

劇評ツール - ユーザーマニュアル

このドキュメントでは、劇評ツールの使い方について説明します。このツールは、Polisコンサルテーションデータを処理し、可視化するための単一スクリプトソリューションです。

目次

- [劇評ツール - ユーザーマニュアル](#)
 - [目次](#)
 - [概要](#)
 - [必要なモジュール](#)
 - [インストール方法](#)
 - [コマンドラインからの実行方法](#)
 - [コマンドラインオプション](#)
 - [設定ファイル \(config.json\) の記載方法](#)
 - [設定項目の説明](#)
 - [input \(入力設定\)](#)
 - [output \(出力設定\)](#)
 - [prompts \(プロンプト設定\)](#)
 - [content \(コンテンツ設定\)](#)
 - [clustering \(クラスタリング設定\)](#)
 - [ai \(AI設定\)](#)
 - [visualization \(可視化設定\)](#)
 - [CSVファイルの記載方法](#)
 - [プロンプトファイルについて](#)
 - [出力結果について](#)
 - [トラブルシューティング](#)

- [APIキーの設定](#)
- [エラーログの確認](#)
- [よくあるエラー](#)

概要

劇評ツールは、Polisコンサルテーションデータを処理し、可視化するためのツールです。以下の処理を一括で行います：

1. CSVデータの読み込み
2. AIプロンプトを用いたデータの処理（抽出、ラベリング、要約）
3. クラスタリング処理
4. 静的なHTML/CSS/JS生成

これにより、コンサルテーションデータを視覚的に理解しやすい形で表示することができます。

必要なモジュール

このツールを実行するには、以下のPythonモジュールが必要です：

```
1 pandas
2 numpy
3 scikit-learn (sklearn)
4 matplotlib
5 requests
6 tqdm
```

インストール方法

必要なモジュールをインストールするには、以下のコマンドを実行します：

```
1 pip install pandas numpy scikit-learn matplotlib requests tqdm
```

コマンドラインからの実行方法

基本的な実行方法：

```
1 python process_all.py
```

オプションを指定して実行する方法：

```
1 python process_all.py [--config CONFIG_FILE] [--input  
INPUT_CSV] [--output OUTPUT_DIR] [--clusters NUM_CLUSTERS] [--  
api {anthropic,openai}]
```

コマンドラインオプション

- `--config`: 設定ファイルのパス（デフォルト: `config.json`）
- `--input`: 入力CSVファイルのパス（設定ファイルの値を上書き）
- `--output`: 出力ディレクトリのパス（設定ファイルの値を上書き）
- `--clusters`: クラスター数（設定ファイルの値を上書き）
- `--api`: 使用するAPI（`anthropic` または `openai`、設定ファイルの値を上書き）
- `--skip-extraction`: 意見抽出をスキップ
- `--skip-clustering`: クラスタリングをスキップ
- `--skip-labelling`: ラベリングをスキップ
- `--skip-summary`: 要約生成をスキップ

例：

```
1 python process_all.py --config my_config.json --input  
my_data.csv --output my_output --clusters 8 --api anthropic
```

設定ファイル（config.json）の記載方法

設定ファイル（config.json）は、ツールの動作を制御するための設定を記述するJSONファイルです。以下は設定ファイルの例です：

```
1 {
2   "input": {
3     "file": "inputs/data.csv",
4     "comment_column": "comment-body"
5   },
6   "output": {
7     "dir": "output/web"
8   },
9   "prompts": {
10    "dir": "prompts",
11    "extraction": "extraction.txt",
12    "labelling": "labelling.txt",
13    "takeaways": "takeaways.txt",
14    "overview": "overview.txt"
15  },
16  "content": {
17    "title": "劇評ライブラリ",
18    "question": "劇団野らぼうの公演「内側の時間」「新装版 内側の時間」の
19    劇評を基にしたAI解析",
20    "description": "このAI生成レポートは、Polisコンサルティングのデ
21    ータに基づいています。"
22  },
23  "clustering": {
24    "num_clusters": 6,
25    "method": "kmeans"
26  },
27  "ai": {
28    "provider": "anthropic",
29    "model": "claude-3-7-sonnet-20250219",
30    "embedding_model": "",
31    "temperature": 0.2,
32    "sample_size": 5,
33    "api_key": "your_api_key_here",
34    "openai_api_key": "",
35    "anthropic_api_key": "your_anthropic_api_key_here"
36  },
37  "visualization": {
38    "colors": [
```

```
37         "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#9467bd",
38         "#8c564b", "#e377c2", "#7f7f7f", "#bcbd22", "#17becf"
39     ],
40     "plotly_url": "https://cdn.plot.ly/plotly-2.29.0.min.js"
41 }
42 }
```

設定項目の説明

input（入力設定）

- `file`: 入力CSVファイルのパス
- `comment_column`: コメント列の名前

output（出力設定）

- `dir`: 出力ディレクトリのパス

prompts（プロンプト設定）

- `dir`: プロンプトファイルのディレクトリ
- `extraction`: 意見抽出用プロンプトファイル
- `labelling`: クラスターラベリング用プロンプトファイル
- `takeaways`: 要点抽出用プロンプトファイル
- `overview`: 全体概要生成用プロンプトファイル

content（コンテンツ設定）

- `title`: ウェブサイトのタイトル
- `question`: コンサルテーションの質問
- `description`: ウェブサイトの説明

clustering（クラスタリング設定）

- `num_clusters`: クラスター数
- `method`: クラスタリング手法（現在は"kmeans"のみサポート）

ai（AI設定）

- `provider`: 使用するAIプロバイダー（"anthropic"または"openai"）
- `model`: 使用するAIモデル
- `embedding_model`: 埋め込みモデル（現在は使用していません）
- `temperature`: 生成時の温度パラメータ
- `sample_size`: サンプルサイズ
- `api_key`: APIキー（一般的なAPIキー）
- `openai_api_key`: OpenAI APIキー
- `anthropic_api_key`: Anthropic APIキー

visualization（可視化設定）

- `colors`: クラスターの色
- `plotly_url`: Plotly.jsのURL

CSVファイルの記載方法

入力CSVファイルは、以下の形式で記述します：

- 1 タイムスタンプ, お名前, 作品, comment-body
- 2 2023/10/15 12:34:56, 鈴木一郎, 内側の時間, 「内側の時間」は時間の概念を独自の視点で描いていて、特に主人公の内的葛藤が印象的でした。
- 3 2023/10/15 13:45:23, 田中花子, 内側の時間, 舞台装置がミニマルでありながら効果的に使われていて、限られた空間で多様な場面転換ができていたのが素晴らしかったです。

必須の列：

- `タイムスタンプ`: コメントの投稿日時

- `お名前`: コメント投稿者の名前
- `作品`: コメントが関連する作品名
- `comment-body`: コメント本文 (`config.json`の `comment_column` で指定した列名)

注意:

- CSVファイルはUTF-8エンコーディングで保存してください。
- 列名は正確に一致させる必要があります。
- コメント列の名前は、`config.json`の `comment_column` で指定した名前と一致させる必要があります。

プロンプトファイルについて

プロンプトファイルは、AIに指示を与えるためのテキストファイルです。以下の4種類のプロンプトファイルが必要です:

1. `extraction.txt`: コメントから意見を抽出するためのプロンプト
2. `labelling.txt`: クラスターにラベルを付けるためのプロンプト
3. `takeaways.txt`: クラスターの要約を生成するためのプロンプト
4. `overview.txt`: 全体の概要を生成するためのプロンプト

各プロンプトファイルは、以下の形式で記述します:

```
1 /system
2   (システムプロンプト)
3
4 /human
5   (ユーザープロンプト)
```

プロンプトファイルは、`config.json`の `prompts.dir` で指定したディレクトリに配置します。

出力結果について

処理が完了すると、config.jsonの `output.dir` で指定したディレクトリに以下のファイルが生成されます：

- `index.html`: メインのHTMLファイル
- `styles.css`: スタイルシート
- `script.js`: JavaScriptファイル
- `data.json`: 処理結果のJSONデータ

生成されたウェブサイトは、以下のセクションで構成されています：

1. 全体の概要
2. 意見の分布（散布図）
3. 作品別分析
4. クラスター分析
5. 全てのコメント

トラブルシューティング

APIキーの設定

APIキーは、以下の方法で設定できます：

1. config.jsonの `ai.api_key`、`ai.openai_api_key`、または `ai.anthropic_api_key` に直接設定
2. 環境変数 `OPENAI_API_KEY` または `ANTHROPIC_API_KEY` に設定

環境変数が設定されている場合は、config.jsonの設定よりも優先されます。

エラーログの確認

エラーが発生した場合は、`process.log` ファイルを確認してください。このファイルには、処理中のログが記録されています。

よくあるエラー

1. CSVファイルの読み込みエラー

- CSVファイルが存在するか確認してください。
- CSVファイルのエンコーディングがUTF-8であるか確認してください。
- 列名が正しいか確認してください。

2. API呼び出しエラー

- APIキーが正しく設定されているか確認してください。
- インターネット接続が正常であるか確認してください。
- APIの利用制限に達していないか確認してください。

3. クラスタリングエラー

- データ量が少なすぎる場合、クラスタリングが正常に機能しない場合があります。
- クラスター数を減らしてみてください。

4. メモリエラー

- 大量のデータを処理する場合、メモリ不足になる可能性があります。
- データ量を減らすか、より多くのメモリを持つマシンで実行してください。