NLP Workshop Practice-Session

IIT Bhilai Data Science Workshop

July 7, 2025

Abstract

This document contains four mini-project assignments designed to test your understanding of the concepts covered in the Natural Language Processing workshop. Each project can be completed within a Google Colab environment. You are provided with an objective, a hint to guide your approach, and a link to a relevant dataset. Good luck!

1 Project 1: Movie Review Sentiment Analysis

Objective

Build a machine learning model to classify movie reviews as either positive or negative. You must use **classic machine learning techniques** for this task, not deep learning. The goal is to practice the fundamentals of the NLP pipeline: text cleaning, feature extraction, and model training.

Hint

Your pipeline should look like this:

1. Load and clean the text data (lowercase, remove stopwords, etc.).
2. Convert the cleaned text reviews into numerical features. A great choice for this is the `TfidfVectorizer` from the `scikit-learn` library.
3. Train a simple and effective classifier on these TF-IDF features. Good models to try are `Naive Bayes` (`MultinomialNB`) or `Logistic Regression`.
4. Evaluate your model's accuracy on a test set.

Dataset

IMDB Dataset of 50K Movie Reviews: This is a classic, balanced dataset perfect for binary sentiment classification.

- **Kaggle Link:** https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-

---

2 Project 2: SMS Spam Prediction with LSTMs

Objective

Develop a deep learning model using a Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) to predict whether an SMS message is ”spam” or ”ham” (not spam). This project will help you understand how to handle sequential data for classification.

Hint

Your deep learning workflow will be:

1. Clean the SMS text.
2. Tokenize the text messages and build a vocabulary of all unique words. Convert each message into a sequence of integers based on this vocabulary.
3. Since RNNs/LSTMs require inputs of the same length, pad all sequences to a fixed length using `pad_sequences` from `TensorFlow/Keras`.
4. Build your model in Keras. It should start with an `Embedding` layer, followed by an `LSTM` or `SimpleRNN` layer, and end with a `Dense` output layer with a sigmoid activation for binary classification.

Dataset

SMS Spam Collection Dataset: A public set of SMS messages tagged as spam or ham. It's small, clean, and ideal for this task.

- **Kaggle Link:** https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

---

3 Project 3: English to Hindi Machine Translation

Objective

Use a modern, pre-trained Transformer model to translate sentences from English to Hindi. The goal is not to train a model from scratch, but to learn how to effectively use the powerful models available through the Hugging Face ecosystem.

Hint

1. The easiest way to accomplish this is by using the `pipeline` function from the Hugging Face `transformers` library.

2. You can initialize a translation pipeline directly by specifying the task and the model. For English to Hindi, a great model to use is `Helsinki-NLP/opus-mt-en-hi`.

3. The pipeline would be initialized like this: `pipeline(’translation_en_to_hi’, model=’Helsinki-NLP/`

4. Use the provided dataset to test your translation pipeline on a few sample sentences and observe the quality of the output.

Dataset

IIT Bombay English-Hindi Parallel Corpus: This is a standard dataset used for translation tasks between English and Hindi.

- **Hugging Face Datasets Link:** https://huggingface.co/datasets/cfilt/iitb-english-hindi

---

4 Project 4: FAQ Chatbot using an LLM

Objective

Create a simple Q&A chatbot that can answer questions based on a provided knowledge base (an FAQ dataset). This project introduces you to the powerful concept of **Retrieval-Augmented Generation (RAG)**, where an LLM’s knowledge is supplemented with external data.

Hint

Instead of just asking an LLM the question directly, follow the RAG approach:

1. **Load Knowledge Base:** Load the provided FAQ dataset into a list or pandas DataFrame. This is your ”knowledge”.

2. **Retrieve:** When a user asks a question (e.g., "How can I improve my sleep?"), your first step is to *find* the most similar question-answer pair from your knowledge base. You can do this by using TF-IDF or sentence embeddings to calculate similarity between the user's query and the questions in your FAQ.

3. **Augment & Generate:** Take the user's original question AND the most relevant FAQ entry you found. Feed both into an LLM (from Hugging Face or Ollama) with a specific prompt, such as:

   > "Using the following context, please answer the user's question. Context: [paste the retrieved FAQ here]. User's Question: [paste the user's question here]."

   This forces the LLM to use your trusted data to form its answer.

Dataset

Mental Health FAQ Dataset: Contains a list of questions and answers related to mental health topics, perfect for a focused chatbot.

- **Kaggle Link:** `https://www.kaggle.com/datasets/narendrageek/mental-health-faq-for-chatbot`