بررسی توهمات در مدل‌های زبانی

اعضای گروه: الهه بدلی، زینب یاسمنی، حسنا اویارحسینی، رضا حیدری، دانیال غریب، علیرضا امیری

چکیده

توهم در مدل‌های زبانی بزرگ باعث تولید اطلاعات نادرست یا بی‌پایه‌ می‌شود که در کاربرد‌های مختلف می‌تواند مشکل‌ساز باشد. در این مقاله، به بررسی راهکارهای مؤثر برای کاهش این مشکل در وظیفه پرسش و پاسخ در زبان فارسی پرداخته‌ایم. برای این منظور، از دو رویکرد روش بازیابی و تولید ترکیبی (RAG) و روش Finetune مدل استفاده کرده‌ایم. در رویکرد RAG ۱۰۰ مقاله از ویکی‌پدیا انتخاب شده است تا به عنوان منابع اطلاعاتی از آن استفاده شود. در این رویکرد با ایندکس‌گذاری و بازیابی داده‌ها، شباهت برداری بین منابع و سوال مورد نظر، به صورت سریع و دقیق انجام می‌شود تا در نهایت، پاسخ مناسبی تولید شود. این رویکرد بر روی دو نوع مدل زبانی مبتنی بر encoder و decoder بررسی شد که با تحلیل نتایج تاثیرگذاری بیشتر رویکرد RAG در مدل‌های مبتنی بر decoder مشخص گردید. از طرفی، در رویکرد دوم، مدل‌ها را از طریق Fine-Tuning تنظیم کردیم تا دقت آن‌ها در تولید پاسخ‌های دقیق‌تر و کمتر توهم‌آمیز افزایش یابد. نتایج این پژوهش نشان می‌دهد که هر دو رویکرد به طور مؤثری به کاهش توهم و بهبود کیفیت پاسخ‌های مدل‌های زبانی بزرگ کمک می‌کند.

کلمات کلیدی: **مدل‌های زبانی بزرگ، توهم، RAG، Fine tune**

1. تعریف مسئله و مقدمه

مدل‌های زبانی بزرگ (LLM) به عنوان یکی از پیشرفته‌ترین دستاوردهای هوش مصنوعی در سال‌های اخیر، توانسته‌اند توانایی‌های چشمگیری در تولید متن‌های پیچیده و انسانی‌گونه به نمایش بگذارند. این مدل‌ها با تکیه بر حجم عظیمی از داده‌های متنی، قادر به درک و تولید زبان به‌صورت پویا و مؤثر هستند. با این حال، یکی از چالش‌های اساسی این مدل‌ها که مانع از پذیرش گسترده و ایمن آنها در کاربردهای عملی می‌شود، مسئله‌ای است که به "توهم زایی" مشهور است.

توهم زایی در مدل‌های زبانی بزرگ به تولید محتوایی اشاره دارد که به ظاهر واقعی و منطقی است اما در واقع پایه و اساسی ندارد. این مسئله به ویژه در کاربردهای حساس مانند پزشکی، حقوق، و امور مالی می‌تواند خطر آفرین باشد، چرا که اطلاعات نادرست ممکن است به تصمیم‌گیری‌های اشتباه منجر شود. علی‌رغم پیشرفت‌های قابل توجه در زمینه هوش مصنوعی، مدل‌های زبانی هنوز از فهم واقعی و عمیق زبان فاصله دارند و در برخی موارد، اطلاعات غلط یا توهم‌زا تولید می‌کنند.

تحقیقات فراوانی برای شناسایی و کاهش این توهمات در مدل‌های زبانی بزرگ انجام شده است، اما بخش عمده‌ای از این تحقیقات بر روی مدل‌های زبان انگلیسی متمرکز بوده است. این در حالی است که برای زبان‌های دیگر، از جمله زبان فارسی، نیاز به پژوهش‌های بیشتری در این زمینه احساس می‌شود. با توجه به رشد روزافزون استفاده از مدل‌های زبانی بزرگ در زبان فارسی و اهمیت دقت و صحت در تولیدات متنی این مدل‌ها، پرداختن به مسئله توهم زایی در این حوزه از اهمیت ویژه‌ای برخوردار است.

پروژه‌ی حاضر به ارائه راهکارهایی برای تشخیص و برطرف کردن توهمات در مدل‌های زبانی بزرگ فارسی می‌پردازد و به دنبال ارائه راهکارهایی برای کاهش این مسئله در کاربردهای واقعی است. هدف از این پژوهش، ارتقاء دقت و کارایی مدل‌های زبانی بزرگ فارسی و فراهم کردن زمینه‌ای برای استفاده ایمن‌تر و گسترده‌تر از این فناوری در کاربردهای مختلف است.

1. پیشینه تحقیق

تحقیقات در زمینه شناسایی و کاهش توهمات در مدل‌های زبان بزرگ (LLMs) توجه زیادی را در تحقیقات اخیر به خود جلب کرده است. در ادامه به بررسی این کار ها در دوسته شناسایی توهمات و کاهش توهمات می پردازیم و در نهایت آشنایی با کارهای انجام شده در زمینه مدل های زبانی بزرگ برای زبان فارسی می پردازیم.

2.1. شناسایی توهمات

پیشرفت‌های اخیر در شناسایی توهمات در مدل‌های زبان بزرگ شامل روش‌های نوآورانه‌ای است. لی و همکاران (2022) [1] بنچمارک FactualityPrompt را توسعه دادند که برای اندازه‌گیری توهمات از خطاهای موجودیت نام‌دار (Hallucination NE (Named Entity) errors) و نسبت‌های انطباق با متن ویکی‌پدیا (Entailment ratios) استفاده می‌کند و نشان می‌دهد که مدل‌های بزرگ‌تر معمولاً عملکرد بهتری دارند. FActScore (مین و همکاران، 2023) [2] و SAFE (وی و همکاران، 2024) [3] ارزیابی‌های طولانی‌مدت را به ترتیب، با اعتبارسنجی حقایق اتمی و استفاده از پرسش‌های جستجوی تکراری، بهبود می‌بخشند. FacTool (چرن و همکاران، 2023) [4] فرآیند کاملی از بررسی واقعیت را در انواع مختلف وظایف اعمال می‌کند، در حالی که SelfCheckGPT (ماناکول و همکاران، 2023) سازگاری میان نمونه‌های مختلف را بدون استفاده از پایگاه‌های داده خارجی و با استفاده از روش های نمونه برداری ارزیابی می‌کند [5]. همچنین، TruthfulQA (لین و همکاران، 2021) [6] و SelfAware (یین و همکاران، 2023) [7] راست‌گویی مدل‌ها و نحوه برخورد آن‌ها با سوالات غیرقابل پاسخ را آزمایش می‌کنند و نشان می‌دهند که مدل‌های بزرگ‌تر معمولاً عملکرد بهتری دارند. در نهایت، آگرول و همکاران (2023) [8] نشان دادند که پرسش‌های غیرمستقیم، که به دنبال جزئیات درباره مراجع تولید شده هستند، در شناسایی توهمات مؤثرتر از پرسش‌های مستقیم هستند.

2.2. کاهش توهمات

تلاش‌ها برای کاهش توهمات در مدل‌های زبان بزرگ شامل روش‌های مختلفی است که به بهبود دقت و اعتبار خروجی‌های مدل کمک می‌کنند. روش‌های تقویت داده به بهبود پاسخ‌های مدل با ادغام اطلاعات متنی اضافی کمک می‌کنند. مدل‌های Retrieval-Augmented Generation (RAG) ]۹[ و (RARR) Retrofit Attribution using Research and Revision [10] از بازیابی مستندات برای بهبود پاسخ‌ها استفاده می‌کنند، در حالی که RARR بر روی مراحل تحقیق و تجدید نظر تمرکز می‌کند. FAVA ]11[ نیز از بازیابی و ویرایش مستندات استفاده می‌کند اما به آن تقویت داده‌های مصنوعی (synthetic data fine-tuning) را اضافه می‌کند. Rethinking with Retrieval (RR) ]۱۲[ و Self-RAG) Self-Reflective Retrieval-Augmented Generation) [13[ این رویکرد را با استفاده از prompting زنجیره‌ای تفکر و توکن‌های بازتابی (reflection tokens) برای اطمینان از دقت و کیفیت بالا بهبود می‌بخشند.

روش‌های prompting پیشرفته نیز نقش مهمی در کاهش توهمات ایفا می‌کنند.Chain-of-Verification (CoVe)]14[ دقت واقعی را با تولید پاسخ اولیه، ایجاد سوالات تایید و اصلاح پاسخ بر اساس پاسخ‌ها بهبود می‌بخشد. RECITE [15[ این را با اضافه کردن recitation بهبود می‌بخشد، جایی که مدل اطلاعات مربوط را قبل از تولید پاسخ تکرار می‌کند، هرچند که هنوز برخی اختلافات با حقیقت وجود دارد.

روش‌های fine-tuning نیز به کاهش توهمات کمک می‌کنند. لی و همکاران (2022) روش‌های TopicPrefix [16[ و Sentence Completion Loss را برای بهبود دقت واقعی معرفی کردند. FLAME (لین و همکاران، 2024) [17] ترکیبی از Supervised Fine-Tuning و Reinforcement Learning with Human Feedback) RLHF) را برای بهبود دقت واقعی ارائه می‌دهد، اگرچه RLHF نتایج مختلطی نشان داده است. Factuality Tuning (تیان و میچل و همکاران، 2024) [18] از روش‌های مبتنی بر مرجع و بدون مرجع برای تخمین حقیقت استفاده می‌کند. مدل‌هایی مانند WebGPT [19[ و GopherCite [20[ از داده‌های موتور جستجو و fine-tuning برای بهبود دقت استفاده می‌کنند.

2.3. مدل‌های زبان بزرگ فارسی

استفاده از مدل‌های زبان بزرگ (LLMs) در زبان فارسی نشان‌دهنده پتانسیل قابل توجهی برای ارتقاء پردازش زبان طبیعی در این زمینه زبانی است. مدل‌های چندزبانه، مانند BERT و GPT، در حال حاضر از فارسی پشتیبانی می‌کنند و تحقیقات اخیر بر پیشرفت مدل‌های خاص فارسی تمرکز کرده است. به عنوان مثال، پناهنده و همکاران (2023) چالش‌های مربوط به مجموعه‌های داده کوچک را با آموزش یک مدل تولید شعر کلاسیک فارسی با استفاده از معماری ترنسفورمر مورد بررسی قرار دادند. پورکمالی و همکاران (2024) [21] بر بهبود ترجمه ماشینی بین فارسی، انگلیسی و روسی از طریق معیارهای ارزیابی و روش‌های prompting بهبود یافته تمرکز کردند. عباس کوهی و همکاران (2024) [22] مطالعه‌ای از مدل‌های زبان بزرگ برای وظایف مختلف فارسی انجام دادند که ویژگی‌های منحصر به فرد زبان فارسی را نشان دادند. مرادبیکی و همکاران (2024) [23] سیستم‌های پاسخ‌گویی به سوالات محاوره‌ای فارسی (CQA) را با ادغام استخراج کلمات کلیدی متنی با مدل‌های زبان بزرگ بهبود دادند. قهرودی و همکاران (2024) چالش خیام [PersianMMLU) [24) را برای ارزیابی تسلط مدل‌های زبان بزرگ به فارسی معرفی کردند. با وجود این پیشرفت‌ها، هنوز شکاف قابل توجهی در تحقیق در مورد توهمات در مدل‌های زبان بزرگ فارسی وجود دارد. در حالی که این مدل‌ها پتانسیل قابل توجهی نشان می‌دهند، تحقیق بیشتری برای پرداختن و کاهش توهمات به‌ویژه در مدل‌های زبان بزرگ فارسی مورد نیاز است.

1. روش تحقیق

3.1. دادگان:

مجموعه داده پرسش و پاسخ فارسی (PersianQA) [25[ مجموعه‌ای از داده‌های درک مطلب است که بر روی ویکی‌پدیا فارسی تهیه شده است. این داده‌مجموعه که به صورت جمعی جمع‌آوری شده، شامل بیش از ۹۰۰۰ ورودی است. هر ورودی می‌تواند شامل سوالی باشد که پاسخ آن غیرممکن است یا سوالی با یک یا چندین پاسخ که در متن (زمینه) مورد نظر که سوال‌کننده سوال را مطرح کرده است، وجود دارد. مانند داده‌مجموعه SQuAD2.0، سوالات غیر ممکن یا غیر قابل پاسخ می‌توانند برای ایجاد سیستمی که "می‌داند که نمی‌داند پاسخ چیست"، استفاده شوند.

علاوه بر این، مجموعه داده دارای ۹۰۰ داده آزمایشی است. تمام افرادی که برای جمع‌آوری داده‌های این مجموعه مشارکت کرده اند، بومیان فارسی‌زبان هستند. همچنین شایان ذکر است که زمینه‌ها از تمام دسته‌های ویکی‌پدیا (تاریخی، مذهبی، جغرافیا، علم و غیره) جمع‌آوری شده‌اند. در حال حاضر، هر زمینه شامل ۷ جفت سوال با یک پاسخ و ۳ سوال غیرممکن است.

همان‌طور که قبلاً اشاره شد، این داده‌مجموعه از داده‌مجموعه معروف SQuAD2.0 الهام گرفته شده و با آن سازگار است و می‌تواند با آن ترکیب شود. اما این همه ماجرا نیست؛ داده‌مجموعه مذکور دارای برخی مزایای نسبی نسبت به منبع الهام اولیه است که برخی از آن‌ها به شرح زیر است: زمینه‌های طولانی‌تر، افزایش تعداد مقالات (با وجود داده‌های کمتر)، سوالات بیشتر در هر زمینه (۷ سوال در مقایسه با ۵ سوال)، شامل ورودی‌های غیررسمی ("محاوره‌ای")، پاسخ‌های متنوع‌تر (نام‌ها، مکان‌ها، تاریخ‌ها و غیره)

اطلاعات آماری این دادگان به صورت دقیق تر در زیر آمده است:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Split** | **# of instances** | **# of unanswerable** | **avg. question length** | **avg. paragraph length** | **avg. answer length** |
| Train | 9,000 | 2,700 | 8.39 | 224.58 | 9.61 |
| Test | 938 | 280 | 8.02 | 220.18 | 5.99 |

نمونه ای از این دادگان را در ادامه میتوان مشاهده کرد:

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Context** | **Question** | **Answer** |
| خوب، بد، زشت | خوب، بد، زشت یک فیلم درژانر وسترن اسپاگتی حماسی است که توسط سرجو لئونه در سال ۱۹۶۶ در ایتالیا ساخته شد. زبانی که بازیگران این فیلم به آن تکلم می‌کنند مخلوطی از ایتالیایی و انگلیسی است. این فیلم سومین (و آخرین) فیلم از سه‌گانه دلار (Dollars Trilogy) سرجو لئونه است. این فیلم در حال حاضر در فهرست ۲۵۰ فیلم برتر تاریخ سینما در وب‌گاه IMDB با امتیاز ۸٫۸ از ۱۰، رتبهٔ هشتم را به خود اختصاص داده‌است و به عنوان بهترین فیلم وسترن تاریخ سینمای جهان شناخته می‌شود. «خوب» (کلینت ایستوود، در فیلم، با نام «بلوندی») و «زشت» (ایلای والاک، در فیلم، با نام «توکو») با هم کار می‌کنند و با شگرد خاصی، به گول زدن کلانترهای مناطق مختلف و پول درآوردن از این راه می‌پردازند. «بد» (لی وان کلیف) آدمکشی حرفه‌ای است که به‌خاطر پول حاضر به انجام هر کاری است. «بد»، که در فیلم او را «اِنجل آیز (اِینجل آیز)» (به انگلیسی: Angel Eyes) صدا می‌کنند. به‌دنبال گنجی است که در طی جنگ‌های داخلی آمریکا، به دست سربازی به نام «جکسون»، که بعدها به «کارسون» نامش را تغییر داده، مخفی شده‌است. | در فیلم خوب بد زشت شخصیت ها کجایی صحبت می کنند؟ | مخلوطی از ایتالیایی و انگلیسی |
| چهارشنبه‌سوری | چهارشنبه‌سوری یکی از جشن‌های ایرانی است که از غروب آخرین سه‌شنبه ی ماه اسفند، تا پس از نیمه‌شب تا آخرین چهارشنبه ی سال، برگزار می‌شود و برافروختن و پریدن از روی آتش مشخصهٔ اصلی آن است. این جشن، نخستین جشن از مجموعه جشن‌ها و مناسبت‌های نوروزی است که با برافروختن آتش و برخی رفتارهای نمادین دیگر، به‌صورت جمعی در فضای باز برگزار می‌شود. به‌گفته ابراهیم پورداوود چهارشنبه‌سوری ریشه در گاهنبارِ هَمَسْپَتْمَدَم زرتشتیان و نیز جشن نزول فروهرها دارد که شش روز پیش از فرارسیدن نوروز برگزار می‌شد. احتمال دیگر این است که چهارشنبه‌سوری بازمانده و شکل تحول‌یافته‌ای از جشن سده باشد، که احتمال بعیدی است. علاوه برافروختن آتش، آیین‌های مختلف دیگری نیز در بخش‌های گوناگون ایران در زمان این جشن انجام می‌شوند. برای نمونه، در تبریز، مردم به چهارشنبه‌بازار می‌روند که با چراغ و شمع، به‌طرز زیبایی چراغانی شده‌است. هر خانواده یک آینه، دانه‌های اسفند، و یک کوزه برای سال نو خریداری می‌کنند. همه‌ساله شهروندانی از ایران در اثر انفجارهای ناخوشایند مربوط به این جشن، کشته یا مصدوم می‌شوند. | نام جشن آخرین شنبه ی سال چیست؟ | No Answer |

ما از این داده برای بررسی عملکرد مدل به خصوص در زمینه تشخیص توهم استفاده کرده ایم.

در بخش RAG که در ادامه توضیح داده می شود از دیتاست PersianWiki ]26[ که شامل تمامی مقالات فارسی تا تاریخ 12 مرداد 1399 می باشند استفاده کرده ایم. برای مثال یک نمونه داده از این دیتاست به صورت زیر می باشد.

عنوان مقاله: سعدی

ابومحمّد مُشرف الدین مُصلِح بن عبدالله بن مشرّف، متخلص به سعدی (۶۰۶ – ۶۹۰ هجری قمری)، شاعر و نویسنده پارسی گوی ایرانی است. اهل ادب به او لقب "استادِ سخن"، "پادشاهِ سخن"، "شیخِ اجلّ" و حتی به طور مطلق، "استاد" داده اند. او در نظامیه بغداد – که مهم ترین مرکز علم و دانش جهان اسلام در آن زمان به حساب می آمد – تحصیل و پس از آن به عنوان خطیب به مناطق مختلفی از جمله شام و حجاز سفر کرد. سعدی سپس به زادگاه خود، شیراز، برگشت و تا پایان عمر در آن جا اقامت گزید. آرامگاه وی در شیراز واقع شده است که به سعدیه معروف است. بیشتر عمر او مصادف با حکومت اتابکان فارس در شیراز و هم زمان با حمله مغول به ایران و سقوط بسیاری از حکومت های وقت نظیر خوارزمشاهیان و عباسیان بود. البته سرزمین فارس، به واسطه تدابیر ابوبکر بن سعد، ششمین و معروف ترین اتابکان سَلغُری شیراز، از حمله مغول در امان ماند. همچنین قرن ششم و هفتم هجری مصادف با اوج گیری تصوف در ایران بود و تأثیر این جریان فکری و فرهنگی در آثار سعدی قابل ملاحظه است. نظر اغلب سعدی پژوهان بر این است که سعدی تحت تأثیر آموزه های مذهب شافعی و اشعری و بنابراین تقدیرگرا است. در مقابل، نشانه هایی از ارادت وی به خاندان پیامبر اسلام مشاهده می شود. سعدی بیش از آن که تابع اخلاق به صورت مطلق و فلسفی آن باشد، مصلحت اندیش است و ازین رو اصولاً نمی تواند طرفدار ثابت و بی چون وچرای قاعده ای باشد که احیاناً در جای دیگری آن را بیان کرده است. برخی از نوگرایان معاصر ایران آثار او را غیراخلاقی، بی ارزش، متناقض و ناهماهنگ قلمداد کرده اند…

و همچنین مشخصات این دیتاست به صورت زیر است:

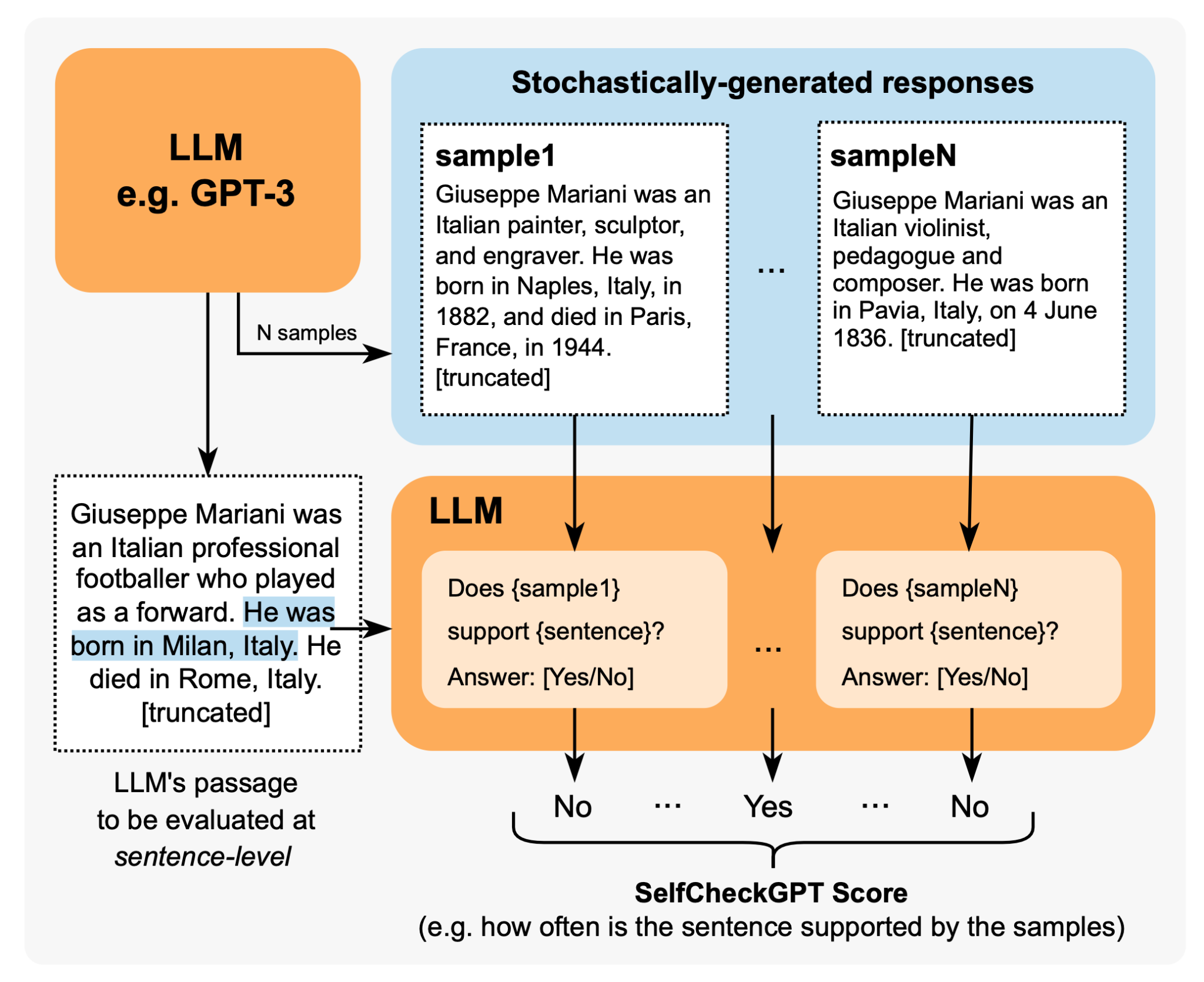
|  |  |  |
| --- | --- | --- |
| **Articles** | **Sentences** | **Tokens** |
| 739870 | 4004765 | 94002094 |

این دیتاست اطلاعات اضافه تر راجع به هر موضوع در دیتاست اول را به ما میدهد زیرا متون دیتاست اول فقط خلاصه شده چند پاراگراف اول هستند. در نهایت نیز تعدادی سوال و جواب از دیتاست دوم به صورت دستی تهیه شده است تا برای بررسی راهکار های ارائه شده به عنوان معیار استفاده شود که در ادامه تعدادی از آن ها را میبینید:

|  |  |  |
| --- | --- | --- |
| **title** | **question** | **answer** |
| شرکت فولاد مبارکه اصفهان | شرکت فولاد مبارکه در حال حاضر مواد اولیۀ چه صنایعی را تامین میکند؟ | فولاد مبارکۀ اصفهان در حال حاضر مواد اولیۀ بیش از ۳٬۰۰۰ کارخانه و کارگاه تولیدی کشور در صنایع لوله و پروفیل، نفت و گاز و پتروشیمی، خودرو، موتورهای الکتریکی، لوازم خانگی، فلزی سبک و سنگین، ساختمانی، غذایی، کشتی سازی و سازه های دریایی و ... را تامین می نماید. |
| ربات | چه کسی اولین بار از کلمه ربات استفاده کرد؟ | نویسنده اهل کشور چک برای اولین بار از کلمه ربات در نمایش نامه خود به عنوان آدم مصنوعی استفاده کرد |
| علوم رایانه | چه دانشگاه‌هایی در ایران رشته علوم کامپیوتر را دادند؟ | در کنار آن علوم کامپیوتر که متاسفانه آشنایی کمی با آن وجود دارد، بیشتر روی مسائل نظری متمرکز است و در واقع چیزی شبیه علوم کامپیوتر نظری در کشورهایی نظیر ایالات متحده است.  در رتبه بندی شانگهای، دانشگاه های تهران، صنعتی امیرکبیر و شریف هر کدام برای مدّت یک سال در بین' ۲۰۰ دانشگاه برتر جهان' در زمینه علوم کامپیوتر بوده اند، که این رده بندی بیشتر بر اساس میزان مقاله های منتشر شده در این زمینه می باش |
| کتاب | کتاب دیجیتال با چه وسیله ای خوانده میشود؟ | کتاب دیجیتالی ("e-Book") است که کتاب در قالب بسته های نرم افزاری مانندِ پی دی اف یا ئی پاب ارائه می شود که وسایلی نیز به نام کتابخوان الکترونیکی مانندِ آمازون کیندل برای خواندن آن به کار می رود |

3.2. تشخیص توهم:

برای تشخیص توهم ما از روش ارائه شده تحت عنوان SelfCheckGPT ]۵[ استفاده کردیم. در این روش یک prompt واحد چندین بار به LLM داده می‌شود و سمپل‌های خروجی از LLM گرفته می‌شود. اگر مدل، پاسخ مربوطه را بر اساس یک واقعیت بیان کند، پاسخ‌ها با هم سازگار خواهد بود. در صورتی که مدل پاسخ را ندارد و بر اساس توهم پاسخ بدهد محتوای تولید شده در هر نمونه متفاوت از بقیه پاسخ‌ها خواهد بود. بر همین اساس معیار selfcheckGPT در اصل میزان عدم سازگاری پاسخ‌ها را اندازه می‌گیرد.

****

3.3. برطرف کردن توهم:

برای برطرف کردن توهم دو روش RAG و Fine tuning بررسی شده است.

**۳.۳.۱. RAG**

همان‌طور که گفته شد استفاده از روش RAG یک راه برای کاهش توهم است. به طور کلی در این روش، ما از مجموعه داده ویکی‌پدیا استفاده کردیم و ۱۰۰ مقاله را انتخاب نمودیم. به طور متناظر، داده‌های مربوط به این مقالات را از دیتای PersianQA جدا کرده و روی آن‌ها کار کردیم.

مدل‌های مورد استفاده برای ایجاد بردارهای embedding از جمله کلیدی‌ترین اجزای فرآیند جستجوی معنایی هستند. در اینجا به توضیح سه مدل رایج که در این زمینه استفاده می‌شوند، می‌پردازیم:

۱. **Paraphrase-MiniLM-L6-v2:** این مدل یکی از مدل‌های کوچک‌تر سری MiniLM است که توسط Microsoft توسعه داده شده و برای وظایف مرتبط با جستجوی معنایی و تشخیص شباهت جملات بهینه شده است. به دلیل سرعت و کارایی بالا، این مدل برای کاربردهای مرتبط با جستجو و بازیابی اطلاعات در مقیاس بزرگ مناسب است.

۲. **Distiluse-base-multilingual-cased:** این مدل نسخه فشرده‌شده (distilled) از مدل USE (Universal Sentence Encoder) است که به صورت چند زبانه (multilingual) آموزش داده شده است. این مدل از BERT به عنوان مدل پایه استفاده می‌کند، اما به صورت فشرده‌شده و سبک‌تر طراحی شده است که باعث کاهش زمان پردازش و افزایش سرعت می‌شود. مدل cased است، به این معنا که به حروف بزرگ و کوچک حساس است. پشتیبانی از چندین زبان را دارد که آن را برای کاربردهای چند زبانه بسیار مفید می‌کند. این مدل در جستجوی معنایی چند زبانه، تطبیق جملات در زبان‌های مختلف و کاربردهایی که نیاز به پشتیبانی از زبان‌های متنوع دارند، کاربرد دارد.

۳. **LABSE (Language-Agnostic BERT Sentence Embedding) :** مدل LABSE یک مدل BERT است که توسط Google توسعه داده شده و به صورت زبان‌خنثی (language-agnostic) آموزش دیده است. به این معنا که می‌تواند جملات را بدون توجه به زبان آن‌ها به بردارهایی در فضای معنایی مشترک تبدیل کند. این مدل برای ایجاد embedding‌های جمله‌ای که در چندین زبان به طور یکنواخت و مشابه عمل می‌کنند، بهینه شده است. به دلیل زبان‌خنثی بودن، جملات با معانی مشابه در زبان‌های مختلف را به بردارهای نزدیک به هم نگاشت می‌کند. از مدل BERT برای استخراج ویژگی‌های زبانی استفاده می‌کند، اما آموزش دیده است تا بتواند جملات در زبان‌های مختلف را به صورت مشترک در فضای برداری تعبیه کند. LABSE برای کاربردهای چند زبانه، تطبیق متون در زبان‌های مختلف، و جستجو و بازیابی اطلاعات در محیط‌های چندزبانه بسیار مناسب است.

این مدل‌ها به طور کلی برای بهبود فرآیند جستجو و بازیابی اطلاعات در سیستم‌های چندزبانه و معنایی بسیار مؤثر هستند. در ادامه برای جستجوی شباهت بین embeddingهای ویکی پدیا و embedding سوال از شاخص FAISS)Facebook AI Similarity Search) استفاده شده است. شاخص FAISS یک ابزار قدرتمند برای جستجوی شباهت برداری است که به ویژه برای مدیریت و بازیابی داده‌های با ابعاد بالا طراحی شده است. این شاخص در مجموعه‌های بزرگی از بردارها (مانند embeddings) به سرعت، نزدیک‌ترین همسایه‌ها (nearest neighbors) را می‌یابد. FAISS بردارهای داده را به گونه‌ای ایندکس می‌کند که جستجوی برداری روی آن‌ها سریع و کارآمد باشد. برای این کار، از روش‌های مختلفی برای فشرده‌سازی و ایندکس‌گذاری داده‌ها استفاده می‌کند. این روش‌ها در ادامه آمده‌اند.

1. **Flat (مسطح):** این ساده‌ترین نوع ایندکس است که بردارها را بدون هیچ فشرده‌سازی ایندکس می‌کند و از یک جستجوی خطی برای پیدا کردن نزدیک‌ترین همسایه‌ها استفاده می‌کند. این روش به حافظه زیادی نیاز دارد و برای مجموعه داده‌های کوچک یا برای انجام جستجوی دقیق (exhaustive search) مناسب است.
2. **IVF)Inverted File Index):** این روش بردارها را به چندین خوشه تقسیم می‌کند و هر بردار را به نزدیک‌ترین خوشه‌اش اختصاص می‌دهد. در هنگام جستجو، تنها خوشه‌های مرتبط بررسی می‌شوند که سرعت جستجو را به طور چشم‌گیری افزایش می‌دهد.
3. **PQ )Product Quantization):** یک تکنیک فشرده‌سازی است که بردارها را به قطعات کوچک‌تر تقسیم کرده و هر قطعه را به صورت جداگانه فشرده می‌کند. این روش به طور قابل‌توجهی میزان حافظه مصرفی را کاهش می‌دهد و سرعت جستجو را افزایش می‌دهد.
4. **HNSW)Hierarchical Navigable Small World graphs):** این روش یک ساختار گرافی برای جستجوی کارآمد نزدیک‌ترین همسایه‌ها ایجاد می‌کند که به ویژه برای داده‌های با ابعاد بسیار بالا مناسب است.

علاوه بر این ما از معیار semantic search نیز استفاده کردیم. این معیار برای انجام جستجوی معنایی در embeddingsها استفاده می‌شود. این معیار بر اساس فاصله اقلیدسی دو بردار یا فاصله کسینوسی آن‌ها تعریف می‌شود.

ما مدل‌های مختلفی به عنوان مدل پایه بررسی کردیم. مدل‌های مورد بررسی در دو دسته مبتنی بر encoder و مدل‌های مبتنی بر decoder بررسی شده‌اند.

مدل‌های مبتنی بر encoder شامل موارد زیر است:

1. Bert-base-multilingual-xquad: مدل fine tune‌شده بر روی مجموعه داده PersianQA
2. marzinouri/parsbert-finetuned-persianQA: مدل Bert که با parsbert‌ بروی مجموعه داده PersianQA فاین تیون شده
3. Bert-base-multilingual-cased: مدل Bert چند زبانه
4. pedramyazdipoor/parsbert\_question\_answering\_PQuAD: مدل parsbert که بر روی مجموعه داده پرسش و پاسخ PQuAD فاین تیون شده

مدل مبتنی بر decoder مدل Mistral است که برای وظیفه پرسش و پاسخ فاین تیون شده است.

1. MehdiHosseiniMoghadam/AVA-Mistral-7B-V2

۳.۳.۲. **Prompt-based fine-tuning**

در روش Fine-tuning مبتنی بر Prompt، تلاش کردیم با طراحی مناسب Prompt‌ها، از ابهام در متون تولیدی مدل بکاهیم. این روش به‌ویژه در سناریوهایی که مدل باید به سؤالات خاصی پاسخ دهد، اهمیت دارد. به طور مشخص، در این روش ما از اطلاعات موجود در دیتاست بهره بردیم؛ دیتاست ما شامل بازه‌های مکانی مشخصی از پاسخ مربوطه به هر سوال ( که با نام start\_span در دیتاست که بصورت یک key, value از ستون answer که یک دیکشنری هست می باشد) است که پاسخ‌ها باید از آنجا شروع شوند. در واقع پاسخ ها دقیقا و عینا از متن استخراج می شوند و مدل در مواجهه با سوالات این دیتاست نیازی به استفاده از دانش پیشین و یا تولید جواب مفهومی و یا از نوع خلاصه سازی ندارد، بلکه پاسخ هر سوال یک بازه مشخص از متن داده شده است.

با توجه به این ویژگی، Prompt‌ها را به گونه‌ای طراحی کردیم که از مدل می خواهد در خط اول پاسخش بازه‌ای را که پاسخ در آن قرار دارد، به صورت مبتنی بر کاراکتر (character-based) مشخص کند. همچنین در خطوط بعدی، مدل باید پاسخ دقیق را از متن استخراج و ارائه دهد.این تغییر در طراحی Prompt، به ما کمک کرد تا اطلاعات دقیق‌تری را از مدل دریافت کنیم و مدل را از تولید پاسخ متوهمانه در رابطه با پرسش های دیتاست دورتر سازیم.

برای اعمال این روش، دیتاست اصلی را با اضافه کردن دو ستون جدید گسترش دادیم. یکی از این ستون‌ها برای ذخیره Prompt اختصاصی هر سؤال بود، و ستون دیگر برای ذخیره پاسخ‌ها به همراه بازه مشخص شده در Prompt بود. در این ستون دوم، بازه مربوط به ابتدای پاسخ عادی دیتاست اضافه شد.

پس از انجام این تغییرات، دیتاست جدید را برای Fine-tuning مدل پایه استفاده کردیم. همچنین برای مقایسه بهتر نتایج تست ها را فقط برای وقتی که prompt مخصوص به نمونه را ایجاد کردیم را نیز گرفتیم که در بخش نتایج مربوط به fine-tuning آمده است. در این حالت درواقع بدون آموزش دادن مدل پایه برروی دیتاست تولید شده، صرفا دیتاست تست را به آن دادیم ولی بجای پرسیدن prompt معمول، prompt ای که در این بخش طراحی کردیم را به مدل دادیم. که درواقع نوعی prompt engineering بدون فاین تیون و ترینینگ می باشد که نتایج آن نیز در کنار سایر نتایج آمده است.

1. نتایج

۴.۱. نتایج رویکرد **RAG**

نتایج اجرای روش RAG بر روی مجموعه داده منتخب و مدل‌های معرفی شده در بخش قبل، در ادامه در جدول زیر آمده است. به ازای هر مدل پایه، مدل embedding متناظر و نحوه قسمت‌بندی (chunk)، معیار استفاده شده و در نهایت نتایج RAG‌ بر اساس شاخص FAISS و معیار semantic search گزارش شده است. به دلیل زمان طولانی اجرا در مدل‌ مبتنی بر decoder، نتایج برای ۲۰ نمونه ابتدایی گزارش شده است. همان‌طور که از نتایج مشخص است، مدل RAG بیشترین تاثیر خود را در مدل‌ مبتنی بر decoder‌ می‌گذارد.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RAG selfcheck**  **With semantic search** | **RAG selfcheck**  **With FAISS** | **Similarity metric**  **(FAISS)** | **[chuck size, overlap, bigger than]** | **Embedding model** | **Base model self-check** | **Base model** | **Type of model** |
| 0.57 | 0.59 | IndexFlatL2 | [500,100,300] | paraphrase-MiniLM-L6-v2 | 0.57 | bert-base-multilingual-xquad | Encoder based Models |
| - | 0.43 | IndexFlatL2 | [1000,100,800] | distiluse-base-multilingual-cased | 0.33 | marzinouri/parsbert-finetuned-persianQA |
| - | 0.37 | IndexFlatL2 | [500,100,300] | paraphrase-MiniLM-L6-v2 | 0.33 | marzinouri/parsbert-finetuned-persianQA |
| **0.35** | **0.34** | **IndexIVFFlat** | **[500,100,300]** | **paraphrase-MiniLM-L6-v2** | **0.33** | **marzinouri/parsbert-finetuned-persianQA** |
| - | 0.698 | IndexIVFFlat | [700,100,300] | paraphrase-MiniLM-L6-v2 | 0.698 | bert-base-multilingual-cased |
| 0.70 | 0.694 | IndexFlatL2 | [700,100,300] | paraphrase-MiniLM-L6-v2 | 0.694 | pedramyazdipoor/parsbert\_question\_answering\_PQuAD |
| **0.70** | **-** | **semantic search** | **[500,100,300]** | **LABSE** | **0.74** | **MehdiHosseiniMoghadam/AVA-Mistral-7B-V2** | Decoder based model |

* 1. **نتایج Prompt-based fine-tuning:**

به طور کلی و در مجموع نتایج این بخش نشان دهنده اثربخشی و کارآمد بودن روش مهندسی کردن promptها برای مدل می باشد و بهبود در عملکرد مدل در مقابله با توهم زایی محسوس می باشد.

نتایج روش Prompt-based fine-tuning قبل و بعد از فاین تیون کردن در جدول زیر قابل مشاهده است. برای مقایسه از متریک [SelfCheckGpt [5 بر روی دیتای تست قبل و بعد از فاین تیون کردن استفاده شده است که مقادیر آنها در ذیل آمده است.

توجه کنید به علت بزرگ بودن مدل مورد استفاده و احتیاج به زمان و منابع سخت افزاری زیاد برای کار با مدل، تنها 20 نمونه از دیتاست برای evaluation در نظر گرفته شده است.

|  |  |  |  |
| --- | --- | --- | --- |
| **Fine-tuned model with prompt SelfCheckGpt**  **Score** | **Base model with prompt SeflCheckGpt Score** | **Base model SelfCheckGpt Score** | **Base model** |
| 0.9067 | 0.6856 | 0.74 | MehdiHosseiniMoghadam/AVA-Mistral-7B-V2 |

همچنین اطلاعات کلی در رابطه با نحوه آموزش و پارامتر های مختلف برای fine-tuning و همچنین منابع استفاده شده برای این بخش در ذیل قابل بررسی است.

|  |  |
| --- | --- |
| **2** | **# of Training Epochs** |
| 30 min | **Time Required for fine tuning** |
| 9e-5 | **Learning Rate** |
| 40 | **# of Samples for training** |
| 0.01 | **Weight Decay** |
| 2 | **Train Batch Size** |
| 2 | **Val Batch Size** |

در مجموع همانطور که از نتایج پیداست درحالت prompt engineering و بدون استفاده از فاین تیون کردن مدل برروی دیتاست، نتایج self check Gpt Score بهبود یافته اند و عدم قطعیت کم شده است. اما در حالت فاین تیونینگ عدم قطعیت بالاتر رفته است و ترینینگ موثر واقع نشده است. باید در نظر گرفت که به علت بزرگ بودن و سنگین بودن مدل مورد استفاده و تعداد پارامتر های بسیار زیاد آن و نداشتن سخت افزار و منابع کافی برای ترینینگ، ما تنها توانستیم آن را برای 2 ایپاک و تنها برروی حدود 40 نمونه از دیتاست آموزش دهیم، احتمالا دلیل عملکرد معکوس مدل هم همین است، چه بسا اگر محدودیت های کمتری برای آموزش وجود می داشت فاین تیونینگ بهترین نتیجه را در میان این 3 میگرفت.

**افزایش امتیاز SelfCheckGPT پس از تنظیم مدل تا حدودی قابل پیش‌بینی بود، با توجه به شرایط موجود. فرآیند تنظیم مدل بر روی یک مجموعه داده نسبتاً کوچک انجام شد که احتمالاً منجر به بیش‌برازش شد و باعث شد مدل به طور بیش از حد بر روی آن داده‌های محدود تخصص پیدا کند. این تخصص‌گرایی معمولاً منجر به کاهش قابلیت تعمیم‌دهی می‌شود و باعث افزایش تولید اطلاعات نادرست یا توهمات می‌گردد. علاوه بر این، پیچیدگی یک مدل بزرگ مانند Mistral می‌تواند این مشکلات را تشدید کند، به‌ویژه زمانی که مجموعه داده‌های تنظیم مدل فاقد تنوع کافی باشد.**

**در نهایت اما در بخش prompt engineering نتایج تا حد ملموسی بهبود یافته اند پس احتمالا ایده کلی در صورت امکان پیاده سازی کامل و کافی، موفق واقع شود.**

1. نتیجه گیری

در این مقاله مسئله تشخیص و تصحیح توهم در مدل‌های زبانی بزرگ ارائه شده است. برای تشخیص از معیار selfcheckGPT استفاده شده که میزان عدم سازگاری پاسخ مدل را می‌سنجد. برای بخش تصحیح توهم از دو رویکرد RAG‌ و finetune کردن مدل استفاده شده است. رویکرد RAG بر روی مدل‌های مبتنی بر decoder‌ بیشترین تاثیر را دارد.

منابع

[1] Lee, Nayeon, et al. "Factuality enhanced language models for open-ended text generation." *Advances in Neural Information Processing Systems* 35 (2022): 34586-34599.

[2] Min, Sewon, et al. "Factscore: Fine-grained atomic evaluation of factual precision in long form text generation." arXiv preprint arXiv:2305.14251 (2023).

[3] Wei, Jerry, et al. "Long-form factuality in large language models." *arXiv preprint arXiv:2403.18802* (2024).

[4] Chern, I., et al. "FacTool: Factuality Detection in Generative AI--A Tool Augmented Framework for Multi-Task and Multi-Domain Scenarios." *arXiv preprint arXiv:2307.13528* (2023).

[5] Manakul, Potsawee, Adian Liusie, and Mark JF Gales. "Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models." *arXiv preprint arXiv:2303.08896* (2023).

[6] Lin, Stephanie, Jacob Hilton, and Owain Evans. "Truthfulqa: Measuring how models mimic human falsehoods." *arXiv preprint arXiv:2109.07958* (2021).

[7] Yin, Zhangyue, et al. "Do Large Language Models Know What They Don't Know?." arXiv preprint arXiv:2305.18153 (2023).

[8] Agrawal, Ayush, et al. "Do Language Models Know When They're Hallucinating References?." *arXiv preprint arXiv:2305.18248* (2023).

[9] Lewis, Patrick, et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." *Advances in Neural Information Processing Systems* 33 (2020): 9459-9474.

[10] Gao, Luyu, et al. "Rarr: Researching and revising what language models say, using language models." arXiv preprint arXiv:2210.08726 (2022).

[11] Mishra, Abhika, et al. "Fine-grained hallucination detection and editing for language models." *arXiv preprint arXiv:2401.06855* (2024).

[12] He, Hangfeng, Hongming Zhang, and Dan Roth. "Rethinking with retrieval: Faithful large language model inference." *arXiv preprint arXiv:2301.00303* (2022).

[13] Asai, Akari, et al. "Self-rag: Learning to retrieve, generate, and critique through self-reflection." arXiv preprint arXiv:2310.11511 (2023).

[14] Dhuliawala, Shehzaad, et al. "Chain-of-verification reduces hallucination in large language models." arXiv preprint arXiv:2309.11495 (2023).

[15] Sun, Zhiqing, et al. "Recitation-augmented language models." *arXiv preprint arXiv:2210.01296* (2022).

[16] Lee, Nayeon, et al. "Factuality enhanced language models for open-ended text generation." Advances in Neural Information Processing Systems 35 (2022): 34586-34599.

[17] Lin, Sheng-Chieh, et al. "Flame: Factuality-aware alignment for large language models." arXiv preprint arXiv:2405.01525 (2024).

[18] Tian, Katherine, et al. "Fine-tuning language models for factuality." arXiv preprint arXiv:2311.08401 (2023).

[19] Nakano, Reiichiro, et al. "Webgpt: Browser-assisted question-answering with human feedback." arXiv preprint arXiv:2112.09332 (2021).

[20] Menick, Jacob, et al. "Teaching language models to support answers with verified quotes, 2022." URL https://arxiv. org/abs/2203.11147 (2022).

[21] Panahandeh, Amir, Hanie Asemi, and Esmaeil Nourani. "TPPoet: Transformer-Based Persian Poem Generation using Minimal Data and Advanced Decoding Techniques." arXiv preprint arXiv:2312.02125 (2023).

[22] Abaskohi, Amirhossein, et al. "Benchmarking Large Language Models for Persian: A Preliminary Study Focusing on ChatGPT." arXiv preprint arXiv:2404.02403 (2024).

[23] Moradbeiki, Pardis, and Nasser Ghadiri. "PerkwE\_COQA: enhance Persian Conversational Question Answering by combining contextual keyword extraction with Large Language Models." arXiv preprint arXiv:2404.05406 (2024).

[24] Ghahroodi, Omid, et al. "Khayyam Challenge (PersianMMLU): Is Your LLM Truly Wise to The Persian Language?." arXiv preprint arXiv:2404.06644 (2024).

[25] Ayoubi, S., & Davoodeh, M. Y. (2021). PersianQA: A dataset for Persian Question Answering. GitHub. <https://github.com/SajjjadAyobi/PersianQA>

[26] Miladfa7. (n.d.). Persian Wikipedia Dataset. GitHub. Retrieved August 14, 2024, from <https://github.com/miladfa7/Persian-Wikipedia-Dataset>