AI Project Report – Module E

Student & Project Details

- **Student Name:** Keerthi Pothireddy
- **Mentor Name:** Kartik Gupta
- **Project Title:** Miscommunication Shield Business Translator

---

1. Problem Statement

Background: International businesses lose $75 billion annually due to cultural miscommunication. Examples include direct refusals offending Japanese clients, informal language deemed unprofessional in French business, and false cognates in Spanish ("I'm embarrassed" → "Estoy embarazada" = "I'm pregnant").

Importance: With 33 million SMEs in international trade, culturally-aware communication tools are critical. Existing translators focus on linguistic accuracy but ignore cultural appropriateness.

AI Task: Build a translation system that detects cultural miscommunication risks before sending, providing real-time warnings and culturally-appropriate alternatives.

Objectives: Detect high-risk patterns, quantify risk (Low/Medium/High), generate culturally-adapted translations, provide explainable warnings.

Constraints: 5 languages (Japanese, French, Spanish, Hindi, Arabic), business context only, synthetic training data, requires human review.

---

2. Approach

System Overview: Hybrid architecture combining rule-based risk detection with LLM translation:

1. Input: message text, target language, business context
2. Risk Detection: pattern-matching scans for cultural violations
3. Translation: GPT-4 generates culturally-adapted output
4. Output: risk level, warnings, safe translation

Data Strategy:

- **Sources:** Expert-curated patterns from business communication research

- **Collection:** 50+ risk words/phrases per language; 100 synthetic examples (35% high, 35% medium, 30% low risk)
- **Preprocessing:** Text normalization, feature extraction (risk word counts, formality markers), language-specific matching

AI/Model Design:

- **Type:** Hybrid - rule-based ML for detection, GPT-4o-mini for translation
- **Risk Detector:** Scoring formula: `(High_Risk × 3) + (Medium_Risk × 1.5) + Penalties`; thresholds: ≥3 → High, ≥1 → Medium
- **Translation:** Prompt engineering with cultural context injection, temperature=0.3

Tools: Python, Jupyter, Pandas, NumPy, OpenAI API, Gradio

Design Decisions: Hybrid for explainability + accuracy, synthetic data due to lack of public datasets, 5 languages for diverse communication styles.

---

3. Key Results

Working Prototype: End-to-end system with risk detection, translation, and warnings. Processes messages in <1 second.

Example Outputs:

High Risk:

- Input: "No, this won't work" (Japanese)
- Risk: HIGH (90% confidence), Warnings: Direct refusal inappropriate
- Recommendation: ❌ DO NOT SEND
- Safe: "We appreciate your proposal and would explore alternatives..."

Low Risk:

- Input: "We appreciate your patience" (Japanese)
- Risk: LOW (80%), Recommendation: ✅ SAFE TO SEND

Evaluation:

- Test: 5 labeled examples
- Accuracy: 80%, Precision (High Risk): 100%, Recall: 67%
- Processing: <1 second/message

Performance: Excels at explicit patterns with 100% precision. Missed subtle false cognate outside pattern database.

Limitations: Pattern dependency, 5 languages only, API costs (~$0.002/message), synthetic data gaps.

4. Learnings

Technical:

- Prompt engineering with cultural context injection
- Simple count-based features outperform complex NLP for explainability
- Hybrid systems: rule-based for detection, neural for generation
- OpenAI API integration and cost optimization

System & Design:

- Explainability vs. accuracy trade-off for business users
- Data quality > quantity: 100 curated examples effective
- Modular architecture enables independent improvements
- Users need actionable recommendations, not just scores

Challenges & Solutions:

1. *No dataset*: Created synthetic data with expert patterns
2. *False positive/negative balance*: Three-tier risk levels with confidence scoring
3. *LLM consistency*: Temperature=0.3 and explicit prompt constraints
4. *Evaluation*: Created expert-labeled test cases

Future Improvements: Collect 10K+ real communications, fine-tune BERT on cultural corpus, add back-translation, build browser extension, support dialects and voice.

References & AI Usage Disclosure

Datasets: Cultural patterns from "The Culture Map" (Erin Meyer), Harvard Business Review, Collins False Friends; 100 synthetic examples (student-generated)

Tools: OpenAI GPT-4o-mini, Python (pandas, numpy, gradio), Google Colab

AI Tools Used: Claude (Anthropic) for structuring and documentation, GitHub Copilot for code completion.All core logic and algorithms independently designed and implemented.

Repository: https://github.com/keerthipothireddy0609/Language-Translation-System.git
Submission Date: [16-01-2026]