

Build an AI Text Rewriter in Google Sheets Using Gemini and Apps Script

Artificial intelligence is no longer limited to chat interfaces — you can now embed powerful AI tools directly into your favourite Google Workspace apps. In this tutorial, we'll walk through how to build an **AI Text Rewriter inside Google Sheets** using **Google Apps Script** and the **Gemini REST API**.

By the end, you'll have a fully functional project that lets you rewrite sentences, paragraphs, and full articles directly inside a spreadsheet — all powered by Google's Gemini model.

What You'll Build

Your final project will include:

- A **custom menu** called "🧠 AI Tools"
- A **sidebar interface** to rewrite text in different tones (e.g., professional, friendly, simplified)
- A **custom function** =AI_REWRITE() that works inside any cell
- Integration with the **Gemini REST API** using Apps Script's `UrlFetchApp`

Here's a sneak peek of what it looks like:

A spreadsheet where column A contains original sentences and column B auto-generates rewritten versions via Gemini.

Step 1: Set Up the Project

1. Open a new Google Sheet.
2. Go to **Extensions → Apps Script**.
3. Delete any existing code.
4. Create three files:
 - **Code.gs** — main backend logic.
 - **Sidebar.html** — user interface.
 - **appsscript.json** — project manifest (we'll enable it later).

Step 2: Write the Backend (Code.gs)

Paste the following code in `Code.gs`:

```
/**  
 * AI Text Rewriter for Google Sheets using Gemini REST API
```

```

* Author: Laurence Svekis
*/

const GEMINI_API_KEY = 'YOUR_API_KEY_HERE';
const GEMINI_MODEL = 'gemini-2.5-flash';
const BASE_URL = 'https://generativelanguage.googleapis.com/v1beta';

// Menu setup
function onOpen() {
  SpreadsheetApp.getUi()
    .createMenu('🧠 AI Tools')
    .addItem('Open Rewriter Sidebar', 'showRewriterSidebar')
    .addItem('Insert Sample Data', 'insertSampleData')
    .addToUi();
}

// Sidebar launcher
function showRewriterSidebar() {
  const html = HtmlService.createHtmlOutputFromFile('Sidebar')
    .setTitle('Gemini Rewriter');
  SpreadsheetApp.getUi().showSidebar(html);
}

// Gemini REST call
function callGeminiRewrite(text, style = 'professional tone',
temperature = 0.6) {
  const url =
`${BASE_URL}/models/${GEMINI_MODEL}:generateContent?key=${GEMINI_API_K
EY}`;
  const payload = {
    contents: [
      { role: "user", parts: [{ text: `Rewrite the following text in a
${style}: \n${text}` } ] }
    ],
    generationConfig: {
      temperature,
      maxOutputTokens: 512
    }
  };

  const options = {
    method: 'post',
    contentType: 'application/json',
    payload: JSON.stringify(payload),
    muteHttpExceptions: true
  };

  try {

```

```

    const res = UrlFetchApp.fetch(url, options);
    const data = JSON.parse(res.getContentText());
    return data?.candidates?.[0]?.content?.parts?.[0]?.text || ' ⚠️ No
rewrite returned.';
  } catch (e) {
    return ' ❌ Error: ' + e.message;
  }
}

// Custom formula: =AI_REWRITE(A2, "friendly tone")
function AI_REWRITE(text, style) {
  return callGeminiRewrite(text, style || 'neutral tone');
}

// Inserts sample rows
function insertSampleData() {
  const data = [
    ['Original Text', 'Rewritten Text'],
    ['Our company values innovation and collaboration across all
departments.', ''],
    ['This policy ensures all employees have equal access to training
opportunities.', ''],
    ['Please submit your report by Friday at noon.', '']
  ];
  const sheet =
SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  sheet.clearContents();
  sheet.getRange(1, 1, data.length, data[0].length).setValues(data);
  SpreadsheetApp.getActive().toast(' ✅ Sample data inserted');
}

```

Step 3: Create the Sidebar (Sidebar.html)

This interface lets users rewrite text manually without formulas.

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      body { font-family: system-ui; padding: 16px; }
      textarea { width: 100%; height: 100px; }
      button { background: #1a73e8; color: white; border: 0; padding:
8px 12px; border-radius: 6px; cursor: pointer; }
      select, input { width: 100%; margin: 4px 0; padding: 4px; }
      pre { background: #f1f3f4; padding: 8px; border-radius: 6px;
white-space: pre-wrap; }
    </style>

```

```

</head>
<body>
  <h2>🧠 Gemini Rewriter</h2>
  <textarea id="text" placeholder="Enter text to
rewrite..."></textarea>
  <label>Choose Tone:</label>
  <select id="tone">
    <option>professional tone</option>
    <option>friendly tone</option>
    <option>simplified explanation</option>
    <option>marketing tone</option>
    <option>academic tone</option>
  </select>
  <label>Temperature (creativity):</label>
  <input type="number" id="temp" min="0" max="1" step="0.1"
value="0.6">
  <button onclick="rewrite()">Rewrite</button>
  <h3>Result:</h3>
  <pre id="output"></pre>

  <script>
    function rewrite() {
      const text = document.getElementById('text').value;
      const tone = document.getElementById('tone').value;
      const temp =
parseFloat(document.getElementById('temp').value);
      document.getElementById('output').textContent = ' ⌚
Rewriting...';
      google.script.run
        .withSuccessHandler(r =>
document.getElementById('output').textContent = r)
        .callGeminiRewrite(text, tone, temp);
    }
  </script>
</body>
</html>

```



Step 4: Add the Manifest (appsscript.json)

```

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

```

```
]
}
```

This ensures your script can:

- Modify Sheets data
- Show sidebars and dialogs
- Make external HTTP requests to Gemini

Step 5: Run and Authorize

1. In Apps Script, click **Run** → **onOpen()**.
2. Approve the authorization scopes.
3. Reload your spreadsheet.
4. You'll see a new menu:  **AI Tools**.

From there:

- **Insert Sample Data** to test
- **Open Rewriter Sidebar** to experiment with custom tones
- Try `=AI_REWRITE(A2, "friendly tone")` in any cell

How It Works

1. `UrlFetchApp.fetch()`

This function sends an HTTP POST request to Gemini's REST endpoint:

POST

`https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-flash:generateContent?key=API_KEY`

The payload defines:

- **contents** — the user's input message.
- **generationConfig** — creativity (temperature) and token length.

2. Parsing the Response

Gemini returns a JSON object.

The relevant text is usually found at:

`data.candidates[0].content.parts[0].text`

If Gemini detects unsafe or truncated content, the function returns helpful messages instead

of errors.

3. Reusable Functions

You can:

- Wrap Gemini calls into your own custom formulas (AI_SUMMARY, AI_REWRITE, AI_TRANSLATE).
- Use them in automation, workflows, or classroom exercises.

Learning Outcomes

After completing this exercise, you now understand:

- How to connect Google Sheets to Gemini's REST API
- How to send structured prompts using Apps Script
- How to create UI extensions (menus + sidebars)
- How to handle and display AI output in Sheets

This pattern forms the foundation for:

- Summarizers
- Grammar Correctors
- Tone Converters
- AI-Powered Report Generators

