

## **BAX 452 HW 6**

Course Instructor : Dr. Rahul Makhijani

Due Date : 4th Mar 2025 @ 11:59PM PST

Please note: You are supposed to submit the solutions by **4th Feb 11:59 pm PST**. Refer to the course syllabus regarding late HW guidelines.

### **Question 1: English to Spanish Translation with Pre-trained Transformers (30 points)**

In this task, you will explore machine translation using pre-trained transformer models from the Hugging Face library. The goal is to perform English-to-Spanish translations and analyze the performance of these models.

#### **(a) Setup (5 points)**

- Environment Preparation: Use Google Colab as the execution environment for this task. Set up the necessary libraries, including the Hugging Face Transformers library. Make sure that the required dependencies (e.g., transformers, torch) are installed in the Colab notebook.
  - Model Selection: Choose a pre-trained English-to-Spanish translation model of your choice from the Hugging Face model hub, such as `Helsinki-NLP/opus-mt-en-es`. Explain why this model is a good choice for the task.
- 

#### **(b) Implementation (10 points)**

- Model Loading: Load the chosen pre-trained model and tokenizer for English-to-Spanish translation from the Hugging Face library.
- Translation Function: Implement a Python function `translate_to_spanish(sentence)` that accepts an English sentence as input and

outputs its Spanish translation. Ensure the function integrates the tokenizer and model inference pipeline.

- **Batch Processing:** Extend the function to handle a list of sentences and return their respective translations. Use padding or truncation to ensure compatibility with the model's input size.
  - **Example Usage:** Demonstrate the function with a few English sentences and display the Spanish translations. Include at least three examples in the notebook.
- 

### **(c) Evaluation (10 points)**

- **Dataset Creation:** Create a test dataset of at least 10 diverse English sentences. The dataset should include simple, complex, and idiomatic expressions to test the model's robustness.
  - **Model Testing:** Use the implemented function to translate the test dataset and display the results alongside the original English sentences.
  - **Analysis:** Evaluate the quality of translations. Focus on:
    - Accuracy of simple translations.
    - Handling of complex sentence structures.
    - Translation of idiomatic expressions.
  - **Metrics:** Compute BLEU scores (Bilingual Evaluation Understudy score is a metric for evaluating the quality of text generated by machine translation models. It measures the similarity between the machine-generated translation and a set of human reference translations) and other relevant metrics to quantitatively assess the translations. Discuss any observed strengths or weaknesses.
- 

### **(d) Analysis (5 points)**

- **Advantages:** Discuss the benefits of using pre-trained models for translation tasks. For instance, address how pre-trained models reduce computational requirements and enable rapid prototyping.

- **Limitations:** Identify limitations of the pre-trained model. Consider factors like:
  - Handling domain-specific or rare vocabulary.
  - Limitations in idiomatic or culturally nuanced translations.
- **Suggestions for Improvement:** Provide suggestions for improving the translation pipeline. For example, consider techniques like fine-tuning the model on a domain-specific dataset or augmenting the data.

## **Question 2: Building a Custom Chatbot with Hugging Face (70 points)**

For this task you need to explore Hugging Face, and create a customized chatbot for a domain-specific application of your choice (health/finance/tax etc.), and evaluate its performance.

### **(a) Model Implementation and Training (50 points)**

#### **1. Model Selection and Fine-Tuning:**

- Choose a pre-trained models from Hugging Face suitable for specific applications (e.g., healthcare, finance, or tax assistance). Document their:
  - Key features (e.g., transformers, attention mechanisms).
  - Strengths and potential weaknesses in handling domain-specific tasks.
- Fine-tune the models on a domain-specific dataset of your choice. Describe the fine-tuning process, including:
  - Optimizer choice (e.g., Adam, SGD).
  - Learning rate and batch size settings.
  - Number of epochs and stopping criteria.

#### **2. Loss Function and Metrics:**

- Choose an appropriate loss function for the chatbot's response generation task. Explain your choice.
- Identify evaluation metrics (e.g., BLEU, user satisfaction score) and justify their relevance.

### 3. Multi-Turn Dialogue Handling:

- Implement mechanisms to maintain context across multiple turns in a conversation.
  - Explain how the architecture supports multi-turn dialogue and handles potential pitfalls like irrelevant responses.
- 

## (b) Evaluation and Analysis (15 points)

### 1. Evaluation:

- Evaluate the accuracy of fine-tuned models using metrics like BLEU.
- Provide sample dialogues generated by the chatbot in your chosen application.
- Highlight improvements or limitations observed in the responses.

### 2. Comparison with Base Model:

- Discuss the trade-offs in terms of accuracy, response relevance, and computational efficiency.
- 

## (c) Future Enhancements (5 points)

### 1. Error Analysis:

- Identify common errors or limitations in the chatbot's responses.
- Suggest potential improvements, such as:
  - Expanding the training dataset.
  - Knowledge distillation.

### 2. Scalability Considerations:

- Discuss how the chatbot could be scaled to handle large datasets or deployed in real-world applications.