🤖 ChatGPT Universal Analyzer

A powerful, production-ready tool for analyzing and extracting insights from your ChatGPT conversations. Features automatic CAPTCHA handling, multiple export formats, and intelligent content extraction.

Show Image

Show Image

Show Image

✨ Features

🎯 Core Capabilities

- 📊 **Export Mode**: Analyze your downloaded ChatGPT data (JSON format)
- 🌐 **Live Mode**: Browse and analyze conversations in real-time with browser automation
- 🔓 **CAPTCHA Handling**: Automatic detection and solving of CAPTCHAs
- 📈 **Smart Analysis**: Identify code snippets, research content, and external links
- 📁 **Multi-Format Export**: HTML, CSV, JSON, Markdown, and PDF reports
- 💾 **Content Extraction**: Export code files and research documents

🔍 Analysis Features

- **Language Detection**: Automatically identifies programming languages
- **Research Classification**: Detects academic and research conversations
- **Keyword Extraction**: Finds important topics and themes
- **Statistics**: Word counts, message counts, and conversation metrics
- **Folder Organization**: Maintains your ChatGPT folder structure

🛡️ Production Features

- **Error Handling**: Robust error recovery and logging
- **Caching**: Fast re-analysis with intelligent caching
- **Multi-threading**: Parallel processing for large datasets
- **Progress Tracking**: Real-time progress indicators
- **Cross-Platform**: Works on Windows, macOS, and Linux

🚀 Quick Start

Prerequisites

- Python 3.8 or higher
- Chrome browser (for live mode)
- 4GB RAM recommended
- 500MB free disk space

Installation

🪟 Windows

```batch
# Run the automated installer
install_windows.bat

# Or install manually
pip install -r requirements.txt
```

🐧 Linux/macOS

```bash
# Make installer executable
chmod +x install_deps.sh

# Run the installer
./install_deps.sh

# Or install manually
pip install -r requirements.txt
```

📥 Export Mode (Recommended for First Use)

1. **Export your ChatGPT data:**

   - Go to ChatGPT → Settings → Data Controls
   - Click "Export data"
   - Wait for the email with your data
   - Download and extract the ZIP file

2. **Run the analyzer:**

   bash

   ```
   python gptscraper.py --mode export --file conversations.json
   ```

3. **View your results:**
   - Open the generated HTML report in your browser
   - Check the `chatgpt_analysis` folder for all outputs

🌐 Live Mode (Browser Automation)

bash

```
# Basic usage
python gptscraper.py --mode live --email your@email.com

# With password (or set CHATGPT_PASSWORD environment variable)
python gptscraper.py --mode live --email your@email.com --password yourpassword

# Headless mode (runs in background)
python gptscraper.py --mode live --email your@email.com --headless
```

📖 Detailed Usage

Command Line Options

bash

```
python gptscraper.py [OPTIONS]

Options:
  --mode        Analysis mode: 'export' or 'live' (default: export)
  --file        Path to conversations.json for export mode
  --email       ChatGPT email for live mode
  --password    ChatGPT password (or use CHATGPT_PASSWORD env var)
  --output-dir  Output directory (default: chatgpt_analysis)
  --formats     Output formats: html csv json markdown pdf all (default: html)
  --extract     Extract code and research content
  --no-cache    Disable caching for fresh analysis
  --headless    Run browser in headless mode (live mode only)
  --debug       Enable debug logging
  --help        Show help message
```

Examples

Analyze and Generate All Reports

bash

```
python gptscraper.py --mode export --file conversations.json --formats all
```

Extract Code from Specific Conversations

bash

```
python gptscraper.py --mode export --file conversations.json --extract
```

Live Analysis with Specific Output

bash

```
python gptscraper.py --mode live --email user@example.com --formats html csv
```

Using Environment Variables

bash

```bash
# Set credentials
export CHATGPT_EMAIL="your@email.com"
export CHATGPT_PASSWORD="yourpassword"

# Run without specifying credentials
python gptscraper.py --mode live
```

📊 Output Formats

🌐 HTML Report

- Interactive, searchable interface
- Visual statistics and charts
- Filter by conversation type
- Direct links to original conversations
- Mobile-responsive design

📄 CSV Export

- Excel-compatible format
- All metadata and metrics
- Easy sorting and filtering
- UTF-8 with BOM for international characters

📋 Markdown Report

- Readable text format
- Hierarchical organization
- GitHub-compatible
- Easy to share and version control

📘 JSON Data

- Complete structured data
- Machine-readable format
- Preserves all metadata
- Ideal for further processing

📑 PDF Report

- Professional presentation

- Printable format
- Includes charts and statistics
- Executive summary included

🔧 Advanced Features

Content Extraction

The analyzer can extract:

- **Code Files**: Organized by language with proper extensions
- **Research Documents**: Formatted markdown with references
- **Metadata**: JSON files with conversation details

```bash
# Interactive extraction
python gptscraper.py --mode export --file conversations.json --extract

# Extract from live mode
python gptscraper.py --mode live --email user@example.com --extract
```

Custom Filtering

Create custom filters in your code:

```python
from gptscraper import ChatGPTAnalyzer

analyzer = ChatGPTAnalyzer(mode='export', export_file='conversations.json')
conversations = analyzer.extract_conversations_from_export()

# Filter by language
python_convs = [c for c in conversations if 'python' in c.languages]

# Filter by date
recent_convs = [c for c in conversations if '2024' in c.update_time]

# Generate custom report
analyzer.generate_reports(python_convs, formats=['html'])
```

Batch Processing

Process multiple export files:

bash

```bash
#!/bin/bash
for file in exports/*.json; do
  python gptscraper.py --mode export --file "$file" --output-dir "analysis/$(basename $file .json)"
done
```

🛠️ Troubleshooting

Common Issues

Chrome Driver Not Found

bash

```bash
# Install webdriver-manager
pip install webdriver-manager

# Or set environment variable
export CHROMEDRIVER_PATH=/path/to/chromedriver
```

CAPTCHA Issues

- The tool includes automatic CAPTCHA handling
- For persistent issues, try:
  - Running in non-headless mode
  - Using export mode instead
  - Checking if your IP is rate-limited

Login Failures

- Ensure credentials are correct
- Check for 2FA (not currently supported)
- Try logging in manually first
- Use export mode as alternative

Memory Issues with Large Datasets

bash

```bash
# Increase Python memory limit
export PYTHONHASHSEED=0
python gptscraper.py --mode export --file large_conversations.json --no-cache
```

Debug Mode

Enable detailed logging:

```bash
python gptscraper.py --debug --mode export --file conversations.json
```

Debug information is saved to:

- Console output with timestamps
- `chatgpt_analysis/debug.log`
- Screenshots for visual debugging

📁 Project Structure

```
chatgpt-analyzer/
├── gptscraper.py         # Main analyzer script
├── captcha_handler.py    # CAPTCHA detection and handling
├── requirements.txt      # Python dependencies
├── install_deps.sh       # Linux/macOS installer
├── install_windows.bat   # Windows installer
├── README.md             # Documentation
├── venv/                 # Virtual environment (created during install)
├── chatgpt_analysis/     # Default output directory
│   ├── *.html            # Generated reports
│   ├── *.csv             # Data exports
│   └── extracted_content/ # Extracted code and documents
└── .cache/               # Cache directory (optional)
```

🔐 Privacy & Security

- **Local Processing**: All analysis happens on your machine
- **No Data Upload**: Your conversations are never sent anywhere
- **Secure Storage**: Credentials are only used for login, not stored
- **Open Source**: Full code transparency

🤝 Contributing

Contributions are welcome! Please:

1. Fork the repository
2. Create a feature branch
3. Add tests for new functionality

4. Submit a pull request

Development Setup

bash

```
# Clone the repository
git clone https://github.com/yourusername/chatgpt-analyzer.git
cd chatgpt-analyzer

# Create virtual environment
python -m venv venv
source venv/bin/activate  # Linux/macOS
# or
venv\Scripts\activate.bat  # Windows

# Install development dependencies
pip install -r requirements.txt
pip install ipython black flake8 pytest

# Run tests
pytest tests/

# Format code
black *.py
```

📄 License

This project is licensed under the MIT License. See LICENSE file for details.

🙏 Acknowledgments

- OpenAI for ChatGPT
- Selenium WebDriver for browser automation
- The Python community for excellent libraries

📞 Support

- **Issues**: GitHub Issues page
- **Discussions**: GitHub Discussions
- **Email**: support@example.com

🗺️ Roadmap

□ Support for ChatGPT Teams/Enterprise exports
□ Real-time monitoring mode

- □ Plugin system for custom analyzers
- □ API endpoint for integration
- □ Mobile app for viewing reports
- □ Support for other chat platforms

⭐ **If you find this tool useful, please star the repository!**

Made with ❤️ by the ChatGPT Analyzer Team