

دوره آموزش تولید محتوای متنی با استفاده از هوش مصنوعی مولد

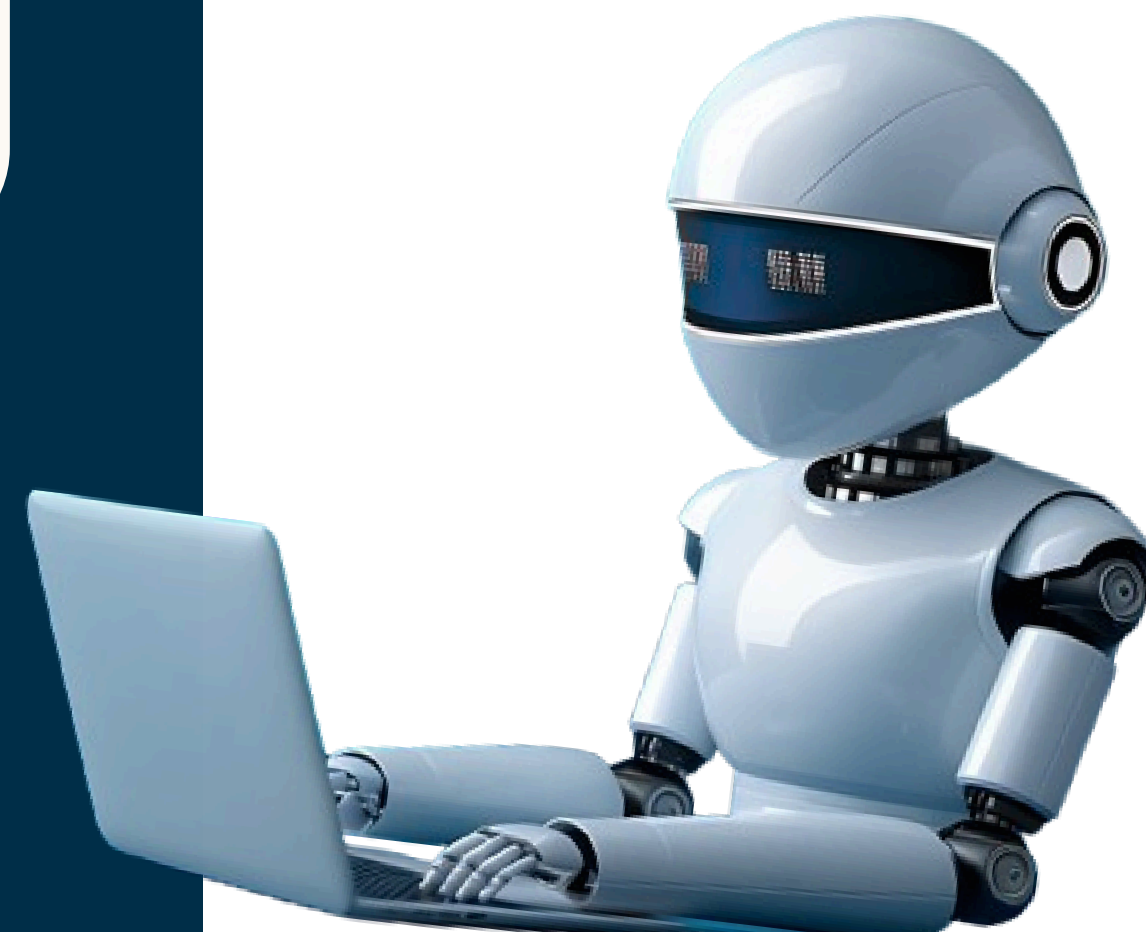
فصل پنجم

Prompt Patterns - بخش سوم

مرکز آموزش فناوری اطلاعات شهرستان گرگان
مهندس مصطفی صادقی



سازمان آموزش فنی و حرفه‌ای کشور



Ask for Input Pattern



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چيست Ask for Input Pattern؟

وقتی به یک مدل زبانی (مثل GPT) یک سری قوانین، نقش‌ها، مراحل یا الگوهای رفتاری می‌دهیم، معمولاً مدل همان لحظه شروع می‌کند به اعمال آن قوانین و تولید خروجی حتی اگر هنوز ورودی اصلی را به آن نداده باشیم.
برای مثال:
اگر بگوییم:

- هر وقت ازت چیزی خواستم، سه روش جایگزین بده
- هر روش را با جزئیات توضیح بده
- و یک پرامپت بساز
- مدل ممکن است همان لحظه از خودش یک "task" بسازد و شروع کند خروجی تولید کردن—چیزی که اصلاً نخواستیم.

Ask for Input Pattern یک تکنیک برای جلوگیری از این رفتار است.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

هدف الگو

این الگو تضمین می‌کند که مدل:

- قوانین را دریافت کند

- آن‌ها را بفهمد

- اما اجرا نکند

- تا وقتی که کاربر «ورودی اصلی» را نفرستاده است.

این الگو در واقع به مدل می‌گوید:

«قواعد را یاد بگیر، اما فعلاً هیچ کاری نکن. فقط از من بپرس ورودی چیست.»



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چرا لازم است؟

- زیرا مدل‌ها وقتی قوانین را می‌بینند:
- شروع می‌کنند به اجرای قوانین
- ورودی‌ای که هنوز ندادیم را حدس می‌زنند
- یا گفتگو را به هم می‌ریزند
- یا حجم زیادی خروجی بی‌مصرف تولید می‌کنند



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

این الگو چگونه کار می کند؟

در پایان پرامپت قوانین مان، یک دستور بسیار ساده اضافه می کنیم:

• Ask me for the first task .

• Ask me for the input .

• Ask me for the first idea .

• Ask me for the next item .

و این جمله سیگنال توقف برای مدل است.

مدل متوجه می شود که:

• باید قوانین را بپذیرد

• اما اجازه ندارد خروجی تولید کند

• باید فقط یک سؤال بپرسد: ورودی چیست؟

Combining Patterns



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

ایده اصلی چیست؟

وقتی پرامپت‌های پیچیده می‌سازیم، معمولاً یک پترن به‌تنهایی کافی نیست.
به‌جای اینکه فقط بپرسیم:

«کدام پترن این مشکل را حل می‌کند؟»

باید بپرسیم:

«هر پترن کدام بخش از مشکل را پوشش می‌دهد؟ و چگونه می‌توانم چند پترن را ترکیب کنم؟»



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چرا ترکیب پترن ها مهم است؟

- وقتی پرامپت های پیچیده می سازیم، معمولاً پرامپت های پیچیده معمولاً چند مرحله یا چند نیاز دارند.
- هیچ پترنی همه نیازهای یک مسئله پیچیده را پوشش نمی دهد.
- با ترکیب پترن ها می توان به یک پرامپت بسیار قدرتمند و کنترل شده رسید.
- در عمل، بیشتر پرامپت های حرفه ای همین طور ساخته می شوند.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

مثال: ترکیب Ask for Input + Alternative Approaches

Ask for Input: مدل قوانین را می پذیرد، ولی صبر می کند تا کاربر ورودی بدهد.
Alternative Approaches: مدل چند راه حل یا چند نوع پاسخ ارائه می دهد.

وقتی این دو را ترکیب می کنیم، پرامپتی می سازیم که:

- ابتدا یک سری قوانین را بیان می کنیم.

- مدل آن را می پذیرد.

- بعد از کاربر ورودی می خواهد.

سپس هنگام پاسخ، چند راه حل یا چند زاویه مختلف ارائه می کند.

این یعنی هم کنترل ورودی داریم، هم خروجی غنی تر می شود.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چگونه باید این کار را انجام داد؟

برای ساخت یک پرامپت پیچیده باید:
(الف) مشکل را به بخش‌های کوچک تقسیم کنیم
مثلاً:

- باید نقش تعریف کنم → Persona Pattern
- باید سؤال بپرسم → Ask for Input
- باید چند پاسخ ممکن بخواهم → Alternatives Pattern
- باید ساختار نتیجه را مشخص کنم → Output Structuring Pattern

(ب) برای هر بخش یک پترن مناسب پیدا کنیم
هر پترن یک سری جملات/دستورالعمل دارد.

(ج) جملات پترن‌ها را کنار هم بچینیم
بعضی وقت‌ها ترکیب ساده است:
هر پترن چند جمله دارد، همه را کنار هم قرار می‌دهیم.

بعضی وقت‌ها ترکیب ظریف‌تر است:
مثلاً Ask for Input همیشه باید در پایان پرامپت قرار بگیرد.
اگر وسط پرامپت بیاید، درست کار نمی‌کند.

بنابراین پترن‌ها علاوه بر محتوا، ترتیب هم دارند.

Outline Expansion Pattern



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چيست Outline Expansion Pattern؟

مدل های زبانی مثل ChatGPT:

- ورودی محدود دارند (نمی توانید 100 هزار صفحه به مدل بدهی)
- خروجی محدود دارند (نمی تواند یک کتاب 300 صفحه ای را یک جا تولید کند)
- در یک مکالمه طولانی، بخشی از کانتکست را فراموش می کنند

پس وقتی می خواهیم یک کار بزرگ انجام دهیم (نوشتن کتاب، طراحی سیستم، تحلیل بزرگ، تولید یک دوره آموزشی، ساخت نرم افزار)، باید کار را به قطعات کوچک بشکنیم و در عین حال بدانیم که هر قطعه کجای ساختار کلی قرار دارد.

اینجاست که Outline Expansion Pattern وارد می شود.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چيست Outline Expansion Pattern؟

ایده این الگو، تولید یک ساختار سلسله‌مراتبی (مانند یک فهرست مطالب یا Outline) به عنوان نقشه راه و چارچوب کلی پروژه است. این ساختار به ما کمک می‌کند:

- درک کلان: از ابتدا دید کلی از پروژه داشته باشیم.
- تقسیم کار: پروژه را به بخش‌های مجزا و تودرتو تقسیم کنیم.
- تولید متمرکز: روی هر بخش به صورت جداگانه و با تمرکز کار کنیم.
- مونتاژ آسان: به دلیل ساختار شفاف، بدانیم هر تکه تولیدشده دقیقاً متعلق به کجای پازل اصلی است.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

این پترن چگونه ساخته می شود؟

در این پترن معمولاً چند الگوی دیگر هم ترکیب می شود:
Persona Pattern .

مدل را وادار می کنیم نقش **Outline Expander** را بازی کند.

Ask-for-Input Pattern .

مدل بعد از هر خروجی، می پرسد مرحله بعد کدام است.

Tail Generation Pattern .

مدل را مجبور می کنیم گفت و گو را با یک سؤال انتهایی تمام کند تا تعامل ادامه یابد.



این پترن چگونه ساخته می شود؟

- Act as an outline expander.
- Generate a bullet-point outline based on the input I give you.
- Then ask me which bullet point you should expand next.
- When I tell you the number, generate a more detailed outline for that bullet only.
- End every response by asking: "Which bullet point should I expand next?"
- Ask me for the initial topic.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Menu Actions Pattern



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چيست Menu Action Pattern؟

Menu Actions Pattern یک الگوی پیشرفته در مهندسی پرامپت است که به کاربر اجازه می‌دهد مجموعه‌ای از «دستورات از پیش تعریف شده» را در ابتدای یک مکالمه مشخص کند؛ به گونه‌ای که مدل زبانی در طول همان مکالمه بتواند این دستورات را به صورت کوتاه، مستقیم و بدون تکرار پرامپت‌های طولانی اجرا نماید.

به بیان دیگر، این الگو امکان ایجاد یک سیستم اکشن‌های قابل فراخوانی را فراهم می‌کند که مشابه منوها و فرمان‌های یک نرم‌افزار عمل می‌کنند.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چیست Menu Action Pattern؟

هدف اصلی Menu Actions عبارت است از:

- کاهش نیاز به تکرار پرامپت‌های پیچیده و طولانی
 - استانداردسازی یک سری عملیات قابل تکرار
 - تسریع جریان کاری کاربر در مکالمات طولانی
 - ایجاد یک ساختار منطقی مشابه «برنامه کوچک» در درون گفت‌وگو
 - ایجاد قابلیت اشتراک‌گذاری دستورهای استاندارد در سطح سازمان یا تیم
- این الگو مخصوص زمانی است که کاربر مکرراً عملیات مشابهی را در یک چت انجام می‌دهد.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چیست Menu Action Pattern؟

این پترن از طریق تعریف مجموعه‌ای از اکشن‌ها (Actions) عمل می‌کند. هر اکشن در حقیقت یک «قالب فرمان» است که:

- یک نام یا کلیدواژه دارد.
- یک ساختار ورودی دارد.
- یک عملکرد تعریف شده دارد.
-

پس از تعریف اکشن‌ها، مدل موظف است هر بار که کاربر فرمان موردنظر را به صورت کوتاه وارد کرد، همان عملیات را اجرا کند.

نمونه ساده:

اگر فقط شماره یک بولت‌پوینت نوشته شود → مدل آن بخش را گسترش می‌دهد.
اگر ساختار فرمانی مانند `<paragraphs> <write> <bullet>` نوشته شود → مدل برای آن بخش متن تولید می‌کند.



نحوه ی طراحی Menu Action Pattern

To use this pattern, your prompt should make the following fundamental contextual statements:

- Whenever I type: X, you will do Y.
- (Optional, provide additional menu items) Whenever I type Z, you will do Q.
- At the end, you will ask me for the next action.

You will need to replace "X" with an appropriate pattern, such as "estimate <TASK DURATION>" or "add FOOD". You will then need to specify an action for the menu item to trigger, such as "add FOOD to my shopping list and update my estimated grocery bill".



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Fact Check List Pattern



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Fact Check List Pattern چیست؟

Fact Checklist Pattern یک الگو در مهندسی پرامپت است که هدف آن کمک به تشخیص و اعتبارسنجی اطلاعاتی است که مدل زبانی در متن خروجی خود تولید می‌کند. این الگو به‌ویژه زمانی مفید است که:

- مدل حجم زیادی از متن تولید می‌کند.
- متن بسیار طبیعی و متقاعدکننده به نظر می‌رسد.
- احتمال وجود اشتباه، فرضیات غلط، یا اطلاعات ساختگی (hallucination) وجود دارد.

در این مواقع، انسان ممکن است به دلیل قانع کننده بودن متن، به صورت ناخودآگاه تمام خروجی را صحیح تصور کند. پترن Fact Checklist سعی می‌کند این مشکل را حل کند.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Fact Check List Pattern چیست؟

• **دستور به مدل:** به مدل دستور داده می‌شود که پس از تولید هر متن، لیستی از فکت‌های بنیادین (Fundamental Facts) موجود در آن متن را نیز ارائه دهد. این فکت‌ها، حقایق کلیدی‌ای هستند که اگر نادرست باشند، اعتبار کل متن زیر سؤال می‌رود.

• **جایگاه فهرست:** این لیست معمولاً در انتهای خروجی متن اصلی اضافه می‌شود.

• **استفاده کاربر از لیست:** کاربر می‌تواند از این لیست به دو روش استفاده کند:

1. **به عنوان حداقل‌ترین وظیفه بررسی:** کاربر می‌تواند مستقیماً هر فکت در لیست را (با جستجوی خارجی، دانش شخصی یا منابع معتبر) بررسی کند.

2. **به عنوان ابزار مقایسه:** کاربر متن اصلی را با لیست فکت‌ها مقایسه می‌کند. اگر جمله یا ادعای مهمی در متن اصلی وجود دارد که در لیست فکت‌ها نیست، باید آن را نیز به لیست بررسی خود اضافه کند. این مقایسه، خود فرآیند تفکر انتقادی درباره محتوا را فعال می‌کند.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

مزایای این الگو:

- **تغییر دیدگاه:** این الگو به ما یادآوری می‌کند که مدل‌های زبانی در درجه اول مولد متن هستند، نه بانک دانش بی‌خطا. ما باید مسئولیت نهایی صحت اطلاعات را بپذیریم.
- **مقابله با واقع‌نمایی متن:** از آنجا که خروجی این مدل‌ها بسیار روان و قانع‌کننده است، ممکن است به راحتی صحت آن را بپذیریم. این چک‌لیست یک توقف‌گاه اجباری برای تفکر انتقادی ایجاد می‌کند.
- **سیستم‌اتیک کردن فرآیند بررسی:** به جای تکیه بر حافظه یا خوانش پراکنده، یک فرآیند ساختاریافته برای استخراج و تأیید اطلاعات ارائه می‌دهد.
- **بهبود کیفیت کار:** با تمرکز بر بررسی فکت‌های کلیدی، احتمال انتشار اطلاعات نادرست کاهش می‌یابد.

محدودیت‌ها:

- **ضمانت قطعی نیست:** خود مدل ممکن است در استخراج کامل و دقیق همه فکت‌های کلیدی شکست بخورد یا برخی مفروضات پنهان را نادیده بگیرد.
- **نیازمند تلاش کاربر است:** این الگو بار بررسی را به عهده کاربر می‌گذارد. تنها در صورت پیگیری و تأیید کاربر، مؤثر خواهد بود.



مثال کاربردی

To use this pattern, your prompt should make the following fundamental contextual statements:

- Generate a set of facts that are contained in the output
- The set of facts should be inserted at POSITION in the output
- The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect

You will need to replace POSITION with an appropriate place to put the facts, such as "at the end of the output".

Examples:

Whenever you output text, generate a set of facts that are contained in the output. The set of facts should be inserted at the end of the output. The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Tail Generation Pattern



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

چيست Tail Generation Pattern؟

Tail Generation Pattern یک الگوی پرامپت نویسی است که برای حفظ قوانین، اهداف و بافت یک گفت و گوی طولانی با مدل زبانی استفاده می شود.

مشکل اصلی: مدل ها در مکالمات طولانی فراموش می کنند که قرار بوده چه کاری انجام دهند. الگوهای رفتاری، قوانین و اهدافی که تعریف کرده ای ممکن است بعد از چند رفت و برگشت محو شود.

راه حل؟

- اینکه در انتهای هر پاسخ مدل، یک "Tail" تولید شود؛ یک جمله ی ثابت یا نیمه ثابت که:
- قوانین بازی را یادآوری می کند،
- هدف فعلی را بازگو می کند،
- مدل را مجبور می کند خودش قوانین را دوباره زنده کند حتی اگر فراموش کرده باشد.

این Tail مثل یک "checkpoint" در پایان هر پیام است که حافظه کوتاه مدت مدل را دوباره شارژ می کند.



Tail دقیقاً چه می‌کند؟

:Tail

- دستورالعمل‌های کلیدی را بازخوانی می‌کند.
- هدف تعامل را خلاصه می‌کند.
- مدل را مجبور می‌کند از "Ask-for-input Pattern" استفاده کند یعنی در پایان سؤال بپرسد: قدم بعدی چیست؟ ورودی بعدی چیست؟
- مدل را "به ریل" برمی‌گرداند اگر در چند پیام قبل از مسیر خارج شده باشد.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation



مثال کاربردی

To use this pattern, your prompt should make the following fundamental contextual statements:

- At the end, repeat Y and/or ask me for X.

You will need to replace "Y" with what the model should repeat, such as "repeat my list of options", and X with what it should ask for, "for the next action". These statements usually need to be at the end of the prompt or next to last.

Examples:

- Act as an outline expander. Generate a bullet point outline based on the input that I give you and then ask me for which bullet point you should expand on. Create a new outline for the bullet point that I select. At the end, ask me for what bullet point to expand next. Ask me for what to outline.
- From now on, at the end of your output, add the disclaimer "This output was generated by a large language model and may contain errors or inaccurate statements. All statements should be fact checked." Ask me for the first thing to write about.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Semantic Filter Pattern



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

Semantic Filter Pattern چیست؟

Semantic Filter الگویی در طراحی پرامپت است که به مدل دستور می‌دهد یک متن ورودی را نگه دارد اما بخش‌هایی از آن را براساس «معنا» حذف یا اصلاح کند. این پترن برخلاف فیلترهای ساده‌ی مبتنی بر کلمه، بر **درک معنایی مدل** تکیه دارد؛ یعنی مدل باید بفهمد چه چیزی از نظر مفهومی، محتوای ناخواسته محسوب می‌شود.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation

سازوکار Semantic Filter Pattern

Semantic Filter از سه مؤلفه اصلی تشکیل می‌شود:

۱. فرمان پایه: **Filter this information ...**

با واژه‌های کلیدی Filter / Remove / Preserve مدل را وادار می‌کنیم که بداند:

ما به دنبال تولید محتوای جدید نیستیم؛

بلکه حفظ متن با حذف بخش‌های ناخواسته هستیم.

۲. قوانین معنایی (Semantic Rules)

این بخش مهم‌ترین جزء است:

اینجا مشخص می‌کنیم چه چیزی باید حذف شود و چرا.

مثال‌ها:

• remove all dates

• remove information revealing diabetes

• remove confidential corporate metrics

• remove personal identifiers

این قوانین باید معنایی باشند، نه سطحی.

مثلاً: «حذف هر چیزی که نشان دهد فرد دیابت دارد» یک قانون معنایی است، نه یک کلمه کلیدی ثابت.

۳. بازنویسی حداقلی (Minimal Rewrite)

پس از حذف، متن ممکن است ناقص، ناخوانا یا عجیب شود.

بنابراین معمولاً می‌گوییم:

rewrite as little as possible to maintain coherence

یعنی حداقل ممکن متن را بازنویسی کن، فقط در حدی که متن بعد از حذف‌ها همچنان منسجم، طبیعی و قابل فهم باقی بماند.



مثال کاربردی

To use this pattern, your prompt should make the following fundamental contextual statements:

- Filter this information to remove X

You will need to replace "X" with an appropriate definition of what you want to remove, such as. "names and dates" or "costs greater than \$100".

Examples:

Filter this information to remove any personally identifying information or information that could potentially be used to re-identify the person.

Filter this email to remove redundant information.



Input Pattern



Combining Patterns



Outline Expansion



Menu Action



Fact Check List



Tail Generation