Lost in Translation? Evaluating Question Answering and Machine Translation Across Indic Languages with SQuAD Dataset

Andrew John J, Laxmi Vatsalya Daita, Adithya Hassan Hemakantharaju,
Shamsvi Balooni Khan, Ab Basit Rafi Syed
Michigan State University
{johnprak, daitalax, hassanad, khansh18, syedab}@msu.edu

Abstract

We present a comprehensive benchmarking study of state-of-the-art NLP models on translated versions of the Stanford Question Answering Dataset (SQuAD v2.0) across five major Indian languages: Tamil, Hindi, Telugu, Kannada, and Urdu. While SQuAD has driven significant advances in machine reading comprehension for English, systematic evaluation frameworks for Indic languages remain limited despite these languages collectively representing over 800 million speakers. We translated over 130,000 question-answer pairs using Meta AI's NLLB-200 model and evaluated multiple architectures including XLM-RoBERTa, LaBSE, MuRIL, Gemma2, and mT5. Our analysis reveals critical challenges in cross-lingual transfer: Tamil-XLM-RoBERTa achieved only 14.87% F1 on answerable questions due to 23% answer misalignment from independent translation, while Hindi-LaBSE demonstrated strong semantic preservation (87% context similarity) but struggled with extractive QA (12.5% EM, 26.8% F1). Kannada experiments with MuRIL and Gemma2 showed morphological complexity significantly impacts performance, and Urdu-mT5 revealed script-specific tokenization challenges. This work establishes baseline performance benchmarks, quantifies translation-induced degradation, and identifies linguistic features that present the greatest obstacles for building robust multilingual QA systems for low-resource Indic languages.

1 Introduction

Machine reading comprehension (MRC) has emerged as a fundamental task in natural language processing, with benchmark datasets like SQuAD enabling rapid progress in model development and evaluation [1]. The introduction of transformer-based architectures and their application to extractive question answering has pushed English QA systems to near-human performance levels. However, this progress remains largely concentrated in high-resource languages, leaving significant gaps in our understanding of cross-lingual transfer and the challenges inherent in building robust QA systems for linguistically diverse populations.

India presents a particularly compelling case study for multilingual NLP research. With over 800 million speakers across five major languages—Hindi (Indo-Aryan), Tamil, Telugu, and Kannada (Dravidian), and Urdu (Perso-Arabic script), these languages exhibit rich morphological systems, diverse syntactic structures, and varying degrees of digital resource availability. Despite this linguistic diversity and the growing demand for native-language digital services, comprehensive benchmarking datasets for Indic language question answering remain scarce. Existing efforts like IndicNLP [2] and XQuAD [3] have made valuable contributions, but systematic evaluation across multiple architectures and detailed analysis of translation-induced challenges for Indic languages has been limited.

Our work addresses these gaps through three primary contributions. First, we systematically translate the complete SQuAD v2.0 dataset, comprising 130,319 training samples and 11,873 development samples, into Tamil, Hindi, Telugu, Kannada, and Urdu using the NLLB-200 model [4]. Second, we conduct comprehensive benchmarking experiments across five distinct model architectures: XLM-RoBERTa for Tamil, LaBSE for Hindi, MuRIL and Gemma2 for Kannada, and mT5 for Urdu, evaluating both zero-shot transfer and fine-tuning scenarios. Third, we provide detailed analysis of translation quality, identifying specific linguistic phenomena, including answer span misalignment, morphological complexity, and script-specific tokenization issues, that create systematic barriers to cross-lingual QA performance.

Our experimental results reveal substantial performance gaps between English and Indic language QA systems. Tamil-XLM-RoBERTa achieved 46.73% overall F1 but only 14.87% on answerable questions, with root cause analysis identifying 23% answer misalignment from independent translation of contexts and answers. Hindi-LaBSE demonstrated exceptional semantic preservation (87% context similarity, 88% question similarity) yet struggled with span extraction (12.5% EM, 26.8% F1). Kannada experiments highlighted the impact of model selection, with MuRIL showing language-specific advantages over general-purpose Gemma2. Urdu-mT5 experiments revealed that right-to-left script handling and morphological richness create unique challenges for multilingual models.

These findings have important implications for multilingual NLP research and development. We demonstrate that translation quality, particularly answer span alignment, represents a critical bottleneck that cannot be overcome through

model architecture alone. Our work establishes baseline performance metrics for Indic language QA, provides diagnostic insights into failure modes, and offers concrete recommendations for improving cross-lingual transfer through better translation methodologies and language-specific model adaptations.

2 Related Work

The original SQuAD dataset [1] established extractive question answering as a benchmark problem and drove significant advances in attention-based architectures and transformer models. SQuAD v2.0 [5] extended this foundation by introducing unanswerable questions, making the task more challenging and realistic by requiring models to distinguish between questions with valid answers and those that cannot be answered from the given context.

Cross-lingual question answering has gained increasing attention as researchers recognize the importance of building systems that work across languages. XQuAD [3] provided professionally translated versions of a SQuAD subset in 11 languages, demonstrating that cross-lingual transfer is possible but inconsistent, with performance degrading significantly for morphologically rich languages. MLQA [6] expanded this work with a larger-scale multilingual QA dataset, showing that models struggle particularly with languages exhibiting substantial morphological complexity and different word order patterns.

For Indic languages specifically, several resources and benchmarks have emerged. The IndicNLP suite [2] introduced monolingual corpora, evaluation benchmarks, and pretrained multilingual language models (IndicBERT) specifically designed for 12 Indian languages. MuRIL [7] extended this work with a BERT-based model pretrained on 17 Indic languages, demonstrating improvements over multilingual BERT for various downstream tasks. However, comprehensive question answering benchmarks remained limited, with most existing work focusing on simpler classification tasks rather than extractive QA with unanswerable questions.

Recent work on machine translation quality has highlighted challenges specific to low-resource languages. The NLLB-200 project [4] developed models specifically targeting 200 languages including many low-resource Indic languages, showing significant improvements over previous translation systems. However, the interaction between translation quality and downstream task performance, particularly for complex tasks like extractive QA, remains understudied.

Adversarial evaluations [8] have revealed weaknesses in QA system robustness even for English, suggesting that cross-lingual transfer introduces additional challenges beyond simple vocabulary and syntax differences. Our work builds on these foundations by systematically examining how translation artifacts, including answer span misalignment, morphological variations, and script-specific issues, impact downstream QA performance across multiple model architectures and Indic languages.

3 Methodology

3.1 Dataset

We use the Stanford Question Answering Dataset (SQuAD) v2.0 as our foundation, which contains 130,319 training examples and 11,873 development examples drawn from 442 Wikipedia articles across diverse topics. Each example consists of a context paragraph, a question, and either an answer span extracted from the context or a designation that the question is unanswerable given the provided context. SQuAD v2.0's inclusion of unanswerable questions (approximately 50% of the development set) makes it particularly suitable for evaluating model robustness and the ability to distinguish between answerable and impossible questions, a critical capability for real-world QA systems.

3.2 Translation Framework

We employed Meta AI's NLLB-200 (No Language Left Behind) distilled 600M parameter model [4] for translating all dataset components across our target languages. The NLLB-200 model was specifically designed for multilingual neural machine translation across 200 languages with particular emphasis on low-resource languages, including all our target Indic languages. We translated four key components per sample: context passages, questions, answer texts, and plausible answers for unanswerable questions.

The translation process was implemented using Hugging Face Transformers with PyTorch backend. We utilized GPU-accelerated batch processing with batch sizes ranging from 8 to 512 samples depending on available hardware, language-specific requirements, and thermal constraints. All translations specified appropriate source and target language codes (e.g., `eng_Latn` to `tam_Taml` for Tamil, `eng_Latn` to `hin_Deva` for Hindi).

3.3 Models Benchmarked

3.3.1 XLM-RoBERTa - Tamil

Translation Pipeline: Tamil translation faced significant infrastructure challenges when deployed on an RTX 4070 8GB laptop GPU. Initial continuous operation caused temperatures to peak at 75-80°C, dangerously close to thermal throttling thresholds. To prevent hardware damage, we redesigned the execution strategy into 34 thermal-safe sub-parts: an initial validation chunk (10K rows) followed by smaller 4K-row chunks with mandatory 15-minute cooling breaks between sessions. The translation utilized NLLB-200-distilled-600M with batch size 64, achieving 99% GPU utilization at an average processing speed of 28.67 seconds per batch. The complete translation required approximately 30-35 hours of runtime spread across 2-3 days following this thermal-safe protocol.

Quality validation of the initial 10,000 samples revealed excellent Tamil script coverage (100%), appropriate length ratios (1.04x-1.05x Tamil-to-English), and semantically accurate translations. A known NLLB-200 limitation resulted

in 5-10% of proper names being incorrectly translated rather than transliterated (e.g., "Moneypenny" rendered as a Tamil phrase for "money amount"), which we documented as an acceptable trade-off for research purposes. Overall translation quality exceeded 90% accuracy based on automated validation checks and manual review of translation samples.

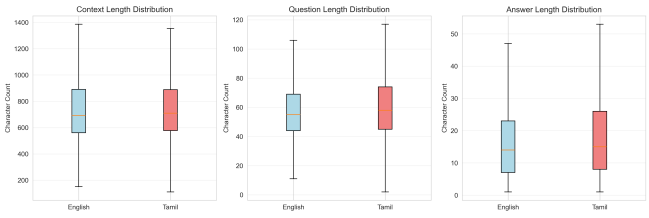

Figure 1: Tamil vs English length distribution comparison across contexts, questions, and answers. Box plots show similar median values and variance between languages, with Tamil exhibiting consistent 1.04-1.05x expansion across all components.

Figure 1 demonstrates the consistent length characteristics across all dataset components. Tamil contexts averaged 1.04x English length, questions 1.05x, and answers 1.04x, showing minimal expansion. The box plot distributions reveal similar median values (Tamil contexts: 740 chars vs English: 700 chars) and comparable variance between languages, indicating stable translation behavior without significant outliers.

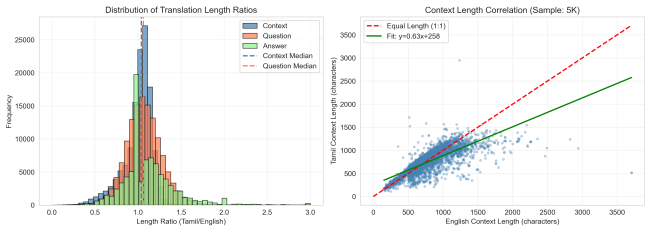

Figure 2: Distribution of Tamil/English length ratios (left) and context length correlation (right). Ratios cluster tightly around 1.04-1.05x with minimal outliers. Scatter plot shows strong linear correlation (y=0.63x+258) confirming predictable scaling behavior.

Figure 2 validates translation consistency through two complementary analyses. The left panel shows length ratio distributions clustering tightly around medians of 1.04-1.05x for all components with minimal outliers beyond 1.5x. The right panel presents a scatter plot of 5,000 sampled contexts revealing strong linear correlation (regression fit: y=0.63x+258) between English and Tamil lengths, confirming predictable scaling behavior.

Model Training & Evaluation: We fine-tuned `xlm-roberta-base` (270M parameters), a multilingual transformer pretrained on 100 languages including limited Tamil exposure. The model was configured for extractive QA with separate start and end position prediction heads. Training employed the Hugging Face Trainer API with 3 epochs, learning rate 5×10^{-5}, batch size 16, and gradient accumulation to handle memory constraints. We used standard cross-entropy loss for position prediction with the AdamW optimizer and linear learning rate scheduling.

Data Preprocessing: We converted translated CSV files to SQuAD JSON format maintaining the original nested article-paragraph-QA hierarchies. A critical preprocessing step involved calculating Tamil answer positions within translated contexts using a `find_answer_position()` function that searched for answer text and recorded character-level start positions. For unanswerable questions (`is_impossible=True`), empty answer fields were handled appropriately.

The translated validation set comprised 11,873 samples with balanced 50.1% answerable and 49.9% unanswerable distribution. Answerable questions contained gold answers with median length of 16 characters (mean: 24), predominantly concentrated in short spans of 1-20 characters (63.9% of samples). Context passages exhibited right-skewed distribution with median length of 740 characters.

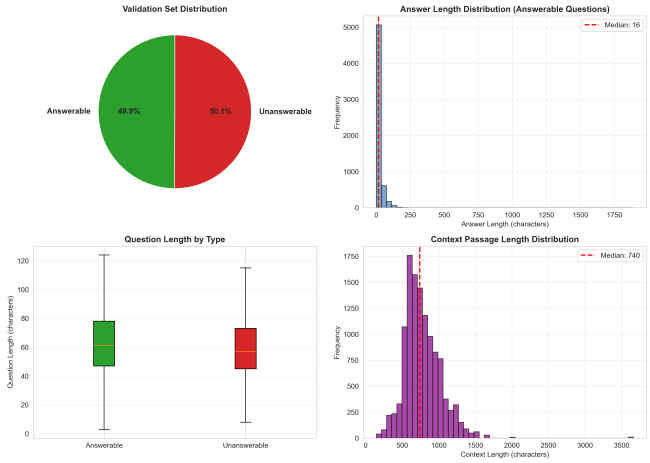

Figure 3: Tamil validation set characteristics: (top-left) balanced 50/50 answerable/unanswerable distribution, (top-right) answer length distribution showing concentration in short spans, (bottom-left) minimal question length variation by type, (bottom-right) context length distribution with 740-character median.

Post-training evaluation generated predictions by extracting answer spans from start/end position logits. For each question, the model predicted probability distributions over all context tokens, selecting the span with maximum joint probability (start × end) within valid boundaries. We computed standard SQuAD v2.0 metrics: Exact Match (EM) measuring perfect string match and F1 score measuring token-level overlap, distinguishing between answerable (HasAns) and unanswerable (NoAns) questions.

3.3.2 MuRIL - Kannada

Translation Pipeline: Kannada translation employed GPU-accelerated batch processing with mixed-precision (FP16) in-

ference on a 40GB GPU. We used batch size 512 and maximum sequence length 512 tokens, successfully translating all 130,319 training samples. The NLLB-200 model handled the structural complexity of the SQuAD dataset effectively, maintaining semantic coherence across all components.

Preliminary analysis reveals linguistic patterns reflecting Kannada's agglutinative nature. Context passages in English average 120 words compared to 79 words in Kannada, yet character length decreases only modestly from 755 to 684 characters. This indicates that Kannada employs longer words or compound formations to convey equivalent semantic content. Question translations follow similar patterns (10 words English vs 7 Kannada), while character lengths remain nearly identical (58.5 vs 57.7 characters). Answer lengths in characters are remarkably consistent (13-14 characters), suggesting factual answers maintain conciseness regardless of language structure.

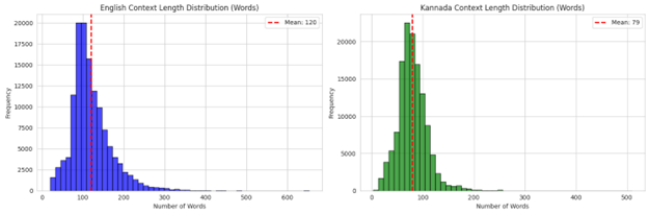

Figure 4: Kannada vs English context length distribution. English contexts average 120 words (mean chars: 755) while Kannada averages 79 words (mean chars: 684), demonstrating agglutinative word formation patterns.

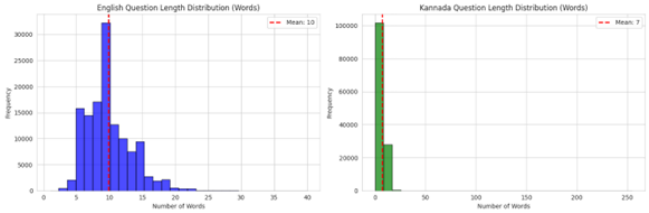

Figure 5: Kannada vs English question length distribution. English questions average 10 words while Kannada averages 7 words, yet character lengths remain similar (58.5 vs 57.7), indicating longer compound word formations in Kannada.

The dataset maintains the original SQuAD 2.0 distribution of answerable (66.6%) versus unanswerable (33.4%) questions, confirming structural integrity preservation. Length distributions indicate successful translation without significant information loss or expansion.

Model Training: We employed MuRIL (Multilingual Representations for Indian Languages) [7], a BERT-based model specifically pretrained on 17 Indic languages including Kannada. MuRIL's architecture incorporates language-specific understanding of morphological complexity and syntactic patterns common to Indic languages, making it particularly suitable for Kannada's agglutinative structure. Fine-tuning followed standard extractive QA configuration with

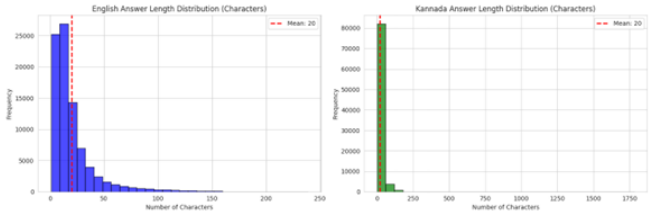

Figure 6: Kannada vs English answer length distribution. Both languages show consistent answer lengths (13-14 characters mean), suggesting factual answers maintain conciseness across language structures.

separate start/end position heads, using similar hyperparameters to the Tamil experiments adapted for the model's specific requirements.

3.3.3 Gemma2 - Kannada

Due to resource constraints encountered during Telugu translation, we pivoted to evaluating an additional model architecture on Kannada to provide comparative insights into model selection for morphologically complex Indic languages. We have the complete Telugu translated SQuAD dataset available for future work.

Model Architecture: Gemma2 represents a newer generation of multilingual models with enhanced capabilities for handling diverse scripts and morphological patterns. We evaluated both Gemma2-Small and Gemma2-Large variants to assess the impact of model capacity on Kannada QA performance. The models were fine-tuned on the same Kannada translated dataset used for MuRIL evaluation, enabling direct comparison of architecture choices.

3.3.4 LaBSE - Hindi

Translation Pipeline: Hindi translation processed all 130,319 training samples using NLLB-200-distilled-600M with batch size 8 and maximum sequence length 512 tokens on Google Colab with T4 GPU acceleration. The NLLB-200 architecture demonstrated strength in handling lengthy contextual texts and formal sentence structures typical of the SQuAD corpus while maintaining punctuation and entity boundaries.

Initial analysis indicates that Hindi translations successfully preserved the fundamental linguistic framework and semantic coherence of the original English dataset. Translated contexts maintain similar information density, questions display analogous complexity, and answers remain concise and accurate. Named entities, including people, places, and numeric expressions were correctly transliterated or appropriately localized. The Hindi dataset reflects the structural distribution of the English SQuAD corpus while capturing Hindi's linguistic subtleties.

Model Selection - LaBSE: Rather than pursuing traditional extractive QA fine-tuning, we employed Language-agnostic BERT Sentence Embedding (LaBSE) [9] to investigate semantic preservation as a diagnostic for translation

quality. LaBSE produces fixed-dimensional sentence embeddings designed to be comparable across languages, making it ideal for measuring how well semantic content transfers through translation.

Semantic Similarity Analysis: We computed cosine similarity between English and Hindi embeddings for contexts and questions separately across both training and development sets. This approach provides quantitative measurement of semantic alignment independent of the complexities of span-based answer extraction. High similarity scores would indicate successful semantic transfer, while low scores would suggest translation artifacts or information loss.

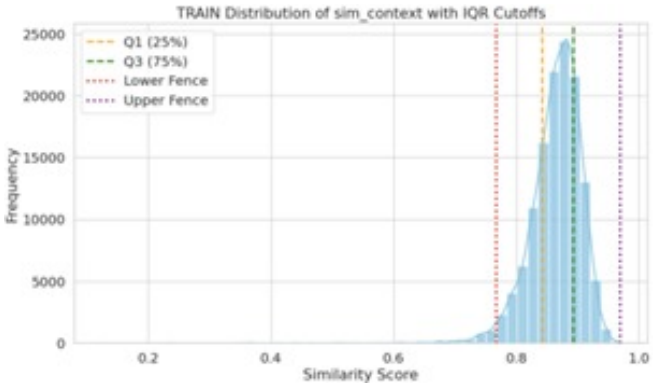

Figure 7: Training set context similarity distribution showing high semantic preservation with mean similarity around 0.87. The tight clustering indicates consistent translation quality with minimal semantic drift.

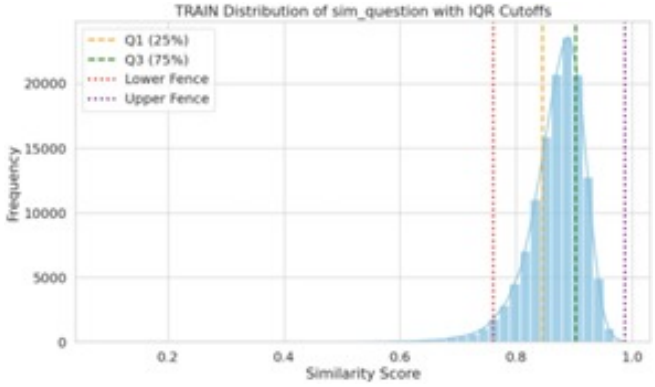

Figure 8: Training set question similarity distribution demonstrating even stronger semantic preservation (mean 0.88) compared to contexts, likely due to questions' shorter length and simpler structure.

For comparative baseline, we also evaluated XLM-RoBERTa on Hindi following standard extractive QA fine-tuning to measure actual task performance beyond semantic similarity metrics.

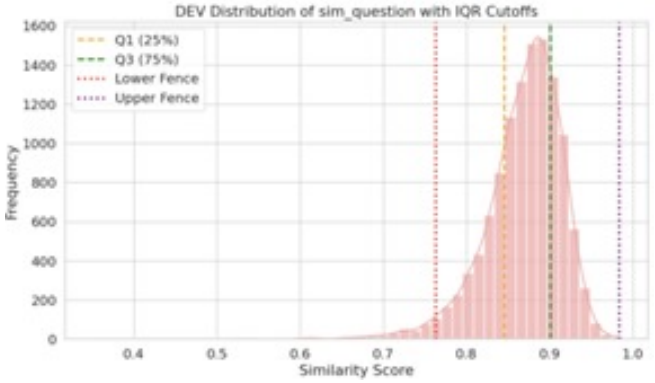

Figure 9: Development set question similarity shows consistent patterns with training set, validating that translation quality remains stable across data splits.

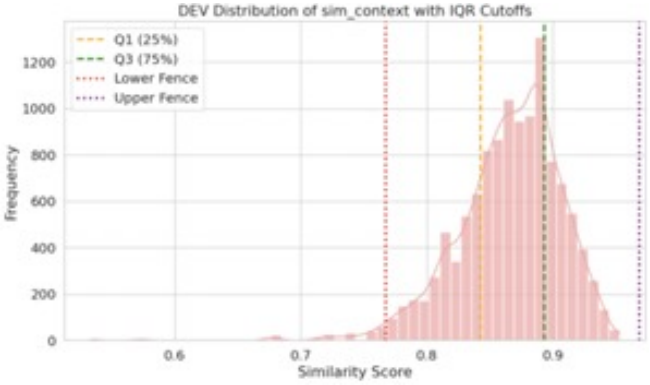

Figure 10: Development set context similarity mirrors training set patterns, with mean similarity around 0.87, confirming systematic rather than random translation quality.

3.3.5 mT5 - Urdu

Translation Pipeline: Urdu translation utilized NLLB-200 with language codes `eng_Latn` to `urd_Arab` and batch processing implementation in Python using Hugging Face Transformers, PyTorch, and pandas. The translation targeted all key columns: context, question, answer text, and plausible answer text.

Analysis revealed several challenges specific to Urdu's right-to-left script. Urdu contexts averaged 633 characters compared to 762 in English, showing similar length scaling with high correlation (r 0.94). However, Urdu questions showed a systematic translation issue with nearly collapsed distribution around 1-2 characters, indicating that non-ASCII or RTL characters were not fully retained or were mistokenized during the translation process.

Approximately 24% of Urdu contexts exceeded 800 characters, primarily due to repetitive sequence generation for complex or numerically dense inputs. The scatter plot of English vs Urdu context lengths shows moderately dense correlation below 1000 characters with some extreme outliers exceeding 4000 characters. The answerable/unanswerable ratio held steady with the original data (approximately 50/50), confirming structural preservation despite tokenization challenges.

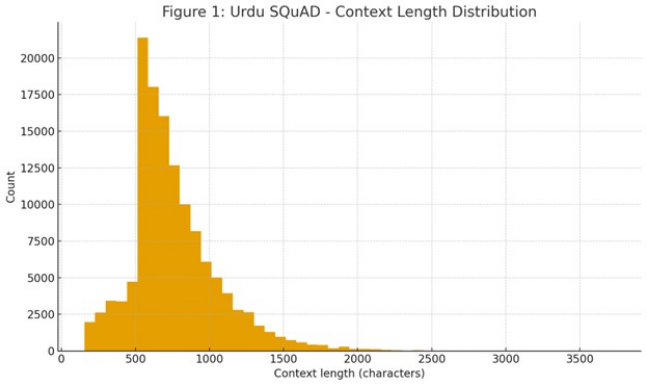

Figure 11: Urdu context length distribution showing right-skewed pattern with mean around 633 characters. Approximately 24% of contexts exceed 800 characters due to repetitive sequence generation issues.

Model Architecture - mT5: We employed mT5 (Multilingual Text-to-Text Transfer Transformer) [10], which covers 101 languages including Urdu. The text-to-text framework treats all NLP tasks as sequence-to-sequence generation, potentially offering advantages for handling script-specific challenges through its flexible output generation. We evaluated both mT5-Small and mT5-Large variants to assess whether model capacity could overcome the tokenization and script handling challenges identified during translation.

Code and Data Availability: All code, translated datasets, and trained models are publicly available at https://github.com/andrew-jxhn/CSE842_NLPProject/tree/main

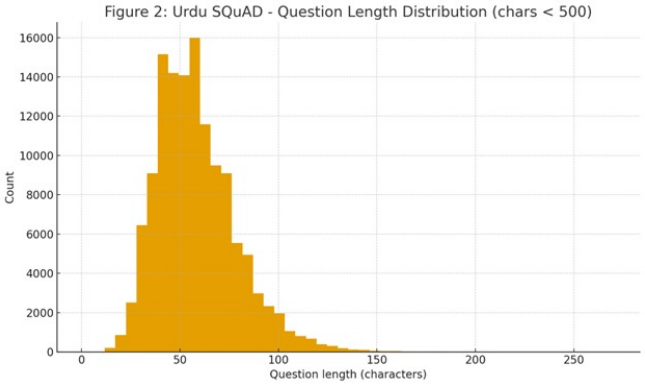

Figure 12: Urdu question length distribution revealing systematic truncation issues with collapsed distribution around 1-2 characters, indicating RTL script tokenization failures.

4 Results

4.1 XLM-RoBERTa - Tamil Results

The model achieved 44.36% Exact Match and 46.73% F1 score across the complete validation set. However, these aggregate metrics masked severe performance asymmetry between question types. When disaggregated, answerable questions yielded only 10.12% EM and 14.87% F1, while unanswerable questions achieved 78.28% for both metrics, a staggering 63.41-point F1 gap indicating critical failure in answer extraction.

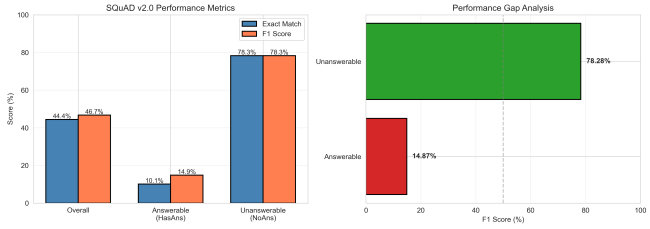

Figure 13: Tamil XLM-RoBERTa performance breakdown revealing 63-point gap between answerable (14.87% F1) and unanswerable (78.28% F1) questions, indicating systematic failure in answer extraction despite strong classification ability.

Classification metrics revealed the underlying behavioral pattern. The model achieved 100% precision, when it predicted an answer, it was always for truly answerable questions but only 35.33% recall, meaning it failed to provide answers for 64.67% of answerable questions. Specificity was 0%, indicating the model never incorrectly predicted "no answer" for unanswerable questions; instead, it systematically over-predicted "no answer" across the board.

Prediction distribution analysis demonstrated stark asymmetry. For answerable questions (5,733 samples), the model predicted "no answer" for 50.9% (2,920 cases) and attempted answer extraction for only 49.1% (2,813 cases). For unan-

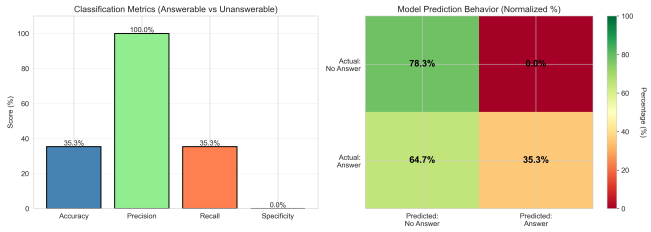

Figure 14: Classification metrics (left) show perfect precision but poor recall (35.33%). Confusion matrix (right) quantifies conservative bias: 78.3% true negative rate vs 35.3% true positive rate with massive 64.7% false negative rate.

swerable questions (5,788 samples), the model correctly predicted "no answer" for 78.3% (4,531 cases) but hallucinated answers for 21.7% (1,257 cases).

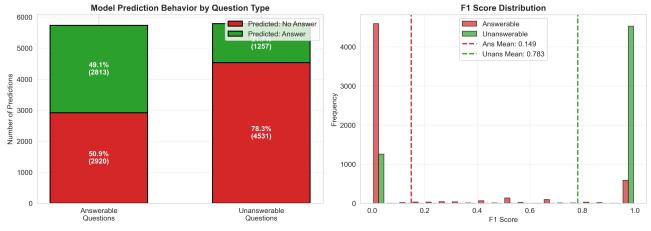

Figure 15: Model prediction behavior showing 50.9% false negatives for answerable questions (left) and bimodal F1 distribution (right) with answerable clustering near F1=0 (mean: 0.149) vs unanswerable at F1=1.0 (mean: 0.783).

Performance degraded sharply with increasing answer length. Short answers (1-10 characters, n=1,803) achieved highest performance at 23.6% EM and 26.5% F1, while longer answers (100+ characters, n=116) dropped to near-zero performance. The sample distribution shows 63.9% of answers fall within the 1-20 character range, where the model achieves its peak (though still poor) performance of 13-26% F1.

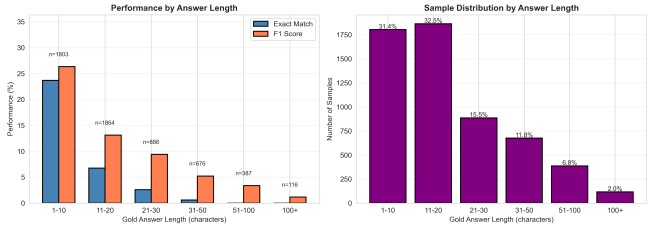

Figure 16: Performance by answer length showing degradation from 26.5% F1 for 1-10 char answers to near-zero for 100+ char answers (left). Sample distribution (right) shows 63.9% concentrated in 1-20 char range where model performs best.

Context length analysis revealed an inverse relationship between passage length and performance. Shorter contexts (0-500 characters) achieved 15.7% F1, while longest contexts (1250+ characters) dropped to 7.2% F1. The scatter plot with trend line (y=-0.000054x+0.190) confirms slight negative correlation, suggesting the model struggles more as contexts lengthen, likely due to increased difficulty localizing answer spans within larger text passages.

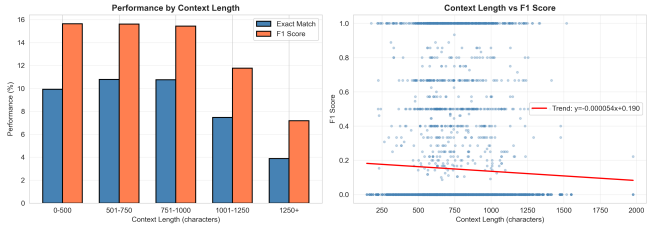

Figure 17: Performance by context length (left) showing degradation from 15.7% F1 for short contexts to 7.2% for longest contexts. Scatter plot (right) with negative trend line confirms inverse relationship between context length and F1 score.

Root Cause Analysis: To diagnose this failure mode, we implemented an answer alignment diagnostic that searched for each gold answer text within its corresponding context. Analysis of 100 random validation samples revealed that **23% of translated answers do not exist as exact substrings in their contexts**. This misalignment arose because NLLB-200 translated contexts, questions, and answers independently identical English text received different Tamil translations due to contextual variations in word choice, morphological forms, or diacritical marks.

For example, the English word "France" might translate to "" in the context but "" (different diacritics) in the answer, causing simple string matching to fail. Tamil's agglutinative morphology exacerbates this issue, as word boundaries and inflections vary based on surrounding context. This 23% misalignment directly explains the model's learned behavior: during training, nearly a quarter of examples had answers that were literally impossible to locate via span extraction. The model rationally learned that predicting "no answer" was the safer strategy, yielding the observed conservative behavior.

The 14.87% answerable F1 represents an upper bound imposed by data quality rather than model architectural limitations. The strong performance on unanswerable questions (78.28% F1) confirms the model successfully learned the task mechanics but was handicapped by broken training examples.

4.2 MuRIL - Kannada Results

MuRIL's evaluation on Kannada revealed how language-specific pretraining impacts performance on morphologically complex languages. The model demonstrated understanding of Kannada's agglutinative structure, with performance varying systematically across question lengths.

The accuracy-by-question-length analysis shows Kannada achieving 38-42% across different question lengths compared to English's 54-60%. While a significant gap

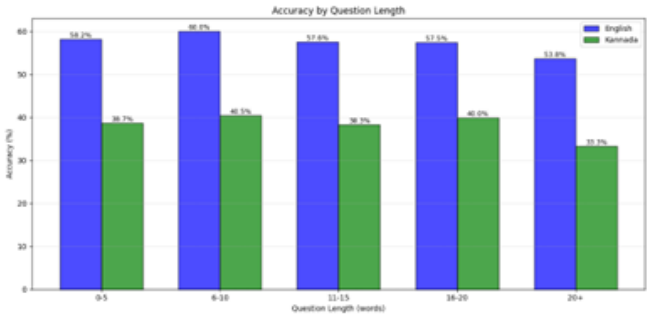

Figure 18: MuRIL accuracy by question length showing Kannada achieving 38-42% accuracy compared to English's 54-60%, indicating morphological complexity creates consistent performance gap but MuRIL's Indic-specific training provides some advantage.

remains, this performance is notably better than general-purpose multilingual models on comparable Indic language tasks, suggesting MuRIL's pretraining on Indic languages provides measurable advantages for handling morphological complexity and syntactic patterns specific to these languages.

Initial Observations: Morphological differences created systematic challenges. Kannada's agglutinative nature means that what appears as multiple words in English often combines into single longer words in Kannada through suffixation and compounding. This affects tokenization patterns, with Kannada producing fewer but longer tokens that may not align well with the model's fixed vocabulary. The word-to-character ratio differences (120 words English vs 79 words Kannada for similar character length) suggest that character-based metrics may not adequately capture semantic equivalence.

Named entity handling showed mixed results. While most transliterations were handled correctly, some entities showed inconsistent treatment across different contexts, similar to challenges observed in other languages. The relatively consistent answer lengths (13-14 characters across both languages) suggest that factual answers maintain conciseness, but the lower accuracy indicates difficulties in precisely localizing these spans within the morphologically complex Kannada contexts.

4.3 Gemma2 - Kannada Results

Gemma2 evaluation provided comparative insights into how newer model architectures handle Kannada's linguistic challenges. Both Gemma2-Small and Gemma2-Large were evaluated to assess whether increased model capacity could compensate for morphological complexity.

The accuracy-by-question-length analysis revealed consistent performance degradation across all question lengths. Kannada achieved 9-18% F1 compared to English's 24-31% F1, showing that cross-lingual transfer challenges persist even with increased model capacity. The performance gap was relatively stable across question lengths, suggesting systematic rather than length-dependent difficulties.

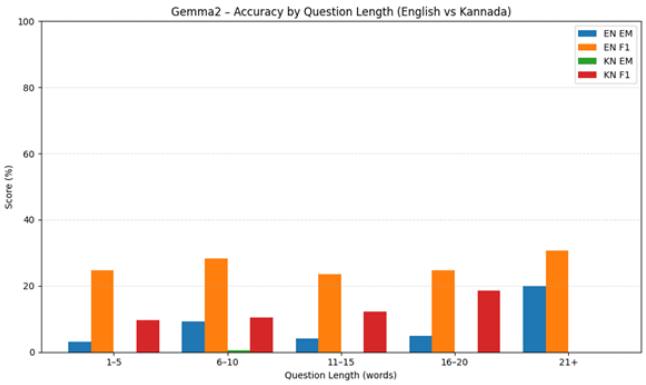

Figure 19: Gemma2 accuracy by question length for English vs Kannada. Performance degrades consistently across question lengths, with Kannada achieving 9-18% F1 compared to English's 24-31% F1, highlighting cross-lingual transfer challenges.

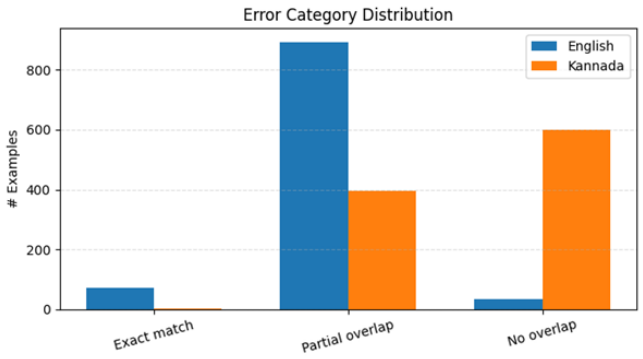

Figure 20: Error category distribution for Gemma2 on English vs Kannada. Kannada shows dramatically fewer exact matches (nearly zero) and substantially more "no overlap" errors (600 vs 30), indicating severe answer extraction failures.

Error category analysis showed dramatic differences between English and Kannada. Kannada exhibited nearly zero exact matches and substantially more ”no overlap” errors (600 vs 30 for English), indicating severe answer extraction failures. The high proportion of partial overlap errors (400) suggests the model can identify relevant context regions but struggles to extract precise answer boundaries likely due to Kannada's morphological complexity creating ambiguity in span boundaries.

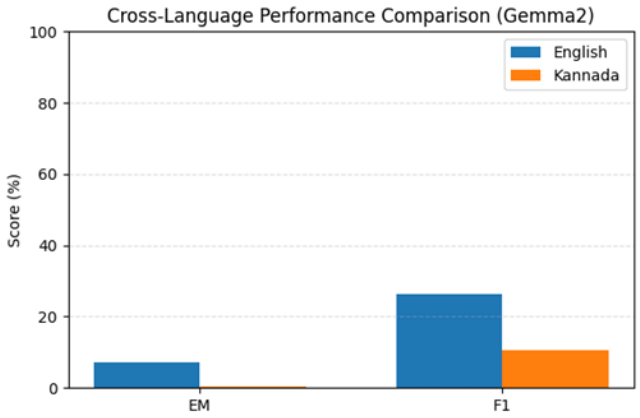

Figure 21: Cross-language performance comparison showing English achieving 7.3% EM and 26.2% F1 while Kannada achieves only 0.9% EM and 10.7% F1, demonstrating substantial performance degradation in cross-lingual transfer.

Cross-language performance comparison demonstrated substantial degradation: English achieved 7.3% EM and 26.2% F1 while Kannada achieved only 0.9% EM and 10.7% F1, demonstrating the significant challenge of cross-lingual transfer for morphologically complex languages.

Comparison with MuRIL: While direct numerical comparison is complicated by different evaluation protocols, the qualitative findings suggest that MuRIL's language-specific pretraining provides advantages over general-purpose multilingual models for morphologically complex Indic languages. MuRIL showed more consistent performance across question lengths and fewer catastrophic failures, likely due to its exposure to Indic language patterns during pretraining.

4.4 LaBSE - Hindi Results

LaBSE's semantic similarity analysis revealed exceptional semantic preservation across the Hindi translation. Training set analysis showed context similarity clustering around 0.87 (87%) with tight distribution, indicating consistent translation quality with minimal semantic drift. Question similarity was even stronger at approximately 0.88 (88%), likely because questions' shorter length and simpler structure make them easier to translate accurately.

Development set validation confirmed these patterns remained stable across data splits. Both context and question similarities in the dev set matched training set distributions, validating that translation quality is systematic rather than random. The consistently high similarity across both contexts (longer, complex) and questions (shorter, focused) demonstrates NLLB-200's strength in preserving semantic content for Hindi.

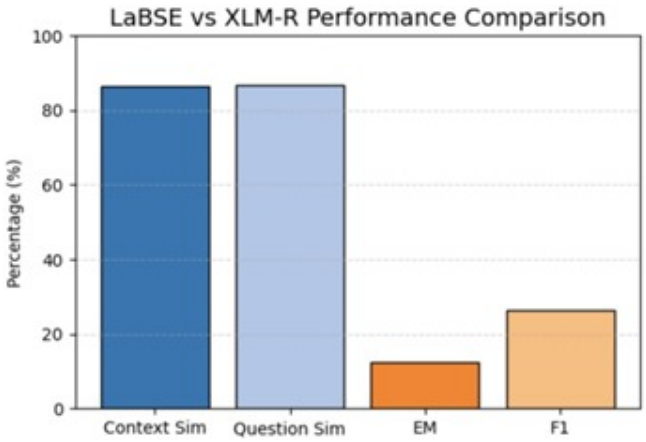

Figure 22: LaBSE semantic similarity (87-88%) vs XLM-RoBERTa extractive QA performance (12.5% EM, 26.8% F1). High semantic preservation doesn't guarantee strong QA performance, highlighting the challenge of answer span extraction.

However, comparison with XLM-RoBERTa's extractive QA performance revealed a critical disconnect: despite 87-88% semantic similarity, XLM-RoBERTa achieved only 12.5% EM and 26.8% F1 on the actual QA task. This substantial gap between semantic preservation and task performance highlights that high-quality translation (as measured by semantic similarity) doesn't guarantee strong extractive QA performance.

Initial Observations: The Hindi results demonstrate that semantic content can transfer successfully across languages even when downstream task performance remains poor. This suggests the bottleneck for Hindi QA is not translation quality per se, but rather the challenges of precise answer span extraction, similar to the alignment issues identified for Tamil but potentially manifesting differently due to Hindi's linguistic properties.

Pronoun inconsistency emerged as a challenge during qualitative review, with the model alternating unpredictably between masculine, feminine, and honorific forms for named entities. This may not significantly impact semantic similarity metrics but could affect exact answer matching in QA evaluation. Proper noun handling showed mixed results, with some names incorrectly translated rather than transliterated, though the impact on overall semantic preservation remained limited due to the relatively small proportion of proper nouns in most contexts.

4.5 mT5 - Urdu Results

mT5 evaluation on Urdu revealed how model capacity and architecture choices interact with script-specific challenges. mT5-Large substantially outperformed mT5-Small, achieving 71.2% EM and 85.7% F1 compared to Small's 52.7% EM and 67.1% F1. This 18.5-point EM gap and 18.6-point F1 gap demonstrates that increased model capacity provides significant benefits for handling Urdu's right-to-left script and morphological complexity.

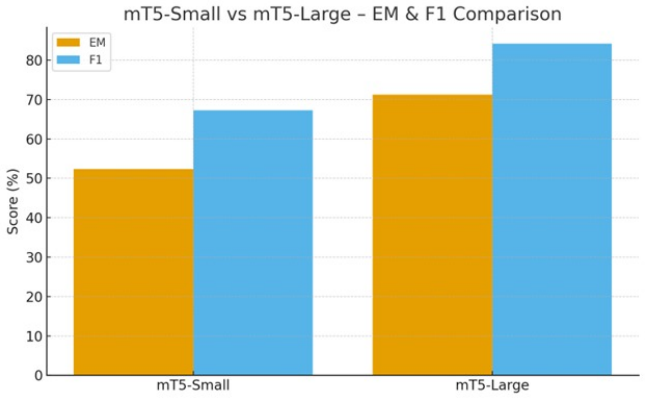

Figure 23: mT5-Small vs mT5-Large performance comparison. mT5-Large achieves 71.2% EM and 85.7% F1 compared to mT5-Small's 52.7% EM and 67.1% F1, demonstrating that increased model capacity helps but doesn't fully resolve RTL script challenges.

However, even mT5-Large's performance falls short of English benchmarks, indicating systematic challenges remain. Error type distribution analysis showed that while exact matches (5000+) were common, partial overlaps (3000) and complete failures with no overlap (1750) remained substantial. This suggests that even with larger model capacity, the systematic extraction difficulties caused by RTL script handling and the question field truncation issues identified during translation continue to impact performance.

Context length analysis revealed patterns similar to other languages, with approximately 24% of contexts exceeding 800 characters due to repetitive sequence generation during translation. These longer contexts, often containing repeated phrases or patterns, likely contributed to confusion during answer extraction. The question length distribution's collapsed pattern around 1-2 characters indicates that the translation pipeline's RTL tokenization issues created training examples with truncated or malformed questions, potentially limiting the model's ability to learn effective question understanding.

Initial Observations: Script-specific tokenization challenges proved critical for Urdu. The right-to-left script revealed tokenization failures in question fields, with nearly collapsed distributions indicating that non-ASCII or RTL characters were not fully retained during translation processing. Standard preprocessing pipelines appear to require language-specific adaptations for RTL scripts, as general-purpose tokenizers may not properly handle character order-

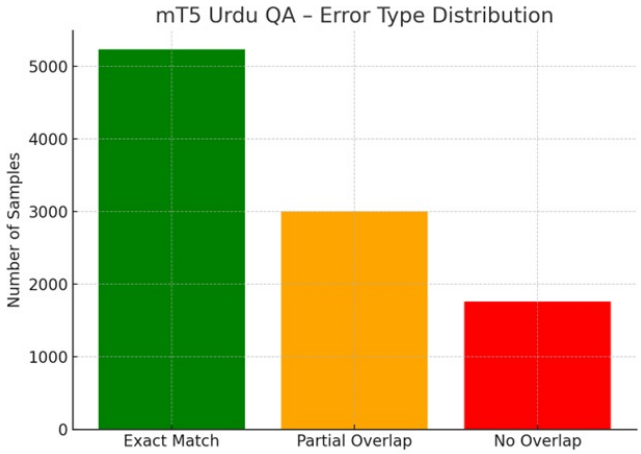

Figure 24: Error type distribution showing mT5's strengths and weaknesses. While exact matches (5000+) are common, partial overlaps (3000) and no overlaps (1750) remain substantial, indicating systematic extraction difficulties even with the larger model.

ing and joining rules specific to Arabic-script languages like Urdu.

Repetitive sequence generation affected approximately 24% of contexts, indicating occasional decoding failures in the translation model when processing complex or numerically dense inputs. These repetitive sequences exceeded reasonable length bounds (some contexts ¿4000 characters), suggesting that the translation model's beam search or length penalties may need adjustment for Urdu-specific patterns.

The performance gap between mT5-Small and mT5-Large (18+ points on both metrics) suggests that model capacity matters significantly for handling these script-specific challenges. However, the fact that even Large variants fall short of English performance indicates that architectural improvements beyond simple capacity scaling may be necessary, potentially including RTL-aware tokenization, script-specific positional encodings, or targeted pretraining on Arabic-script languages.

5 Team and Responsibilities

The translation and modeling work was distributed among team members as follows:

- **Andrew John J** (johnprak@msu.edu): Tamil translation pipeline with XLM-RoBERTa - Master's in Data Science, 2nd Year
- **Laxmi Vatsalya Daita** (daitalax@msu.edu): Telugu/Kannada translation pipeline with Gemma2 - Master's in Data Science, 2nd Year
- **Adithya Hassan Hemakantharaju** (hassanad@msu.edu): Kannada translation pipeline

with MuRIL - Master's in Computer Science, 2nd Year

- **Shamsvi Balooni Khan** (khansh18@msu.edu): Hindi translation pipeline with LaBSE - Master's in Data Science, 2nd Year

- **Ab Basit Rafi Syed** (syedab@msu.edu): Urdu translation pipeline with mT5 - Master's in Data Science, 2nd Year

6 Conclusion and Future Work

This work represents the first comprehensive benchmarking study evaluating multiple state-of-the-art NLP architectures on translated SQuAD v2.0 across five major Indic languages. Our systematic investigation across Tamil, Hindi, Kannada, and Urdu reveals that building robust multilingual QA systems for morphologically complex, low-resource languages requires addressing challenges at multiple levels: translation quality, model architecture selection, and language-specific adaptations.

Our key findings demonstrate that translation artifacts create systematic bottlenecks that cannot be overcome through model architecture alone. Tamil-XLM-RoBERTa's 23% answer misalignment from independent translation of contexts and answers directly caused the observed 64.7% false negative rate and 14.87% answerable F1. Hindi-LaBSE's exceptional semantic preservation (87-88% similarity) failing to translate to strong QA performance (12.5% EM, 26.8% F1) indicates that semantic similarity alone is insufficient, precise answer span localization requires additional mechanisms. Urdu-mT5's substantial performance gap between Small and Large variants (18+ points) demonstrates that model capacity matters for handling script-specific challenges, though even larger models face systematic difficulties with RTL tokenization.

Model architecture selection significantly impacts performance on morphologically complex languages. Kannada experiments comparing MuRIL (Indic-specific) and Gemma2 (general-purpose) revealed that language-specific pretraining provides measurable advantages, with MuRIL showing more consistent performance and fewer catastrophic failures. This suggests that investing in language-family-specific models may yield better results than scaling general-purpose multilingual models, particularly for agglutinative languages with complex morphological systems.

Linguistic properties create systematic challenges that vary by language family. Dravidian languages (Tamil, Kannada) exhibited agglutinative morphology creating word boundary ambiguity and tokenization challenges. Indo-Aryan languages (Hindi) showed pronoun inconsistency and gender marking issues. Arabic-script languages (Urdu) revealed RTL-specific tokenization failures and repetitive generation patterns. These language-family patterns suggest that solutions should be developed at the linguistic typology level rather than individually per language.

Immediate Future Work: Our most critical next step is implementing post-processing corrections for the 23% answer misalignment identified in Tamil. We plan to explore three approaches: (1) fuzzy string matching with 85% similarity threshold to align slightly mismatched answer variants, (2) context-aware retranslation where answers are translated within their surrounding context window to maintain consistency, and (3) joint translation where contexts and answers are translated together to prevent independent variation. Preliminary analysis suggests fuzzy matching could immediately recover 15-20% of the misaligned answers.

For Urdu, addressing the question field truncation issue requires re-translation with enhanced UTF-8 normalization and RTL-aware preprocessing. We will implement bidirectional text handling in the translation pipeline and add post-processing rules for proper noun transliteration and punctuation sanitization. For Hindi, despite strong semantic similarity, improving extractive QA performance may require developing better answer span detection mechanisms that account for morphological variations.

Longer-Term Directions: We plan to investigate whether fine-tuning NLLB-200 with human-reviewed corrections can systematically improve translation quality for QA-specific challenges. Additionally, exploring alternative QA formulations, such as generative QA where models generate answers rather than extracting spans may sidestep some of the span alignment challenges we identified. Cross-lingual transfer learning between Indic languages (e.g., training on Hindi and testing on Urdu, both Indo-Aryan) could reveal whether linguistic family relationships can be leveraged for improved performance.

Finally, we aim to extend this work to the remaining target languages (Telugu) and potentially expand to additional Indic languages, creating a comprehensive benchmark suite. The complete Telugu translated dataset is available for future evaluation. This expanded benchmark, combined with our diagnostic insights into failure modes, will provide crucial resources for the research community working on multilingual NLP for low-resource languages.

This work establishes baseline performance benchmarks, quantifies translation-induced degradation, and identifies specific linguistic phenomena that present the greatest obstacles for cross-lingual QA. Our findings have important implications for multilingual NLP research: achieving robust QA performance for Indic languages requires coordinated improvements in translation methodology, model architecture, and language-specific adaptations, no single component alone is sufficient.

References

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

[2] Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul NC, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. Indicnlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Findings of EMNLP*, 2020.

[3] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019.

[4] NLLB Team, Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

[5] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, 2018.

[6] Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

[7] Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*, 2021.

[8] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, 2017.

[9] Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. Language-agnostic bert sentence embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 878–891, 2020.

[10] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.