

テキストデータの解析に基づく動画の生成

1 はじめに

近年、動画配信サービスが盛んになり、動画コンテンツの重要性が増している。その中には公知のミーム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人気になっている。

本研究では、テキストデータから必要な情報を抽出し、公知のミームを用いた動画を生成する手法について検討する。テキストから動画の生成が実現できれば、動画の生成に関して時間の節約、コスト削減に加えて大量のコンテンツの作成が可能になる。具体的には、テキストの解析を実施し、それを基にした背景画像とミーム素材の決定、これらの素材を使用した動画の生成という手順で進めていく。

また、現在 Google や OpenAI が動画生成 AI を発表しているが、一般公開はされていない。今回の実験で扱った猫ミームは、動画素材が有限であるため、ストーリーを入力した時に素材の中から適切なものを選択することで動画の生成が容易になる。

2 要素技術

2.1 Adobe After Effects (AE)

Adobe After Effects [1] 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、ビジュアルエフェクトを動画に加えることに利用され、映画、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。

このソフトウェアは、タイムラインベースのインターフェースを持ち、エフェクトの適用やアニメーションの作成が容易である。キーフレームを使ったアニメーションの制御が強力で、精緻な動きの調整が可能である。また、3D 空間での作業やカメラの操作もサポートしており、複雑なビジュアルエフェクトを作成できる。

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様々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。

2.2 ExtendScript

ExtendScript [2] 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。これにより、反復作業の自動化、複雑なタスクの簡略化、独自のツールやパネルの作成を容易に実行できる。

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。ExtendScript はプロのクリエイターや開発者にとって、効率的なワークフローを支援する強力な道具である。

2.3 GPT-3.5 Turbo

GPT-3.5 Turbo [3] は、OpenAI が開発した言語モデルで、GPT-3.5 の改良版である。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。

このモデルは、大規模なトレーニングデータと豊富な言語知識を基にしており、一般的な会話から専門的なトピックまで幅広い領域で対応が可能である。文法の正確さや文脈に即した回答の提供にも優れている。

2.4 ミーム

ミーム [4] とは、インターネット上で Web サイトや SNS を通して拡散され、話題となった文章や画像、動画のことである。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、文化的に人から人へと広がる現象を指す。元々「ミーム」は、進化生物学者リチャード・ドーキンスが命名した、人から人へ広がる行動やアイデアの概念である。

3 提案手法

本研究では、テキストの解析を実施し、動画を自動生成する手法を提案する。

1. テキストから情報を抽出
2. 背景画像、猫ミーム素材に関する情報をまとめた csv ファイルを利用し、適切なファイルパスを入手

表 1: GPT-3.5 Turbo のパラメータ

temperature	0.7
max_tokens	100

3. ファイルパス, テキスト情報をもとに, テンプレートに当てはめて ExtendScript を作成, 実行

テキストからの情報の抽出とファイルパスの決定の2つに GPT-3.5 Turbo を使用した. 表 1 に実験パラメータを示す.

3.1 テキストからの情報の抽出

以下に, テキストから情報を抽出する際に用いたプロンプトを示す.

プロンプト

```
#命令
入力文を以下の形式で表現してください.
背景
時間:
場所:
登場人物の状態:
テキスト情報:
#入力文
勉強している時は、一休みすると気分が楽になる。
#条件
上記のような形式で表現してください. 背景 (時間と場所), 登場人物 (私) の状態, テキスト情報を出力してください.
時間に関して, 朝, 昼, 夜などの具体的な時間を示してください.
テキスト情報に関しては, 発言や状況などを簡潔に説明したいです. 発言がない場合は状況を適切に表現してテキスト情報を出力してください.
#例
「朝起きて、気分よくおはようと言った。」という
入力が与えられた場合,
背景
時間: 朝
場所: 自宅の寝室
登場人物の状態: 起きたばかりで気分が良い
テキスト情報: おはよう
のように出力してください. テキスト情報に関して, 上記の場合はおはようのような単語だけでお願いします.
```

表 2: 背景画像データの csv ファイル (一部抜粋)

画像コード	画像 (場所)	画像 (時間帯, 状態)	ファイルパス
1	ATM コーナー	日中	/Users/-
2	ATM コーナー	夜	/Users/-
3	ATM コーナー	夕方	/Users/-
...

表 3: 猫ミームデータの csv ファイル (一部抜粋)

猫ミームコード	猫ミーム名	特徴	ファイルパス
1	DJ 猫	ひっかく	/Users/-
2	EDM 猫	EDM, 踊る	/Users/-
3	Girlfriend 猫	ポップ, 踊る	/Users/-
...

このプロンプトを使用して, 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. この出力を正規表現で処理することで, time, location, status, text_info という変数にそれぞれ格納し, 後の操作に使用する.

3.2 ファイルパスの決定

抽出した情報をもとに動画に使用する素材を決定する. 表 2 に背景画像, 表 3 猫ミーム素材に関する情報をまとめた csv ファイルの一部を示す.

背景画像はフリー素材サイト「みんちりえ」¹から収集する. 背景画像データの csv ファイルには, 画像コード, 場所, 時間帯・状態, ファイルパスを保存している. 時間帯は [日中, 夜, 夕方], 状態は [照明の ON・OFF, カーテンやシャッターの開閉等] について記述している.

猫ミームデータ²の csv ファイルもインターネットからダウンロードし, 猫ミームコード, 猫ミームの名前, 特徴, ファイルパスを保存している.

変数に保存した情報を使用して, csv ファイルの場所や時間帯, 特徴をもとに適切なファイルパスを決定する. 以下に, ファイルパスを決定する際に用いたプロンプトを示す.

¹<https://min-chi.material.jp>

²https://sasalabo.net/2024/02/27/猫ミームとは/#google_vignette

プロンプト

#Command

Please answer in Japanese.

The location is {user_location} and the time is {user_time}.

Please compare this to the candidate list, choose the location name and time from the candidate list that you think is most appropriate, and return the file path for that location.

The candidate list is: {'', '.join(candidate_list)}

まず、事前に csv ファイルから場所、時間とファイルパスの対応関係を読み込み、そのうちの場所と時間についてリスト形式で保持する。このプロンプトでは、テキストから得た user_location(location) と user_time(time) をもとに GPT-3.5 Turbo によって候補リストの中から対応する場所と時間を選択する。こうして GPT-3.5 Turbo の応答から抽出された場所と時間を利用して、適切なファイルパスを決定する。

猫ミームデータに関しても同様の処理を実行し、ファイルパスを決定する。

3.3 ExtendScript の作成, 実行

3.1 で得たテキスト情報と 3.2 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。

4 実験結果

本実験は 3 種類の入力文を用意した。それぞれについてスクリプトを実行しその出力について考察した。また、入力文の種類に対応して実験 1, 実験 2, 実験 3 とした。入力文、抽出した情報、ファイルパスを示す。

4.1 実験 1

入力文: 外を歩いていたらひどい雨にあった。

時間: 外出時 場所: 街の中

登場人物の状態: 雨にあたっている

テキスト情報: 雨に濡れてしまった

背景画像のファイルパス: /Users/shun/Documents/みんなちりえ_データ/商店街 | みんなちりえ/商店街 (夕方).jpg

動画素材のファイルパス: /Users/shun/Documents/猫ミーム用素材/cat_meme/濡れた猫.mp4

4.2 実験 2 失敗例

入力文: 自販機で買うのはやっぱり限定ものだ。

時間: 不明 場所: 自販機の前

登場人物の状態: 買い物をしている

テキスト情報: 限定品を購入した

背景画像のファイルパス: ファイルパスが見つかりませんでした

動画素材のファイルパス: /Users/shun/Documents/猫ミーム用素材/cat_meme/お菓子食べる猫.mp4

背景画像のファイルパスの出力に失敗したため、GPT の応答を確認した。

「自販機の前と言う場所と時刻が不明なので、選択肢の中から最も適切な候補を選び、その場所と時刻に対応するファイルパスを返す必要がありますね。選択肢の中から「街中の自動販」

4.3 実験 2 成功例

入力文: 自販機で買うのはやっぱり限定ものだ。

時間: 不明 場所: 自販機の前

登場人物の状態: 買い物をする

テキスト情報: 限定品を買う

背景画像のファイルパス: /Users/shun/Documents/みんなちりえ_データ/街中の自動販売機 | みんなちりえ/街中の自動販売機 (日中).jpg

動画素材のファイルパス: /Users/shun/Documents/猫ミーム用素材/cat_meme/お菓子食べる猫.mp4

ファイルパスの出力に成功したため、GPT の応答を確認した。

「自販機の前」の時間が不明であるため、選択された場所は「街中の自動販売機 (日中)」です。この場所のファイルパスは「./images/街中の自動販売機 (日中).jpg」です」

4.4 実験 3 失敗例

入力文: 勉強している時は、一休みすると気分が楽になる。

時間: 勉強中 場所: 自宅の書斎
登場人物の状態: 勉強疲れている
テキスト情報: 一休み
背景画像のファイルパス: /Users/shun/Documents/みんなちりえ_データ/パソコン部屋 | みんなちりえ/パソコン部屋 (日中).jpg
動画素材のファイルパス: ファイルパスが見つかりませんでした

動画素材のファイルパスの出力に失敗したため、GPTの応答を確認した。

「特徴として最も適切だと思われるのは、「爆睡」です。その特徴に該当するファイルパスは次の通りです: 爆睡.wav」

4.5 実験3 成功例

入力文: 勉強している時は、一休みすると気分が楽になる。

時間: 勉強中 場所: 学習机
登場人物の状態: 勉強から一休みして気分が楽になる
テキスト情報: 一休み
背景画像のファイルパス: /Users/shun/Documents/みんなちりえ_データ/学校のグラウンド | みんなちりえ/学校のグラウンド (日中).jpg
動画素材のファイルパス: /Users/shun/Documents/猫ミーム用素材/cat_meme/WOP 猫.mp4

5 考察

実験1のように抽出した状態に対応する猫ミーム素材が存在すれば、生成が成功する確率が上がる。実験1の入力文には「ひどい雨にあった」という部分があり、状態を雨にあたっているとして抽出されている。また、実験2でも入力文に「自販機で」という部分があり、場所を自販機の前と適切に抽出ができていた。このことから場所や状態に関して、抽出されるもとの情報があれば成功する傾向にあるとわかる。また、不十分なテキストだと時間や場所に関して不明と返すことがあったため、適切なテキストの入力か柔軟に対応できるプロンプトが必要だと考える。

実験2で失敗した際、GPTの応答を確認するとプロンプトの内容を復唱しておりトークン数が足りていなかった。そのため、トークン数を増やすか、プロンプトの改善が必要だと考察する。実験3で失敗した際に、csvファイルの特徴に一致するものが存在したが、選択に失

敗した。特徴を画像コードや猫ミームコードに対応して出力させ、ファイルパスはPythonで選択することで改善が見込めると考える。

これらを通して、エラーが発生したのはテキストから情報を抽出する部分とファイルパスを決定する部分に集中しており、いずれもGPTを使用しているため、プロンプトの改善が有効であると考えられる。

6 まとめと今後の展望

本研究では、テキストの解析を実施し、それをもとにした背景画像と動画素材の決定、これらの素材を使用した動画の生成という流れを自動化するスクリプトを作成し、実行した。

今後の課題として、出力の安定、背景画像の自動生成、時間、場所、登場人物の状態、テキスト情報以外の柔軟な変更が挙げられる。

出力の安定に関して、与えられていない情報の補完方法の模索、素材のパスの正確な決定を実装する。また、本実験では元々インターネット上に存在する背景画像を採用したが、文章から抽出した時間や場所の情報を基に、背景画像の自動生成をしたいと考えている。また、テンプレートで固定していた位置やサイズなどを柔軟に対応できるように検討していきたい。

参考文献

- [1] Adobe. After effects と premiere pro の違い: 映画制作に最適なツールの選び方. <https://www.adobe.com/jp/creativecloud/video/premiere-pro-vs-after-effects.html>. Accessed: 2024-05-20.
- [2] MrDogeBro. Extendscript overview. <https://extendscript.docsforadobe.dev/introduction/extendscript-overview.html#:~:text=ExtendScript%20provides%20a%20common%20scripting,DOM%20and%20the%20messaging%20framework.,2016>.
- [3] OpenAI. Models. <https://platform.openai.com/docs/models/gpt-4-turbo-and-gpt-4>. Accessed: 2024-06-25.
- [4] ソフトバンクニュース. 【インターネットミーム】1分でわかるキーワード. https://www.softbank.jp/sbnews/entry/20230420_02, 2023.