–
TRANSLATED S.R.L.
VAT. IT 071 73 52 10 01
R.E.A. Roma 1015467

Junior ML Engineer Test

Functional Specifications

Build a RAG backend server in Java or Python, featuring a REST API that delivers Retrieval-Augmented translation prompts by executing similarity searches using pre-stored translation examples. A **translation prompt request consists** of:

- **Source Language**: ISO 639-1 two-letter code (e.g., "en" for English, "it" for Italian)
- **Target Language**: ISO 639-1 two-letter code
- **Query Sentence**: A plain-text string

RAG Database

Integrate a database of your choice capable of storing and retrieving translation pairs. Some valid examples of libraries or frameworks include Vector DBs, Lucene/ElasticSearch, and MySQL with text search functionality.

The quality of retrieval will be evaluated, but the primary focus is on the overall functionality and usability of the tool.

You have the discretion to choose the similarity metric and algorithm, as well as how to apply it for database queries. Retrieve up to the **4 translation pairs** that are **most similar** based on the selected similarity score, given the translation request.

A translation pair is considered useful if it meets the following criteria:

- It contains sentences in both the source and target languages.
- The source language sentence is similar to the query sentence.

–
TRANSLATED S.R.L.
VAT. IT 071 73 52 10 01
R.E.A. Roma 1015467

API Specification

Your backend server will provide two main REST API endpoints for interacting with the translation service.

`POST /pairs` *- add a new translation pair to your database*.

- **INPUT**: source language, target language, sentence, translation.
- **OUTPUT**: a simple “ok” response.

`GET /prompt` *- returns a translation prompt for a given sentence.*

- **INPUT**: source language, target language, input sentence.
- **OUTPUT**: the translation prompt to use for LLM translation including the suggestions returned by the database.

Stammering Detection API **(ADVANCED)**

Implement a **stammering detection algorithm** to analyze a translated sentence.

In Machine Translation, stammering refers to the non-natural repetition of text parts in the translated output, resulting in awkward or nonsensical sentences. Stammering examples will be provided in a separate file. Implement the following API endpoint:

`GET /stammering`

- **INPUT**: an input sentence and its translation.
- **OUTPUT**: a boolean value indicating whether a stammering has been detected.

–
TRANSLATED S.R.L.
VAT. IT 071 73 52 10 01
R.E.A. Roma 1015467

Restrictions

ALLOWED:	Standard libraries and language features. Libraries for basic HTTP server functionality and JSON handling. Any NLP and information retrieval library. Any AI-powered code assistance tool.
NOT ALLOWED:	External stammering detection systems.

Success Criteria

MINIMUM SUCCESS RESULT:	Successfully stores translation pairs. The API is operational and able to receive and correctly respond to addition and translation requests as specified.
IDEAL SUCCESS RESULT:	The application is fully operational, incorporating all required and advanced features. Successfully conducts similarity searches, and retrieves similar pairs. Provides setup and run instructions in the README. Code is well-organized, maintainable, and adheres to best practices.
SCORE BOOSTERS:	Containerize the application using Docker.

Considerations

Balance **complexity** and **simplicity:**

- Avoid overly complex solutions that are hard to maintain.
- Ensure the solution is not overly simplistic to the point of lacking essential features.

Every developer develops their own approach to coding over time to avoid falling into either extreme. The purpose of this test is to see **how aligned your approach is with ours**.

–
TRANSLATED S.R.L.
VAT. IT 071 73 52 10 01
R.E.A. Roma 1015467

Repository

- Host the code on a **GIT repository** on any platform of your choice.
- Ensure that the evaluation team has access.
- Include all necessary files and documentation.
- A **set of Python scripts is provided** for database population, prompt-generation requests, and stammering detection tests.

Disclaimer

The code you produce is and will remain your property. We will not use the artifacts you produce in this test.

Contacts

Davide Caroselli: davide.caroselli@translated.com (github user: davidecaroselli)
Valerio Giannini: valerio.giannini@translated.com (github user: ValeGian)

Good luck with your test! We look forward to reviewing your work.