

Software Requirements Specification (SRS)

NLP-Based Language Translator Web Application

1. Introduction

- 1.1 Purpose: This document defines the detailed functional and non-functional requirements for an NLP-based Language Translator Web Application.
- 1.2 Scope: The system translates text between multiple languages using Natural Language Processing (NLP) models and Machine Learning-based translation engines.
- 1.3 Objectives: Provide accurate, context-aware, scalable, and real-time translation services accessible via web browsers.

2. Overall Description

- 2.1 Product Perspective: The system integrates NLP models such as Transformer-based Neural Machine Translation (NMT) models.
- 2.2 Product Functions: Text preprocessing, language detection, tokenization, translation using NLP model, post-processing, and output formatting.
- 2.3 User Classes: Students, professionals, researchers, and general users.
- 2.4 Operating Environment: Cloud-hosted backend, REST API services, modern browsers, mobile-responsive interface.

3. System Architecture

- 3.1 Frontend: HTML, CSS, JavaScript (React or similar framework).
- 3.2 Backend: Python (Flask/FastAPI) handling API requests.
- 3.3 NLP Engine: Pre-trained Transformer models (e.g., sequence-to-sequence neural networks).
- 3.4 Database (Optional): Store user history and logs.
- 3.5 Deployment: Cloud infrastructure (AWS/Azure/GCP).

4. Functional Requirements

- 4.1 The system shall allow users to input text up to 5000 characters.
- 4.2 The system shall automatically detect the source language using NLP language identification models.
- 4.3 The system shall tokenize input text before translation.

- 4.4 The system shall use Neural Machine Translation (NMT) for context-aware translation.
- 4.5 The system shall perform post-processing including detokenization and grammar refinement.
- 4.6 The system shall display translation within 2-3 seconds.
- 4.7 The system shall support at least 100 languages.
- 4.8 The system shall provide copy-to-clipboard functionality.
- 4.9 The system shall provide speech-to-text and text-to-speech (future scope).

5. NLP-Specific Requirements

- 5.1 Language Detection Model Accuracy should be at least 95%.
- 5.2 Translation BLEU score should meet acceptable benchmark standards.
- 5.3 The model shall handle idioms and contextual phrases.
- 5.4 The system shall support continuous learning and model updates.
- 5.5 Bias mitigation techniques should be applied in NLP model outputs.

6. Non-Functional Requirements

- 6.1 Performance: Response time < 3 seconds under normal load.
- 6.2 Scalability: Support 1000+ concurrent users using load balancing.
- 6.3 Security: HTTPS encryption, API key protection, secure authentication (if login enabled).
- 6.4 Reliability: 99% uptime excluding scheduled maintenance.
- 6.5 Usability: Responsive UI, accessible design standards.
- 6.6 Maintainability: Modular backend, documented APIs, version-controlled codebase.

7. External Interface Requirements

- 7.1 User Interface: Input field, language selectors, translate button, output field.
- 7.2 API Interface: RESTful API using JSON request/response.
- 7.3 Hardware Interface: Compatible with desktops, tablets, and smartphones.

8. Data Flow Process

- Step 1: User inputs text.
- Step 2: Language detection using NLP classifier.
- Step 3: Text preprocessing and tokenization.
- Step 4: Neural Machine Translation model inference.
- Step 5: Post-processing and formatting.
- Step 6: Output displayed to user.

9. Future Enhancements

- 9.1 Real-time voice translation.
- 9.2 Offline model deployment.
- 9.3 Document and PDF translation.
- 9.4 AI-powered grammar correction.
- 9.5 Personalized translation memory.

10. Conclusion

- The NLP-Based Language Translator Web Application leverages advanced Neural Machine Translation techniques to deliver accurate, scalable, and context-aware multilingual communication services.