CS 4372 Assignment 4

NLP Using Transformers

November 2025

Group Members:
Madison Humphries (mrh210006)

Late Days Used: 1

1 Introduction

The goal of this project is to perform machine translation on the classic English text "A Christmas Carol" by Charles Dickens, by translating it into Spanish. We are translating the book through a transformer-based sequence-to-sequence model and evaluating performance with BLEU and ROUGE metrics. In this project, "A Christmas Carol" by Charles Dickens was obtained from Project Gutenberg, while the reference used in the evaluation is a Spanish publication of "A Christmas Carol" provided by "Elejandria libros de dominio publico" (Alexandria Public Domain Books).

This task demonstrates how transformer models can be applied to natural language processing problems beyond standard datasets. Machine translation is a core NLP task that involves generating semantically accurate translations while preserving grammatical structure and meaning.

2 Transformers in Translation

Machine translation is the task of converting text from one language to another while preserving the meaning, grammar, and style. Traditional machine translation methods relied on phrase-based statistical models, but these struggled to capture long-range dependencies and context. Transformers, introduced by Vaswani et al. (2017), revolutionized machine translation by using self-attention mechanisms to model relationships between all words in a sentence simultaneously, rather than sequentially.

Key Components in Transformers

1. Encoder-Decoder Architecture:
   (a) Encoder: processes the source sentence (input sentence) and generates a sequence of contextualized embeddings representing each word in context.
   (b) Decoder: uses the embeddings to generate the target sentence (translated output) one token at a time, attending to the source sentence via cross-attention
2. Self-Attention Mechanism
   (a) Allows the model to weigh the importance of each word in the sentence relative to others.
   (b) Crucial for handling long sentences, dependencies, and word reordering between languages
3. Positional Encoding: provides information about the order of words, essential for generating coherent translations
4. Beam Search in Decoding:
   (a) Keeps multiple candidate sequences and selects the most probable translation
   (b) Hyperparameters like num_beams and length_penalty allow control over translation quality and fluency

Why Transformers?

1. Captures long-range dependencies better than RNNs or LSTMs
2. Computations performed in parallel enable faster training and inference
3. Scales effectively for large multilingual datasets, allowing pre-trained models to generalize across many languages
4. Produces fluent, contextually accurate translations for complex texts

Transformers have overall influenced the scope of machine learning, dominating in machine translation due to their capabilities. Furthermore, with the addition of transfer learning, transformers can be replicated and further trained or tuned without requiring all of the effort and time to create transformers from scratch.

3 Transformer Architecture

We are using a transformer model from HuggingFace called **Helsinki-NLP/opus-mt-en-es**. This model implements the standard encoder-decoder transformer architecture introduced in Vaswani et al., 2017.

1. Encoder
   (a) The encoder will process the input of "A Christmas Carol" by Charles Dickens and will transform it into a sequence of contextual embedding
   (b) Each encoder layer contains:
      i. Multi-head self-attention: captures dependencies between all tokens in the input simultaneously, allowing the model to understand context beyond local word sequences
      ii. Feed-forward neural network (FFN): applies nonlinear transformations to each token representation
      iii. Residual connections and layer normalization: stabilize training and improve gradient flow
2. Decoder
   (a) The decoder generates the output Spanish text autoregressively, predicting one token at a time
   (b) Each decoder layer contains:
      i. Masked multi-head self-attention: prevents the decoder from seeing future tokens, ensuring proper sequence generation
      ii. Encoder-decoder cross-attention: allows the decoder to attend to the encoder's output, providing contextual information from the source sentence
      iii. Feed-forward network, residual connections, and layer normalization
3. Tokenization
   (a) Uses SentencePiece tokenizer trained for English and Spanish, splitting text into subword units to handle rare words and morphological variations efficiently.
   (b) Input text is converted into token IDs before being fed into the encoder, and decoder output token IDs that are converted back to text
4. Generation and Hyperparameters
   (a) Translation is generated using beam search, which keeps multiple candidate sequences and selects the most likely output
   (b) Key generation hyperparameters:
      i. num_beams: number of beams in beam search, controlling exploration of candidate sequences
      ii. length_penalty: adjusts preference for shorter or longer sentences
      iii. max_length: maximum number of tokens in a generated sequence to prevent truncation
5. Pre-Training
   (a) The model is pre-trained on multilingual parallel corpora and fine-tuned for English-to-Spanish translation
   (b) Pre-training allows the model to generalize well to literary text, despite differences from everyday language

4 Model Hyper-Parameter Tuning

To optimize translation quality, we performed a systematic hyperparameter search over key generation parameters of the transformer model:

Hyperparameter	Values Tested	Description
num_beams	3, 5, 7	Beam search width. Higher values explore more candidate sequences, potentially improving translation fluency and accuracy.
length_penalty	0.8, 1.0, 1.2	Adjusts preference for sentence length. Values <1 favor shorter translations, while values >1 favor longer translations.

Table 1: Hyperparameter settings used for tuning the transformer translation model.

Testing Notes

1. Each hyperparameter combination was tested on the first 750 sentences of the English version, to match the available Spanish reference and minimize computational cost.
2. The translation was also performed chunk-by-chunk to respect the model's token limit (About 512 tokens per chunk), and translated chunks were combined before evaluation.
3. A progress bar was implemented to monitor translation progress in Google Colab for easy visual status.

Evaluation Methodology

For evaluating Machine Translation, both BLEU (Bilingual Evaluation Understudy) Score and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score can be utilized, as well as other methods. We only used BLEU and ROUGE for evaluation for each hyperparameter combination.

Evaluation Metric	Description	Score
BLEU (Bilingual Evaluation Understudy) Score	Measures n-gram overlap between the translated output and the reference translation.	Higher BLEU indicates a more accurate translation.
ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score	Measures overlap of unigrams (ROUGE-1), bigrams (ROUGE-2), and longest common subsequence (ROUGE-L) between the translation and a provided reference.	Higher ROUGE scores indicate better coverage of content and fluency.

Table 2: Evaluation metrics used to assess translation quality.

Expected Results

1. Beam search with num_beams = 5 or 7 is likely to provide higher BLEU and ROUGE scores than num_beams = 3, at the cost of longer computation time.
2. length_penalty = 1.0 is expected to generate translations of balanced sentence length, while values <1 or >1 may result in sentences that are too short or too long, slightly affecting BLEU and ROUGE scores.

5 Actual Results & Analysis

Table 3: Translation Evaluation Scores (Paragraph-Level)

num_beams	length_penalty	BLEU	ROUGE-1	ROUGE-2	ROUGE-L
5	1.0	0.092672	0.572632	0.204072	0.323743
7	1.0	0.092116	0.572630	0.204206	0.320878
5	1.2	0.091352	0.571161	0.203414	0.322973
7	1.2	0.090702	0.573028	0.203081	0.320579
7	0.8	0.090655	0.571963	0.203699	0.320150
5	0.8	0.089356	0.571429	0.202300	0.321839
3	1.0	0.086751	0.572831	0.201637	0.319383
3	0.8	0.086389	0.572766	0.200843	0.318671
3	1.2	0.085380	0.572564	0.201076	0.319233

Methodology
We used paragraph-level separation and evaluation without strict sentence-by-sentence alignment. In this approach, each paragraph is treated as a single evaluation unit rather than evaluating each sentence individually. The English and Spanish texts are split into corresponding paragraphs, and the translation is compared at the paragraph level.

This approach is motivated by the characteristics of literary texts, which often contain long and complex sentences that can be split, merged, or restructured in translation. Strict sentence-by-sentence alignment can misrepresent translation quality because a translator might naturally adjust sentence boundaries to preserve flow, meaning, or style in the target language. Evaluating whole paragraphs avoids penalizing these natural variations.

BLEU Score
The highest BLEU score achieved was 0.092672, which is low given that BLEU scores range from 0 to 1. BLEU calculates translation quality based on n-gram overlap between the translation and the reference. When sentences are split or combined differently in translation, many n-grams are seen as mismatches, even though the meaning is preserved.

For example, the English sentence *"He was tired. He went home."* could be translated directly as *"Él estaba cansado. Fue a casa."*, but a more natural translation might be *"Cansado, se fue a casa."* BLEU would penalize this restructuring despite the meaning being the same.

ROUGE Score
The highest ROUGE scores were ROUGE-1 = 0.573028, ROUGE-2 = 0.204206, and ROUGE-L = 0.323743. ROUGE measures overlap of words or phrases between candidate and reference translations, without requiring exact ordering. This makes it more tolerant of structural changes, such as sentence splitting or merging.

While ROUGE scores are higher than BLEU, they are still relatively low. This can be attributed to differences in word choice between a machine-generated translation and a published literary translation, as multiple phrases or words can convey the same meaning.

Best Hyper-parameters
The best hyperparameters were found to be num_beams = 5 and length_penalty = 1.0, achieving the highest

BLEU score as well as the highest ROUGE-1 and ROUGE-L scores among all tested configurations. While the absolute values of these scores remain low due to the nature of literary text translation, these settings represent the optimal trade-off between beam search breadth and sentence length bias for this model. Furthermore, these hyperparameters align with theoretical expectations, as moderate beam widths often balance translation diversity with accuracy, and a length penalty of 1.0 maintains natural sentence length without favoring overly long or truncated outputs.

Conclusion
BLEU underestimates translation quality for literary texts, whereas ROUGE provides a somewhat more accurate estimation. Paragraph-level evaluation better captures meaning preservation without penalizing natural restructuring in the target language.

In literary works like "A Christmas Carol", sentence complexity, idiomatic expressions, and stylistic flourishes make literal word-for-word matching less meaningful. Unique words or phrases may not have direct equivalents in the target language. Comparing a machine-translated version to a high-quality published translation naturally leads to lower BLEU and ROUGE scores, reflecting stylistic differences rather than translation quality.

Ultimately, translation quality should be assessed based on whether the meaning, tone, and context of the original text are effectively conveyed. Low BLEU and ROUGE scores in this case do not imply poor translation, but rather highlight the limitations of automatic metrics when applied to literary texts.