Fine-tuning LLM for Enterprise Applications

-

Test Case 2 Medical Misinformation Detection in LLMs

Submitted by Joyal Poulose
S223721123
Master of Data Science

Onset:

This is a documentation of what I've done so far (end week) for this project, initially I had tried to make all of the project locally since I believed I had ample requirements for loading and training the model. However, I later realized that this wouldn't be possible because of technical and resource limitations. Therefore, I referred to Google Colab for the rest of the project however, it too had some technical limitations such as limited compute units, ever after restarting the kernel. I tried my best by creating different accounts to further utilize the free compute units given, which proved to be fruitless.

Note that, I had completed the code locally for Dataset preparation pipeline, Dataset Analysis pipeline, Training pipeline, Evaluation matrices pipeline and a final report. But it was the tinkering with the parameters and fine tuning which made me choose to opt for Google Colab and thus I had to tweak my code a lot which led me down a spiral chasing errors and warnings.

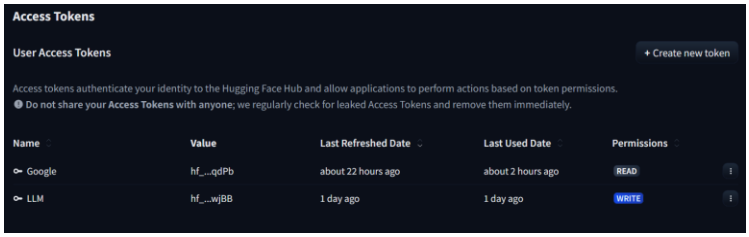
Project:

This project involves fine-tuning an LLM model on medical misinformation datasets to enhance its ability to detect and classify fake medical news in real-time. Specifically, the model is trained on datasets such as HealthFact, SciFact, and the COVID-19 Fake News Dataset. The goal is to fine-tune the LLM to accurately classify text as either factual or misinformation in the medical domain.
For my project, I had chosen the Falcon-2 with 7 billion parameters to classify and detect. I used the HuggingFace to load the model from tiiuae/falcon-7b" using an API key.

This project is crucial due to the rapidly growing problem of misinformation, particularly in the healthcare and medical sectors. Many people are prone to this false information, which is widely present in the internet unchecked, so a model like mine can help mitigate this issue. Misinformation can have serious consequences, such as individuals making health decisions based on false or misleading information. Therefore, by fine-tuning LLMs for this specific task, this project aims to contribute to creating more reliable AI models that can assist in safeguarding public health, ultimately improving the trustworthiness of online health information.

API/Token:

As mentioned earlier, I had used the Huggingface to load my model, I created a key by first creating an account and then publishing a new token with the necessary requirements.



Then, I integrated this into my code by logging in using the huggingface_hub library and login() function which accepts the hf token value as a parameter.

```
from huggingface_hub import login

# Configuration settings
login(token="hf_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx")
MODEL_NAME = "tiiuae/falcon-7b"  # Base model
```

Environment:

Most of the project was developed in Google Colab for easy access to GPU resources, which is essential for large model training. Ultimately, I had to buy more compute units, so GPUs were utilized during model training to speed up computation times significantly. I had initially tried using the Nvidia T4 GPU, but I figured it would be easier and better to run on the A100 instead of the former. Python version 3.8 was used for this model for its compatibility with necessary libraries such as PyTorch, transformers and so on, a short list of libraries that I've mainly used are:

1. Pandas,
2. Numpy,
3. Torch,
4. Transformers,
5. PEFT,

6. Accelerate,
7. Bitsandbytes,
8. Accelerate,
9. Logging,
10. Sklearn.metrics and so on

LLM Setup:

Towards the beginning of the week, I hadn't much idea of Falcon model, but researching a bit on the internet, I realised that Falcon 7B model was a powerful transformer model and it has good performance for its parameters. Therefore, I selected this model for fine-tuning due to its ability to process and generate human-like text. It is particularly effective for text classification tasks like my project which is medical misinformation detection.

For my model, Hugging Face is the primary provider, offering a large selection of pre-trained models, including Falcon 7B. The platform makes it easy to load and fine-tune models on specific datasets using its simple-to-use API. For loading my model from my saved output directory, I would make use of the function from_pretrained() to call in a saved model from a given directory.

Datasets

For this project, we were allowed to use 3 different datasets for which we were advised to pretrain on 2 of these datasets and then fine tune on the 3rd. These datasets are containing medical articles labelled as factual or fake, providing a solid base for training models in the domain of medical misinformation.

The datasets were pre-processed by removing any irrelevant columns, handling missing values, and tokenizing the text data using the Hugging Face tokenizer to convert text into tokens the model can process. Moreover, the data was also split into training, validation, and test sets. Some information about the datasets is given below:

1. Scifact: This dataset had in total 1711 records with 3 columns, it also included 3 labels which were "Misleading", "False" and "True". It also included a unique column which wasn't present in any other dataset named "evidence_text".
2. HealthFact: In this dataset, there were approximately 12251 records which made it the largest dataset in comparison. It also had 5 columns and 3 labels which were "false", "MISLEADING" and "true". In this dataset the unique columns included "explanation" a bogus entry "Unnamed" and "claim_id".
3. COVID-19 Fake News: This dataset contained 10700 records and 3 columns which also had labelling such as "real", "fake" and "nan". Moreover, the columns were named "id" and "tweet".

These were the datasets available for my project, as you can all 3 of these datasets had different values, different columns as well as labels. My team leader asked for 3 of us to process the dataset by removing unnecessary columns and then unifying all the labels for which I had volunteered but one of my peers had started working on it.

Healthfact before processing:

--- HealthFact Train Data (Initial) ---

	Unnamed: 0	claim_id	claim	explanation	label
0	0	15661	"The money the Clinton Foundation took from fr...	"Gingrich said the Clinton Foundation ""took m...	false
1	1	9893	Annual Mammograms May Have More False-Positives	This article reports on the results of a study...	MISLEADING
2	2	11358	SBRT Offers Prostate Cancer Patients High Canc...	This news release describes five-year outcomes...	MISLEADING
3	3	10166	Study: Vaccine for Breast, Ovarian Cancer Has ...	While the story does many things well, the ove...	true
4	4	11276	Some appendicitis cases may not require 'emerg...	We really don't understand why only a handful ...	true

--- HealthFact Dev Data (Initial) ---

	Unnamed: 0	claim_id	claim	explanation	label
0	0	34656	A baby died at an unnamed medical facility be...	Fellow Twitter users suggested @FierceFemtivis...	MISLEADING
1	1	3632	Bat from Shawnee County tests positive for rab...	A bat found in northeastern Kansas has tested ...	true
2	2	29558	Germany has banned pork from school canteens b...	What's true: Some politicians complained that ...	false
3	3	8416	Coronavirus prompts Canada to roll out safe dr...	Canada's Pacific province of British Columbia ...	true
4	4	7169	Wayne National Forest plans fires for tree, wi...	Nearly 2,000 acres of Wayne National Forest in...	true

--- HealthFact Test Data (Initial) ---

	Unnamed: 0	claim_id	claim	explanation	label
0	0	33456	A mother revealed to her child in a letter aft...	The one-eyed mother story expounds upon two mo...	false
1	1	2542	Study says too many Americans still drink too ...	On any given day in the United States, 18 perc...	true
2	2	26678	Viral image Says 80% of novel coronavirus case...	The website Information is Beautiful published...	true
3	3	40705	An email says that 9-year old Craig Shergold o...	Send greeting or business cards to cancer vict...	false
4	4	35718	Employees at a Five Guys restaurant in Daphne,...	What's undetermined: As of this writing, Five ...	MISLEADING

Healthfact after processing:

--- HealthFact Train Data (Processed) ---

	text	evidence	label
0	"The money the Clinton Foundation took from fr...	"Gingrich said the Clinton Foundation ""took m...	FALSE
1	Annual Mammograms May Have More False-Positives	This article reports on the results of a study...	MISLEADING
2	SBRT Offers Prostate Cancer Patients High Canc...	This news release describes five-year outcomes...	MISLEADING
3	Study: Vaccine for Breast, Ovarian Cancer Has ...	While the story does many things well, the ove...	TRUE
4	Some appendicitis cases may not require 'emerg...	We really don't understand why only a handful ...	TRUE

--- HealthFact Dev Data (Processed) ---

	text	evidence	label
0	A baby died at an unnamed medical facility be...	Fellow Twitter users suggested @FierceFemtivis...	MISLEADING
1	Bat from Shawnee County tests positive for rab...	A bat found in northeastern Kansas has tested ...	TRUE
2	Germany has banned pork from school canteens b...	What's true: Some politicians complained that ...	FALSE
3	Coronavirus prompts Canada to roll out safe dr...	Canada's Pacific province of British Columbia ...	TRUE
4	Wayne National Forest plans fires for tree, wi...	Nearly 2,000 acres of Wayne National Forest in...	TRUE

--- HealthFact Test Data (Processed) ---

	text	evidence	label
0	A mother revealed to her child in a letter aft...	The one-eyed mother story expounds upon two mo...	FALSE
1	Study says too many Americans still drink too ...	On any given day in the United States, 18 perc...	TRUE
2	Viral image Says 80% of novel coronavirus case...	The website Information is Beautiful published...	TRUE
3	An email says that 9-year old Craig Shergold o...	Send greeting or business cards to cancer vict...	FALSE
4	Employees at a Five Guys restaurant in Daphne,...	What's undetermined: As of this writing, Five ...	MISLEADING

Scifact before processing:

```
--- SciFact Train Data (Initial) ---
```

	claim	evidence_text	label
0	0-dimensional biomaterials lack inductive prop...		Misleading
1	1 in 5 million in UK have abnormal PrP positiv...	RESULTS Of the 32,441 appendix samples 16 were...	False
2	1-1% of colorectal cancer patients are diagnos...		Misleading
3	10% of sudden infant death syndrome (SIDS) dea...		Misleading
4	32% of liver transplantation programs required...	Policies requiring discontinuation of methadon...	True

```
--- SciFact Dev Data (Initial) ---
```

	claim	evidence_text	label
0	0-dimensional biomaterials show inductive prop...		Misleading
1	1,000 genomes project enables mapping of genet...	We propose as an alternative explanation that ...	True
2	1,000 genomes project enables mapping of genet...	In conclusion, uncommon or rare genetic varian...	True
3	1/2000 in UK have abnormal PrP positivity.	RESULTS Of the 32,441 appendix samples 16 were...	True
4	5% of perinatal mortality is due to low birth ...		Misleading

```
--- SciFact Test Data (Initial) ---
```

	id	claim
0	7	10-20% of people with severe mental disorder r...
1	8	25% of patients with melanoma and an objective...
2	16	50% of patients exposed to radiation have acti...
3	23	8% of burn patients are admitted for hospitali...
4	29	A breast cancer patient's capacity to metaboli...

```
SciFact test set doesn't have labels. Using a portion of dev set as test set.
```

Scifact after processing:

```
--- SciFact Train Data (Processed) ---
```

	text	evidence	label
0	0-dimensional biomaterials lack inductive prop...		MISLEADING
1	1 in 5 million in UK have abnormal PrP positiv...	RESULTS Of the 32,441 appendix samples 16 were...	FALSE
2	1-1% of colorectal cancer patients are diagnos...		MISLEADING
3	10% of sudden infant death syndrome (SIDS) dea...		MISLEADING
4	32% of liver transplantation programs required...	Policies requiring discontinuation of methadon...	TRUE

```
--- SciFact Dev Data (Processed) ---
```

	text	evidence	label
373	The amount of publicly available DNA data doub...		MISLEADING
263	N348I mutations cause resistance to zidovudine...	Biochemical analyses of recombinant RT contain...	TRUE
74	CCL19 is absent within dLNs.		MISLEADING
143	Flexible molecules experience greater steric h...		MISLEADING
147	Functional consequences of genomic alterations...		MISLEADING

```
--- SciFact Test Data (Processed) ---
```

	text	evidence	label
405	The treatment of cancer patients with co-IR bl...		MISLEADING
255	Monoclonal antibody targeting of N-cadherin in...	In vivo, these antibodies slowed the growth of...	TRUE
98	Cells undergoing methionine restriction may ac...		MISLEADING
171	Hyperfibrinogenemia decreases rates of femorop...	CONCLUSIONS Plasma fibrinogen concentration wa...	FALSE
387	The minor G allele of FOXO3 is related to more...	We identify a noncoding polymorphism in FOXO3A...	FALSE

Covid before processing:

```
Analyzing COVID-19 Fake News dataset...

--- COVID-19 Train Data (Initial) ---
```

	id	tweet	label
0	1	The CDC currently reports 99031 deaths. In gen...	real
1	2	States reported 1121 deaths a small rise from ...	real
2	3	Politically Correct Woman (Almost) Uses Pandem...	fake
3	4	#IndiaFightsCorona: We have 1524 #COVID testin...	real
4	5	Populous states can generate large case counts...	real

```
--- COVID-19 Dev Data (Initial) ---
```

	id	tweet	label
0	1	Chinese converting to Islam after realising th...	fake
1	2	11 out of 13 people (from the Diamond Princess...	fake
2	3	COVID-19 Is Caused By A Bacterium, Not Virus A...	fake
3	4	Mike Pence in RNC speech praises Donald Trump'...	fake
4	5	6/10 Sky's @EdConwaySky explains the latest #C...	real

```
--- COVID-19 Test Data (Initial) ---
```

	id	tweet	label
0	1	Our daily update is published. States reported...	True
1	2	Alfalfa is the only cure for COVID-19.	False
2	3	President Trump Asked What He Would Do If He W...	False
3	4	States reported 630 deaths. We are still seein...	True
4	5	This is the sixth time a global health emergen...	True

Covid after processing:

```
--- COVID-19 Train Data (Processed) ---
```

	text	evidence	label
0	The CDC currently reports 99031 deaths. In gen...		TRUE
1	States reported 1121 deaths a small rise from ...		TRUE
2	Politically Correct Woman (Almost) Uses Pandem...		FALSE
3	#IndiaFightsCorona: We have 1524 #COVID testin...		TRUE
4	Populous states can generate large case counts...		TRUE

```
--- COVID-19 Dev Data (Processed) ---
```

	text	evidence	label
0	Chinese converting to Islam after realising th...		FALSE
1	11 out of 13 people (from the Diamond Princess...		FALSE
2	COVID-19 Is Caused By A Bacterium, Not Virus A...		FALSE
3	Mike Pence in RNC speech praises Donald Trump'...		FALSE
4	6/10 Sky's @EdConwaySky explains the latest #C...		TRUE

```
--- COVID-19 Test Data (Processed) ---
```

	text	evidence	label
0	Our daily update is published. States reported...		TRUE
1	Alfalfa is the only cure for COVID-19.		FALSE
2	President Trump Asked What He Would Do If He W...		FALSE
3	States reported 630 deaths. We are still seein...		TRUE
4	This is the sixth time a global health emergen...		TRUE

Code:

I started off my code with importing all necessary libraries as mentioned earlier. Moreover, I also initialized *Logger* for tracking all events during my program execution. Then, in the next code cell, I created a class named “ProjectConfig” where I have declared and set most variables for my model such as the model’s name, batch size, lora configurations, label mapping and so on.

```python
class ProjectConfig:
    model_name: str = "tiiuae/falcon-7b"
    max_length: int = 128
    batch_size: int = 1
    num_epochs: int = 2
    learning_rate: float = 2e-5
    weight_decay: float = 0.001
    lora_r: int = 16
    lora_alpha: int = 32
    lora_dropout: float = 0.05
    output_dir: str = "/content/drive/MyDrive/Model/"
    eval_steps: int = 100
    save_steps: int = 100
    label_map: Dict[str, int] = None
    num_labels: int = 3

# Initialize with proper label mapping
config = ProjectConfig()
config.label_map = {"TRUE": 0, "True": 0, "real": 0,
                    "FALSE": 1, "False": 1, "fake": 1,
                    "MISLEADING": 2, "Misleading": 2, "misleading": 2}
```

Class MedicalDataProcessor

Then I created a class “MedicalDataProcessor” which configures and loads all the datasets in this project. I did try mitigating the redundancy of the code, but since some of the datasets contained nan values or some dataset had 2 classes and not 3, I figured it would be better to just create 3 distinct functions to handle the 3 datasets. The functions load_healthfact(), load_scifact and load_covid_fake_news each handles and standardizes the datasets healthfact, scifact and covid19 fake news dataset respectively.

Moreover, I also balance the datasets tuned to the minimum count across all labels. Process all datasets, tokenize them and then map them using the label_map.

Class ModelTrainer

Moving on to my “ModelTrainer” class, this is the main class where I’ve defined and implemented my model with LoRA efficiency. At first, I tried to load the model without any quantization to see if my colab setup would be sufficient enough to handle Falcon 7 b. However, it wasn’t nearly enough, and I had to quantize it to 4 bits just to get it to load.

In the __init__() I do check if I’ve already pretrained or finetuned before, because I had already saved my model once and in order to debug one of the functions again I had to rerun the code, which would lead it to run the training again. So, I avoided it by using if statement to check for an existing model. These are the parameters for my base model:

```python
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True, # Load in 8-bit
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.bfloat16,
    bnb_4bit_use_double_quant=True,
)

# Load base model
model = AutoModelForSequenceClassification.from_pretrained(
    self.model_name,
    num_labels=self.num_labels,
    torch_dtype=torch.bfloat16,
    quantization_config=bnb_config,
    device_map="auto"
)
```

As mentioned before, I quantized the model in 4 bits using BitsAndBytesConfig() to pass parameters to quantize it. Then, I load the model using the parameters as shown in the photo. I also added LoRA configurations for efficient tuning, below are the given parameters.

```
# LoRA configuration
peft_config = LoraConfig(
    task_type=TaskType.SEQ_CLS,
    r=self.config.lora_r,
    lora_alpha=self.config.lora_alpha,
    lora_dropout=self.config.lora_dropout,
    bias="none",
    target_modules=["query_key_value"],
)
```

Moreover, in this class, I have also defined a few other functions like evaluate() and inside evaluate() compute_metrics() which calculates all the metrics for evaluation such as F1, recall, ROCAUC, accuracy, false_positive_rate_for_true and precision.

Lastly, we have the train() function, which as the name suggests trains the model using a given set of arguments.
The arguments are show in the picture below:

```
# Training arguments
training_args = TrainingArguments(
    output_dir=output_dir,
    eval_strategy="steps",
    eval_steps=self.config.eval_steps,
    save_strategy="steps",
    save_steps=self.config.save_steps,
    learning_rate=self.config.learning_rate,
    per_device_train_batch_size=self.config.batch_size,
    per_device_eval_batch_size=self.config.batch_size,
    num_train_epochs=num_epochs,
    weight_decay=self.config.weight_decay,
    load_best_model_at_end=True,
    metric_for_best_model="accuracy",
    push_to_hub=False,
    logging_dir=os.path.join(output_dir, "logs"),
    logging_steps=100,
    bf16=True,
    gradient_accumulation_steps=8,
)
```

Class RAGConfidenceScorer

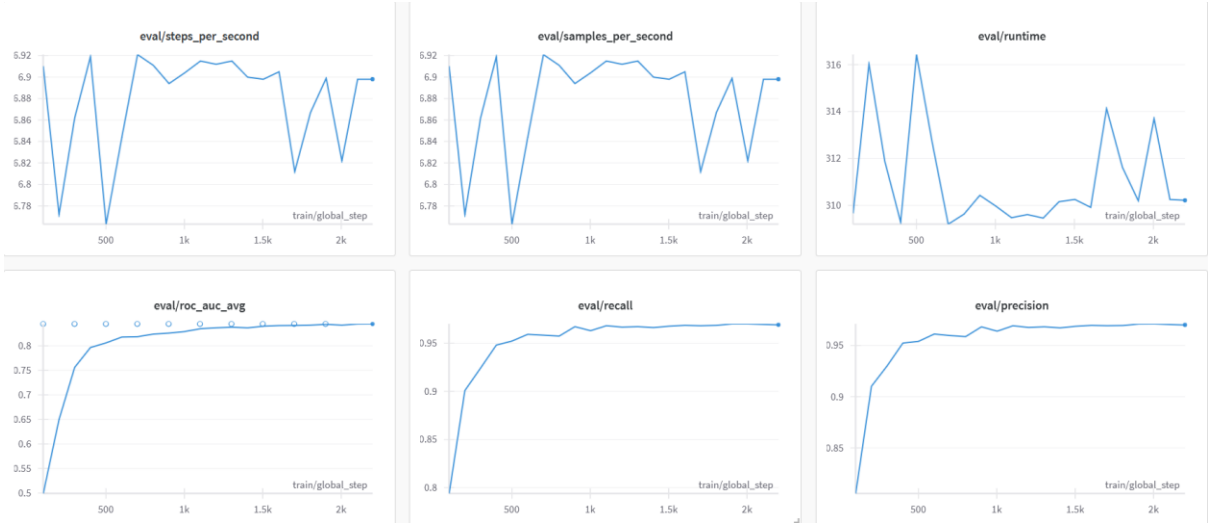
In this class, I had a bit of trouble whilst setting it up, when I was compiling it locally, this pipeline would work fine, but afterwards when I tried to convert it into google colab friendly version, it would create havoc raising many errors. Therefore, I've implemented the loading from a checkpoint code here again as removing it created errors on my end.

The core method search_pubmed() queries the PubMed API for article metadata relevant to a given claim, using retries and XML parsing with logging for event tracking. The PubMed search currently returns a list of dicts, not abstract texts directly. We need to fetch or use abstract texts for embedding. Since the current search_pubmed only returns metadata (title, date), I could not directly embed abstract texts here but fixed it in code.

The rest of the class provides functionality to extract claims from text, compute semantic similarity between claims and retrieved evidence using SentenceTransformers(), and calculate a trust_score. It then combines this score with the model's own classification confidence to generate a final combined_score reflecting both statistical and evidence-backed trust in a claim.

I got the below metrics for my mode:
s
Evaluation results: {'healthfact_metrics': {'eval_loss': 3.898899555206299, 'eval_model_preparation_time': 0.0071, 'eval_accuracy': 0.3093278463648834, 'eval_precision': 0.21323175391889176, 'eval_recall': 0.3093278463648834, 'eval_f1': 0.1502641193525452, 'eval_false_positive_rate_for_true': 0.4849108367626886, 'eval_roc_auc_avg': 0.5295956177995308, 'eval_runtime': 144.5705, 'eval_samples_per_second': 10.085, 'eval_steps_per_second': 10.085}, 'covid_metrics': {'eval_loss': 2.722978353500366, 'eval_model_preparation_time': 0.0073, 'eval_accuracy': 0.4766355140186916, 'eval_precision': 0.22718141322386234, 'eval_recall': 0.4766355140186916, 'eval_f1': 0.3077014077842186, 'eval_false_positive_rate_for_true': 0.5233644859813084, 'eval_roc_auc_avg': nan, 'eval_runtime': 212.3862, 'eval_samples_per_second': 10.076, 'eval_steps_per_second': 10.076}}

Using wandb, I got these graphs, though, I'm not pretty sure, if I saved them before or after my finetuning started, due to the drop in some metrics:



We move on to the run_evaluation() function, this function is designed to perform model evaluation on preprocessed datasets using an already fine-tuned model within a Google Colab environment. It begins by mounting Google Drive to access saved model files, then sets up logging to monitor execution progress and potential errors. It defines paths for both the pretraining and fine-tuning outputs based on a config object, moreover it checks for the existence of the fine-tuned model directory.
The function then evaluates the model on two distinct test sets: the first being HealthFact test set (used during pretraining) and the second COVID Fake News test set (which was used for fine-tuning), logging their respective performance metrics. The evaluation results, which

include metrics like accuracy or F1-score (depending on implementation), are returned as a dictionary.

These were my metrics for the model:
PreTraining and Evaluation:

Step	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1	False Positive Rate For True	Roc Auc Avg
100	1.922100	1.154756	0.469076	0.404587	0.469076	0.412662	0.124392	0.498975
200	1.199800	1.077018	0.553857	0.558738	0.553857	0.552907	0.138985	0.649846
300	1.072500	0.906322	0.609451	0.599504	0.609451	0.602008	0.102154	0.756260
400	0.877000	0.806608	0.618485	0.634532	0.618485	0.625145	0.134816	0.796693
500	0.896900	0.815419	0.612926	0.670193	0.612926	0.629814	0.176511	0.806241
600	0.795600	0.806325	0.620570	0.667652	0.620570	0.635765	0.154969	0.818051
700	0.812300	0.801846	0.626129	0.684275	0.626129	0.643146	0.170257	0.818697
800	0.787500	0.786155	0.630994	0.668512	0.630994	0.641366	0.136901	0.824071
900	0.769600	0.781446	0.624739	0.664665	0.624739	0.633042	0.128562	0.826069
1000	0.701800	0.789597	0.624739	0.692081	0.624739	0.643607	0.164003	0.829267
1100	0.737600	0.744068	0.644197	0.677936	0.644197	0.655836	0.134816	0.834944
1200	0.771200	0.738800	0.645587	0.679417	0.645587	0.657464	0.136901	0.836874
1300	0.788700	0.743052	0.641418	0.693639	0.641418	0.657691	0.153579	0.838185
1400	0.716800	0.762245	0.633079	0.701485	0.633079	0.651642	0.159138	0.836795
1500	0.748000	0.706430	0.654621	0.679431	0.654621	0.663879	0.134816	0.840073
1600	0.682800	0.725718	0.653926	0.707548	0.653926	0.670186	0.154274	0.841385
1700	0.723700	0.716444	0.658791	0.686421	0.658791	0.667383	0.120222	0.841648
1800	0.658200	0.723054	0.651842	0.702083	0.651842	0.667579	0.149409	0.842388
1900	0.656300	0.716996	0.653926	0.696835	0.653926	0.668132	0.138290	0.843977
2000	0.713600	0.742914	0.643502	0.712788	0.643502	0.663042	0.165393	0.842339
2100	0.680500	0.717814	0.654621	0.694069	0.654621	0.667353	0.133426	0.844569
2200	0.670800	0.714860	0.658791	0.698567	0.658791	0.672037	0.136206	0.844714

For fine tuning:

Step	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1	False Positive Rate For True
100	2.812900	0.508630	0.793925	0.805486	0.793925	0.796955	0.133645

Step	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1	False Positive Rate For True
200	0.334600	0.314804	0.900935	0.910259	0.900935	0.903607	0.028037
300	0.297500	0.238276	0.924299	0.930150	0.924299	0.926658	0.051402
400	0.167600	0.198575	0.948131	0.952416	0.948131	0.950124	0.033178
500	0.237600	0.188275	0.952336	0.954136	0.952336	0.953222	0.022430
600	0.236300	0.172044	0.959346	0.961273	0.959346	0.960248	0.024766
700	0.128200	0.176424	0.958411	0.959762	0.958411	0.958872	0.028505
800	0.150500	0.185844	0.957477	0.958792	0.957477	0.957890	0.013551
900	0.125400	0.170017	0.967290	0.968329	0.967290	0.967748	0.020561
1000	0.091200	0.179254	0.963084	0.964030	0.963084	0.963526	0.015888
1100	0.167700	0.168696	0.968224	0.969330	0.968224	0.968684	0.021028
1200	0.129400	0.174722	0.966822	0.967745	0.966822	0.967270	0.014953
1300	0.172900	0.168767	0.967290	0.968360	0.967290	0.967749	0.021028
1400	0.261500	0.164192	0.966355	0.967301	0.966355	0.966811	0.019159
1500	0.110700	0.164350	0.967757	0.968812	0.967757	0.968216	0.020561
1600	0.152100	0.161326	0.968692	0.969717	0.968692	0.969150	0.019626
1700	0.136300	0.162380	0.968224	0.969409	0.968224	0.968684	0.021963
1800	0.081500	0.162361	0.968692	0.969598	0.968692	0.969144	0.015888
1900	0.171700	0.159823	0.970093	0.971015	0.970093	0.970549	0.016355
2000	0.128300	0.160069	0.970093	0.971106	0.970093	0.970552	0.018692
2100	0.077400	0.159516	0.969626	0.970566	0.969626	0.970083	0.017290
2200	0.113700	0.159183	0.969159	0.970125	0.969159	0.969616	0.018224

There was one more column for the finetuning one, but the roc_auc_avg was filled with nan values, but when I printed a few debug statements to find out what happened, it didn't seem anything was wrong.

Class MedicalMisinformationDetector

This class is a comprehensive inference pipeline designed to detect and explain medical misinformation in textual claims using a fine-tuned transformer model enhanced with a PEFT (Parameter-Efficient Fine-Tuning) adapter and RAG-based (Retrieval-Augmented Generation) trust scoring. Upon initialization, it loads a tokenizer, and a quantized base transformer model (Falcon 7 B) resizes embeddings to match the tokenizer and integrates the PEFT adapter from the specified model directory.

The class maps numerical prediction outputs to textual labels which is TRUE, FALSE or MISLEADING. However, so far, I've noticed a big error in my model which I'll talk about soon. Apart from that, it utilizes a RAGConfidenceScorer to retrieve relevant medical evidence from

sources like PubMed (mentioned in the given draftcase), generating a trust score that reflects factual support from literature.

In the end, we have the get_explanation() function interprets the classification, elaborating on why a claim was classified a certain way and summarizing the level of confidence and support in medical literature. This setup enables the system to not only classify claims but also explain the rationale and evidence behind each prediction, making it suitable for applications in fact-checking, healthcare, and public information platforms.

On running the above code, I get the output:
===== MEDICAL MISINFORMATION DETECTION RESULTS =====

Claim: Vitamin C prevents COVID-19.
Classification: FALSE
Trust Score: 91
Combined Score: 0.70
Explanation: This claim is classified as FALSE and contradicts established medical knowledge. There is good confidence in this classification. The claim has strong support in medical literature (Trust Score: 91).

Relevant medical literature:
1. Vitamin C-An Adjunctive Therapy for Respiratory Infection, Sepsis and COVID-19. (2020 Dec 7)
2. The Long History of Vitamin C: From Prevention of the Common Cold to Potential Aid in the Treatment of COVID-19. (2020)
3. Vitamin C and COVID-19. (2020)

Claim: Regular exercise can help improve mental health.
Classification: FALSE
Trust Score: 69
Combined Score: 0.77
Explanation: This claim is classified as FALSE and contradicts established medical knowledge. There is good confidence in this classification. The claim has moderate support in medical literature (Trust Score: 69).

Relevant medical literature:
1. Sleep physiology, pathophysiology, and sleep hygiene. (2023 Mar-Apr)
2. Management of Lumbar Disc Herniation: A Systematic Review. (2023 Oct)
3. Role of Physical Activity on Mental Health and Well-Being: A Review. (2023 Jan)

Claim: Vaccines cause autism in children.
Classification: FALSE
Trust Score: 80
Combined Score: 0.74
Explanation: This claim is classified as FALSE and contradicts established medical knowledge. There is good confidence in this classification. The claim has moderate support in medical literature (Trust Score: 80).

Relevant medical literature:
1. Measles, Mumps, Rubella Vaccination and Autism: A Nationwide Cohort Study. (2019 Apr 16)
2. Vaccines are not associated with autism: an evidence-based meta-analysis of case-control and cohort studies. (2014 Jun 17)
3. The MMR Vaccine and Autism. (2019 Sep 29)

Claim: Mammograms may have false positives which increase with annual screening.
Classification: FALSE
Trust Score: 61
Combined Score: 0.79
Explanation: This claim is classified as FALSE and contradicts established medical knowledge. There is good confidence in this classification. The claim has moderate support in medical literature (Trust Score: 61).

Relevant medical literature:
1. Benefits and Harms of Breast Cancer Screening: A Systematic Review. (2015 Oct 20)
2. Breast density implications and supplemental screening. (2019 Apr)
3. Breast Cancer Screening Using Mammography, Digital Breast Tomosynthesis, and Magnetic Resonance Imaging by Breast Density. (2024 Oct 1)

Model complete. Ready for deployment

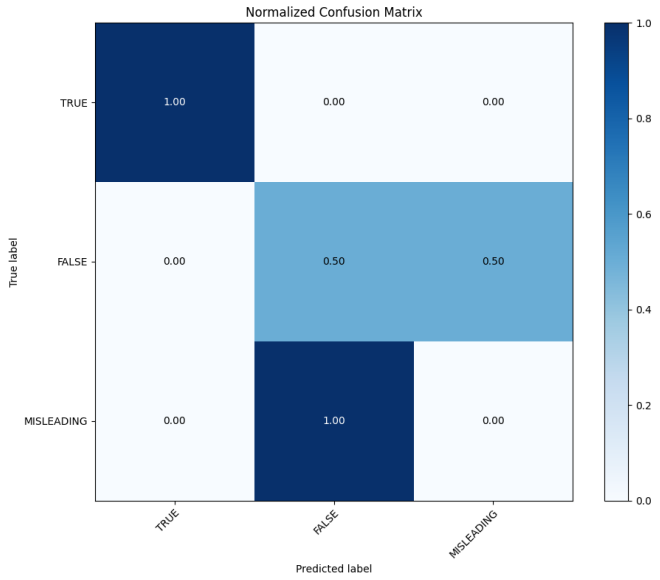
As you can see my model's working fine, however it does misclassify some statements, but if we look at the trust score it should be outputting the correct outputs however it doesn't. I tried debugging this all week but to no avail.

In the next few cells, I was trying my best to debug the code and analyse why my model was performing as such. One cell, evaluate_model_performance() function defines a comprehensive evaluation pipeline for assessing the performance of my multi-class classification model. The evaluate_model_performance function first standardizes label inputs by mapping various string representations ("True", "fake", "Misleading") to numerical indices.

It then calculates key metrics including accuracy, precision, recall, and F1-score using scikit-learn functions, and also generates a detailed classification report. The confusion matrix is computed and returned for further visualization. The plot_confusion_matrix function visualizes this confusion matrix, allowing optional normalization to reflect proportions instead of raw counts.

It uses matplotlib to generate a labeled heatmap of the confusion matrix, enhancing interpretability of model performance across classes. The code concludes with an example usage where true and predicted labels are evaluated, performance metrics are printed, and both raw and normalized confusion matrix plots are saved. This evaluation workflow is essential for understanding classification behavior, diagnosing model weaknesses, and supporting transparency in medical AI systems.

Note that, I did this on a sample dataset (for testing)

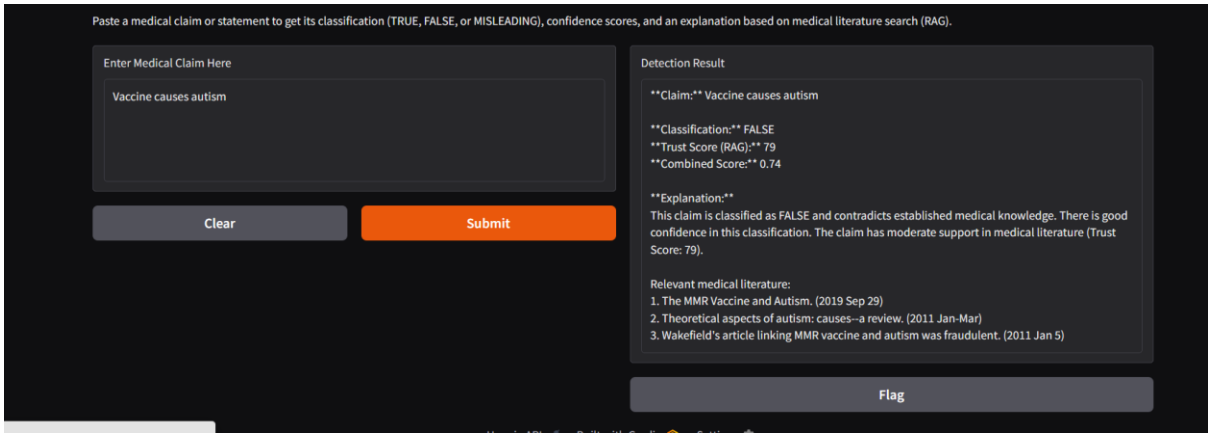


Gradio

The gradio_misinformation_detector code snippet sets up a Gradio web interface for my Medical Misinformation Detection System, enabling users to interactively input a medical claim and receive classification results in real time. The key component is the radio_misinformation_detector function, which acts as the backend callback for the Gradio app. Once a claim is submitted, the function returns a detailed output that includes:

- The original claim text
- The classification label (TRUE, FALSE, or MISLEADING)
- The Trust Score derived from the RAG (Retrieval-Augmented Generation) pipeline
- The Combined Score, which fuses model confidence with RAG evidence
- An Explanation that summarizes why the claim is classified as such, using retrieved medical literature

For instance,



Finally, the Gradio interface is launched with a title, input textbox (for entering the claim), and a read-only output textbox as shown above.

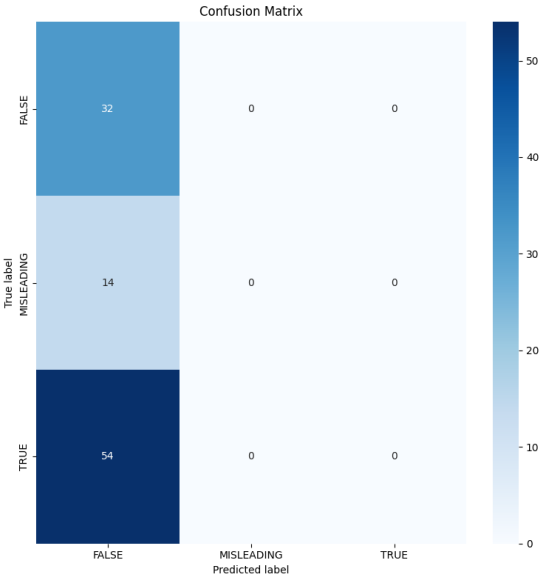
Possible Causes:

1. Imbalance in Class Labels: A prevalent issue is the imbalance in the dataset, where the number of "FALSE" instances far exceeds the "TRUE" and "MISLEADING" categories. When training, the model focuses on minimizing overall loss, which may result in a preference for predicting the majority class to enhance accuracy. If "FALSE" is the dominant class, the model is likely to classify uncertain instances as "FALSE" by default. But since I balanced all my datasets, this shouldn't be a problem for my model.
2. Label Quality and Distribution: Erroneous or Misleading Labels: The existence of incorrectly labelled instances in the training dataset like a "TRUE" or "MISLEADING" assertion inaccurately classified as "FALSE" can lead the model to learn inappropriately. When the model encounters conflicting associations tied to TRUE/MISLEADING, its confidence in accurately recognizing them diminishes, causing it to lean towards categorizing them as FALSE.
3. Modelling Challenges like absence of class weighting in training: When the model is trained with standard cross-entropy loss without modifications for class imbalance, it tends to favour the majority class. Incorporating class weights, determined inversely to the frequency of classes, can help mitigate this bias and promote more equitable learning.

Apart from that, the model may just be overfitting the small datasets or many training epochs, the model may be overfit to superficial patterns found primarily in "FALSE" claims.

In addition to that, most probably the main reason could only be that if the base model wasn't adequately pretrained on related data, or if fine-tuning steps lacked validation checkpoints and learning rate tuning, it may fail to form good representations for TRUE or MISLEADING claims.

My confusion matrix shown below could confirm the same:



But, apart from that, there could also be the reason for poor/inconsistent tokenization all over the code. If certain tokens appear mostly in FALSE claims, the tokenizer might skew representation weights toward those examples, enhancing class imbalance during input representation.

Thank you for going through my documentation of fine tuning Falcon 2 7b for medical misinformation detection.