TEAM 17 : Valyrian-to-English Benchmark for Translation Tasks

I. Introduction

This project introduces a **benchmark for Valyrian-to-English translation** to evaluate neural machine translation models on fictional and non-traditional language. The benchmark includes:

1. A **parallel corpus** of Valyrian sentences paired with English translations. These sentences were manually generated using linguistic rules and annotations to create a challenging dataset.
2. A baseline **LSTM-based encoder-decoder model** to serve as a reference point for future comparisons.
3. Evaluation metrics, including **BLEU Score** and **Perplexity**, to assess model performance.

II. Motivation

The motivation behind this benchmark stems from the need to test and refine translation models on languages with limited datasets. Valyrian, a constructed language from the fictional world of Game of Thrones, provides a challenging test case due to its unique structure and limited vocabulary. It provides an opportunity to explore how well machine translation systems adapt to non-standard linguistic inputs. The dataset and baseline results encourage further research and experimentation, enabling comparisons between different models.

III. Dataset

Data Collection :

- **Valyrian vocabulary :** The vocabulary was first compiled using a PDF file titled *'High Valyrian Glyph Dictionary,'* which can be found here. It served as the initial source. This information was then used to complete the file *'glyph.txt,'* where Valyrian-English word mappings were manually recorded. Using regular expressions, the vocabulary was extracted from *'glyph.txt'* to identify the Valyrian words and their English equivalents. Finally, the data was converted into a JSON format, with each entry mapping a Valyrian word to its English translation.
- **Parallel corpus :** Using the vocabulary file, we generated a structured parallel corpus with Valyrian sentences and their English translations. This corpus was constructed based on grammatical rules (e.g., Subject-Verb-Object or Subject-Object-Verb structures) and enriched with linguistic annotations (e.g., parts-of-speech tags) using SpaCy.

Each entry in the vocabulary was annotated with its part of speech (POS) using SpaCy's English model. The annotations were then used to construct sentences with varied syntactic structures. This method ensured that generated sentences were grammatically consistent

.

Dataset Size

The dataset includes:

- **Vocabulary size** : 706 word mappings between Valyrian and English.
- **Corpus size** : 50000 parallel sentences across different syntactic structures. This size was chosen to balance computational efficiency and statistical robustness. The main bottleneck in scaling was the manual validation of translations and the creation of diverse sentence templates.

Data Loading

To load the data, the code is in the cell 'Load the data' of the Notebook 'Valyrian_English_Translation.ipynb'.

IV. Experimental Setup

We compare our baseline model, an **LSTM-based** encoder-decoder architecture for Valyrian-to-English translation, with **Google T5 Small Model** with about 60 million parameters. We also tried with Meta LLama 3 but we ran out of resources.

Baseline model architecture

The Valyrian-to-English translation task is performed using an encoder-decoder architecture with Long Short-Term Memory (LSTM) networks. The encoder processes the input sequence (Valyrian sentences) and generates a context vector, which is used by the decoder to predict the output sequence (English translations).

The model uses an encoder-decoder architecture, where the encoder consists of an embedding layer (256-dimensional vectors) and an LSTM layer (512 hidden size) to process input sequences. The decoder has a similar embedding layer, followed by an LSTM layer that generates output sequences based on the encoder's context and previous predictions, with a dense layer applying softmax to predict the next token.

For data preparation, both Valyrian and English sentences were tokenized using Keras' Tokenizer. The sentences were then padded with pad_sequences to ensure a uniform input size. The dataset was split into 80% for training and 20% for testing to evaluate the model's performance.
In terms of training configuration, the Adam optimizer with the default learning rate was used, and the loss function was Sparse Categorical Crossentropy with masking to ignore padded tokens. The training process utilized a batch size of 16 samples per batch and ran for 5 epochs over the entire training dataset.

The training loop updates the encoder and decoder parameters to minimize the loss between the predicted and actual sequences. The decoder predicts tokens one step at a time, using teacher forcing by providing the actual previous token during training.

Evaluation Metrics

The benchmark evaluates translation models using:

- BLEU Score : Measures the overlap between predicted and reference translations at the n-gram level, emphasizing precision.
- Perplexity : Assesses the model's confidence in its predictions by calculating the exponential of the average negative log-likelihood.

These metrics were chosen because they provide complementary insights into model performance: BLEU focuses on accuracy, while perplexity evaluates model uncertainty.

V. Results :

Model	BLEU Score	Perplexity
Baseline Model	0.2617	1.4944
Google T5	0.3922	1.4335

BLEU Score: A BLEU score of 26.2% indicates that the model captures some level of alignment between predicted translations and reference sentences, but there is still significant room for improvement. This score reflects the model's partial ability to handle the unique sentence structures and linguistic patterns of Valyrian. The Google T5 model, with a higher BLEU score of **39.2%**, demonstrates improved alignment and translation quality compared to the baseline.

Perplexity: A perplexity of 1.49 suggests that the model is relatively confident in its predictions. This low perplexity demonstrates that the model performs well on the given test data but does not necessarily reflect true generalization. The slightly lower perplexity of **1.43** for Google T5 indicates even greater confidence in its predictions, suggesting that it better captures the relationships within the data.

Interpretation of Results

Strengths:

- The model shows a good degree of coherence when generating translations, achieving a low perplexity score.
- Despite the challenges of a low-resource dataset, the model was able to learn basic mappings between Valyrian and English.

Weaknesses:

- The BLEU score remains moderate, indicating that while individual words are often correct, sentence structure and grammatical accuracy are inconsistent.
- The model struggles to generalize beyond the training data, especially when encountering unseen or complex linguistic structures.

Limitations:

- The dataset used for training and evaluation was generated using a basic structure. Some of the sentences may contain patterns that are not fully coherent or representative of natural Valyrian sentences.
- The lack of diversity and natural fluency in the generated data limits the model's ability to generalize effectively.

Contributions : *Rizlaine & Ashley : Data preparation, Mahnoor & Hoogwang : Model Construction, Training and Evaluation : Mahnoor & Ashley*