English to Greek Neural Machine Translation Model Report

William Hayden and Aris Antoniadis

Data Source and Research Question

Our project aimed to build a neural machine translation (NMT) model for translating English to Greek. The motivation stemmed from a desire to tackle the academic challenge of implementing state-of-the-art machine translation techniques and a personal interest in the Greek language, given its unique alphabet and its personal significance as the native tongue of Aris's relatives. Language translation has long been a cornerstone application of machine learning, making it an ideal domain for exploring cutting-edge neural architectures.

To train our model, we utilized the Europarl Parallel Corpus, a well-regarded dataset spanning 1996-2011, containing parliamentary proceedings translated into multiple languages. This dataset provided 1,235,977 English-Greek sentence pairs, offering contextually consistent and high-quality data ideal for machine translation tasks. By leveraging this dataset, we aimed to construct a functional model capable of translating general text effectively. Recognizing that achieving state-of-the-art results was beyond the scope of our computational resources, we focused instead on demonstrating the principles of neural machine translation and developing a foundational understanding of the challenges involved in translating between two linguistically distinct languages.

Descriptive Data Analysis

The Europarl Parallel Corpus provided 1,235,977 sentence pairs, which we split into training (80%), validation (10%), and test (10%) sets. After preprocessing, the training set contained 366,815 sentences, while the validation and test sets each contained 45,852 sentences. Preprocessing involved converting text to lowercase, removing excess whitespace, and filtering sentences based on length (3-20 tokens). To address missing data, 2,354 entries from the English column and 5,767 from the Greek column were excluded. This ensured a clean and consistent dataset.

Tokenization was performed using the SentencePiece algorithm with Byte Pair Encoding (BPE), creating a shared vocabulary of 32,000 tokens. This approach effectively handled rare

and unseen words, enabling the model to generalize across diverse sentence structures. While preprocessing standardized the data and reduced noise, the dataset's domain-specific nature, focusing on formal parliamentary language, limited its ability to capture the nuances of contemporary Greek usage.

Model Architecture and Implementation

Our NMT model was based on the Transformer architecture introduced in the seminal paper *Attention is All You Need* by Vaswani et al. (2017). The Transformer architecture revolutionized the field of machine translation, achieving state-of-the-art results on benchmarks like WMT 2014 English-to-German with a BLEU score of 28.4. This result was obtained through training over 3.5 days on 8 GPUs, demonstrating the scalability and efficiency of the Transformer model compared to earlier approaches. As the paper notes, "the Transformer attains state-of-the-art quality while being more parallelizable and requiring significantly less time to train" (Vaswani et al., 2017).

Our implementation followed the TensorFlow documentation, adapting the Transformer architecture to translate English to Greek. We chose TensorFlow over scikit-learn because TensorFlow is optimized for deep learning, offering built-in layers for natural language processing (NLP), seamless GPU support (critical for training on the more powerful Colab A100 GPU), and ready-to-use components like attention mechanisms. In contrast, scikit-learn is better suited for traditional machine learning tasks and lacks the native support for large-scale neural networks and GPU acceleration required for our project. TensorFlow provides extensive documentation on how to implement its Transformers-related functions, which helped us to quickly begin developing our model without significant experience with the library prior to this project.

Following the architecture described in the original Transformers paper, the model consisted of 6 encoder and decoder layers, an embedding dimension of 256, 8 attention heads, and a feed-forward network dimension of 1,024, with a dropout rate of 0.1. The encoder used self-attention mechanisms to process the input sentence, while the decoder incorporated both self-attention and encoder-decoder attention to generate translations. As described in the original paper, "Each sublayer (attention or feed-forward) in both the encoder and decoder is followed by

a residual connection and layer normalization," (Vaswani et al., 2017) ensuring stability and robust learning.

Our inspiration also drew from the methodology established by AlexNet, which popularized deep learning by demonstrating the value of large-scale neural networks. While AlexNet focuses on computer vision, its emphasis on GPU acceleration and advanced architectures influenced our approach to NLP tasks.

Training and Evaluation

The model was trained over 20 epochs using the Adam optimizer with a custom learning rate schedule, incorporating a warm-up phase as recommended in the original Transformer paper: "For the base models, we use a warmup of 4000 steps" (Vaswani et al., 2017). Training utilized Sparse Categorical Cross Entropy as the loss function, and the learning rate schedule enabled efficient convergence while avoiding instability.

Performance was evaluated using BLEU, CHRF, and Translation Edit Rate (TER) metrics. Our model achieved a BLEU score of 0.23 (23.0), a CHRF score of 26.64, and a TER of 168.98. While these metrics indicate modest performance, they underscore the challenges of translating between English and Greek, given the syntactic and morphological differences between the languages. The relatively low token-level accuracy reflected the difficulty of training with limited computational resources and domain-specific data. However, considering that our model was trained on a single Colab A100 GPU in under a day, we achieved a BLEU score that approaches the 28.4 reported by the *Attention is All You Need* Transformer architecture a decade ago, accomplished with significantly more computational resources and training time.

BLEU (Bilingual Evaluation Understudy) is a widely used metric in the field of Machine Translation. BLEU is meant to automatically assess the quality of machine-generated translations by comparing them to human-generated reference translations. BLEU measures the similarity of n-grams between the machine-generated results and human-written examples and also multiplies it with a "brevity penalty" that encourages the system to produce translations that are as long as the reference text. We chose to utilize BLEU score as our main evaluation metric because it is

probably the most relevant industry standard when it comes to NMT evaluation, as well as being the main metric provided to evaluate the results of the model in *Attention is All You Need*.

Key Features of the Transformer Architecture

The Transformer's success lies in its ability to model complex relationships between words using attention mechanisms. In the encoder, self-attention computes the relevance of each word to every other word in the input sentence. For example, in the sentence "The cat sat on the mat," self-attention may emphasize the relationship between "cat" and "sat," enabling the model to understand context and grammatical structure. This process is enhanced by residual connections and dropout, as noted by Vaswani et al.: "We employ a dropout rate of P=0.1."

The decoder extends this mechanism by combining self-attention with encoder-decoder attention, focusing on both the target sentence generated so far and the encoded representation of the source sentence. As the paper explains, "The decoder inserts a third sublayer, which performs multi-head attention over the output of the encoder stack" (Vaswani et al., 2017). This ensures that the translation aligns with the input sentence's meaning.

Improvements and Future Directions

Several enhancements could significantly improve our model's performance. First, expanding the dataset to include contemporary and colloquial text would provide a broader representation of English-Greek translation contexts. Online resources could augment the corpus, addressing gaps in domain coverage. Improved preprocessing, such as advanced text normalization and techniques for handling out-of-vocabulary words, would further refine input data quality.

On the model side, scaling the architecture by increasing the depth and width of layers, experimenting with advanced attention mechanisms, and incorporating additional residual connections could enhance learning capacity. Systematic hyperparameter optimization using grid search or Bayesian methods would likely improve performance. Training on distributed systems with multiple GPUs would enable larger models and longer training durations, overcoming computational constraints.

Exploring transfer learning through pre-trained models like mBERT or multilingual translation models could provide a significant boost, particularly for low-resource language pairs like English-Greek. Ensemble methods, combining predictions from multiple models, could improve translation quality through techniques like voting or averaging.

Finally, domain adaptation strategies, such as fine-tuning the model for specialized domains like legal or medical text, could create targeted translation systems. Continuous learning mechanisms would ensure that the model remains relevant as language evolves, addressing one of the major challenges in machine translation.

Conclusion

Our project demonstrated the principles of neural machine translation, providing valuable insights into the complexities of translating between linguistically distinct languages like English and Greek. Despite modest performance, the Transformer architecture proved effective, showcasing its potential for scaling and adaptation. With enhanced resources and methodologies, our model could evolve into a robust system capable of tackling diverse translation tasks and being able to assist in communication with relatives in Greece!

Appendix: Code

We used Google Colab as our primary development tool for this project. This is because it had easy and consistent access to cloud instances of a Nvidia A100 GPU. We used this to facilitate the deep learning techniques that the Transformers architecture facilitates. The TensorFlow library is also designed to work with GPUs or TPUs to reach optimum performance in model training. The original final is also shared with you on Google Drive (**Translation Engine 2.0 - Transformers**). The file contains a path to the data as well as the model weights that we acquired through our training, you should be able to run the evaluation cells on the pre-trained model to achieve the same evaluation results that we referred to in the above Training and Evaluation section of our report. If you'd like to connect your own GPU instance to train the model yourself, please feel free to do so, and if you achieve better results with your hyperparameter selection/training configuration, we would love to know!