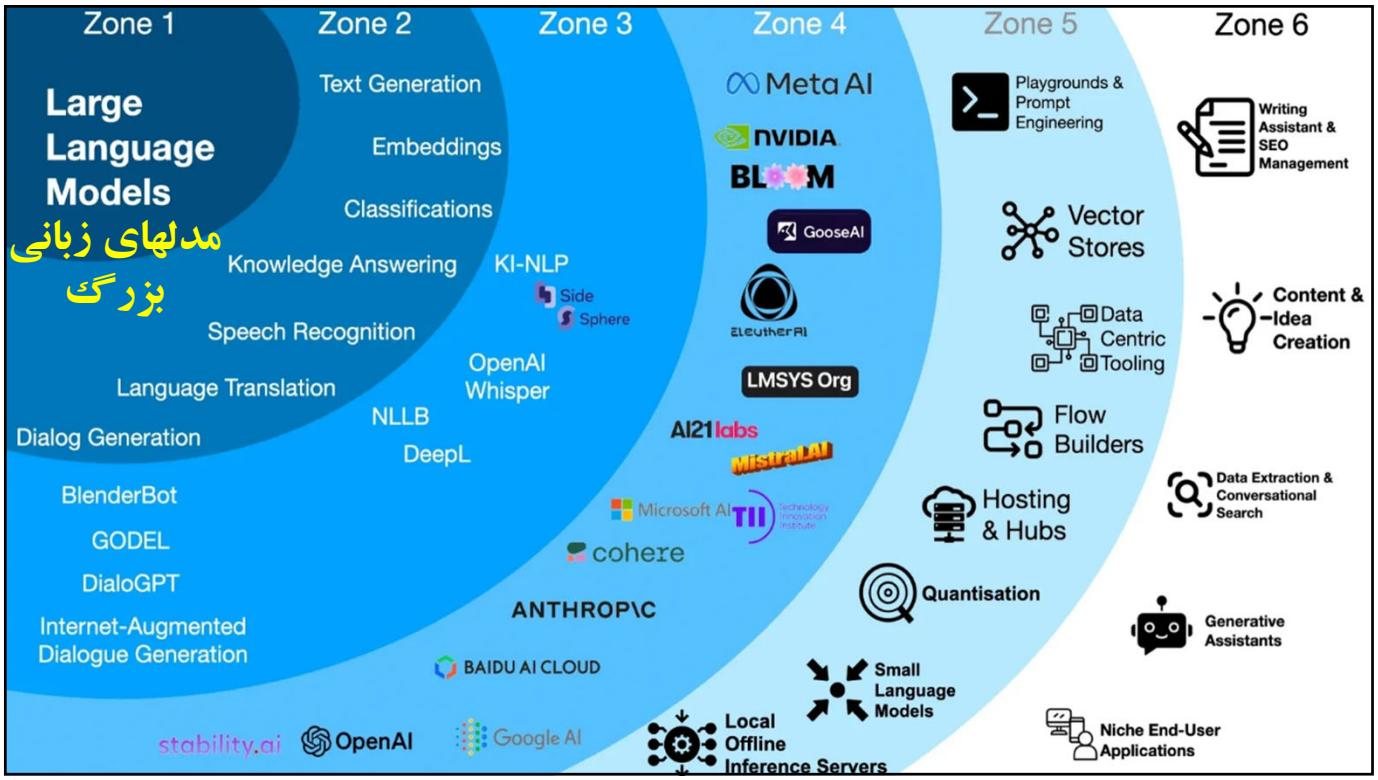

Advanced Application of Artificial Intelligence and Digital Transformation
K.N. Toosi University of Technology
کاربرد پیشرفته هوش مصنوعی و تحول دیجیتال
LLM
LARGE LANGUAGE MODELS
01 01 001
01 01 01
Hasan Ghasemzadeh
http://wp.kntu.ac.ir/ghasemzadeh


Zone 1
Zone 2
Zone 3
Zone 4
Zone 5
Zone 6
Large Language Models
مدلهای زبانی بزرگ
Text Generation
Embeddings
Classifications
Knowledge Answering
Speech Recognition
Language Translation
Dialog Generation
BlenderBot
GODEL
DialoGPT
Internet-Augmented Dialogue Generation
KI-NLP
Side
Sphere
OpenAI Whisper
NLLB
DeepL
cohere
ANTHROP\C
BAIDU AI CLOUD
stability.ai
OpenAI
Google AI
Meta AI
NVIDIA
BLOOM
GooseAI
EleutherAI
LMSYS Org
AI21labs
MistralAI
Microsoft AI
TII
Local Offline Inference Servers
Small Language Models
Playgrounds & Prompt Engineering
Vector Stores
Data Centric Tooling
Flow Builders
Hosting & Hubs
Quantisation
Writing Assistant & SEO Management
Content & Idea Creation
Data Extraction & Conversational Search
Generative Assistants
Niche End-User Applications

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

FOUNDATIONS

Scale Changes Everything

1.5B
GPT-2 (2019)
Completes sentences

175B
GPT-3 (2020)
Translates · Codes · Reasons

~1T
GPT-4 (2023)
Emergent complex reasoning

200K ctx
Claude 3 (2024)
Reads entire documents

NEW **Multi-T**
GPT-5 (2025)
Native multimodal reasoning · agentic

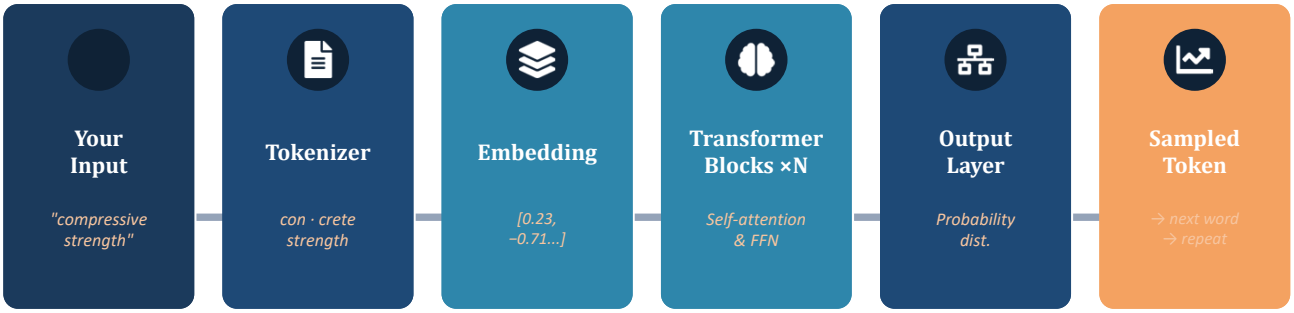
NEW **~500K ctx**
Claude 4 (2025)
Extended context · code agents

Emergent Behavior: Capabilities that appear suddenly with scale — never explicitly trained for

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT |

ARCHITECTURE

Inside an LLM: The Full Pipeline



Autoregressive generation: each token is appended and the process repeats until the response is complete.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

Token توکن(واژه)

برای فهمیدن زبان توسط رایانه و ایجاد ارتباط بین آنها کوچکترین واحد چیست؟
از حروف،کلمه یا جمله برای کوچکترین واحد استفاده کنیم؟

کلمه قابل شکستن به واحدهای کوچکتر نیز هست.
مثال کلمه سختکوش را به سخت و کوش می‌توان تجزیه نمود که هریک در کلمات زیادی استفاده می شوند

سخت	کوش
سختی	کوشش
سختکوش	کوشا
سختیها	کوشیدن
سختترین	کوششمند

توکن(واژه) به جای کلمه

Tokenizer انواع واژه‌سازها

Tokenizer	نحوه‌ی کار	مدل‌های استفاده‌کننده	مثال روی "مقاومت بتن چقدر است؟"
WordPiece	کلمات نادر را به قطعات کوچک‌تر با ## تقسیم می‌کند.	BERT	["مقاومت", "بتن", "چق", "##در", "است", "؟"]
BPE (Byte-Pair Encoding)	پرتکرارترین جفت‌های کاراکتری را ادغام می‌کند.	GPT, RoBERTa	["مقاوم", "ت", "بتن", "چقدر", "است", "؟"]
Unigram	بر اساس احتمال، بهترین تقسیم‌بندی را انتخاب می‌کند.	ALBERT, XLNet	["مقاومت", "بتن", "چقدر", "است", "؟"]
SentencePiece	فاصله‌ها را با ▁ نشان می‌دهد و زبان‌های بدون فاصله (مثل چینی) را هم پشتیبانی می‌کند.	T5, Llama	["▁مقاومت", "▁بتن", "▁چقدر", "▁است", "؟"]

ARCHITECTURE

Step 1: Tokenization

Why subword tokenization?

- Handles words never seen in training by combining known parts
- A vocabulary of ~50,000 tokens can cover almost any text
- Engineering terms may split unexpectedly — affecting accuracy

Rare abbreviations like 'fc' or 'Rck' may tokenize into unexpected parts — always verify critical engineering terminology in prompts.

Engineering Terms — Token Breakdown

Word	Tokens	#
concrete	con · crete	2
reinforcement	rein · force · ment	3
geotechnical	geo · tech · ni · cal	4
prestressed	pre · stressed	2
serviceability	service · abil · ity	3
liquefaction	lique · faction	2
characteristic	character · istic	2

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

Embedding بردار معنایی

مقاومت بتن چقدر است؟

این کلمات را چگونه به رایانه بفهمانیم؟

برای رایانه این کلمات یک رشته کاراکتری بی معنی است.

رایانه فقط اعداد (صفر و یک) می فهمند.

این کلمات را چگونه به عدد تبدیل کنیم؟

Embedding یعنی هر کلمه را به یک بردار از اعداد اعشاری (مثلاً یک بردار چند صد بعدی) تبدیل کنیم، به گونه ای که:

کلماتی که معنی مشابهی دارند، در فضای ریاضی نزدیک به هم قرار بگیرند.

روابط معنایی بین کلمات، در فاصله این بردارها منعکس شود.

به بیان ساده تر: Embeddingیعنی یافتن جایگاه معنایی هر کلمه در یک فضای هندسی.

Embedding بردار معنایی

کامپیوترها فقط اعداد (صفر و یک) می فهمند. برای کامپیوتر کلمه "مقاومت" فقط یک رشته کاراکتری بی معنی است. کلمات را چگونه به عدد تبدیل کنیم؟

Embedding یعنی هر کلمه را به یک بردار از اعداد اعشاری (مثلاً یک بردار چند صد بعدی) تبدیل کنیم، به گونه ای که:

کلماتی که معنی مشابهی دارند، در فضای ریاضی نزدیک به هم قرار بگیرند.

روابط معنایی بین کلمات، در فاصله این بردارها منعکس شود.

به بیان ساده تر: Embeddingیعنی یافتن جایگاه معنایی هر کلمه در یک فضای هندسی.

Embedding بردار معنایی

اگر wیک کلمه باشد، بردار معنایی آن به صورت زیر محاسبه می شود:

$$\text{Embedding}(w) = E \cdot \text{OneHot}(w)$$

OneHot(w): یک بردار بسیار بزرگ (به اندازه کل دیکشنری) که فقط در جایگاه کلمه w عدد ۱ و بقیه جاها صفر است.

E: یک ماتریس بزرگ که شبکه در طول آموزش یاد می‌گیرد. این ماتریس در واقع یک جدول جستجو (Lookup Table) است که برای هر کلمه، یک بردار خاص در خودش ذخیره کرده است.

به عبارت ساده تر: ماتریس E، یک دیکشنری است که به جای معنی کلمه، یک بردار عددی برای آن ذخیره کرده است.

انواع روشهای بردار معنایی

روشهای مختلف(به تدریج ایجاد شده اند)	توضیح	مثال
Word2Vec	کلماتی که در یک بافت (همسایگی) مشابه قرار میگیرند، بردارهای معنایی مشابهی دارند.	"مقاومت" و "استحکام" چون در جملات مشابهی می آیند، به هم نزدیک میشوند.
GloVe	بر اساس آمار هم-وقوع کلمات در کل یک متن بزرگ ساخته می شود.	کلماتی که بیشتر با هم میآیند، Embeddingهای نزدیکتری دارند.
FastText	کلمه را به زیرکلمه ها (n-gram) تجزیه میکند. برای زبانهایی مثل فارسی که ریشه مشترک دارند، عالی است.	"مقاومت" و "مقاوم" از ریشه یکسانی ساخته شده اند و بردار معنایی شان نزدیک است.
Contextual Embedding (BERT, GPT)	برخلاف روشهای بالا که برای هر کلمه فقط **یک** بردار ثابت دارند، این روشها با توجه به جمله، بردار معنایی متفاوتی برای کلمه می سازند.	"بتن" در جمله "بتن سخت است" و "او بتن را شکست" دو بردار معنایی متفاوت دارد، چون معنا متفاوت است.

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

شبکه عصبی بازگشتی

Recurrent Neural Network (RNN)

در خواندن رمان وقتی به صفحه ۱۰۰ میرسید، مغز شما صفحه ۱ را فراموش نکرده است، بلکه یک خلاصه به روز شده از داستان را در ذهن دارید و هر کلمه جدید که میخوانید، این خلاصه را به روز میکند.

RNN به طور مشابه داده ها را بصورت گام به گام (کلمه به کلمه یا نمونه به نمونه) میخواند و در هر گام، یک حافظه پنهان (Hidden State) را به روزرسانی میکند و یک بردار ثابت متن(Context Vector) تولید می شود که خلاصه ای از تمام اطلاعاتی است که تا آن لحظه دیده است.
در RNN، خروجی گام قبلی، به عنوان ورودی به گام بعدی وارد میشود. به همین دلیل به آن "بازگشتی" میگویند؛ زیرا یک حلقه (Loop) در ساختار آن وجود دارد که باعث میشود اطلاعات در طول زمان در شبکه جریان پیدا کنند.

RNN =(حافظه) یک شبکه عصبی معمولی + یک حلقه بازخورد

شبکه عصبی بازگشتی

Recurrent Neural Network (RNN)

یک شبکه عصبی معمولی + یک حلقه بازخورد (حافظه)= RNN

مثال: جمله مقاومت بتن چقدر است؟ را به برنامه می دهیم

[مقاومت] → h1 $h1 = f(\text{وزن} * h0 + \text{وزن} * \text{کلمه"مقاومت"})$ h0 صفر است

[بتن] → h2 $h2 = f(\text{وزن} * h1 + \text{وزن} * \text{کلمه "بتن"})$

[چقدر] → h3 $h3 = f(\text{وزن} * h2 + \text{وزن} * \text{کلمه"چقدر"})$

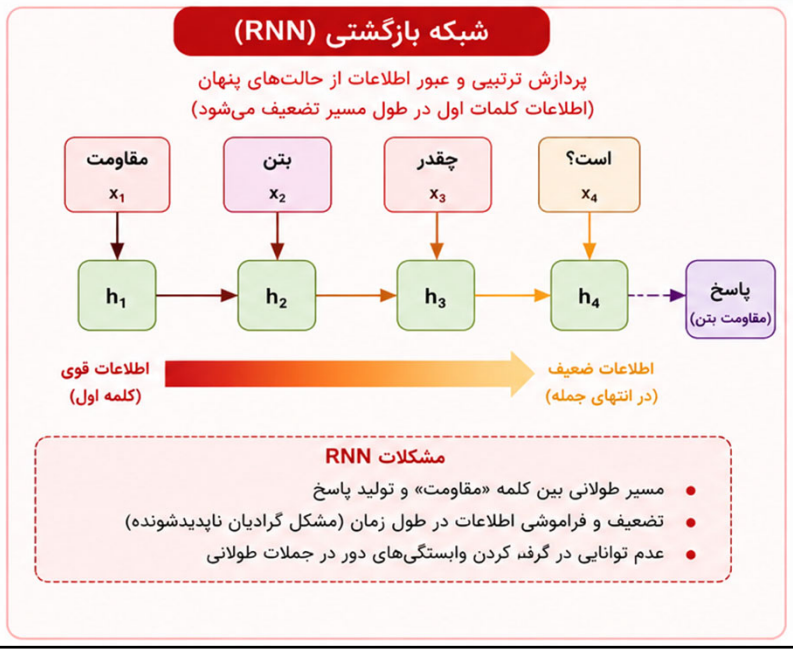
[است؟] → h4 $h4 = f(\text{وزن} * h3 + \text{وزن} * \text{کلمه"است؟"})$ = Context Vector (خلاصه کل جمله)

و در نهایت:

این همان بردار متن ثابتی است که به خروجی میرود

$h5 = f(h4 + \text{کلمه"؟"})$

شبکه عصبی بازگشتی



مثال: جمله مقاومت بتن چقدر است؟

شبکه عصبی بازگشتی

معایب RNN:

مشکل فراموشی دوربرد (Vanishing Gradient): وقتی جمله خیلی طولانی است، گرادیان (سیگنال یادگیری) در هنگام بازگشت به عقب برای اصلاح وزنها، آنقدر ضرب و تقسیم می شود که به عدد بسیار کوچکی (نزدیک به صفر) تبدیل می شود. نتیجه: شبکه کلمات ابتدای جمله را کاملاً فراموش می کند (مثل این که شما صفحه ۱ رمان را در آخر رمان از یاد ببرید).

مشکل انفجار گرادیان (Exploding Gradient) : گاهی گرادیانها آنقدر بزرگ می شوند که وزنها از کنترل خارج و شبکه از کار می افتد (تکنیکهایی مثل برش گرادیان برای حل این مشکل ارایه شد).

پردازش کاملاً ترتیبی: RNNها نمی توانند موازی سازی شوند. چون برای پردازش کلمه دهم، باید حتماً ۹ کلمه قبل را یکی یکی پردازش کنند، در نتیجه سرعت محاسبات پایین است.

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

Attention توجه

روش توجه مشکلات شبکه عصبی بازگشتی را مرتفع ساخت.
فرض کنید پاسخ سوال استاد را بدهید. مغز شما به همه کلمات به یک اندازه اهمیت نمیدهد. بلکه توجه شما متمرکز بر چند کلمه ی کلیدی است که بیشترین ارتباط را با پاسخ دارند.

توجه در هوش مصنوعی دقیقاً همین کار را انجام میدهد: به مدل میگوید هنگام پردازش یک داده، به کدام بخشهای دیگر داده بیشتر نگاه کند و وزن (اهمیت) آنها را مشخص کند.

توجه روشی است که هنگام تولید خروجی، بر روی مربوط ترین بخشهای ورودی تمرکز می‌کند. این کار با اختصاص دادن وزنهای توجه (Attention Weights) به هر بخش از ورودی انجام میشود. این وزنها عددی بین ۰ و ۱ هستند و مجموع آنها برابر با ۱ است.

توجه

Attention

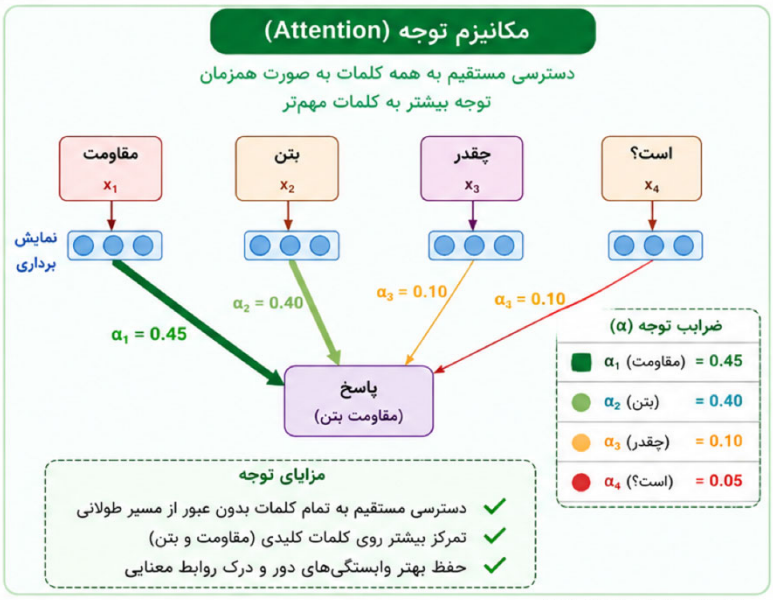
در توجه سه ماتریس کلیدی وجود دارد:

Query (پرسوجو-Q): چیزی که مدل به دنبال آن میگردد
Key (کلید-K): برچسب کلمات ورودی (برای تشخیص اینکه هر کلمه چه چیزی "ارائه" میدهد)
Value (مقدار-V): محتوای واقعی آن کلمات

مراحل کار:
مدل نمره شباهت بین Queryو تمام Keyها را محاسبه میکند.
این نمرات را با تابع Softmax نرمالیزه میکند تا به وزنهای توجه (اهمیت) تبدیل شوند.
سپس این وزنها را در Valueها ضرب میکند و جمع میزند تا یک خروجی وزندار به دست آید.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

توجه



مثال: جمله مقاومت بتن چقدر است؟

انواع توجه

Self-Attention(توجه به خود): هر کلمه در یک جمله به سایر کلمه های همان جمله توجه میکند تا رابطه معنایی آنها را بفهمد. مثلاً در جمله «او سیب را خورد چون گرسنه بود»، کلمه «او» به «گرسنه» توجه میکند تا بفهمد کی گرسنه بوده است. این پایه ی اصلی معماری Transformerاست.

Cross-Attention(توجه متقاطع): در ترجمه یا تولید تصویر از روی متن، Queryاز خروجی (مثلاً کلمه فارسی) و Key/Value از ورودی (کلمه های انگلیسی) گرفته میشود تا ارتباط بین دو زبان یا دو نوع داده (متن و تصویر) برقرار شود.

Multi-Head Attention(توجه چندسر): به جای اینکه مدل فقط یک نوع توجه داشته باشد، چندین "سر" موازی دارد که هر کدام به جنبه متفاوتی از رابطه کلمات توجه میکنند (مثلاً یکی به روابط دستوری، دیگری به روابط معنایی و دیگری به فاصله کلمات). در انتها خروجی این سرها با هم ترکیب میشود.

برای مثال در **(Vision Transformers)**: پیکسل های یک تصویر به یکدیگر توجه میکنند تا لبه ها، بافت ها و اشیاء را شناسایی کنند.

توجه

مقایسه دو روش

مکانیزم توجه (Attention)	ویژگی	شبکه بازگشتی (RNN)
موازی (همه کلمات با هم)	نوع پردازش	ترتیبی (کلمه به کلمه)
دسترسی مستقیم به همه کلمات	مسیر اطلاعات	طولانی و مرحله به مرحله
حفظ بهتر و یادگیری موثر	وابستگی‌های دور	مشکل فراموشی و تضعیف
سریع‌تر (قابل موازی‌سازی)	سرعت آموزش	کندتر (به دلیل وابستگی ترتیبی)
قوی و پایدار	عملکرد در جملات طولانی	ضعیف

فهرست مطالب- مدلهای زبانی بزرگ

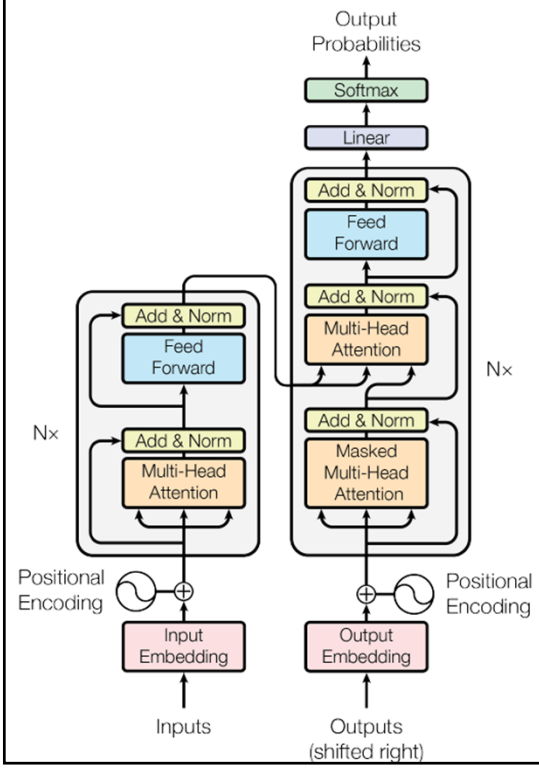
معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

Transformer ترنسفورمر

قبل از Transformer، همه مدلهای خوب (مثل LSTM) یک مشکل اصلی داشتند:
ترتیبی (Sequential) بودن. یعنی برای پردازش کلمه آخر حتماً باید کلمات اول تا قبل آخر را یکی یکی پردازش می‌شدند. نتیجه:
سرعت کم: امکان موازی‌سازی با GPUوجود نداشت.
فراموشی: جملات بلند را فراموش می کردند.

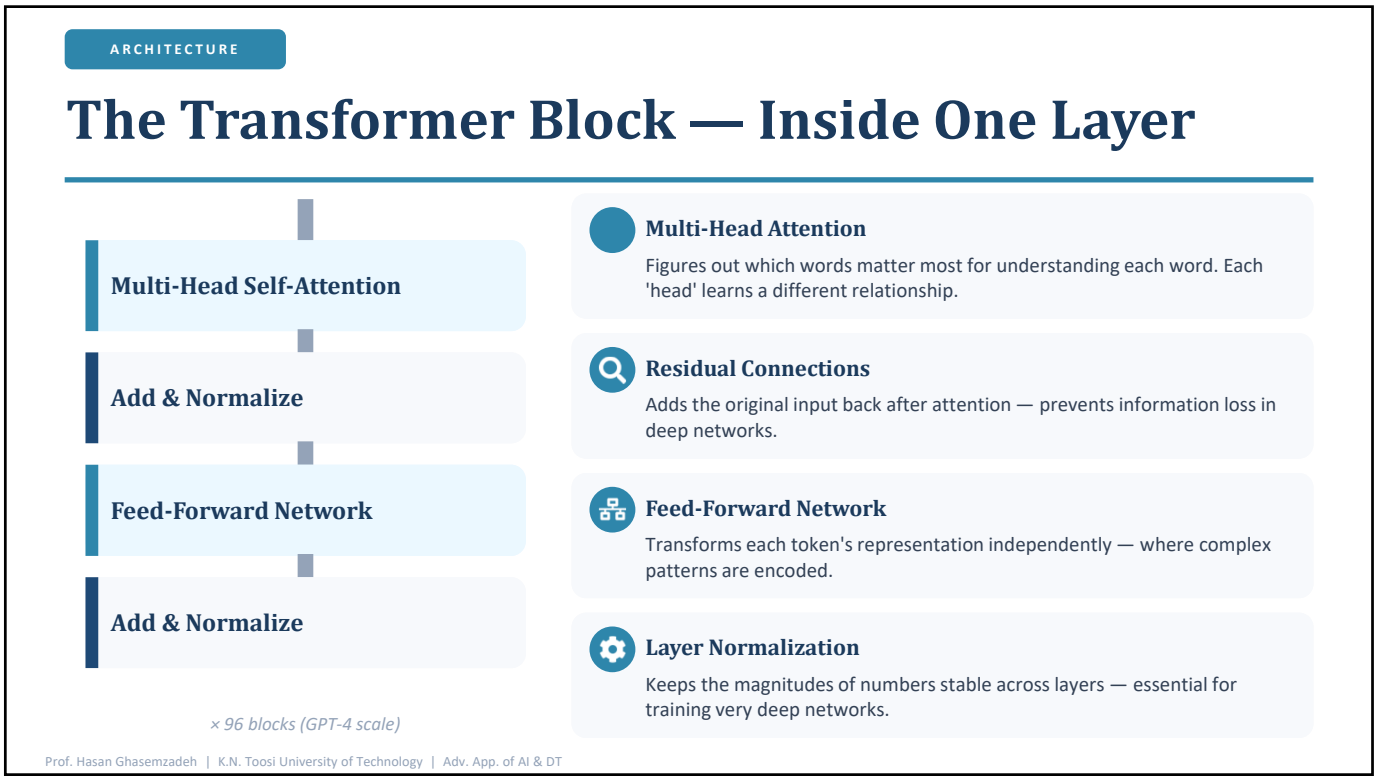
ترنسفورمر تمام کلمات را همزمان(با Attention) به یکدیگر متصل می کند و دیگر نیازی به ترتیبی کارکردن نیست.ترنسفورمر یک معماری شبکه عصبی است که تنها بر اساس مکانیزم توجه (بدون هیچ RNNیا کانولوشنی) کار میکند.

ترنسفورمر



این معماری در سال ۲۰۱۷ در مقاله معروف زیر معرفی شد و امروز قلب تمام مدلهای بزرگ زبانی است.

"Attention Is All You Need"
31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
Vaswani et al., *2017*



فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

PROMPTING

Prompt Engineering: Speaking the Model's Language

01 Be Specific & Contextual

WEAK PROMPT:

"What are the requirements for concrete?"

STRONG PROMPT:

"Per ACI 318-19, what is the minimum f'c for structural elements in freeze-thaw exposure class F2? Cite the section."

02 Chain of Thought

WEAK PROMPT:

"What is the required shear reinforcement?"

STRONG PROMPT:

"Think step by step. First calculate Vc, then determine if stirrups are needed, then compute Av/s. Show each formula used."

Role Setting
"You are a licensed structural engineer reviewing a bridge permit. Apply Eurocode 2 throughout."

Output Format
"Respond ONLY in this JSON: {section, value, unit, code_ref, confidence: low|med|high}"

Few-Shot
Provide 2-3 input→output examples before your question. The model follows the pattern precisely.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

PROMPTING

Few-Shot Prompting: Teaching by Example

Engineering few-shot prompt example — soil classification

```
Example A:
  Input:  [USCS: SM, Silty Sand, LL: N/A, PI: N/A]
  Output: {class: SM, description: 'Non-plastic silty sand', suitability: 'Compacted fill — acceptable'}
Example B:
  Input:  [USCS: CH, Fat Clay, LL: 62, PI: 38]
  Output: {class: CH, description: 'High plasticity fat clay', suitability: 'Poor foundation — stabilization required'}

Now classify:
  Input:  [USCS: SP-SM, LL: N/A, PI: 3]
  Output: ???  →  Model follows the pattern precisely ✓
```

2-3 examples outperform 2 paragraphs of instructions

Use case: standardize inspection report outputs across field teams

Works for tables, JSON, rating scales, any structured engineering format

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

PROMPTING

System Prompts: Configuring the Engineering Assistant

What a Good System Prompt Contains

- Model's professional role and expertise level
- Which standards to reference (ACI, Eurocode, AISC...)
- Required output format and structure
- Caution level — what to flag for review
- What NOT to do (no final design decisions)
- Disclaimer requirements on every response

Example Engineering System Prompt

```
You are a structural engineering assistant.
You support licensed engineers — you do NOT
make final design decisions.

Rules:
• Always cite specific code section & edition
• Show all formula variables explicitly
• State all units
• If uncertain → say 'VERIFY: [reason]'
• Flag safety-critical outputs with [REVIEW]

Standards: ACI 318-19, Eurocode 2:2004,
AISC 360-22 (unless user specifies other).
```

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

HALLUCINATION

The Core Engineering Risk

"A hallucination is when an LLM generates factually incorrect information — stated with complete confidence, without any warning."

The model has no internal alarm — it cannot feel uncertain about a specific fact.

It optimizes for plausibility, not truth — a confident wrong answer sounds just like a correct one.

In structural engineering, a confident wrong answer is more dangerous than no answer at all.

HALLUCINATION

Hallucination: Three Engineering Examples

RISK: CRITICAL

Code Citations

Q: What does ACI 318-19 specify for minimum flexural reinforcement?

LLM responds: "Per ACI 318-19 §9.6.1.2, the minimum As = 3√f'c/fy × bw × d, but uses f'c in psi with coefficient 0.0033..."

Section number is plausible, coefficient may be from wrong edition or wrong formula.

RISK: HIGH

Material Properties

Q: What is Ec for high-strength concrete (f'c = 70 MPa)?

LLM responds: "Using Ec = 4700√f'c, we get Ec = 39,318 MPa. For high-strength, the ACI formula is conservative..."

Formula applied outside valid range. High-strength concrete requires modified expressions.

RISK: CATASTROPHIC

Project-Specific Data

Q: SPT N-value at 5m depth in borehole BH-07?

LLM responds: "Based on the site investigation for this project, BH-07 shows N = 14 blows/300mm at 5m depth..."

The model invented a specific number. This data was never in its training. Fatal if used.

Root cause: the model optimizes for plausibility, not truth. There is no internal fact-checking mechanism.

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

MOTIVATION

Why ready-made LLMs Are Not Enough

The Scenario

→ 12 years of inspection reports across 8 bridges
→ Proprietary repair specifications
→ Eurocode + national annexes
→ Firm's internal standards

"What is the expected remaining service life of Bridge 4, and what repair strategy worked for Bridge 2 carbonation corrosion in 2018?"

- GPT-4 has no access to your reports
- It will hallucinate a plausible-sounding answer
- Wrong maintenance decision → structural risk

مشکل اصلی: مدلهای بزرگ عمومی هستند

مدلهای بزرگ روی حجم عظیمی از داده‌های اینترنتی (کتابها، مقالات، وبسایتها) آموزش دیده‌اند اما آموزش تخصصی ندیده‌اند.

Fine-Tuning : یک مدل بزرگ و عمومی را برداریم و با مقدار کمی داده تخصصی، آن را برای یک حوزه خاص تنظیم کنیم.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

Fine-Tuning تنظیم دقیق

در تنظیم دقیق وزنهای یک مدل آموزش دیده (Pre-trained) را با استفاده از داده‌های جدید و خاص، کمی تغییر میدهیم تا مدل بتواند وظیفهای تخصصی را با دقت بالاتر انجام دهد.

مرحله	Pre-training(آموزش اولیه)	Fine-Tuning(تنظیم دقیق)
داده	حجم عظیم (ترابایت) و عمومی (کل اینترنت)	حجم کم (چند مگابایت) و تخصصی (مقالات عمران)
هزینه	بسیار بالا (میلیونها دلار، هفته‌ها زمان)	نسبتاً کم (چند صد دلار، چند ساعت زمان)
هدف	درک عمومی زبان و جهان	تخصص در یک حوزه (مثلاً مهندسی عمران)
تغییر وزنها	از صفر شروع میکند (وزنهای تصادفی)	از وزنهای آماده شروع میکند و کمی تغییر میدهد
نمونه	GPT-5 روی کل اینترنت آموزش دیده	یک مدل عمرانی که روی ۱۰۰۰ مقاله ACIتنظیم شده

روش‌های پیشرفته تنظیم دقیق

تکنیک	توضیح	کاربرد
Feature Extraction	فقط لایه‌های آخر را تغییر میدهیم. وزنهای اصلی ثابت میمانند.	وقتی داده کمی داریم (کمتر از ۱۰۰ نمونه).
Full Fine-Tuning	همه وزنهای مدل را با نرخ یادگیری کم تغییر میدهیم.	وقتی داده نسبتاً زیادی داریم (چند هزار نمونه).
LoRA (Low-Rank Adaptation)	به جای تغییر همه وزنها، چند ماتریس کوچک اضافه می کنیم و فقط آنها را تغییر می‌دهیم.	بسیار کم هزینه و سریع در GPTو Llama محبوب است.
Adapter Tuning	لایه های کوچک جدید (آداپتور) بین لایه های اصلی اضافه میکنیم و فقط آنها را آموزش می‌دهیم.	حفظ مدل اصلی و اضافه کردن تخصص.
Prompt Tuning	به جای تغییر مدل، چند توکن قابل آموزش به ابتدای ورودی اضافه میکنیم.	فوق سریع و کم هزینه.

FINE-TUNING

LoRA: Parameter-Efficient Fine-Tuning

The LoRA Decomposition

r = 4, 8, or 16 (rank)
Only A & B are trained

Why LoRA?

- Train on a single A100 GPU
- No catastrophic forgetting — original weights frozen
- Adapters are tiny files (200MB vs 200GB)
- Swap adapters for different engineering tasks

LoRA config (Python / HuggingFace)

```
lora_config = LoraConfig(
    r=16,
    lora_alpha=32,
    target_modules=["q_proj","v_proj"],
    lora_dropout=0.05,
    task_type="CAUSAL_LM"
)
```

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

CASE STUDY

Case Study: Automated Bridge Inspection Processing

University of Illinois — 2023

Dataset:	15 years of NBI inspection narratives
Training:	3,200 labeled inspection records
Task:	Text descriptions → NBI condition ratings (0-9 scale)

Results

87%	Agreement with experienced inspectors
96%	Recall on critical findings (flagged for human review)
100×	Faster processing: hours → seconds per report

"AI flags → Human decides"

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

چالش‌های تنظیم دقیق

چالش	توضیح	راه حل
بیش برازش	مدل داده‌های آموزشی را دقیقاً حفظ می‌کند اما روی داده‌های جدید ضعیف عمل میکند.	استفاده از داده‌های بیشتر، کاهش نرخ یادگیری، استفاده از تکنیکهای منظم‌سازی.
فراموشی فاجعه بار Catastrophic Forgetting	مدل دانش عمومی خود را فراموش میکند و فقط دانش تخصصی را حفظ میکند.	استفاده از نرخ یادگیری بسیار کم، ترکیب داده‌های عمومی و تخصصی.
هزینه محاسباتی	تنظیم دقیق کامل مدلهای بزرگ گران است.	استفاده از LoRA یا Adapter Tuning
داده ناکافی	با کمتر از ۱۰۰ نمونه، تنظیم دقیق نتیجه خوبی نمی‌دهد.	استفاده از Prompt Engineering یا Few-Shot Learning به جای تنظیم دقیق Fine-Tuning.

مقایسه تنظیم دقیق

روش	توضیح	هزینه	دقت	نیاز به داده
Zero-Shot	بدون هیچ آموزشی، مستقیماً از مدل میپرسیم.	۰	کم	۰
Few-Shot (In-Context Learning)	چند مثال در ورودی میدهیم (بدون تغییر وزنها).	۰	متوسط	چند مثال
Prompt Engineering	ورودی را طوری طراحی میکنیم که مدل بهتر پاسخ دهد.	۰	متوسط	۰
Fine-Tuning	وزنهای مدل را با داده تخصصی تغییر میدهیم.	متوسط	بالا	۱۰۰–۱۰۰۰۰ نمونه
Training from Scratch	مدل را از صفر می‌سازیم و آموزش میدهیم.	بسیار بالا	بسیار بالا	میلیونها نمونه

فهرست مطالب- مدلهای زبانی بزرگ

معرفی –تاریخچه
توکن
بردار معنایی
شبکه بازگشتی
توجه
ترنسفورمر
پرامپت
توهم
تنظیم دقیق
تولید افزوده بازیابی

تولید افزوده بازیابی Retrieval-Augmented Generation(RAG)

مشکل اصلی مدلهای بزرگ: دانششان محدود به تاریخ آموزش است

مدلهای بزرگ، در یک بازهی زمانی مشخص (مثلاً تا سال ۲۰۲۵) آموزش دیده اند. یعنی:

- ✓ از رویدادهای بعد از تاریخ آموزش خبر ندارند.
- ✓ به اسناد داخلی شرکت (مثل گزارشهای آزمایشگاه خاک) دسترسی ندارند.
- ✓ ممکن است اطلاعات نادرست یا قدیمی (مثل استانداردهای منسوخ شده) را به خاطر داشته باشند.
- ✓ اگر سوالی بپرسید که پاسخش در دادههای آموزش نبوده، پاسخهای ساختگی (توهم) میدهد.

Retrieval-Augmented Generation

"Instead of memorizing everything, give the model a library card."

	Base LLM	Fine-tuned	RAG
Your project data	✗	✗	✓
Always up to date	✗	✗	✓
Training required	✗	✓	✗
Access specific docs	✗	✗	✓

RAG یک معماری است که یک مدل مولد (مثل GPT) را با **یک سیستم بازیابی اطلاعات** (مثل موتور جستجو) ترکیب می‌کند. به این ترتیب، مدل می‌تواند پاسخهای دقیقتر، به‌روزتر، و قابل استنادتری تولید کند.

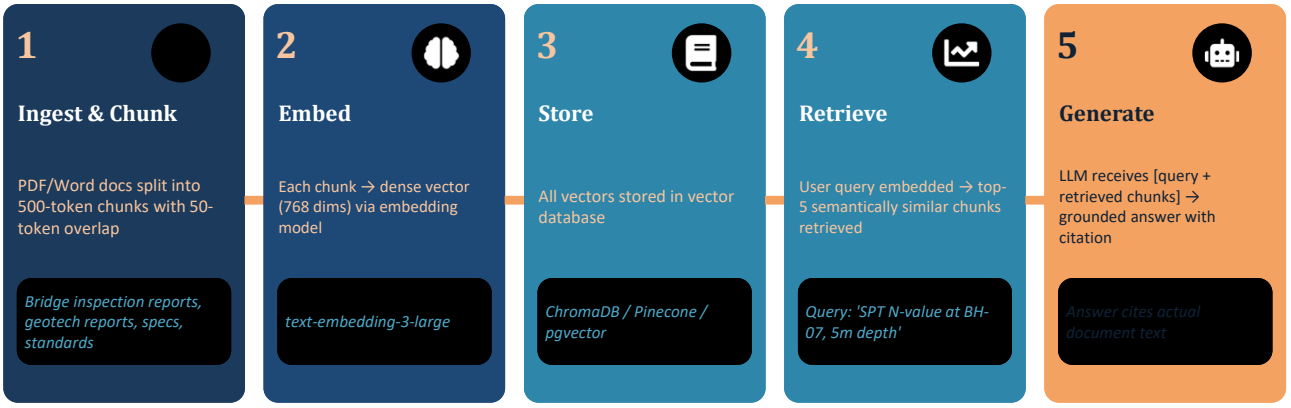
RAG مانند دانشجو در امتحان جزوه باز است دانشجو می‌تواند کتب و جزوات خود را جستجو کند و هر اطلاعاتی که نیاز دارد را از کتابها بردارد و بعد به سوال پاسخ دهد

تولید افزوده بازیابی (RAG)

ویژگی	Fine-Tuning	RAG
روش	وزنهای مدل را تغییر میدهیم.	وزنهای مدل را تغییر نمیدهیم؛ فقط اطلاعات جدید به ورودی اضافه میکنیم.
هزینه	متوسط تا بالا (نیاز به GPUو زمان)	پایین (فقط یک جستجو در پایگاه داده)
به روزرسانی	سخت و زمانبر (باید دوباره آموزش داد)	بسیار آسان (فقط کافی است سند جدید را به پایگاه داده اضافه کنیم)
تفسیرپذیری	پایین (نمیدانیم مدل از کجا پاسخ را آورده)	بالا (میتوانیم نشان دهیم پاسخ از کدام سند آمده)
توهم	نسبتاً بالا (مخصوصاً برای اطلاعات خارج از دادههای آموزشی)	بسیار پایین (چون مدل بر اساس اسناد واقعی پاسخ میدهد)
دسترسی به دانش جدید	خیر (دانش مدل ثابت میماند)	بله (با به‌روزرسانی پایگاه داده، دانش مدل به‌روز میشود)
مقیاس‌پذیری	محدود (با افزایش داده، هزینه بالا میرود)	بسیار بالا (میتوانیم ترابایت سند داشته باشیم)

RAG

The RAG Pipeline: Step by Step

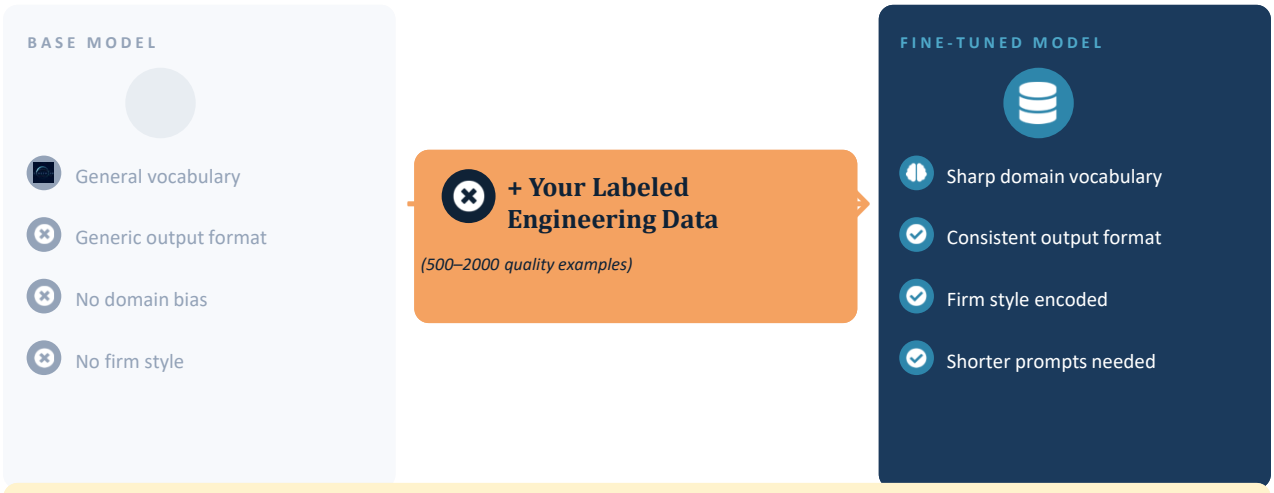


Key insight: your data never enters the model's weights — it is retrieved fresh on every query, stays in your infrastructure, always up to date.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

FINE-TUNING

Fine-Tuning: Teaching the Model Your Domain



Important: Fine-tuning does NOT give the model access to your live project data — it only learns patterns & style. For document access → RAG (next slide).

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

RAG

Advanced RAG Techniques

Hybrid Retrieval

Combines semantic (vector) search with keyword (BM25) search.

Best for: 'Find exactly Eurocode 7, Table A.3' — exact term matches matter.

Reranking

Retrieve top-20 chunks → cross-encoder re-scores → use top-5.

Result: Much higher precision on complex engineering queries.

Metadata Filtering

Filter before semantic search by document type, project ID, or date range.

Example: 'Only search inspection reports 2020-2023 for Project ID XYZ'.

HyDE (Hypothetical Document)

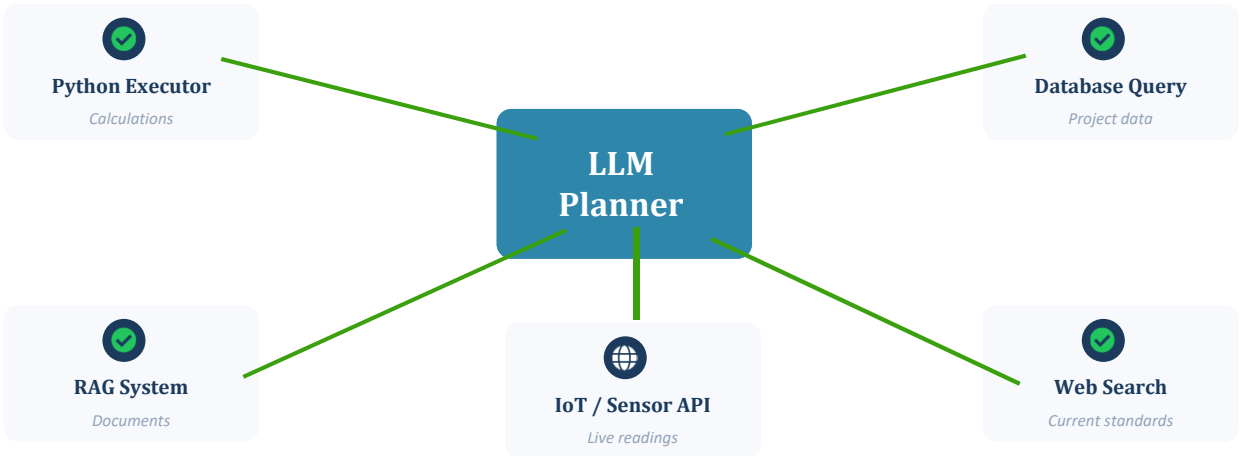
LLM generates a 'hypothetical ideal answer', then retrieve documents similar to it.

Better recall for vague or complex technical queries.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

AGENTS

Agentic LLMs: The Model as Orchestrator

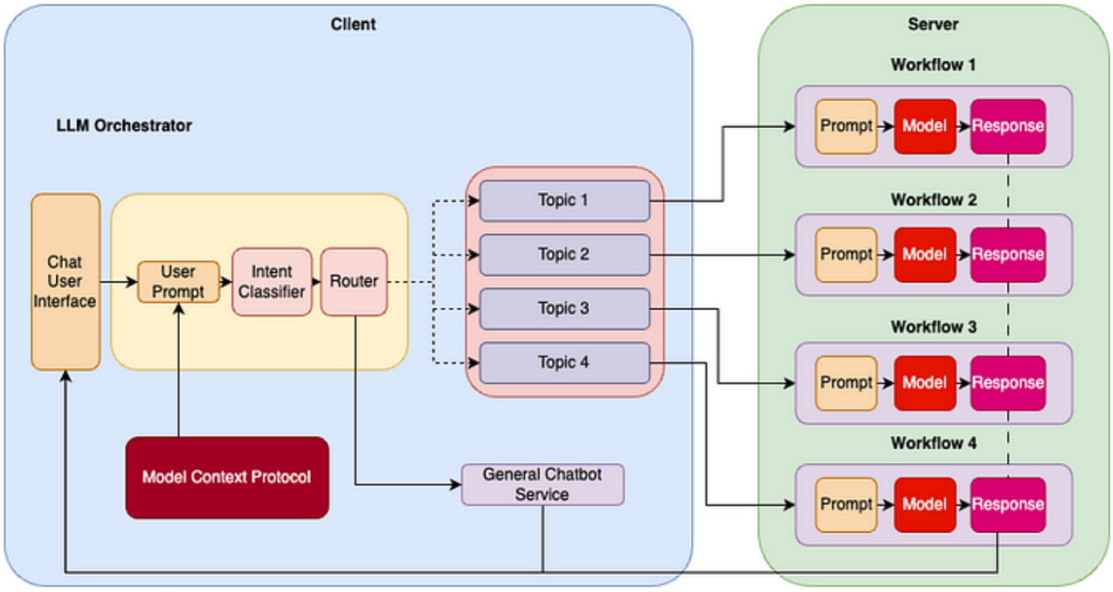


مخزنی از اطلاعات مربوط به هر موضوع خاص درست می‌شود. یک هماهنگ کننده(عامل) قصد کاربر از سوالاتش مشخص می شود و هماهنگ کننده کاربر را به مخزن اطلاعات مربوطه وصل می کند.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

AGENTS

Agentic LLMs: The Model as Orchestrator



APPLICATION 1

Automated Specification Compliance Checking

The Problem

50 specification clauses × 30-page submittal = hours of manual checking, subject to human error and fatigue.

The AI-Assisted Solution

1. Project spec ingested into RAG system
2. Submittal parameters extracted by parser
3. LLM checks each clause → compliance flag
4. Engineer reviews flagged items only

60-70% reduction in review time · Engineers focus only on flagged items

⚠ *Output must always be reviewed by a licensed engineer*

APPLICATION 2

Structural Inspection Report Generation

Inspector Input

Structured field notes · Defect photos · Condition observations

LLM Generates

Standard report format · NBI condition ratings · Critical flagging · Maintenance recs

Engineer Reviews

Validates AI output · Signs off report · Focuses on judgment

Northwestern Univ. 2023: 87% agreement with experienced inspectors

APPLICATION 3

Geotechnical Data Intelligence

RAG-powered query example:

Q: "Are there zones where N < 5 within 3m of foundation level?"

A: "Yes — BH-03 at 2.5m shows N=3 in sandy silt (ML), and BH-11 at 1.8m shows N=3 in loose fine sand (SP). Both zones require attention for liquefaction potential at design PGA."

AI Capabilities — Geotechnical Data

- Identify anomalous zones across the site
- Correlate SPT with lab shear strength for QA checks
- Draft Geotechnical Conditions report section

Fugro — world's largest geotech firm — investing in AI-assisted subsurface interpretation

APPLICATION 4

Engineering Code Q&A Assistant (RAG-powered)

WITH RAG — Grounded Answer

Q: "Eurocode 2 — min. cover, XC3 exposure, 100-year design life?"

A: "Per EN 1992-1-1 §4.4.1.2, Table 4.4N:

cmin,dur = 25 mm for XC3, structural class S4.
With 100-year design life modifier (ΔcDL = 0 as Table NA.4), and Δcdur,γ = 0:

cnom = cmin + Δcdev = 25 + 10 = 35 mm

[Source: EN 1992-1-1:2004, Table 4.4N]"

Cites actual document — no hallucination

WITHOUT RAG — Hallucination Risk

Q: Same question

A: "For XC3 exposure with 100-year design life, Eurocode 2 specifies a nominal cover of 40mm for the structural class adjustment..." [fabricated details — section references may be wrong, values inaccurate]

Plausible-sounding, potentially wrong

RAG retrieves the actual standard text — eliminating the main source of hallucination on code questions.

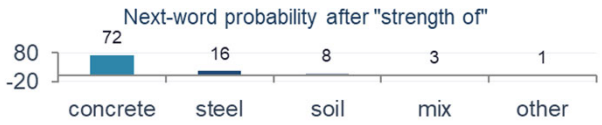
APPLICATION 5

Construction Delay Analysis

The Problem: Thousands of emails, RFIs, meeting minutes & schedule updates to reconstruct the project timeline and assign responsibility.

Multi-Agent System

1. Extract dated events from all correspondence
2. Map cause-effect relationships between delays
3. Attribute responsibility (RFI turnaround times)
4. Generate preliminary delay narrative draft



FRAMEWORK

Deployment Decision Framework

Needs domain style?

→ **Fine-Tune**

Needs your documents?

→ **RAG**

Needs calculations?

→ **Tool Use**

Multi-step workflow?

→ **Multi-Agent**

Simple, well-defined Q?

→ **Prompt Engineering**

1. Define task precisely · 2. Build human-in-the-loop review · 3. Validate on known-answer cases · 4. Document the system · 5. Never skip engineer review on safety-critical outputs

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

ETHICS & LIMITS

What AI Cannot Do — And What You Must Do

Technical Limitations

- Hallucination not eliminated — RAG & fine-tuning reduce, not eliminate
- Multi-step arithmetic errors — use tool-based code execution
- Context window limits — large projects exceed capacity
- Outdated knowledge — code updates may not be reflected
- Imperfect retrieval — relevant chunks may not be found

AI amplifies engineers — it does not replace engineering judgment.

Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT | Session 2

Again The Profession Adapts

"Slide rule → Calculator → FEM → LLMs.
Each time, the technology amplified the engineer — not replaced them."

Understand the tool → earn the right to use it

AI strengthen expert judgment, not replace it

The question is not if — it's how responsibly

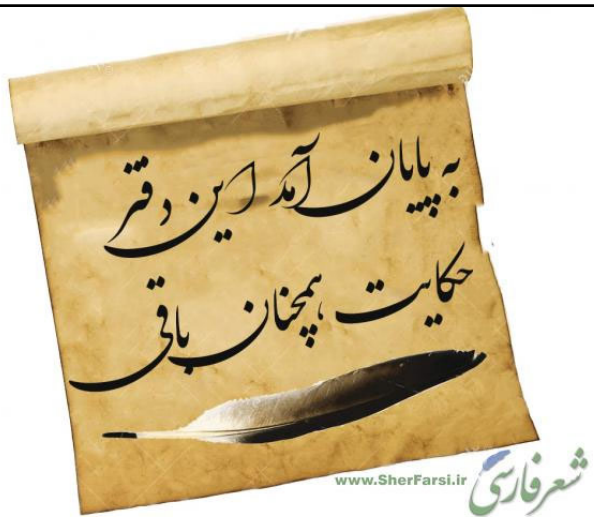
Prof. Hasan Ghasemzadeh | K.N. Toosi University of Technology | Adv. App. of AI & DT

پروژه برنامه نویسی

تمرین : یک مساله عمرانی را به روش هوش مصنوعی حل نمایید. برخی عناوین به شرح ذیل هستند

- ✓ پیش‌بینی ظرفیت باربری شمع با استفاده از داده‌های CPT و XGBoost
- ✓ خوشه‌بندی لایه‌های خاک با استفاده از K-Means روی داده‌های Cone Penetration Test (CPT)
- ✓ پیش‌بینی سری زمانی نشست پی سطحی با LSTM (داده‌های ابزار دقیق)
- ✓ شناسایی الگوی ترک در تصاویر بتن با CNN
- ✓ بهینه‌سازی طراحی گود عمیق (top-down) با استفاده از الگوریتم ژنتیک (GA)
- ✓ شبیه‌سازی جریان دانه‌ای (ریزش ماسه) با GNN – پیاده‌سازی ساده شده مقاله Choi & Kumar 2024
- ✓ ایجاد یک دوقلوی دیجیتال ساده برای یک شیروانی خاکی (نشست لحظه‌ای vs هشدار)
- ✓ استفاده از مدل زبانی بزرگ برای کنترل دفترچه محاسبات سازه یا ژئوتکنیک

Adv. App. of AI and DT 62



به دنبال آرزوهایم خواهم رفت
مگر نمیرم عهد بسته ام قبل از