卒 業 研 究 報 告 書

題　　目

生成 AI を用いた
シナリオデータに基づく動画生成手法の提案

研 究 グ ル ー プ　　第 1 研究グループ

指 導 教 員　　森 直樹 教授

令和　6　年（　　2024　年 ）度卒業

（No.　　1211201039　　）　　河地 駿太朗

大阪府立大学工学域電気電子系学類情報工学課程

目 次

図目次

表目次

1　はじめに

近年, 動画配信サービスの普及に伴い, 動画コンテンツの需要が急速に高まっている. その中にはネットコミュニティにおける共有知である表現を用いることで, 必要な素材の数を減らし, 容易に制作が可能なコンテンツが存在する.

このような表現はミーム [6] と呼ばれる. ミームとは, 進化生物学者リチャード・ドーキンスにより提唱された文化における情報の伝達単位であり, ここではインターネット上で話題となったコンテンツを指す. ミームには, 多くの人が共通で認識するネタやコラージュ画像などが含まれており, Web サイトや SNS を通じて人から人へ広がる現象として捉えられている. 現代のインターネット文化では, ミームが日常的に共有され, コンテンツ制作でも積極的に活用されている.

本研究では, シナリオデータから必要な情報を抽出し, 公知のミームを活用した動画を自動生成する手法を提案する. この手法により, シナリオデータをもとにして動画生成が容易になるため, 制作時間の短縮やコスト削減が期待される. また, コンテンツの大量生産が可能となり, より多くの動画を短期間で制作することができる. 具体的には, まずシナリオデータを解析し, そこから抽出した情報に基づいて適切な背景画像やミーム素材を選定する. その後, それらを組み合わせて動画を生成することで, コンテンツが自動的に完成する仕組みを構築する.

現在, さまざまな動画生成 AI が登場しており, 映像表現の高度化が進んでいる. しかし, すべての動画が高度な映像技術を必要とするわけではない. むしろ, コンテンツによっては視覚的な演出よりもシナリオそのものに重点を置いた動画が有効な場合もある. 特に, 本研究で扱う猫ミームは公知であるため, 視聴者はそのミームに対して既存のイメージを共有しており, 複雑な映像表現がなくてもシナリオの内容を容易に理解できる.

本研究の手法は, ミームの持つ高い認識性と親しみやすさを利用し, 効果的かつ低コストな動画生成を実現するものである. これにより, 動画制作の効率化だけでなく, 視聴者にとっても分かりやすく魅力的なコンテンツを提供できることが期待される.

以下に本論文の構成を示す. まず 2 章では, 本研究で使用する要素技術につ

いて概説する. 3 章では, 本研究で提案するシナリオ解析に基づく動画生成の方法を順を追って説明し, 4 章で実験の概要と結果, 考察を示す. 最後に 5 章で本研究の成果のまとめと今後の課題について述べる.

2 要素技術

2.1 Large Language Model (LLM)

大規模言語モデル (Large Language Model, LLM) とは, 大規模なデータセットを使用して訓練された自然言語処理のモデルを指す. ゼロショットや少数ショット学習を活用することで, 自然言語理解, 感情分析, 文章生成などのさまざまなタスクを実現できる.

大規模言語モデルの代表例としては, 2018 年に Google が発表した BERT [7] や, 2020 年に OpenAI が発表した GPT-3 [8], 2023 年に同じく OpenAI が発表した GPT-4 [9], 2023 年に Google が発表した Gemini [10] などが挙げられる.

2.2 Generative Pre-trained Transformer (GPT)

Generative Pre-trained Transformer (GPT) は, OpenAI が開発した言語モデルである. 自然言語や生成タスクにおいて, 高速で効率的な処理を実現し, 複雑な文脈を理解する能力を有する.

図 2.1 に, 2018 年に OpenAI が発表した GPT-1 [1] のアーキテクチャを示す. GPT-1 は, 埋め込み層, Transformer Decoder 12 層, 出力層の 3 つの部分に分かれている. また, 図 2.2 に, さまざまなタスクを微調整するための入力の変換について示す.

2.2.1 埋め込み層

埋め込み層は, モデルが入力トークンを処理するための層である. 主に, トークン埋め込みと位置埋め込みの 2 つの要素を組み合わせて, トークンの意味的特徴と順序情報をモデルに伝える. トークン埋め込みと位置埋め込みを加算して, 各トークンの埋め込みベクトルを作成することで, モデルが文脈と順序を理解できるようになる.

具体的には, 2.1 式に示すように, トークンの意味的特徴と位置情報を同時に含んだ h_0 を計算する.

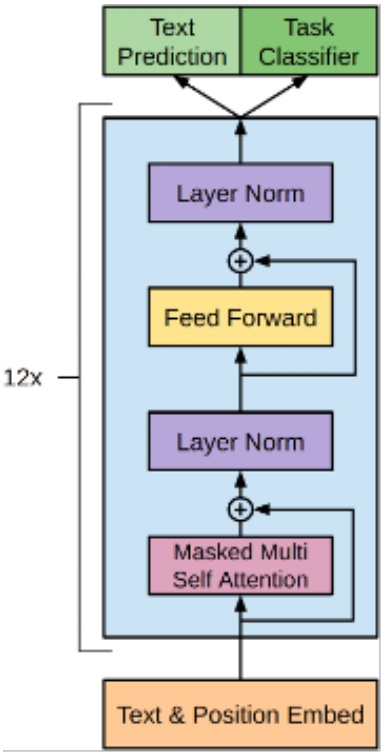

図 2.1: GPT-1 のアーキテクチャ[1]

$$h_0 = UW_e + W_p \tag{2.1}$$

ここで, U は入力トークン, W_e はトークン埋め込み行列, W_P は位置埋め込み行列を表す.

2.2.2　Transformer Decoder 12 層

このモデルでは, 言語モデルの訓練に 12 層の Transformer [11] のデコーダアーキテクチャを使用している. 自己注意 (Self Attention) 機構を使用して, 入力トークン間の関係を捉え, 情報を伝達する. 具体的には, 多頭自己注意 (Masked Multi Self Attention) とフィードフォワード層 (Feed Forward Layer) を組み合わせた層が繰り返し適用される. 12 層のデコーダが処理を終えた後, 最終的な出力はソフトマックス層に渡され, 次のトークンの予測が実行される.

2.2 式, 2.3 式にデコーダの処理を示す.

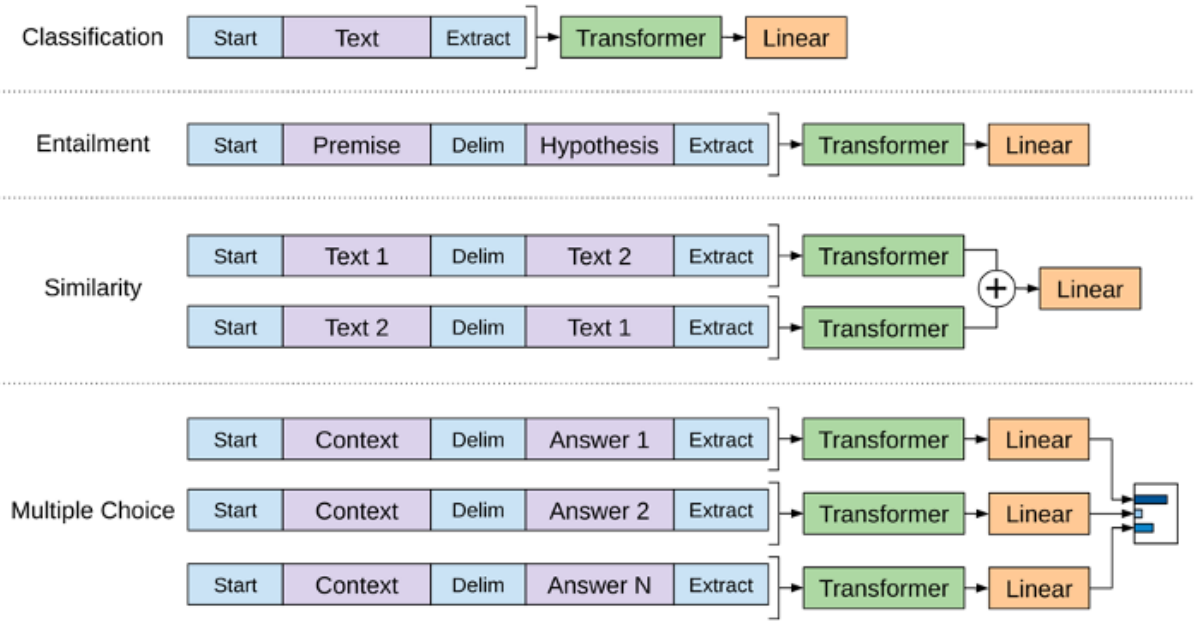

図 2.2: タスクに合わせた入力の変換 [1]

$$h_l = \text{transformer\_block}(h_{l-1}) \quad \forall l \in [1, n] \tag{2.2}$$

$$P(u) = \text{softmax}(h_n W_e^T) \tag{2.3}$$

ここで, $P(u)$ はモデルが予測する次のトークンの確率分布を表す.

2.2.3　出力層

出力層は, 文章予測 (Text Prediction) とタスク分類 (Task Classifier) に分かれている. 前者は, GPT-1 が学習する言語モデルの出力のための層であり, 後者はファインチューニングにおいて, タスクごとに応じた出力を得るための層である.

2.2.4　タスクに合わせた入力の変換

文章分類 (Classification) には, 調整がほとんど必要ない. 文章の初めと終わりが認識できるように, ターゲットとなる文章の初めに Start, 終わりに Extract というトークンを差し込む.

文章含意 (Entailment) には, 前提と仮定が与えられ, 前提が真のとき, 仮定も真になるかを判定するタスクである. 前提 (Premise) と仮定 (Hypothesis) が入力となるため, この 2 つの間に区切り文字 (Delimiter) トークンを差し込み, これを 1 つの入力とする.

類似性 (Similarity) は, 入力として与える文の順番に意味がなく, 順番の要素を排除するために, 文章 1・区切り文字・文章 2 と文章 2・区切り文字・文章 1 という順番の異なる 2 つを入力として与え, それぞれの出力の和を線形出力層に渡す.

質疑応答 (Question Answering) と常識推論 (Commonsence Reasoning) に関しては, 文脈と質問, 回答として複数の選択肢が与えられる. 文脈と質問・区切り文字・回答の 1 つという形式ですべての回答について入力を作成し, その後ソフトマックス層で正規化され, 最終的な回答の分布が生成される.

本研究では, シナリオデータからの情報の抽出, 背景画像の生成用プロンプトの作成, ミームの決定において, gpt-4 というモデルを使用した.

2.3　Gemini

Gemini [10] は, Google が開発したマルチモーダル AI モデルで, 画像, 音声, 動画, テキストなど複数のデータ形式を統合的に処理することができる. 文脈理解や応答精度に優れており, 多岐にわたるタスクに対応可能である. 高性能なアルゴリズムにより, リアルタイムでの処理能力が高く, ユーザにとって直感的で柔軟なインタラクションが可能となっている.

本研究では, テキストから情報を抽出する際に, GPT と比較する形で使用した. 実験には, gemini-1.5-flash というモデルを使用した.

2.4　Structured Query Language (SQL)

Structured Query Language (SQL) [12] は, リレーショナルデータベース [13] においてデータの操作や管理をするための標準的な言語である. SQL は, データの検索, 追加, 更新, 削除などの操作に加えて, テーブルの作成や変更, アクセ

ス権限の設定といったデータベース構造の管理機能も備えている. SQL の構文は比較的単純であり, 直感的な記述が可能であるため, 広く利用されている.

本研究ではデータ管理と情報抽出のために MySQL を採用した. MySQL はオープンソースのリレーショナルデータベース管理システムであり, 高いパフォーマンスと拡張性を持つことから, 商用・非商用を問わず多くのプロジェクトで使用されている. 加えて, SQL を用いることで複雑なデータ操作も効率的に実現可能であり, 本研究のデータ処理においてもその利点を十分に活用した.

2.5　Facebook AI Similarity Search (FAISS)

Facebook AI Similarity Search (FAISS) [14] は, 高次元ベクトルの効率的な類似性探索およびクラスタリングを実施するためのオープンソースライブラリである. 主に大規模データセットに対する近似最近傍探索 (Approximate Nearest Neighbor Search) [15] を目的として開発されており, メモリ内での高速検索やインデックス圧縮技術を駆使して, 高速かつスケーラブルな処理を実現する. FAISS には多様なインデックス手法が用意されており, データサイズや精度要件に応じて適切な方法を選択できる.

本研究では, 背景画像の選択において FAISS を活用し, GPT を使用したアプローチと比較した. FAISS の高速な検索性能は, 大量の画像データに対しても効率的な探索を可能とし, より正確な画像の選択を実現した.

2.6　Stable Diffusion

Stable Diffusion [2] は, テキストから高品質な画像を生成する画像生成 AI モデルで, 拡散モデル (Diffusion Model) [16] を基盤としている. 拡散モデルは, ノイズを加えていった画像を逆方向に再構築するプロセスで画像を生成する仕組みである. 従来の敵対的生成ネットワーク (Generative Adversarial Network, GAN) [17] と比べ, 生成画像の多様性が高く, 学習が安定しやすいという利点がある.

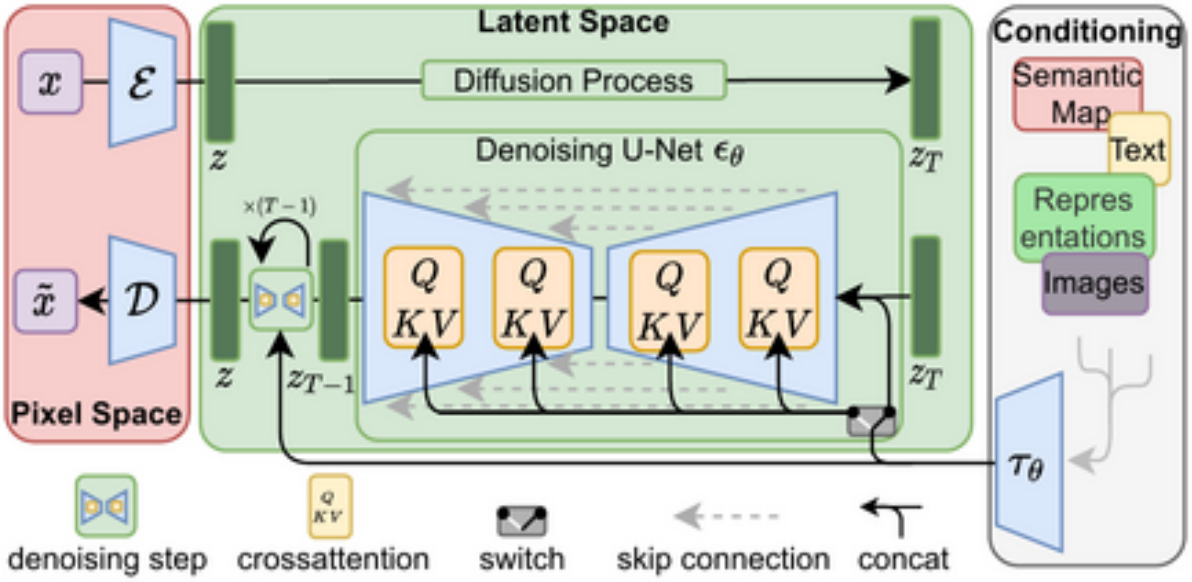

図 2.3: Stable Diffusion のアーキテクチャ[2]

Stable Diffusion の大きな特徴はオープンソースとして公開されている点で, 商用利用も含めて誰でも自由に活用・カスタマイズが可能である. この柔軟性により, 画像の部分編集 (インペインティング), スタイル変換, 画像から画像への変換 (イメージ・トゥ・イメージ) といった多様な応用が実現している.

図 2.3 に, Stable Diffusion のアーキテクチャを示す.

2.7　Adobe After Effects (AE)

Adobe After Effects は, Adobe が提供する高度なデジタル映像編集・合成ソフトウェアである. 主にモーショングラフィックスやビジュアルエフェクトを動画に加えることに利用され, 動画, テレビ番組, ウェブコンテンツ, 広告などで活用されている. After Effects は, 他の Adobe 製品 (Premiere Pro, Photoshop, Illustrator 等) との連携が強力である.

また, After Effects には, 多数の内蔵エフェクトがあり, これらを組み合わせることでさまざまな表現が可能である. さらに, プラグインやスクリプトの追加で機能を拡張し, 特定のニーズに合わせたカスタマイズが可能である.

2.8　ExtendScript

ExtendScript は, Adobe 製品向けの JavaScript ベースのスクリプト言語で, After Effects, Photoshop, Illustrator などの自動化やカスタマイズを可能にする.

通常, ExtendScript は Visual Studio Code の拡張機能である ExtendScript Debugger を使用する. これにより, プログラムでソフトの操作を制御し, 特定の作業フローに合わせた柔軟な解決策を構築できる.

日本語で書かれたこの文から, 時間, 場所, 登場人物の状態, 状況を簡単に説明する 10 文字程度のテキストの 4 つの情報を以下の形式で抽出してください. 時間, 場所, 登場人物の状態に関して仮にわからない場合, 不明と出力してください. テキストに関しては必ず出力してください.: {input_sentence}
時間:
場所:
登場人物の状態:
テキスト情報:

図 3.1: GPT に与えるプロンプトの内容

3 提案手法

本研究では, シナリオデータを解析し, 動画を自動生成する手法を提案する. 動画を生成する手順を以下に示す.

1. LLM を使用して, シナリオデータから情報を抽出する.
2. 登場人物の状態に適するミーム素材[1]を決定する.
3. 時間と場所をもとに背景画像を決定する.
4. こうして得た素材のファイルパスとテキスト情報をもとに ExtendScript を作成し, 実行する.

3.1 LLM を使用したシナリオデータからの情報の抽出

LLM を使用して入力文から時間, 場所, 登場人物の状態, テキスト情報を入手する. ここでは, 2 つの LLM を個別に使用している. 使用した LLM は GPT と Gemini である.

図 3.1, 図 3.2 にそれぞれ GPT, Gemini で情報を抽出する際に使用したプロンプトを示す. また, 表 3.1 に GPT と Gemini の使用したモデルと温度を示す. 以降はモデルと温度をこれに統一する.

[1] https://sasalabo.net/2024/02/27/猫ミームとは/#google_vignette

日本語で書かれたこの文から, 時間, 場所, 登場人物の状態, 状況を簡単に説明する 10 文字程度のテキストの 4 つの情報を以下の形式で抽出してください. 時間, 場所, 登場人物の状態に関して仮にわからない場合, 不明と出力してください. テキストに関しては必ず出力してください.: {text}
時間:
場所:
登場人物の状態:
テキスト情報:

図 3.2: Gemini に与えるプロンプトの内容

表 3.1: GPT と Gemini のパラメータ

	model	temperature
GPT	gpt-4	0.5
Gemini	gemini-1.5-flash	1.0

3.2　ミーム素材の決定

GPT を使用してミーム素材をまとめた MySQL のデータベースから, 抽出した登場人物の状態に適する素材を決定する. 表 3.2 にミーム素材に関する情報をまとめたデータベースの一部を示す. 表 Meme_Features にはミーム ID, 特徴 ID, ファイルパスが存在している. 同一ミーム ID が複数の特徴を持つ場合があり, また複数のミーム ID が同一の特徴を持つ場合もある.

以下にミーム素材の決定方法を示す.

1. 抽出された状態を入力し, GPT を使用して特徴の表から適切な特徴の ID と名前を取得
2. その特徴 ID を持つミーム ID を取得
3. 再度状態を入力し, GPT を使用して, ミーム名を含めて再度適切なミームの ID と名前を取得

表 3.2: ミーム素材に関する表の構成

meme_id	feature_id	file_path
1	1	/Users/...
2	2	/Users/...
2	3	/Users/...
3	2	/Users/...

(左) 表 Meme_Features の構成

meme_id	meme_name
1	DJ 猫
2	EDM 猫
3	Girlfriend 猫
4	oiia 猫

(中) 表 Memes の構成

feature_id	feature_name
1	ひっかく
2	踊る
3	EDM
4	ポップ

(右) 表 Features の構成

3.3　背景画像の決定

背景画像の決定には, 3 つの手法を使用した. 1 つ目は, ミーム素材の決定と同様に GPT を使用して, MySQL のデータベースから時間と場所に適する背景画像[2]を決定するものである. 2 つ目は, FAISS を使用したベクトル検索により, MySQL のデータベースから時間と場所に適する背景画像を決定するものである. 3 つ目は, 抽出した時間, 場所の情報をもとに GPT で背景画像を説明するプロンプトを作成し, Stable Diffusion を使用して背景画像を生成するというものである.

表 3.3 に背景画像に関する情報をまとめたデータベースの一部を示す. MySQL を使用する場合に, この表を使用する. 表 images には, 画像 ID, 場所 ID, 時間 ID, ファイルパスが存在し, 1 つの場所に複数の時間が存在する形で保存している. これにより, 場所を決定した後に時間を決定するという形式で実装が可能になる.

3.3.1　GPT ・ MySQL を使用した背景画像の決定

以下に背景画像の決定方法を示す.

1. データベースから表 locations の情報をリスト形式で保持
2. 抽出された場所名と場所のリストを GPT に渡し, 適する場所を出力
3. その場所に対する場所 ID を取得し, 表 images を通して取り得る画像 ID と時間 ID を獲得

[2] https://min-chi.material.jp

表 3.3: 背景画像に関する表の構成

image_id	location_id	time_condition_id	file_path
1	1	3	/Users/...
2	1	2	/Users/...
3	1	1	/Users/...
4	2	3	/Users/...

location_id	location_name
1	ATM コーナー
2	アーケード商店街
3	アイランドキッチン
4	アジト

time_condition_id	time_and_condition
1	日中
2	夜
3	夕方
4	夜・照明 OFF

(上) 表 images の構成　　(左下) 表 locations の構成　　(右下) 表 time_conditions の構成

4. こうして得た時間 ID についてのみデータベースから表 time_condition の情報を取り出し, リスト形式で保持
5. 抽出された時間名と時間のリストを GPT に渡し, 適する時間を出力
6. 場所 ID と時間 ID から画像 ID を決定

図 3.4 から図 3.11 にデータベースを使用したシステムの概要を示す.

3.3.2　FAISS ・ MySQL を使用した背景画像の決定

以下に背景画像の決定方法を示す.

1. 抽出された場所名を入力し, ベクトル検索により適切な場所を出力
2. その場所に対応する ID を取得し, 抽出された時間を入力して, ベクトル検索により適切な時間を出力
3. 場所 ID と時間 ID から画像 ID を決定

3.3.3　Stable Diffusion を使用した背景画像の生成

以下に背景画像の決定方法を示す.

1. 抽出した場所と時間を入力とし, GPT により画像生成のプロンプトを作成

あなたは画像生成プロンプトを作成する AI です. 以下の情報をもとに, Stable Diffusion 用の最適な画像生成プロンプトを作成してください.

- 場所: {location_input}
- 時間: {time_input}

プロンプトの要件:
1. 美しい風景描写にすること
2. 背景画像として使える高品質な描写を想定
3. 必ず英語で書くこと
4. 複数の要素をカンマで区切る形式で生成すること（例: ”Quiet beach at sunset, orange sky, calm waves, distant lighthouse”）

最適なプロンプトを 1 行で出力してください.

図 3.3: GPT に与えるプロンプトの内容

2. これをもとに Stable Diffusion で画像を生成

画像生成には, gsdf/Counterfeit-V2.5 という Stable Diffusion モデルを使用した. 図 3.3 に GPT に与える画像生成プロンプトの作成プロンプトを示す.

3.4　ExtendScript の生成・実行

3.1. で得たテキスト情報と 3.2 と 3.3 で得たファイルパスを使用して動画を作成するためのスクリプトを生成する. 動画の長さ, 素材の位置やサイズ, テキストの表示時間等を事前に指定したテンプレートに対して, テキスト情報とファイルパスを入力することで, 動画を生成するための ExtendScript ファイルを出力させる. このコードを実行することで動画が生成される.

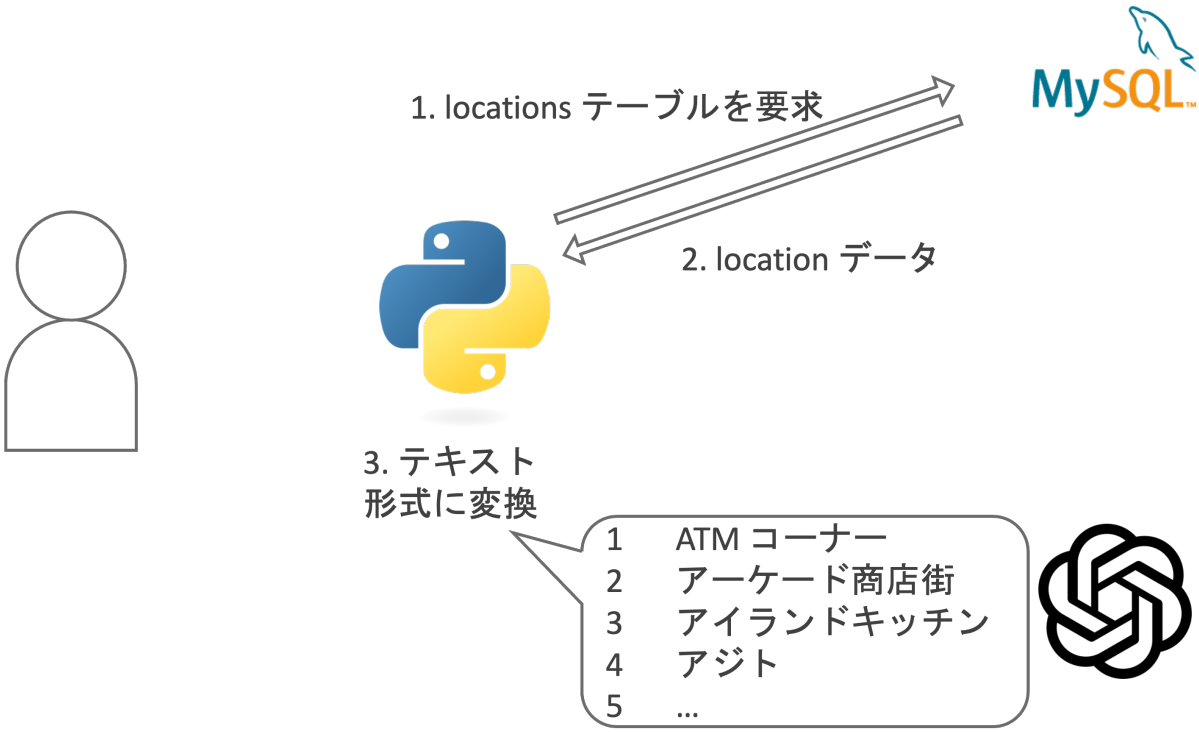

図 3.4: システムの概要 (1) [3] [4] [5]

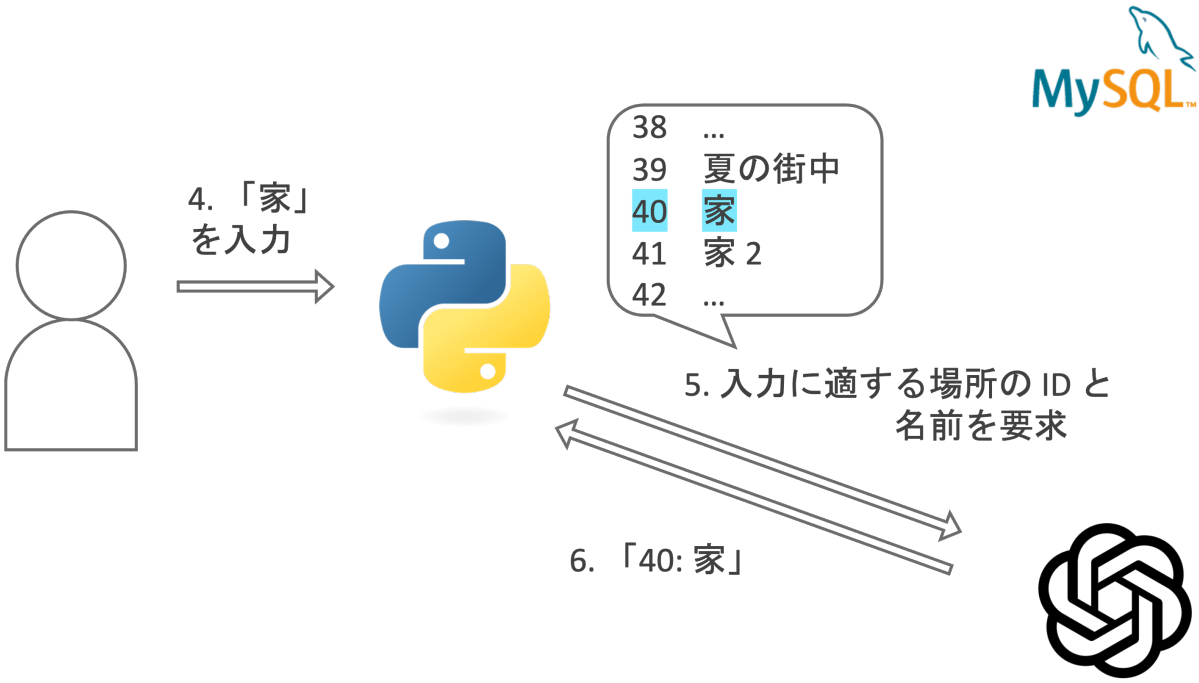

図 3.5: システムの概要 (2) [3] [4] [5]

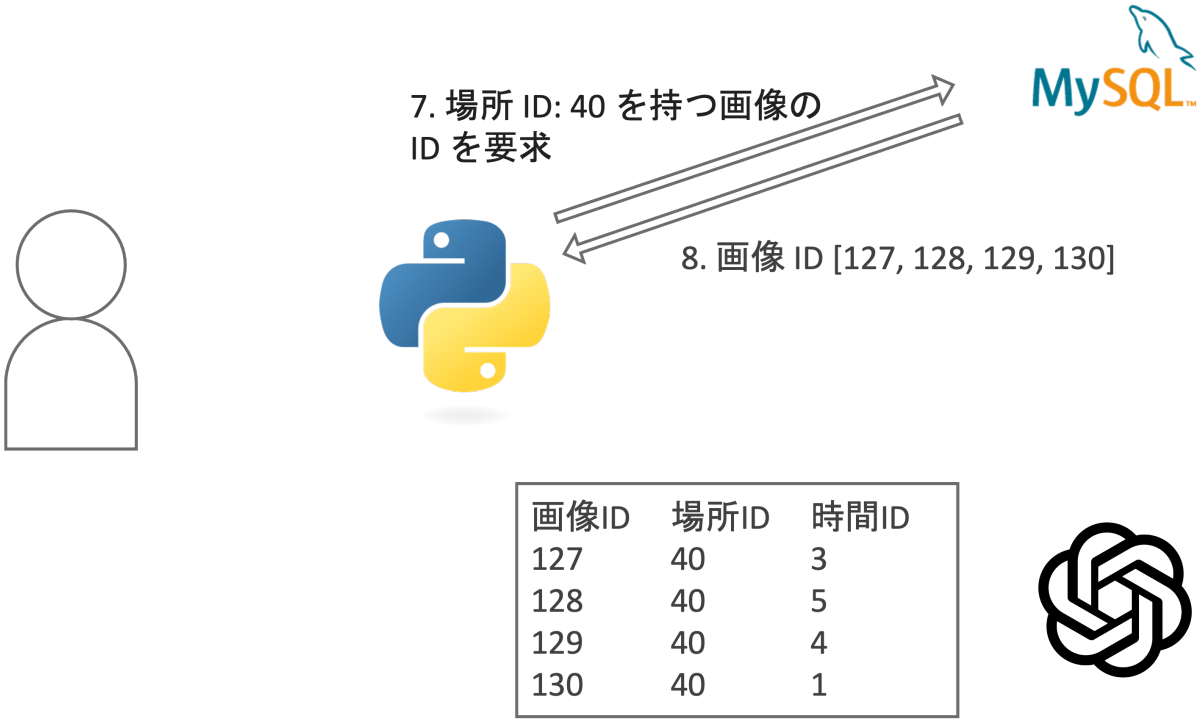

図 3.6: システムの概要 (3) [3] [4] [5]

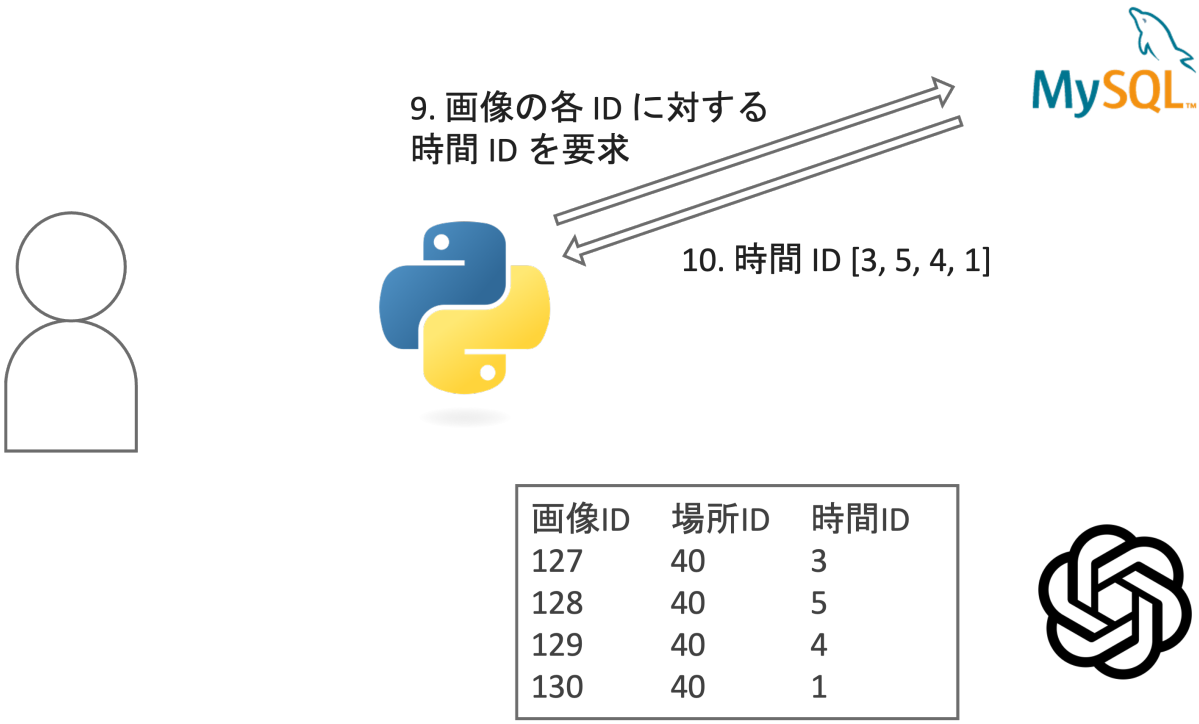

図 3.7: システムの概要 (4) [3] [4] [5]

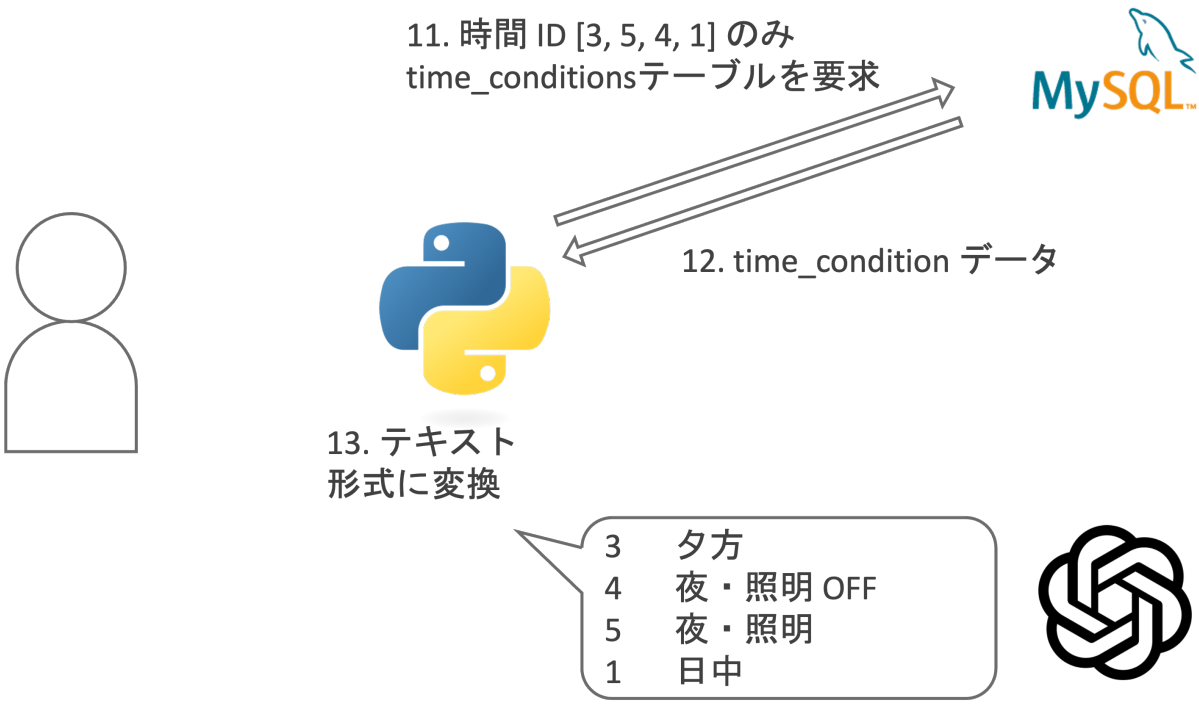

図 3.8: システムの概要 (5) [3] [4] [5]

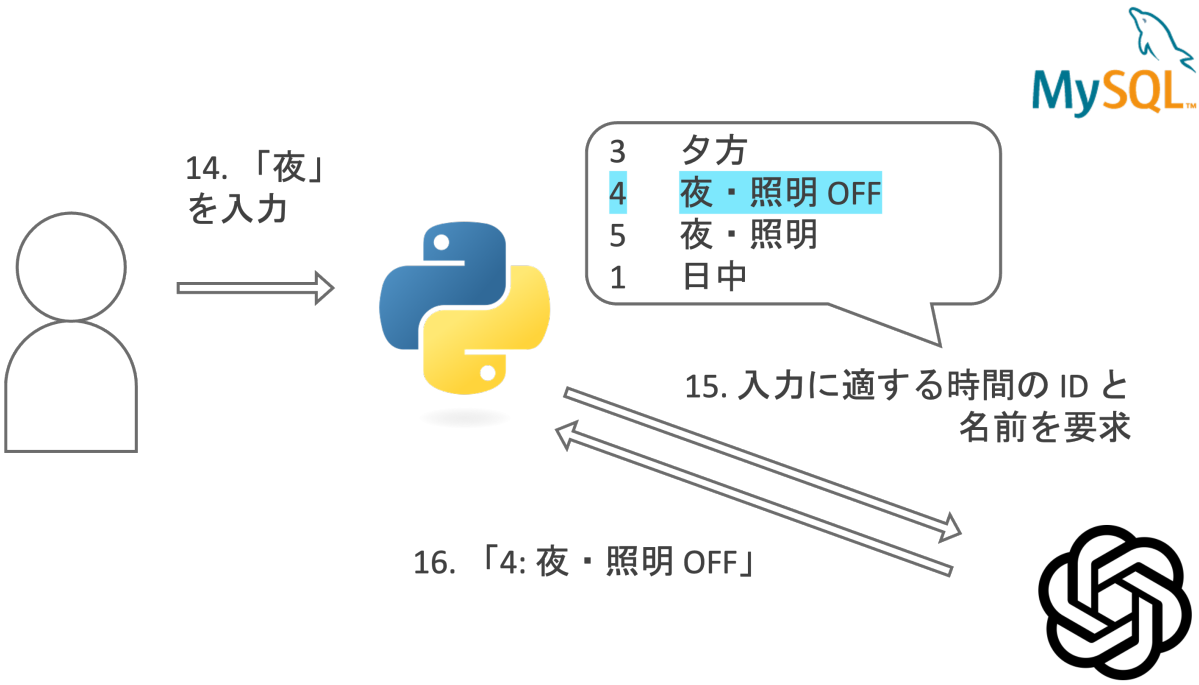

図 3.9: システムの概要 (6) [3] [4] [5]

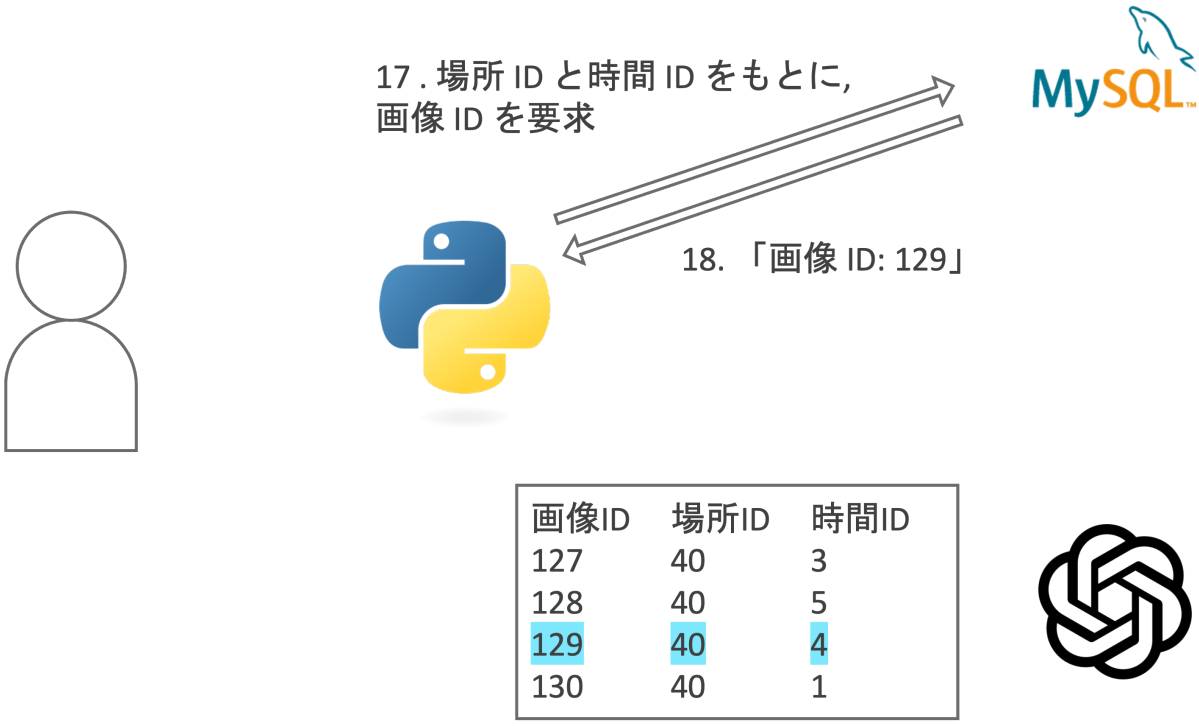

図 3.10: システムの概要 (7) [3] [4] [5]

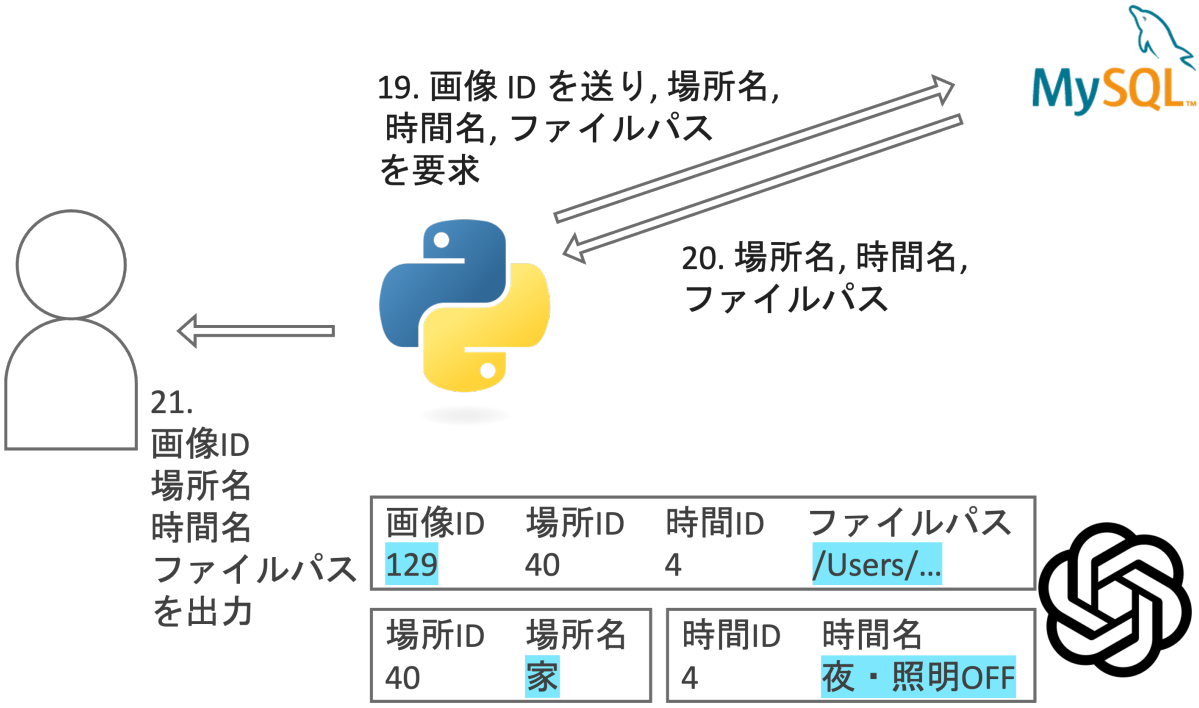

図 3.11: システムの概要 (8) [3] [4] [5]

4　実験

4.1　実験方法

本研究では, 3 つの実験を実施した.

1 つ目は, シナリオデータから情報を抽出する際に, 使用する LLM を GPT と Gemini で比較する実験である. 同一のプロンプトを使用して, 同じ入力文から抽出した情報を 5 回試行で比較した.

2 つ目は, 背景画像を決定する際に, GPT を使用した場合と FAISS を使用した場合, Stable Diffusion で生成した場合で性能を比較する実験である. GPT と FAISS を使用したものについては, 異なる 5 つの場所と時間を入力とし, 得られた場所名, 時間名を比較した. Stable Diffusion を使用したものについては, 先ほどと同じ異なる 5 つの場所と時間の入力をもとに生成したプロンプトと生成画像を示す.

3 つ目は, シナリオデータに基づいて実際に生成した動画を, シナリオデータからの情報の抽出とミーム素材の決定の観点で評価した.

4.2　実験結果

4.2.1　実験 1

以下に使用した入力文を示す.

入力文: 勉強している時は、一休みすると気分が楽になる。

表 4.1 に GPT を使用した場合の 5 回試行の結果を示す. 表 4.2 に Gemini を使用した場合の 5 回試行の結果を示す.

4.2.2　実験 2

表 4.3 に GPT を使用した場合の 5 つの入力を与えた結果を示す. 表 4.4 に FAISS を使用した場合の 5 つの入力を与えた結果を示す. 表 4.5 に場所と時間をもとに GPT を使用して作成された画像生成プロンプトを示す. 図 4.1 から図 4.5 は表 4.5 のプロンプトにより生成された画像である.

4.2.3　実験 3

以下に使用したシナリオデータを示す.

シナリオデータ:

学生にとって学校での勉強はとても重要である。部屋で長時間勉強していると、疲れてしまうこともあるだろう。そんな時は、少し休憩すると気分がとても楽になる。

表 4.6 にシナリオデータから抽出した情報を示す. 空欄は特定できなかったことを表す. シナリオデータと比較すると, 時間, 場所, 登場人物の状態がシナリオデータと一致していることがわかる.

表 4.7 に決定したミーム素材に関する情報を示す. 勉強中に対して PC 猫, 疲れてしまうに対してしょんぼり猫のように登場人物の状態とミームの雰囲気が一致している.

表 4.1: 実験 1: GPT 5 回試行

時間	場所	登場人物の状態	テキスト情報
勉強中	不明	一休み中	気分が楽になる
勉強中	不明	一休み中	気分が楽になる
勉強中	不明	一休み中	気分が楽になる
勉強中	不明	一休み中	気分が楽になる
勉強中	不明	一休み中	気分が楽になる

表 4.2: 実験 1: Gemini 5 回試行

時間	場所	登場人物の状態	テキスト情報
勉強中	不明	疲れている	気分転換が必要
勉強中	不明	疲れている	気分転換必要
勉強中	不明	疲れている	気分転換が必要
勉強中	不明	疲れている	気分転換必要
勉強中	不明	疲れている	一休みで気分転換

表 4.3: 実験 2: GPT 出力結果

入力場所	入力時間	出力場所	出力時間
ファミレス	昼	レストラン	日中
図書館	夜	市立図書館	照明 ON
学校	昼休み	学校のベンチ	日中
街中	夕方	街中のビル	夕方
自宅	夜	家	夜・照明 OFF

表 4.4: 実験 2: FAISS 出力結果

入力場所	入力時間	出力場所	出力時間
ファミレス	昼	アジト	照明 ON
図書館	夜	図書室	夕方
学校	昼休み	学校のベンチ	日中
街中	夕方	都会の街中	夕方
自宅	夜	家	夕方

表 4.5: 実験 2: GPT で作成した画像生成プロンプト

入力場所	入力時間	生成プロンプト	生成画像
ファミレス	昼	A bustling family restaurant at noon, vibrant colors of the interior, sunlight streaming through the windows, patrons enjoying their meals, detailed food items on the tables	図 4.1
図書館	夜	A serene library at night, softly glowing lamps illuminating rows of books, tranquil atmosphere, high vaulted ceilings, grand wooden bookshelves, a lone reading desk	図 3.2
学校	昼休み	Peaceful schoolyard during lunch break, lush green trees, students enjoying their meals, clear blue sky, grand school building in the background	図 4.3
街中	夕方	A bustling city street at dusk, glowing street lamps, long shadows stretching across the pavement, vibrant sunset sky reflecting off glass buildings	図 4.4
自宅	夜	Cozy home interior at night, warm lighting, comfortable furniture, large windows with a view of the starry sky, peaceful ambiance	図 4.5

図 4.1: ファミレス_昼 画像

図 4.2: 図書館_夜 画像

図 4.3: 学校_昼休み 画像

図 4.4: 街中_夕方 画像

図 4.5: 自宅_夜 画像

表 4.6: シナリオから抽出した情報

	時間	場所	登場人物の状態	テキスト情報
文 1		学校	勉強中	勉強の重要性
文 2	長時間	部屋	疲れてしまう	勉強中
文 3	そんな時		休憩中	気分が楽になる

表 4.7: ミーム素材の決定

	登場人物の状態	特徴名	ミーム名
文 1	勉強中	タイピング	PC 猫
文 2	疲れてしまう	落ち込む	しょんぼり猫
文 3	休憩中	爆睡	いびきをかく猫

4.3　考察

実験 1 では, GPT と Gemini のいずれも時間と場所に関しては同様の結果を示したが, 登場人物の状態とテキスト情報には明確な違いが見られた. GPT を使用した場合, 入力文にある情報をもとに, 登場人物の状態を「一休み中」とし, テキスト情報を「気分が楽になる」と出力している. 一方で Gemini の場合は, 勉強して一休みするという状況から登場人物が疲れていると想定したのか, 登場人物の状態を「疲れている」とし, テキスト情報で「気分転換が必要」という説明を入れている. 与えた入力文には, 読点の前後で登場人物の状態が変化していると捉えることができ, Gemini が文の前半を, GPT が文の後半を反映した出力になっていると考えられる.

実験 2 では,「ファミレス」のような直接データベースに存在していないものを使用した. GPT はこれを正確に解釈し, 他の場所や時間に対しても正確に出力している. LLM を使用しているため, 高度な検索が可能になっていることがわかる. 一方で, FAISS を使用したベクトル検索に基づくファイルパスの決定は,「ファミレス」などの存在していない単語や略称を処理できていない.

また, Stable Diffusion を使用した画像の生成については, 入力した場所と時間に対して想定通りの画像が出力され, 十分な性能を確保できていることから, GPT でのプロンプト生成が十分に機能していると考えられる.

実験 3 では, シナリオデータから時間, 場所, 登場人物の状態, テキスト情報を抽出できていることが確認できた. これをもとにしたミーム素材の決定においても, 期待される結果を得ることができた. 一方で, 時間情報の抽出は,「長時間」 や 「そんな時」 のように曖昧な表現がそのまま抽出されてしまう課題が見られた. このため, これを補完する手法が必要である. したがって, シナリオデータからの情報抽出に関して, 時間以外は適切に抽出されていることから, 明確な情報の抽出には GPT を利用する手法が十分有効である. しかし, 曖昧な要素を含んだ場合には, GPT がそのまま処理することは難しいことが示唆される.

5　まとめと今後の課題

本研究では, シナリオデータの解析を実施し, SQL 等を使用して背景画像とミーム素材を決定し, これらを使用した動画の生成という一連の流れを自動化するスクリプトを作成し, 実行した. 実験結果として, GPT と Gemini によるシナリオデータの解析が十分に可能であることが確認された. また, 背景画像とミーム素材の決定において GPT を使用した決定方法が有効であり, 提示される結果がシナリオの内容と整合性を保っていることが分かった. さらに, Stable Diffusion を使用した背景画像の生成に際しても, GPT によるプロンプト生成が十分に利用できることが確認できた. この結果から, テキストベースのデータ解析と生成モデルを組み合わせることで, 自動動画生成システムの構築が可能であることが示唆された.

今後の課題としては, 以下の点が挙げられる.

- 本研究では, 46 種類のミーム素材に対して特徴を手作業でラベル付けしているが, 十分なラベル付けができていないケースが見受けられた. また, シナリオ中の登場人物の状態や感情に適するミーム素材が存在しない場合もあり, これが素材選択の精度低下につながることがあった. ミーム素材に対して適切なラベルを付加することで, 前者の問題は解決できると考えられる. そのため, LLM を使用した自動ラベル付け手法の検討が今後の重要な課題である.

- 本研究では, もともとインターネット上に存在するミーム素材を使用している. しかし, 既存のミームだけでは補いきれない状態や表現が存在するため, これを補完するための手法が必要である. 動画生成 AI の利用も視野に入れ, 多くの人が共通で認識できるというミームの特徴を意識しながら, オリジナルのミーム生成についても検討したい. 特に, 固有の登場人物や特定の状況に特化したミームを新たに生成することで, シナリオの表現力をさらに高めることが可能である.

謝辞

本研究を進めるにあたり, 日々の研究の進捗や発表資料の作成にご指導いただき, 研究方針についても快くご相談に乗ってくださった森直樹教授には心から感謝申し上げます. また, 研究会などで貴重なご意見をいただいた岡田真助教にも深く感謝申し上げます.

さらに, 毎週の研究会において研究や発表に関するアドバイスや資料の添削など, 貴重な助言をいただいた Creation 班の先輩方, 小泉尚輝さん, 西村昭賢さん, 村上一真さん, 村田知弥さんにも心より感謝申し上げます.

最後に, 同期の皆さんには, さまざまなアドバイスをいただいただけでなく, 精神的な支えもいただきました. 環境構築や資料作成の方法について相談に乗ってくださった先輩方をはじめ, 創発ソフトウェア研究室に関わるすべての皆様に心より御礼申し上げます.

2025 年 2 月 21 日

参考文献

[1] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[3] Python Software Foundation. Python logo, n.d. Accessed: 2025-02-09.

[4] OpenAI. Openai logo, n.d. Accessed: 2025-02-09.

[5] Oracle Corporation. Mysql logo, n.d. Accessed: 2025-02-09.

[6] H. Nakao. The meme's-eye view. *Studies in Philosophy and History of Science*, 4:45–64, 2010.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[8] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

[9] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, and Y. Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

[10] G. T. et al. Gemini: A family of highly capable multimodal models, 2024.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[12] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD International Conference on Management of Data*, SIGMOD '79, p. 23–34, New York, NY, USA, 1979. Association for Computing Machinery.

[13] B. E. F. Codd. A relational model of data for large shared data banks. *M.D. computing : computers in medical practice*, 15 3:162–6, 1970.

[14] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The faiss library. 2024.

[15] W. Li, Y. Zhang, Y. Sun, W. Wang, W. Zhang, and X. Lin. Approximate nearest neighbor search on high dimensional data — experiments, analyses, and improvement (v1.0), 2016.

[16] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin eds., *Advances in Neural Information Processing Systems*, Vol. 33, pp. 6840–6851. Curran Associates, Inc., 2020.

[17] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.