VIP Cheatsheet:
ترانسفورمرها و مدل‌های زبانی بزرگ

افشین عمیدی و شروین عمیدی
ترجمه توسط امیر ضیائی

۲۲ اردیبهشت ۱۴۰۴

این چیت‌شیت ویژه، مروری کلی بر محتوای کتاب *"Super Study Guide: Transformers & Large Language Models"* ارائه می‌دهد – کتابی شامل حدود ۶۰۰ نگاره در ۲۵۰ صفحه که مفاهیم زیر را به‌صورت عمیق تحلیل می‌کند. اطلاعات بیشتر در `https://superstudy.guide` در دسترس است.

۱ بنیادها

۱.۱ توکن

❐ **تعریف** – توکن کوچک‌ترین واحد پردازش متن است، مثل کلمه، زیر کلمه یا کاراکتر و از یک دایره واژه مشخص گرفته می‌شود.

نکته: توکن `[UNK]` برای نمایش بخش‌های ناشناختهٔ متن به‌کار می‌رود، و توکن `[PAD]` برای پر کردن فضاهای خالی جهت یکسان‌سازی طول دنبالهٔ ورودی استفاده می‌شود

❐ **توکن ساز** – توکن‌ساز T متن را بر اساس سطح دلخواهی از جزئیات به توکن‌ها تقسیم می‌کند.

انواع اصلی توکن‌سازها عبارت‌اند از:

نوع	مزایا	معایب	نگاره
واژه	• تفسیر آسان • دنباله کوتاه	• دایره واژه بزرگ • فاقد توانایی در پردازش واژگان متنوع	تدی خرس
زیرواژه	• ریشه‌های واژه‌ها به‌کار گرفته می‌شوند • تعبیه‌های قابل‌تفسیر	• افزایش طول دنباله • فرایند توکنیزاسیون پیچیده‌تر است	##ی تد خرس
کاراکتر بایت	• مشکل واژه‌های خارج از دایره واژه وجود ندارد • دایره واژه کوچک باقی می‌ماند	• طول دنباله بسیار بیشتر است • تفسیر الگوها دشوار است چون اطلاعات خیلی جزئی هستند	ی د ت س ر خ

نکته: *BPE (Byte-Pair Encoding)* و *Unigram* از توکن‌سازهای رایج در سطح زیرواژه‌ای هستند.

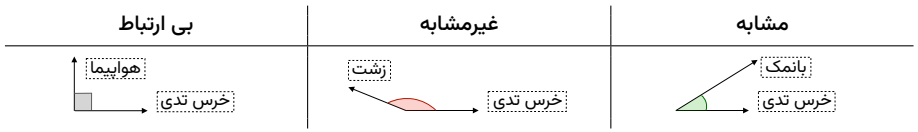
۲.۱ بردارهای تعبیه

❐ **تعریف** – تعبیه، نمایش عددی یک عنصر است (مثلاً یک توکن یا جمله) که با یک بردار $x \in \mathbb{R}^n$ مشخص می‌شود.

❐ **شباهت** – شباهت کسینوسی بین دو توکن t_1 و t_2 با فرمول زیر اندازه‌گیری می‌شود:

$$\text{شباهت}(t_1, t_2) = \frac{t_1 \cdot t_2}{||t_1||\,||t_2||} = \cos(\theta) \in [-1, 1]$$

زاویهٔ θ معیاری برای سنجش شباهت میان دو توکن است:

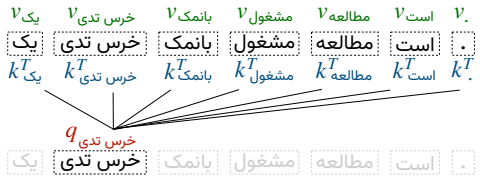


نکته: روش‌های *ANN (Approximate Nearest Neighbors)* و *LSH (Locality Sensitive Hashing)* شباهت را در پایگاه‌های داده بزرگ به‌صورت تقریبی و بهینه محاسبه می‌کنند.

۲ ترانسفورمرها

۱.۲ توجه

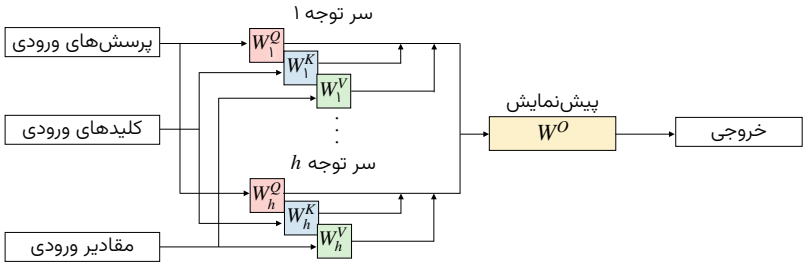
❐ **فرمول** – با داشتن پرسش q، می‌خواهیم بدانیم این پرسش باید نسبت به کدام کلید k با توجه به مقدار مرتبط v «توجه» کند.



توجه را می‌توان به‌طور کارآمد با استفاده از ماتریس‌های Q، K، V که به‌ترتیب شامل پرسش‌ها q، کلیدها k و مقادیر v هستند، همراه با بُعد کلیدها d_k، محاسبه کرد:

$$\text{توجه} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

❐ **توجه چندسَر** – لایهٔ توجه چندسَری (***M**ulti-**H**ead **A**ttention*, MHA) توجه را در چند سر موازی محاسبه کرده و خروجی نهایی را به فضای خروجی نگاشت می‌دهد.

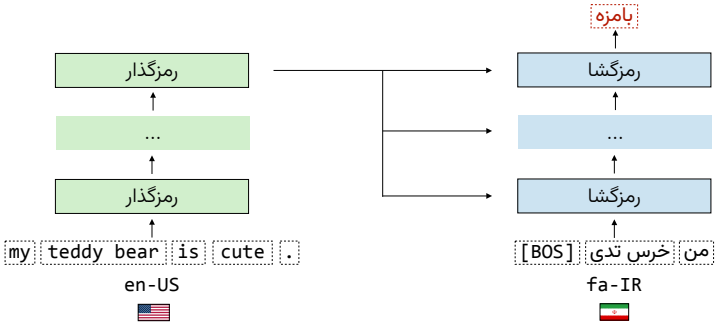


این ساختار شامل h سر توجه و ماتریس‌های W^Q، W^K، W^V برای تولید Q، K و V از ورودی است. نگاشت خروجی با ماتریس W^O انجام می‌شود.

نکته: دو نوع متداول به‌نام توجه با پرسش گروهی *(Grouped-Query Attention, GQA)* و توجه چندپرسشی *(Multi-Query Attention, MQA)* برای کاهش پیچیدگی محاسباتی، کلیدها و مقادیر را بین سرها به‌اشتراک می‌گذارند.

۲.۲ معماری

❐ **نمای کلی** – ترنسفورمر مدلی کلیدی مبتنی بر توجه خودکار است که از ترکیب رمزگذار و رمزگشا ساخته شده است. رمزگذار تعبیه‌های معنایی از ورودی تولید می‌کند و رمزگشا از آن‌ها برای پیش‌بینی توکن بعدی بهره می‌گیرد.

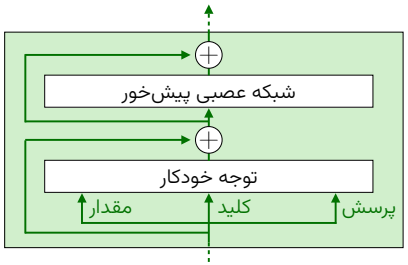


نکته: ترنسفورمر گرچه در آغاز برای ترجمه طراحی شد، اکنون به‌صورت گسترده در کاربردهای مختلف به‌کار می‌رود.

❐ **اجزا** – رمزگذار و رمزگشا دو مؤلفهٔ کلیدی ترنسفورمر با نقش‌های مجزا هستند:

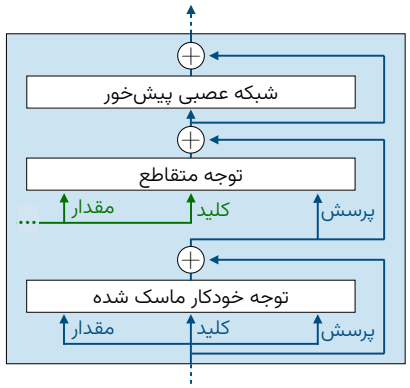
رمزگذار

تعبیه‌های رمزگذار نمایانگر معنای ورودی هستند



رمزگشا

تعبیه‌های رمزگشا نمایانگر معنای ترکیبیِ ورودی و خروجیِ تولیدشده تا کنون‌اند

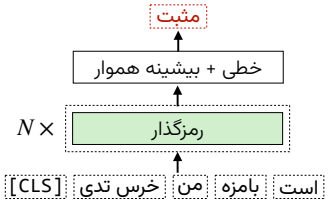


❐ **تعبیه موقعیتی** – تعبیه‌های موقعیت جایگاه توکن را در جمله مشخص می‌کنند و هم‌ابعاد با تعبیه‌های توکن هستند. این تعبیه‌ها یا به‌صورت دلخواه تعریف می‌شوند یا از داده آموزش می‌بینند.

نکته: تعبیه‌های موقعیت چرخشی *(Rotary Position Embeddings, RoPE)* q و k، اطلاعات موقعیت نسبی را وارد مدل می‌کنند.

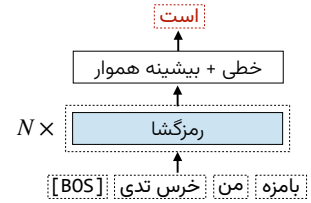
۳.۲ گونه‌ها

❐ **فقط رمزگذار** – بازنمایی‌های رمزگذار دوسویه از ترانسفورمرها (***B**idirectional **E**ncoder **R**epresentations from **T**ransformers*, BERT)، مدلی مبتنی بر ترنسفورمر بوده که از چندین رمزگذار تشکیل شده است. این مدل با دریافت متن به‌عنوان ورودی، تعبیه‌هایی معنادار تولید می‌کند که می‌توان از آن‌ها در وظایف دسته‌بندی بعدی بهره برد.



توکن `[CLS]` به ابتدای دنباله افزوده می‌شود تا بازنمایی کلی جمله را استخراج کند. تعبیهٔ آن در وظایف پایین‌دستی مانند استخراج عقاید به‌کار می‌رود.

❐ **فقط رمزگشا** – ***G**enerative **P**re-trained **T**ransformer* (GPT) مدلی خودبازگشتی مبتنی بر ترنسفورمر است که تنها از پشته‌ای از رمزگشاها تشکیل شده است. برخلاف BERT، این مدل همهٔ وظایف را به‌صورت تبدیل متن به متن حل می‌کند.



اکثر مدل‌های پیشرفتهٔ امروزی از معماری فقط رمزگشا استفاده می‌کنند، مانند سری GPT، LLaMA، Mistral، Gemma و DeepSeek.

نکته: مدل‌های رمزگذار–رمزگشا نظیر *T5* نیز خودبازگشتی هستند و با معماری فقط رمزگشا شباهت‌های زیادی دارند.

۴.۲ بهینه‌سازی‌ها

❐ **تقریب در محاسبات توجه** – محاسبات توجه دارای پیچیدگی $\mathcal{O}(n^2)$ هستند که با افزایش طول دنباله، هزینه محاسباتی آن بالا می‌رود. برای کاهش این هزینه، دو روش اصلی استفاده می‌شود:

- پراکندگی: در توجه خودکار، ارتباط بین تمام توکن‌ها در دنباله بررسی نمی‌شود، بلکه فقط بین توکن‌های مرتبط‌تر انجام می‌شود.

- تقریب با رتبه پایین: با بازنویسی فرمول توجه به‌صورت حاصل‌ضرب ماتریس‌های کم‌رتبه، محاسبات سبک‌تر می‌شود.

❐ **توجه فلش** – توجه فلش یک روش دقیق برای بهینه‌سازی محاسبات توجه است که با استفادهٔ هوشمندانه از سخت‌افزار GPU، محاسبات ماتریسی را ابتدا در حافظهٔ سریع ***S**tatic **R**andom-**A**ccess **M**emory* (SRAM) انجام می‌دهد و سپس نتایج را در حافظهٔ کندتر ولی با پهنای باند بالا یعنی ***H**igh **B**andwidth **M**emory* (HBM) ذخیره می‌کند.

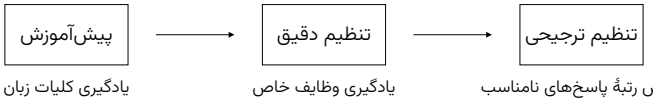
نکته: این روش در عمل باعث کاهش مصرف حافظه و افزایش سرعت محاسبات می‌شود.

۳ مدل زبانی بزرگ

۱.۳ نمای کلی

❐ **تعریف** – مدل زبانی بزرگ (***L**arge **L**anguage **M**odel*, LLM) یک مدل ترنسفورمر با توانایی‌های پیشرفتهٔ زبان طبیعی است که معمولاً میلیاردها پارامتر دارد.

❐ **مراحل آموزش** – آموزش مدل‌های زبانی بزرگ در سه مرحله انجام می‌شود: پیش‌آموزی، تنظیم دقیق، و تنظیم ترجیحی.

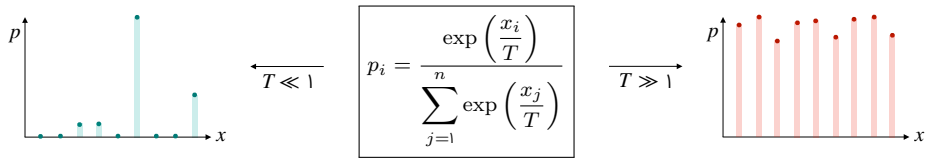


تنظیم دقیق و تنظیم ترجیحی دو رویکرد پس از آموزش هستند که هدفشان هم‌راستا کردن مدل برای انجام وظایف مشخص است.

۲.۳ پرسش‌دهی

❐ **طول زمینه** – طول زمینهٔ مدل، حداکثر تعداد توکن ورودی است و معمولاً بین ده‌ها هزار تا میلیون‌ها توکن متغیر است.

❐ **نمونه‌گیری در رمزگشایی** – توکن‌ها از توزیع احتمال p_i انتخاب می‌شوند که با دمای T کنترل می‌شود.



نکته: دمای بالا منجر به خروجی‌های خلاقانه‌تر می‌شود، در حالی که دمای پایین خروجی‌های قطعی‌تری تولید می‌کند.

❐ **زنجیره فکر** – زنجیره فکر (***C**hain-**o**f-**T**hought*, CoT) یک فرآیند استدلالی است که در آن مدل یک مسئلهٔ پیچیده را به سلسله‌ای از گام‌های میانی تقسیم می‌کند. این روش به مدل کمک می‌کند تا پاسخ نهایی صحیح را تولید کند. درخت افکار (***T**ree **o**f **T**houghts*, ToT) نسخه‌ای پیشرفته‌تر از CoT است.

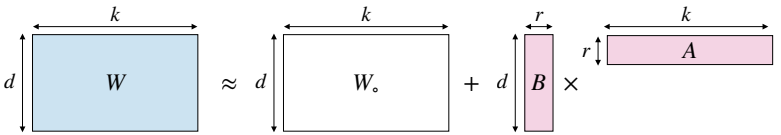
نکته: خود سازگاری روشی است که پاسخ‌ها را در مسیرهای استدلال زنجیره‌ی فکر تجمیع می‌کند.

۳.۳ تنظیم دقیق

❐ **تنظیم دقیق تحت نظارت** – تنظیم دقیق تحت نظارت (***S**upervised **F**ine**T**uning*, SFT) یک روش پس‌ازپیش‌آموزش است که رفتار مدل را با وظیفه نهایی هم‌راستا می‌کند. این روش بر جفت‌های ورودی–خروجی باکیفیت و هم‌راستا با آن وظیفه تکیه دارد.

نکته: اگر داده‌های *SFT* دربارهٔ دستورالعمل‌ها باشند، این مرحله «تنظیم بر اساس دستورالعمل» نامیده می‌شود.

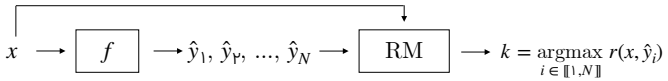
❐ **تنظیم دقیق با بهره‌وری در تعداد پارامترها** – تنظیم دقیق با بهره‌وری در تعداد پارامترها (***P**arameter-**E**fficient **F**ine**T**uning*, PEFT) دسته‌ای از روش‌ها است که برای اجرای کارآمد SFT به کار می‌روند. به‌طور خاص، تطبیق کم‌رتبه (***Lo**w-**R**ank **A**daptation*, LoRA) با ثابت نگه‌داشتن ماتریس وزن پیش‌آموزش‌دیده $W_\circ$ و یادگیری ماتریس‌های کم‌رتبه A و B ، وزن‌های قابل آموزش W را تقریب می‌زند:



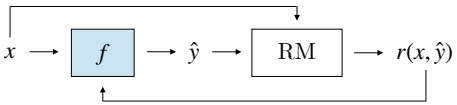
نکته: سایر تکنیک‌های *PEFT* شامل تنظیم پیشوند و افزودن لایهٔ آداپتور هستند.

۴.۳ تنظیم ترجیحات

❐ **مدل پاداش** – مدل پاداش (***R**eward **M**odel*, RM) مدلی است که پیش‌بینی می‌کند خروجی $\hat{y}$ تا چه حد با رفتار مطلوب برای ورودی x هم‌راستا است. نمونه‌گیری بهترین از N (***B**est-**o**f-**N***, BoN) که به «نمونه‌گیری کنارگذاری» نیز شناخته می‌شود، روشی است که با استفاده از مدل پاداش، بهترین پاسخ را از میان N پاسخ تولیدشده انتخاب می‌کند.



❐ **یادگیری تقویتی** – یادگیری تقویتی (***R**einforcement **L**earning*, RL) رویکردی است که از مدل پاداش بهره می‌برد و مدل f را بر اساس پاداش‌های خروجی‌های تولیدشده‌اش به‌روزرسانی می‌کند. اگر مدل پاداش بر پایه ترجیحات انسانی باشد، این فرایند یادگیری تقویتی از بازخورد انسانی (***R**einforcement **L**earning from **H**uman **F**eedback*, RLHF) نامیده می‌شود.



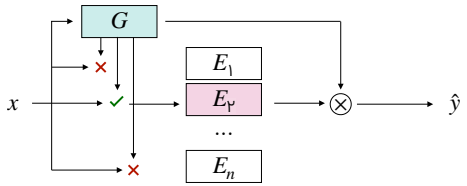
بهینه‌سازی سیاست مجاور (***P**roximal **P**olicy **O**ptimization*, PPO) یک الگوریتم RL پرکاربرد است که با تشویق به دریافت پاداش‌های بالاتر، در عین حال مدل را به مدل پایه نزدیک نگه می‌دارد تا از سوءاستفاده از پاداش جلوگیری کند.

نکته: رویکردهای نظارت‌شده‌ای نیز وجود دارند، مانند بهینه‌سازی مستقیم ترجیحات (*Direct Preference Optimization*, *DPO*)، که مدل پاداش و یادگیری تقویتی را در یک گام نظارت‌شده ترکیب می‌کنند.

۵.۳ بهینه‌سازی‌ها

❐ **ترکیب متخصصان** – مدل ترکیب متخصصان (***M**ixture **o**f **E**xperts*, MoE) مدلی است که تنها بخشی از نورون‌های خود را در زمان استنتاج فعال می‌کند. این مدل بر پایه یک دروازه‌ی G و متخصصان $E_1, \ldots, E_n$ کار می‌کند.

$$\hat{y} = \sum_{i=1}^{n} G(x)_i E_i(x)$$



مدل‌های زبانی بزرگ مبتنی بر MoE از این سازوکار دروازه‌بندی در FFNN خود استفاده می‌کنند.

نکته: آموزش یک مدل زبانی بزرگ مبتنی بر *MoE* به‌طور شناخته‌شده‌ای چالش‌برانگیز است؛ همان‌طور که در مقالهٔ *LLaMA* آمده، نویسندگان آن با وجود کارایی بالای این معماری در زمان استنتاج تصمیم گرفتند از آن استفاده نکنند.

❐ **تقطیر** – تقطیر فرآیندی است که در آن یک مدل دانش‌آموز (کوچک) S بر اساس خروجی‌های پیش‌بینی‌شده‌ی یک مدل معلم (بزرگ) T آموزش می‌بینند. این مدل با استفاده از زیان واگرایی KL آموزش داده می‌شود:

$$\mathrm{KL}(\hat{y}_T || \hat{y}_S) = \sum_i \hat{y}_T^{(i)} \log\left(\frac{\hat{y}_T^{(i)}}{\hat{y}_S^{(i)}}\right)$$

نکته: برچسب‌های آموزشی به‌عنوان «برچسب‌های نرم» در نظر گرفته می‌شوند، زیرا احتمال‌های کلاس را نمایش می‌دهند.

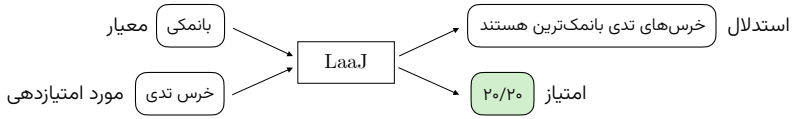
❐ **کوانتیزاسیون** – کوانتیزاسیون مدل دسته‌ای از روش‌ها است که دقت وزن‌های مدل را کاهش می‌دهد در حالی که تأثیر آن بر عملکرد نهایی مدل را محدود می‌کند. در نتیجه، این کار حجم حافظهٔ مورد نیاز مدل را کم کرده و سرعت استنتاج آن را افزایش می‌دهد.

نکته: $QLoRA$ یک واریانت کوانتیزه‌شدهٔ پرکاربرد از $LoRA$ است.

۴ کاربردها

۱.۴ LLM به‌عنوان قاضی

❐ **تعریف** – LLM به‌عنوان قاضی (***L**LM-**a**s-**a**-**J**udge*, LaaJ) روشی است که از یک مدل زبان بزرگ برای امتیازدهی به خروجی‌های داده‌شده بر اساس معیارهای مشخص‌شده استفاده می‌کند. از نکات برجسته این است که LaaJ می‌تواند برای امتیاز خود یک استدلال تولید کند که به قابلیت تفسیرپذیری کمک می‌کند.



بر خلاف معیارهای عصر پیش از LLM مانند (***R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation*, ROUGE)، LaaJ به هیچ متن مرجعی نیاز ندارد و این امر ارزیابی هر نوع وظیفه‌ای را آسان می‌کند. به‌ویژه وقتی LaaJ بر مدل قدرتمند بزرگی (مثلاً GPT-4) تکیه می‌کند، با رتبه‌بندی‌های انسانی همبستگی قوی‌ای نشان می‌دهد، چرا که برای عملکرد مطلوب نیازمند توانایی‌های استدلالی است.

نکته: $LaaJ$ برای انجام دورهای سریع ارزیابی مفید است؛ اما ضروری است که تطابق خروجی‌های $LaaJ$ با ارزیابی‌های انسانی را زیر نظر داشته باشیم تا از بروز هرگونه انحراف جلوگیری شود.

❐ **سوگیری‌های متداول** – مدل‌های LaaJ ممکن است دارای سوگیری‌های زیر باشند:

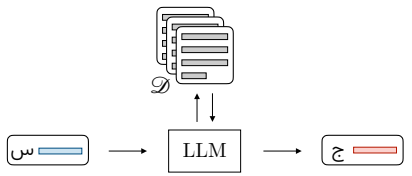
	سوگیری نسبت به موقعیت	سوگیری پرگویی	سوگیری خودستایی
مسئله	تعلق به موقعیت اول در مقایسه‌های دوتایی	تعلق به محتوای پرحرف‌تر	تعلق به خروجی‌هایی که خودشان تولید کرده‌اند
راه‌حل	میانگین‌گیری معیار بر موقعیت‌های تصادفی‌شده	اعمال جریمه بر طول خروجی	استفاده از قاضی ساخته‌شده بر پایهٔ مدل پایه‌ای متفاوت

راه‌حلی برای برطرف‌کردن این مشکلات می‌تواند تنظیم دقیق یک LaaJ سفارشی باشد، اما این نیازمند تلاش زیادی است.

نکته: فهرست سوگیری‌های بالا جامع نیست.

۲.۴ RAG

❐ **تعریف** – روش (***R**etrieval-**A**ugmented **G**eneration*, RAG) این امکان را می‌دهد که مدل LLM برای پاسخ به یک پرسش، به دانش خارجی مرتبط دسترسی پیدا کند. این روش به‌ویژه زمانی مفید است که بخواهیم اطلاعاتی را وارد کنیم که پس از تاریخ آموزش مدل LLM به وجود آمده‌اند.



با داشتن پایگاه دانشی چون $\mathscr{D}$ و یک پرسش، یابنده اسناد مرتبط را بازیابی می‌کند، سپس پرسش را با این اطلاعات غنی‌سازی کرده و خروجی را تولید می‌کند.

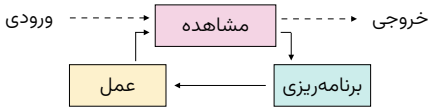
نکته: مرحلهٔ بازیابی معمولاً بر پایه بردارهای تعبیه‌شده‌ای است که از مدل‌های صرفاً رمزگذار به‌دست می‌آیند.

❐ **ابر پارامتر ها** – پایگاه دانش $\mathscr{D}$ ابتدا با تقسیم اسناد به بخش‌هایی به اندازهٔ n_c ساخته می‌شود و سپس این بخش‌ها به بردارهایی با ابعاد $\mathbb{R}^d$ تبدیل می‌گردند.

۳.۴ عامل‌ها

❐ **تعریف** – عامل سیستمی است که به‌صورت مستقل برای رسیدن به هدف‌ها تلاش کرده و وظایف را از طرف کاربر انجام می‌دهد. این سیستم می‌تواند از زنجیره‌های مختلفی از فراخوانی‌های مدل زبانی بزرگ بهره ببرد.

❐ **ReAct** – ***Reason** + **Act*** (ReAct) امکان استفاده از چند زنجیره فراخوانی مدل زبانی بزرگ را برای حل وظایف پیچیده فراهم می‌کند.



این چارچوب از مراحل زیر تشکیل شده است:

- مشاهده: خلاصه‌سازی اقدامات قبلی و بیان صریح آنچه اکنون می‌دانیم.
- برنامه‌ریزی: مشخص‌کردن وظایف مورد نیاز و ابزارهایی که باید فراخوانی شوند.
- عمل: انجام یک اقدام از طریق API یا جستجو در پایگاه دانش برای اطلاعات مرتبط.

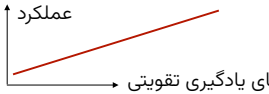
نکته: ارزیابی یک سامانه عاملی چالش‌برانگیز است، اما همچنان می‌توان آن را در سطح مؤلفه‌ها از طریق ورودی-خروجی‌های محلی و در سطح سامانه از طریق زنجیره‌های فراخوانی ارزیابی کرد.

۴.۴ مدل های استدلالی

❐ **تعریف** – مدل استدلال مدلی است که برای حل وظایف پیچیده‌تر در ریاضی، کدنویسی و منطق، به ردپاهای استدلالی مبتنی بر زنجیره تفکر تکیه می‌کند. نمونه‌هایی از مدل‌های استدلال شامل سری o و DeepSeek-R1 و Gemini Flash Thinking هستند.

نکته: مدل $DeepSeek\text{-}R1$ به‌صورت صریح ردپای استدلالی خود را بین برچسب‌های `<think>` تولید می‌کند.

❐ **مقیاس‌پذیری** – دو نوع مقیاس‌پذیری برای تقویت توانایی‌های استدلالی استفاده می‌شوند:

	توضیح	نگاره
مقیاس‌پذیری در زمان آموزش	با اجرای طولانی‌تر یادگیری تقویتی، مدل می‌تواند پیش از پاسخ، نحوه تولید مسیرهای استدلالی سبک زنجیره تفکر را یاد بگیرد	
مقیاس‌پذیری در زمان آزمون	با اعمال بودجه زمانی و واژه‌هایی مانند "`Wait`"، اجازه دهید مدل تأمل بیشتری قبل از پاسخ داشته باشد