Educational Bot

BOSTON UNIVERSITY
CS688 FINAL PROJECT

Utilizing LLama 2 chatbot technology to deliver responses to educational questions, and recommending relevant articles leveraging LLM - based keyword analysis of chatbot responses.

Sanjana Prasad
U35186791

Motivation

- **Personalized Learning:** By analyzing chatbot responses and recommending relevant articles based on keyword analysis, learners can access supplementary materials tailored to their specific interests and learning goals.

- **Efficiency:** LLama 2 chatbots are capable of processing vast amounts of information quickly and accurately. This efficiency enables prompt responses to user queries, saving time for both students and educators.

- **Keeps Content Updated:** By leveraging keyword analysis to recommend articles, educators can ensure that learners have access to the latest and most relevant information in their field of study.

Installations

- **Hugging Face Transformers:** Provides us with a straightforward way to use pre-trained models.
- **PyTorch:** Serves as the backbone for deep learning operations.
- **Accelerate:** Optimizes PyTorch operations, especially on GPU.
- **Hugging face authentication:** To access private models and datasets in Hugging Face.
- **Beautifulsoup:** Library for web scraping, allowing you to extract data from HTML and XML files.
- **KeyBERT:** Stands for "Keyword Extraction with BERT". KeyBERT leverages BERT's contextual embeddings to extract keywords or key phrases from a given text.

Llama-2 7B model

- **Training:** Llama 2 was pretrained on 2 trillion tokens of data from publicly available sources. The fine-tuning data includes publicly available instruction datasets, as well as over one million new human-annotated examples.

- **Architecture:** Llama 2 is an auto-regressive language model that uses an optimized transformer architecture.

- **Variations:** Llama 2 comes in a range of parameter sizes — 7B, 13B, and 70B — as well as pretrained and fine-tuned variations.

Llama-2 7B model

- **Tokenization Process:** Text prompts are converted into tokens, which are numerical representations understandable by the model, enabling effective communication.

- **GPU Utilization:** Leveraging the GPU's computational power, all tokenized inputs are transferred to it for efficient processing, ensuring faster response times

- **Model Generation:** Our system generates responses based on the input tokens, using parameters like max_length to control response length and ensure coherent answers.

- **Memory Considerations:** We carefully manage memory resources, particularly in environments with constraints like free-tier Colab, where setting max_length too high may strain GPU memory. This ensures optimal performance within resource limitations.

Loading the model and tokenizer and creating the LLama pipeline

```python
from transformers import AutoTokenizer
import transformers
import torch

model = "meta-llama/Llama-2-7b-chat-hf" # meta-llama/Llama-2-7b-hf

tokenizer = AutoTokenizer.from_pretrained(model, use_auth_token=True)
```

The tokenizer will help convert text prompts into a format that the model can understand and process

```python
from transformers import pipeline

llama_pipeline = pipeline(
    "text-generation",  # LLM task
    model=model,
    torch_dtype=torch.float16,
    device_map="auto",
)
```

This pipeline simplifies the process of feeding prompts to our model and receiving generated text as output

By using 16-bit floating point format (float16), the memory usage during inference can be reduced compared to the default 32-bit floating point format.

Model parameters and output

```python
def get_llama_response(prompt: str) -> str:
    """
    Generate a response from the Llama model.

    Parameters:
        prompt (str): The user's input/question for the model.

    Returns:
        str: The model's response.
    """
# Generate sequences using the Llama pipeline
    sequences = llama_pipeline(
        prompt,  # User's input/question
        do_sample=True,  # Use sampling-based decoding
        top_k=10,  # Consider top-10 tokens during sampling
        num_return_sequences=1,  # Generate one sequence
        eos_token_id=tokenizer.eos_token_id,  # End-of-sequence token ID
        max_length=1000,  # Maximum length of the generated sequence
    )
    # Return the generated response
    return sequences[0]['generated_text']

prompt = 'Can you explain what dark matter is?\n'
doc2 = get_llama_response(prompt)
```

Sampling Tokens:
Instead of deterministically selecting the token with the highest probability (greedy decoding), sampling-based decoding involves randomly selecting tokens from this probability distribution according to their probabilities.

KeyBERT

- KeyBERT stands for "Keyword Extraction with BERT".
- KeyBERT utilizes pre-trained transformer-based models, such as BERT, to encode and understand the contextual information within the text.
- KeyBERT can automatically extract keywords or key phrases that represent the most important or informative terms within the text

```
from keybert import KeyBERT
```

+ Code + Text

```
model = KeyBERT(model="distilbert-base-nli-mean-tokens") #distilled version of the BERT model, which is smaller and faster but still maintains good performance
```

```
model.extract_keywords(
    doc2,
    top_n=10, #considers the top 10 keywords
    keyphrase_ngram_range=(1, 1), #1 word
    stop_words="english", #removes stopwords such as "the"
)
```

Keywords and relevance score:

```
[('astrophysicist', 0.4303),
 ('neutrinos', 0.384),
 ('telescopes', 0.3555),
 ('galaxies', 0.3535),
 ('cosmic', 0.3365),
 ('radiation', 0.3264),
 ('microwave', 0.3161),
 ('black', 0.2946),
 ('astronomical', 0.2933),
 ('invisible', 0.2776)]
```

Scraping for relevant articles

- The extracted keywords are utilized to filter the articles based on their titles, ensuring that only articles containing relevant keywords are included in the list of recommended articles.
- The code utilizes the BeautifulSoup library for parsing HTML content retrieved from a web page.
- It makes use of the requests library to send HTTP requests to the web server and retrieve the HTML content of the page.
- The code employs web scraping techniques to extract relevant information, such as article headings and URLs, from the parsed HTML.

Output & Streamlit UI

Educational Chatbot

Type your question:

Can you explain what dark matter is?

Chatbot Response:

Dark matter is a hypothetical form of matter that is thought to exist in the universe but cannot be directly detected by telescopes or other astronomical instruments. It is called "dark" because it does not emit, absorb, or reflect any electromagnetic radiation, such as light, making it invisible to our telescopes. Despite its elusive nature, dark matter's presence can be inferred through its gravitational effects on visible matter and the way galaxies and galaxy clusters move.

The existence of dark matter was first proposed by Swiss astrophysicist Fritz Zwicky in the 1930s, and since then, a wealth of observational evidence has accumulated to support its existence. Some of the key lines of evidence include:

1. Galactic rotation curves: The rotation curves of galaxies are the rate at which stars and gas orbit around the center of the galaxy. These curves are typically flat or even rise slightly towards the edges of the galaxy. This is unexpected, as the stars and gas in the outer regions of the galaxy should be moving slower than those closer to the center due to the decreasing gravitational pull. The flat rotation curves suggest that there is an unseen mass of dark matter surrounding the galaxy, which is responsible for the increased gravitational pull.
2. Galaxy clusters: Galaxy clusters are the largest structures in the universe, containing hundreds or even thousands of galaxies. The galaxies within these clusters are observed to be moving at high speeds, indicating that they are held together by a large amount of mass. However, the visible mass of the galaxies and hot gas within the cluster is not enough to account for the high speeds of the galaxies. Dark matter provides the additional mass necessary to hold the cluster together.
3. Large-scale structure of the universe: The universe is made up of vast networks of galaxy clusters and superclusters, which are separated by vast distances of empty space. The distribution of these structures on large scales is sensitive to the presence of dark matter. Simulations of the universe that include dark matter can reproduce the observed large-scale structure of the universe, while simulations without dark matter fail to do so.
4. Cosmic microwave background radiation: The cosmic microwave background radiation (CMB) is the leftover radiation from the Big Bang. The CMB is a snapshot of the universe when it was just 380,000 years old, and it contains a wealth of information about the universe's composition and evolution. The CMB is consistent with the presence of dark matter, as it suggests that the universe is made up of about 27% dark matter and 68% ordinary matter.

While the existence of dark matter is well established, its exact nature is still unknown. Dark matter could be composed of as-yet-undiscovered particles, such as weakly interacting massive particles (WIMPs), axions, or sterile neutrinos. Alternatively, dark matter could be made up of ordinary matter that is somehow "hidden" from our detection, such as in the form of black holes or neutrinos.

Recommended articles:

Scientists are one step closer to knowing the mass of ghostly neutrinos — possibly paving the way to new physics: https://www.livescience.com/physics-mathematics

Improvements

Llama 2:

Given enough computational power:

- Capable of experimenting with larger models like 17B and 70B.
- Can increase the max_length of output, enabling more detailed responses.
- Can use 32-bit floating point format instead of the 16-bit format for inference for improved accuracy.

KeyBERT:

- The model architecture is the base version, which generally contains fewer parameters compared to larger variants like "large" or "xlarge”, which can be explored.