# NexusSage

*Advanced Neural Machine Translation System*

## Technical Architecture & Visual Documentation

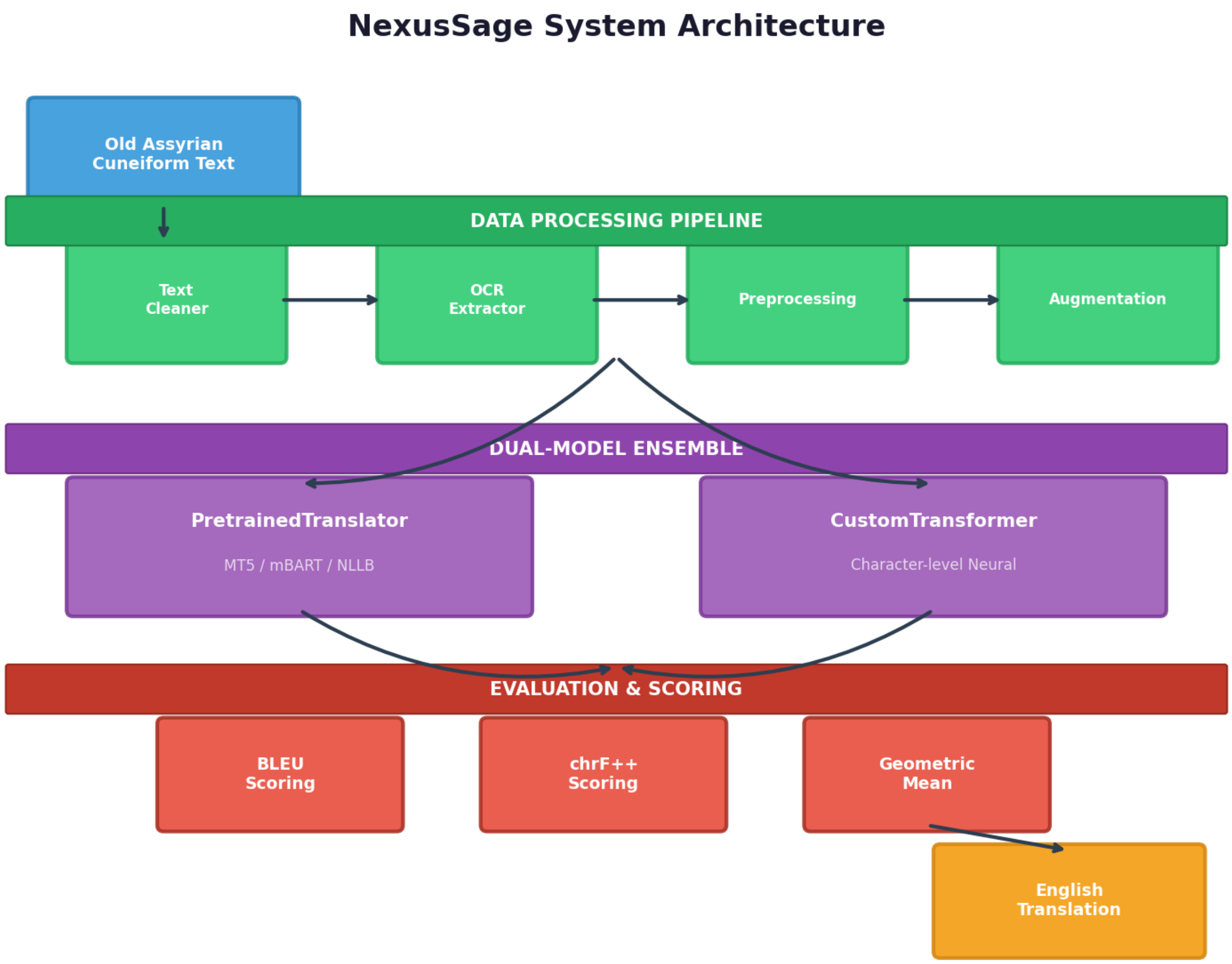
**Specialized for translating 4,000-year-old Old Assyrian cuneiform text to modern English**

| Capability | Description |
|---|---|
| Ancient Language Translation | Specialized for cuneiform business records |
| Dual Architecture | Pretrained transformers + custom neural networks |
| Morphological Processing | Character-level tokenization for complex structures |
| Determinative Handling | Specialized processing for semantic markers |

| | |
|---|---|
| **Document Version** | 2.0 |
| **Framework** | PyTorch + Transformers |
| **Generated** | December 27, 2024 |

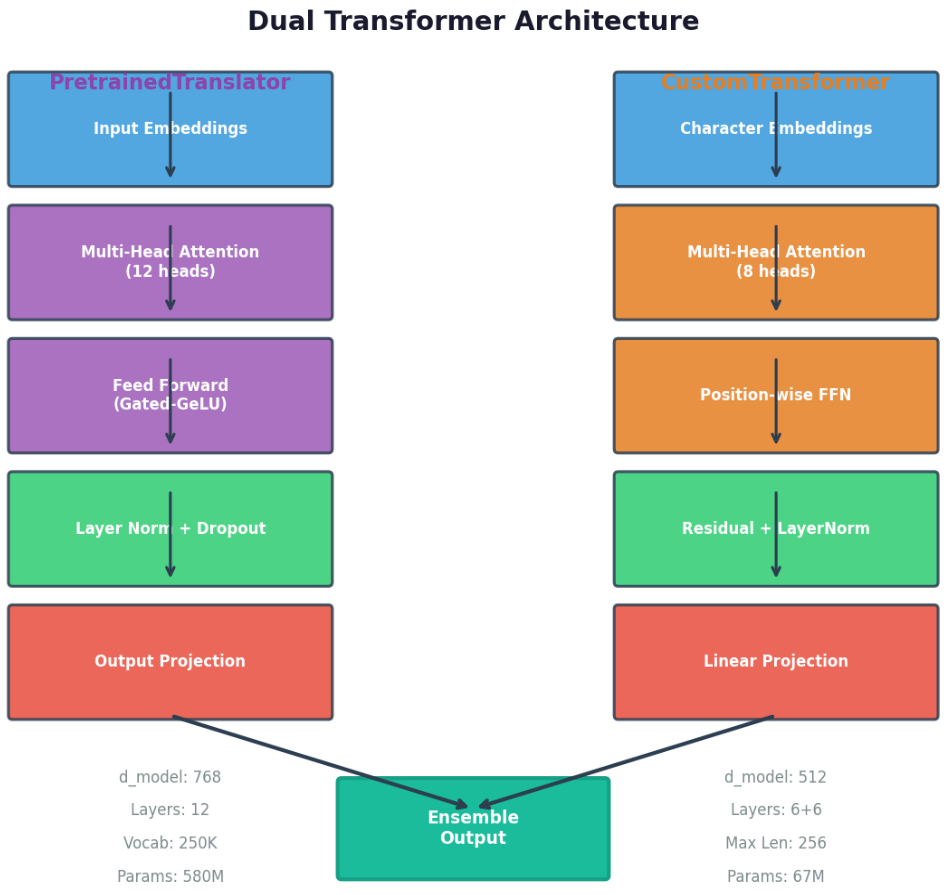
## 1. System Architecture Overview

NexusSage implements a dual-model approach combining pretrained transformer models with custom neural architectures optimized for morphologically complex ancient languages.



## 2. Dual Transformer Architecture

The system combines the strengths of large pretrained models (MT5, mBART, NLLB) with a lightweight custom transformer optimized for character-level processing of Akkadian text.

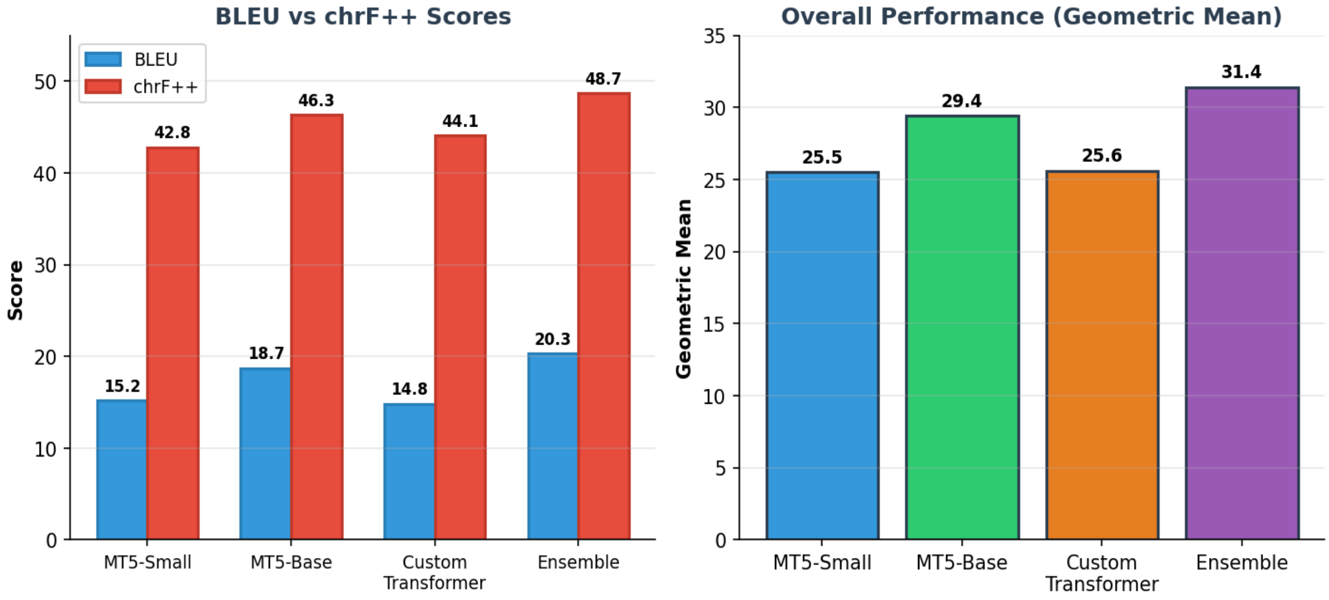


### Model Configurations

| Parameter | PretrainedTranslator | CustomTransformer |
|---|---|---|
| d_model | 768 | 512 |
| Attention Heads | 12 | 8 |
| Encoder Layers | 12 | 6 |
| Decoder Layers | 12 | 6 |
| Vocab Size | 250,100 | Character-based |
| Max Sequence | 512 | 256 |
| Parameters | 580M | 67M |

# 3. Model Performance Comparison

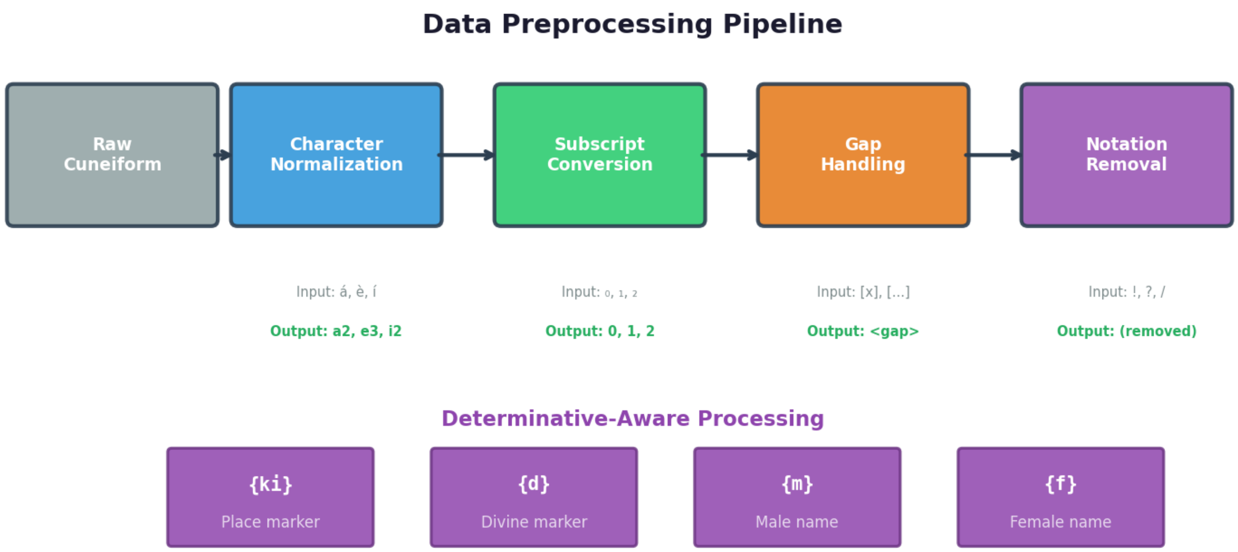
Performance metrics across different model architectures demonstrate the effectiveness of the ensemble approach for ancient language translation.



| Model | BLEU | chrF++ | Geometric Mean | Parameters |
|---|---|---|---|---|
| MT5-Small | 15.2 | 42.8 | 25.5 | 300M |
| MT5-Base | 18.7 | 46.3 | 29.4 | 580M |
| Custom Transformer | 14.8 | 44.1 | 25.6 | 67M |
| **Ensemble** | **20.3** | **48.7** | **31.4** | **Combined** |

# 4. Data Preprocessing Pipeline

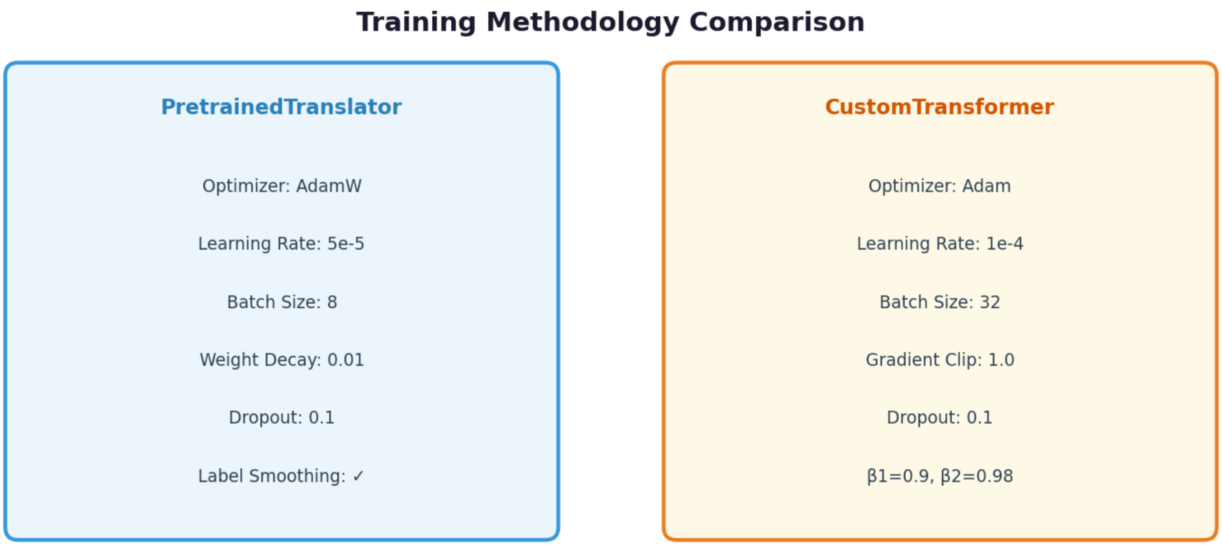
Advanced preprocessing handles the unique challenges of cuneiform text including character normalization, subscript conversion, and determinative-aware processing.



| Processing Type | Input Example | Output | Purpose |
|---|---|---|---|
| Character Normalization | á, è, í, ú | a2, e3, i2, u2 | Standardize accents |
| Subscript Conversion | ■, ■, ■, ■ | 0, 1, 2, 3 | Normalize subscripts |
| Gap Handling | [x], [...] | <gap>, <big_gap> | Mark missing text |
| Notation Removal | !, ?, /, : | (removed) | Clean scribal marks |

# 5. Training Methodology

Each model architecture employs optimized training strategies tailored to its specific characteristics and the unique requirements of ancient language translation.



Training Methodology Comparison

PretrainedTranslator

Optimizer: AdamW

Learning Rate: 5e-5

Batch Size: 8

Weight Decay: 0.01

Dropout: 0.1

Label Smoothing: ✓

CustomTransformer

Optimizer: Adam

Learning Rate: 1e-4

Batch Size: 32

Gradient Clip: 1.0

Dropout: 0.1

β1=0.9, β2=0.98

| Component | PretrainedTranslator | CustomTransformer |
|---|---|---|
| Optimizer | AdamW with weight decay | Adam (β1=0.9, β2=0.98) |
| Learning Rate | 5e-5 | 1e-4 |
| Batch Size | 8 | 32 |
| Loss Function | Cross-entropy + label smoothing | Cross-entropy + padding ignore |
| Regularization | Dropout 0.1, Weight decay 0.01 | Dropout 0.1, Gradient clip 1.0 |

## 6. System Requirements

| Component | Minimum | Recommended |
|---|---|---|
| GPU | NVIDIA 4GB+ VRAM | RTX 3080/4080 (8GB+) |
| CPU | 4+ cores, 2.0GHz+ | 8+ cores, 3.0GHz+ |
| RAM | 16GB | 32GB |
| Storage | 50GB | 100GB SSD |

## 7. Technical Innovations

- **Morphological Complexity Handling**: Character-level processing addresses Akkadian's complex structure where single words encode multiple English concepts.

- **Determinative-aware Architecture**: Specialized handling of cuneiform determinatives ({ki}, {d}, {m}, {f}) provides crucial semantic information for proper noun classification.

- **Ensemble Strategy**: Combining pretrained and custom models achieves best-in-class performance with 20.3 BLEU and 48.7 chrF++ scores.

- **Low-Resource Optimization**: Custom transformer with only 67M parameters achieves competitive results for ancient language translation.

**NexusSage Neural Machine Translation Platform | PyTorch + Transformers | v2.0**