

UESI Examination Build System

取扱説明書（PC 仕様）

UESI プロジェクト / Streamlit アプリ

作成日: 2026-01-17

改訂履歴

版	日付	変更内容	担当
0.1	2026-01-18	初版	-

目次

改訂履歴	2
目次	エラー! ブックマークが定義されていません。
1. 本システムの概要	5
1.1 想定利用者	5
1.2 できること（機能一覧）	5
2. 動作環境・起動方法	5
2.1 必要環境	5
2.2 データ保存場所（重要）	6
3. 画面構成と基本操作（サイドバー）	6
3.1 データベース	6
3.2 絞り込み	7
4. 各タブの使い方	8
4.1 スタート（登録済み入試一覧）	8
4.2 閲覧（パッセージ一覧・詳細）	9
4.3 新規解析(JSON)（追加/更新）	10
4.4 DB 編集（フォーム編集）	11
4.5 文法抜粋（文法タグ一致の検索）	12
4.6 語彙一覧（キーワード/コロケーション）	13
4.7 JSONL 書き出し（条件付きエクスポート）	13
4.8 DB ダウンロード(※重要)	14
5. 入力 JSON 仕様（新規解析/DB 編集）	14
5.1 必須項目	15
5.2 JSON 構造（概要）	15
5.3 サンプル JSON	15
5.4 上書き判定（replace_if_exists）	16
6. UESI 解析・生成エンジンについて	17
6.1 解析エンジンの使い方	18
6.2 英文生成エンジンの使い方	19
6.3 使用モデルについて	19
7. 推奨ワークフロー	20
8. トラブルシューティング	20

8.1 よくあるエラーメッセージ	20
8.2 アップロードした DB が反映されない	21

1. 本システムの概要

『公開用 KB ビルダー』は、解析済みの英文（入試英文など）を SQLite データベースに格納し、Web UI（Streamlit）上で閲覧・編集・検索・書き出し（JSONL）できるアプリケーションです。

格納対象は、本文（英文/和訳）に加え、段落役割、論理関係、文法項目、語彙（キーワード/コロケーション）、出題意図、再利用メモ等の付随情報です。

1.1 想定利用者

- 英文解析データを登録・整理したい方
- 登録済みデータを閲覧し、授業/教材作成に活用したい方
- 文法タグや語彙で横断検索し、例文・抜粋を作成したい方

1.2 できること（機能一覧）

- SQLite DB のアップロード/切り替え
- 空の DB 作成（スキーマ自動生成）
- 条件で絞り込み（レベル、学校/入試、年度、文章タイプ、キーワード）
- パッセージ一覧・詳細閲覧（英文/和訳/文法/構造/語彙/出題意図）
- JSON 入力による新規追加/更新
- フォームによる DB 編集（メタ情報・本文・構造・文法・語彙・出題意図・再利用メモ）
- 文法タグ一致の抜粋検索
- 語彙一覧（キーワード/コロケーション）検索
- 条件付きで JSONL ヘクスポート
- SQLite DB のダウンロード

2. 動作環境・起動方法

2.1 必要環境

利用者側はインストール不要で、Web ブラウザからアクセスするだけで使用できます（推奨：Chrome/Edge/Firefox/Safari 最新版）。必要に応じて、生成されたデータ（例：SQLite DB）を端末へダウンロードして保存してください。

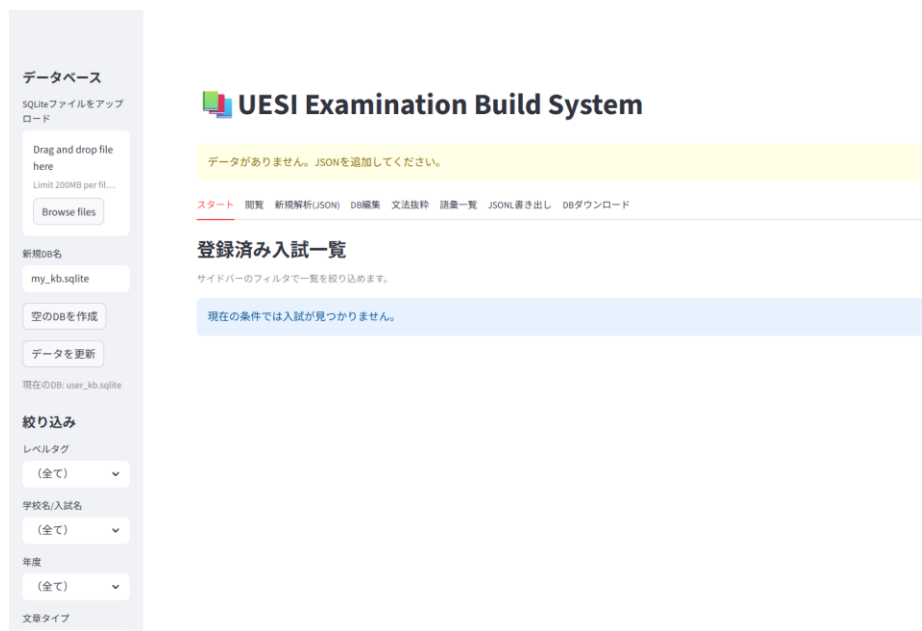
2.2 データ保存場所（重要）

本アプリは、セッションごとにランダムな session_id を払い出し、『user_dbs/<session_id>/』配下に SQLite ファイルを保存します。

公開運用する場合は、サーバ上にアップロードされた DB が残る設計になり得るため、個人情報や秘匿情報を含むデータはアップロードしない/させない運用を推奨します。

3. 画面構成と基本操作（サイドバー）

画面左側のサイドバーでは、データベース操作と絞り込み条件の設定を行います。



3.1 データベース

主な操作:

- SQLite ファイルのアップロード (.sqlite / .db)
- アップロードした DB を読み込む
- 空の DB を作成（新規 DB 名を指定）
- データを更新（キャッシュクリア）
- 現在の DB 名の表示

※アップロードした DB を読み込むボタンを押すと、ファイルはセッション領域に保存され、以降その DB が操作対象になります。

初めて使う場合は、ローカルに保存する SQLite DB がないため、4.3 で説明する「新規解析」というタブから解析 JSON を入力し、DB に英文情報を蓄積させる形になります。ページを閉じる前に DB をダウンロードし、端末に保存します。

2 回目以降、端末に保存された DB を使って情報を蓄積したい場合は、「Browse Files」を押して端末の DB を読み込みます。すると「アップロードした DB を読み込む」というボタンが出るので、そのボタンを押すと以前蓄積していた DB が読み込まれます。

「データを更新」ボタンは「新規解析」タブで入力した解析 JSON を閲覧画面に反映させるためのものです。

3.2 絞り込み

絞り込みは『レベルタグ』『学校名/入試名』『年度』『文章タイプ』『キーワード/テーマ/タグ』で行えます。

『（全て）』を選択すると、その条件は適用されません。キーワードはメタ情報（学校名/テーマ/文章タイプ/レベル）に対して部分一致で適用されます。

4. 各タブの使い方

4.1 スタート（登録済み入試一覧）

データベース

SQLiteファイルをアップロード

Drag and drop file here
Limit 200MB per file...

Browse files

englis-
2.0MB

アップロードしたDBを読み込む

新規DB名
my_kb.sqlite

空のDBを作成

データを更新

現在のDB
engfish_kb.sqlite

絞り込み

レベルタグ
(全て)

学校名/入試名
(すべて)

UESI Examination Build System

スタート 閲覧 新規解群(JSON) DB編集 文法抜粋 読解一覧 JSONL書き出し DBダウンロード

登録済み入試一覧

サイドバーのフィルタで一覧を絞り込みます。

	source_exam	year	level_tag	passages
12	2025年度 北海道大学 前期日程 第4問		2025 [KJU-TE]	1
13	2025年度 名古屋大学 前期日程 第3問		2025 [KJU-TE]	1
14	2025年度 北海道大学 前期日程 第1問		2025 [KJU-TE]	1
15	2025年度 名古屋大学 前期日程 第1問		2025 [KJU-TE]	1
16	2025年度 通研模試 (11月実施) 第5問		2025 [PRIV]	1
17	2025年度 高1進研模試 (11月実施)		2025 [PRIV]	1
18	京都大学 2025年度 第1問		2025 [TK]	1
19	北海道大学 前期日程 2025 第2問		2025 [KJU-TE]	1
28	同志社大学 2025年度 全学部日程 1第1問		2025 [KKO-MARCH]	1
29	同志社大学 2025年度 全学部日程 1第2問		2025 [KKO-MARCH]	1
30	名古屋大学 前期日程 2025 第2問		2025 [KJU-TE]	1
41	大阪大学 2025年度 (外国語学部以外) 第1問(A)		2025 [KJU-TE]	1
42	大阪大学 2025年度 (外国語学部以外) 第1問(B)		2025 [KJU-TE]	1

現在の絞り込み条件に合致するデータから、入試（source_exam）ごとの件数・最新年度・レベルタグを一覧表示します。

用途例:

- どの入試データが登録されているかを俯瞰する
- 特定の学校/年度の登録状況を素早く確認する

4.2 閲覧（パッセージ一覧・詳細）

The screenshot shows the UESI Examination Build System interface. On the left, there's a sidebar with database management options. The main area displays a table of passages. The right panel shows details for a selected passage, including the text in English and Japanese, and a list of tags and roles.

	id	year	source_exam	level_tag	topic
0	188	2025	2025年度 北海道大学 前期日程 第4問	[KYU-TEI]	"Boomer"というス
1	191	2025	2025年度 名古屋大学 前期日程 第3問	[KYU-TEI]	未来予測 (新語・
2	186	2025	2025年度 北海道大学 前期日程 第1問	[KYU-TEI]	Universal Basic Inc
3	189	2025	2025年度 名古屋大学 前期日程 第1問	[KYU-TEI]	英国鉄道の開発と
4	148	2025	2025年度 高1 進研模試 (11月実施) 第3問	[PRIV]	Noise pollution an
5	147	2025	2025年度 高1 進研模試 (11月実施)	[PRIV]	Deaf musician Eve
6	122	2025	京都大学 2025年度 第1問	[TK]	言語の平等性と
7	187	2025	北海道大学 前期日程 2025 第2問	[KYU-TEI]	Grandparents' infl
8	123	2025	同志社大学 2025年度 全学部日程 第1問	[KOD-MARCH]	人間と道具の身
9	119	2025	同志社大学 2025年度 全学部日程 第2問	[KOD-MARCH]	Noise pollution: hi
10	190	2025	名古屋大学 前期日程 2025 第2問	[KYU-TEI]	Language, discou
11	210	2025	大阪大学 2025年度 (外国語学部以外) 第1	[KYU-TEI]	Sport participator
12	211	2025	大阪大学 2025年度 (外国語学部以外) 第2	[KYU-TEI]	言語メッセージ
13	124	2025	大阪府公立高校 2025年度 A 問題 第1問	[HS]	マンホールのふた
14	129	2025	大阪府公立高校 2025年度 B 問題 第2問	[HS]	Kasutoriと世界
15	120	2025	大阪府公立高校 入試 2025年度 B 問題 第1	[HS]	Snow crystals and

左側にパッセージ一覧、右側に選択した本文 ID の詳細を表示します。パッセージ一覧を見て、ある大学の英文情報を閲覧したい場合は、「本文 ID を選択」というところで該当する本文 ID を選択しましょう。

右側詳細には以下のサブタブがあります：

- 英文: 本文 (passage_en)
- 和訳: 和訳 (passage_jp)
- 文法: 文法項目 (span、タグ、解説、難易度)
- 構造: 段落の役割 (paragraph_roles) と論理関係 (logic_relations)
- 語彙: キーワード/コロケーション
- 出題意図: why_this_text と想定設問タイプ

※パッセージ一覧は『id, year, source_exam, level_tag, topic, text_type, word_count』を表示します。

4.3 新規解析(JSON)（追加/更新）



解析結果を JSON として貼り付け、DB へ保存します。入力 JSON は本アプリの『統一フォーマット』に従う必要があります。この統一フォーマットに従った JSON は、通常 6 で説明する UESI 解析・生成 GPT を用いて ChatGPT 上で生成します。

操作手順:

1. (任意) 更新対象 passage_id を指定 (既存 ID を直接更新したい場合)
2. 『source_exam/year/topic/text_type が一致する場合は上書き』を必要に応じて設定
3. JSON を入力欄に貼り付け
4. 『保存』を押下

保存後、追加または更新の結果と passage_id が表示されます。

4.4 DB 編集（フォーム編集）

データベース

SQLiteファイルをアップロード

Drag and drop file here
Limit 200MB per file...

Browse files

アップロードしたDBを読み込む

新規DB名
my_kb.sqlite

空のDBを作成

データを更新

現在のDB:
english_kb.sqlite

絞り込み

レベルタグ
(全て)

学校名/入試名
(全て)

UESI Examination Build System

スタート 閲覧 新規解析(JSON) **DB編集** 文法検定 語彙一覧 JSON書き出し DBダウンロード

DBエディタ

本文ID
188

DBを再読み込み

メタ情報

出題元 (source_exam)
2025年度 北海道大学 前期日程 第4問

年度 (year)
2025

レベルタグ (level_tag)
[KYU-TEI]

テーマ (topic)
"Boomer" という世代ラベルとテクノロジー・言語不安

語数 (word_count)
787

文章タイプ (text_type)

本文

英文 (passage_en)
American University Professor (AP): I'm feeling a little down.
Japanese Assistant Professor (JP): What's up?
AP: One of the international students in my class was referring to me as a "Boomer".
JP: Boomer. I don't know what that is. What does it mean, and why do you feel bad about it?
AP: Boomer is a shortened way of referring to "Baby Boomers", that generation of people who was born after World War II, when global safety improved and most people thought it was a good idea to have lots of children, leading to a boom in baby births.

和訳 (passage_jp)
米国の大学教授(AP): ちょっと落ち込んでるんだ。
日本人の准教授(JP): どうしたの？
AP: 授業の留学生の一人が、僕のことを「ブーマー」って呼んだんだ。
JP: ブーマー？それが何かわからないな。どういう意味で、なぜ嫌な気分になったの？
AP: ブーマーは「ベビーブーマー」を短く言ったもので、第二次世界大戦後に生まれた世代のことだよ。世界の安全が改善して、子どもをたくさん持つのが良いことだと多くの人が考え、その結果、出生数が急増した。

登録済みデータを画面上のフォームで編集し、保存できます。解析結果が間違っていた場合や、補足情報を付け加えたい場合には、この機能を使うことでデータを修正することができます。

編集できる主な項目:

- メタ情報（出題元/年度/レベル/テーマ/語数/文章タイプ）
- 本文（英文・和訳）
- 構造（段落役割、論理関係）
- 文法項目（span、タグ、解説、難易度）
- 語彙（キーワード、コロケーション）
- 出題意図、再利用メモ

注意: 『削除する』は取り消しできません。チェックボックスで確認した上で実行してください。

4.5 文法抜粋（文法タグ一致の検索）

The screenshot displays the 'UESI Examination Build System' interface. On the left is a sidebar with database management options like 'データベース', 'SQLiteファイルを上アップロード', and '絞り込み'. The main area is titled 'UESI Examination Build System' and contains a navigation bar with links like 'スタート', '閲覧', and '文法抜粋'. Below this, the '文法タグ一致の抜粋' section is active, showing a search for '文法タグ' with 'Choose options' and 'キーワード/フレーズ' filters. The results list two items, each with a detailed explanation in English and Japanese, and associated tags like 'RG2-4A, RG3-1B, RG4-1A, RG4-3A, RG5-1A'.

登録済みの文法項目（grammar_items）から、文法タグとキーワードで抜粋を検索します。

文法タグは、タグ JSON 内から正規表現『RG 数字-数字+任意の英字（例: RG12-3a）』に一致するコードのみ抽出されます。文法タグは章末の「文法タグ体系一覧」をご覧ください。

キーワード検索では、文法事項を指定して、それにヒットした入試英文を閲覧することができます。

操作手順:

1. 文法タグを複数選択（未選択の場合はタグ条件なし）
2. （任意）キーワード/フレーズを入力（span_en または explanation_jp に部分一致）
3. 『検索』を押下

4.6 語彙一覧（キーワード/コロケーション）

データベース

SQLiteファイルをアップロード

Drag and drop file here
Limit 200MB per file

Browse files

englis...
2.0MB

アップロードしたDBを読み込む

新規DB名
my_kb.sqlite

空のDBを作成

データを更新

現在のDB:
englis_kb.sqlite

絞り込み

レベルタグ
(全て)

学校名/入試名
(全て)

キーワード/コロケーション一覧

学校名の絞り込みは source_exam の部分一致です。

学校名 (部分一致)
同志社大学

レベルタグ
(全て)

年度
(全て)

単語/表現/注記

検索

キーワード コロケーション

キーワード 392 件

page_id	word	level	note_jp	source_exam	year	level_tag	topic
0	123	embody / embodied	[KKD-MARCH] embody は「具現化する、体現する」「身体の一部として取り込む」の二つの二	同志社大学2025年度全学部日程1第1問	2025	[KKD-MARCH]	人間と道具の身体化（チニストラットと身体
1	123	extension	[KKD-MARCH] 「延長、拡張」の意味。フェデラーの発言「it's the extension of my arm」で「手	同志社大学2025年度全学部日程1第1問	2025	[KKD-MARCH]	人間と道具の身体化（チニストラットと身体
2	123	innovation	[KKD-MARCH] 「革新、革新的な考案」。ここでは Wingfield's innovation として、既存の室内ゲ	同志社大学2025年度全学部日程1第1問	2025	[KKD-MARCH]	人間と道具の身体化（チニストラットと身体
3	123	neocortex	[KKD-MARCH] 「新皮質」。大脳皮質の一部で、高次の感覚処理や認知を担う。本文では体性	同志社大学2025年度全学部日程1第1問	2025	[KKD-MARCH]	人間と道具の身体化（チニストラットと身体
4	123	patent	[KKD-MARCH] 名詞「特許」、動詞「〜の特許を取る」。本文では「had patented sphairistike」	同志社大学2025年度全学部日程1第1問	2025	[KKD-MARCH]	人間と道具の身体化（チニストラットと身体
5	123	prosthetics	[KKD-MARCH] 名詞で「義肢学」「義肢（器具）」、本文では道具の例として「prosthetics, wheel	同志社大学2025年度全学部日程1第1問	2025	[KKD-MARCH]	人間と道具の身体化（チニストラットと身体

語彙（vocab_keywords / vocab_collocations）を横断的に検索し一覧表示します。

学校名は source_exam に対する部分一致、単語/表現/注記も部分一致で検索します。

4.7 JSONL 書き出し（条件付きエクスポート）

データベース

SQLiteファイルをアップロード

Drag and drop file here
Limit 200MB per file

Browse files

englis...
2.0MB

アップロードしたDBを読み込む

新規DB名
my_kb.sqlite

空のDBを作成

データを更新

現在のDB:
englis_kb.sqlite

絞り込み

レベルタグ
(全て)

学校名/入試名
(全て)

UESI Examination Build System

スタート 閲覧 新規解析(JSON) DB閲覧 文法注釈 語彙一覧 JSONL書き出し DBダウンロード

JSONL書き出し

条件に合うデータのみを書き出します。

学校名 (カンマ区切り)

レベルタグ
Choose options

開始年
(指定なし)

終了年
(指定なし)

JSONL出力

Download JSONL

指定条件に合うパッセージを、統一JSON形式のJSONL（1行1JSON）として出力します。JSONLファイルとは 6 で説明する、ChatGPT に特定の大学と同系統の英文を生成させるために用いるファイルです。

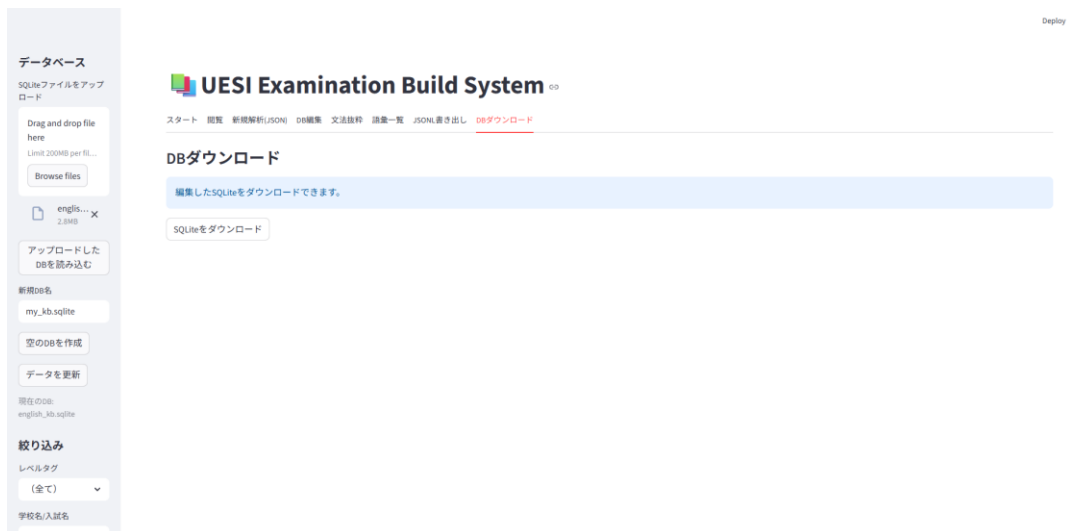
データベースは多くの大学を蓄積すると量が膨大となるので、「この大学だけの傾向に似せた英文を作りたい」と思った時は、以下の指定可能な条件を用いて絞り込み、ChatGPT が分析して英文を出せるようにファイルを生成します。

指定可能な条件:

- 学校名（カンマ区切り。source_exam の部分一致）
- レベルタグ（複数選択）
- 開始年/終了年

ファイル名として 'english_kb_export.jsonl' が生成されると、ダウンロードできるため、端末に保存してください。

4.8 DB ダウンロード(※重要)



現在操作中の SQLite DB をダウンロードできます。データベースの編集が終わり、ページを閉じる際は必ずこれをダウンロードし、バックアップを取ってください。アプリの性質上、ユーザーの DB をクラウドに保存する機能はありません。

5. 入力 JSON 仕様（新規解析/DB 編集）

本アプリは、解析結果を『統一JSON』として取り込みます。新規解析(JSON)タブの入力、およびDB 編集タブの保存データは、同一の構造を前提とします。

5.1 必須項目

必須条件: text.passage (英文本文) が空でないこと。

それ以外の項目は空でも保存可能ですが、運用上はメタ情報の整備を推奨します。

5.2 JSON 構造 (概要)

- meta: 出題元/年度/レベル/テーマ/語数/文章タイプなど (meta_json として丸ごと保存)
- text: passage (英文), jp_translation (和訳)
- structure: paragraph_roles (段落役割), logic_relations (論理関係)
- grammar_analysis: 文法項目の配列 (span_en, tag_ver2, explanation_jp, difficulty)
- vocab_analysis: keywords / collocations
- exam_intent: why_this_text, likely_question_types
- reuse_notes: avoid_next_time.grammar / avoid_next_time.topic

5.3 サンプル JSON

```
{
  "meta": {
    "source_exam": "〇〇大学",
    "year": "2024",
    "level_tag": "難関",
    "topic": "環境と社会",
    "text_type": "論説",
    "word_count": "520"
  },
  "text": {
    "passage": "(English passage here...)"
    "(2nd paragraph...)",
    "jp_translation": "(和訳...)"
  },
  "structure": {
    "paragraph_roles": [
      {"p": 1, "role": "導入", "summary_en": "...", "summary_jp": "..."},
      {"p": 2, "role": "主張", "summary_en": "...", "summary_jp": "..."}
    ],
    "logic_relations": [
      {"type": "contrast", "from_p": 1, "to_p": 2, "note": "However ..."}
    ]
  },
  "grammar_analysis": [
    {
      "span_en": "Not only A but also B",
      "tag_ver2": ["RG12-3"],
      "explanation_jp": "not only A but also B の構文",

```

```

      "difficulty": "B"
    }
  ],
  "vocab_analysis": {
    "keywords": [
      {"word": "sustainable", "level": "B2", "note_jp": "持続可能な"}
    ],
    "collocations": [
      {"expr": "play a role", "note_jp": "役割を果たす"}
    ]
  },
  "related_sources": [],
  "exam_intent": {
    "why_this_text": "論理展開と要旨把握を問う",
    "likely_question_types": ["内容一致", "要旨", "語句整序"]
  },
  "reuse_notes": {
    "avoid_next_time": {
      "grammar": ["RG12-3"],
      "topic": ["環境"]
    }
  }
}

```

5.4 上書き判定 (replace_if_exists)

『source_exam/year/topic/text_type が一致する場合は上書き』を ON にすると、同一キー (source_exam, year, topic, text_type) の行が存在する場合に更新扱いになります。

このキーは DB 側で UNIQUE 制約があるため、同一キーの重複登録はできません。

6. UESI 解析・生成エンジンについて

このアプリでは、ユーザーが自由に英文の情報を DB に保存できるように構成されています。そこで、4.3 でも説明した通り、入力 JSON というテキストが必要になります。入力 JSON は通常、ChatGPT 上の専用 GPT で生成したものを利用します。

また、この GPT には 4.7 で説明している「JSONL 書き出し」タブで生成した JSONL を用いた入試準拠の英文生成エンジンとしての機能も持ち合わせています。

また、GPT のリンクはスタート画面にも掲載されています。

<https://chatgpt.com/g/g-696b901249748191ba3b9b9e80f70c3c-uesijie-xi-sheng-cheng-gpt>

UESI解析・生成GPT 5.2



UESI解析・生成GPT

作成者: community builder

UESI入試英語データベースの補助ツールです。最初に通会問を解析する場合は/GENERATE、英文を生成する場合は/ANALYZEを送信してください。

+ 質問してみましょう



6.1 解析エンジンの使い方

UESI解析・生成GPT 5.2

共有する ...

/ANALYZE

了解しました。これから私は JSON解析エンジンとして動作します。
次のターンで 英文のメタ情報（大学名、年度、大同名、レベルタグなどの出典）と 該当入試英文を私に提示してください。

また、念のため最後に
「指示に基づきJSONコードブロックを返してください」
と打っておくと、より正確かつ安定した解析が行われます。

他にも要望（語彙を多めに、文法を詳しく、など）があれば何でもおっしゃってください。

🗨️ 🗨️ ...

出典はサンプルとします。
Look at your phone or laptop keyboard and you will probably see the same curious pattern: QWERTY. At first it seems almost accidental, as if someone scattered letters without a plan, but it is a fossil from the age of early typewriters. Those machines used metal arms that could collide and jam when certain letters were struck too quickly. To reduce this trouble, designers placed common letter pairs farther apart, trading speed for reliability in a very physical way. Later typewriters improved, and computers removed the jamming problem entirely. Even so, the layout remained, partly because millions of people had already trained their fingers to follow it and because offices wanted workers who could start immediately. Here a simple lesson appears. A tool is not chosen only for its technical quality; it is chosen for the network built around it, which includes te. ↓, rs, manuals, and software that expects a familiar standard. Even when a rival system promises greater speed, switching

+ 質問してみましょう

🗨️ 🗨️

ChatGPT の回答はどれも正しいとは限りませんが、重要な情報は確認するようにしてください。cookie の設定を参照してください。

まずリンクを開いた後、「/ANALYZE」とだけ打ちます。そうすると固定で写真のようなメッセージが出るので、指示に従って英文の出典情報と英文全体を送信します。すると以下のようなJSON コードブロックが出力されます。もしコードブロックではなくテキストで出てきた場合はアプリが読み取れないので、テキストをコピーして、「このテキストをJSONコードブロックに直してください」とすると修正できるはずです。

UESI解析・生成GPT 5.2

共有する ...

```
{
  "meta": {
    "source_exam": "サンプル（出典はサンプル扱い）:contentReference[alcite:0][index=0]",
    "year": "0000",
    "level_tag": "[HS-ADV](高校入試上位)",
    "topic": "QWERTY配列と「切り替えコスト」による標準の固定化（パス・ディペンデンス）",
    "word_count": "229",
    "text_type": "論評文"
  },
  "text": {
    "passage": "Look at your phone or laptop keyboard and you will probably see the same curious pattern: QWERTY. At first it seems almost accidental, as if someone scattered letters without a plan, but it is a fossil from the age of early typewriters. Those machines used metal arms that could collide and jam when certain letters were struck too quickly. To reduce this trouble, designers placed common letter pairs farther apart, trading speed for reliability in a very physical way. Later typewriters improved, and computers removed the jamming problem entirely. Even so, the layout remained, partly because millions of people had already trained their fingers to follow it and because offices wanted workers who could start immediately. Here a simple lesson appears. A tool is not chosen only for its technical quality; it is chosen for the network built around it, which includes te. ↓, rs, manuals, and software that expects a familiar standard. Even when a rival system promises greater speed, switching",
    "jp_translation": "スマホやノートパソコンのキーボードを見てみると、たぶん同じ不思議な並びに気づくはず",
    "structure": {
      "paragraph_roles": [
        {
          "p": 1,
          "role": "導入→歴史的背景→一般化（標準の固定化）→結論",
          "summary_en": "The passage explains why the QWERTY keyboard layout survived after its original purpose was obsolete.",
          "summary_jp": "本文は、QWERTY配列が本来の技術的理由（ジャム防止）が消えた後も残った理由を、学習者の視点から説明している。"
        }
      ],
      "logic_relations": [
        {
          "type": "原因→結果",
          "from_p": "1",
          "to_p": "1",
          "note": "タイプライターの機械的制約→QWERTY配列、制約の消滅後も学習・職場環境・周辺ネットワーク→標準"
        },
        {
          "type": "具体化→一般化",
          "from_p": "1",
          "to_p": "1",
          "note": ""
        }
      ]
    }
  }
}
```

+ 質問してみましょう

🗨️ 🗨️

ChatGPT の回答はどれも正しいとは限りませんが、重要な情報は確認するようにしてください。cookie の設定を参照してください。

6.2 英文生成エンジンの使い方



リンクを開いた後、「/GENERATE」とだけ打ちます。同様に写真のようなメッセージが出るので、指示に従ってアプリで生成したJSONLと、英文生成にあたっての要望を送信します。

6.3 使用モデルについて

解析・生成ともに ChatGPT の使用モデルは GPT Thinking 以上を推奨しています。多くの場合は無料での使用などで Thinking を利用できない場合が多いと思いますが、通常使用とされる Instant でも動作はするのでご安心ください。今後、OpenAI API などを利用して端末のアプリ内で解析、

生成を行える使用にしたいと思っています。API 利用や開発費の制約上、有料にはなるとは思います
が、低価格で運用できるように努めたいと思います。

7. 推奨ワークフロー

代表的な作業フロー例です。運用に合わせて調整してください。

1. (初回) 空の DB を作成し、運用 DB として保存する
2. 英文を専用 GPT により解析し JSON を生成する
3. 解析済み JSON を『新規解析(JSON)』で登録する
4. サイドバーの『データを更新』ボタンを押して、『閲覧』で表示確認し、必要に応じて『DB 編集』で修正する
5. 『文法抜粋』『語彙一覧』で横断チェック（表記揺れ/タグ漏れ）を行う
6. 必要に応じて『DB 編集』で修正を行う
5. 英文生成が必要な場合、『JSONL 書き出し』でエクスポートの上、専用 GPT に JSONL を送信する
6. ページを閉じる際は、『DB ダウンロード』でバックアップを取得する

8. トラブルシューティング

8.1 よくあるエラーメッセージ

- text.passage が空です。

入力 JSON の text.passage（英文本文）が空です。必ず英文本文を入れてください。

- passage_id が見つかりません。

更新対象に指定した passage_id が DB に存在しません。ID を確認するか、空欄で追加してください。

- 開始年は終了年以下にしてください。

開始年/終了年の指定が逆です。設定を入れ替えてください。

8.2 アップロードした DB が反映されない

『アップロードした DB を読み込む』ボタンを押した後、画面が再読み込みされます。反映されない場合は『データを更新』でキャッシュをクリアし、DB 名表示が変わっているか確認してください。