

# Script Generation Pipeline

## ▼ Ý tưởng

Tôi có một ý tưởng như vậy về công cụ hỗ trợ tạo kịch bản tận dụng gemini.

Ví dụ người dùng sẽ nhập vào mô tả về yêu cầu của họ, từ đó gemini sẽ phân tích yêu cầu / mô tả đó để cho ra cách filter trên tập dữ liệu và sau đó sẽ xem bộ dữ liệu đã được lọc để rút ra insight phục vụ cho việc viết kịch bản.

Ví dụ người dùng nhập: "tôi muốn làm video quảng bá cho món bánh tráng chấm của nhà tôi..."

Thì gemini sẽ phân tích và gợi ý các trường để filter dữ liệu, ví dụ như này:

```
{  
  "category": ["Review sản phẩm ăn uống"],  
  "main_content_focus": ["Sản phẩm ăn uống"],  
  "structure_style": ["Hướng dẫn", "Review trải nghiệm"]  
}
```

xong nó sẽ sinh thêm một số trường khác nếu có thể, ví dụ như nó nghĩ pacing nên chậm...

Xong nó lọc và xem các trường content như là hook, cta, highlight...

Và dùng nó để sinh kịch bản cuối cùng.

## ▼ Tóm tắt pipeline ý tưởng:

### Bước 1: Người dùng nhập mô tả mục tiêu

Ví dụ: "Tôi muốn làm video quảng bá cho món bánh tráng chấm của nhà tôi, cách nói chuyện gần gũi, có hướng dẫn cách ăn, giọng điệu từ tốn."

### Bước 2: Gemini phân tích và đề xuất cấu hình lọc

```
{  
  "category": ["Review sản phẩm ăn uống"],  
  "main_content_focus": ["Sản phẩm ăn uống"],  
  "structure_style": ["Hướng dẫn", "Review trải nghiệm"],  
  "tone_of_voice": ["Thân mật"],  
  "pacing": ["Chậm"]  
}
```

→ Có thể sinh thêm nếu mô tả đủ rõ: `hook_type`, `cta_type`, v.v.

### Bước 3: Filter dữ liệu đã annotate theo cấu hình trên

→ Lấy ra những video phù hợp với mục tiêu

### Bước 4: Trích xuất:

- `hook_content` phổ biến
- `cta_content` hay dùng
- `highlight_phrases` ấn tượng

→ Có thể dùng clustering hoặc sort theo tần suất

### Bước 5: Sinh kịch bản gợi ý

Dựa trên insight từ dữ liệu lọc được:

- Hook gợi ý: "Bạn đã thử kiểu bánh tráng chấm thế này chưa?"
- Body: hướng dẫn ngắn gọn + chia sẻ trải nghiệm cá nhân
- CTA: "Nhớ follow để mình chia sẻ thêm công thức chấm nhé!"

### ✅ Điều bạn đang tạo:

- Là **content engine** cho TikTok ẩm thực
- Giao diện đơn giản: người dùng chỉ cần gõ mô tả tự nhiên
- Backend "hiểu" nhu cầu → gợi ý filter + trích insight + sinh script

### ▼ Vấn đề: yêu cầu quá chi tiết

### ✅ Ví dụ cụ thể

Người dùng mô tả:

"Mình muốn làm video giới thiệu món ăn, giọng thân mật, hướng dẫn từ tốn"

### Cấu hình Gemini sinh ra:

```
json
CopyEdit
{
  "filters_strict": {
    "category": ["Review sản phẩm ăn uống"],
    "tone_of_voice": ["Thân mật"]
  }
}
```

```

},
"filters_flexible": {
  "structure_style": ["Hướng dẫn"],
  "pacing": ["Chậm"]
}
}

```

## Diễn giải quy trình:

### 1. Bước đầu tiên:

- Lọc video có `category=Review`, `tone=Thân mật` → ra 5 video

### 2. Thấy ít quá → bỏ bớt điều kiện flexible:

- Chấp nhận video chỉ có `category=Review`, `tone=Thân mật` nhưng `pacing` khác
- Hoặc `structure_style` không hẳn là "Hướng dẫn"

→ Có thể nới lên 15–20 video

## Kết luận:

- Flexible không phải là điều kiện bắt buộc**
- Nó giúp **mở rộng dữ liệu đầu vào**, chứ không **siết thêm**
- Là cơ chế **"fallback" thông minh** để đảm bảo có dữ liệu đủ cho việc sinh insight/script

## ▼ Các trường bổ sung

Trường mới	Mục tiêu	Dạng	Suy ra từ đâu?
dish_names	Lọc món ăn/sản phẩm cụ thể	List[str]	Mention rõ trong transcript
brand_or_place	Nhận diện quán/thương hiệu	List[str]	"Ở tiệm bánh X...", "mua ở..."
personas_in_video	Biết có ai xuất hiện, đối tượng	List[str]	"Minh với mẹ đi ăn", "anh bán..."
language_style_keywords	Cụm từ mang style đặc trưng	List[str]	"ăn xiu luôn", "ngon dã man"...
script_template_type	Xác định script có thể tái dùng theo format nào	List[str]	Suy ra từ cấu trúc + hook

 **Gợi ý cách dùng** `script_template_type`

Giống như `structure_style`, nhưng hướng tới tái sử dụng trong viết script:

Giá trị có thể có	Ý nghĩa
Giới thiệu món → cảm nhận → CTA	Format phổ biến trong review ăn uống
Hướng dẫn từng bước → lời khuyên	Video dạy cách ăn, cách chọn, cách dùng món
Câu chuyện cá nhân → món ăn	Hook theo kể chuyện, chuyển sang review

## ✅ Đề xuất các trường phân tích "hữu ích – không cảm tính":

Trường	Ý nghĩa cụ thể	Dữ liệu đầu ra (dạng)
<code>hook_analysis</code>	Mô tả cách mở đầu (gây tò mò? nhắm vào ai? dùng phrasing gì?)	string
<code>cta_analysis</code>	CTA xuất hiện lúc nào? Có rõ ràng không? Nhắm hành động gì?	string
<code>structure_pattern</code>	Diễn giải cấu trúc triển khai: theo thứ tự gì, nhấn mạnh gì	string (format: A → B → C)
<code>tone_analysis</code>	Giọng điệu + nhịp điệu + level cảm xúc	string
<code>style_highlights</code>	Phong cách nổi bật có thể bắt chước: hài hước, đều đều, story	list[string]

## ▼ 🧩 Các loại insight nên trích xuất (từ transcript gốc)

Nhóm	Trường phân tích cụ thể	Ghi chú
🎯 Hook	<code>hook_goal</code> – Hook nhắm tới cảm xúc nào? tò mò, shock, trải nghiệm?	Label
	<code>hook_phrasing_style</code> – Cấu trúc hook (câu hỏi? khẳng định?)	Label
	<code>hook_viral_keywords</code> – Các cụm từ "viral" mở đầu	List[str]
🍽️ Nội dung	<code>dish_or_product</code> – Món/quán/sản phẩm được đề cập	List[str]
	<code>eating_instruction_present</code> – Có hướng dẫn cách ăn không?	Bool
	<code>structure_pattern</code> – Trình tự nội dung gốc (giới thiệu → cảm nhận...)	String
	<code>flavor_keywords</code> – Từ miêu tả vị/texture	List[str]
	<code>personal_emotion_keywords</code> – Cảm xúc người nói (phấn khích, thất vọng...)	List[str]
📢 CTA	<code>cta_present</code> – Có CTA không?	Bool
	<code>cta_type</code> – Follow, Like, Comment, Mời trải nghiệm...	List[str]
	<code>cta_phrases</code> – Câu gọi hành động thực tế	List[str]

Nhóm	Trường phân tích cụ thể	Ghi chú
✨ Phong cách	<code>tone_of_voice</code> – Giọng điệu tổng thể	Label
	<code>pacing</code> – Nhịp điệu (nh nhanh, chậm, thay đổi)	Label
	<code>style_keywords</code> – Các cụm phong cách (ăn xiu, dã man, đỉnh của chóp)	List[str]
🧱 Tái sử dụng	<code>script_template_type</code> – Format tổng thể (Hook → Review → CTA)	String
	<code>template_potential</code> – Có dùng làm khung tái sinh không?	Bool