Proposed Method: Transformer Character-Level Encoders with Cross-Attention and Improved Morpheme Segmentation*

George Kaceli
University of Windsor
Windsor, ON, Canada
kaceli@uwindsor.ca

Vimanga Umange
University of Windsor
Windsor, ON, Canada
umange@uwindsor.ca

ABSTRACT

We propose an enhanced pipeline for automated interlinear glossing. Our approach uses a Transformer-based character encoder, and introduces a full Transformer encoder–decoder architecture with cross-attention. Our pipeline integrates an unsupervised morpheme segmentation module that leverages adaptive thresholding and structured prediction via a forward–backward algorithm with utility masking to handle variable-length inputs. In addition, a translation encoder supplements the segmented representations, and a Transformer decoder generates gloss sequences that accurately reflect the morphological structure. The resulting model is optimized to generate precise gloss sequences (e.g., `shortage-FEM.NOM.SG wine-NEUT.GEN.SG`) from source words (e.g., `inopi-a`) and the translation (e.g., `a wine shortage`).

KEYWORDS

Transformer, Interlinear Glossing, Morpheme Segmentation, Relative Positional Encoding, Sequence-to-Sequence

1 PROBLEM DEFINITION

Given a source word represented as a sequence of characters

$$W = \{c_1, c_2, \ldots, c_n\},$$

our goal is to automatically generate a gloss sequence

$$G = \{g_1, g_2, \ldots, g_K\},$$

that accurately reflects the morphological structure and semantic content of W.

Each character c_i is converted into an embedding via

$$E : C \to \mathbb{R}^d,$$

and processed by a Transformer-based encoder to produce contextual representations:

$$H = \{h_1, h_2, \ldots, h_n\}.$$

An unsupervised segmentation module then partitions W into morphemes

$$S = \{M_1, M_2, \ldots, M_k\},$$

by computing segmentation probabilities for each h_i and applying an adaptive threshold. Aggregated morpheme representations, with an auxiliary translation T, serve as memory for a decoder that generates the gloss G [5].

Thus, the problem is to learn a mapping

$$f : (W, T) \to G,$$

which produces a gloss sequence that captures both the morphological and semantic properties of the source word.

*https://github.com/gfkaceli/COMP8730_Project

2 PROPOSED APPROACH

Our method integrates advanced sequence modeling with adaptive segmentation and joint optimization to generate high-quality glosses from source words.

2.1 Source Encoding and Contextual Representation

The process begins with a Transformer-based encoder that converts source words, represented as sequences of one-hot encoded characters, into rich contextual embeddings. Discrete inputs are projected into a continuous embedding space. Multi-head self-attention, augmented with relative positional encodings [3, 4], captures both local morphological patterns and long-range dependencies.

Definition 2.1. **Relative Positional Encoding:** A method to represent the positions of tokens relative to each other, allowing the model to better capture sequence order information without relying solely on absolute positions.

The resulting high-dimensional, context-aware vectors, denoted by h_i for the ith character, form the foundation for subsequent segmentation and gloss generation.

2.2 Adaptive Morpheme Segmentation and Aggregation

After encoding, the segmentation module partitions the continuous representations into morphemes. For each encoded vector h_i, a segmentation probability is computed as

$$s_i = \sigma(W h_i + b),$$

where σ denotes the sigmoid function, W is a weight matrix, and b is a bias term.

Definition 2.2. **Segmentation Probability s_i:** The likelihood that a boundary exists after the ith character.

To determine where to segment, summary statistics of the encoder outputs are aggregated:

$$\mathbf{z} = \left[\max_i h_i;\ \text{mean}_i(h_i);\ \text{var}_i(h_i) \right].$$

This aggregated vector is then passed through an MLP to predict a dynamic threshold τ:

$$\tau = \sigma(\text{MLP}(\mathbf{z})).$$

A forward-backward algorithm in log space computes marginal probabilities for segmentation boundaries, ensuring only valid positions contribute[1]. The resulting segmentation mask is used to aggregate contiguous character embeddings into morpheme-level representations.

2.3 Gloss Generation via Cross-Attention Decoder

The aggregated morpheme representations are combined with an auxiliary translation representation—obtained by averaging translation embeddings—to form a comprehensive memory [5].

Definition 2.3. **Memory:** In this context, the memory is the combined set of morpheme-level embeddings and the translation representation used by the decoder.

A Transformer-based decoder then generates gloss tokens autoregressively. It employs cross-attention to condition each output token on the full memory, ensuring that both morphological structure and semantic context are utilized in the prediction.

2.4 End-to-End Training and Gradient Estimation

The entire pipeline is trained end-to-end using a joint loss function that integrates:

Gloss Prediction Loss: A cross-entropy loss over the gloss tokens generated by the decoder.

Segmentation Loss: A binary cross-entropy loss on the segmentation probabilities s_i, encouraging accurate boundary detection.

Morpheme Count Loss: An auxiliary loss that minimizes the discrepancy between the predicted and target morpheme counts.

A key challenge in training is the non-differentiability of hard segmentation decisions. To overcome this, we employ a straight-through estimator with a Gumbel-Softmax relaxation [2]:

$$\tilde{s}_i = \text{GumbelSoftmax}(s_i, \tau_{\text{temp}}),$$

where τ_{temp} is a temperature parameter that controls the smoothness of the relaxation.

Definition 2.4. **Gumbel-Softmax Relaxation:** A differentiable approximation to sampling from a categorical distribution, enabling gradient flow through discrete decisions.

This relaxation, along with additional variance reduction techniques and regularization terms, ensures smooth gradient propagation and consistent decoding, ultimately leading to improved performance on the gloss generation task.

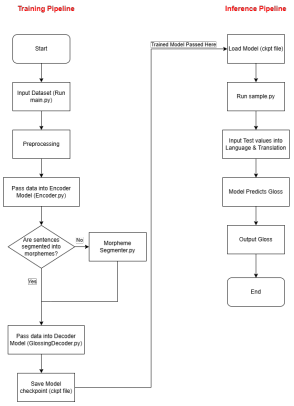

Figure 1: Overview of the proposed glossing pipeline.

3 ALGORITHM

Algorithm 1 provides the high-level pseudocode for our method.

Algorithm 1 Transformer-Based Glossing Pipeline

procedure GlossingPipeline(W, T, θ)
  Input: Source word $W = \{c_1, \ldots, c_n\}$, Translation T, Parameters θ
  Encoder:
  for $i = 1$ to n **do**
    $e_i \leftarrow E(c_i)$
    $p_i \leftarrow \text{RelPosEnc}(i)$
    $\tilde{e}_i \leftarrow e_i + p_i$
  $H \leftarrow \text{TransformerEncoder}(\{\tilde{e}_1, \ldots, \tilde{e}_n\})$

  Unsupervised Segmentation (for Track 1):
  for $i = 1$ to n **do**
    $s_i \leftarrow \sigma(W_{\text{seg}}\, h_i + b_{\text{seg}})$
  Compute rich summary z from H using max, mean, and variance.
  Predict adaptive threshold: $\tau \leftarrow g(z)$
  Compute structured marginals m via a forward–backward algorithm over s_i, using utility masks.
  Generate binary segmentation mask: $b_i = \mathbf{1}\{m_i > \tau\}$
  Aggregate contiguous character representations using b_i:

$$S \leftarrow \text{Aggregate}(H,\, b_1, \ldots, b_n)$$

  Translation Encoding:
  $T_{\text{repr}} \leftarrow \text{AvgPool}(E_{\text{trans}}(T))$

  Decoder (Glossing):
  Form memory: $M \leftarrow \text{Concat}(T_{\text{repr}}, S)$
  for each decoding step t **do**
    Compute query q_t from previously generated gloss tokens
    $c_t \leftarrow \text{CrossAttention}(q_t, M)$
    Generate gloss token $g_t \leftarrow \text{MLP}(c_t)$
  Output: Gloss sequence $G = \{g_1, \ldots, g_K\}$

REFERENCES

[1] Leander Girrbach. 2023. Tü-CL at SIGMORPHON 2023: Straight-Through Gradient Estimation for Hard Attention. In *SIGMORPHON*.

[2] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

[3] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 464–468.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[5] Changbing Yang et al. 2024. Embedded Translations for Low-resource Automated Glossing. In *SIGMORPHON*.