AIDE User Guide

Introduction

AIDE (AI-Assisted Data Extraction) is a web-based tool designed to accelerate the data extraction process for systematic reviews and meta-analysis. It uses Large Language Models (LLMs) to automatically extract information from PDF documents based on your custom prompts.

Getting Started

Prerequisites

- A modern web browser (Chrome, Firefox, Safari, or Edge)
- An API key for an OpenAI-compatible LLM service, OR
- A local LLM server (llama.cpp or LMStudio)

Supported LLM Providers

AIDE works with any OpenAI-compatible API endpoint:

1. **OpenAI (GPT-4, GPT-3.5)**
   - Endpoint: `https://api.openai.com/v1/chat/completions`
   - Get API key: https://platform.openai.com/api-keys
   - Pricing: Pay-per-use

2. **Anthropic Claude (via OpenRouter)**
   - Endpoint: `https://openrouter.ai/api/v1/chat/completions`
   - Get API key: https://openrouter.ai/keys
   - Pricing: Varies by model

3. **Google Gemini (via OpenRouter)**
   - Endpoint: `https://openrouter.ai/api/v1/chat/completions`
   - Get API key: https://openrouter.ai/keys
   - Pricing: Varies by model

4. **Mistral AI**
   - Endpoint: `https://api.mistral.ai/v1/chat/completions`
   - Get API key: https://console.mistral.ai/api-keys/
   - Pricing: Free tier available

5. **Local Models (llama.cpp)**
   - Endpoint: `http://localhost:8080/v1/chat/completions`
   - Setup: https://github.com/ggerganov/llama.cpp
   - Pricing: Free (requires local hardware)

6. **Local Models (LMStudio)**
   - Endpoint: `http://localhost:1234/v1/chat/completions`
   - Setup: https://lmstudio.ai/
   - Pricing: Free (requires local hardware)

Step-by-Step Workflow

Step 1: Configure API Settings

1. Navigate to the **Setup** page
2. Enter your API endpoint URL
3. Enter your API key
4. (Optional) Set context window size
5. Click **Save Settings**

Important Notes:
- Your API key is stored only in sessionStorage
- It will be automatically cleared when you close the browser
- Never share your API key with anyone

Step 2: Create and Upload Your Coding Form

Creating a Coding Form

Your coding form is an Excel or CSV file where:
- **First row** = Your prompts for the LLM
- **Subsequent rows** = Extracted data from your PDFs

Example coding form:

Study Authors	Publication Year	Sample Size	Intervention Type	Effect Size
empty	empty	empty	empty	empty

Tips for Writing Good Prompts:

✅ **Good Prompts:**
- "What is the total sample size of the study?"
- "List all authors of this study, separated by commas"
- "What intervention was tested in this study?"
- "Extract the primary outcome measure reported"

❌ **Poor Prompts:**
- "Authors" (too vague)
- "Number" (ambiguous)
- "Results" (too broad)

Uploading Your Coding Form

1. Go to the **Setup** page
2. Click **Choose File** under "Coding Form"
3. Select your .csv, .xls, or .xlsx file
4. The app will display your prompts for confirmation

Step 3: Analyze PDFs

1. Navigate to the **Analyze** page
2. Click **Choose PDF** and select your document

3. Choose processing mode:
   - **Send PDF file**: Sends the complete PDF (includes images, formatting)
   - **Send text only**: Extracts and sends text only (faster, no images)
4. Click **Analyze PDF**

What Happens During Analysis:

1. The app processes your PDF
2. Sends ONE API request with ALL your prompts
3. The LLM analyzes the document and returns structured JSON
4. Responses appear in the coding form fields

Step 4: Review and Record Responses

For each prompt:

1. **Review the response** - Is it accurate?
2. **Check the source** - Click “Source” to see where the information came from
3. **Edit if needed** - You can modify the response before recording
4. **Click “Record”** - Saves the response to your coding form

Source Information:

- Shows the exact text from the PDF that the LLM used
- Displays the page number where the information was found
- Helps you verify the accuracy of the extraction

Step 5: Process Multiple PDFs

1. After recording responses, click **Next PDF**
2. Upload a new PDF
3. Click **Analyze PDF**
4. Review and record responses
5. Repeat for all your documents

Step 6: Download Your Results

1. Go back to the **Setup** page
2. Click **Download Form** to get your completed Excel file
3. Or click **View Form** to see your data in a new tab

Advanced Features

PDF Processing Modes

Send PDF File:
- Sends the complete PDF to the LLM
- Includes images, tables, formatting
- Best for: Documents with important visual elements
- Slower and may use more tokens

Send Text Only:
- Extracts text using pdf.js
- Sends only the text content

- Best for: Text-heavy documents without crucial images
- Faster and uses fewer tokens

Context Window Size

- Some LLMs have limited context windows
- If your PDF is very long, you may need to adjust this
- Leave blank to use the model’s default
- Larger values = can process longer documents

Structured JSON Output

AIDE uses structured JSON for reliable data extraction:

```json
{
  "responses": [
    {
      "prompt": "What is the sample size?",
      "response": "N = 150",
      "source": "A total of 150 participants were recruited...",
      "page": "3"
    }
  ]
}
```

This ensures:
- Consistent formatting
- Reliable parsing
- Source attribution
- Page number tracking

Best Practices

1. Prompt Design

- **Be specific**: “Extract the primary outcome measure” vs “Outcome”
- **Use examples**: “List authors as: LastName, FirstName; LastName, FirstName”
- **Request format**: “Report sample size as: N = X”
- **Be explicit**: “If not found, respond with ‘Not reported’“

2. Batch Processing

- Process similar documents together
- Use the same coding form for consistency
- Download your form regularly (after every 10-20 PDFs)

3. Quality Control

- Always review responses before recording
- Use the source information to verify accuracy
- Manually correct any errors
- Keep notes on problematic extractions

4. Performance Optimization

- Use “text-only” mode for text-heavy documents

- Process shorter PDFs first to test your prompts
- Adjust context window if you get errors
- Consider using faster models for initial testing

Troubleshooting

Problem: “Please configure API settings”

Solution:
1. Go to Setup page
2. Enter API endpoint and key
3. Click Save Settings

Problem: “Invalid response format from LLM”

Solutions:
1. Check your API key is correct
2. Verify the endpoint URL is accurate
3. Try a different model
4. Check if you have API credits/quota

Problem: “Failed to extract text from PDF”

Solutions:
1. Ensure the PDF is not corrupted
2. Try re-saving the PDF
3. Check if the PDF has selectable text
4. Use “send-pdf” mode instead of “text-only”

Problem: Responses are inaccurate

Solutions:
1. Improve your prompts (be more specific)
2. Try “send-pdf” mode for better context
3. Use a more capable model (e.g., GPT-4)
4. Break complex prompts into simpler ones

Problem: “API request failed”

Solutions:
1. Check your internet connection
2. Verify API key is valid
3. Check if you have API quota remaining
4. Try again in a few minutes (rate limits)

Problem: Can’t download coding form

Solutions:
1. Check if popups are blocked
2. Try a different browser
3. Make sure you recorded some responses
4. Refresh the page and try again

Data Privacy & Security

What Data is Stored?

In Your Browser (sessionStorage):
- API endpoint URL
- API key (cleared when browser closes)
- Coding form data
- Current PDF text

Not Stored Anywhere:
- Your PDFs are never uploaded to our servers
- All processing happens in your browser
- PDFs are only sent to your chosen LLM API

Security Best Practices

1. **Use HTTPS** - Always access AIDE over HTTPS
2. **Close browser** - Close your browser when done to clear API keys
3. **Download regularly** - Download your coding form frequently
4. **API key safety** - Never share your API key
5. **Local processing** - Consider local LLMs for sensitive documents

Tips for Success

For Systematic Reviews

1. **Create a pilot coding form** - Test with 5-10 PDFs first
2. **Refine prompts** - Adjust based on pilot results
3. **Dual extraction** - Have two reviewers check random samples
4. **Document everything** - Keep notes on prompt changes

For Meta-Analysis

1. **Standardize extraction** - Use consistent prompts
2. **Extract raw data** - Get exact numbers, not summaries
3. **Note units** - Make sure prompts specify units
4. **Effect sizes** - Be specific about which effect size to extract

For Literature Reviews

1. **Key themes** - Extract main themes or concepts
2. **Methodology** - Capture research methods used
3. **Conclusions** - Extract author conclusions
4. **Quality indicators** - Include study quality metrics

Frequently Asked Questions

Q: How much does it cost?

A: AIDE is free to use. You only pay for the LLM API you choose:
- OpenAI: Pay-per-use (typically $0.01-0.10 per document)

- Mistral: Free tier available
- Local models: Free (but requires hardware)

Q: Can I use this offline?

A: Partially. The app works offline, but you need internet to:
- Call cloud-based LLM APIs
- Load the PDF.js worker (cached after first load)

With a local LLM (llama.cpp or LMStudio), you can work completely offline.

Q: How accurate is the extraction?

A: Accuracy depends on:
- Quality of your prompts (most important!)
- LLM model used (GPT-4 > GPT-3.5)
- Document quality (readable PDFs work best)
- Complexity of information requested

Expect 80-95% accuracy with well-designed prompts and good models. Always review and verify responses.

Q: Can I process scanned PDFs?

A: If the PDF has selectable text (OCR already applied), yes. If it's a pure image PDF, you'll need to OCR it first using tools like Adobe Acrobat or online OCR services.

Q: What's the maximum PDF size?

A: Limits depend on:
- Your LLM's context window (typically 8k-128k tokens)
- Browser memory (modern browsers handle most PDFs fine)
- API limits (some providers limit file size)

Most academic papers (10-30 pages) work without issues.

Q: Can I extract tables or figures?

A:
- With "send-pdf" mode: Some models can describe tables/figures
- With "text-only" mode: Only if table text is extractable
- For complex tables: Manual extraction may be more reliable

Q: How do I cite AIDE?

A: See the Cite page in the app, or use:

Schroeder, N. L., Jaldi, C. D., & Zhang, S. (2025). Large Language Models with Human-In-The-Loop Validation for Systematic Review Data Extraction. https://doi.org/10.48550/arXiv.2501.11840

Support and Contributing

Getting Help

1. Check this user guide
2. Review the README.md
3. Check GitHub issues

4. Create a new issue with:
    - What you were trying to do
    - What happened instead
    - Browser and OS information
    - Any error messages

Contributing

Contributions are welcome! See the GitHub repository for:
- Bug reports
- Feature requests
- Code contributions
- Documentation improvements

Acknowledgments

This React version of AIDE is based on the original R Shiny application developed by Noah L. Schroeder, Chris Davis Jaldi, and Shan Zhang. We’re grateful for their groundbreaking work in applying LLMs to systematic review data extraction.