Table of Contents

# Portable PsyAgent User Manual

## Table of Contents

1. [Project Overview](#project-overview)
2. [Features](#features)
3. [System Requirements](#system-requirements)
4. [Installation Guide](#installation-guide)
5. [Quick Start](#quick-start)
6. [Detailed Usage](#detailed-usage)
7. [Assessment Module](#assessment-module)
8. [Analysis Module](#analysis-module)
9. [Stress Testing Module](#stress-testing-module)
10. [Configuration Files](#configuration-files)
11. [Batch Processing](#batch-processing)
12. [Troubleshooting](#troubleshooting)
13. [API Reference](#api-reference)
14. [License](#license)

## Project Overview

Portable PsyAgent is a portable psychological assessment agent system that supports multiple large model evaluators and local Ollama models. The system can perform multi-dimensional personality assessments on AI agents, generate detailed analysis reports, and support targeted enhanced stress testing.

## Features

* 🧠 **Multi-dimensional Personality Assessment** - Supports Big Five personality trait analysis
* 🤖 **Multi-evaluator Support** - Supports OpenAI, Claude, Gemini, DeepSeek, GLM, Qwen, and local Ollama
* 🔧 **Configuration-driven** - Easily switch models and parameters through configuration files
* 📊 **Detailed Analysis Reports** - Generates comprehensive reports including motivation analysis, personality traits, and behavioral patterns
* 🛡️ **Local Assessment** - Supports fully localized Ollama model evaluation
* 🔍 **Debug Logs** - Complete conversation logs and debugging information
* 🚀 **Batch Analysis** - Automatically processes large numbers of assessment reports with intelligent batch processing and progress tracking
* 💪 **Stress Testing** - Supports emotional stress, cognitive traps, and context load stress testing

## System Requirements

* Python 3.8 or higher
* Windows, Linux, or macOS operating system
* At least 4GB RAM (8GB or higher recommended)
* Disk space: At least 10GB available space

## Installation Guide

### 1. Clone the Project

git clone https://github.com/ptreezh/AgentPsyAssessment  
cd AgentPsyAssessment

### 2. Install Dependencies

# Install base dependencies  
pip install -r requirements.txt  
  
# Optional: Install Google Gemini support  
pip install google-generativeai

### 3. Configure Environment Variables

Create a .env file or set environment variables:

# OpenAI  
OPENAI\_API\_KEY=your\_openai\_key  
  
# Anthropic Claude   
ANTHROPIC\_API\_KEY=your\_claude\_key  
  
# Google Gemini  
GOOGLE\_API\_KEY=your\_gemini\_key  
  
# Alibaba Cloud Qwen  
DASHSCOPE\_API\_KEY=your\_qwen\_key  
  
# DeepSeek  
DEEPSEEK\_API\_KEY=your\_deepseek\_key  
  
# GLM  
GLM\_API\_KEY=your\_glm\_key

## Quick Start

### Using Ollama Local Models (Recommended)

#### Install Ollama

# Windows  
# Download from https://ollama.ai/download  
  
# Linux  
curl -fsSL https://ollama.ai/install.sh | sh  
  
# macOS  
brew install ollama

#### Download Models

# Start Ollama service  
ollama serve  
  
# Download recommended models  
ollama pull llama3:latest  
ollama pull qwen3:8b  
ollama pull mistral-nemo:latest

### Basic Assessment

# Use default evaluator  
python shared\_analysis/analyze\_results.py data/your\_data.json  
  
# Use specific evaluators  
python shared\_analysis/analyze\_results.py data/your\_data.json --evaluators gpt claude  
  
# Use local Ollama evaluators  
python shared\_analysis/analyze\_results.py data/your\_data.json --evaluators ollama\_llama3 ollama\_qwen3

## Detailed Usage

### Project Structure

portable\_psyagent/  
├── llm\_assessment/ # Assessment module  
│ ├── run\_assessment\_unified.py # Unified assessment entry  
│ ├── roles/ # Role definition files  
│ ├── test\_files/ # Test question banks  
│ ├── results/ # Assessment results  
│ └── services/ # Core services  
├── shared\_analysis/ # Analysis module  
│ ├── analyze\_results.py # Results analysis main program  
│ ├── analyze\_big5\_results.py # Big Five analysis  
│ ├── analyze\_motivation.py # Motivation analysis  
│ └── ollama\_evaluator.py # Ollama evaluator  
├── interference\_materials/ # Interference materials (stress testing)  
├── config/ # Configuration files  
├── batch\_analysis\_output/ # Batch analysis output  
└── docs/ # Documentation

## Assessment Module

### Running Assessments

# Basic assessment  
python llm\_assessment/run\_assessment\_unified.py --model\_name gemma3:latest --test\_file big5 --role\_name a1  
  
# Assessment with stress testing  
python llm\_assessment/run\_assessment\_unified.py --model\_name gemma3:latest --test\_file big5 --role\_name a1 --emotional-stress-level 3 --cognitive-trap-type p  
  
# Set temperature parameter  
python llm\_assessment/run\_assessment\_unified.py --model\_name gemma3:latest --test\_file big5 --role\_name a1 --tmpr 0.7  
  
# Set context length  
python llm\_assessment/run\_assessment\_unified.py --model\_name gemma3:latest --test\_file big5 --role\_name a1 --context-length-mode static --context-length-static 4

### Assessment Parameters

* --model\_name: Model identifier (e.g., ollama/gemma3:latest)
* --test\_file: Test file name or path
* --role\_name: Role name
* --emotional-stress-level: Emotional stress level (0-4)
* --cognitive-trap-type: Cognitive trap type (p, c, s, r)
* --tmpr: Model temperature setting
* --context-length-mode: Context length mode (auto, static, dynamic, none)
* --timeout: Model response timeout (seconds)

## Analysis Module

### Motivation Analysis

# Run motivation analysis (no API required)  
python shared\_analysis/analyze\_motivation.py data/your\_data.json --debug

### Big Five Personality Analysis

# Basic Big Five analysis  
python shared\_analysis/analyze\_big5\_results.py data/your\_data.json

### Comprehensive Analysis

# Use default evaluator  
python shared\_analysis/analyze\_results.py data/your\_data.json  
  
# Use specific evaluators  
python shared\_analysis/analyze\_results.py data/your\_data.json --evaluators gpt claude  
  
# Use local Ollama evaluators  
python shared\_analysis/analyze\_results.py data/your\_data.json --evaluators ollama\_llama3 ollama\_qwen3

## Stress Testing Module

### Supported Stress Test Types

1. **Emotional Stress Testing** - Affects model performance through different levels of emotional stress
2. **Cognitive Trap Testing** - Introduces paradoxes, circularity, semantic fallacies, and procedural traps
3. **Context Load Testing** - Tests model processing capabilities by increasing context length

### Stress Testing Parameters

* -esL, --emotional-stress-level: Emotional stress level (0-4)
* -ct, --cognitive-trap-type: Cognitive trap type
  + p: Paradox trap
  + c: Circularity trap
  + s: Semantic fallacy trap
  + r: Procedural trap
* --context-length-mode: Context length mode
  + auto: Automatic detection
  + static: Fixed length
  + dynamic: Dynamic ratio
  + none: Disable context injection

## Configuration Files

### Ollama Configuration (config/ollama\_config.json)

{  
 "ollama": {  
 "base\_url": "http://localhost:11434",  
 "timeout": 120,  
 "models": {  
 "llama3": {  
 "name": "llama3:latest",  
 "temperature": 0.1,  
 "max\_tokens": 1024,  
 "description": "Meta Llama 3 - General-purpose large model"  
 },  
 "qwen3": {  
 "name": "qwen3:8b",  
 "temperature": 0.1,  
 "max\_tokens": 1024,  
 "description": "Alibaba Cloud Qwen3 - 8B parameter version"  
 },  
 "mistral": {  
 "name": "mistral-nemo:latest",  
 "temperature": 0.1,  
 "max\_tokens": 1024,  
 "description": "Mistral NeMo - High-performance reasoning model"  
 }  
 }  
 },  
 "evaluators": {  
 "ollama\_llama3": {  
 "provider": "ollama",  
 "model": "llama3",  
 "description": "Llama3 local evaluator"  
 },  
 "ollama\_qwen3": {  
 "provider": "ollama",  
 "model": "qwen3",  
 "description": "Qwen3 local evaluator"  
 },  
 "ollama\_mistral": {  
 "provider": "ollama",  
 "model": "mistral",  
 "description": "Mistral NeMo local evaluator"  
 }  
 }  
}

## Batch Processing

### Batch Analysis

# View file statistics  
python ultimate\_batch\_analysis.py --stats  
  
# Quick test (5 files)  
python ultimate\_batch\_analysis.py --quick  
  
# Analyze specific model (e.g., deepseek)  
python ultimate\_batch\_analysis.py --filter deepseek  
  
# Complete batch analysis (all 294 files)  
python ultimate\_batch\_analysis.py  
  
# One-click start for Windows users  
start\_batch\_analysis.bat

### Supported Assessment Data

The system supports automatic analysis of assessment reports in the results/results directory, containing test data from multiple model series.

### Batch Analysis Features

* 🔄 **Automatic Format Conversion** - Supports original assessment data formats
* 📊 **Intelligent Batch Processing** - Supports resume and error recovery
* ⏱️ **Progress Tracking** - Real-time display of analysis progress and estimated time
* 📋 **Detailed Reports** - Generates JSON and Markdown format summaries
* 🎯 **Flexible Filtering** - Filters by model, sample size, and other conditions

## Troubleshooting

### Common Issues

1. **Ollama Connection Failure**

* # Check Ollama service  
  ollama ps  
  curl http://localhost:11434/api/tags

1. **Batch Analysis Interrupted**

* # Check output directory  
  ls -la batch\_analysis\_results/  
    
  # Re-run (will automatically skip completed files)  
  python ultimate\_batch\_analysis.py --filter deepseek

1. **Insufficient Memory**

* # Reduce batch size  
  python ultimate\_batch\_analysis.py --sample 10

1. **API Key Issues**

* # Check environment variables  
  echo $OPENAI\_API\_KEY

1. **Missing Modules**

* # Install missing dependencies  
  pip install google-generativeai

### Debug Mode

# Enable detailed debug output  
python shared\_analysis/analyze\_results.py data.json --evaluators ollama\_llama3

Check log files: - logs/evaluator\_conversation\_log.txt - Conversation logs - logs/debug\_info.json - Debug information

## API Reference

### Assessment API

#### run\_assessment\_unified.py

Main parameters: - --model\_name (required): Model identifier - --test\_file (required): Test file - --role\_name: Role name - --debug: Enable debug mode - --test\_connection: Test model connectivity only

Stress testing parameters: - --emotional-stress-level: Emotional stress level (0-4) - --cognitive-trap-type: Cognitive trap type (p, c, s, r) - --tmpr: Model temperature setting - --context-length-mode: Context length mode - --timeout: Response timeout

### Analysis API

#### analyze\_results.py

Main functions: - Comprehensive analysis of assessment results - Generates Big Five and MBTI personality assessments - Supports multiple evaluators

#### analyze\_motivation.py

Main functions: - Motivation analysis of assessment results - Generates motivation test reports - Supports Markdown format output

#### analyze\_big5\_results.py

Main functions: - Specialized analysis of Big Five personality traits - Generates detailed Big Five scoring reports

## License

This project is for research and educational purposes only.

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.