شناسایی و استخراج سوال‌های تکراری در Stack Overflow

پویا خانی[1]، پدرام شاطری[2]

1 دانشکده کامپیوتر، دانشگاه صنعتی شریف، تهران، **pouya.khani@sharif.edu**

2 دانشکده کامپیوتر، دانشگاه صنعتی شریف، تهران، **pedramshateri@sharif.edu**

چکیده

امروزه با پیشرفت تکنولوژی و افزایش استفاده از زبان‌های برنامه نویسی در حوزه‌های مختلف، بیش از هر زمان دیگری نیاز به یک انجمن برای تعامل بین برنامه‌نویسان احساس می‌شود. از این‌رو پلتفرم‌ها و وبسایت‌های بسیاری برای این هدف پیاده سازی شده‌اند که به‌وسیله آن‌ها برنامه‌نویسان و متخصصان در این حوزه بتوانند پرسش‌های خود را در آن مطرح کنند و متخصصین دیگر به آن پاسخ دهند و به این‌گونه هم‌افزایی اطلاعات صورت پذیرد. وبسایت stack overflow یکی از قدیمی‌ترین و بهترین انجمن‌های پرسش و پاسخ مخصوص برنامه‌نویسان است و به‌همین دلیل در این پژوهش ما بر روی داده‌های این وبسایت آزمایشات خود را انجام می‌دهیم. به دلیل گسترش روزافزون برنامه‌نویسی و تعدد برنامه‌نویسان فعال در این حوزه، طبیعی است که نرخ بسیار زیادی از پرسش و پاسخ‌های مرتبط، در این‌گونه از وبسایت‌ها در حال اتفاق افتادن است لذا بسیار پیش می‌آید که سوال‌های تکراری توسط کاربران مختلف به مرور زمان مطرح شود و این چالش یکی از اصلی ترین چالش‌های پیش‌امده برای این‌گونه پلتفرم‌ها می‌باشد. از این‌رو این سایت از کاربران می‌خواهد که در صورتی که سوال تکراری را مشاهده کردند، با ذکر آدرس سوال اولیه، آن‌را گزارش کنند اما با توجه به حجم بسیار زیاد سوالات مطرح شده در بازه‌های زمانی مختلف، شناسایی و حذف تمامی سوالات تکراری به این روش که روشی دستی محسوب می‌شود، غیرممکن است. با پیشرفت هوش مصنوعی و بالخصوص الگوریتم‌ها و روش‌های یادگیری ماشین، این نوع چالش‌ها قابل حل شده است زیرا می‌توانیم با استفاده از این رویکردها، به صورت اتوماتیک سوالات تکراری را شناسایی کنیم.

در این مقاله در ابتدا پژوهش‌های پیشین در حوزه شناسایی سوالات تکراری به کمک یادگیری ماشین مرور شده و سپس این رویکردها با یکدیگر مقایسه می‌شوند. در آخر روش پیشنهادی جدیدی برای این مسئله ارائه شده و بر روی مجموعه داده‌های سایت stack overflow آزمایش می‌شود.

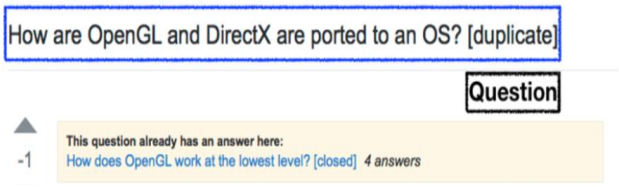
کلمات کلیدی

پرسش‌های تکراری، شباهت‌یابی معنایی، پردازش زبان طبیعی، انجمن‌های پرسش و پاسخ، مجموعه دادگان stack overflow

۱- مقدمه

امروزه با پیشرفت تکنولوژی و بستر اینترنت، مردم می‌توانند به راحتی با یکدیگر ارتباط برقرار کرده و هم‌چنین از نظرات یکدیگر در حوزه‌های مختلف آگاه شوند. به این منظور پلتفرم‌ها و وبسایت‌های زیادی برای پرسش و پاسخ بین کاربران ایجاد شده و روز به روز کاربران این‌گونه بسترها در حال افزایش هستند. هم‌چنین به‌دلیل کاربرد بسیار زیاد کامپیوتر در زندگی امروزی، برنامه‌نویسی محبوبیت بسیار زیادی پیدا کرده است و همان‌طور که گفته شد، برنامه‌نویسان نیز مانند دیگر حوزه‌ها تمایل به هم‌افزایی و آگاهی از نظرات یکدیگر دارند. لذا وبسایت‌ها و پلتفرم‌هایی نیز برای این دسته از کاربران یعنی برنامه‌نویسان توسعه داده شده است. یکی از قدیمی‌ترین، قدرتمندترین و معروف‌ترین وبسایت‌های حال حاضر در این حوزه، وبسایت Stack Overflow است که طبق آمار گرفته شده، تا اواسط سال ۲۰۱۹ میلادی، ۱۸۱۷۹۷۸ پست(شامل سوال و پاسخ‌های آن) در این وبسایت به اشتراک گذاشته شده است.

به دلیل حجم بسیار زیاد کاربران و سوال‌های پرسیده شده در این وبسایت، امکان طبقه‌بندی و نمایش تمام سوالات به صورت یکپارچه وجود ندارد از این رو، احتمال اینکه سوالی تکراری توسط کاربر پرسیده شود که در گذشته پرسیده شده و پاسخ به آن داده شده است، بسیار زیاد است. از طرفی مشخصا سوالات تکراری حداقل در یک کلمه متفاوت هستند و نمی‌توان به صورت سنتی و مبتنی بر شباهت‌یابی جملات، این‌گونه سوال‌ها را شناسایی کرد. هم‌چنین ممکن است دو سوال فقط در یک کلمه متفاوت باشند اما کاملا سوال‌های متفاوتی از لحاظ معنایی به حساب بیایند. لذا شناسایی این‌گونه سوالات، بسیار چالش برانگیز بوده است. به عنوان مثال، شکل ۱ دو سوال تکراری را نشان می‌دهد که توسط وبسایت شناسایی شده اند و در ظاهر تفاوت بسیار زیادی با یکدیگر دارند.



شکل (۱): دو پرسش تکراری در سایت stack overflow که توسط وبسایت شناسایی شده است.

با توجه به این مشکل، در ابتدا سایت از کاربران می‌خواست که در صورتی که سوال تکراری مشاهده کردند، با ارجاع دادن به سوال پیشین، تکراری بودن سوال جدید را گزارش دهند. دور از ذهن نیست که این روش کارایی بسیار کمی دارد زیرا اولا یک کاربر نمی‌تواند تمام سوالات را در ذهن خود داشته باشد و شناسایی سوالات تکراری به این روش بسیار پرهزینه و زمان‌بر اتفاق می‌افتد. ثانیا ممکن است سوالی که تکراری نباشد به عنوان یک سوال تکراری گزارش شود لذا بدون اعتبارسنجی نمی‌توان به این‌گونه گزارش‌ها استناد کرد و اعتبارسنجی آن‌ها نیز بسیار زمان‌بر و پرهزینه است. برای این مسئله [۱] روشی ارائه می‌کند که به وسیله آن بتوانیم سوالاتی که به اشتباه تکراری در نظر گرفته شده و گزارش شده‌اند را شناسایی کنیم. حتی اگر از اعتبار تمام گزارش‌های کاربران مطمئن شویم و روشی کم‌هزینه برای این‌کار پیدا کنیم، بازهم نیازمند یک شناسایی اتوماتیک غیر انسانی برای سوالات تکراری هستیم زیرا مهم است که بتوانیم درصد بسیار زیادی از سوالات تکراری را شناسایی کنیم که این‌کار توسط عامل انسان تقریبا غیرممکن است.

با پیشرفت هوش مصنوعی و حوزه یادگیری ماشین، پژوهشگران و صنعت‌گران از این روش‌ها برای حل مسئله‌های بسیاری کمک گرفته‌اند لذا پژوهشگرانی که در حوزه شناسایی سوالات تکراری که مسئله مورد بحث در این پژوهش نیز هست تحقیق می‌کردند نیز بر آن شدند که از این روش‌ها برای طراحی الگوریتم و سیستمی برای شناسایی سوالات تکراری که به صورت متن هستند، استفاده کنند. در این پژوهش‌ها از الگوریتم‌های یادگیری ماشین مانند جنگل تصادفی، شبکه بیزین و غیره استفاده شده است. هم‌چنین در سال‌های اخیر با پیشرفت حوزه یادگیری ماشین و پیدایش شبکه‌های عصبی عمیق، استفاده از الگوریتم‌ها و روش‌های یادگیری ژرف در این حوزه نیز مورد استقبال قرار گرفته است.

مشارکت های عمده در این پژوهش شامل موارد زیر است:

- در ابتدا مروری بر روش‌های پیشین که مبتنی بر یادگیری ماشین و یادگیری ژرف هستند برای شناسایی سوال‌های تکراری در مخزن stack overflow انجام می‌شود.
- سپس روش پیشنهادی بر پایه روش‌های پیشین و با اعمال تغییراتی در آن‌ها برای شناسایی سوالات تکراری بیان می‌شود.

ادامه پژوهش به این صورت است. در قسمت ۲ پژوهش‌های پیشین در راستای شناسایی سوال‌های تکراری مرور می‌شود. سپس در قسمت ۳ روش پیشنهادی جدیدی برای شناسایی این‌گونه سوالات ارائه شده و در قسمت ۴ به مقایسه پژوهش‌های پیشین در این حوزه میپردازیم و در نهایت در قسمت ۵ هم نتیجه گیری نهایی را انجام می‌دهیم.

۲- کارهای پیشین

همان‌طور که گفته شد، شناسایی سوالات تکراری در انجمن‌های مختلف یکی از چالش‌های معروف این حوزه بوده و هست لذا پژوهش‌های زیادی در این راستا راه‌حل های خود را ارائه داده‌اند. در این بخش به پژوهش‌های پیشین اشاره می‌شود.

[۱] از سه معیار برای شباهت یابی سوالات استفاده می‌کند. معیار اول شباهت برداری[۱] است که لازم است هر سوال به یک بردار به نام بردار نهان تبدیل شده و سپس این بردارها توسط این معیار مقایسه شوند. معیار دوم شباهت موضوعی[۲] است که موضوع هر سوال توسط یک الگوریتم بر پایه LDA شناسایی شده و سپس این موضوع‌ها با یکدیگر مقایسه می‌شوند. معیار سوم، معیاری مبتنی بر مقایسه زوج کلمات استفاده شده در سوالات است به این صورت که ابتدا تمام زوج کلمات موجود در متن سوال استخراج شده و سپس زوج کلمات سوالات مختلف با یکدیگر مقایسه می‌شوند. هرچه اشتراک این زوج کلمات در یک سوال

با سوال دیگر بیشتر باشد، یعنی به احتمال بیشتری سوالات تکراری هستند. برای بردارسازی از سوالات نیز از روش Doc2Vec استفاده می‌شود. در این مقاله ادعا می‌شود که در جفت سوالات تکراری مختلف، یک سری جفت واژه‌های تکراری، تکرار می‌شوند به همین منظور نیز از روش‌های Association pair mining برای بوجود آوردن ویژگی دیگری استفاده کرده است. مدل‌های یادگیری ماشین استفاده شده در این مقاله بسیار زیاد هستند که k نزدیک‌ترین همسایه و جنگل تصادفی و SVM خطی از جمله بهترین مدل‌های استفاده شده در آن است. در قسمت ارزیابی این مقاله، روش پیشنهادی خود را با روش پیشین که بر اساس term frequency بوده مقایسه می‌کنند و سپس recall آن‌ها را نسبت به یکدیگر میسنجند. مشاهده می‌شود که روش پیشنهادی آن‌ها بهتر عمل می‌کند.

در [۲] بهبودهایی بر روی مقاله قبلی انجام داده است به این صورت که به جای آن‌که تکراری بودن یک سوال جدید با تمام سوالات موجود مقایسه شود، با در نظر گرفتن برچسب سوال در مرحله اول تعداد زیادی از سوالات هرس می‌شوند سپس مدل احتمالاتی آن با دیگر سوالات مقایسه شده و تعداد دیگری هرس اتفاق می‌افتد. در آخر با استفاده از روش‌های BM25 و Jelinek-Mercer و دیریکله و DFRS و LDA هرس‌های نهایی اتفاق می‌افتد و به این‌گونه تعداد سوالات برای بررسی بسیار کاهش می‌یابد و آن سوال با مجموعه کوچک‌تری از سوالات موجود مقایسه می‌شود. هم‌چنین برای شباهت‌یابی، به جای شباهت موضوعی، از Relevance Feature استفاده می‌کند و از Recall برای اعلام نتایج بدست آمده بهره می‌گیرد. مشاهده می‌شود که Recall این روش از روش قبلی بهتر شده و در عین حال در زمان و انرژی صرفه‌جویی می‌شود.

[۳] با استفاده از شبکه‌های کانولوشنی، سوالات را به شکل یک بردار کد می‌کند و به این وسیله بردارهای مشابه که نمایانگر سوالات تکراری هستند را شناسایی می‌کند. در این شبکه کانولوشنی، هرچه خروجی شبکه عدد بیشتری باشد، یعنی ورودی مربوطه که همان سوال جدید است، با احتمال بیشتری تکراری است. هم‌چنین در این پژوهش وقتی شبکه را با داده‌های مربوط به همان حوزه به اصطلاح fine-tune می‌کند، نتیجه بهتری نیز می‌گیرد. ورودی شبکه عصبی در این روش، خروجی الگوریتم Word2Vec است. در قسمت ارزیابی این پژوهش، روش پیشنهادی که از شبکه کانولوشنی استفاده می‌کند با دوتا از بهترین روش‌های قبلی نیز مقایسه می‌شود که نتایج نشان می‌دهند که روش پیشنهادی آن‌ها precision بهتری می‌دهد.

به دلیل آن‌که پیاده‌سازی‌های روش‌های ارائه‌شده در [۴] و [۵] در دسترس عموم قرار نگرفت، در [۶] بدون ارائه دادن روش جدیدی، به پیاده‌سازی عمومی این دو روش روی آورده است. روش موجود در [۴] این‌طور است که ابتدا داده‌ها پیش‌پردازش شده و سپس به عنوان آرگومان‌های ورودی یک تابع به آن پاس داده می‌شوند. این تابع یک جمع وزن‌دار انجام می‌دهد به این صورت که ۴ فاکتور موضوع سوال، برچسب(تگ) سوال، متن سوال و عنوان سوال را در دو سوال ورودی بررسی کرده و میزان شباهت بین آن‌ها را به صورت وزن‌دار جمع می‌کند. هم‌چنین همپوشانی مولفه‌ای[۳]، نوع این همپوشانی و شباهت شبکه‌کلمات[۴] نیز در این روش لحاظ می‌شوند. برای کاهش فضای حالت هم از روش BM25 نیز استفاده می‌شود.

روش موجود در [۵] این‌طور است که یک مدل Logistic Regression را برای دسته‌بندی آموزش می‌دهد که این مدل دو سوال را توسط روش BOW به دو بردار تبدیل کرده و سپس شباهت کسینوسی برچسب، عنوان و متن هردو سوال را محاسبه کرده و سپس با روش LDA شباهت موضوع آن‌هارا نیز محاسبه کرده و این اطلاعات را به عنوان ورودی می‌گیرد و پیش‌بینی می‌کند که آیا دوسوال تکراری هستند یا خیر.

در [۶] همان‌طور که گفته شد، دو روش مقالات قبلی پیاده‌سازی شده است اما Recall ای که به عنوان خروجی نتیجه می‌شود از دو مقاله اصلی کم‌تر می‌شود. در این مقاله گفته شده که عنوان یک سوال از متن آن بسیار ویژگی تاثیرگذارتری است. هم‌چنین گفته شده که موضوع سوالات کم اهمیت‌ترین ویژگی برای تشخیص تکراری بودن دو سوال است.

در [۷] از شبکه‌های عصبی برای تشخیص تکراری بودن سوالات استفاده می‌شود به این صورت که ابتدا یک Representation از سوالات به عنوان Embedded Vector تولید شده سپس از سه بلاک شبکه عصبی کانولوشنی[۵]، بازگشتی[۶] و بازگشتی با حافظه طولانی‌مدت[۷] عبور می‌کند تا مدل اصلی یادگرفته شود. استفاده از این روش از روش‌های قبلی Recall بهتری را نتیجه می‌دهد.

در [۸] ابتدا به صورت عملی نتیجه می‌شود که تجربه و قدمت یک کاربر در پرسیدن سوال تکراری توسط او تاثیر دارد به طوری‌که اغلب سوالات تکراری توسط کاربران جدید و یا قدیمی با فعالیت کمتر پرسیده می‌شود. هم‌چنین گفته شده که تعداد سوالاتی که کاربر به آن پاسخ داده است تاثیری در پرسیدن یا نپرسیدن سوال تکراری از جانب او ندارد. هم‌چنین در این مقاله گفته شده که سوال‌های تکراری می‌توانند مفید باشند، از این جهت که انواع مختلف پاسخ و رویکردهای جدیدتر برای پاسخ‌دهی به سوال مربوطه با حضور سوالات تکراری می‌توانند در دسترس باشند.

[۹] ابتدا بردار کلماتی توسط الگوریتم Word2Vec از سوالات ساخته شده سپس کلمات موجود در این بردار کلمات با توجه به تعداد تکرار آن در متن و کل داده‌ها وزن‌دار می‌شود و در آخر با استفاده از روش SimHash برداری برای هر کدام از متن‌ها ساخته می‌شود. سپس برای پیدا کردن سوال‌های تکراری از فاصله همینگ این دو بردار استفاده می‌شود. برای ارزیابی این روش از سه معیار Accuracy و Recall و F1 استفاده می‌شود.

در [۱۰] ابتدا سوالات توسط روش Word2Vec به یک بردار تبدیل شده سپس به عنوان ورودی به سه شبکه کانولوشنی و بازگشتی و بازگشتی با حافظه بلندمدت داده می‌شود. Recall این روش از روش‌های قبلی بهتر می‌شود.

۳- روش پیشنهادی

روش پیشنهادی این نوشته از ۲ مرحله تشکیل شده است. در مرحله اول داده‌های ورودی پیش‌پردازش شده تا کیفیت نتیجه خروجی افزایش یابد و هم‌چنین با استفاده از یک سری جفت سوال خصوصیات مختلف هر یک از جفت‌ها را بدست می‌آوریم و در مرحله بعد با تزریق آن‌ها به مدل‌های مختلف، مدل‌ها را آموزش می‌دهیم. برخی از این جفت سوال‌ها تکراری یکدیگر هستند و بقیه جفت‌ها به صورت تصادفی با هم جفت شده‌اند(تکراری یکدیگر نیستند) سپس معیارهای ارزیابی معروف را بر روی نتایج بدست آمده اعمال می‌کنیم. در این مرحله با استفاده از مدل آموزش داده‌شده، به ازای هر سوال ورودی، مجموعه‌ای از سوالات تکراری شناسایی شده و در فایلی ذخیره می‌شوند. در ادامه دو مرحله گفته شده به صورت کامل توضیح داده خواهند شد.

۳-۱- پیش‌پردازش

ابتدا سوال‌های تکراری و غیر تکراری را از مخزن stackexchange استخراج کرده و سپس بر روی آن‌ها پیش‌پردازش‌های زیر را به ترتیب انجام می‌دهیم (به دلیل کمبود قدرت پردازشی و با توجه به حجیم بودن مجموعه‌داده‌های مورد نظر، از ۰/۲ داده‌ها استفاده می‌کنیم):

۱. در ابتدا متن و عنوان و برچسب‌های سوالات را جداگانه در متغیرهایی ذخیره کرده تا کار کردن با آن‌ها آسان شود.
۲. تگ‌های HTML را در تمامی بخش‌های سوال حذف می‌کنیم چون اطلاعات غیرمفیدی را در خود دارند.
۳. تمامی حروف بخش‌های سوال را به حالت کوچک[۸] تبدیل می‌کنیم.
٤. تمامی اعداد را نیز در این بخش‌ها حذف می‌کنیم.
٥. تمامی حروف اضافه را نیز حذف می‌کنیم.
٦. جملات و کل متون را تبدیل به توکن‌های مجزا کرده و سپس کلمات پایانی[۹] را از آن‌ها حذف می‌کنیم.
۷. تمامی توکن‌ها را مبنی بر ریشه کلمات، بررسی کرده و توکن‌های با ریشه‌های یکسان را حذف می‌کنیم.
۸. به دلیل اینکه می‌خواهیم از الگوریتم Doc2Vec استفاده کنیم و این الگوریتم از Tf-idf برای بررسی اهمیت توکن‌ها استفاده می‌کند، تمامی توکن‌های پیش پردازش شده را مجددا به همدیگر متصل کرده تا بتوانیم شبه‌جمله هایی همانند جمله‌های اصلی داشته باشیم.
۹. توسط الگوریتم Doc2Vec از متون بردارهایی استخراج کرده تا ورودی مدل‌های یادگیری ماشین که در بخش‌های بعدی گفته می‌شود، فراهم شود.
۱۰. موضوع سوالات توسط الگوریتم LDA استخراج شده تا ویژگی بیشتری برای مدل‌ها فراهم شود و سپس به وسیله فاصله کسینوسی، شباهت موضوع‌های سوالات با یکدیگر محاسبه می‌شود.
۱۱. شباهت معنایی جفت سوالات نیز توسط الگوریتم Relevance Similarity محاسبه شده و ویژگی بیشتری نیز فراهم می‌کنیم.
۱۲. سوالات پیش‌پردازش شده را به همراه برچسب حقیقی آن‌ها در فایل‌های جداگانه‌ای ذخیره کرده تا برای آموزش مدل یادگیری ماشین استفاده شوند.

۳-۲- آموزش و آزمایش مدل یادگیری ماشین

در این بخش داده‌های پیش‌پردازش‌شده‌ای که در بخش قبل فراهم آورده شدند که شامل جفت سوالات تکراری و غیرتکراری با برچسب‌های درست هستند، به عنوان ورودی مدل در نظر گرفته می‌شوند. از سه مدل رگراسیون لجیستیک[۱۰]، Adaboost و جنگل تصادفی[۱۱] به دلیل اینکه در پژوهش‌های پیشین مانند [۱] و [۲] و در کل در صنعت و پژوهش‌های حوزه‌های دیگر، این سه الگوریتم به دلیل قدرت و دقت بالا مورد استقبال و استفاده قرار گرفته‌اند، استفاده می‌کنیم و سپس نتایج بدست آمده توسط آن‌ها را نسبت به معیارهای Accuracy و Precision و Recall و F1 مقایسه می‌کنیم. مشاهده می‌شود که هر سه مدل Accuracy حدود ۹۰ درصد را نتیجه می‌دهند که از بین آن‌ها، مدل جنگل تصادفی بهترین عملکرد یعنی حدود ۹۱ درصد Accuracy را بدست می‌اورد. دلیل برتر بودن الگوریتم جنگل تصادفی و Adaboost، این است که این روش‌ها در حقیقت روش‌های Ensemble از درخت‌ها هستند. البته قابل به ذکر است که جنگل تصادفی، ترکیب درخت‌های تصمیم و الگوریتم Adaboost ترکیب کنده‌های تصمیم[۱۲] هستند و از این جهت با یکدیگر تفاوت دارند. می‌دانیم درخت‌های تصمیم و مشتقات آن مانند کنده‌های تصمیم، یکی از معروف ترین الگوریتم‌های کلاسیک برای دسته‌بندی محسوب می‌شوند لذا با ترکیب آن‌ها قدرت و دقت مدل بالا می‌رود و به همین دلیل در صنعت هم به وفور از این الگوریتم‌ها استفاده می‌شود. تفاوت‌های دیگر این دو روش، نوع نمونه‌برداری آن‌ها است که در جنگل تصادفی، از روش Bagging و در Adaboost از روش Boosting استفاده می‌شود. از تفاوت‌های دیگر آن‌ها بحث وزن یال‌های درخت است که در جنگل تصادفی، از وزن‌های یکسان اما در Adaboost از وزن‌های متغیر استفاده می‌شود. تفاوت دیگر نیز در وابستگی

درخت‌ها یا کنده‌ها با یکدیگر است که در جنگل تصادفی درخت‌ها مستقل از یکدیگر و بدون ترتیب ظاهر می‌شوند اما در Adaboost کنده‌ها ترتیب دارند و مستقل نیستند.

۴- نتیجه‌گیری

در این مقاله ما یک روش تلفیقی از پژوهش‌های پیشین برای شناسایی و استخراج سوالات تکراری مربوط به مخزن Stack Overflow ارائه دادیم. در این روش پیش‌پردازش و استخراج ویژگی چندمرحله‌ای انجام شد و سپس سه مدل یادگیری ماشین منتخب توسط این داده‌ها و ویژگی‌ها آموزش داده شد که بتوانند سوالات تکراری را در مرحله آزمایش، تشخیص دهند. این روش‌ها در صورتی که داده‌های آموزشی به مقدار کافی موجود نباشد، ممکن است اعبارشان را از دست بدهند (تهدید علیه اعتبار) زیرا مدل به خوبی fit نمی‌شود. از طرف دیگر اگر مجموعه‌داده آموزشی متوازن نباشد، یعنی تعداد سوالات تکراری و غیرتکراری برابر نباشند، مدل به سمت سوالاتی که تعداد بیشتری دارند bias می‌شود و اعتبار این روش از بین می‌رود.

مراجع

[1] Wei Emma Zhang, Quan Z. Sheng, Jey Han Lau, Ermyas Abebe (2017).*Detecting Duplicate Posts in Programming QA Communities via Latent Semantics and Association Rules*

[2] Wei Emma Zhang, Quan Z. Sheng, Jey Han Lau, Ermyas Abebe, WENJIE RUAN (2018).*Duplicate Detection in Programming Question Answering Communities*

[3] Dasha Bogdanova, Cícero dos Santos, Luciano Barbosa and Bianca Zadrozny (2015).*Detecting Semantically Equivalent Questions in Online User Forums*

[4] M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy, and K. A. Schneider (2016)*Mining duplicate questions in Stack Overflow*

[5] Rodrigo F. G. Silva Klérisson Paixão Marcelo de A. Maia (2018).*Duplicate Question Detection in Stack Overflow: A Reproducibility Study*

[6] Doris Hoogeveen, Andrew Bennett, Yitong Li, Karin M. Verspoor, Timothy Baldwin (2018).*Detecting Misflagged Duplicate Questions in Community Question-Answering Archives*

[7] Yun Zhang, David Lo, Xin Xia ,Jian-Ling Sun(2016). *Multi-Factor Duplicate Question Detection in Stack Overflow*

[8] Liting Wang, LiZhang, JingJiang (2019).*Detecting Duplicate Questions in Stack Overflow via Deep Learning Approaches*

[9] Durham Abric, Oliver E. Clark, Matthew Caminiti, Keheliya Gallaba, and Shane McIntosh (2019).*Can Duplicate Questions on Stack Overflow Benefit the Software Development Community?*

[10] Jin Gao, Yahao He, Xiaoyan Zhang, Yamei Xia, (2017). *Duplicate short text Detection Based on Word2vec*

پانویس‌ها

۱ Vector similarity
۲ Topic similarity
۳ Entity Overlap
۴ WordNet Similarity
۵ CNN
۶ RNN
۷ LSTM
۸ Lower case
۹ Stop Word
۱۰ Logistic Regression
۱۱ Random Forest
۱۲ Decision Stump