

Exercise 12 — AI Translator with Terminology Glossary

A3 | fx Download the syllabus and follow

	A	B	C
1	Source (EN)	Target	
2	Our AI tutor inte	Notre tuteur IA s'intègre directe	
3	Download the sy	Téléchargez le plan de cours et su	
4	The marketing campaign launches next Monday at		
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

 **AI Translator (Gemini)**

Target language (e.g., French, es, de)

Text to translate

The marketing campaign launches next Monday at 9 AM.

Test Connection

Translate

La campagne marketing se lance lundi prochain à 9h.

Bring the power of Gemini into Google Sheets to translate text — and keep your organization's terminology consistent.

Most translation tools ignore your company's unique language: acronyms, product names, or preferred wordings. In this project, you'll build an **AI Translator** directly inside Google Sheets that not only translates text but also follows a **custom glossary** of approved terms.

What You'll Build

- A **custom menu** in Sheets → 🧠 AI Tools
- Automatic translation of selected cells → next column
- A **Glossary sheet** defining preferred terms
- A **Sidebar translator** for one-off translations
- A **custom formula**: `=AI_TRANSLATE(A2, "French", Glossary!A:B)`

English Text

French Translation

Our AI tutor integrates with the LMS. Notre tuteur IA s'intègre au SGA.

Step 1 — Set Up the Project

1. Open a new Google Sheet.
 2. Choose **Extensions** → **Apps Script**.
 3. Name the project: **AI Translator (Gemini)**
-

Step 2 — Add the Code

Copy this full script into **Code.gs** (see below).

It defines the menu, Gemini API calls, glossary logic, and the sidebar function.

👉 Full Code.gs from Exercise 12 (see your GitHub project)

Step 3 — Add Your Gemini API Key

In the Apps Script editor:

- Go to **Project Settings** → **Script Properties** → **Add property**

- **Name:** GEMINI_API_KEY
- **Value:** (your Gemini API key)

Save it.



Step 4 — Create Your Glossary Sheet

Add a sheet named **Glossary** with two columns:

Source Term	Preferred Translation
AI tutor	tuteur IA
LMS	SGA
syllabus	plan de cours
marketing campaign	campagne marketing

Gemini will be instructed to **respect** these mappings exactly.



Step 5 — Use the Translator

1. In your Sheet, reload and you'll see a new menu: 🧠 **AI Tools**
2. Choose **Insert Sample Text** and **Insert Sample Glossary**.
3. Select the English column.
4. Click **Translate Selection** → **next column**.
5. Watch as Gemini fills the next column with translations that obey your glossary.

You can also use the Sidebar (**AI Tools** → **Open Translator Sidebar**) for quick one-off translations.

How It Works

1 Prompt Builder

Each translation request is built like this:

```
Translate the text to French.  
Output ONLY the translated text.  
Use these exact glossary terms:  
- "AI tutor" → "tuteur IA"  
- "LMS" → "SGA"  
Text:  
Our AI tutor integrates with the LMS template.
```

2 Gemini Call

`UrlFetchApp` sends the prompt to:

```
https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-fla  
sh:generateContent
```

and parses the JSON response for the translation text.

3 Retry Logic

If Gemini returns nothing, the script retries automatically with a simplified prompt ("Return ONLY the translation").

4 Glossary Support

The glossary sheet is read as pairs (A:B) and inserted into the prompt. Gemini treats them as strict mappings.

Testing and Customization

Try different target languages:

```
=AI_TRANSLATE(A2, "Spanish", Glossary!A:B)
```

Increase `DEFAULT_TOKENS` to 1024 for longer paragraphs.
Or add a temperature slider in the sidebar for creative translations.



Why This Matters

- Ensures terminology consistency across teams.
 - Empowers non-developers to use Gemini within Sheets.
 - Demonstrates how prompt engineering and Apps Script combine to solve real workflow problems.
-



Learning Outcomes

After completing Exercise 12 you can:

- Call Gemini via Apps Script REST API.
- Build custom menus and sidebars in Sheets.
- Pass structured data (like glossaries) into AI prompts.
- Handle API errors and parse responses safely.

appscript.json

```
{
  "timeZone": "America/Toronto",
  "runtimeVersion": "V8",
  "exceptionLogging": "STACKDRIVER",
  "oauthScopes": [
    "https://www.googleapis.com/auth/spreadsheets",
    "https://www.googleapis.com/auth/script.external_request",
    "https://www.googleapis.com/auth/script.container.ui"
  ]
}
```

Sidebar.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      body { font-family: system-ui, -apple-system, Segoe UI, Roboto, Arial,
sans-serif; padding:16px; }
      textarea, input { width:100%; }
      textarea { min-height:140px; }
      button { background:#1a73e8; color:#fff; border:0; padding:8px 12px;
border-radius:6px; cursor:pointer; }
      pre { background:#f1f3f4; padding:10px; border-radius:6px; white-space:pre-wrap;
}
      label { display:block; margin-top:10px; font-weight:600; }
    </style>
  </head>
  <body>
    <h2>🤖 AI Translator (Gemini)</h2>

    <label>Target language (e.g., French, es, de)</label>
    <input id="lang" placeholder="French" value="French"/>

    <label>Text to translate</label>
    <textarea id="t" placeholder="Paste text here..."></textarea>

    <p>
      <button onclick="doPing()">Test Connection</button>
      <button onclick="doTranslate()">Translate</button>
    </p>

    <pre id="out">Waiting...</pre>

    <script>
      function set(msg) { document.getElementById('out').textContent = msg; }

      function doPing() {
        set('⌚ Pinging server...');
        google.script.run
          .withSuccessHandler(r => set('✅ ping: ' + r))
          .withFailureHandler(e => set('❌ ping failed: ' + e.message))
          .ping();
      }
    </script>
  </body>
</html>
```

```

    }

    function doTranslate() {
      const lang = document.getElementById('lang').value || 'French';
      const t = document.getElementById('t').value || '';
      if(!t.trim()){ set('⚠ Please paste some text.');
```

Code.gs

```

/**
 * Exercise 12 — AI Translator with Terminology Glossary (Sheets + Gemini)
 * Author: Laurence Svekis
 *
 * Features
 * - Translate text via Gemini REST API (UrlFetchApp)
 * - Enforce custom terminology from a “Glossary” sheet (A:B)
 * - Batch: Selection → next column
 * - Custom function: =AI_TRANSLATE(A2, "French", Glossary!A:B)
 * - Sidebar: quick single-shot translator (+ ping test)
 */

// ===== CONFIG =====
// Prefer Script Properties (Project Settings → Script properties: GEMINI_API_KEY)
// You can set a fallback value here if you want:
const GEMINI_API_KEY = ""; // e.g., 'Alza...'
const GEMINI_MODEL   = 'gemini-2.5-flash';
const GEMINI_BASE    = 'https://generativelanguage.googleapis.com/v1beta';

const DEFAULT_TEMP   = 0.2;
const DEFAULT_TOKENS = 1024; // raised from 512 for long strings
const COOLDOWN_MS    = 120;

// ===== MENU / UI =====

```

```

function onOpen() {
  SpreadsheetApp.getUi()
    .createMenu('🧠 AI Tools')
    .addItem('Translate Selection → next column', 'translateSelectionToRight')
    .addItem('Open Translator Sidebar', 'showTranslatorSidebar')
    .addSeparator()
    .addItem('Insert Sample Text', 'insertSampleText')
    .addItem('Insert Sample Glossary', 'insertSampleGlossary')
    .addToUi();
}

// Sidebar loader (HTML file must be named "Sidebar")
function showTranslatorSidebar() {
  try {
    const html = HtmlService.createHtmlOutputFromFile('Sidebar')
      .setTitle('AI Translator (Gemini)');
    SpreadsheetApp.getUi().showSidebar(html);
  } catch (e) {
    SpreadsheetApp.getUi().alert('Failed to open sidebar: ' + e.message);
  }
}

// ===== DIAGNOSTICS =====
// Ping from Sidebar to prove wiring/scopes are OK
function ping() { return 'pong'; }

// Simple model listing to verify API key
function testGemini() {
  const key = PropertiesService.getScriptProperties().getProperty('GEMINI_API_KEY') ||
GEMINI_API_KEY;
  if (!key) return Logger.log('❌ No GEMINI_API_KEY set');
  const url =
`https://generativelanguage.googleapis.com/v1beta/models?key=${encodeURIComponent(key)}
`;
  const res = UrlFetchApp.fetch(url, { muteHttpExceptions: true });
  Logger.log(res.getContentText());
}

// ===== API KEY =====
function getApiKey() {
  const p = PropertiesService.getScriptProperties().getProperty('GEMINI_API_KEY');
  return (p && p.trim()) ? p.trim() : (GEMINI_API_KEY || "").trim();
}

```



```
// ===== ROBUST EXTRACTOR =====
// Handles candidates with empty parts/text, and MAX_TOKENS truncation.
function extractGeminiText_(json) {
  try {
    if (json?.error?.message) return `❌ API error: ${json.error.message}`;

    const cand = json?.candidates?.[0];
    if (!cand) return "";

    const parts = cand?.content?.parts || [];
    const joined = parts.map(p => (p?.text || "")).join("").trim();
    if (joined) return joined;

    if (cand?.text && String(cand.text).trim()) return String(cand.text).trim();

    if (cand?.finishReason === 'MAX_TOKENS') {
      return '⚠️ Truncated (MAX_TOKENS) and no visible text. Try lowering input length or
increasing max tokens.';
    }
    return "";
  } catch (e) {
    return "";
  }
}

// ===== GEMINI CALL (with logs + extractor) =====
function callGemini_(prompt) {
  const key = getApiKey();
  if (!key) return `❌ Missing GEMINI_API_KEY.`;

  const payload = {
    contents: [{ role: "user", parts: [{ text: prompt }] }],
    generationConfig: { temperature: DEFAULT_TEMP, maxOutputTokens: DEFAULT_TOKENS,
topP: 0.9, topK: 40 }
    // If you want to try plain text with v1beta, add at top level (NOT inside generationConfig):
    // responseMimeType: "text/plain"
  };

  const url =
`${GEMINI_BASE}/models/${encodeURIComponent(GEMINI_MODEL)}/generateContent?key=
${encodeURIComponent(key)}`;
  const res = UrlFetchApp.fetch(url, {
    method: 'post',
    contentType: 'application/json',

```

```

    payload: JSON.stringify(payload),
    muteHttpExceptions: true
  });

const code = res.getResponseCode();
const body = res.getContentText();

if (code < 200 || code >= 300) {
  Logger.log(` Gemini HTTP ${code}\n${body}`);
  return `❌ HTTP ${code}: ${body}`;
}

let out = "";
try {
  const j = JSON.parse(body);
  out = extractGeminiText_(j);
} catch (e) {
  Logger.log('Parse error: ' + e);
  return `❌ Parse error`;
}

if (!out) {
  Logger.log('Empty translation body: ' + body);
  return `⚠️ Empty translation`;
}
return out.trim();
}

// ===== PROMPT BUILDER (strict) =====
function buildTranslatePrompt_(text, targetLang, glossaryPairs) {
  const rules = [
    `Translate the text to ${targetLang}.`,
    `Output ONLY the translated text. No quotes, no brackets, no preface, no code fences.`,
    `Preserve meaning and tone. Respect proper nouns and capitalization.`
  ];

  if (Array.isArray(glossaryPairs) && glossaryPairs.length) {
    const entries = glossaryPairs
      .filter(r => r && r.length >= 2 && String(r[0]).trim())
      .map(r => `-${String(r[0]).trim()} → "${String(r[1]||"").trim()}"`)
      .join("\n");
    if (entries) {
      rules.push(
        `Use these exact terminology mappings wherever applicable:`,

```

```

        entries
    );
}
}

return `${rules.join("\n")}\n\nText:\n${String(text || "").trim()}`;
}

// ===== PUBLIC TRANSLATE (with retry) =====
function translateWithGlossary_(text, lang, glossary) {
    // 1) Strict "just the translation"
    const prompt1 = buildTranslatePrompt_(text, lang, glossary);
    let out = callGemini_(prompt1);
    if (out && !/^!|^X/.test(out)) return out;

    // 2) Retry once with a simpler instruction
    const prompt2 = [
        `Translate to ${lang}.`,
        `Return ONLY the translated sentence. No quotes, no commentary, no extra lines.`,
        `Text:\n${String(text || "").trim()}`
    ].join("\n");
    out = callGemini_(prompt2);
    return out;
}

// ===== SHEETS ACTIONS =====
function translateSelectionToRight() {
    const ui = SpreadsheetApp.getUi();
    const resp = ui.prompt('Translate Selection', 'Enter target language (e.g., French, es, de):',
        ui.ButtonSet.OK_CANCEL);
    if (resp.getSelectedButton() !== ui.Button.OK) return;

    const lang = (resp.getResponseText() || "").trim();
    if (!lang) { ui.alert('Please provide a language.');
```

```

const out = [];
for (let r = 0; r < values.length; r++) {
  const src = String(values[r][0] || "").trim();
  if (!src) { out.push(""); continue; }
  const translated = translateWithGlossary_(src, lang, glossary);
  out.push([translated]);
  Utilities.sleep(COOLDOWN_MS);
}

const outCol = range.getColumn() + range.getNumColumns();
sheet.getRange(range.getRow(), outCol, out.length, 1).setValues(out);
SpreadsheetApp.getActive().toast(`Translated ${out.length} row(s) → column ${outCol}`);
}

// ===== CUSTOM FUNCTION =====
// Usage: =AI_TRANSLATE(A2, "French", Glossary!A:B)
function AI_TRANSLATE(text, targetLanguage, glossaryRange) {
  const glossVals = Array.isArray(glossaryRange) ? glossaryRange : [];
  return translateWithGlossary_(text, targetLanguage || 'French', glossVals);
}

// ===== SAMPLE DATA =====
function insertSampleText() {
  const data = [
    ['Source (EN)', 'Target (→ after run)'],
    ['Please reset your password using the link in your email.', ""],
    ['Our AI tutor integrates directly with the LMS template.', ""],
    ['Download the syllabus and follow the weekly schedule closely.', ""],
    ['The marketing campaign launches next Monday at 9 AM.', ""],
    ['Customer data is stored securely according to our policy.', ""],
  ];
  const sheet = SpreadsheetApp.getActiveSheet();
  sheet.clearContents();
  sheet.getRange(1, 1, data.length, data[0].length).setValues(data);
  SpreadsheetApp.getActive().toast("✅ Sample text inserted (A–B).");
}

function insertSampleGlossary() {
  let sh = SpreadsheetApp.getActive().getSheetByName('Glossary');
  if (!sh) sh = SpreadsheetApp.getActive().insertSheet('Glossary');
  sh.clearContents();
  sh.getRange(1, 1, 1, 2).setValues([['Source Term', 'Preferred Translation']]);
  const rows = [
    ['learning management system', 'système de gestion de l'apprentissage'],
  ];

```

```

    ['LMS', 'SGA'],
    ['AI tutor', 'tuteur IA'],
    ['syllabus', 'plan de cours'],
    ['privacy policy', 'politique de confidentialité'],
    ['marketing campaign', 'campagne marketing']
  ];
  sh.getRange(2, 1, rows.length, 2).setValues(rows);
  SpreadsheetApp.getActive().toast('✅ Sample Glossary inserted (Glossary!A:B).');
}

// ===== SIDEBAR ENTRYPOINT =====
function translateFromSidebar(text, targetLang) {
  try {
    const glossarySheet = SpreadsheetApp.getActive().getSheetByName('Glossary');
    const glossaryValues = glossarySheet
      ? glossarySheet.getRange(1, 1, glossarySheet.getLastRow(), 2).getValues()
      : [];
    const translation = translateWithGlossary_(text || '', targetLang || 'French', glossaryValues);
    Logger.log('Sidebar -> translate ok');
    return translation || '⚠ Empty translation';
  } catch (e) {
    const msg = `Sidebar error: ${e && e.message ? e.message : e}`;
    Logger.log(msg);
    return '❌ ' + msg;
  }
}

// ===== OPTIONAL: quick local test =====
function manualTest() {
  const out = translateWithGlossary_(
    'Our AI tutor integrates directly with the LMS template.',
    'French',
    [['AI tutor', 'tuteur IA'], ['LMS', 'SGA']]
  );
  Logger.log(out);
}

```