User Manual

Nonsense Generator Project

Lorenzo Gatto, Jacopo Di Lauro, Riccardo De Nat, Asia Chiarion Sileni

09/06/2025

Abstract

This manual provides detailed information on the installation, configuration, and usage of the Nonsense Generator program.

1 Introduction

The **Nonsense Generator** is a Java-based web application designed to produce grammatically correct but logically absurd sentences. It analyzes user-provided text to extract relevant words (nouns, verbs, and adjectives). These words, or randomly selected words from its dictionaries, are then used to fill predefined sentence templates.

The application leverages the Google Cloud Language API for advanced linguistic analysis, including syntax analysis to identify parts of speech from input sentences, and content moderation to check generated sentences for potentially sensitive content.

1.1 Principal functions

The principal functions of the program are as follows:

- **Sentence Analysis**: Users can input a sentence, which the application will analyze to extract nouns, verbs, and adjectives using the Google Cloud Language API.
- **Nonsense Sentence Generation**: Based on words extracted from the input sentence or random words from internal dictionaries, the application generates new sentences by populating chosen or random templates.
- **Verb Tense Control**: Users can specify the tense (Present, Past, or Future) for the verbs in the generated sentence.
- **Template Selection**: Users can either manually select a sentence template from a predefined list or let the application pick a compatible template randomly.
- **Dynamic Dictionary Management**: The application allows users to add new nouns, adjectives, and verbs to its dictionaries directly through the web interface. These dictionaries are stored as text files.
- **Content Moderation**: Generated sentences are submitted to the Google Cloud Language API for content moderation to identify and flag any potentially sensitive or problematic content if it's over a certain threshold (10%).
- **Web-Based User Interface**: The application provides an HTML interface for easy interaction.

2 Installation and Setup

2.1 Requirements

To set up and run the Nonsense Generator, ensure you have the following installed:

- Java Development Kit (JDK) 17 or higher.
- Maven 3.6+.
- Internet connection (for Google's API access).

- A Google Cloud's key to access the APIs (the instructions to implement it, where and how to save it, are in the next step).
- Recommended IDE: IntelliJ, Eclipse, or VS Code

2.2 Installation Steps

1. **Get the Project Code**: You can obtain the source code in two ways:

   - **Option A: Clone the repository with Git (Recommended)** Open a terminal and run the following commands:

     ```
     git clone https://github.com/jacopodilauro/Nosense_Generator
     cd Nosense_Generator
     ```

     This will download the project into a folder named `Nosense_Generator` and move you inside it.

   - **Option B: Download the ZIP archive from GitHub**
     (a) Visit the repository page on GitHub: https://github.com/jacopodilauro/Nosense_Generator
     (b) Click the green "Code" button, then select "Download ZIP".
     (c) Extract the downloaded ZIP file to a location of your choice on your computer.
     (d) Open a terminal or command prompt and navigate to the main directory of the extracted project (e.g., `cd path/to/folder/Nosense_Generator-main` or similar, depending on how your system extracts archives).

   Ensure your terminal is positioned in the project's root directory (the one containing the `pom.xml` file) before proceeding with the next steps.

2. **Configure Google Cloud Credentials**:

   (a) Obtain your Google Cloud service account key file, you can find it in https://console.cloud.google.com/ it's a `.json` file that you can download from the Google Cloud Console.
   (b) Rename the downloaded key file to `google-key.json`.
   (c) Move the `google-key.json` file into the `src/main/resources/` directory within the project folder. The final path of the file should be similar to: `Nosense_Generator/src/main/resources/google-key.json` (or `Nosense_Generator-main/src/main/resources/google-key.json` if you downloaded the ZIP).

3. **Compile and Run the Project**: In the terminal, ensuring you are still in the project's root directory, run the following command:

   - **For Linux**:

     ```
     export GOOGLE_APPLICATION_CREDENTIALS="$PWD/src/main/resources/google-key.json"
     ```

   - **For Windows (Command prompt - CMD)**:

     ```
     set GOOGLE_APPLICATION_CREDENTIALS=%CD%\src\main\resources\google-key.json
     ```

     %CD% is the path of your current folder.

   - **For Windows (PowerShell)**:

     ```
     $env:GOOGLE_APPLICATION_CREDENTIALS = "$PWD\src\main\resources\google-key.json"
     ```

     $PWD is the path of your current folder.

   To run the project:

   ```
   mvn spring-boot:run
   ```

4. **Access the Application**: Once the application has started successfully (you should see a message similar to `Started LanguagewebApplication in X seconds` in the terminal log), it will be accessible via a web browser at:

   ```
   https://localhost:8080
   ```

3 Execution Environment

- Operating System: Windows or Linux
- Java Version: 17+
- Required API: Google Natural Language
- Input Language: English

4 External APIs Used

- **Syntax Analysis:** https://cloud.google.com/natural-language/docs/analyzing-syntax
- **Toxicity Detection:** https://cloud.google.com/natural-language/docs/moderate-text

5 Project Structure

The project is organized as follows:

- `src/main/java/com/nonsense/languageweb/`: Contains all the Java source code.
  - `NonsenseApplication.java`: The main controller handling web requests, UI generation, and core application logic.
  - `LanguagewebApplication.java`: The Spring Boot main application class.
  - `SyntaxAnalyzer.java`: Class responsible for analyzing sentence syntax using Google Cloud API.
  - `TemplateFiller.java`: Class for filling sentence templates with words.
  - `TemplateSelector.java`: Class for selecting sentence templates.
  - `Noun.java, Verb.java, Adjective.java`: Classes to load and provide random nouns, verbs, and adjectives from text files.
- `/resources/`: Contains application resources.
  - `sentence_templates.txt`: File storing sentence templates. Each template includes word count requirements (e.g., nouns:1, verbs:1, adjectives:1).
  - `past.txt`: File storing a list of verbs and their past tense forms.
  - `nouns.txt, adjectives.txt, verbs.txt`: Dictionary files for nouns, adjectives and verbs.
- `output.txt` : contains the random generated sentences.
- Test Result.html: contains the result of the maven test build by the IDE IntelliJ.
- `src/test/java/com/nonsense/langugeweb`: Contains unit tests for the application.
- `Documents/`: Designed for detailed documentation, User Manual, User Stories, System Test Report.
- `pom.xml`: The Maven Project Object Model file, managing dependencies and the build process.

6 Basic Usage Instructions

1. **Access the application**: Open your web browser and navigate to the application's URL (`https://localhost:8080` if running locally).
2. **Enter a Sentence (Optional)**: In the "Sentence to analyze" field, you can type a sentence. The application will extract nouns, verbs, and adjectives from this sentence to use in the nonsense generation. If left blank, the application will use random words from its dictionaries.
3. **Select Verb Tense**: Choose the desired tense for verbs (Present, Past, or Future) from the dropdown menu.
4. **Choose a Template (Optional)**: You can select a specific sentence structure from the "Choose a Template" dropdown. If you don't select one, the application will pick a compatible template randomly based on the available words (either from your input sentence or entirely random if no sentence was provided).
5. **Generate Nonsense**: Click the "Analyze and Generate Nonsense" button otherwise press Enter.
6. **View Results**: The page will display:
   - **Analysis of Input Sentence**: If you provided a sentence, this section will show the extracted nouns, verbs, and adjectives.
   - **Selected Template**: Shows the template structure used for generation (either manually chosen or randomly selected).
   - **Generated Nonsense Sentence**: The final, generated sentence.
   - **Content Moderation Status**: Indicates if any potentially sensitive content was detected in the generated sentence, along with categories and confidence scores if issues are found.
7. **Manage Dictionaries**:
   - Click the "Manage Dictionaries" button to reveal options for adding words.
   - Enter a new noun, adjective, or verb into the respective input field and click the corresponding "Add" button.
   - The word will be added to the appropriate dictionary file if it's valid (letters and spaces only) and not a duplicate.