NLP CHALLENGE

Tensor Decomposition presents

CARE

Customer Assistance and Response Engine
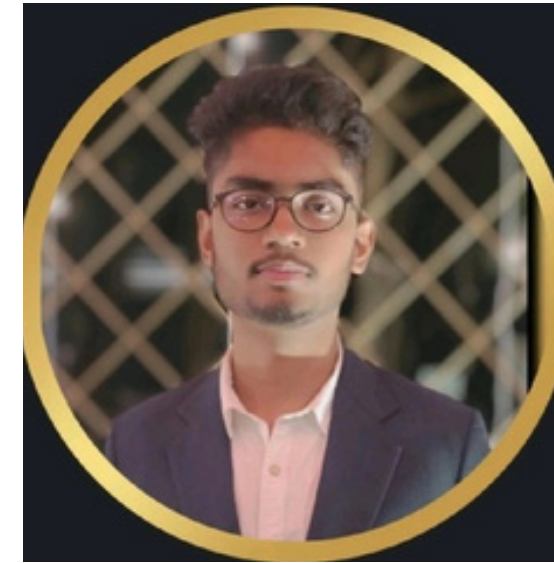
Meet Our Team

Nabarup Ghosh

Ankit Dutta
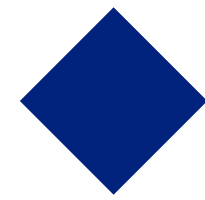

Ankush Chatterjee

Overview

Problem Statement

To create a **Smart Customer Support Chatbot** powered by **NLP** that can handle diverse customer inquiries, offer **empathetic responses** based on **user sentiment**, and escalate unresolved issues. The chatbot should be versatile enough to serve industries such as **e-commerce**, **banking**, **healthcare**, or **telecommunications**.

Our Solution: CARE

```
    "role": "user",
    "content": "I am having depression what should i do? \n"
  }
]
# Extract and print the response
print(text.split("assistant")[1])
```

```
|> Depression is a common mental health condition that can affect your mood, thoughts, and daily activities. Here
are some steps you can take to manage your depression:

1. Seek professional help: Consult a mental health professional, such as a psychiatrist or psychologist, to discus
s your symptoms and receive a proper diagnosis.
2. Medication: Your doctor may prescribe antidepressant medications to help regulate your mood and alleviate sympt
oms.
3. Therapy: Engage in therapy, such as cognitive-be-havioral therapy (CBT) or interpersonal therapy, to address un
derlying issues and develop coping strategies.
4. Lifestyle changes: Incorporate regular exercise, a balanced diet, and sufficient sleep into your routine.
5. Social support: Connect with friends and family for emotional support.
6. Mindfulness and relaxation techniques: Practice meditation, deep breathing exercises, or yoga to reduce stress
and improve your mental well-being.
7. Avoid alcohol and drugs: Limit or avoid substances that can worsen your symptoms.
8. Join a support group: Connect with others who are experiencing similar challenges to share experiences and copi
ng strategies.
```
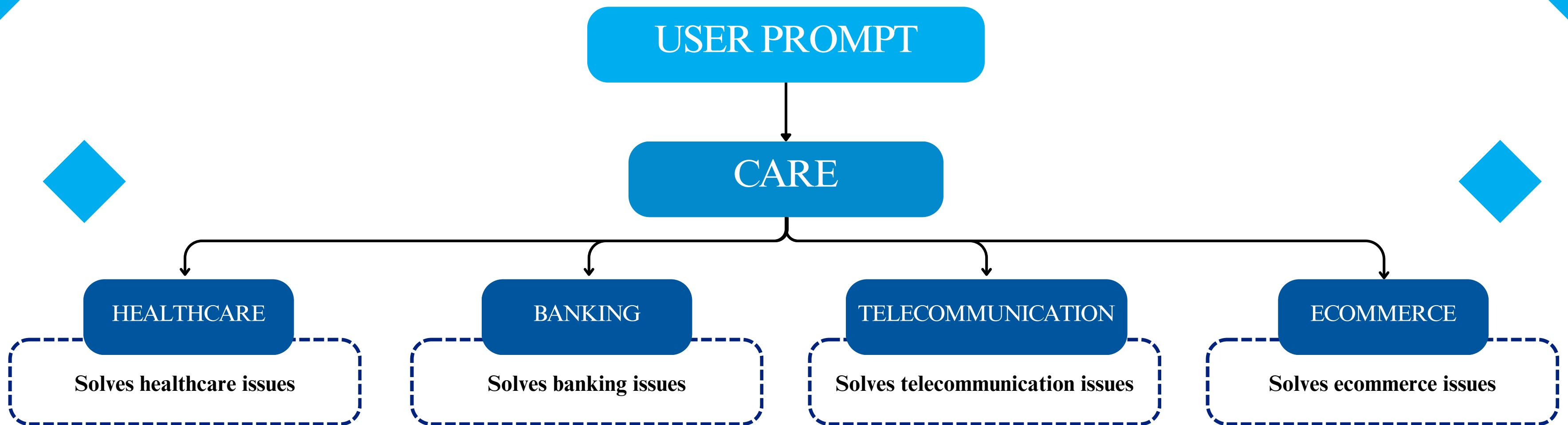
CARE (Customer Assistance and Response Engine) is an advanced chatbot powered by **Phi 3.5 LLM**, designed to deliver **efficient**, **empathetic**, and **domain-specific** support across medical, telecom, banking, and e-commerce sectors.
It automates customer interactions, analyzes sentiment, and ensures seamless issue resolution.
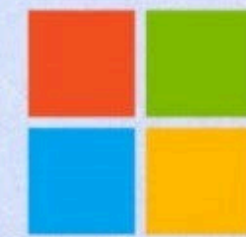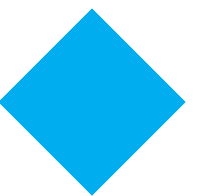
Execution Framework

Methodology
phi-3.5-mini-instruct

- **Phi 3.5 Mini Instruct:** A compact language model with approximately 3.8 billion parameters.
- Optimized for efficient fine-tuning and deployment.
- Uses **instruction-tuned training** for improved prompt adherence and contextual understanding.
- **Built on transformer architecture,** balancing performance and computational efficiency.
- **Size Benefits:**
  - Enables faster inference.
  - Reduces memory requirements.
- **Ideal Usage:**
  - Suitable for resource-constrained settings.
  - Capable of generating coherent, domain-specific chatbot responses.

Phi 3.5

Dataset

Banking:
https://www.kaggle.com/datasets/shashwatwork/consume-complaints-dataset-fo-nlp

Medical:
https://huggingface.co/datasets/ruslanmv/ai-medical-chatbot?row=0

Telecommunication:
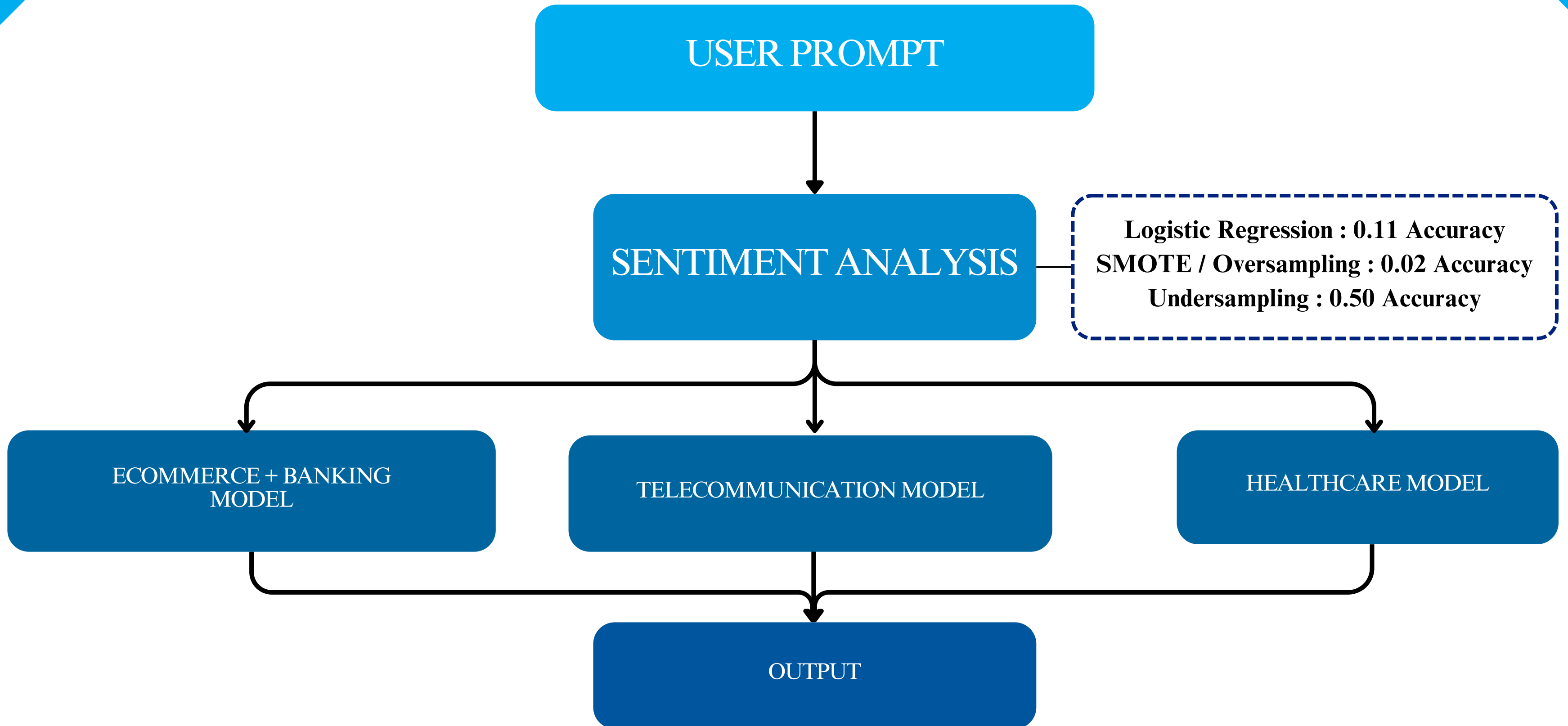https://huggingface.co/datasets/bitext/Bitext-telco-llm-chatbot-training-dataset

Ecommerce :
https://github.com/imsoumya18/E-commerce_FAQ/blob/main/Ecommerce_FAQs.csv + Custom Made

Approach 1

(Not feasible due to **poor classification results**)

```
┌─────────────────────┐
│    USER PROMPT      │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐        ┌──────────────────────────────────────┐
│ SENTIMENT ANALYSIS  │────────│ Logistic Regression : 0.11 Accuracy  │
└─────────────────────┘        │ SMOTE / Oversampling : 0.02 Accuracy │
           │                   │ Undersampling : 0.50 Accuracy        │
           │                   └──────────────────────────────────────┘
```

| ECOMMERCE + BANKING MODEL | TELECOMMUNICATION MODEL | HEALTHCARE MODEL |

OUTPUT

Approach 2

(Not feasible due to **high memory requirement**)

```
                    ┌──────────────────────┐
                    │     USER PROMPT       │
                    └──────────────────────┘
                               │
                               ▼
                    ┌──────────────────────┐
                    │     LLAMA TEXT        │
                    │   CLASSIFICATION      │
                    └──────────────────────┘
```

ECOMMERCE + BANKING MODEL	TELECOMMUNICATION MODEL	HEALTHCARE MODEL

OUTPUT
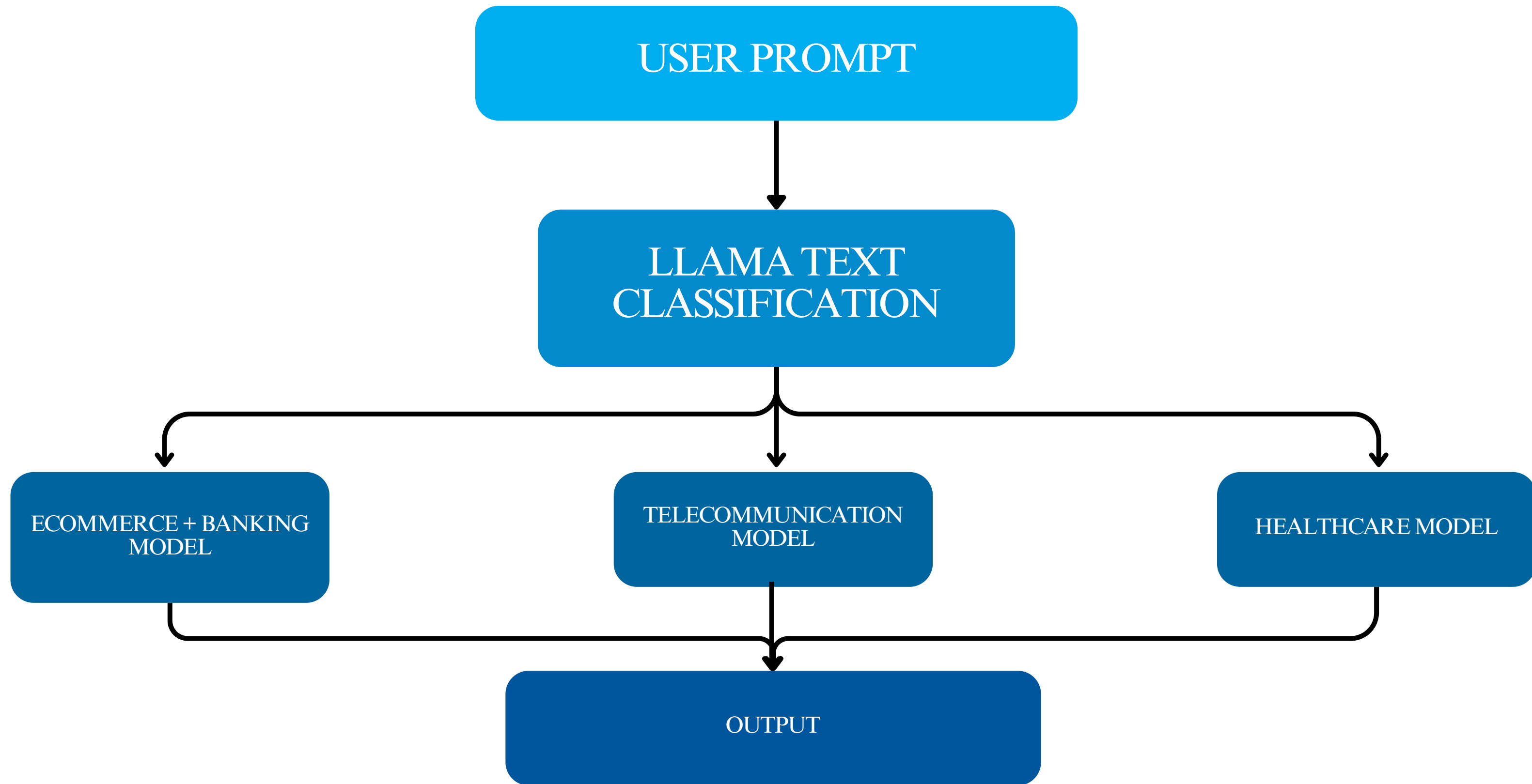
Approach 3

(Not feasible due to **low accuracy**)

Final Approach

| HEALTHCARE (HC) | BANKING (BC) | ECOMMERCE (EC) | TELECOMMUNICATION (TC) |

USER PROMPT

CONCAT

MERGING FINETUNED MODEL TO ITS MOTHER MODEL ← **QLORA FINETUNING PHI 3.5** ← **DOWNSIZING HC/TC DATA** ← **MAIN DATASET**

To avoid skewness

OUTPUT
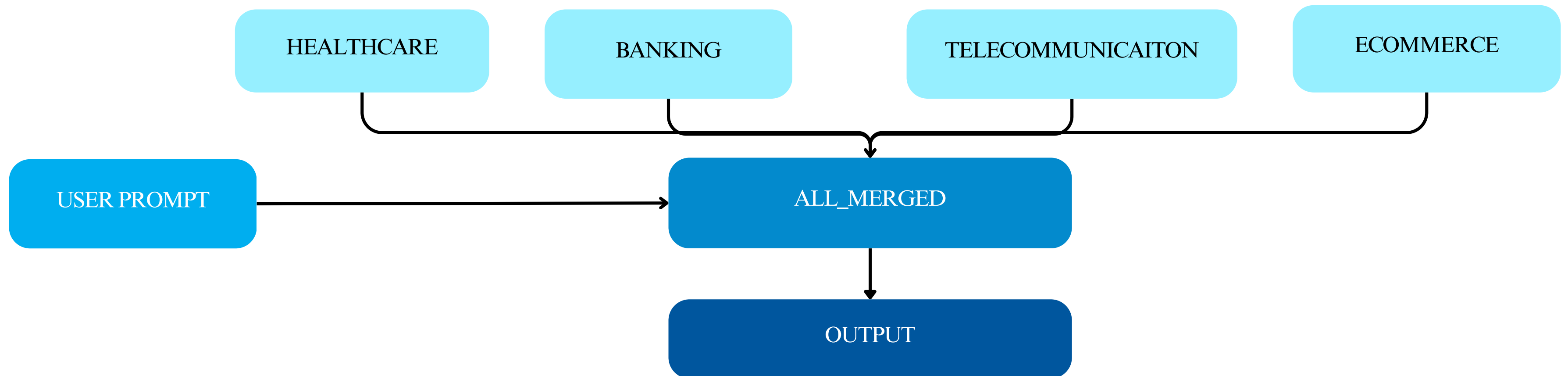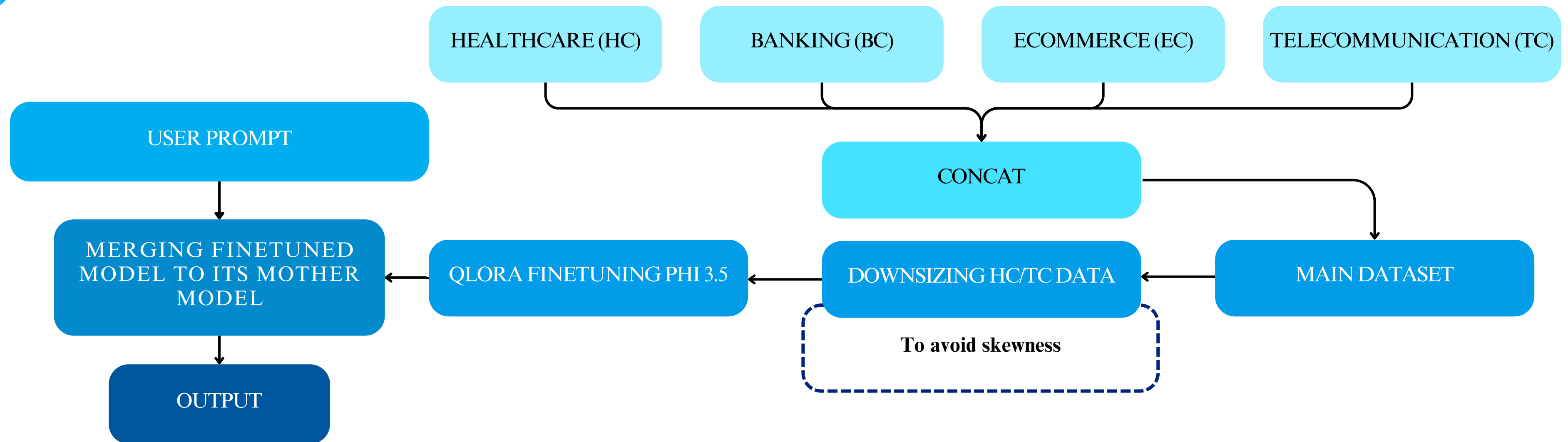
Final Model Made: ankitruler567/Phi-3.5-mini-instruct-Combined-Model
Final Testing Notebook: https://www.kaggle.com/code/ankitd7752/testing-production-code-tensor-decomposition

User Prompts

Banking Prompt

```python
messages = [
    {
        "role": "user",
        "content": "Hi, what documents do i need to open a bank account? \n"
    }
]

# Apply the chat template to the message
prompt = tokenizer.apply_chat_template(messages, tokenize=False,
                                       add_generation_prompt=True)

# Tokenize and move inputs to GPU
inputs = tokenizer(prompt, return_tensors='pt', padding=True,
                   truncation=True).to("cuda")

# Move the model to GPU
model = model.to("cuda")

# Generate outputs
outputs = model.generate(**inputs, max_length=150, num_return_sequences=1)

# Decode the output
text = tokenizer.decode(outputs[0], skip_special_tokens=True)

# Extract and print the response
print(text.split("assistant")[1])
```

|> To open a bank account, please follow these steps:

1. Visit our website at {{WEBSITE_URL}} or go to a branch in person.
2. Choose the type of account you wish to open (e.g., checking, savings, or credit card).
3. Complete the online application form or fill out the necessary paperwork at the branch.
4. Provide the required identification documents, which typically include:
    - A valid government-issued photo ID (e.g., driver's license, passport).
    - Proof of address (e.g., utility bill

Healthcare Prompt

```python
messages = [
    {
        "role": "user",
        "content": "I am having a severe headache, what medications do I need to take? \n"
    }
]

# Apply the chat template to the message
prompt = tokenizer.apply_chat_template(messages, tokenize=False,
                                       add_generation_prompt=True)

# Tokenize and move inputs to GPU
inputs = tokenizer(prompt, return_tensors='pt', padding=True,
                   truncation=True).to("cuda")

# Move the model to GPU
model = model.to("cuda")

# Generate outputs
outputs = model.generate(**inputs, max_length=150, num_return_sequences=1)

# Decode the output
text = tokenizer.decode(outputs[0], skip_special_tokens=True)

# Extract and print the response
print(text.split("assistant")[1])
```

|> To manage your severe headache, please adhere to the following steps:

1. Take 500 mg of ibuprofen orally every 6 to 8 hours as needed for pain relief.
2. Ensure you are well-hydrated by drinking plenty of water throughout the day.
3. Rest in a quiet, dimly lit room to minimize headache triggers.
4. Apply a cool compress to your forehead or the back of your neck to help alleviate discomfort.
5. If your headache persists or worsens, consult a healthcare professional for further evaluation and treatment.

If you have any additional questions or require further assistance, please do not hesitate to reach out. <|end|>

User Prompts

Ecommerce Prompt

```python
messages = [
    {
        "role": "user",
        "content": "How do I return or exchange a product? \n"
    }
]

# Apply the chat template to the message
prompt = tokenizer.apply_chat_template(messages, tokenize=False,
                                       add_generation_prompt=True)

# Tokenize and move inputs to GPU
inputs = tokenizer(prompt, return_tensors='pt', padding=True,
                   truncation=True).to("cuda")

# Move the model to GPU
model = model.to("cuda")

# Generate outputs
outputs = model.generate(**inputs, max_length=350, num_return_sequences=1)

# Decode the output
text = tokenizer.decode(outputs[0], skip_special_tokens=True)

# Extract and print the response
print(text.split("assistant")[1])
```

```
|> To return or exchange a product, please follow these steps:

1. Log into your account on our website.
2. Navigate to 'My Orders' and select the order you wish to return.
3. Choose 'Return or Exchange' and follow the instructions provided.
4. If required, print the return label and attach it to your package.
5. Drop off the package at the designated return center or schedule a pickup if available.

If you encounter any issues, please contact our customer support team for further assistance. <|end|>
```

Telecommunication Prompt

```python
messages = [
    {
        "role": "user",
        "content": "How do I enable international roaming on my number? \n"
    }
]

# Apply the chat template to the message
prompt = tokenizer.apply_chat_template(messages, tokenize=False,
                                       add_generation_prompt=True)

# Tokenize and move inputs to GPU
inputs = tokenizer(prompt, return_tensors='pt', padding=True,
                   truncation=True).to("cuda")

# Move the model to GPU
model = model.to("cuda")

# Generate outputs
outputs = model.generate(**inputs, max_length=350, num_return_sequences=1)

# Decode the output
text = tokenizer.decode(outputs[0], skip_special_tokens=True)

# Extract and print the response
print(text.split("assistant")[1])
```

```
|> To enable international roaming on your number, please follow these steps:

1. Log in to your account on {{WEBSITE_URL}}.
2. Navigate to the {{ROAMING_SECTION}} section.
3. Select the option to enable international roaming.
4. Follow the on-screen instructions to complete the process.

If you encounter any issues, please contact our customer support team for further assistance. <|end|>
```

Possible Improvements

Some possible improvements that we felt could be made for our chatbot

- **Multi-Language Support**: Expand the chatbot's capability to handle queries in multiple languages for global accessibility.
- **Voice Integration:** Implement real-time voice-to-text and text-to-voice features for better accessibility.
- **Using RLHF and RAG**: We can integrate out chatbot system with RLHF and RAG to continuously making our chatbot more and more accurate.
- **Knowledge Base Expansion**: Continuously update domain-specific databases to improve accuracy and relevance.
- **Deploying the model as an API:** We can deploy the model as an API to serve even a larger base of customers creating a more large impact.

Conclusion

At the end we would like to thank GENPACT and KSHITIJ for giving this amazing project to work upon which can be a really helpful solution for which would impact a lot of people in a positive way.
With this, Team Tensor Decomposition signs off!

Thank you