# Advanced Chatbot Design
## (Using Cornell Movie Dialogs Corpus)

AAI 520

Final Project - Group 6

Adam Graves, Paul Parks, Alden Caterio

# Introduction

**Goal:**

Design and build a chatbot that can carry out multi-turn conversations, adapt to context, and provide a variety of answers related to movies.

Explain and demonstrate the evolution of chatbots from the past popular Retrieval-Based to the newer Generative Models

**Output:**

A web or app interface where users can converse with the chatbot, and ask questions and receive answers about movies taken from the Cornell movie data corpus we have trained the model against. The Retrieval-Based will be query based, while the Generative Model will be based on pre-trained LLM model, and able to have multi-turn conversations, adapt to context, and handle a variety of topics

# Dataset

The dataset consists of 617 movies with several files containing:

1. Five main files with basic movie info such as title, genre, release year, and IMBd rating
2. Character info such as character name and gender
3. Movie lines by a character
4. Conversations between characters
5. Full movie scripts (url link)

   https://www.kaggle.com/datasets/rajathmc/cornell-moviedialog-corpus

# Retrieval Based Chatbots

- The concept of retrieval-based chatbots has been around since the early days of artificial intelligence and natural language processing.
- Chatbot that generates responses by selecting pre-existing responses from a predefined set or database
- The chatbot then compares the input to its predefined database or knowledge base, which contains a collection of responses or potential answers. It uses techniques of NLP to identify the intent and assign to the lookup fields for database matching.
- Database query style with the use of If and Else statements.
- Use of utilizing pre-defined BERT models and Natural Language Understanding (NLU) to improve tokenizing and intent identification is used by our code.

# Generative Chatbot Model

- A **generative chatbot** can generate original combinations of language to respond to prompts rather than selecting from a database of predefined responses.
- Generative chatbot models utilize **transformers** and pre-trained **Large Language Models** (LLMs).
    - A transformer is applied to an LLM in order for the model to understand human language.

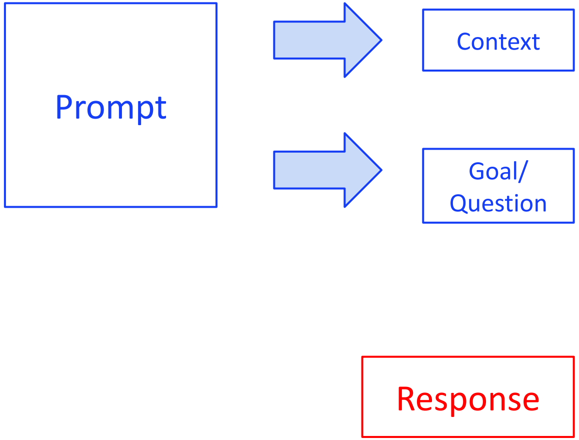
# Our Chatbot Model

- Our chatbot model uses the Meta Llama2 LLM
    - This LLM has been pre-trained on several billions of parameters.
- Using this model, our chatbot is able to respond to questions in a conversational manner.
- Utilizing HuggingFace libraries, we can further train, or "fine-tune" the model.
    - We generated a train.csv file that had questions and answers based on the Cornell Movie corpus.

# Training/Fine-Tuning

- **Fine-tuning** is the processing of applying additional training data to the model in an effort to steer the model towards a specific field of knowledge.
- The training data must be formatted with a prompt and a response.
    - The quality of the training data will affect how well the chatbot responds.
- Training parameters such as number of epochs and learning rate affect the chatbot performance.

# Sample of Training Data



<s>[INST] <<SYS>>

Below is an instruction that describes a movie related question. Write a response that appropriately answers the question using the Cornell Movie-Dialog Corpus.

<</SYS>>

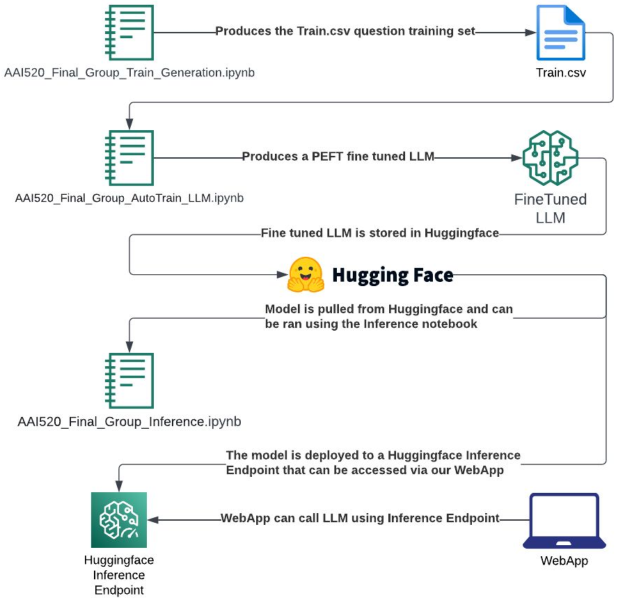
what is the genre of the movie zulu dawn?

[/INST]

according to the Cornell Movie-Dialog Corpus, The genres for the movie zulu dawn are action, adventure, drama, history, and war

</s>

# LLM Chatbot Architecture



Produces the Train.csv question training set

Produces a PEFT fine tuned LLM

Fine tuned LLM is stored in Huggingface

Model is pulled from Huggingface and can be ran using the Inference notebook

The model is deployed to a Huggingface Inference Endpoint that can be accessed via our WebApp

WebApp can call LLM using Inference Endpoint

AAI520_Final_Group_Train_Generation.ipynb
Train.csv
AAI520_Final_Group_AutoTrain_LLM.ipynb
FineTuned LLM
Hugging Face
AAI520_Final_Group_Inference.ipynb
Huggingface Inference Endpoint
WebApp

# Hosting and Inference

- The fine-tuned LLM is stored in HuggingFace
    - https://huggingface.co/guitarnoob/msaai520_with_HF_format_v1
- The LLM is hosted and ran using the HuggingFace Inference Endpoint
- The Endpoint is utilized by our front end user interface
    - https://msaai-520-final-project-web-app.vercel.app/

# Multi-turn Conversation

- The LLM is not stateful and does not retain information about previous conversations.
- The WebApp uses the Llama2 question format to add all chat history into each request
    - <s>[INST] <<SYS>> {{ system_prompt }} <</SYS>> {{ user_msg_1 }} [/INST] {{ model_answer_1 }} </s><s>[INST] {{ user_msg_2 }} [/INST]

# Demonstration

# Collaborative Efforts

## Adam Graves

- Team leader; organized meetings, discussions, and workflow
- Developed code for extracting information from the dataset and formatting into usable dataframes
- Developed code for creation of training data
- Tested chatbot functionality; documented findings
- Maintained GitHub documentation and files

## Paul Parks

- Researched multiple LLMs and chatbot methodologies
- Developed main body of code for chatbot training and deployment
- Researched and tested various methods of deploying the chatbot
- Developed online interface for easy chatbot interaction

## Alden Caterio

- Researched multiple LLMs and chatbot methodologies
- Contributed code for creation of training data
- Tested chatbot functionality; documented findings
- Contributed to overall documentation and reporting