

انتخاب مدل LLM

با توجه به اینکه محدودیت سخت افزاری داریم بر طبق آمار انتخابات ما به مدلهایی با نهایتاً ۷ بیلیون پارامتر محدود میشود. از این رو در مدلهایی با ۷ بیلیون پارامتر مدل مناسب را باید انتخاب کنیم.

از جمله مدلهای معروف قابل اجرا روی کولب موارد زیر موجود است:

Falcon-7b

LLama-2-7b

Mistral-7b

LLma-2-13b

Wizard-vicuna-13b

معیارهای سنجش در hugging face برای LLM ها موارد زیر اند:

ARC: ARC stands for "AI2 Reasoning Challenge". It's a dataset designed to test a model's ability to solve challenging reasoning problems. Performance on this dataset indicates how well a model can understand and apply logical reasoning skills.

HellaSwag: A dataset for evaluating a model's ability to generate coherent and contextually relevant completions for partially completed stories. It tests creativity and understanding of narrative structure.

MMLU: MMLU (Multi-Modal Language Understanding) is a benchmark focusing on models' abilities to understand and generate text based on visual inputs. It's part of the broader trend towards multimodal AI, where models process and integrate information from different modalities (text, images, audio).

TruthfulQA: A dataset designed to assess a model's truthfulness in answering questions. It includes questions that require distinguishing between factual information and misinformation, testing the model's ability to discern accurate information.

Winogrande: An extension of the Winograd Schema Challenge, which focuses on a model's ability to resolve ambiguous pronouns and prepositions in sentences. It's a test of a model's comprehension and inference capabilities.

GSM8K: GSM8K (General Sentiment Analysis) is a dataset for evaluating sentiment analysis models. It involves determining the emotional tone behind words, phrases, symbols, emoticons, etc., in texts.

با توجه به کارکرد مدل ما که پاسخگویی به سوالات بر اساس داک هست، معیار ARC از جهت سنجش منطق llm و TruthfulQA از جهت پرسش و پاسخ برای ما حائز اهمیت است. سایر موارد ارزیابی و امتیاز مدل در مسئله ما اولویت نیست.

با توجه به این دو معیار داریم:

Mistral

Model: Mistral-7B-Instruct-v0.1

Select Columns to Display:

- ☐ Average
- ☒ ARC
- ☐ HellaSwag
- ☐ MMLU
- ☒ TruthfulQA
- ☐ Winogrande
- ☐ GSM8K
- ☒ Type
- ☐ Architecture
- ☐ Precision
- ☐ Merged
- ☐ Hub License
- ☒ #Params (B)
- ☒ Hub
- ☐ Model sha

Model types:

- ☒ base merges and moerges
- ☐ fine-tuned on domain-specific datasets
- ☒ chat models (RLHF, DPO, IFT, ...)
- ☒ continuously pretrained
- ☒ pretrained

Precision:

- ☒ bfloat16
- ☒ float16
- ☒ 4bit
- ☒ 8bit
- ☒ GPTQ

Select the number of parameters (B): 7 to 10

Hide models:

- ☒ Private or deleted
- ☒ Contains a merge/moerge
- ☐ MoE
- ☒ Flagged

T	Model	ARC	TruthfulQA	Type	#Params (B)	Hub
	mistralai/Mistral-7B-Instruct-v0.1	54.52	56.28	chat models (RLHF, DPO, IFT, ...)	7	1442

Llama-13b

Model: llama-13b

Select Columns to Display:

- ☐ Average
- ☒ ARC
- ☐ HellaSwag
- ☐ MMLU
- ☒ TruthfulQA
- ☐ Winogrande
- ☐ GSM8K
- ☒ Type
- ☐ Architecture
- ☐ Precision
- ☐ Merged
- ☐ Hub License
- ☒ #Params (B)
- ☒ Hub
- ☐ Model sha

Model types:

- ☒ base merges and moerges
- ☐ fine-tuned on domain-specific datasets
- ☒ chat models (RLHF, DPO, IFT, ...)
- ☒ continuously pretrained
- ☒ pretrained

Precision:

- ☒ bfloat16
- ☒ float16
- ☒ 4bit
- ☒ 8bit
- ☒ GPTQ

Select the number of parameters (B): 7 to 10

Hide models:

- ☒ Private or deleted
- ☒ Contains a merge/moerge
- ☐ MoE
- ☒ Flagged

T	Model	ARC	TruthfulQA	Type	#Params (B)	Hub
	abhinand/tamil-llama-13b-instruct-v0.1	54.52	41.22	chat models (RLHF, DPO, IFT, ...)	13	6
	huggyllama/llama-13b	56.14	39.48	pretrained	13	136
	codellama/Codellama-13b-Instruct-hf	44.54	45.88	chat models (RLHF, DPO, IFT, ...)	13	137
	TheBloke/Codellama-13b-Instruct-fp16	44.62	45.88	chat models (RLHF, DPO, IFT, ...)	13	29
	OpenAssistant/codellama-13b-oasst-sft-v10	45.39	45.02	chat models (RLHF, DPO, IFT, ...)	13	65
	codellama/Codellama-13b-hf	40.87	43.79	pretrained	13	95
	codellama/Codellama-13b-Python-hf	32.59	44.59	chat models (RLHF, DPO, IFT, ...)	13	46

Wizard-Vecuna

LLM Benchmark

Metrics through time

About

FAQ

Submit

Search

wizard-vecuna-13b

Select Columns to Display:

☐ Average

☒ ARC

☐ HellaSwag

☐ MMLU

☒ TruthfulQA

☐ Winogrande

☐ GSM8K

☒ Type

☐ Architecture

☐ Precision

☐ Merged

☐ Hub License

☒ #Params (B)

☒ Hub

☐ Model sha

Model types

☒ base merges and moerges

☐ fine-tuned on domain-specific datasets

☒ chat models (RLHF, DPO, IFT, ...)

☒ continuously pretrained

☒ pretrained

Precision

☒ bfloat16

☒ float16

☒ 4bit

☒ 8bit

☒ GPTQ

Select the number of parameters (B)

7

10

Hide models

☒ Private or deleted

☒ Contains a merge/moerge

☐ MoE

☒ Flagged

Llama-2-7b

Search

Llama-2-7b

Select Columns to Display:

☐ Average

☒ ARC

☐ HellaSwag

☐ MMLU

☒ TruthfulQA

☐ Winogrande

☐ GSM8K

☒ Type

☐ Architecture

☐ Precision

☐ Merged

☐ Hub License

☒ #Params (B)

☒ Hub

☐ Model sha

Model types

☒ base merges and moerges

☐ fine-tuned on domain-specific datasets

☒ chat models (RLHF, DPO, IFT, ...)

☒ continuously pretrained

☒ pretrained

Precision

☒ bfloat16

☒ float16

☒ 4bit

☒ 8bit

☒ GPTQ

Select the number of parameters (B)

7

10

Hide models

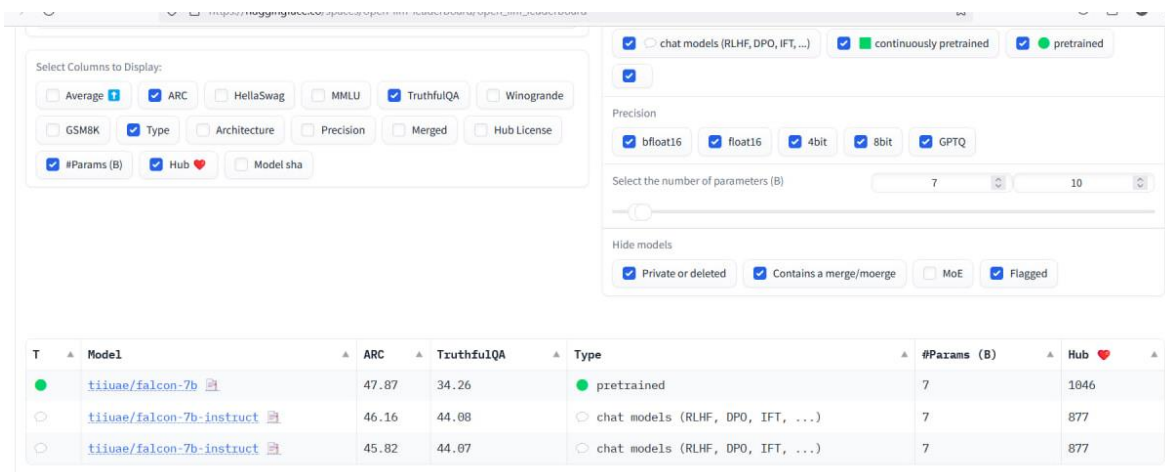
☒ Private or deleted

☒ Contains a merge/moerge

☐ MoE

☒ Flagged

Falcon-7b



The screenshot shows the Hugging Face Model Index interface. On the left, there are filters for 'Select Columns to Display' and 'Precision'. The 'Select Columns to Display' section includes checkboxes for Average, ARC, HellaSwag, MMLU, TruthfulQA, Winogrande, GSM8K, Type, Architecture, Precision, Merged, Hub License, #Params (B), Hub, and Model sha. The 'Precision' section includes checkboxes for bfloat16, float16, 4bit, 8bit, and GPTQ. On the right, there are filters for 'chat models (RLHF, DPO, IFT, ...)', 'continuously pretrained', 'pretrained', and 'Select the number of parameters (B)' with a slider from 7 to 10. Below these filters is a table of models.

T	Model	ARC	TruthfulQA	Type	#Params (B)	Hub
●	tiiuae/falcon-7b	47.87	34.26	● pretrained	7	1846
○	tiiuae/falcon-7b-instruct	46.16	44.08	○ chat models (RLHF, DPO, IFT, ...)	7	877
○	tiiuae/falcon-7b-instruct	45.82	44.07	○ chat models (RLHF, DPO, IFT, ...)	7	877

از بین موارد فوق mistral انتخاب مناسب تری است همچنین به استناد این [لینک](#) از مدل ۱۳ بیلیون پارامتری llama نیز عملکرد بهتری دارد.

انتخاب مدل Embedding

۱. ویژگی های مدل

معماری: این مدل بر اساس MiniLM (مدل زبان کوچک) است که نسخه کوچکتر و مقطر BERT (نمایش رمزگذار دوطرفه از ترانسفورماتور) است. MiniLM بسیاری از اثربخشی BERT را حفظ می کند در حالی که به طور قابل توجهی سبک تر و سریع تر است.

Sentence Transformers: این مدل بخشی از کتابخانه Sentence Transformers است که به طور خاص برای تولید جملات با کیفیت بالا طراحی شده است. این کتابخانه بر وظایف شباهت متنی تمرکز دارد، و آن را برای ایجاد جاسازی‌هایی ایده‌آل می‌سازد که معنای جملات را به جای کلمات منحصر به فرد به تصویر می‌کشد.

کارایی: "all-MiniLM-L6-v2" به دلیل تعادل بین عملکرد و راندمان محاسباتی شناخته شده است. از ۶ لایه ترانسفورماتور استفاده می کند و برای اجرای سریع حتی بر روی CPU ها بهینه شده است، و برای برنامه های بلادرنگ و سناریوهایی که منابع محاسباتی محدود هستند مناسب است.

۲. معیارهای عملکرد

مدل "all-MiniLM-L6-v2" بر روی چندین مجموعه داده معیار برای وظایف مختلف NLP ارزیابی شده است. در اینجا برخی از معیارهای مرتبط وجود دارد:

تشابه متنی معنایی (STS):

معیار STS (STS-B): این مدل به نمره همبستگی اسپیرمن بالایی، اغلب در حدود ۷۷-۸۰ درصد دست می یابد. این نشان دهنده عملکرد قوی در گرفتن شباهت معنایی بین جملات است.

جاسازی جملات با هدف عمومی:

میانگین وظایف انتقال (ارزیابی شده بر روی وظایف مختلف مانند طبقه بندی، خوشه بندی، و غیره): مدل به صورت رقابتی عمل می کند، اغلب در بین مدل های برتر با مزیت کارایی قابل توجهی رتبه بندی می شود.

معیارهای کارایی:

سرعت نسل جاسازی: این مدل می تواند جاسازی ها را با سرعت تقریبی ۱۲۰۰ جمله در ثانیه در یک GPU مدرن و حدود ۱۰۰ جمله در ثانیه در یک CPU استاندارد ایجاد کند که آن را برای برنامه های بلادرنگ بسیار کارآمد می کند.

۳. رتبه بندی مقایسه ای

در مقایسه با سایر مدل های تعبیه شده:

BERT و RoBERTa: در حالی که مدل های BERT و RoBERTa دقت بسیار بالایی در بسیاری از وظایف NLP ارائه می کنند، اما از نظر محاسباتی گران هستند. MiniLM-L6-v2 با دقت کمی پایین تر اما سرعت و کارایی بسیار بالاتر، به یک معامله خوب دست می یابد.

انواع SBERT: مدل "all-MiniLM-L6-v2" اغلب درست پایین تر از مدل های SBERT با عملکرد برتر مانند "all-roberta-large-v1" قرار می گیرد، اما ردپای محاسباتی به میزان قابل توجهی کاهش می یابد.

DistilBERT: مشابه DistilBERT از نظر اینکه یک نسخه تقطیر شده از یک مدل بزرگتر است، MiniLM عملکرد بهتری را در بسیاری از معیارها ارائه می کند و در عین حال کارایی مشابهی را حفظ می کند.

۴. مناسب برای پاسخگویی به سوالات PDF

برای یک سیستم پاسخگویی به پرسش PDF با استفاده از RAG، الزامات کلیدی برای یک مدل جاسازی عبارتند از:

درک معنایی بالا: برای درک دقیق معنی جملات یا پاراگراف ها در PDF.

کارایی: برای پردازش اسناد بزرگ بالقوه به سرعت و در زمان واقعی.

تطبیق پذیری: برای مدیریت انواع متن موجود در PDF، از متن ساختاریافته گرفته تا جملات پیچیده تر.

مدل "Sentence-transformers/all-MiniLM-L6-v2" این الزامات را به طور موثر برآورده می کند. توانایی آن برای ایجاد جاسازی های با کیفیت بالا که در عین حال از نظر محاسباتی کارآمد هستند، آن را به یک انتخاب عالی برای جاسازی متن در یک سیستم پاسخدهی به پرسش PDF تبدیل می کند.

انتخاب روش split برای خواندن pdf

```
pdf_folder_path = "."
documents = []

for file in os.listdir(pdf_folder_path):
    if file.endswith('.pdf'):
        pdf_path = os.path.join(pdf_folder_path, file)
        loader = PyPDFLoader(pdf_path)
        documents.extend(loader.load())

text_splitter_rc = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=100)
chunked_documents_rc = text_splitter_rc.split_documents(documents)
chunked_documents_rc
```

رویکرد RecursiveCharacterTextSplitter از کتابخانه Langchain برای تقسیم کارآمد و مؤثر اسناد به قطعات کوچکتر برای پردازش طراحی شده است. در اینجا به این می پردازیم که چرا این رویکرد به ویژه برای یک سیستم پاسخگویی به پرسش PDF مناسب است و چگونه با سایر روش های تقسیم بندی مقایسه می شود:

۱. دانه بندی و کنترل

اندازه قطعه و همپوشانی: RecursiveCharacterTextSplitter امکان کنترل دقیق بر اندازه تکه (مثلاً ۱۰۰۰ کاراکتر) و همپوشانی بین تکه ها (مثلاً ۱۰۰ کاراکتر) را فراهم می کند. این مفید است زیرا:

اندازه قطعه بهینه: اندازه تکه ای از ۱۰۰۰ کاراکتر تعادلی را بین اندازه کافی برای ثبت زمینه معنادار و کوچک بودن برای پردازش کارآمد توسط مدل هایی مانند MiniLM-L6-v2 ایجاد می کند. این تضمین می کند که هر تکه حاوی اطلاعات کافی برای ایجاد تعبیه های دقیق بدون تحت تأثیر قرار دادن مدل است.

همپوشانی برای حفظ متن: همپوشانی ۱۰۰ کاراکتر تضمین می کند که اطلاعات مهم در مرزهای تکه ها از بین نمی روند. این برای حفظ زمینه بسیار مهم است، زیرا همپوشانی به ایجاد تداوم بین تکه ها کمک می کند. این می تواند کیفیت تعبیه ها و دقت سیستم پاسخگویی سوال را بهبود بخشد.

۲. تقسیم بازگشتی

رویکرد سلسله مراتبی: RecursiveCharacterTextSplitter از یک رویکرد بازگشتی برای تقسیم متن استفاده می کند، که می تواند شامل شکستن متن در سطوح مختلف (به عنوان مثال، پاراگراف ها، جملات) قبل از توسل به تقسیم در سطح کاراکتر باشد. این چند مزیت دارد:

مرزهای طبیعی: با تلاش برای تقسیم در مرزهای طبیعی (به عنوان مثال، پاراگراف ها، جملات)، این روش یکپارچگی معنایی متن را بهتر از روش های تقسیم دلخواه حفظ می کند. این منجر به تکه هایی می شود که منسجم تر و از نظر محتوایی معنادارتر هستند.

مکانیسم بازگشتی: اگر مرزهای طبیعی در اندازه قطعه مشخص شده یافت نشد، روش می تواند به تقسیم در سطح کاراکتر بازگردد. این تضمین می کند که فرآیند تقسیم قوی است و می تواند ساختارهای مختلف سند را بدون شکست یا تولید تکه های بیش از حد بزرگ اداره کند.

۳. مقایسه با سایر روش های تقسیم

تقسیم با طول ثابت:

مزایا: ساده و سریع، اندازه تکه های یکنواخت را تضمین می کند.

معایب: می تواند در وسط جملات یا کلمات تقسیم شود و منجر به تکه هایی شود که انسجام معنایی ندارند. این می تواند کیفیت تعبیه ها و عملکرد سیستم پاسخگویی به سوال را کاهش دهد.

تقسیم جمله یا پاراگراف:

مزایا: مرزهای معنایی را حفظ می کند که منجر به تکه های منسجم می شود.

معایب: تکه های حاصل می توانند از نظر اندازه متفاوت باشند. پاراگراف ها یا جملات بسیار طولانی می توانند از حد پردازش مدل فراتر بروند، در حالی که موارد بسیار کوتاه می توانند برای پردازش ناکارآمد باشند. این تنوع می تواند خط لوله پردازش را پیچیده کند.

تقسیم پنجره کشویی:

مزایا: زمینه را از طریق پنجره های همپوشانی، مشابه RecursiveCharacterTextSplitter، حفظ می کند.

معایب: معمولاً از پنجره های با اندازه ثابت بدون توجه به مرزهای متن طبیعی استفاده می کند که منجر به تقسیم احتمالی میان جمله یا کلمه می شود. این می تواند منجر به تکه های منسجم کمتری در مقایسه با تقسیم بازگشتی شود.

۴. مناسب برای پاسخگویی به سوالات PDF

برای یک سیستم پاسخگویی به پرسش PDF، اطمینان از اینکه تکه های متن وارد شده به مدل تعبیه شده هم از نظر اندازه قابل مدیریت و هم از نظر محتوایی غنی هستند، بسیار مهم است. رویکرد RecursiveCharacterTextSplitter به خوبی با این الزامات مطابقت دارد:

تضمین انسجام: تقسیم متن در مرزهای طبیعی تا حد امکان برای حفظ معنای معنایی.

حفظ زمینه: استفاده از همپوشانی برای اطمینان از حفظ اطلاعات مهم در سراسر مرزهای تکه.

انعطاف پذیری: مدیریت ساختارهای مختلف سند به طور موثر، و آن را برای محتوای متنوعی که اغلب در فایل های PDF یافت می شود مناسب می کند.

انتخاب Prompt

```
template = ""Using the information contained in the context,
give a comprehensive answer to the question.
Respond only to the question asked, and provide concise and relevant responses.
Mention the number of the source document when relevant.
If the answer cannot be deduced from the context, state that you don't know.
Context: {context}
Question: {question}
your Answer: ""

prompt = PromptTemplate(
    template=template, input_variables=["context", "question"]
)
```

۱. وضوح و تمرکز

دستورالعمل برای تمرکز بر زمینه:

اعلان به طور صریح به مدل دستور می دهد تا از اطلاعات موجود در زمینه استفاده کند و اطمینان حاصل کند که پاسخ تولید شده بر اساس اسناد بازیابی شده است نه بر دانش قبلی یا توهومات مدل. این به حفظ دقت پاسخ ها کمک می کند.

پاسخ جامع:

دستورالعمل ارائه "پاسخ جامع" مدل را تشویق می کند تا پاسخی دقیق و کامل ارائه دهد و تمام جنبه های مربوط به سوال را از زمینه ارائه شده پوشش دهد.

۲. مختصر و مرتبط

پاسخ های مختصر:

درخواست به طور خاص پاسخ های مختصر را می خواهد. این تضمین می کند که مدل از پر حرفی های غیرضروری اجتناب می کند و به اطلاعات ضروری پایبند است و پاسخ ها را خواناتر و دقیق تر می کند.

ارتباط:

با تأکید بر ارتباط، اعلان مدل را راهنمایی می کند تا اطلاعات اضافی را فیلتر کند و صرفاً بر آنچه مربوط به سؤال است تمرکز کند. این باعث افزایش کیفیت و سودمندی پاسخ ها می شود.

۳. منبع

ذکر منبع سند:

گنجاندن دستورالعمل "ذکر شماره سند منبع در صورت لزوم" به پاسخ ها شفافیت و اعتبار می بخشد. این به کاربران امکان می دهد اطلاعات را به منبع اصلی آن ردیابی کنند، که برای تأیید صحت و قابل اعتماد بودن پاسخ بسیار مهم است.

۴. مدیریت عدم قطعیت

بیان صریح عدم قطعیت:

دستورالعمل بیان "نمی دانم" در صورتی که پاسخ را نمی توان از متن استنتاج کرد، برای حفظ یکپارچگی سیستم حیاتی است. هنگامی که متن پاسخ روشنی ارائه نمی دهد، مدل را از ایجاد اطلاعات جلوگیری می کند و در نتیجه از اطلاعات نادرست جلوگیری می کند.

۵. ساختار قالب

جداسازی متن و سوال:

جداسازی واضح متن و سوال در اعلان به مدل کمک می کند تا نقش های متمایز هر جزء را درک کند. این رویکرد ساختاریافته به مدل در تمرکز بر استخراج اطلاعات مرتبط از زمینه برای پاسخ دقیق به سؤال کمک می کند.