summarization"- the output of this pipeline will summarize the input.
10. "zero-shot-classification"- this pipeline is used to make zero shot classification on a pre trained model without training.

**Q2.3** The model page for deepset/roberta-base-squad2 has a section labelled “In Transformers” that uses the pipeline method, and it can do exactly the same thing that you just did in part 1 of this Section. Review and copy that code, make sure you can run the three example QA problems. Find a different model that can perform inference at least 25% faster (in wall clock time) than roberta-base-squad2 that can be used in that same pipeline, and see if it gives the same answers - report which model you chose, its inference time (and the time for roberta-base-squad2), and whether the answers were comparable to roberta-base-squad2 or not.

**A-**

Model- deepset/roberta-base-squad2

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

Model- deepset/tinyroberta-squad2

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

From the outputs above we can see the model tinyroberta- squad2 performed just as well as Roberta-base-squad2 but got the outputs in almost half the time as Roberta-base-squad2.

**Q3.1.a** If you do not wish to upload the newly trained model to the Hugging face hub, is it

necessary to execute the notebook login line of the notebook?

**A-**

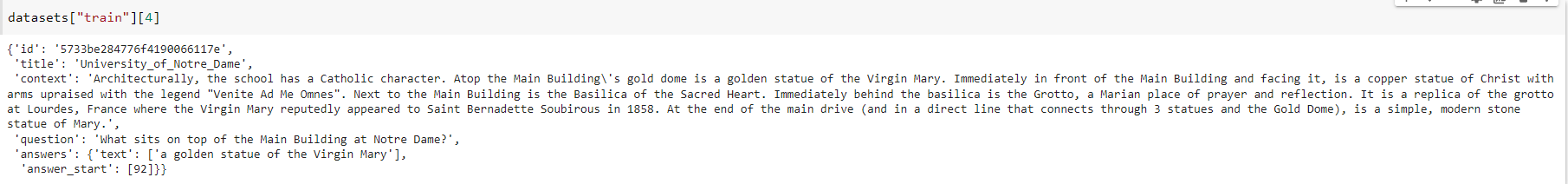
NO

**Q3.1.b** Give a single line of code that will print the 5th data point in the training split of the

SQuAD dataset?

**A-**

datasets["train"][4]



**Q3.1.c** Some input examples may exceed the maximum input length of the model. The code

in the tutorial truncates inputs that are greater than a maximum length, as part of the

tokenizer. It can split the input into two segments, one segment which fits and another

segment that is what remains. When calling the tokenizer, which argument causes the

tokenizer to return both the segments after truncation? If we want to have the two

segments have some overlap (just in case the text containing the answer is truncated),

which argument should be used to control the extent of overlapping?

**A-**

The return\_overflowing\_tokens argument when set to true causes the tokenizer to return both the segments after truncation.

The stride argument is used to control the extent of overlap. This is assigned by the variable doc\_stride.

**Q3.1.d** How are input examples that are not the same length handled in this code?

**A-**

They are padded to the right to make the examples the same length. pad\_on\_right = tokenizer.padding\_side == "right"

**Q3.1.e** In the function prepare train features(), are the start positions and end positions

referring to the token indices or the character indices?

**A-**

It is referring to the token indices

**Q3.1.f** Which gradient descent optimizer is used in this code? If we want to change the optimizer, how is this done?

**A-**

The gradient descant optimizer used is AdamW(adam with weight decay). To change the optimizer we can use the Create\_optimizer argument in the trainer class to set a up a new optimizer. A new optimizer from the  .optimization module can passed as a argument. (Eg. AdaFactor, AdamW)

**Q3.1.g** What is “checkpoint saving”, and why is it done? How often is checkpoint saving done in the model training in this code?

**A-**

Checkpoint saving saves the parameters of the model. It is done to load a model from any check point so we don’t need to train the model from scratch.

Checkpoint saving is done every 500 steps (updates of parameters) as this is the default values for the trainer.

**Q3.1.h** How is the performance of the model evaluated in this code?

**A-**

The model makes predictions on the preprocessed test set. The model gives the start and end position of the answers along with the logits. Using both of this the model computes a score.

The preprocessed test set has the ids and offset masks added to it to get the final output.

This final output is then compared against the ground truth to give the final evaluation score.

**Q3.2** Fine-tune the models, using only 1/10 of the training dataset, with your choice of hyperparameters. Note: You need to enable the colab’s GPU runtime for fine-tuning, and keep the page open for around 20 minutes, so the colab runtime does not disconnect. Try a few different hyperparameters and report the best set.

**A-**

Text

Description automatically generated

1. Total runtime of a single training run- 13:23 mins
2. Performance- training loss after 3 epochs- 0.85 and validation loss of 2.32

Evaluation-

Text

Description automatically generated with low confidence

Exact match of 50% and f1 score of 65%

1. Hyperparameters-
   1. Learning rate- 2e-2
   2. Weight decay- 0.05
2. Find 3 examples where the model failed and suggest why?

Failed examples-

'context': "Notre Dame alumni work in various fields. Alumni working in political fields include state governors, members of the United States Congress, and former United States Secretary of State Condoleezza Rice. A notable alumnus of the College of Science is Medicine Nobel Prize winner Eric F. Wieschaus. A number of university heads are alumni, including Notre Dame's current president, the Rev. John Jenkins. Additionally, many alumni are in the media, including talk show hosts Regis Philbin and Phil Donahue, and television and radio personalities such as Mike Golic and Hannah Storm. With the university having high profile sports teams itself, a number of alumni went on to become involved in athletics outside the university, including professional baseball, basketball, football, and ice hockey players, such as Joe Theismann, Joe Montana, Tim Brown, Ross Browner, Rocket Ismail, Ruth Riley, Jeff Samardzija, Jerome Bettis, Brett Lebda, Olympic gold medalist Mariel Zagunis, professional boxer Mike Lee, former football coaches such as Charlie Weis, Frank Leahy and Knute Rockne, and Basketball Hall of Famers Austin Carr and Adrian Dantley. Other notable alumni include prominent businessman Edward J. DeBartolo, Jr. and astronaut Jim Wetherbee.", 'question': 'Mariel Zagunis is notable for winning what?'

Graphical user interface, text, application

Description automatically generated

'context': "In the British West Indian islands (and also in the United States), the majority of enslaved blacks brought across the Atlantic came from West Africa (roughly between modern Senegal and Ghana). Very little of Bermuda's original black emigration came from this area. The first blacks to arrive in Bermuda in any numbers were free blacks from Spanish-speaking areas of the West Indies, and most of the remainder were recently enslaved Africans captured from the Spanish and Portuguese. As Spain and Portugal sourced most of their slaves from South-West Africa (the Portuguese through ports in modern-day Angola; the Spanish purchased most of their African slaves from Portuguese traders, and from Arabs whose slave trading was centred in Zanzibar). Genetic studies have consequently shown that the African ancestry of black Bermudians (other than those resulting from recent immigration from the British West Indian islands) is largely from the a band across southern Africa, from Angola to Mozambique, which is similar to what is revealed in Latin America, but distinctly different from the blacks of the West Indies and the United States.", 'question': 'Why is the black population in Bermuda different from that in the British West Indies and the United States?'

A picture containing table

Description automatically generated

'context': 'House music is a genre of electronic dance music that originated in Chicago in the early 1980s. It was initially popularized in Chicago, circa 1984. House music quickly spread to other American cities such as Detroit, New York City, and Newark – all of which developed their own regional scenes. In the mid-to-late 1980s, house music became popular in Europe as well as major cities in South America, and Australia. Early house music commercial success in Europe saw songs such as "Pump Up The Volume" by MARRS (1987), "House Nation" by House Master Boyz and the Rude Boy of House (1987), "Theme from S\'Express" by S\'Express (1988) and "Doctorin\' the House" by Coldcut (1988) in the pop charts. Since the early to mid-1990s, house music has been infused in mainstream pop and dance music worldwide.', 'question': 'What genre does House music fall into?'

Graphical user interface, text, application, email

Description automatically generated

For the first example the prediction might be wrong as it is quite a hard context as it is very verbose. For the next 2 examples we can see the answers are partially right they are not an exact match this may be fixed with a larger training set as the Training set that was used was quite small.

**Q4.1** To get a sense of the influence of some of the generation parameters, explore at least 10 combinations of temperature and top p to generate a maximum of 30 tokens using autoregressive generation with the GPT2 model. Comment on how the generation differs across the range of parameters that you have selected. You must choose your own range, and you’ll have to do some exploration to do that.

**A-**

top\_p=0.01 temperature=0.9

Text

Description automatically generated

top\_p=0.01 temperature=0.7

Text

Description automatically generated with medium confidence

top\_p=0.01 temperature=0.2

A picture containing diagram

Description automatically generated

top\_p=0.2 temperature=0.7

Text

Description automatically generated

top\_p=0.5 temperature=0.7

A picture containing diagram

Description automatically generated

top\_p=0.9 temperature=0.7

A picture containing text

Description automatically generated

top\_p=0.5 temperature=1

A picture containing text

Description automatically generated

top\_p=0.5 temperature=2

Text

Description automatically generated

top\_p=0.7 temperature=1

A picture containing diagram

Description automatically generated

top\_p=0.2 temperature=4

Graphical user interface, application

Description automatically generated

In the first 3 examples we have a very low top\_p value and no matter what the temperature value is we always obtain the same result and this result is a meaningful one.

In the next 3 examples- Choosing a temperature value of 0.7 and varying the top\_p in 0.2, 0.5, 0.9 it can be seen the output slowly becomes worse or less objective and more narrative as we increase the top\_p value.

In the next 2 examples- we set the top\_p value to 0.5 and try higher temperature values of 1 and 2. Here we find the outputs to be objective, but they do not make much sense.

The last 2 examples the values of top\_p and temperature are arbitrarily chosen and again the outputs are quite subjective or in the form of a story.

After exploring the parameters I think top\_p of 0.2 and temperature of 0.7 is the best as is does not use an extremely low top\_p value which might restrict our sampling and it can also be seen higher temperature values tend to make outputs that are more narative.

**Q4.2** Modify the code to output the probabilities of the each word that is generated. You’ll need to set these two generate parameters:return\_dict\_in\_generate=True and, output\_scores=True, and extract the probabilities that come in the returned dictionary one call at a time. Provide a table that shows these probabilities, similar to Assignment 3. Comment on the probabilities.

**A-**

Text

Description automatically generated

|  |  |
| --- | --- |
| Word | Probability |
| it | 0.16 |
| is | 0.33 |
| a | 0.15 |
| key | 0.06 |
| part | 0.15 |
| of | 0.98 |
| the | 0.26 |
| global | 0.11 |
| economy | 0.19 |
| . | 0.33 |
| /n | 0.21 |
| /n | 0.99 |
| “ | 0.28 |
| we | 0.14 |
| need | 0.23 |
| to | 0.78 |

Most of the words have small probabilities as we are dealing with a very large vocabulary. But there are times the model is very confident such as predicting the word ‘of’.

**Q4.3** Write new code that generates the probability tree (like the one drawn on the board in Lecture 6) using the treelib package that you can find here. Generate the tree for the above sequence as input, providing the top 3 probabilities for each word position, as far as is practical to see. (You’ll have to apply some common sense here to visualize the tree). Comment on the what you see in the tree. Is the tree affected by the top p parameter or the temperature parameter? Why or why not?

**A-**

Generated tree with without random sampling

Diagram, schematic

Description automatically generated

Tree generated with top\_p=0.2 and temperature= 0.7

Chart, box and whisker chart

Description automatically generated with medium confidence

By reducing the top\_p to 0.1 and increasing the temperature to 1.5 we can see better results for the tree.

Chart

Description automatically generated

The 2 parameters effect the tree as they control the space and how to sample the space so as we build the tree sometimes the sampling may not be ideal and can cause slightly random words to be chosen (words that don’t fit the context perfectly).

**Q5.1** Using the second QA problem from Section 1 (ID 56fa3d788f12f319006300ff), find a way (i.e.engineer the input) to get GPT-3 to answer this question (with all the information given) as best it can. Then, pose the same question to GPT-3 without giving it the context. Compare and contrast the results to the one you obtained in Section 2.

**A-**

When we provide the context to GPT3-

Text

Description automatically generated

When the context is not given-

Graphical user interface, text

Description automatically generated with medium confidence

When the context is given to GPT3 we can see it gives the same answer as the section 2 but it is framed in a more human like manner.

When the context is not provided the model answers the question with its pre trained knowledge as no context was given to it.

**Q5.2** You are to “design” a prompt for GPT-3 that takes direct statements of fact and turns

them into such “softer” statements. Here is one more example of the original statement:

You’re having trouble getting focused. Try experimenting with different prompts, and

the generation parameters (Temperature, top-p, etc.) to see how well you can make such a

converter, for this example and one of your own choosing. For both of those examples, report the best 4 results, and show how you achieved them.

**A-**

Example 1- best 4 results

The images below show the prompts used and the temperature and top p parameters used

Text

Description automatically generated with medium confidence

Text

Description automatically generated with low confidence

Text

Description automatically generated with low confidence

Text

Description automatically generated

Example2- You are having a hard time. (best 4 results)

The images below show the prompts used and the temperature and top p parameters used

Text

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

Text

Description automatically generated with medium confidence

Text

Description automatically generated with low confidence