# Capstone Project Title: Multi-lingual Chatbot for e-commerce start up venturing in the Kenyan Market

## 1.0 Project Overview

This project aims to develop a natural language chatbot capable of generating human-like responses and understanding informal customer feedback expressed in English, Kenyan Swahili, and Sheng.

Designed for a startup expanding into the Kenyan market, the chatbot will help the company engage users more naturally and analyze feedback from social platforms and online conversations. By training on locally relevant dialogue data — including YouTube comments and Kenyan media — the system will capture the linguistic and cultural nuances often missed by standard models.

## 1.1 Problem Statement

Startups entering new markets often struggle to understand customer feedback when it's expressed in local dialects or informal language. In Kenya, much of this communication occurs in Swahili and Sheng — a blend of local slang, English, and Swahili spoken fluidly and often without strict structure.

Existing chatbot systems trained on formal English frequently fail to grasp the tone, intent, or meaning behind such messages. This project aims to bridge that gap by building a chatbot trained specifically on real-world Kenyan conversations, enabling it to interpret and respond to customer queries and feedback with local context and cultural relevance.

## 1.2 Objectives

- Collect and preprocess Kenyan user dialogue from YouTube, social media, and local content featuring Swahili and Sheng.
- Fine-tune the chatbot using foundational dialogue data while emphasizing local language patterns.
- Build a sequence-to-sequence model capable of handling informal, code-switched conversations.
- Evaluate the chatbot's performance with emphasis on contextual relevance and local linguistic understanding.
- Present a working prototype that simulates real-world customer feedback scenarios.

## 2.0 Data Scraping and understanding

## 📁 Data Sources

This project uses data from two primary sources:

### 1. Cornell Movie Dialogues Corpus

A dataset of fictional conversations extracted from movie scripts. We used two key files:

- `movie_lines.txt` : Contains individual lines of dialogue.
- `movie_conversations.txt` : Contains sequences of line IDs that form complete conversations.

### 2. YouTube Comments from Kenyan Local Content Channels

A collection of YouTube comments scraped from videos that feature local Kenyan content. These comments reflect more natural, informal, and code-switched language patterns (English, Swahili, Sheng).

# 2.1 📊 Data Understanding

## Cornell Movie Dialogs Dataset:

- Loaded and inspected datasets
- Read `movie_lines.txt` and `movie_conversations.txt`
- Parsed line-level and conversation-level files
- Reconstructed conversations from line IDs
- Viewed sample conversations
- Saved reconstructed conversations into the Chatbot repository

## YouTube Comments Dataset:

- Merged YouTube datasets from multiple Kenyan channels
- Pre-processed the YouTube comments and saved them in the Chatbot repo
- Analyzed comment length distribution
- Examined the top words in the comments

## Dataset Combination:

- Combined YouTube and Cornell datasets for downstream chatbot training

# 2.1. Language Detection and Pre-processing of YouTube Data

- Checked the quantity of Swahili/Sheng in the YouTube dataset to assess its suitability for training.
- Swahili/Sheng word percentage: **0.46%**
- Used `langdetect` to detect Swahili words:
  - Code-switched comments (Swahili-English): **1.79%**

> ⚠️ **Low prevalence** of Swahili/Sheng (less than 0.5%) in the dataset.

This suggests:

- Most comments are in English.
- The Swahili/Sheng vocabulary list may be too limited to capture Swahili/Sheng expressions accurately.
- Manual inspection of code-switched phrases shows that only a small fraction of the data contains both English and Swahili in the same comment.

## 2.2 Introducing Dataset with More Swahili Words

To increase the prevalence of Swahili in the dataset:

- Loaded the `allenai/c4` dataset (Swahili subset) from Hugging Face.
- Merged the combined YouTube dataset with the additional Swahili dataset.
- Shuffled the merged dataset to ensure better mixing of both sources, helping prevent model overfitting to source structure/order.

### 📊 Dataset Sizes:

- Original YouTube rows: **26,095**
- CC100 (Swahili) sample rows: **9,857**
- Combined total rows: **35,952**

## 🧹 Cleaning & Preprocessing the Combined Dataset

- Dropped rows where the `comment` field is missing.
- Stripped text, converted to lowercase, removed URLs, and cleaned extra whitespace.
- Recalculated comment length to support filtering (e.g., extremely short or long comments).
- Detected language if it was missing.
- Filled missing `response` values with `None` (if column exists).
- Filled missing `code-switched` flags with `False`.
- Dropped unnecessary columns.
- Reset the index for a clean final dataset.

## 2.3 Preparing input-reponse pairs from the Cornell dataset.

- Defining file paths
- Loading and parsing movie_lines.txt
- Loading and parsing movie_conversations.txt
- Creating comment-response pairs
- Creating a DataFrame
- Cleaning paired Dataframe - lowercase, Punctuations, extra whitespaces

# 3. Modeling

This stage involves designing and implementing the core logic that enables the chatbot to understand user queries and generate relevant responses. We used the TF-IDF and Cosine Similarity as the baseline model, follwed by a retrieval model and fine-tunned with mT5 for conversational dialogues

## 3.1 TF-IDF and Cosine Similarity

TF-IDF (Term Frequency–Inverse Document Frequency) - a statistical measure that reflects how important a word is to a document in a collection. Cosine Similarity measures how similar two texts (like a user query and a response) are in terms of their word usage.

Cosine Similarity measures how similar two texts (like a user query and a response) are in terms of their word usage. Its computed between the prompts-response and selects the response with the highest similarity score.

# 3.2. Retrieval Model

Selects the most appropriate response from a pre-defined list of responses. Does not generate new text. When prompted, it retrieves the best-matching response using similarity metrics (like cosine similarity)

# 3.3 Fine-Tuning mT5 for Conversational Generation

**mT5** is a multilingual version of the T5 (Text-To-Text **Transfer Transformer**) model. It is pre-trained on many languages and can handle a variety of tasks such as text generation, translation, and question answering.

Fine-tuning mT5 on a conversational dataset enables the model to learn how to generate responses that are:

- Contextually appropriate
- Grammatically fluent
- Aligned with the chatbot's domain

**Fine-tuning was done using** `accelerate>=0.26.0` , a Hugging Face library for training transformer models.
The fine-tuning process included:

- Encoding the input and candidate responses
- Computing cosine similarities

## 3.4 Evaluation

Evaluation was done using two main approaches:

### Semantic Similarity Evaluation

- Sample sentence pairs and scores
- Human-labeled similarity scores
- Evaluated using Pearson and Spearman cosine similarity

## 3.5 Deployment

- Gradio Interface
- Streamlit platform