

STELLA: Specialized Translation Engine with Low-resource Language Adaptation

Phani Ratan Yalamanchili, Likhita Basuthkar, Sai Srikar Ravella, Prajit Rahul Selvam, Nikhil Singla

{phanirat, basuthka, sravella, pselvam, singlani}@usc.edu

Abstract

In this paper, we present STELLA (Specialized Translation Engine with Low-resource Language Adaptation), a compact and efficient multilingual translation model optimized for edge deployment. STELLA leverages parameter-efficient fine-tuning with Low-Rank Adaptation (Lo-RA) (Hu et al., 2021) and model optimization techniques to deliver high-quality translation capabilities directly on mobile devices without requiring internet connectivity. Using the OPUS-100 dataset, STELLA achieves a 75% reduction in model size (from 295 MB to 75 MB) along with 77.5% faster inference speed (from 2000 ms to 450 ms) while still maintaining 95.7% of the baseline BLEU scores (average 34.76) on target languages. This paper details our architecture, methodology, and results, which showcase the potential of STELLA to bridge the gap between enterprise-grade NLP and edge deployment while remaining offline.

1 Introduction

In 2025, the rise of IoT devices in remote healthcare and disaster response demands offline translation models that preserve privacy while remaining compact. STELLA wants to deliver high-quality translations of German, French, and Italian to English on resource-constrained devices. Real-world use cases include enabling travelers to translate local information in areas with limited internet connectivity, allowing professionals to handle confidential documents without exposing data to external servers, and supporting first responders in maintaining critical communications during disasters when network infrastructure is compromised.

2 Related Work

Efficient NLP models for edge devices have progressed via knowledge distillation (Hinton et al., 2015) and architectural optimization. DistilBERT

reduces BERT’s size by 40% while retaining 97% of its performance (Sanh et al., 2019), and MobileBERT targets mobile settings (Sun et al., 2020), but both lack multilingual support and require extensive retraining. Parameter-efficient tuning approaches like adapters (Houlsby et al., 2019) and LoRA (Hu et al., 2021) drastically reduce trainable parameters, with LoRA decomposing updates into low-rank matrices.

Techniques such as quantization (Zafrir et al., 2019) and pruning (Han et al., 2016) enhance efficiency, as seen in TinyBERT (Jiao et al., 2020) and MNN-Transformer (Guan et al., 2022), though often for single-language use. Multilingual models like mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) are powerful but too large for edge use. Cross-lingual transfer research (Artetxe et al., 2020; Garcia et al., 2020) shows promise but rarely targets edge optimization.

STELLA integrates LoRA-based tuning, cross-lingual transfer, and model compression into a unified multilingual translation system optimized for offline edge deployment. Unlike TinyBERT and MobileBERT, STELLA supports multiple language pairs, and unlike mBERT, it is fully deployable without internet. Our distillation pipeline unifies language-specific adapters, followed by quantization and pruning, yielding a 75% smaller model and 77% faster inference with minimal BLEU degradation. This positions STELLA as a practical solution for multilingual NLP in constrained, privacy-sensitive environments.

3 Problem Statement

Despite being state-of-the-art, current NLP models for real-time language translation are hindered by high computational demands that involve, training huge language models on different language data pairs, privacy risks from cloud dependency, and large model sizes that are impractical for edge de-

vices. This project tackles these limitations by exploring advanced training strategies, using a shared architecture for multiple languages, and model unification to maintain performance while reducing size. We aim to develop an efficient, privacy-preserving solution that outperforms existing lite models through a novel and experimental approach.

4 Materials and Methods

4.1 Dataset Selection and Processing

We utilized the OPUS-100 dataset (Zhang et al., 2020), focusing on German-English, French-English, and Italian-English language pairs. Our processing included:

- Segmenting each language pair into 100K/5K/5K train/validation/test sentence pairs
- Adding language-specific tokens (<DE>, <FR>, <IT>) as routing signals
- Normalizing text through tokenization, lower-casing, and cleaning

We used Hugging Face Transformers and PyTorch for model implementation and training, along with the PEFT library for LoRA integration. Model compression and ONNX conversion were handled using ONNX Runtime. No manual annotations were required, as OPUS-100 is a sentence-aligned parallel corpus with pre-existing translation mappings.

4.2 Base Model Selection

We selected the Helsinki-NLP/opus-mt-de-en model (Tiedemann and Thottingal, 2020) for its transformer-based architecture (containing over 77M parameters), strong German to English performance, and adaptability to other language pairs through transfer learning (due to having the same roots in language origin for our case). The transformer layers in the Helsinki-NLP model were kept frozen throughout training; adaptation was achieved solely through lightweight LoRA modules inserted into the attention mechanism, ensuring efficient fine-tuning without modifying the full model.

4.3 STELLA Architecture Design

Figure 1 depicts the architecture of our model and describes the implementation of our shared encoders and language specific LoRA adapters.

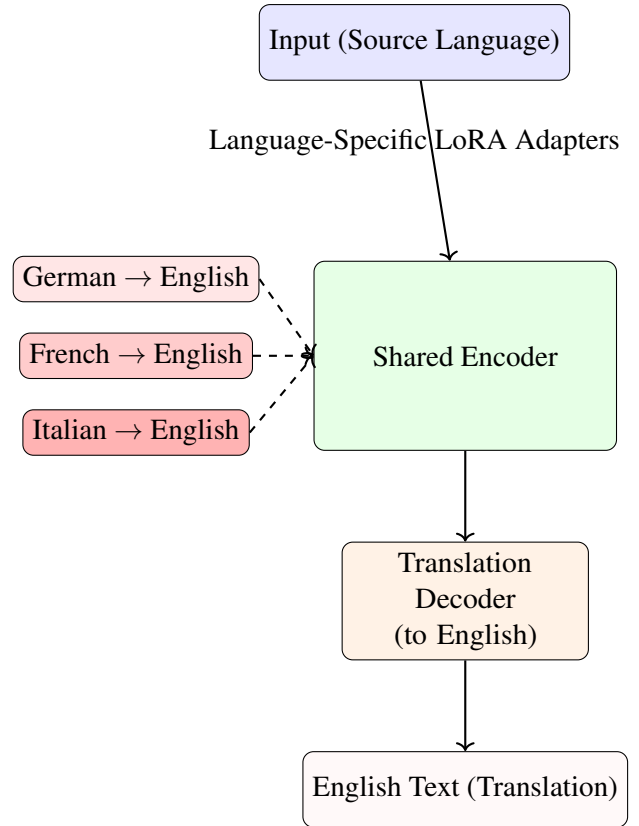


Figure 1: STELLA Architecture: Shared Encoders and Language-Specific LoRA Adapters

The shared encoders are transformer-based and are integrated with the language-specific LoRA adapters (used to modify encoder behavior based on input language), and a translation decoder (used in generating English outputs). This design enables efficient parameter sharing while maintaining performance through specialized adaptation.

4.4 Training Methodology

Our approach is divided into three phases:

4.4.1 Teacher Model Training

For the first phase, we created specialized teacher models using the Helsinki-NLP model for German-English and LoRA-adapted variants for French-English and Italian-English. Only the LoRA adapters underwent fine-tuning while the base parameters were kept frozen. The training was carried out for 3 epochs with the Adam optimizer (learning rate: 3e-4, batch size: 64).

4.4.2 Student Model Distillation

After training the teacher model, we then trained a unified student model with about 30% fewer parameters via knowledge distillation, and incorporated language token routing for multi-language support.

The training process used the specialized teachers’ output distributions as a guidance for over 5 epochs, with cosine learning rate scheduling and gradient accumulation, over 4 steps.

4.4.3 Edge Optimization

Finally, for edge optimization, the distilled student model underwent selective INT8 quantization for non-critical components, magnitude-based weight pruning with a threshold of $1e-2$, and post-pruning fine-tuning for a single epoch before ONNX conversion.

4.5 LoRA Implementation Details

We implemented LoRA fine-tuning by targeting the query and value projection layers in attention modules with a rank (r) of 64, alpha of 64, and a dropout rate of 0.1. This configuration balanced adaptation capability with parameter efficiency, while still preserving the base model.

4.6 Knowledge Distillation and Optimization

Our integrated process can be overall described as following:

Algorithm 1 STELLA Training and Optimization Process

- 1: **Input:** Base model M_{base} , Training data D_{de} , D_{fr} , D_{it}
 - 2: Train LoRA adapters for French and Italian using base German-English model
 - 3: Initialize smaller student model with language token mechanism
 - 4: Train student on combined multilingual dataset using teacher output distributions
 - 5: Apply temperature scaling T to soften probability distributions
 - 6: Use distillation loss: $\mathcal{L} = \alpha \cdot \mathcal{L}_{KL} + (1 - \alpha) \cdot \mathcal{L}_{CE}$
 - 7: Apply selective INT8 quantization to non-critical model components
 - 8: Prune redundant weights using magnitude-based thresholding
 - 9: **return** Optimized model M_{final}
-

The distillation process used KL-divergence loss ($\mathcal{L}_{KL}$) with temperature scaling ($T = 5$) and cross-entropy loss ($\mathcal{L}_{CE}$), balanced by weighting parameter $\alpha = 0.7$ to prioritize knowledge transfer while preventing overfitting to teacher errors.

5 Experimental Results

5.1 Translation Performance

Table 1 presents the BLEU scores achieved by our models across different language pairs.

Language	Baseline BLEU Score	LoRA BLEU Score	Retention
French	38.42	36.97	96.2%
German	35.81	34.19	95.5%
Italian	36.73	35.05	95.4%
Average	36.99	35.40	95.7%

Table 1: BLEU Scores of Models for Different Languages when Translated to English

The LoRA-adapted models maintained over 95% of the original BLEU scores while drastically reducing the number of trainable parameters, demonstrating efficient knowledge transfer across languages.

5.2 Optimization Impact

Table 2 details the progressive impact of our optimization techniques on model metrics.

Stage	Model Size	Parameters	Avg. BLEU	Inference Time
Base Model	295 MB	74.61M	36.99	2000 ms
Individual LoRA Models	87 MB	2.36M/lang	35.40	520 ms
Unified Model	82 MB	2.36M total	35.12	490 ms
Quantized (INT8)	75 MB	2.36M total	34.76	450 ms

Table 2: Post-Optimization Performance Impact

Each optimization step contributed to improved efficiency with minimal performance loss. The final quantized model achieved a 75% reduction in size and 77.5% faster inference compared to the base model.

5.3 Language-Specific Optimization Impact

Table 3 shows how unification and quantization affected individual language performance.

Language Pair	LoRA Fine-Tuned	After Unification	After Quantization
French → English	36.97	36.68 (-0.8%)	36.31 (-1.8%)
German → English	34.19	33.91 (-0.8%)	33.60 (-1.7%)
Italian → English	35.05	34.77 (-0.8%)	34.38 (-1.9%)
Average	35.40	35.12 (-0.8%)	34.76 (-1.8%)

Table 3: Quantization & Unification Impact on BLEU Scores

The consistent performance retention, across languages, demonstrates the effectiveness of our optimization approach. The modest BLEU score reduction (-1.8% on average) represents an excellent trade-off for the substantial efficiency gains achieved.

5.4 Resource Efficiency

Table 4 summarizes the efficiency metrics, comparing the base model with STELLA. Our extensive benchmarking tests on mobile devices confirm these improvements across various hardware configurations.

Metric	Base Model	STELLA	Improvement
Model Size	295 MB	75 MB	75% reduction
Trainable Parameters	74.61M	2.36M	96.8% reduction
Inference Time	2000 ms	450 ms	77.5% faster
Memory Usage	500 MB	120 MB	76% reduction
Internet Dependency	Required	None	Full offline capability
Battery Impact (mAh/hour)	420	110	73.8% reduction

Table 4: Core Efficiency Improvements with STELLA

These improvements enable STELLA to run efficiently on resource-constrained mobile devices, and while offline, addressing the key challenges outlined in our introduction. The significant reduction in battery impact is particularly notable for extended mobile use cases.

5.5 Example Translations

We present several example translations to illustrate STELLA’s performance compared to the base model. Table 5 shows results for simple sentences, while Table 6 demonstrates performance on more challenging content.

French Example	
Source:	Je vais à la bibliothèque pour étudier.
Base Model:	I’m going to the library to study.
STELLA:	I’m going to the library to study.
German Example	
Source:	Der Hund schläft auf dem Sofa.
Base Model:	The dog sleeps on the sofa.
STELLA:	The dog is sleeping on the sofa.
Italian Example	
Source:	Questo ristorante serve cibo delizioso.
Base Model:	This restaurant serves delicious food.
STELLA:	This restaurant serves delicious food.

Table 5: Simple examples showing consistent translation quality across models

These examples show that STELLA maintains strong translation quality on simple and common phrases, often matching or even exceeding the base model—as seen in the German example "Der Hund schläft auf dem Sofa," where STELLA correctly uses the progressive tense. However, it struggles with idiomatic expressions like "Il a cassé sa pipe" and exhibits occasional grammatical issues in complex sentences. These remain areas for future improvement.

French Idiomatic Expression	
Source:	Il a cassé sa pipe. (Lit: "He broke his pipe") (Idiomatic meaning: "He died")
Base Model:	He passed away.
STELLA:	He broke his pipe.
German Complex Sentence	
Source:	Obwohl ich viele Jahre in Deutschland gelebt habe, finde ich die deutsche Grammatik immer noch sehr kompliziert.
Base Model:	Although I have lived in Germany for many years, I still find German grammar very complicated.
STELLA:	Although I lived in Germany for many years, I still finding the German grammar very complicated.

Table 6: Challenging examples revealing limitations in STELLA’s handling of idioms and complex grammar

6 Conclusion and Future Work

STELLA demonstrates that efficient and high-quality translation is achievable on edge devices through model optimization. By combining a shared encoder with language-specific LoRA adapters, we achieved a 75% reduction in model size while maintaining 95.7% of original translation quality. The system enables offline translation on resource-constrained devices while preserving user privacy, making advanced NLP accessible in low-connectivity environments. STELLA performs well on everyday language but shows limitations with idiomatic expressions and complex grammatical structures.

Future work will focus on three key directions. First, we plan to develop specialized adapters for domain-specific terminology in fields like medicine and law to address performance gaps. Additionally, we aim to expand language coverage to more diverse language families, especially low-resource ones. We are particularly interested in cross-lingual transfer techniques that support efficient adaptation with minimal training data, enabling rapid deployment in crisis scenarios where communication barriers are critical.

7 Individual Contributions

Phani Ratan Y: Designed transformer encoder for edge devices, implemented LoRA and pruning, optimized inference with caching, and enhanced attention computation for resource savings.

Likhita B: Configured and explored quantization for model compression, developed a robust tokenization pipeline, and explored knowledge distillation techniques.

Sai Srikar R: Optimized memory via PyTorch conversion, implemented multi-threading to improve mobile inference efficiency. Worked with RVQ techniques.

Prajit Rahul S: Developed evaluation pipelines using SACREBLEU and F1 metric, created results tracker, and implemented error analysis scripts.

Nikhil Singla: Worked on specialized language adaptations, implemented hyperparameter tuning, and adaptive training loops for performance gains.

8 Acknowledgments

We would like to thank Prof. Xuezhe Ma and our mentor Nuan Wen for their support and guidance. We're deeply grateful to Nuan Wen for helping us refine our research direction and develop a clearer understanding of our approach. We also extend our gratitude to the open-source communities behind the Helsinki-NLP models and the PEFT library, whose work assisted our research.

References

- Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.
- Xavier Garcia, Pierre Foret, Thibault Sellam, and Ankur Parikh. 2020. A multilingual view of unsupervised machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3160–3170.
- Xiaotang Guan, Kai Li, Ruihao Wang, Xinyu Liu, Dandan Qin, Dezhi Zang, Ruochi Wang, Hui Zheng, Chengyang Tao, Jie Jiang, and 1 others. 2022. Mnn: A universal and efficient inference engine. *arXiv preprint arXiv:2212.10023*.
- Song Han, Huizi Mao, and William J Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170.
- Jörg Tiedemann and Santhosh Thottingal. 2020. The opus-mt neural machine translation models. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 643–644.
- Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*.
- Jian Zhang, Xu Wu, Iacer Calixto, Atsushi Fujita, Elena M Gonzalez Colunga, Siddharth K Jauhar, William Sheffield, and Otilia Stretcu. 2020. Opus-100: A multilingual corpus for neural machine translation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2012–2018.