# مجموعه‌داده‌ی چند سطحی فارسی برای بازیابی اطلاعات


**علی عابدزاده**
کارشناسی ارشد
دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران

**رضا رمضانی ***
دکتری تخصصی
استادیار، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران

**افسانه فاطمی**
دکتری تخصصی
دانشیار، دانشکده مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران



**دریافت:** ۱۴۰۱/۱۲/۲۷ | **پذیرش:** ۱۴۰۲/۰۸/۲۷ | مقاله برای اصلاح به مدت ۵ ماه و ۱۶ روز نزد پدیدآوران بوده است.



**چکیده:** یک سامانه‌ی بازیابی اطلاعات وظیفه دارد با دریافت یک پرسش یا پرسه[۱]، اسناد مرتبط با آن پرسه[۲] را بازیابی کند؛ که این بازیابی از میان مجموعه‌ای بزرگ از هزاران تا میلیون‌ها سند انجام می‌شود. در سال‌های اخیر، پژوهش‌های زیادی برای توسعه‌ی سامانه‌های بازیابی اطلاعات با استفاده از مدل‌های زبان[۳] انجام شده است؛ اما در این زمینه، پژوهشی برای زبان فارسی یافت نشد. یکی از علت‌های اصلی این امر، وجود نداشتن یک مجموعه‌داده‌ی فارسی مناسب برای آموزش مدل‌های زبان می‌باشد. در این پژوهش، ابتدا یک مجموعه‌داده‌ی بازیابی اطلاعات فارسی ارائه شده است. پس از آن، روش‌هایی برای غنی‌سازی این مجموعه‌ی داده مورد بحث قرار گرفته است. این غنی‌سازی با کمک چند سطحی کردن ارتباط میان پرسه و سند انجام می‌شود؛ به نحوی که مجموعه‌داده‌ی جدید می‌تواند رابطه بین پرسه و سند را بجای دو سطح (کاملا نامرتبط – کاملا مرتبط) در چهار سطح (نامرتبط – مرتبط – بسیار مرتبط – کاملا مرتبط) نشان دهد. مجموعه‌داده ایجاد شده PersianMLIR[۴] نام دارد[۵]. آزمایش‌ها بیانگر بهبود عملکرد سامانه هم برای زبان فارسی و هم برای زبان انگلیسی است که این میزان بهبود برای زبان فارسی ۱٫۸۷٪ می‌باشد.



نشریهٔ علمی (رتبه ین المللی)
پژوهشگاه علوم و فنّاوری اطلاعات ایران
شاپا(چاپی) ۸۲۲۳-۲۲۵۱
شاپا(الکترونیکی) ۸۲۳۱-۲۲۵۱
نمایه در LISTA، SCOPUS و ISC
http://jipm.irandoc.ac.ir
دوره xx | شمارهٔ x | ص ص xx-xx
۱۳xx x

**نوع مقاله: پژوهشی**

**به این مقاله به شکل زیر استناد کنید:**
**درون متن:**
(عابدزاده، رمضانی، فاطمی، زودآی‌ند)
**در فهرست منابع:**
عابدزاده، علی. رمضانی، رضا. فاطمی، افسانه. زودآی‌ند، مجموعه‌داده‌ی چند سطحی فارسی برای بازی‌ابی اطلاعات، *پژوهشنامه پردازش و مدیریت اطلاعات*)
(دسترسی در
http://jipm.irandoc.ac.ir


[۲] در این مقاله، به منظور یکنواختی، از کلمه پرسه بجای پرسش یا پرسه استفاده می‌شود

[۳] Language Models

[۴] Persian Multi-Level Information Retrieval (PersianMLIR)

[۵] لینک دانلود مجموعه‌داده: https://github.com/BigData-IsfahanUni/PersianMLIR

پژوهش‌نامهٔ پردازش و مدیریت اطلاعات

X xx ۱۳ | دورهٔ xx | شمارهٔ x

**کلیدواژه ها:** باز یابی اطلا عات، مدل های ز بان، مجمو عه‌داده باز یابی اطلا عات، مجموعه‌داده فارسی

* رضا رمضانی r.ramezani@eng.ui.ac.ir

## ۱. مقدمه

سامانه‌های بازیابی اطلاعات متنی کاربردهای بسیاری دارند. طی دهه‌ها پژوهش در این زمینه، محققان روش‌های متفاوتی برای ساخت سامانه‌های بازیابی اطلاعات ارائه کرده‌اند. از مشهورترین و پراستفاده‌ترین روش‌های بازیابی اطلاعات می‌توان به TF-IDF (Salton and Buckley 1988) و BM25 (Robertson and Zaragoza 2009) اشاره کرد. این دو روش، بر اساس مدل بسته کلمات و شباهت لغوی عمل می‌کنند. بر اساس معیارهای Trec-DL (Craswell et al. 2020; Craswell, Mitra, Yilmaz, and Campos 2021)، سامانه‌های بازیابی اطلاعات را بر اساس راه‌کار شباهت‌سنجی، می‌توان به سه دسته تقسیم کرد:

- سامانه‌های سنتی: این سامانه‌ها بر اساس روش‌های قدیمی مانند BM25 و TF-IDF (و به طور کلی، شباهت لغوی میان اسناد و پرسه) کار می‌کنند. در این سامانه‌ها استفاده از نمایش تُنُک اسناد بسیار رایج می‌باشد.
- سامانه‌های شبکه‌عصبی: در این سامانه‌ها، برای بازیابی از شبکه‌های عصبی کمک گرفته می‌شود. تمام سامانه‌هایی که بر اساس شبکه‌ی عصبی هستند، اما شبکه‌ی عصبی مورد استفاده، یک مدل زبان نمی‌باشد در این دسته قرار می‌گیرند. در این سامانه‌ها استفاده از نمایش متراکم اسناد بسیار رایج می‌باشد.
- سامانه‌های شبکه‌عصبی مدل زبان: اگر شبکه‌عصبی مورد استفاده در یک سامانه از نوع مدل زبان (مانند BERT (Devlin et al. 2019)) باشد، آن سامانه در این دسته قرار می‌گیرد. در این سامانه‌ها نیز، استفاده از نمایش متراکم اسناد بسیار رایج می‌باشد.

از اصلی‌ترین اشکالات روش‌های سنتی مبتنی بر نمایش تُنُک اسناد، بوجود آمدن اختلاف واژگانی میان سند و پرسه کاربر می‌باشد (برای نمونه «سرخ» و «قرمز») (Mitra and Craswell 2018; Qu et al. 2021). اشکال دیگر نمایش تُنُک اسناد زمانی پدیدار می‌شود که یک کلمه دو یا چند مفهوم متفاوت داشته باشد (مانند «شیر»). در نحوه‌ی نمایش متراکم اسناد که معمولا با کمک مدل‌های زبان و شبکه‌های عصبی تولید می‌شوند، معایب روش‌های قبل تا حد زیادی حل شده است. در این روش‌ها، به جای توجه به تشابه لغوی، تلاش می‌شود تا معنا و مفهوم پرسه و سند درک شود. در این حالت، نه تنها مشکل اختلاف واژگانی حل می‌شود، بلکه یک بازیابی معنایی وجود خواهد داشت (Z. Liu et al. 2021; Qu et al. 2021). اگرچه این روش‌ها دقت بالاتری دارند، اما در مواجهه با واژگان نادر و کمیاب، عملکرد خوبی ندارند و در چنین مواردی، نحوه‌ی نمایش تُنُک اسناد می‌تواند نتایج بهتری را حاصل کند (Mitra and Craswell 2018).

در سال‌های اخیر، پژوهش‌های بسیاری برای ساخت سامانه‌های بازیابی اطلاعات با کمک مدل‌های زبان انجام شده است. از آنجایی که مدل‌های زبان عمدتا بر اساس شبکه‌های عصبی عمیق هستند، وجود مجموعه‌داده‌های بزرگ‌مقیاس برای آموزش آن‌ها ضروری می‌باشد (Zhang, Yates, and Lin 2020). در طی سال‌های اخیر، مدل‌های زبان زیادی ساخته شده‌اند و تعداد قابل توجهی از این مدل‌ها قادر به درک و فهم زبان فارسی می‌باشند. با این حال، بدلیل عدم وجود یک مجموعه‌داده‌ی مناسب، پژوهشی روی بازیابی اطلاعات فارسی با کمک مدل‌های زبان انجام نشده است.

در این پژوهش، ابتدا روشی برای ساخت یک مجموعه‌داده‌ی بازیابی اطلاعات با کمک مجموعه‌داده‌های درک مطلب ماشینی[۶] فارسی موجود ارائه شده است[۷]. سپس روشی برای غنی‌سازی این مجموعه‌داده‌ی جدید ارائه گردیده است. در مجموعه‌داده تولید شده اولیه، همانند مجموعه‌داده‌های بازیابی اطلاعات مرسوم، به ازای هر پرسه و سند، تنها یکی از دو رابطه «کاملا نامرتبط» یا «کاملا مرتبط» وجود دارد. هرچند این مجموعه‌داده تولید شده، اولین مجموعه‌داده بازیابی اطلاعات برای زبان فارسی می‌باشد، اما در ادامه به منظور بهبود عملکرد آن، سطح روابط میان هر پرسه و سند را از دو سطح به چهار

[۶] Machine Reading Comprehension

[۷] این مجموعه‌داده درک زبان ماشینی فارسی در خوشه تحقیقاتی کلان داده دانشگاه اصفهان توسعه داده شده است.

سطح افزایش می‌دهیم به نحوی که ارتباط بین پرسه و سند می‌تواند یکی از چهار حالت «نامرتبط»، «مرتبط»، «بسیار مرتبط» و «کاملا مرتبط» باشد. سپس یک سامانه بازیابی اطلاعات مبتنی بر مدل زبان روی هر دو مجموعه‌داده آموزش داده شده است؛ که نتایج ارزیابی نشان می‌دهد مجموعه‌داده چهار سطحی می‌تواند بطور میانگین عملکرد مجموعه‌داده دو سطحی را ۱٫۸۷٪ بهبود دهد.

## ۲. پیشینه

از آنجایی که نوآوری اصلی این پژوهش ارائه یک مجموعه‌داده بازیابی اطلاعات فارسی چند سطحی است، در این بخش به بررسی مجموعه‌داده‌های بازیابی اطلاعات و درک مطلب ماشینی انگلیسی و فارسی می‌پردازیم. علت بررسی مجموعه‌داده‌های درک مطلب ماشینی این است که این مجموعه‌داده‌ها شباهت زیادی به مجموعه‌داده‌های بازیابی اطلاعات دارند و در مواردی می‌توان با اعمال تغییراتی در آن‌ها، یک مجموعه‌داده‌ی بازیابی اطلاعات ایجاد کرد. البته تبدیل کردن یک مجموعه‌داده‌ی درک مطلب ماشینی به یک مجموعه‌داده‌ی بازیابی اطلاعات می‌تواند مشکلاتی چون سوگیری داده‌های آموزشی، بی معنا بودن پرسه‌ها در صورت وجود نداشتن پاراگراف مربوطه، همپوشانی لغوی زیاد بین پرسه و سند مرتبط را بوجود آورد. با این وجود، تاکنون مجموعه‌داده‌های بازیابی اطلاعات مختلفی از روی این مجموعه‌داده‌های درک مطلب ماشینی ایجاد شده است.

مجموعه‌داده MS Marco (Bajaj et al. 2016; Craswell, Mitra, Yilmaz, Campos, et al. 2021) را می‌توان بزرگ‌ترین مجموعه‌داده بازیابی اطلاعات انگلیسی دانست که حاوی بیش از ۵۰۰ هزار پرسه و حدود ۸٫۸ میلیون سند (پاراگراف) است. این مجموعه‌داده، یک مجموعه‌داده دوسطحی محسوب می‌شود و مدل‌های بازیابی اطلاعات بسیاری در مرحله‌ی آموزش از آن استفاده می‌کنند.

Trec-DL یکی دیگر از مجموعه‌داده‌های معروف بازیابی اطلاعات است که در آن برای هر پرسه، تعداد زیادی سند نشانه‌گذاری شده وجود دارد. بر خلاف مجموعه‌داده‌ی MS Marco که یک مجموعه‌ی دوسطحی می‌باشد، مجموعه‌داده Trec-DL چندسطحی می‌باشد که سطوح روابط بین پرسه‌ها و اسناد به صورت زیر می‌باشد:

- کاملا مرتبط (سطح ۳): سند متعلق به پرسه می‌باشد و پاسخ دقیق پرسه در آن قرار دارد.
- بسیار مرتبط (سطح ۲): سند تا حدودی پاسخ پرسه را در خود دارد؛ اما پاسخ دقیق واضح نیست و یا در میان اطلاعات اضافه، پنهان شده است.
- مرتبط (سطح ۱): سند به نظر در ارتباط با پرسه می‌رسد، اما پاسخ در آن وجود ندارد.

- نامرتبط (سطح ۰): سند هیچ ارتباطی با پرسه ندارد.

مجموعه‌داده SQuAD (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018) یک مجموعه‌داده درک مطلب ماشینی انگلیسی است که شامل بیش از ۱۰۰ هزار پرسه و پاراگراف می‌باشد. مجموعه‌داده‌های درک مطلب ماشینی معمولا یک مجموعه از سه‌تایی‌ها به فرم (پاراگراف، پرسه، پاسخ پرسه) می‌باشند. با استفاده از این سه‌تایی‌ها، می‌توان درک مطلب ماشینی را آموزش داد که با دریافت پرسه و پاراگراف، سعی می‌کند تا پاسخ پرسه را از داخل پاراگراف پیدا کند. بطور مشابه، مجموعه‌داده‌ی بازیابی اطلاعات ANTIQUE (Hashemi et al. 2020) شامل بیش از ۲۵۰۰ پرسه پیچیده و دشوار است که از تالارهای گفت و گوی آنلاین جمع‌آوری شده‌اند. در این مجموعه‌داده، به ازای هر پرسه، حدود ۱۳ پاراگراف با سطوح ارتباطی مختلف توسط انسان نشانه‌گذاری شده است. مجموعه‌داده‌ی NaturalQuestions (Kwiatkowski et al. 2019) نیز یک مجموعه‌داده‌ی نسبتا بزرگ مقیاس با بیش از ۳۰۰ هزار داده‌ی آموزشی می‌باشد. در این مجموعه‌داده که برای آموزش و ارزیابی درک مطلب ماشینی طراحی شده است، هر پرسه معمولا با یک پاسخ کوتاه و یک پاسخ بلند و نیز پاراگراف مربوطه همراه شده است.

مجموعه‌داده‌هایی که تا کنون مورد بحث قرار گرفتند، مجموعه‌داده‌های انگلیسی بودند.با بررسی‌های صورت گرفته، مجموعه‌داده بازیابی اطلاعات فارسی که برای آموزش یک مدل زبان مناسب باشد، شناسایی نشد. مجموعه‌داده مورد نظر این پژوهش، باید ساختاری شبیه به مجموعه‌داده‌های انگلیسی بازیابی اطلاعات (که در بالا بررسی شدند) داشته باشد. خوشبختانه در سال‌های اخیر، فعالیت‌های زیادی در زمینه‌ی پرسش و پاسخ فارسی و درک مطلب ماشینی فارسی صورت گرفته و مجموعه‌داده‌هایی برای این زمینه‌های پژوهشی فراهم شده که می‌توان از آن‌ها به منظور ایجاد مجموعه‌داده بازیابی اطلاعات فارسی استفاده نمود. به عنوان مثال مجموعه ParsiNLU (Khashabi et al. 2021) متشکل از تعدادی مجموعه‌داده است که می‌توان از آن برای وظایف مختلف پردازش زبان طبیعی مانند استنتاج متنی، بازنویسی پرسه، پرسش پاسخ چند جوابی، ترجمه ماشینی، تحلیل احساسات و درک مطلب ماشینی استفاده کرد. بخش درک مطلب ماشینی این مجموعه شامل ۱۳۰۰ پرسه می‌باشد که تمام این پرسه‌ها با کمک موتور جست‌وجوی گوگل جمع‌آوری شده است.

مجموعه‌داده PersianQA (Ayoubi Sajjad & Davoodeh 2021) یک مجموعه‌داده‌ی فارسی برای درک مطلب ماشینی است که نحوه‌ی ایجاد و ساختار کلی آن مشابه SQuAD می‌باشد. این مجموعه‌داده شامل حدود ۱۰ هزار داده‌ی آموزشی می‌باشد و اسناد این مجموعه از ویکی‌پدیای فارسی استخراج شده است. بطور مشابه، مجموعه‌داده ParSQuAD (Abadani et al. 2021) که یک مجموعه‌ی فارسی برای درک مطلب ماشینی است، نسخه‌ی ترجمه شده‌ی SQuAD می‌باشد. مجموعه‌داده PersianQuAD (Kazemi, Mozafari, and Nematbakhsh 2022) نیز مانند سایر مجموعه‌های فارسی، یک مجموعه‌داده‌ی درک مطلب ماشینی می‌باشد که فرآیند ساخت این مجموعه‌داده کاملا مشابه مجموعه‌داده‌یSQuAD بوده و شامل حدود ۲۰ هزار داده‌ی آموزشی شامل سه‌تایی‌های (پرسه، پاسخ، سند مرتبط) می‌باشد. در این مجموعه نیز از اسناد ویکی‌پدیای فارسی برای طراحی مجموعه استفاده شده و برای طراحی پرسه‌ها و نشانه‌گذاری آن‌ها از افراد فارسی زبان کمک گرفته شده است.

از میان مجموعه‌داده‌های فارسی نام برده شده، مجموعه‌ی PersianQuAD مناسب‌ترین انتخاب برای ساخت یک مجموعه‌داده‌ی جدید بازیابی اطلاعات فارسی می‌باشد. علت این امر آن است که بر خلاف ParsQuAD (که یک نسخه‌ی ترجمه شده می‌باشد)، اسناد مورد استفاده برای ساخت این مجموعه در دسترس می‌باشند (ویکی‌پدیای فارسی) و این امر برای ساخت یک مجموعه‌داده‌ی بازیابی اطلاعات ضروری می‌باشد. همچنین این مجموعه‌داده، تعداد داده‌ی آموزشی بیشتری نسبت به PersianQA و ParsiNLU دارد و تعداد نمونه‌های آموزشی از اهمیت زیادی برخوردار می‌باشد (Zhang et al. 2020). در مورد ساخت مجموعه‌داده بازیابی اطلاعات با استفاده از این مجموعه‌داده، در بخش بعد صحبت شده است.

### ۳. ساخت مجموعه‌داده بازیابی اطلاعات

همانطور که پیش‌تر بیان شد، یکی از نوآوری‌های این پژوهش ارائه یک مجموعه‌داده بازیابی اطلاعات فارسی می‌باشد. در این بخش ابتدا نحوه‌ی تبدیل مجموعه‌داده PersianQuAD (که یک مجموعه‌داده‌ی درک مطلب ماشینی است) به یک مجموعه‌داده‌ی بازیابی اطلاعات بیان شده است. مجموعه‌داده‌ی ساخته شده اولیه در این بخش یک مجموعه‌ی دوسطحی است. سپس نحوه غنی‌سازی این مجموعه‌داده به کمک افزایش سطح روابط از دو سطح به چهار سطح مورد بحث قرار گرفته است.

### ۳-۱ ساخت مجموعه‌داده‌ی بازیابی اطلاعات فارسی دوسطحی

همان‌طور که اشاره شد، برای زبان فارسی، مجموعه‌داده‌ی بازیابی اطلاعاتی که برای آموزش و تنظیم دقیق مدل‌های زبان مناسب باشد وجود ندارد، اما با کمک مجموعه‌داده‌های فارسی درک مطلب ماشینی، می‌توان یک مجموعه‌داده‌ی جدید تولید کرد. برای این منظور در این پژوهش از مجموعه‌داده‌ی PersianQuAD (Kazemi et al. 2022) استفاده شده است. در این مجموعه داده، به ازای هر پرسه، چندین سند مرتبط وجود دارد. در این نوشتار، منظور از «سند» یک سند کامل، یک پاراگراف یا حتی یک جمله است.

در این پژوهش از DPR به عنوان مدل بازیابی اطلاعات استفاده خواهد شد (Karpukhin et al. 2020). لذا ساختار مجموعه‌داده بازیابی اطلاعات فارسی بایستی به نحوی شود که قابل استفاده توسط DPR باشد. DPR یک مدل بازیابی اطلاعات می‌باشد که با کمک آن می‌توان اسناد مرتبط با یک پرسه را بازیابی کرد. در این مدل از دو شبکه‌ی عصبی مجزا برای تعبیه کردن پرسه‌ها و اسناد استفاده می‌شود و شباهت میان خروجی‌های این دو شبکه نمایانگر میزان ارتباط میان پرسه و سند خواهد بود. مجموعه‌داده‌ی مورد استفاده در DPR شامل سه فایل می‌باشد:

- **فایل مجموعه‌ی اسناد:** یک پیکره متنی بزرگ که بازیابی بر روی آن انجام می‌شود. در این پیکره، هر سند می‌تواند یک سند کامل، یک پاراگراف یا یک جمله باشد.
- **فایل مجموعه‌ی آموزشی:** داده‌های آموزشی مدل که هر نمونه شامل یک پرسه، اسناد مرتبط و همینطور اسناد نامرتبط با آن پرسه می‌باشد.
- **فایل مجموعه‌ی ارزیابی:** مانند مجموعه‌ی آموزشی است اما با هدف ارزیابی سامانه استفاده می‌شود.

نحوه‌ی ساخت این سه فایل در ادامه بیان شده است.

#### *۳-۱-۱ ساخت مجموعه‌ی اسناد*

در ساخت PersianQuAD ابتدا با کمک الگوریتم PageRank حدود ۴۴۰۰ سند ویکی‌پدیا انتخاب شده است (Kazemi et al. 2022). سازندگان این مجموعه‌داده با استفاده از PageRank اطمینان حاصل کرده‌اند که صفحات مهم و پر ارجاع ویکی‌پدیا که شامل اطلاعات داغ و محبوبی هستند انتخاب شده‌اند. سپس با کمک ۱۹۰۵ سند (که می‌توان آن‌ها را به ۲۶ هزار پاراگراف تبدیل کرد)،

مجموعه‌ی داده‌ی درک مطلب ماشینی تولید شده است[۸]. در این پژوهش برای ساختن مجموعه‌ی اسناد، از این ۲۶ هزار پاراگراف استفاده شده است. با این وجود، از آنجایی که قصد داریم مجموعه‌ی اسناد مورد استفاده در این پژوهش، کمی بزرگ مقیاس‌تر از این تعداد سند باشد، لازم است تا اسناد دیگری نیز به این مجموعه اضافه شوند. برای این منظور، ابتدا یک نسخه‌ی متنی کامل از ویکی‌پدیای فارسی بارگیری شده است. پس از آن، متون داخل این فایل استخراج شده و به پاراگراف‌هایی با اندازه‌ی حداکثر ۵۰۰ واژه تبدیل شده‌اند. در نهایت، با انتخاب پاراگراف‌ها بصورت تصادفی، اندازه‌ی مجموعه‌ی اسناد به ۲۰۰ هزار پاراگراف رسیده است. لازم به ذکر است که برای جلوگیری از وارد شدن پاراگراف‌های تکراری در مجموعه‌ی اسناد، از وارد شدن پاراگراف‌هایی که عنوان سند آن‌ها، در لیست اسناد مورد استفاده‌ی PersianQuAD وجود داشته است خودداری شده است.

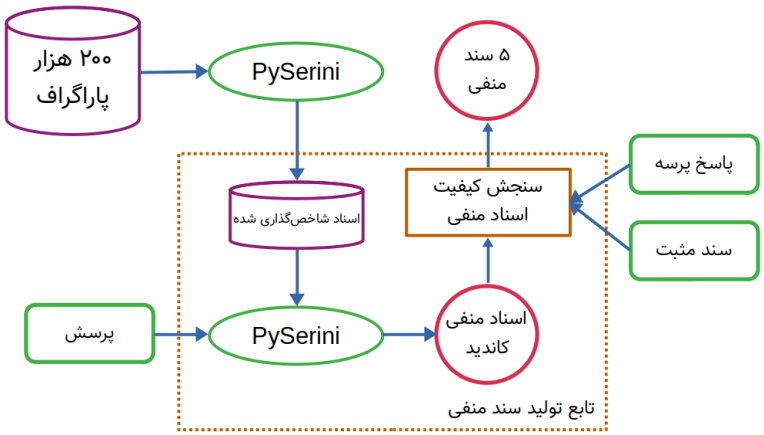
### ۳-۱-۲ ساخت مجموعه‌ی آموزشی

در مجموعه‌داده‌ی آموزشی بازیابی اطلاعات، هر نمونه یک سه‌تایی به فرم (پرسه، مجموعه اسناد مرتبط، مجموعه اسناد نامرتبط) می‌باشد. با کمک PersianQuAD می‌توان دو عنصر اول این سه‌تایی را تولید کرد، اما برای تولید مجموعه اسناد نامرتبط، باید از مجموعه‌ی کل اسناد (که در مرحله‌ی قبل ساخته شده) کمک گرفته شود. اسناد مرتبط معمولا نشانه‌گذاری شده‌اند؛ اما برای انتخاب اسناد نامرتبط، نیاز به یک مجموعه‌ی اسناد بزرگ است و لذا فرض می‌شود که هر سندی در این مجموعه، به غیر از اسناد مرتبط که نشانه‌گذاری شده‌اند، اسناد نامرتبط و منفی هستند (Karpukhin et al. 2020). روش‌های زیادی برای انتخاب سند منفی وجود دارد (Karpukhin et al. 2020; Y. Liu et al. 2021; Qu et al. 2021) اما یکی از رایج‌ترین روش‌ها، استفاده از اسناد با رتبه‌ی BM25 بالا می‌باشد که حاوی پاسخ نمی‌باشند. در این پژوهش برای انتخاب اسناد منفی، ابتدا اسناد با کمک ابزار PySerini (Lin et al. 2021; Yang, Fang, and Lin 2017) شاخص‌گذاری شده‌اند. PySerini ابزاری برای شاخص‌گذاری و بازیابی اطلاعات بر روی اسناد می‌باشد که می‌توان از آن برای بازیابی اسناد با کمک BM25 استفاده کرد. پس از شاخص‌گذاری اسناد، به ازای هر پرسه، پنج سند برتر BM25 که شامل پاسخ نمی‌باشند انتخاب می‌شوند. علاوه بر شرایط ذکر شده، اطمینان حاصل می‌شود که اسناد منفی انتخاب شده، عنوان سند متفاوتی از عنوان سند مرتبط داشته باشند. به این ترتیب، می‌توان اطمینان بیشتری حاصل نمود که اسناد منفی انتخاب شده، ارتباطی با پرسه مد نظر ندارند. در شکل ۱ کلیات فرآیند انتخاب

[۸] این ۱۹۰۵ سند از طریق سازندگان PersianQuAD در اختیار ما قرار گرفته است.

پنج سند منفی به ازای هر پرسه نشان داده شده است. همانطور که در شکل دیده می‌شود، پس از شاخص‌گذاری اسناد با کمک PySerini، به ازای هر پرسه‌ی ورودی، تعدادی سند بازیابی می‌شود که به عنوان اسناد منفی کاندید در نظر گرفته می‌شوند. پس از آن با کمک محدودیت‌هایی که به آن‌ها پرداخته شد (عدم وجود پاسخ در سند کاندید و تفاوت عنوان سند کاندید با سند مثبت)، این اسناد منفی فیلتر می‌شوند تا اسناد منفی باکیفیت‌تری تولید شود. اعمال کردن این محدودیت‌ها به عهده‌ی بخش «سنجش کیفیت اسناد منفی» است. از میان اسناد منفی فیلتر شده، پنج سند با بالاترین امتیاز BM25 به عنوان اسناد منفی انتخاب می‌شوند.

با کمک روش بیان شده، می‌توان مجموعه‌داده‌ی آموزشی مورد نیاز برای آموزش DPR را ایجاد نمود. چگونگی عملکرد و نحوه‌ی آموزش مدل DPR پایه در بخش (۴-۱) توضیح داده شده است.




**شکل ۱: فرآیند انتخاب اسناد منفی برای هر پرسه**

### ۳-۱-۳ ساخت مجموعه‌ی ارزیابی

روند ساخت مجموعه‌ی ارزیابی بازیابی اطلاعات، بسیار مشابه روند ساخت مجموعه‌ی آموزشی می‌باشد؛ با این تفاوت که در این مجموعه‌ی آموزشی، نیازی به وجود اسناد نامرتبط و منفی نمی‌باشد؛ اما در عوض، نیاز به دانستن پاسخ مربوط به پرسه می‌باشد. در این حالت سه‌تایی‌های (پرسه، پاسخ پرسه، سند مرتبط) تشکیل می‌شوند. استفاده از پاسخ پرسه در سامانه‌های بازیابی اطلاعات امر رایجی نمی‌باشد، اما DPR از این پاسخ‌ها برای محاسبه‌ی سنجه‌ی Exact Match و سنجش عملکرد مدل تحت آموزش

استفاده می‌کند. در بخش (۵-۱) به شرح سنجه‌های مورد استفاده در DPR و این پژوهش پرداخته شده است.

**۳-۲ ساخت مجموعه‌داده‌ی بازیابی اطلاعات فارسی چندسطحی**

مجموعه‌داده‌ی ارائه شده در بخش (۳-۱) یک مجموعه‌داده‌ی دوسطحی است؛ چرا که برای هر پرسه، تنها اسناد «مرتبط» و «نامرتبط» تعیین شده است. یک مجموعه‌داده‌ی چندسطحی می‌تواند رابطه‌ی یک پرسه و سند را در چند سطح نمایش دهند. در این پژوهش از ساختار مجموعه‌داده Trec-DL که یک مجموعه‌داده‌ی بازیابی اطلاعات چندسطحی است (Craswell et al. 2020; Craswell, Mitra, Yilmaz, and Campos 2021) استفاده شده است.

مجموعه‌داده Trec-DL روابط را در چهار سطح نشان می‌دهد: کاملا مرتبط (سطح ۳)، بسیار مرتبط (سطح ۲)، مرتبط (سطح ۱)، نامرتبط (سطح ۰). برای نمونه، اگر پرسه‌ی مورد بررسی «مصدق در چه سالی متولد شده؟» باشد، پاراگراف «مصدق در سال ۱۲۶۱ چشم به جهان گشود و ...» *کاملا مرتبط*، پاراگراف «در سال ۱۲۸۰ خورشیدی، مصدق که در آن زمان نوزده سال داشت ...» *بسیار مرتبط*، پاراگراف «... ازدواج این دو ۶۴ سال تا پایان زندگانی‌شان ادامه یافت» *مرتبط* و پاراگراف «پس از مرگ مظفرالدین شاه پسرش محمد علی شاه در تاریخ ۲۹ دی ۱۲۸۵ تاج‌گذاری کرد.» *نامرتبط* می‌باشد. در واقع می‌توان گفت که سطح ۰ همان سند منفی است و سطح ۳ نیز همان سند مثبت است. در این حالت، اطلاعات اضافه‌ای میان این دو سطح اضافه شده است (سطح ۱ و ۲) که مدل می‌تواند با کمک آن‌ها، بازیابی دقیق‌تر و بهتری انجام دهد.

در این پژوهش، برای تبدیل کردن مجموعه‌داده‌ی دوسطحی ساخته شده در بخش (۳-۱) به یک مجموعه‌ی چندسطحی، از دو روش استفاده شده است:

- روش مبتنی بر قوانین دست‌ساز
- روش مبتنی بر طبقه‌بندی و ویژگی‌های لغوی

### ۳-۲-۱ روش مبتنی بر قوانین دست‌ساز

برای ساخت یک مجموعه‌داده‌ی چند سطحی با کمک قوانین دست‌ساز، یکصد سند برتر مرتبط با هر پرسه با کمک PySerini استخراج شده‌اند تا بتوان از آن‌ها به عنوان اسناد کاندید[۹] استفاده کرد. پس

[۹] سند کاندید سندی است که از میان کل اسناد انتخاب شده است اما میزان ارتباط آن با پرسه‌ی مورد بررسی مشخص نمی‌باشد و می‌تواند به صورت بالقوه در یکی از چهار سطح ارتباطی قرار بگیرد.

از آن، با کمک قوانینی که به صورت دستی تعریف شده‌اند، هر سند کاندید به یکی از سطوح ۰ الی ۲ منتصب شده است (نامرتبط، مرتبط، بسیار مرتبط). سطح ۳ (کاملا مرتبط) تنها شامل یک سند می‌باشد و آن سند همان سند مثبت می‌باشد و قوانین مورد بحث نقشی در انتخاب آن ندارند. برای تعیین سطح یک سند کاندید، علاوه بر خود سند کاندید و عنوان آن، سند مثبت و عنوان آن و همینطور پاسخ کوتاه و ریشه‌های آن مورد نیاز هستند. برای استخراج ریشه های پاسخ از ابزار Hazm استفاده شده است[۱۰].

قوانین تعریف شده به ترتیب زیر می‌باشند:

- قانون ۱: اگر عنوان سند کاندید و عنوان سند مثبت یکی باشد و همچنین پاراگراف کاندید شامل بخشی از ریشه لغت‌های پاسخ باشد، سند کاندید یک سند سطح ۲ خواهد بود.
- قانون ۲: اگر پاسخ عینا در سند کاندید وجود داشته باشد، سند کاندید یک سند سطح ۲ خواهد بود.
- قانون ۳: اگر عنوان سند کاندید و عنوان سند مثبت یکی باشد یا پاراگراف کاندید شامل بخشی از ریشه لغت‌های پاسخ باشد، سند کاندید یک سند سطح ۱ خواهد بود.
- قانون ۴: اگر عنوان سند کاندید و عنوان سند مثبت متفاوت باشد و همینطور سند کاندید شامل پاسخ دقیق نباشد، سند کاندید یک سند سطح ۰ خواهد بود.

از میان قوانین تعریف شده، اولین قانونی که برقرار شود، تعیین‌گر سطح سند می‌باشد و قوانین بعد از آن اثری نخواهند داشت.

### ۳-۲-۲ روش مبتنی بر طبقه‌بندی و ویژگی‌های لغوی

ایده‌ی اصلی این بخش این است که با کمک یک مجموعه‌داده‌ی چندسطحی انگلیسی موجود، یک مدل آموزش داده شود که بتواند با کمک این مجموعه‌ی داده، میزان ارتباط اسناد را یاد بگیرد. در واقع هدف آن است تا دانش موجود در یک مجموعه‌داده‌ی چندسطحی به یک مدل منتقل شود و با کمک آن مدل، یک مجموعه‌داده‌ی جدید ایجاد شود. برای این منظور در این پژوهش، از مجموعه‌داده‌ی Trec-DL (که یک مجموعه‌داده‌ی چندسطحی انگلیسی است) برای آموزش مدل استفاده گردیده و از

[۱۰] ابزاری است مانند NLTK پایتون برای تمیز کردن متن، تجزیه نحوی، ریشه‌یابی و ...: https://www.roshan-ai.ir/hazm

مدل ایجاد شده، برای ساخت یک مجموعه‌داده‌ی فارسی استفاده شده است. برای ان منظور، ابتدا نیاز به یک تابع برای استخراج ویژگی می‌باشد، که این ویژگی‌ها عبارتند از:

- آیا سند کاندید شامل پاسخ کامل می‌باشد؟ به عبارتی آیا پاسخ ارائه شده برای پرسه، عینا در سند کاندید وجود دارد.
- آیا سند کاندید و سند مثبت، عنوان مشترک دارند؟
- لیستی از شباهت‌های لغوی میان سند کاندید و سند مثبت. در این لیست، با کمک سه تابع شباهت کسینوسی، جاکارد و اقلیدوسی، شش مقدار مشابهت میان سند کاندید و سند مثبت محاسبه می‌شوند (دو مقدار به ازای هر تابع شباهت)[11].
- لیستی از شباهت معنایی میان سند کاندید و سند مثبت. در این لیست، از مدل زبان‌های تولید شده توسط DPR دوسطحی استفاده می‌شود. برای این منظور، DPR به تعداد یک، سه و شش دور تنظیم دقیق می‌شود و از مدل‌های هر اجرا، برای تعبیه کردن و به دست آوردن شباهت میان بردار سند مثبت و بردار سند کاندید استفاده می‌شود. به طور مشابه، میزان شباهت پرسه و سند کاندید نیز با کمک این مدل‌ها محاسبه می‌شود. علاوه بر سه مدل تنظیم دقیق شده‌ی بالا، یک مدل BERT نیز (بدون تنظیم دقیق شدن) برای تعبیه کردن اسناد و پرسه‌ها استفاده می‌شود. برای به دست آوردن شباهت، از توابع ضرب داخلی و همینطور شباهت کسینوسی استفاده شده است. حاصل تمام این محاسبات، ۱۶ مقدار مشابهت (دو تابع شباهت‌سنجی اعمال شده بر روی هشت مدل زبان) می‌شود که در یک لیست قرار می‌گیرند.

با در دست داشتن این تابع، می‌توان با کمک PySerini یکصد سند کاندید را استخراج و ویژگی‌های هر یک از آنان را استخراج کرد. سپس با وارد کردن این ویژگی‌ها به یک طبقه‌بند می‌توان سطح این اسناد کاندید را تعیین کرد. مانند قبل، تعیین سطح سند تنها برای سطوح ۰ الی ۲ می‌باشد و مجموعه اسناد سطح ۳ مرتبط با هر پرسه تنها شامل یک عضو است که همان سند مثبت می‌باشد و طبقه‌بند نقشی در تعیین آن ندارد.

---

[11] برای محاسبه‌ی این شباهت‌ها، ابتدا یک بردار از سند کاندید و یک بردار از سند مثبت ساخته می‌شود. هر عنصر در این دو بردار نشان‌دهنده‌ی یک واژه می‌باشد. مقداردهی این عناصر می‌تواند به صورت دودویی (۱ برای وقوع واژه در سند و ۰ برای عدم وقوع آن) و غیر دودویی (تعداد وقوع واژه) باشد. در نتیجه به ازای هر سند یک بردار دودویی و یک بردار غیر دودویی وجود خواهد داشت و با کمک سه تابع شباهت یاد شده، می‌توان شش مقدار شباهت تولید کرد.

مدل‌هایی که در این بخش برای طبقه‌بندی اسناد مورد استفاده قرار گرفته‌اند، عبارتند از: درخت تصمیم، جنگل تصادفی، ماشین بردار پشتیبان، شبکه‌ی عصبی، نزدیک‌ترین همسایه، AdaBoost[۱۲]، SGD[۱۳]

برای آموزش این مدل‌ها، از ویژگی‌های استخراج شده در مراحل قبل استفاده شده است و برای سطح ارتباط که بایستی پیش‌بینی شود نیز از سطوح تعین شده در Trec-DL (که توسط سازندگان مجموعه‌ی داده برای هر پاراگراف مشخص شده است) استفاده گردیده است. در میان این ویژگی‌ها، مواردی که از پاسخ استفاده می‌کنند وجود ندارند؛ چراکه مجموعه‌داده‌ی Trec-DL چنین اطلاعاتی را در خود ندارد. از آنجایی که در میان این ویژگی‌ها، ویژگی‌های مرتبط با پاسخ وجود ندارد، سایر ویژگی‌های باقی‌مانده مستقل از زبان می‌باشند و می‌توان توابع مورد استفاده برای مجموعه‌ی انگلیسی را برای استخراج ویژگی از مجموعه‌ی فارسی استفاده کرد. پس از آموزش و ارزیابی مدل، می‌توان مانند بخش قبل، اسناد کاندید فارسی را با کمک PySerini و BM25 تولید نموده، ویژگی‌های آن‌ها را استخراج کرد و سپس آن‌ها را طبقه‌بندی نمود. در نهایت با کمک طبقه‌بندی‌های انجام شده، می‌توان یک مجموعه‌داده‌ی چندسطحی فارسی ساخت.

## ٤. آموزش مدل بازیابی اطلاعات

### ٤-١ مدل DPR پایه

برای استفاده از مجموعه‌داده بازیابی اطلاعات دوسطحی ساخته شده، یک مدل بازیابی اطلاعات به عنوان مدل پایه مورد نیاز است. بدین منظور در این پژوهش از مدل DPR[۱۴] (Karpukhin et al. 2020) استفاده شده است. DPR یک مدل بازیابی اطلاعات می‌باشد که با کمک مدل‌های زبان قادر است تا اسناد مرتبط با یک پرسه را بازیابی کند. DPR از دو رمزگذار BERT مستقل استفاده می‌کند، اما نیازی به پیش‌آموزش آن‌ها از قبل ندارد. در عوض، DPR بر یادگیری یک مدل بازیابی اطلاعات قوی با استفاده از پرسش‌ها و پاسخ‌های دوتایی تمرکز می‌کند. در واقع DPR راه‌هایی را برای انتخاب و

[۱۲] Adaptive Boosting

[۱۳] Stochastic Gradient Descent

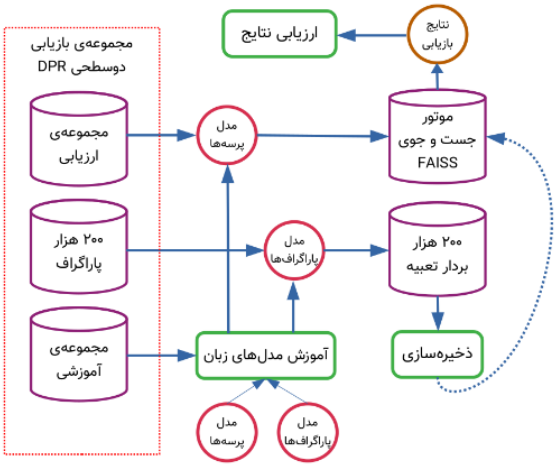
[۱۴] Dense Passage Retrieval

استفاده از نمونه‌های منفی برای یک سؤال درنظر گرفته است. نمونه‌های منفی می‌توانند هرگونه سند تصادفی از مجموعه‌داده یا اسناد برتر بازگردانده شده توسط BM25 که حاوی پاسخ صحیح نیستند، باشد. از آنجایی که مدل DPR پایه تنها دو نوع رابطه «مرتبط» و «نامرتبط» را می‌شناسد، در ابتدا ساختار و نحوه‌ی آموزش مدل DPR بر روی مجموعه‌داده ارائه شده دوسطحی توضیح داده شده است. سپس در بخش بعدی تغییرات مورد نیاز در DPR برای درک و فهم مجموعه‌داده ارائه شده چهارسطحی مورد بحث قرار گرفته است.

آموزش و ارزیابی مدل DPR پایه در سه مرحله انجام می‌شود:

- آموزش دادن رمزگذارهای دوگانه با کمک مجموعه‌داده آموزشی
- تعبیه کردن مجموعه‌ی اسناد با کمک رمزگذار مربوط به اسناد و ذخیره‌سازی مقادیر تعبیه شده با کمک ابزار FAISS (Johnson, Douze, and Jégou 2019)
- بازیابی اطلاعات برای پرسه‌های داخل مجموعه‌ی ارزیابی با کمک رمزگذار مربوط به پرسه‌ها و FAISS و سپس ارزیابی کردن کارایی سامانه بر اساس نتایج حاصل شده

در شکل ۲ می‌توان جایگاه این سه مرحله را مشاهده کرد. در این شکل می‌توان دید که علاوه بر مجموعه‌داده‌ی دوسطحی، دو نمونه از مدل زبان مورد نیاز می‌باشد تا بتوان آن‌ها را آموزش داد و سپس از آن‌ها برای تعبیه کردن اسناد و پرسه‌ها استفاده کرد.




**شکل ۲: مراحل آموزش، تعبیه و ارزیابی در DPR**

برای آموزش مدل بازیابی اطلاعات، یک مدل زبان فارسی و یا یک مدل زبان چندزبانه که از زبان فارسی نیز پشتیبانی کند مورد نیاز است. در این پژوهش، از مدل BERT چندزبانه[۱۵] استفاده شده است. همان‌طور که قبلا نیز اشاره شد، جهت آموزش مدل، به دو نمونه از این مدل زبان نیاز می‌باشد. یک نمونه برای تعبیه کردن اسناد (پاراگراف) که آن را با $E_P$ نشان می‌دهیم و نمونه‌ی دیگر برای تعبیه کردن پرسه‌ها که آن را با $E_Q$ نشان می‌دهیم. در زمان آموزش، یک دسته از داده‌های آموزشی انتخاب می‌شوند و همه به طور موازی پردازش می‌شوند. در هر دسته، هر پرسه شامل تعدادی سند مثبت و منفی می‌باشد که در مجموعه‌ی داده تعیین شده‌اند. علاوه بر اسناد منفی داخل مجموعه، اسناد مثبت پرسه‌های دیگر نیز می‌توانند به عنوان سند منفی پرسه جاری استفاده شوند (Karpukhin et al. 2020). پس از تعبیه کردن پرسه‌ها و اسناد، با استفاده از ضرب داخلی بردارها، شباهت میان هر پرسه و سند محاسبه می‌شود که آن را به صورت $Sim(q,p)$ نشان می‌دهیم که q یک پرسه و p یک پاراگراف است و برای محاسبه‌ی آن از رابطه‌ی ۱ استفاده می‌شوند.

$$sim(q,p) = E_Q(q)^T E_P(p) \qquad (1)$$

[۱۵] BERT-base-multilingual-uncased

هرچه مقدار ضرب داخلی یک سند و پرسه بیشتر باشد، ارتباط میان آن دو قوی‌تر می‌باشد. برای نمونه، اگر فرض شود که برای آموزش مدل از دسته‌های سه‌تایی استفاده می‌شود و هر پرسه به همراه یک سند مثبت و یک سند منفی باشد، با کمک رابطه‌ی ۱ می‌توان ماتریس رسم شده در جدول ۱ را بدست آورد. در این جدول، سند $d_1$ یک سند مثبت برای پرسه $q_1$ می‌باشد؛ اما همین سند برای پرسه‌های $q_2$ و $q_3$ یک سند منفی محسوب می‌شود.

**جدول ۱: ماتریس فرضی برای محاسبه‌ی شباهت سه پرسه**

| | سند مثبت برای پرسه ۱ | سند منفی برای پرسه ۱ | سند مثبت برای پرسه ۲ | سند منفی برای پرسه ۲ | سند مثبت برای پرسه ۳ | سند منفی برای پرسه ۳ |
|---|---|---|---|---|---|---|
| | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $d_5$ | $d_6$ |
| $q_1$ | $sim(q_1, d_1)$ | $sim(q_1, d_2)$ | $sim(q_1, d_3)$ | $sim(q_1, d_4)$ | $sim(q_1, d_5)$ | $sim(q_1, d_6)$ |
| $q_2$ | $sim(q_2, d_1)$ | $sim(q_2, d_2)$ | $sim(q_2, d_3)$ | $sim(q_2, d_4)$ | $sim(q_2, d_5)$ | $sim(q_2, d_6)$ |
| $q_3$ | $sim(q_3, d_1)$ | $sim(q_3, d_2)$ | $sim(q_3, d_3)$ | $sim(q_3, d_4)$ | $sim(q_3, d_5)$ | $sim(q_3, d_6)$ |

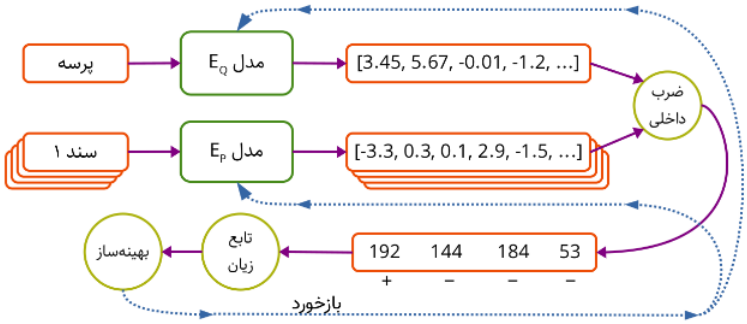
سپس با کمک ماتریس محاسبه شده، به ازای هر سطر از ماتریس، تابع زیان محاسبه می‌شوند تا مقادیر زیان را به دست آوریم. رابطه‌ی ۲ این تابع زیان را نشان می‌دهد که Negative Log Likelihood (و یا به اختصار NLL) نام دارد (Karpukhin et al. 2020). در این رابطه، $(q_i, p_i^+, p_i^-{}_1, \dots, p_i^-{}_n)$ نشان‌دهنده‌ی یک سطر از ماتریس با شاخص $i$ است که شامل یک پرسه $(q_i)$، یک سند مثبت $(p_i^+)$ و چند سند منفی $(p_i^-{}_1, \dots, p_i^-{}_n)$ می‌باشد. بخش کسری این رابطه همواره در بازه‌ی $[0, 1]$ است و هرچه مقدار این کسر به ۱ نزدیک‌تر باشد، به این معنا است که مدل پیش‌بینی دقیق‌تری انجام داده و شباهت پرسه و سند مثبت را بالاتر ارزیابی کرده است. به طور مشابه، اگر مدل ما شباهت میان یک یا چند سند منفی با پرسه را بالا محاسبه و ارزیابی کند، مقدار مخرج این کسر افزایش پیدا می‌کند و مقدار نهایی کسر به صفر نزدیک‌تر می‌شود. مقدار شباهت محاسبه شده توسط بخش کسری این رابطه، با کمک عملیات لگاریتمی و منفی‌سازی آن، به یک مقدار زیان تبدیل می‌شود. شایان ذکر است که خروجی این رابطه همواره در بازه‌ی $[0, \infty)$ می‌باشد.

$$L\left(q_i, p_i^+, p_i^-{}_1, \dots, p_i^-{}_n\right) = -Log \frac{e^{sim(q_i, p_i^+)}}{e^{sim(q_i, p_i^+)} + \sum_{i=1}^{n} e^{sim\left(q_i, p_{(i,j)}^-\right)}} \qquad (2)$$

با در دست داشتن مقادیر زیان، می‌توان به رمزگذار $E_P$ (مربوط به پاراگراف‌ها) و همینطور رمزگذار $E_Q$ (مربوط به پرسه‌ها) یک بازخورد از نحوه‌ی عملکرد آن‌ها داد. اگر مقدار تابع زیان نزدیک به صفر

باشد، پارامترهای مدل کمتر دستخوش تغییرات می‌شوند و هرچه مقدار تابع زیان بیشتر شود، تغییرات در پارامترهای مدل بیشتر خواهد شد. اعمال این تغییرات تا زمانی که مدل قادر به پیش‌بینی صحیح شباهت میان پرسه‌ها و اسناد مثبت شود ادامه می‌یابد. با کمک این تابع زیان می‌توان مدل را تنظیم دقیق کرد و در طی این مرحله، مدل یاد می‌گیرد تا برای اسناد و پرسه‌هایی که از منظر بازیابی اطلاعات مرتبط هستند، بردارهای مشابه تولید کند تا میزان شباهت بردارهای آن‌ها، بیشینه شود.

در شکل ۳ می‌توان فرآیند کلی آموزش (تنظیم دقیق) را مشاهده کرد. در این شکل (که در آن فرآیند آموزش ساده‌سازی شده است) می‌توان دید که یک پرسه، یک سند مثبت (سند ۱) و سه سند منفی (اسناد ۲ تا ۴) به رمزگذارهای $E_Q$ و $E_P$ داده شده‌اند و به چندین بردار عددی تبدیل شده‌اند. سپس با کمک تابع شباهت ضرب داخلی، شباهت هر یک از بردارهای اسناد با بردار پرسه محاسبه شده و یک لیست از این شباهت‌ها تولید می‌شود که عنصر اول آن مربوط به سند مثبت و بقیه‌ی عناصر مربوط به اسناد منفی است. تابع زیان با دانستن جایگاه سند مثبت می‌تواند عملکرد رمزگذارها را ارزیابی کند. پس از آن با کمک بهینه‌ساز بازخوردی برای تنظیم پارامترهای دو مدل تولید شده و بدین ترتیب، مدل تنظیم دقیق می‌شود.




**شکل ۳: فرآیند آموزش DPR**

پس از آن‌که دو مدل زبان مورد استفاده در مرحله‌ی قبل آموزش دیدند، تمامی اسناد، تعبیه و شاخص‌گذاری می‌شوند. برای تعبیه کردن، لازم است تا از مدل مربوط به اسناد $E_P$ استفاده کرده و تمامی اسناد را به بردارهای معادل آن نگاشت کنیم. پس از تعبیه شدن اسناد، با کمک ابزار FAISS باید تمام بردارها ذخیره شوند تا بتوان بر روی آن‌ها جست‌وجو انجام داد. این ابزار قادر است تا پس از

ذخیره‌سازی اسناد، یک بردار دریافت کند و بردارهای شبیه به آن بردار را با سرعت بسیار بالا بازیابی کند. در این مرحله می‌توان با کمک FAISS و مدل زبان $E_Q$، بازیابی اطلاعات را انجام داد.

برای ارزیابی سامانه، از بخش ارزیابی مجموعه‌داده که شامل سه‌تایی‌های (پرسه، پاسخ پرسه، سند مرتبط) می‌باشد استفاده می‌شود. با تعبیه کردن هر پرسه با کمک مدل $E_Q$ یک بردار حاصل می‌شود و با وارد کردن این بردار به FAISS تعداد مشخصی سند که بردارهای مشابه دارند حاصل می‌شود. امتیاز نهایی این سندها همان ضرب داخلی دو بردار است و اسناد بر اساس این مقدار مرتب می‌شوند. حال می‌توان با کمک سنجه‌های ارزیابی، کیفیت بازیابی اطلاعات این سامانه را سنجید. سنجه‌های ارزیابی در بخش (۵-۱) معرفی شده‌اند.

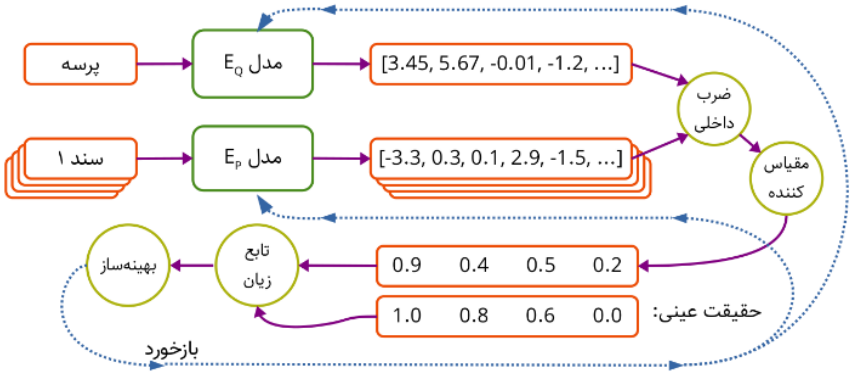
### ٤-۲ مدل DPR چندسطحی

به‌منظور استفاده از مجموعه‌داده‌ی بازیابی اطلاعات چندسطحی، نیاز به تغییر دادن مدل DPR می‌باشد تا بتواند از سطوح تعریف شده در این مجموعه‌داده‌ی جدید استفاده کند. مهم‌ترین تغییری که بایستی در مدل DPR اعمال شود، بخش مربوط به تابع زیان است. این بخش باید قادر باشد تا به رتبه‌بندی مدل از اسناد چندسطحی، امتیاز دهد. در بخش (۴-۱) دیده شد که شباهت میان بردار اسناد و پرسه با کمک ضرب داخلی انجام می‌شود و تابع زیان مدل DPR پایه نیز، تابع NLL بود. در این بخش نیز از ضرب داخلی برای محاسبه‌ی شباهت استفاده شده است؛ اما تابع زیانی که در این بخش استفاده شده، باید قادر باشد تا یک بردار از شباهت‌ها (به صورت نرمال شده) دریافت کند و به این بردار امتیاز دهد. لذا برای مدل DPR چندسطحی، از تابع RankCosine (Xia et al. 2008) به عنوان تابع زیان استفاده شده است.

برای محاسبه‌ی تابع فوق، لازم است تا در ابتدا، شباهت‌های تولید شده توسط ضرب داخلی به بازه‌ی ۰ تا ۱ نگاشت شود. برای این منظور از یک مقیاس‌کننده‌ی min-max استفاده شده است. پس از آن با کمک تابع زیان، عملکرد سامانه سنجیده می‌شوند. در این بخش، تابع زیان علاوه بر دریافت بردار شباهت که حاصل محاسبات دو مدل زبان و ضرب داخلی بردارها می‌باشد، یک بردار حقیقت عینی[۱۶] نیز دریافت می‌کند که مقادیر آن به طور ضمنی توسط مجموعه‌داده تعیین می‌شوند و در واقع فاصله‌ی این دو بردار ورودی محاسبه می‌شود. بردار حقیقت با کمک میزان ارتباط اسناد با پرسه حاصل می‌شود. برای نمونه، اگر سند اول در بردار مشابهت، یک سند *کاملا مرتبط* (سطح ۳) باشد، عنصر نظیر آن در بردار حقیقت

[۱۶] Ground Truth

۱ خواهد بود که معادل بالاترین مقدار ممکن برای امتیاز یک سند می‌باشد. به طور مشابه، عنصر نظیر یک سند *نامرتبط* برابر ۰ خواهد بود که معادل پایین‌ترین مقدار ممکن برای امتیاز یک سند می‌باشد. در شکل ۴ می‌توان جایگاه بردار حقیقت در مدل بازیابی اطلاعات چندسطحی را مشاهده کرد. مقادیری که به ازای هر سطح در نظر گرفته می‌شوند، یک پارامتر برای مدل محسوب می‌شود و از همین جهت، برای یافتن مقدار مناسب، نیاز به تنظیم پارامتر می‌باشد.




**شکل ٤: فرآیند آموزش DPR چندسطحی**

برای محاسبه‌ی RankCosine ابتدا باید شباهت کسینوسی دو بردار ورودی محاسبه شود که این امر با کمک رابطه‌ی ۳ امکان‌پذیر می‌باشد. در این رابطه، $V_{Dot}$ بردار شباهت تولید شده توسط ضرب داخلی است و هر عنصر از این بردار، شباهت میان پرسه و یک سند را نشان می‌دهد. همان‌طور که قبلا نیز اشاره شد، مقادیر این بردار به بازه‌ی ۰ الی ۱ مقیاس شده‌اند. بردار $V_{Ground}$ همان بردار حقیقت است که هر عنصر آن، نشانگر سطح ارتباط سند با پرسه می‌باشد. مقادیر این بردار نیز در بازه‌ی ۰ الی ۱ می‌باشند. مقدار محاسبه شده توسط رابطه‌ی ۳ همواره میان ۱- و ۱ است.

$$Cosin(V_{Dot}, F_{Ground}) = \frac{V_{Dot}{}^{T} V_{Ground}}{||V_{Dot}||\ ||V_{Ground}||} \quad (3)$$

برای محاسبه‌ی مقدار زیان از رابطه‌ی ۴ استفاده می‌شود (Xia et al. 2008). برای محاسبه‌ی این مقدار، ابتدا شباهت کسینوسی دو بردار محاسبه شده و سپس مقدار این شباهت به مقدار زیان تبدیل می‌شود.

$$L(V_{Dot}, V_{Ground}) = \frac{1}{2}\left[1 - Cosine(V_{Dot}, V_{Ground})\right] \quad (4)$$

با کمک این تابع زیان، می‌توان مدل را با کمک مجموعه‌داده‌ی چندسطحی آموزش داد. پس از فرآیند آموزش، نحوه‌ی استفاده از مدل‌ها دقیقا مانند یک مدل دوسطحی می‌باشد. با کمک مدل‌های آموزش دیده، می‌توان اسناد و پرسه‌ها را تعبیه کرد، ضرب داخلی بردارهای حاصل شده را محاسبه نمود و اسناد را بر اساس امتیاز ضرب داخلی، مرتب کرد. در واقع در مرحله‌ی استفاده، معنا و مفهوم دوسطحی و چندسطحی بودن مدل‌ها از میان می‌رود و تنها رتبه‌بندی اسناد اهمیت خواهد داشت. به عبارت دیگر، تفاوت مدل چندسطحی و دوسطحی، تنها در بخش آموزش و مجموعه‌داده‌ی آموزشی می‌باشد و در مراحل تعبیه‌کردن اسناد و ارزیابی، از عملیات و مجموعه‌های ارزیابی کاملا یکسانی استفاده می‌شود. علت این امر آن است که مدل‌ها (چه دوسطحی و چه چندسطحی) پس از مرحله‌ی آموزش، رفتار یکسانی از خود نشان می‌دهند و با دریافت یک پرسه و سند، یک عدد اعشاری به عنوان امتیاز تولید می‌کنند.

## ۵. ارزیابی

همانطور که پیشتر گفته شد، تفاوت مدل چندسطحی و دوسطحی، تنها در بخش آموزش و مجموعه‌داده‌ی آموزشی می‌باشد؛ و در مرحله‌ی ارزیابی، استفاده از مجموعه‌داده‌های دوسطحی و یا چندسطحی برای هر دو مدل امکان‌پذیر می‌باشد و این امر هیچ ارتباطی با نحوه‌ی آموزش دوسطحی یا چندسطحی مدل‌ها نخواهد داشت. برای نمونه، می‌توان یک مدل را به صورت دوسطحی آموزش داد و در ارزیابی آن، از یک مجموعه‌داده‌ی چندسطحی و یا دوسطحی استفاده کرد.. در این پژوهش، برای داشتن یک ارزیابی صحیح و عادلانه، از یک مجموعه ارزیابی دوسطحی برای ارزیابی هر دو مدل استفاده شده است تا عملکرد مدل‌ها در شرایط برابر و یکسان سنجیده شود.

### ۵-۱ سنجه‌های ارزیابی مدل بازیابی اطلاعات

برای سنجش کیفیت بازیابی انجام شده در بازیابی اطلاعات، سنجه‌های مختلفی قابل استفاده هستند. از رایج‌ترین این سنجه‌ها می‌توان به سنجه‌ی Recall اشاره کرد که در رابطه‌ی ۵ نشان داده شده است (Mitra and Craswell 2018). در این رابطه $(i, d) \in R_q$ نشان‌دهنده‌ی تمام اسناد بازیابی شده توسط سامانه‌ی بازیابی اطلاعات می‌باشد. مقدار $d \in D$ نیز نشان‌دهنده‌ی هر سند مرتبط با پرسه $q$ می‌باشد. مقادیر $rel_q(d)$ همواره ۱ یا ۰ بوده و نشان‌دهنده‌ی ارتباط داشتن یا نداشتن سند مورد بررسی

$d$ با پرسه $q$ می‌باشد. به طور خلاصه، این رابطه نسبت تعداد اسناد مثبت بازیابی شده (صورت کسر) را به تعداد کل اسناد مثبت داخل مجموعه‌داده (مخرج کسر) می‌سنجد.

$$Recall_q = \frac{\sum_{(i,d)\in R_q} rel_q(d)}{\sum_{d\in D} rel_q(d)} \quad (5)$$

از دیگر سنجه‌های مورد استفاده در بازیابی اطلاعات می‌توان به MRR[۱۷] اشاره کرد. برای محاسبه‌ی این سنجه، به ازای هر پرسه و نتایج بازیابی شده‌ی آن به کمک مدل، مقدار RR[۱۸] (که در رابطه‌ی ۶ نشان داده شده است) محاسبه می‌شود و سپس میانگین این مقادیر RR محاسبه و بعنوان MRR در نظر گرفته می‌شود (Mitra and Craswell 2018). در این رابطه، $i$ را می‌توان به عنوان رتبه‌ی هر سند بازیابی شده تفسیر کرد. به عبارت دیگر، برای محاسبه‌ی RR هر سند، معکوس رتبه‌ی اولین سند مثبت محاسبه می‌شود و سپس با محاسبه‌ی میانگین برای این مقادیر RR، مقدار MRR محاسبه می‌شود.

$$RR_q = \max_{(i,d)\in R_q} \frac{red_q(d)}{i} \quad (6)$$

سنجه‌ی دیگری که استفاده از آن در بازیابی اطلاعات رایج می‌باشد، سنجه‌ی Precision می‌باشد که بسیار شبیه به سنجه Recall بوده و در رابطه‌ی ۷ نشان داده شده است. در این رابطه $|R_q|$ نشان‌دهنده‌ی تعداد کل اسناد بازیابی شده توسط سامانه می‌باشد. به عبارت دیگر، این سنجه نسبت تعداد اسناد مثبت بازیابی شده به تعداد کل اسناد بازیابی شده را می‌سنجد.

$$Precision_q = \frac{\sum_{(i,d)\in R_q} rel_q(d)}{|R_q|} \quad (7)$$

همچنین استفاده از گونه‌ی تغییریافته‌ای از سنجه‌ی Precision که با نام Exact Match شناخته می‌شود در ارزیابی‌های سامانه‌های درک مطلب ماشینی بسیار رایج می‌باشد. در این سامانه‌ها، یافتن پاسخ دقیق یک پرسه در سند، یک موفقیت محسوب می‌شود و از همین‌رو، تمامی اسنادی که حاوی پاسخ باشند، مثبت فرض می‌شوند. به عبارت دیگر، برای محاسبه‌ی سنجه‌ی Exact Match کافی است نسبت اسناد بازیابی شده که حاوی پاسخ هستند به کل اسناد بازیابی شده محاسبه شود.

---

[۱۷] Mean Reciprocal Rank

[۱۸] Reciprocal Rank

در این پژوهش، برای ارزیابی از سنجه‌های Recall و MRR استفاده شده است. همینطور در صورتی که پاسخ هر پرسه در دسترس باشد، مقادیر Exact Match نیز محاسبه شده‌اند. لازم به ذکر است که تعداد اسناد بازیابی شده به ازای هر سنجه به صورت num@ و در کنار نام سنجه نشان داده می‌شود. برای نمونه، اگر برای محاسبه‌ی MRR از ۱۰ سند و برای محاسبه‌ی Recall از ۱۰۰ سند استفاده شده باشد، آنها را به ترتیب با MRR@10 و Recall@100 نمایش می‌دهند. همینطور شایان ذکر است که مقادیر حاصل شده از هر یک از این سنجه‌ها یک مقدار اعشاری بین ۰ و ۱ می‌باشد. در این پژوهش، مقادیر حاصل شده از این سنجه‌ها در ۱۰۰ ضرب شده‌اند تا تفسیر کردن آن‌ها راحت‌تر و برحسب درصد باشد.

### ۵-۲ نتایج آموزش و ارزیابی بر روی Trec-DL

مجموعه‌داده‌ی Trec-DL یک مجموعه‌داده‌ی چندسطحی انگلیسی است. در این پژوهش با کمک مجموعه‌داده‌های منتشر شده در سال‌های ۲۰۱۹ و ۲۰۲۰، دو مدل آموزش داده می‌شوند. مدل اول، یک مدل DPR دوسطحی است که تنها قادر به استفاده از سطوح ۰ و ۳ به عنوان اسناد مثبت و منفی است. مدل دوم، یک DPR چندسطحی است که می‌تواند بر خلاف مدل قبلی، علاوه بر سطوح ۰ و ۳، از سطوح ۱ و ۲ نیز استفاده کند. در ادامه‌ی این بخش، با کمک سنجه‌های Recall و MRR عملکرد این دو مدل مقایسه و بررسی می‌شوند. از آنجایی که پاسخ هر پرسه در مجموعه‌داده‌ی Trec-DL وجود ندارد، نمی‌توان سنجه‌هایی چون Exact Match را برای ارزیابی استفاده کرد.

برای ارزیابی دقیق‌تر، فرآیند آموزش هر یک از مدل‌های دوسطحی و چندسطحی سه مرتبه و با دانه‌های تصادفی[۱۹] مختلف انجام شده است[۲۰]. پس از آن میانگین این نتایج محاسبه و به عنوان نتیجه‌ی نهایی ارایه گردیده است. نتایج ارزیابی برای Recall و MRR بر روی مجموعه‌داده‌ی Trec-DL را می‌توان در جدول ۲ و جدول ۳ مشاهده نمود. همانطور که دیده می‌شود، مدل چندسطحی عملکرد بهتری نسبت به همتای دوسطحی خود دارد (۷٫۰۴٪ برای Recall@100 و ۴٫۰۲٪ برای MRR@100). این بهبود عملکرد نشان می‌دهد که تغییرات اعمال شده در مدل اصلی DPR و تبدیل آن به یک مدل چندسطحی، موثر واقع شده و مدل جدید چندسطحی DPR قادر بوده به درستی از اطلاعات اضافه‌ی

---

[۱۹] Random Seed

[۲۰] دانه تصادفی عددی تصادفی است که برای مقداردهی پارامترهای مدل و نوع درهم‌سازی داده‌های آموزشی و ... بکار میرود و از همین رو، تغییر دادن آن می‌تواند بر نحوه‌ی آموزش دیدن مدل و نتایج حاصل شده از مدل در مرحله‌ی ارزیابی موثر باشد. از آنجایی که آموزش دادن DPR یک فرآیند زمانبر می‌باشد، در این بخش برای ارزیابی، آموزش‌ها تنها با سه دانه‌ی تصادفی مختلف انجام شده.

فراهم شده (روابط سطح ۱ و ۲) در جهت افزایش دقت مدل بازیابی اطلاعات استفاده کند. حال که بر اساس نتایج بدست آمده، از عملکرد صحیح مدل چندسطحی اطمینان حاصل شد، در بخش بعدی به ارزیابی مجموعه‌داده‌ی فارسی و بررسی درستی مجموعه‌داده و نیز ارزیابی کارآیی روش‌های تولید آن پرداخته می‌شود.

**جدول ۲ : مقادیر Recall برای مجموعه‌داده‌های دوسطحی و چندسطحی Trec-DL**

| مجموعه داده | Recall@1 | Recall@10 | Recall@20 | Recall@100 |
|---|---|---|---|---|
| دوسطحی - اجرای اول | 4.26 | 14.25 | 18.86 | 33.58 |
| دوسطحی - اجرای دوم | 4.95 | 16.37 | 21.68 | 37.58 |
| دوسطحی - اجرای سوم | 3.20 | 11.25 | 15.31 | 28.66 |
| چهارسطحی - اجرای اول | 3.88 | 13.73 | 18.31 | 31.22 |
| چهارسطحی - اجرای دوم | 8.28 | 22.38 | 27.72 | 43.33 |
| چهارسطحی - اجرای سوم | 9.22 | 24.98 | 31.23 | 46.38 |
| دوسطحی - میانگین | 4.14 | 13.96 | 18.62 | 33.27 |
| چهارسطحی - میانگین | 7.13 | 20.36 | 25.75 | 40.31 |

**جدول ۳: مقادیر MRR برای مجموعه‌داده‌های دوسطحی و چندسطحی Trec-DL**

| مجموعه داده | MRR@1 | MRR@10 | MRR@20 | MRR@100 |
|---|---|---|---|---|
| دوسطحی - اجرای اول | 4.26 | 6.97 | 7.29 | 7.64 |
| دوسطحی - اجرای دوم | 4.95 | 8.04 | 8.40 | 8.78 |
| دوسطحی - اجرای سوم | 3.20 | 5.37 | 5.65 | 5.95 |
| چهارسطحی - اجرای اول | 3.88 | 6.55 | 6.87 | 7.17 |
| چهارسطحی - اجرای دوم | 8.28 | 12.17 | 12.55 | 12.91 |
| چهارسطحی - اجرای سوم | 9.22 | 13.58 | 14.01 | 14.37 |
| دوسطحی - میانگین | 4.14 | 6.79 | 7.11 | 7.46 |
| چهارسطحی - میانگین | 7.13 | 10.77 | 11.14 | 11.48 |

**۵-۳ نتایج آموزش و ارزیابی بر روی مجموعه‌داده‌ی فارسی**

در بخش (۳-۲) روش‌های مربوط به ساخت یک مجموعه‌داده‌ی چندسطحی جدید به کمک قوانین دست‌ساز و طبقه‌بندی خودکار مورد بحث قرار گرفت. در حالت طبقه‌بندی خودکار، به منظور انتخاب بهترین طبقه‌بند جهت ساخت مجموعه‌داده چندسطحی جدید، تعدادی طبقه‌بند مرسوم آزموده شدند که دقت طبقه‌بندی آن‌ها در جدول ۴ ذکر شده است. همان‌طور که مشاهده می‌شود، از میان این طبقه‌بندها، جنگل تصادفی عملکرد بهتری را ارائه کرده است. به همین دلیل، در این پژوهش نیز از همین طبقه‌بند جهت ساخت مجموعه‌داده‌ی چندسطحی فارسی استفاده شده است.

**جدول ٤: دقت طبقه‌بندهای آزموده شده برای طبقه بندی اسناد**

| طبقه بند | F1 (micro) | F1 (macro) |
|---|---|---|
| جنگل تصادفی | **91.10** | **92.88** |
| درخت تصمیم | 78.00 | 82.39 |
| ماشین بردار پشتیبان | 50.60 | 59.89 |
| SGD | 41.20 | 44.75 |
| نزدیکترین همسایه | 29.60 | 28.34 |
| AdaBoost | 40.70 | 45.38 |
| شبکه عصبی (۴ لایه مخفی) | 48.60 | 58.43 |

به منظور ارزیابی تاثیرات مثبت مجموعه‌داده چندسطحی در مقابل مجموعه‌داده دوسطحی در سامانه‌های بازیابی اطلاعات، عملکرد هر دو مجموعه‌داده چندسطحی فارسی ساخته شده (یکی با قوانین دست‌ساز و یکی با طبقه‌بند جنگل تصادفی) با عملکرد مجموعه‌داده دوسطحی فارسی (به عنوان مدل پایه) مقایسه شده است. در جدول ۵ می‌توان مقادیر Exact Match@100 برای مدل دوسطحی پایه، مدل چندسطحی آموزش دیده بر روی مجموعه‌داده‌ی مبتنی بر قوانین دست‌ساز (با عنوان «چندسطحی قانون-محور») و مدل چندسطحی آموزش دیده بر روی مجموعه‌داده‌ی ایجاد شده با کمک جنگل تصادفی (با عنوان «چندسطحی جنگل تصادفی») را مشاهده کرد. هر یک از این مقادیر، نشان‌دهنده‌ی میانگین هفت اجرای مختلف با دانه‌های متفاوت می‌باشند. همان‌طور که مشاهده می‌شود، هر دو مدل چندسطحی، عملکرد بهتری نسبت به همتای دوسطحی خود دارند. مدل چندسطحی قانون-محور بهبود ۱٫۸۷ درصدی Exact Match را حاصل می‌کند و یا به تعبیر دیگر، مدل چندسطحی توانسته به ازای هر پرسه، حدودا ۲ سند بیشتر (که شامل پاسخ نیز می‌باشد) نسبت به مدل دوسطحی بازیابی کند. مشاهده‌ی

دیگری که در این جدول می‌توان داشت، برتری روش قانون-محور نسبت به روش جنگل تصادفی می‌باشد. در واقع علت این امر در استفاده از ویژگی‌های مرتبط با پاسخ در این روش می‌باشد.

**جدول ۵: مقادیر Exact Match برای مجموعه‌داده‌های دوسطحی و چندسطحی فارسی**

| مجموعه داده | Exact Match @100 |
|---|---|
| دوسطحی | 6.21 |
| چندسطحی قانون-محور | 8.08 |
| چندسطحی جنگل تصادفی | 7.35 |

در نهایت در جدول ۶ و جدول ۷ می‌توان به ترتیب ارزیابی Recall و MRR برای سه مجموعه‌داده را مشاهده کرد. در این جداول می‌توان عملکرد بهتر مجموعه‌داده‌ها و مدل‌های چندسطحی (به جز Recall@100 برای مدل چندسطحی جنگل تصادفی) را مشاهده کرد. از آنجایی که در بخش (۵-۲) عملکرد صحیح مدل‌ها مورد بررسی قرار گرفت، می‌توان ادعا کرد که بهبود حاصل شده در این بخش مربوط به کارا بودن دو روش ارائه شده این پژوهش (روش مبتنی بر قانون و روش مبتنی بر طبقه‌بند، معرفی شده در بخش‌های ۳-۲-۱ و۳-۲-۲) می‌باشد. به عبارت دیگر، با کمک این دو روش، می‌توان به طور موثر یک مجموعه‌داده بازیابی اطلاعات دوسطحی را به یک مجموعه‌ی چندسطحی تبدیل کرد.

**جدول ٦: مقادیر Recall برای مجموعه‌داده‌های دوسطحی و چندسطحی فارسی**

| مجموعه داده | Recall@1 | Recall@10 | Recall@20 | Recall@100 |
|---|---|---|---|---|
| دوسطحی | 14.62 | 39.73 | 48.04 | 67.81 |
| چندسطحی قانون-محور | 17.97 | 41.58 | 48.60 | 68.62 |
| چندسطحی جنگل تصادفی | 16.63 | 42.58 | 50.15 | 67.50 |

**جدول ۷: مقادیر MRR برای مجموعه‌داده‌های دوسطحی و چندسطحی فارسی**

| مجموعه داده | MRR@1 | MRR@10 | MRR@20 | MRR@100 |
|---|---|---|---|---|
| دوسطحی | 14.62 | 21.96 | 22.53 | 23.02 |
| چندسطحی قانون-محور | 17.97 | 24.74 | 25.23 | 25.71 |
| چندسطحی جنگل تصادفی | 16.63 | 24.36 | 24.89 | 25.33 |

## ٦. نتیجه گیری

در مباحث پردازش زبان طبیعی، زبان فارسی یک زبان کم منبع می‌باشد و نیازمندی‌های زیادی در آن وجود دارد. در چند سال اخیر و با پیشرفت مدل‌های زبان، سامانه‌های بازیابی اطلاعات زیادی توسط محققان بر پایه‌ی این ابزارهای جدید ارائه گردیده است؛ اما برای زبان فارسی، هیچ پژوهشی در این زمینه انجام نشده و در نتیجه، هیچ مجموعه‌داده مناسبی برای آن وجود ندارد.

در این پژوهش، پس از تبدیل یک مجموعه‌داده‌ی درک مطلب ماشینی به یک مجموعه‌داده‌ی بازیابی اطلاعات دوسطحی، دو روش مختلف برای تبدیل کردن مجموعه‌ی ساخته شده به یک مجموعه‌داده‌ی چندسطحی ارائه شد. سپس مدل DPR به عنوان مدل پایه‌ی این پژوهش مورد استفاده قرار گرفت و تغییرات لازم در آن برای درک مجموعه‌داده‌های چندسطحی مورد بحث قرار گرفت.

با کمک دو مجموعه‌داده‌ی چند سطحی جدید و مجموعه‌داده‌ی دوسطحی و همینطور مدل‌های دوسطحی و چندسطحی، می‌توان به ارزیابی راه‌کارهای ارائه شده پرداخت. ارزیابی‌های انجام شده بر روی این مجموعه‌ها نشان می‌دهد که در صورت فراهم بودن مجموعه‌داده‌ی چندسطحی، مدل بازیابی اطلاعات می‌تواند نتایج بهتری نسبت به مدل دوسطحی حاصل کند. همینطور با کمک روش‌های پیشنهاد شده در این پژوهش، می‌توان یک مجموعه‌ی دوسطحی را به یک مجموعه‌ی چندسطحی تبدیل کرد تا بتوان از مزایای مدل چندسطحی، بهره برد.

## ۷. منابع

Abadani, Negin, Jamshid Mozafari, Afsaneh Fatemi, Mohammd Ali Nematbakhsh, and Arefeh Kazemi. 2021. "*ParSQuAD: Machine Translated SQuAD Dataset for Persian Question Answering*." in 2021 7th International Conference on Web Research (ICWR). IEEE.

Ayoubi Sajjad & Davoodeh, Mohammad Yasin. 2021. "*PersianQA: A Dataset for Persian Question Answering*." GitHub Repository.

Bajaj, Payal, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, and Tri Nguyen. 2016. "*Ms Marco: A Human Generated Machine Reading Comprehension Dataset*." ArXiv Preprint ArXiv:1611.09268.

Craswell, Nick, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. "*Overview of the TREC 2020 Deep Learning Track*." ArXiv Preprint ArXiv:2102.07662.

Craswell, Nick, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. "*Ms Marco: Benchmarking Ranking Models in the Large-Data Regime*." pp. 1566–76 in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.

Craswell, Nick, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M. Voorhees. 2020. "*Overview of the TREC 2019 Deep Learning Track*." ArXiv Preprint ArXiv:2003.07820.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "*BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*." pp. 4171–86 in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics.

Hashemi, Helia, Mohammad Aliannejadi, Hamed Zamani, and W. Bruce Croft. 2020. "*ANTIQUE: A Non-Factoid Question Answering Benchmark*." Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 12036 LNCS:166–73. doi: 10.1007/978-3-030-45442-5_21.

Johnson, Jeff, Matthijs Douze, and Hervé Jégou. 2019. "*Billion-Scale Similarity Search with GPUs*." IEEE Transactions on Big Data 7(3):535–47.

Karpukhin, Vladimir, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. "*Dense Passage Retrieval for Open-Domain Question Answering*." pp. 6769–81 in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Kazemi, Arefeh, Jamshid Mozafari, and Mohammad Ali Nematbakhsh. 2022. "*PersianQuAD: The Native Question Answering Dataset for the Persian Language*." IEEE Access 10:26045–57. doi: 10.1109/ACCESS.2022.3157289.

Khashabi, Daniel, Arman Cohan, Siamak Shakeri, Pedram Hosseini, Pouya Pezeshkpour, Malihe Alikhani, Moin Aminnaseri, Marzieh Bitaab, Faeze Brahman, and Sarik Ghazarian. 2021. "*ParsiNLU: A Suite of Language Understanding Challenges for Persian*." Transactions of the Association for Computational Linguistics 9:1163–78.

Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, and Kenton Lee. 2019. *"Natural Questions: A Benchmark for Question Answering Research*." Transactions of the Association for Computational Linguistics 7:453–66.

Lin, Jimmy, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. "*Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations*." pp. 2356–62 in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.

Liu, Ye, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and S. Yu Philip. 2021. "*Dense Hierarchical Retrieval for Open-Domain Question Answering*." pp. 188–200 in Findings of the Association for Computational Linguistics: EMNLP 2021.

Liu, Zhenghao, Kaitao Zhang, Chenyan Xiong, Zhiyuan Liu, and Maosong Sun. 2021. "*OpenMatch: An Open Source Library for NEU-IR Research*." pp. 2531–35 in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.

Mitra, Bhaskar, and Nick Craswell. 2018. "*An Introduction to Neural Information Retrieval*." Foundations and Trends® in Information Retrieval 13(1):1–126.

Qu, Yingqi, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. "*RocketQA: An Optimized Training Approach to Dense Passage Retrieval for Open-Domain Question Answering*." pp. 5835–47 in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Rajpurkar, Pranav, Robin Jia, and Percy Liang. 2018. "*Know What You Don't Know: Unanswerable Questions for SQuAD*." pp. 784–89 in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers).

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. "*SQuAD: 100,000+ Questions for Machine Comprehension of Text*." pp. 2383–92 in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Robertson, Stephen, and Hugo Zaragoza. 2009. "*The Probabilistic Relevance Framework: BM25 and Beyond*." Foundations and Trends® in Information Retrieval 3(4):333–89.

Salton, Gerard, and Christopher Buckley. 1988. "*Term-Weighting Approaches in Automatic Text Retrieval*." Information Processing & Management 24(5):513–23.

Xia, Fen, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. "*Listwise Approach to Learning to Rank: Theory and Algorithm*." pp. 1192–99 in Proceedings of the 25th international conference on Machine learning.

Yang, Peilin, Hui Fang, and Jimmy Lin. 2017. "*Anserini: Enabling the Use of Lucene for Information Retrieval Research*." pp. 1253–56 in Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval.

Zhang, Xinyu, Andrew Yates, and Jimmy Lin. 2020. "*A Little Bit is Worse than None: Ranking with Limited Training Data*." pp. 107–12 in Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing.

# Multi-level Persian Dataset for Information Retrieval

Ali Abedzadeh

Master of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, a.abedzadeh@eng.ui.ac.ir

Reza Ramezani
Assistant Professor, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, r.ramezani@eng.ui.ac.ir

Afsaneh Fatemi
Associate Professor, Faculty of Computer Engineering,
University of Isfahan, Isfahan, Iran, a_fatemi@eng.ui.ac.ir

**Abstract.** Information retrieval systems are an essential part of many smart systems. The applications of this research field include search engines such as Google and Bing, question-answering systems, modern databases, etc. An information retrieval system tries to retrieve documents related to a question/query. The retrieval is done from a large collection of documents, and the size of this collection can be from a few thousand documents to millions of documents. In recent years, a lot of research has been done to develop information retrieval systems using language models. However, in this research field, no research has been done for the Persian language. One of its main reasons is the lack of a suitable Persian dataset for training language models. In this research, first, a Persian dataset for information retrieval is presented. After that, methods for enriching this data set are investigated. This enrichment is done by defining multi-level relationships between a document and a question. In this regard, the new dataset can show the relationship between question and document in four levels (unrelated - related - highly related - completely related) instead of two levels (completely unrelated - completely related). The name of the generated dataset is PersianMLIR. Experiments show that by using multi-level relationships, the performance of the system improves for both Persian and English languages, where the improvement is 1.87% for the Persian language. The results conclude that enriching information retrieval datasets by increasing the number of relations between query and document lead to improving the performance of information retrieval systems.

**Keywords.** Information Retrieval, Language Models, Information Retrieval Dataset, Persian Dataset

**علی عابدزاده**

متولد سال ۱۳۷۵ دارای مدرک تحصیلی کارشناسی ارشد در رشته مهندسی کامپیوتر از دانشگاه اصفهان است. عنوان پایان‌نامه ایشان «ساخت مجموعه‌داده چندسطحی جهت آموزش مدل‌های زبان برای زبان کم‌منبع فارسی» می‌باشد که تحت راهنمایی جناب آقای دکتر رضا رمضانی و مشاوره سرکار خانم دکتر افسانه فاطمی به پایان رساندند.

**رضا رمضانی**

متولد سال ۱۳۶۸ دارای مدرک تحصیلی دکتری تخصصی در رشته مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه فردوسی مشهد است. ایشان هم‌اکنون استادیار گروه مهندسی نرم‌افزار دانشکده‌ی مهندسی کامپیوتر دانشگاه اصفهان است. پردازش زبان طبیعی، تحلیل داده و وب معنایی از جمله علایق پژوهشی وی است.

**افسانه فاطمی**

متولد ۱۳۵۲ دارای مدرک تحصیلی دکتری تخصصی در رشته مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه اصفهان است. ایشان هم‌اکنون دانشیار گروه مهندسی نرم‌افزار دانشکده‌ی مهندسی کامپیوتر دانشگاه اصفهان است. سیستم‌های پیچیده، کلان‌داده، سیستم‌های پرسش‌پاسخ و سیستم‌های گفتگو از جمله علایق پژوهشی وی است.