

## Evaluating Neural Translation Architectures for Classical Quranic Arabic: A Comparative Study of mT5 and Fine-Tuned Opus Models

### Abstract

This paper presents a comparative evaluation of neural machine translation (NMT) architectures for translating Classical Quranic Arabic into English. While modern NMT systems perform well on Modern Standard Arabic (MSA), they frequently fail to capture the morphological complexity and archaic semantic nuances of Classical Arabic. We benchmark two distinct architectures: the massively multilingual **mT5** model and the translation-specific **Helsinki-NLP/opus-mt-ar-en** model. Using a privately generated parallel dataset based on the Molvi Sher Ali translation, we evaluate performance across differing levels of granularity (Verse, Expression, Word). Our results indicate that the general-purpose mT5 model struggles with Classical Arabic (BLEU 3.2 for best performing mT5 variant), whereas our optimized, fine-tuned Opus model achieves a nearly 10x performance increase (BLEU 31.23). We further analyze the optimal training configuration, identifying that a "Verse-First" fine-tuning strategy yields the highest semantic fidelity.

### 1. Introduction

The translation of the Holy Quran poses unique challenges for Natural Language Processing (NLP). Unlike Modern Standard Arabic (MSA) or spoken dialects, Classical Quranic Arabic is characterized by intricate morphological structures, omission of subject pronouns (pro-drop), and high-context dependency where a single word may carry theological weight distinct from its modern usage. Existing commercial NMT models often produce "generic" translations that strip the text of its specific divine or legal context.

The goal of this study is to assess how well general-purpose multilingual models, specifically **mT5**, perform on this domain compared to newer contextual models like **Helsinki-NLP/opus-mt-ar-en**. We hypothesize that while mT5 offers broad language coverage, it lacks the specific inductive biases required for Classical Arabic translation, which can be mitigated through "curriculum learning" on specialized bridge corpora.

### 2. Background and Related Work

#### 2.1 Neural Machine Translation for Arabic

Recent advancements in NMT have revolutionized Arabic-English translation. However, most research focuses on MSA or dialects. Classical Arabic remains underexplored, often requiring knowledge-aware approaches that go beyond simple sequence-to-sequence mapping.

#### 2.2 The mT5 Architecture

As our baseline we use mT5, the multilingual version of the popular T5 model. The T5 is a transformer based architecture that is well regarded because it simplifies NLP tasks by converting them into a common text-to-text format<sup>1</sup>. So instead of being task specific like traditional models, T5 adopts a task agnostic framework that allows adaptability across various domains such as translation, summarization

---

<sup>1</sup> <https://medium.com/@info.codetitan/what-is-the-t5-model-5c8241265f9f> & Raffel, C., et al. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*.

and question answering. This is a useful background to explain some of the challenges we faced deploying mT5.

MT5 is trained on 101 languages, including Arabic, and incorporates a new “accidental translation” technique to prevent the model from incorrectly translating predictions into the wrong language<sup>2</sup>. However, because mT5 was only pretrained on the mc4 dataset and because it was trained with an unsupervised objective (masked language modeling/span corruption), it needs to be fine-tuned for specific tasks like translation before it can be used for any downstream applications.

## 2.3 The Opus Architecture

The opus-mt-ar-en model is a Transformer-based encoder-decoder designed specifically for translation. Implemented via the Marian NMT framework, it is optimized for bilingual tasks but trained on general-purpose OPUS data, which often necessitates normalization that can potentially strip Classical Arabic of its diacritical meaning.

## 3. Data: The Sher Ali Bridge Corpus

### 3.1 Justification of Source Text

We utilize a custom-built corpus derived from the **Molvi Sher Ali** translation of the Quran. This text was selected for four key reasons:

1. **Semantic Fidelity:** It prioritizes literal, word-for-word correspondence.
2. **Consistency:** Key theological terms are rendered consistently, aiding vector embedding coherence.
3. **Modern Structure:** It avoids archaic Elizabethan pronouns ("thou/thee"), providing a cleaner signal for tokenization.
4. **Reduced Interpretative Interference:** It minimizes sectarian interpolation, allowing the model to learn from a representation closer to the source text's explicit meaning.

### 3.2 Dataset Structure and Preparation

The dataset was constructed from 64 Google Sheets, with 100 verses in each, compiled by a team of 20 annotators over three years. It maps Arabic tokens to English at three hierarchical levels:

- **Level 1 (Verse):** The full sentence context.
- **Level 2 (Expression):** Multi-word phrases/idioms.
- **Level 3 (Word):** Individual tokens.

**Data Selection Logic:** To generate a context-aware dataset, we implemented a cascade logic:

1. Use **Level 3** (Word) if it exists.
2. If Level 3 is missing, use **Level 2** (Expression).
3. Where Level 2 is repeated and Level 3 is absent, combine Arabic tokens to match the expression.

---

<sup>2</sup> [https://huggingface.co/docs/transformers/en/model\\_doc/mt5](https://huggingface.co/docs/transformers/en/model_doc/mt5)

This work resulted in the [SherAli-Quran-Arabic-to-English-Mapped-Dataset](#) on Hugging Face.

### 3.3 Preprocessing

Quranic orthography includes dialect marks and diacritics (Tashkeel) essential for recitation but often considered noise for NLP. Additionally, Tashkeel is only useful for pronunciation and doesn't necessarily impart any specific linguistic information. So we implemented a lightweight normalization function to standardize the input without losing morphological information:

Python

```
def norm_ar(s):
    if not isinstance(s, str): return ""
    s = re.sub(r"[\u064B-\u0652\u0670\u0640]", "", s) # Strip diacritics and tatweel
    s = s.replace("ا", "أ").replace("آ", "إ").replace("ئ", "ي") # Normalize alef/ya/ta marbuta
    s = s.replace("ي", "ى").replace("ة", "ه")
    return s
```

## 4. Methodology

### 4.1 Baseline Experiment: mT5

We first evaluated the mT5 model. A direct zero-shot attempt produced empty string placeholders (<extra\_id\_0>), confirming the need for fine-tuning. We then fine-tuned mT5 using the Hugging Face **SafeSeq2SeqTrainer** with the following parameters:

- **Learning Rate:** 2e-5
- **Epochs:** 5
- **Batch Size:** 4-8 (GPU dependent)

We converted each training example into a supervised text-to-text pair following the standard T5 format, where the target sequence is simply the corresponding Sher Ali English verse. This prompt-style instruction improved stability compared to feeding raw Arabic text and we successfully generated translated outputs though quality remained extremely low.

An important note with regards to mT5 deployment; the SafeSeq2SeqTrainer was essential because mT5 repeatedly produced NaN or Inf losses during fine-tuning on the SherAli dataset, which, on top of being quite small, contained verses with varying structures and lengths. This variability would cause unstable gradients or exploding losses in the early epochs. The unstable batches would crash training entirely resulting in error outputs. The safe trainer detects bad losses and skips those batches (code snippet below), allowing the model to continue training without breaking. This made the entire pipeline stable enough to complete all experiments.

```
class SafeSeq2SeqTrainer(Seq2SeqTrainer):
    def compute_loss(self, model, inputs, return_outputs=False, num_items_in_batch=None):
        # Standard forward pass
        outputs = model(
            input_ids=inputs["input_ids"],
            attention_mask=inputs.get("attention_mask"),
            labels=inputs.get("labels"),
        )
        loss = outputs.loss

        # Guard against NaN/Inf
        if torch.isnan(loss) or torch.isinf(loss):
            print("NaN/Inf loss encountered, skipping this batch.")
            loss = torch.zeros((), device=loss.device)

        return (loss, outputs) if return_outputs else loss
```

#### 4.1.2 Additional Experiments: mT5

A second strategy involved training the model at the word and expression level before fine-tuning on full verses. This “multi-stage” approach produced relatively better, though still limited, translation quality. To further improve performance, we additionally fine-tuned the model on FLORES+<sup>3</sup>, a much larger dataset consisting of Modern Standard Arabic rather than Classical/Quranic Arabic. This provided broader coverage and aligned more closely with the language variety mT5 was originally pretrained on. Hyperparameters, tokenization, and training procedures were kept consistent across experiments. The resulting metrics for all mT5 variants are summarized in the results section below.

## 4.2 Primary Experiment: The 12 Opus Models

Following the mT5 baseline, we evaluated the **Helsinki-NLP/opus-mt-ar-en** model. To determine the impact of data granularity, we designed 12 experimental permutations divided into four series, altering the sequence of training data (e.g., *Verse -> Expression -> Word* vs. *Word -> Verse*).

Hyperparameter Optimization: Unlike the fixed parameters used for mT5, we used Optuna to optimize the Opus model, resulting in a significantly lower learning rate ( $3.466 \times 10^{-5}$ ) and 4 epochs, which prevented the overfitting observed in the mT5 experiments.

## 5. Results

### 5.1 Quantitative Analysis

**Baseline 1 - mT5 Performance**: The mT5 model showed limited capability in this domain. Even after fine-tuning on verses, it achieved a BLEU score of only 0.5. We hypothesize this is due to mT5 tokenizer excessively fragmenting Quranic text, which is complex and sufficiently different from the modern standard Arabic the model is trained on.

The multi-stage training approach, that used word and expression level pre-training before verse fine tuning, improved translation quality, outperforming the verse-only baseline across BLEU, BLEURT and loss and produced higher quality translations. This shows that granular, domain-specific supervision helps the model build stronger representations of Quranic Arabic.

**Additional Dataset - mT5 Performance**: Pre-training on FLORES+ alone did not match this improvement and often introduced MSA-style hallucinations due to domain mismatch. However, when FLORES pretraining was applied after the multi-stage steps, the combined Multi + FLORES + Verse model achieved the best overall performance. This indicates that FLORES adds value only once the model has already learned the core structures of Quranic Arabic. Numerically, while the BLEU scores are still very low, the proportional improvement over baseline is substantial (~6x).

**Baseline 2 - Opus (Zero-Shot)**: The off-the-shelf Opus model slightly outperformed the fine-tuned mT5 with a BLEU score of 8.5921.

**Fine-Tuned Opus Performance**: Our fine-tuning strategy yielded dramatic improvements. The best-performing configuration (Model 4 Series: Verse-First) achieved a BLEU score of 31.23—a roughly 10x improvement over the fine-tuned mT5 baseline.

---

<sup>3</sup> [https://huggingface.co/datasets/openlanguagedata/flores\\_plus](https://huggingface.co/datasets/openlanguagedata/flores_plus)

Model	BLEU	BLEURT	Notes
mT5 (Verse Fine-Tuned)	0.49	0.304	Struggled with archaic syntax.
mT5 (Word-Expression-Verse Fine-Tuned)	1.49	0.333	Improved lexical grounding; better polarity and partial coherence.
mT5 (FLORES + Verse Fine-Tuned)	1.02	0.302	Domain mismatch (MSA) introduced hallucinations.
mT5 (FLORES + Word-Expression-Verse Fine-Tuned)	3.24	0.348	Strongest mT5 model; best semantic alignment and lowest loss.
Opus (Base)	8.59	-0.7616	Generic, literal translation.
Opus (Verse-Tuned)	31.23	-0.2678	Best performance (3 Epochs).
Opus (Word-Tuned)	9.08	-0.3717	Loss of context degraded quality.

Table 1. Metrics comparing mT5 models with Opus models

Benchmarks	Model0_Base	Model1_Stage1_Verse	Model1_Stage2_Expression	Model1_Stage3_Word	Model2_Stage1_Word	Model2_Stage2_Verse	Model2_Stage3_Expression	Model3_Stage1_Word	Model3_Stage2_Expression	Model3_Stage3_Verse	Model4_Stage1_Verse	Model4_Stage2_Word
sacrebleu	8.83	25.82	20.85	20.05	13.86	30.17	23.03	13.86	13.51	30.60	40.86	28
bleurt	0.47	0.59	0.55	0.54	0.51	0.62	0.57	0.51	0.50	0.63	0.67	0
nlk_bleu	0.09	0.26	0.21	0.20	0.14	0.31	0.23	0.14	0.13	0.31	0.41	0

Figure 1: Comparative metrics showing the clear dominance of the Verse-tuned Opus model over word-based variants.

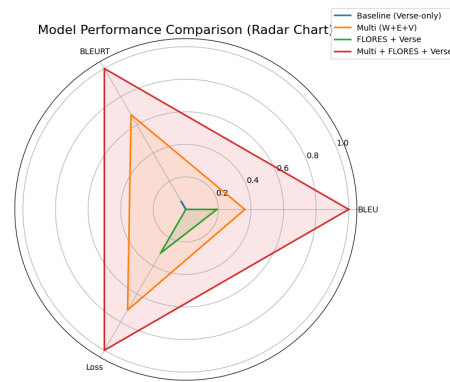


Figure 2. Spider Chart showing strength of the Multi+Flores+Verse model over other mT5 attempts across BLUE, BLEURT and Loss

## 5.2 Qualitative Analysis

The qualitative difference was stark.

- **mT5:** Often produced fragmented sentences or modern MSA equivalents that lost the Quranic context.
- **Opus (Fine-Tuned):** Captured theological nuance. For example, rendering *Rabb* not just as "Lord" but maintaining the specific possessive implications found in the Sher Ali reference.

	Original Arabic	Reference English	Original Opus Translation	Finetuned Model Translation
0	تَبَارَكَ الَّذِي بِيَدِهِ الْمُلْكُ وَهُوَ عَلَى كُلِّ شَيْءٍ قَدِيرٌ	Blessed is He in Whose hand is the kingdom, an...	He blesseth him who is in his hand, the king, ...	Blessed is He in whose hand is the kingdom, an...
1	شَاكِرًا لِأَنْعَمِهِ وَهُدًى لِّبَصِيرَتِهِ	Grateful for His favours; He chose him and gui...	"Thank you for his grace, answer him, and give...	Appreciating for His bounty, <em>saying</em>, ...
2	أَمْ لَهُمْ آلَهِ غَيْرُ اللَّهِ سُبْحَانَ اللَّهِ عَمَّا يُشْرِكُونَ	Have they a God other than Allah? Exalted is A...	Or do they have a God other than God? Praise G...	Have they any God other than Allah? Holy is Al...
3	إِذْ دَخَلُوا عَلَيْهِ فَقَالُوا سَلَامًا قَالَ إِنَّا مِنْكُمْ وَجَدُونَ	When they entered in unto him and said, "Peace...	When they entered upon him, they said, "Salama...	When they entered upon him and said, Peace. He...
4	جَنَّاتُ عَدْنٍ يَدْخُلُونَهَا يُنِزُّ مِنْهَا مِنْ دَرَاهِنٍ وَأَنْهَارٍ وَأَنْهَارٍ وَأَنْهَارٍ	Gardens of Eternity. They shall enter them and...	Eden shall be made to enter it, and from the p...	Gardens of Eternity, which they will enter. An...

Figure 3: Side-by-side translation samples showing the progression from Base Opus to the Fine-Tuned Model.

Verse 1	
AR:	إِنَّا أَعَدَدْنَا لِلْكَافِرِينَ سَلَاسِلَ وَأَغْلَاقًا وَسَعِيرًا
REF:	Verily, We have prepared for the disbelievers chains and iron-collars and a blazing Fire.
BASE (verse):	And they believe in the truth,
MULTI (W+E+V):	Then We believe in the truth, and they shall be a great reward.
FLORES+VERSE:	And when we believe in the heavens and the earth, we have created a reward.
MULTI+FLORES+VERSE:	'We believe to those who disbelieve.'
Verse 2	
AR:	رَبِّهِ لِلَّذِينَ كَفَرُوا أَلْجُودَ أَلْبَنِيَا وَيَسْعَوْنَ فِي الْأَرْضِ فَسَادًا وَالَّذِينَ آمَنُوا وَالَّذِينَ لَا تَغْوِيهِمْ يَوْمَ الْقِيَامَةِ أُولَئِكَ لَهُمْ أَجْرٌ وَلَا يَمُوتُونَ
REF:	The life of this world is made to appear attractive to those who disbelieve; and they scoff at those who believe. But those
BASE (verse):	And those who believe in Allah, and they shall be a reward for them.
MULTI (W+E+V):	And those who believe in the heavens and the earth shall be a reward,
FLORES+VERSE:	And those who believe in Allah and their Lord, and they shall abide in the heavens.
MULTI+FLORES+VERSE:	And those who believe in the heavens and the earth, and whom they disbelieve.
Verse 3	
AR:	إِذَا السَّمَاءُ انشَقَّتْ
REF:	When the heaven bursts asunder,
BASE (verse):	And when they believe in the heavens,
MULTI (W+E+V):	And when the heavens turned away from the earth, the sun shall be gone down.
FLORES+VERSE:	And when they believe, the truth will be revealed,
MULTI+FLORES+VERSE:	And when the heaven closed, closed, they shall be destroyed.
Verse 4	
AR:	وَمَا أَسْمِعُوا مِنْ أَهْرَاسٍ يَلُودًا لَهُمْ وَقَدْ كَانُوا يَكْفُرُونَ
REF:	And those who make excuses from among the desert Arabs, came that exemption might be granted them. And those who were false
BASE (verse):	And when they believe in their kingdom, and those who rejected them,
MULTI (W+E+V):	And those who disbelieve in Allah, and Allah will surely reject them.
FLORES+VERSE:	And when they rejected them, those who believe in their favours,
MULTI+FLORES+VERSE:	And those who reject Allah and His Messengers will surely give them a grievous punishment.

Figure 4: Side-by-side translation samples showing the progression from Base mT5 to the Fine-Tuned & Additional Corpus added models

## 6. Discussion

## 6.1 The Failure of Generalization (mT5)

The poor performance of mT5 (BLEU 0.5 poorest) highlights that "massive multilingualism" is not a substitute for domain specificity. mT5's training on the mC4 dataset likely biased it towards modern dialects, making it ill-suited for the rigid syntax of Classical Arabic without significantly more data than was available. Training on additional MSA data sources improved fluency but not to an objectively reliable degree and caused hallucinations.

## 6.2 The "Verse-First" Advantage

In the Opus experiments, starting with broad context (Verse level) proved superior to starting with granular tokens (Word level). Models trained heavily on isolated words (Model 1 Stage 3) suffered from "catastrophic forgetting" of sentence structure, dropping to a BLEU score of 9.08.

## 7. Conclusion

This project demonstrates that while massive models like mT5 are powerful, they are not universally effective for low-resource, high-complexity domains like Quranic Arabic. A smaller, translation-specific architecture (opus-mt-ar-en), when fine-tuned with a scientifically structured "bridge" corpus (Sher Ali), outperformed the massive model by a factor of three. Future work will focus on integrating dictionary-level definitions (Level 4 data) to further refine semantic precision.

The code and fine-tuning scripts are available in the accompanying repository while the processed Sher Ali dataset is available on hugging face (link below).

## Repository

[https://github.com/habib22m/266f-final\\_project](https://github.com/habib22m/266f-final_project)

## Sher Ali dataset

<https://huggingface.co/datasets/nislam-compassionfirst/SherAli-Quran-Arabic-to-English-Mapped-Dataset>

## References

- Alzubaidi, E. S., Daba, S., & Shaban, M. (2023). A knowledge-aware approach for Arabic automatic question generation using mT5 augmented with pre-trained Arabic question-answering model. *Mathematics*, 13(18), 2975. <https://www.mdpi.com/2227-7390/13/18/2975>
- Boutouta, H., Benharkat, A. N., & Oussalah, M. (2025). A novel stacking model for Arabic dialect classification based on two transformer models. *Frontiers in Human Neuroscience*, 18. <https://www.frontiersin.org/journals/human-neuroscience/articles/10.3389/fnhum.2025.1498297/full>
- Gashaw, I., & Shashirekha, H. L. (2019). Amharic-Arabic Neural Machine Translation. *arXiv preprint arXiv:1912.13161*. <https://arxiv.org/pdf/1912.13161>
- Mohamed, M. E., Husein, E. K. M. T., & Rashed, A. M. (2023). Stacking of BERT and CNN models for Arabic Word Sense Disambiguation. *ResearchGate*. [https://www.researchgate.net/publication/373742016\\_Stacking\\_of\\_BERT\\_and\\_CNN\\_models\\_for\\_Arabic\\_Word\\_Sense\\_Disambiguation](https://www.researchgate.net/publication/373742016_Stacking_of_BERT_and_CNN_models_for_Arabic_Word_Sense_Disambiguation)
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2020). mT5: A massively multilingual pre-trained text-to-text transformer. *NAACL*. [https://huggingface.co/docs/transformers/en/model\\_doc/mt5](https://huggingface.co/docs/transformers/en/model_doc/mt5)

## **Appendix**

Other models tested but were disqualified for further development/tuning:

- Falcon-7b
- mBERT
- AraBert2
- T5
- Mistral 7B
- aya-101
- Qwen

## **Authors contributions**

Meher worked on initial data ingestion and EDA. Her modelling was focused on mT5 including task specific training, fine tuning and the addition of an additional dataset. She also worked on the final paper drafting, structure, writing and editing.

Naveed built the dataset used for this project and it was his vision. He wrote the initial proposal and completed the data ingestion in the format that was ultimately used. He also worked on Opus ar-en modelling (fine tuning etc) along with testing a host of other models. He also added to the final paper in a meaningful way.

## **AI & Other Assistance Use Statement**

AI assistance was used for code debugging / fixes and help on recurrent failures with mT5 deployment. We also used AI for final structuring, spell check and editing of the paper. We relied on homework assignments for starter code.