وزارت علوم، تحقیقات، و فناوری
پژوهشگاه علوم و فناوری اطلاعات ایران
(ایرانداک)

طرح پژوهشی

طراحی و ساخت یک ماژول نرم‌افزاری برای استخراج و بازنمایی خودکار پیشنهادهای پژوهش از پایان‌نامه‌ها و رساله‌های پایگاه اطلاعات علمی ایران (گنج)

مجری
نصراله پاک‌نیت

علی‌اصغر حجت‌پناه

آبان‌ماه ۱۴۰۱


بسم الله الرحمن الرحیم

طراحی و ساخت یک ماژول نرم‌افزاری برای استخراج و بازنمایی خودکار پیشنهادهای پژوهش از پایان‌نامه‌ها و رساله‌های پایگاه اطلاعات علمی ایران (گنج)

مدیر طرح پژوهشی: نصراله پاک‌نیت، پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)

نشانی: تهران، خیابان انقلاب، چهارراه فلسطین، شماره ۱۰۹۰، طبقه ۴، اتاق ۴۰۳

صندوق پستی: ۱۳۱۵۷۷۳۳۱۴ **تلفن:** ۶۶۴۹۴۹۸۰-۰۲۱(داخلی ۴۲۲)

وبگاه: https://www.irandoc.ac.ir

رایانامه: *pakniat@irandoc.ac.ir*

همکاران: علی‌اصغر حجت‌پناه

به نام خدا

تولید دانش‌بنیان، اشتغال آفرین

گواهی

انجام طرح پژوهشی:

- **عنوان:** طراحی و ساخت یک ماژول نرم‌افزاری برای استخراج و بازنمایی خودکار پیشنهادهای پژوهش از پایان‌نامه‌ها و رساله‌های پایگاه اطلاعات علمی ایران (گنج)
- **پدیدآوران:** دکتر نصراله پاک‌نیت، مهندس علی اصغر حجت‌پناه
- **شماره (و تاریخ) قرارداد:** ۳۳/۳۹۱ (۱۴۰۰/۱/۲۱)
- **تاریخ شروع:** ۱۴۰۰/۱/۲۱
- **تاریخ پایان** (تصویب گزارش پایانی در شورای پژوهش): ۱۴۰۱/۸/۲۹ (نشست ۴۷۷)
- **ناظران:** دکتر امیرحسین صدیقی (ناظر علمی)، دکتر سیروس علیدوستی (ناظر بهره‌بردار)، مهندس عطیه صیادی (ناظر فنی)

در پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک) گواهی می‌شود. از سوی ایرانداک، از پدیدآوران و ناظران گرامیِ این طرحِ ارزشمند سپاسگزارم.

تندرست باشید
لیلا نامداریان
سرپرست معاونت پژوهش و آموزش

چکیده

انتخاب مساله پژوهش مناسب همواره یکی از دغدغه‌های اساسی پژوهشگران بوده است. یکی از مهم‌ترین ملاک‌های موجود در این راستا، به‌روز بودن یا به عبارتی عامیانه‌تر «جای کار داشتن» مساله انتخابی است. از آن‌جا که پژوهشگران «مسائل نیازمند پژوهش بیشتر» و بعضا ایده‌های قابل‌استفاده برای حل آن‌ها را در قالب «کارهای آتی» یا «پیشنهادهای پژوهش» در اسناد علمی بیان می‌کنند، مطالعه این اسناد راه‌کار اصلی در نظر گرفته شده برای شناسایی این مسائل است. با این حال، با توجه به رشد روزافزون مدارک علمی، مطالعه تمام اسناد موجود و استخراج پیشنهادهای پژوهش امری دشوار بوده و روزبه‌روز نیز بر این دشواری افزوده می‌شود. با توجه به این دشواری، به‌ویژه برای پژوهشگران تازه‌کار، در این پژوهش به مساله استخراج خودکار پیشنهادهای پژوهش از پایان‌نامه‌ها و رساله‌ها (پارساها)ی موجود در پایگاه اطلاعات علمی ایران (گنج) پرداخته شده است. در این راستا، در ابتدا روش‌های موجود برای این مساله مورد بررسی قرار گرفته و در ادامه، با به‌کارگیری رویکرد یادگیری ماشینی با نظارت، روشی خودکار برای استخراج پیشنهادهای پژوهش از پارساهای فارسی‌زبان ارائه شده است. برای به‌کارگیری رویکرد با نظارت در روش پیشنهادی، به مجموعه داده‌ای برچسب‌خورده نیاز است که با توجه به عدم وجود چنین مجموعه داده‌ای برای زبان فارسی، به عنوان یکی از دست‌آوردهای این پژوهش، چنین مجموعه داده‌ای ایجاد شده و برای آموزش و آزمایش روش پیشنهادی به کار گرفته شده است. برای پیاده‌سازی روش پیشنهادی از ابزارهای یادگیری ماشینی مختلفی مانند بیز ساده، رگرسیون لجستیک و ماشین بردار پشتیبان استفاده شده است. نتایج آزمایشات صورت گرفته بیانگر این است که به‌کارگیری دسته‌بندهای رگرسیون لجستیک و ماشین بردار پشتیبان نسبت به استفاده از دسته‌بند دیگر منجر به نتایج بهتری شده است. در پایان، روش پیشنهادی در قالب یک ماژول نرم‌افزاری پیاده‌سازی و تحت عنوان سامانه «گنجینه پیشنهادهای پژوهش پایان‌نامه‌ها و رساله‌ها» به کار گرفته شده است. گنجینه پیشنهادهای پژوهش از داده‌های پایگاه «اطلاعات علمی ایران» (گنج) استفاده کرده و پیشنهادهای پژوهش استخراج شده را در اختیار کاربران قرار می‌دهد.

کلیدواژه‌ها: متن‌کاوی؛ کارهای آتی؛ استخراج خودکار؛ پیشنهادهای پژوهش؛ پایگاه اطلاعات علمی ایران.

فهرست مطالب

عنوان **شماره صفحه**

فهرست شکل‌ها

عنوان **شماره صفحه**

فهرست جدول‌ها

عنوان **شماره صفحه**

سپاس‌گزاری

اکنون که به یاری خدا طرح "طراحی و ساخت یک ماژول نرم‌افزاری برای استخراج و بازنمایی خودکار پیشنهادهای پژوهش از پایان‌نامه‌ها و رساله‌های پایگاه اطلاعات علمی ایران (گنج)" به پایان رسیده است، بر خود لازم می‌دانم

از جناب آقای دکتر امیرحسین صدیقی و سرکار خانم مهندس عطیه صیادی که زحمت نظارت علمی و فنی این طرح را بر عهده داشته‌اند،

از جناب آقای دکتر سیروس علیدوستی، ریاست گرامی پیشین پژوهشگاه که علاوه بر پیشنهاد موضوع پژوهش، به عنوان ناظر بهره‌بردار نیز از راهنمایی ایشان بهره برده‌ام،

و همچنین اعضای گرامی شورای پژوهش به‌ویژه جناب آقای دکتر پرویز شهریاری معاون گرامی سابق پژوهش و آموزش پژوهشگاه، که با پیشنهادهای سازنده خود، منجر به افزایش کیفیت این پژوهش شده‌اند،

کمال تشکر را داشته باشم.

فصل اول

معرفی پژوهش

۱-۱. مقدمه و مساله پژوهش

بخش پیشنهادهای پژوهش یکی از بخش‌های مهم در اسناد و مدارک علمی است. این بخش ارزش زیادی برای پژوهشگران داشته و سرنخ‌هایی در رابطه با موضوعات پژوهشی و ایده‌های مناسب برای حل آن‌ها در اختیار پژوهشگران قرار می‌دهد. پژوهشگران به طور معمول از اطلاعات ارائه شده در این بخش برای انتخاب مساله پژوهش مناسب استفاده می‌کنند. با توجه به زمان انتشار اسناد علمی، می‌توان با بررسی بخش پیشنهادهای پژوهش، مساله‌های پژوهشی به‌روز یا به عبارتی عامیانه‌تر، مساله‌های نیازمند کار بیشتر، را شناسایی کرد و در راستای حل آن‌ها تلاش نمود. با این حال، با توجه به رشد روزافزون مدارک علمی، مطالعه اسناد موجود و استخراج پیشنهادهای پژوهش برای پژوهشگران امری دشوار بوده و روزبه‌روز نیز بر دشواری آن افزوده می‌شود. با توجه به شناخت کمتر، پژوهشگران تازه‌کار نیز دشواری بیشتری را در این راستا متحمل می‌شوند. برای مثال، یک دانشجوی کارشناسی ارشد را در نظر بگیرید که هنوز مفهوم پژوهش را به درستی درک نکرده اما از او خواسته می‌شود که پیشنهاده پایان‌نامه خود را ارائه کند. این دانشجو باید چه مساله‌ای را برای پژوهش خود انتخاب کند؟ به دلایل مختلف، اغلب اساتید راهنما و مشاور نیز در این زمینه کمک چندانی نمی‌کنند. به وضوح، به‌روز بودن مساله پژوهش و جای کار داشتن آن، مساله‌ای اساسی برای دانشجویان است؛ چرا که نمره پایان‌نامه آن‌ها و همچنین ادامه تحصیل آن‌ها در مقطع بالاتر در ارتباط مستقیم با خروجی پژوهش آن‌ها در قالب مقاله خواهد بود. با توجه به سختی استخراج دستی پیشنهادهای پژوهش از اسناد علمی (به‌ویژه برای پژوهشگران تازه‌کار)، در این پژوهش مساله استخراج خودکار این اقلام اطلاعاتی از پایان‌نامه‌ها و رساله‌ها (پارساها)ی موجود در پایگاه اطلاعات علمی ایران (گنج) پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)، در نظر گرفته شده و با بررسی روش‌های ارائه

شده بدین منظور در زبان انگلیسی و همچنین ویژگی‌های زبان فارسی، یک روش خودکار بدین منظور ارائه شده است.

لازم به ذکر است که استخراج پیشنهادهای پژوهش از تمام‌متن پارساها با چالش‌های بیشتری نسبت به استخراج این فراداده‌ها از مقالات مواجه است. این مهم به این دلیل است که اولا حجم مقالات نسبت به پارساها کمتر بوده و متن کمتری نیازمند پردازش است؛ دوما، با توجه به استانداردهای نشریات و نظارت موجود بر روی آن‌ها، شناسایی بخشی از مقاله که پیشنهادهای پژوهش در آن ذکر شده‌اند نسبت به شناسایی بخش مشابه در پارساها راحت‌تر است؛ و سوما، بخش زیادی از پیشنهادهای ارائه شده در پارساها اجرایی بوده و پژوهشی نیستند.

۲-۱. هدف و گام‌های پژوهش

هدف اصلی این پژوهش عبارت است از «طراحی و پیاده‌سازی یک ماژول نرم‌افزاری برای استخراج خودکار پیشنهادهای پژوهش از پارساهای ثبت شده در ایرانداک» که این هدف را می‌توان به اهداف جزئی‌تر زیر تقسیم نمود:

- شناسایی روش‌های به‌کار رفته برای استخراج خودکار پیشنهادهای پژوهش از اسناد علمی در زبان انگلیسی.
- طراحی و پیاده‌سازی آزمایشی روش پیشنهادی و ارزیابی کیفیت نتایج به‌کارگیری آن با توجه به پارامترهای دقت، صحت و فراخوانی.
- پیاده‌سازی نهایی روش پیشنهادی تحت عنوان یک ماژول نرم‌افزاری در پلت‌فرم موردنظر مرکز فناوری اطلاعات پژوهشگاه و به‌کارگیری آن.

با توجه به اهداف تعریف شده و نیازمندی‌ها، این پژوهش در سه گام انجام شده است.

- **گام نخست: بازطراحی و پیاده‌سازی مکانیزم انتقال داده‌ها به سامانه گنج.** انتقال داده‌ها از سامانه‌های ثبت و ویرایش به پایگاه گنج قبل از انجام پژوهش حاضر نیز صورت می‌گرفته است. با این حال، ابزار سابق مورد استفاده بدین منظور فاقد کارایی لازم بوده و با توجه به سرعت بسیار پایین (در حد ۲۰۰ مدرک در روز)، برای ارسال حجم‌های بالا از اطلاعات نامناسب است. با توجه به این مهم، این ابزار فاقد توانایی لازم برای انتقال اطلاعات استخراج شده و منتج از خروجی طرح حاضر به پایگاه گنج است. برای حل این مساله، در گام نخست طرح حاضر و به عنوان بخشی از دست‌آوردهای آن، ابزار فعلی به

گونه‌ای بازطراحی شده تا علاوه بر بهبود عملکرد و کارایی آن، قابلیت ارسال اطلاعات استخراج شده در این طرح به پایگاه گنج نیز وجود داشته باشد.

- **گام دوم: بررسی مساله، طراحی و پیاده‌سازی آزمایشی روش پیشنهادی، و ارزیابی آن**. در این گام، با بررسی روش‌های ارائه شده برای استخراج پیشنهادهای پژوهش از مدارک علمی در زبان انگلیسی و همچنین ویژگی‌های زبان فارسی، یک روش خودکار برای استخراج پیشنهادهای پژوهش از اسناد علمی فارسی‌زبان ارائه شده است. روش پیشنهادی با دریافت تمام‌متن یک پارسا در قالب فایل Word یا Tex، جملات آن را به دو دسته ۱) دربردارنده پیشنهاد پژوهش و ۲) فاقد پیشنهاد پژوهش، دسته‌بندی می‌کند. در ادامه و پس از پیاده‌سازی آزمایشی روش پیشنهادی، از پارامترهای دقت، صحت و فراخوانی برای بررسی کیفیت آن استفاده شده است.
- **گام سوم: پیاده‌سازی و ارائه ماژول نرم‌افزاری**. در این گام، پیاده‌سازی نهایی روش پیشنهادی در قالب یک ماژول نرم‌افزاری و با توجه به نیازهای فنی موجود صورت گرفته است. ماژول نرم‌افزاری طراحی شده تمام‌متن پارساها را به عنوان ورودی دریافت کرده، پیشنهادهای پژوهش را از آن‌ها استخراج نموده و پیشنهادهای استخراج شده را به عنوان فراداده‌ای جدید به مدارک اضافه می‌کند. برای دسترسی کاربران به پیشنهادهای پژوهش استخراج شده، ماژول نرم‌افزاری جدید در قالب یک سامانه جدید به نام «گنجینه پیشنهادهای پژوهش پایان‌نامه‌ها و رساله‌ها» ارائه شده و از طریق آدرس https://ganjpro.irandoc.ac.ir/ قابل دسترس است. در این سامانه، قلم اطلاعاتی «پیشنهادهای پژوهش» به عنوان یک فراداده همانند آن‌چه در مورد چکیده، کلیدواژه‌ها، فهرست نوشته‌ها و فهرست‌های منابع در سامانه گنج داشتیم، به صورت یک Tab جدید به هر پارسا اضافه شده است.

۳-۱. ساختار پژوهش

ادامه این طرح پژوهشی به شرح زیر است. در فصل دوم جزئیات بازطراحی و پیاده‌سازی مکانیزم انتقال داده‌ها به سامانه گنج ارائه شده است. در فصل سوم، به پیشینه مساله اصلی این پژوهش پرداخته شده است. در فصل چهارم، مفاهیم پیش‌نیاز لازم برای درک ادامه این طرح پژوهشی شامل روش‌های بیز ساده، رگرسیون لجستیک و ماشین بردار پشتیبان مورد مطالعه قرار گرفته است. در ادامه، جزئیات روش پیشنهادی برای استخراج پیشنهادهای پژوهش از پارساهای موجود در پایگاه گنج و همچنین

ارزیابی آن در فصل پنجم بیان شده است. نتیجه‌گیری و پیشنهادهای پژوهش در فصل ششم ارائه شده است.

فصل دوم
بازطراحی و پیاده‌سازی مکانیزم انتقال داده‌ها به سامانه گنج

۲-۱. مقدمه

همانطور که در فصل قبل ذکر شد سازوکار فعلی انتقال داده‌ها از سامانه‌های ثبت و ویرایش به پایگاه گنج فاقد کارایی لازم بوده و با توجه به سرعت بسیار پایین (در حد ۲۰۰ مدرک در روز)، برای ارسال حجم‌های بالا از اطلاعات نامناسب است. با توجه به این مساله و نیاز این پژوهش به انتقال حجم بالایی از اطلاعات از سامانه‌های ثبت و ویرایش به پایگاه گنج، در گام نخست این طرح و به عنوان بخشی از دست‌آوردهای آن، ابزار فعلی به‌گونه‌ای بازطراحی می‌شود تا علاوه بر بهبود عملکرد و کارایی آن (با تاکید بر انتقال حداقل ۲۰۰۰ مدرک در روز)، امکان ارسال اطلاعات استخراج شده در این طرح به پایگاه گنج، و همچنین اجرای فرایند انتقال به صورت زمان‌بندی شده و غیربرخط نیز میسر شود.

لازم به ذکر است که گزینه‌های دیگری مانند توسعه یک ماژول جدید برای انتقال داده‌ها نیز وجود داشت که با توجه به افزایش هزینه‌ها در صورت استفاده از چندین راه‌کار به صورت هم‌زمان، به‌کارگیری یک راه‌کار کمکی مقرون‌به‌صرفه و قابل قبول نبود. علاوه‌براین، به دلیل مشکلات موجود، که منجر به کاهش شدید کارایی راه‌کار فعلی شده، توسعه آن نیز میسر نبود. با توجه به این موارد، بهترین راه‌کار در دسترس بازطراحی و پیاده‌سازی راه‌کار حاضر بوده است.

در ادامه این فصل، در ابتدا راه‌کارهای مختلف انتقال اطلاعات بین پایگاه‌های داده SQL Server و mySQL را بررسی کرده، سپس راه‌کار انتخابی و مشخصات فنی پیاده‌سازی آن بیان شده و در نهایت، مقایسه‌ای بین ماژول انتقال داده سابق و جدیدا توسعه یافته ارائه می‌شود.

۲-۲. راه‌کارهای مختلف انتقال اطلاعات بین پایگاه‌های داده SQL Server و mySQL

برای انتقال اطلاعات بین پایگاه‌های داده SQL Server و mySQL سه معماری مختلف وجود دارد:

- **Web Service**: این معماری، که راه‌کار فعلی انتقال داده‌ها نیز بر اساس آن است، از مشکلاتی مانند سرعت کم، هزینه نگهداری بالا و ... رنج می‌برد. همچنین، تعریف اقلام اطلاعاتی انتقالی جدید در این راه‌کار نیازمند بازبینی و تغییر دو برنامه از آن (برنامه ارسال داده که با C# ایجاد شده و برنامه ثبت اطلاعات که با روبی توسعه داده شده) است.

- **ETL/ELT**: معروف‌ترین ابزارهای موجود در این دسته SSIS و CData Data Sync هستند. ماهیت این معماری برای جریان انتقال و روان داده است. با این حال، وجود توابع اختصاصی برای دست‌کاری داده‌ها، استفاده از منابع بیرونی مانند اصطلاح‌نامه‌ها و پایگاه‌های مرجع دیگر، و همچنین استفاده از روش‌های رمزگذاری داده، منجر به نامناسب شدن این معماری برای به‌کارگیری در کاربرد حاضر می‌شود. نکته دیگری که در استفاده از ابزارهای موجود در این دسته باید مورد توجه گیرد نیاز به حضور یک متخصص باتجربه در این حوزه بوده که با توجه به شرایط، در حال حاضر امکان آن فراهم نیست.
- **Data Connectorها**: در این معماری، ارتباط مستقیم دو پایگاه داده از طریق Data Connectorها ایجاد می‌شود. از رایج‌ترین Data Connectorهای موجود می‌توان به ODBC اشاره کرد. در این رویکرد، کنترل همه اجزاء در اختیار برنامه‌نویس بوده، در استفاده از منابع بیرونی انعطاف بالایی وجود داشته، انواع دست‌کاری داده ممکن بوده، انتقال با سرعت بالا صورت گرفته و مدیریت خطاها به راحتی قابل انجام است.

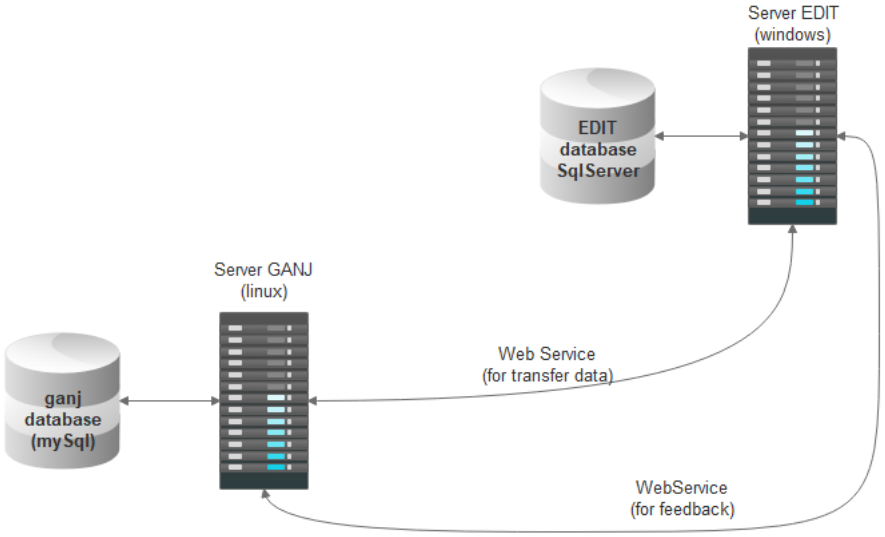
با توجه به مزایا و معایب ذکر شده برای روش‌های موردنظر، در این پژوهش از رویکرد سوم، یعنی استفاده از Data Connectorها، برای بازطراحی مکانیزم موجود برای انتقال داده‌ها استفاده شده است. جزئیات فنی ماژول جدید توسعه یافته در جدول ۲-۱ ارائه شده است.

جدول ۲-۱. مشخصات فنی ماژول توسعه‌یافته.

عنوان ابزار	مشخصات	شرح
ارسال اطلاعات از سامانه ثبت به پایگاه گنج	.net Core 3.1 console application SQLServer to MySQL Installed on SABT Server	انتقال مدارک جدید ثبت شده به پایگاه گنج
ارسال اطلاعات از سامانه ویرایش به پایگاه گنج	.net Core 3.1 console application SQLServer to MySQL Installed on EDIT Server	انتقال مدارک ویرایش شده به پایگاه گنج انتقال فرادادههای حاصل از استخراج پیشنهادهای پژوهش به گنج
نمایه اطلاعات دریافت شده	Ruby Solr	بعد از تزریق اطلاعات در پایگاه گنج توسط ابزارهای بالا، توسط این برنامه اطلاعات نمایه شده و قابلیت جستجو توسط Solr را خواهند داشت.

۲-۳. مقایسه ماژول‌های انتقال داده قبلی و جدید

در مقایسه با ماژول قبل (که معماری آن در شکل ۲-۱ نشان داده شده)، ماژول جدید، که معماری آن در شکل ۲-۲ نشان داده شده، دارای مزایایی مانند کارایی، سرعت انتقال و انعطاف‌پذیری بالاتر، نگهداری و مدیریت خطای آسان‌تر و تنوع بیشتر در داده‌های قابل انتقال است. جزئیات بیشتر در مورد مقایسه ماژول جدید و قبل، در جدول ۲-۲ ارائه شده است. لازم به ذکر است که موارد مشخص شده با ستاره در این جدول، شرایط تعیین شده توسط مرکز فناوری اطلاعات پژوهشگاه برای بهبود بوده است.



شکل ۲-۱. ماژول انتقال داده قبل و نحوه استفاده آن از معماری وب‌سرویس برای انتقال داده.

شکل ۲-۲. ماژول توسعه سافته با استفاده از معماری Data Connectorها.

جدول ۲-۲. مقایسه ماژول‌های انتقال داده‌های قبل و جدیدا توسعه یافته.

معیار مقایسه	روش قبل	روش جدید
معماری	وب‌سرویس	ارتباط مستقیم با هر دو پایگاه داده از طریق Data connector

کارایی	**پایین**: وب‌سرویس‌ها در انتقال داده بین دو پایگاه داده که برقراری ارتباط مستقیم بین آن‌ها و یا دسترسی به آن‌ها محدود شده است کارایی دارند.	**بالا**: در شرایط فراهم بودن امکان برقراری ارتباط مستقیم، این روش سریع‌ترین راه‌کار انتقال داده است.
سرعت انتقال*	به دلیل سربارگذاری و عبور داده در بسترهای مختلف، جزء روش‌های کند محسوب می‌شود. سرعت انتقال داده از پایگاه ویرایش به گنج در بهترین حالت ۴۰۰ مدرک در روز است. این کندی فقط مربوط به معماری نیست زیرا تراکنش‌های زیادی به‌ازای درج یک مدرک باید انجام شود که بخشی از این کندی به دلیل استفاده از فریم‌ورک ActiveRecord است. همچنین بخشی از منابع سرور برای سرویس دادن به جستجو و خدمات پایگاه گنج اختصاص یافته و بخش‌های دیگر را با کندی مواجه می‌کند.	سرعت این روش در هر روز بیش از ۱۰ هزار مدرک است. اطلاعات ورودی توسط دانشجویان ۳۰۰ مدرک در روز و اطلاعات ویرایش شده توسط بخش نمایه‌سازی ۱۰۰ مدرک در روز است. تعداد رکوردهای باقیمانده در صف نیز زیاد است: - ۳۰ هزار پیشنهاده - ۵ هزار مدرک ویرایش شده - ۸ هزار ثبت جدید - ۴۰ هزار مدرک دانشگاه تهران ماژول طراحی شده روی ماشین دیگری اجرا می‌شود و از منابع سرور گنج استفاده نمی‌کند.
انعطاف‌پذیری*	هر گونه تغییر در تعداد اقلام اطلاعاتی انتقالی باید در هر دو برنامه ارسال داده و دریافت داده اعمال شود که با توجه به محدودیت توسعه در سمت روبی (کمبود نیروی متخصص) اینکار با تاخیر زیاد همراه است.	فقط یک برنامه نیاز به ویرایش دارد.
نگهداری*	دو برنامه در دو زیرساخت متفاوت، یکی در ویندوز و یکی در لینوکس باید نگهداری شود: - وب‌سرویس ارسال اطلاعات به گنج با #C در ویندوز. - وب‌سرویس دریافت اطلاعات و ذخیره اطلاعات در گنج با روبی و تحت لینوکس.	فقط یک برنامه ویندوزی باید نگهداری شود: - برنامه تزریق اطلاعات در پایگاه گنج
فریم‌ورک‌ها	در #C با Entity Framework و در روبی با ActiveRecord	در ارتباط با پایگاه داده ویرایش از Entity Framework استفاده شده و برای درج اطلاعات در پایگاه گنج از AdoMySQL به عنوان Data Connector استفاده شده است.
مدیریت خطا	از مکانیسم TXT Log استفاده شده است. سامانه ارسال اطلاعات از بروز خطا در سامانه دریافت مطلع نمی‌شود و موارد مشکل‌دار را در	Logها در پایگاه داده و با ساختار مشخصی ذخیره شده و قابلیت اجرای پرسمان بر روی آن‌ها وجود دارد. همچنین از طریق به‌روزرسانی Flagها،

	فراخوانی‌های بعدی نیز مجدد ارسال می‌کند.	امکان تلاش مجدد روی موارد مشکل‌دار و یا پرش از روی این موارد وجود دارد.
داده‌های قابل ارسال	فقط پایان‌نامه/رساله	پایان‌نامه/رساله، پیشنهاده و در گام دوم اقلام اطلاعاتی مربوط به طرح پژوهشی حاضر

٤-٢. نتیجه‌گیری

همان‌طور که ذکر شد ماژول سابق انتقال داده مابین سامانه‌های ثبت، ویرایش و گنج از کارایی پایینی برخوردار بود به‌گونه‌ای که حتی بدون توجه به طرح حاضر نیز، محدودیت‌های زیادی را ایجاد کرده بود. با توجه به این مساله، در این گام از این پژوهش، به بازطراحی و پیاده‌سازی این ماژول پرداخته شده تا محدودیت‌های موجود برطرف گردد و امکان به‌کارگیری نتایج طرح حاضر، که در گام‌های بعد حاصل خواهند شد، نیز ایجاد شود. لازم به ذکر است که اجرای موفقیت‌آمیز ماژول توسعه‌یافته منجر به انتقال ۱۵۰۰۰ مدرک به پایگاه گنج و خالی شدن صف‌های تشکیل شده برای ارسال اطلاعات به این پایگاه شده است.

فصل سوم

پیشینه پژوهش

در این فصل به مرور کارهای صورت گرفته در زمینه استخراج پیشنهادهای پژوهش و موارد مشابه، شامل استخراج تقدیر و تشکرها و پیشینه پژوهش از مدارک علمی، پرداخته خواهد شد.

۱-۳. استخراج پیشنهادهای پژوهش

مساله استخراج پیشنهادهای پژوهش را می‌توان به عنوان حالتی خاص از مساله استخراج خودکار فراداده از اسناد در نظر گرفت. پژوهش‌های محدودی در زمینه استخراج پیشنهادهای پژوهش از اسناد علمی صورت گرفته است. عبارات منظم[۱] و الگوریتم‌های یادگیری ماشینی ابزارهایی هستند که در این پژوهش‌ها مورد استفاده قرار گرفته‌اند. در ادامه این بخش، به بررسی روش‌های ارائه شده در این زمینه خواهیم پرداخت.

در مطالعه (Hu and Wan 2015)، از «عبارات منظم» برای استخراج پیشنهادهای پژوهش استفاده شده است. در این راستا، نویسندگان دو دسته متفاوت از عبارات منظم تعریف نموده‌اند. دسته اول سخت‌گیرانه‌تر و موکدتر بوده، جملاتی که منطبق با عبارات منظم موجود در این دسته باشند با احتمالی بسیار بالا دربردارنده پیشنهاد برای پژوهش‌های آتی بوده و به طور مستقیم در مجموعه جملات نهایی در نظر گرفته می‌شوند. پس از یافتن جمله‌ای منطبق با عبارات منظم تعریف شده در دسته اول، جملاتی که منطبق با عبارات منظم موجود در دسته دوم هستند، نیز می‌توانند به عنوان جملات دربردارنده پیشنهاد برای پژوهش‌های آتی در نظر گرفته شوند. دلیل در نظر گرفتن دو دسته عبارات منظم در این مقاله این است که جملاتی که منطبق با عبارات منظم دسته دوم هستند، می‌توانند جملات مربوط به نتیجه‌گیری نیز باشند. این مهم به این دلیل است که نویسندگان به طور معمول ابتدا نتیجه‌گیری کار خود را ارائه کرده و در ادامه پیشنهادها برای پژوهش‌های آتی را در بخش

[۱] regular expression

نتیجه‌گیری ارائه می‌کنند. بنابراین، در صورت یافتن یک جمله دربردارنده پیشنهاد پژوهش، فرض بر این است که نویسندگان شروع به توصیف پیشنهادهای پژوهش نموده و جملات بعد از این جمله در بخش مربوطه نیز، حتی اگر در دسته دوم عبارات منظم صدق کنند، مربوط به پیشنهادهای پژوهش هستند.

جهت بررسی کیفیت روش پیشنهادی، نویسندگان با برچسب‌گذاری جملات ۴۰۰ مقاله به صورت دستی، مجموعه داده‌ای شامل ۷۲۴ جمله دربردارنده پیشنهاد پژوهش تهیه کرده‌اند. مقالات موردنظر از «مجموعه ACL»[۱] انتخاب شده و شامل مقالات منتشر شده توسط ACL و سازمان‌های مرتبط و نشریه «زبان‌شناسی محاسباتی»[۲] در ۵ دهه است. مقالات ورودی در قالب PDF بوده و نویسندگان از «PDFlib»[۳] برای استخراج متن از آن‌ها و از «ParsCit»[۴] برای استخراج ساختار فیزیکی بخش‌ها، زیربخش‌ها و پاراگراف‌ها استفاده کرده‌اند. در صورتی که مقاله شامل بخش کارهای آتی باشد، این بخش و در غیر این صورت، بخش نتیجه‌گیری در ساخت پیکره مورد استفاده قرار گرفته است. نتایج آزمایشات صورت گرفته توسط نویسندگان بیانگر مقادیر ۸۵/۶، ۹۴/۷ و ۸۹/۹ به ترتیب برای پارامترهای دقت[۵]، فراخوانی[۶] و F1 روش پیشنهادی بوده است.

در (Li and Yan 2015)، یک خط سیر[۷] پردازش زبان طبیعی برای شناسایی پیشنهادهای پژوهش در مقالات علمی و استخراج کلیدواژه از آن‌ها ارائه کرده‌اند. خط سیر ارائه شده از گام‌های زیر تشکیل شده است:

۱. شناسایی بخش پیشنهادهای پژوهش: (Li and Yan 2015) برای افزایش صحت[۸] الگوریتم ارائه شده، تجزیه و تحلیل‌های خود را تنها به بخش‌هایی از مقالات که با بیشترین احتمال دربردارنده پیشنهادهای پژوهش هستند، محدود کرده‌اند. بدین منظور، نویسندگان تنها بخش‌های با عنوان Discussion یا Conclusion و یا بخش‌هایی که کلمه Future در عنوان آن‌ها وجود دارد را بدین منظور مورد تجزیه و تحلیل قرار داده‌اند.

---

[۱] https://aclanthology.org/

[۲] Computational Linguistics Journal

[۳] https://www.pdflib.com/

[۴] https://wing.comp.nus.edu.sg/projects/parscit-an-open-source-crf-reference-string-and-logical-document-structure-parsing-package/

[۵] precision

[۶] recall

[۷] pipeline

[۸] accuracy

۲. شناسایی جملات درباره پیشنهادهای پژوهش: در این قسمت، نویسندگان با استفاده از الگوریتمی برای بررسی تطابق کلیدواژه‌ها، الگوهای زبانی اطراف کلیدواژه‌ها و ویژگی‌های نحوی به شناسایی جملات درباره پیشنهادهای پژوهش پرداخته‌اند. در این راستا، از ۱۰۰ مقاله انتخاب شده به صورت تصادفی از نشریه Science Advances استفاده شده و مرتبط‌ترین الگوهای نحوی و اصطلاح‌شناختی[۱] مورد استفاده در گذاره‌های درباره پیشنهادهای پژوهش شناسایی شده است. نویسندگان بیان کرده‌اند که ۱۱ واژه future، further، additional، will، would، need، needs، needed، remain، warrant و warrants به عنوان «اصطلاحات هسته»[۲] انتخاب شده و جملات دربردارنده این واژگان برای ارزیابی‌های بیشتر وارد خط سیر شده‌اند. در ادامه، برای هر اصطلاح هدف در جملات انتخاب شده قواعدی جهت شناسایی دیگر اصطلاحات مهم در کنار آن ایجاد شده است.
۳. استخراج کلیدواژه: در این گام، نویسندگان برای هر مقاله دو دسته کلیدواژه، یکی از پیشنهادهای پژوهش مقاله و دیگری از عنوان و چکیده آن ایجاد کرده‌اند.
۴. تجزیه و تحلیل داده‌ها: در این گام، نویسندگان کلیدواژه‌ها را با توجه به منبع، دسته‌بندی کرده و به مقایسه کلیدواژه‌های استخراج شده از پیشنهادهای پژوهش و عنوان و چکیده پرداخته‌اند و در صورت وجود حداقل یک کلیدواژه مشترک مابین دو دسته استخراج شده برای یک مقاله، آن مقاله را به عنوان یک مقاله دارای ارتباط منطقی[۳] در نظر گرفته‌اند.

برای ارزیابی خط سیر پیشنهادی، نویسندگان آزمایش‌هایی نیز بر روی مجموعه‌ای شامل ۱۶۲۰ مقاله تمام‌متن از نشریه Science advances انجام داده‌اند. نتایج بررسی‌های صورت گرفته بیانگر این است که از بین ۱۶۲۰ مقاله بررسی شده، تنها ۵۳۷ مقاله دارای پیشنهاد پژوهش بوده و از این بین نیز تنها ۲۱۸ مقاله ارتباط منطقی داشته‌اند. در زمینه استخراج پیشنهادهای پژوهش، نتایج آزمایش‌های صورت گرفته نشانگر مقدار ۹۶/۴٪ برای پارامتر دقت و ۷۶/۸٪ برای پارامتر فراخوانی روش پیشنهادی بوده است.

در (Hao et al. 2020)، در ابتدا با توجه به تمام متن مقالات دانشگاهی، مجموعه داده‌ای از پیشنهادهای پژوهش ایجاد شده است. در ادامه، نویسندگان به بررسی پیشنهادهای پژوهش ارائه شده

---

[۱] terminological

[۲] seed term

[۳] conceptual connection

در مجموعه داده تهیه شده پرداخته و آن‌ها را به صورت دستی در شش دسته روش[۱]، منابع[۲]، ارزیابی[۳]، کاربرد[۴]، مساله[۵] و سایر[۶]، و ۱۷ زیردسته دسته‌بندی کرده‌اند. نتایج بررسی‌های صورت گرفته بیانگر این است که بیشترین جملات درباره پیشنهادهای پژوهش از دسته روش بوده و تعداد جملات موجود در دیگر دسته‌ها تقریبا با یکدیگر برابر بوده است.

(Zhu et al. 2019) به بررسی مقالات نشریه Journal of the Association for Information Science and Technology با توجه به پیشنهادهای پژوهش موجود در آن‌ها پرداخته‌اند. در این راستا، نویسندگان مدل یادگیری عمیق Bert را روی مقالات شامل پیشنهادهای پژوهش آموزش داده‌اند. نویسندگان بیان کرده‌اند که یکی از مزایای استفاده از مدل Bert این است که این مدل، قبل از آغاز آموزش برای مساله استخراج پیشنهادهای پژوهش، روی «بازنمایی زبانی»[۷] آموزش دیده است. برای ارزیابی مدل آموزش دیده، نویسندگان آن را بر روی مجموعه داده‌ای برچسب‌گذاری نشده مورد آزمایش قرار داده‌اند. نتیجه به دست آمده بیانگر مقدار ۸۲/۶۵٪ برای پارامتر میانگین هارمونیک بیشینه[۸] بوده است. Zhu و همکاران در کار خود، علاوه بر به‌کارگیری مدل‌های یادگیری عمیق برای استخراج پیشنهادهای پژوهش، دسته‌بندی جدیدی شامل چهار دسته حمایت‌گر[۹]، مربوط به روش[۱۰]، برجسته‌سازی عوامل تاثیر بالقوه[۱۱] و نمایشگر اهداف آتی[۱۲] برای پیشنهادهای پژوهش ارائه کرده‌اند.

(Chen and Fang 2019) به مساله جدید بودن ایده‌های به کار رفته در مقالات پرداخته و روشی خودکار برای استخراج ایده‌های جدید از چکیده مقالات در حوزه‌های مهندسی و فناوری ارائه

---

۱ method

۲ resources

۳ evaluation

۴ application

۵ problem

۶ other

۷ linguistic representation

۸ maximum harmonic mean

۹ supportive

۱۰ methodological

۱۱ potential influence factor

۱۲ future target

کرده‌اند. در روش ارائه شده، نویسندگان n-تایی‌ها[۱] را با استفاده از برچسب‌گذاری اجزای واژگانی کلام[۲] (POS) استخراج کرده و با بررسی پایگاه داده اسکوپوس جدید بودن آن‌ها را بررسی می‌کند.

(Hao et al. 2021)، از روش‌های یادگیری ماشینی بیز ساده[۳]، رگرسیون لجستیک[۴]، ماشین بردار پشتیبان[۵]، و جنگل تصادفی[۶] برای استخراج پیشنهادهای پژوهش از مقالات علمی استفاده کرده‌اند. برای آموزش و آزمایش روش، نویسندگان از مجموعه داده ارائه شده در (Hao et al. 2020) استفاده کرده‌اند. نتایج به دست آمده بیانگر مقادیر ۹۲/۴٪، ۹۲/۳۹٪، ۹۲/۵۸٪ و ۹۱/۲۲٪ برای پارامتر دقت و ۸۱٪/۰۵، ۸۶٪/۸۷، ۹۲٪/۶۳ و ۹۷/۵۸٪ برای پارامتر فراخوانی، به ترتیب برای روش‌های رگرسیون لجستیک، ماشین بردار پشتیبان، جنگل تصادفی و بیز ساده بوده است. همچنین نویسندگان، دقت و فراخوانی روش‌های پیشنهادی را بر روی انواع جملات درباره پیشنهادهای پژوهش در نظر گرفته شده در (Hao et al. 2020)، یعنی روش، منابع، ارزیابی، کاربرد، مساله و سایر نیز بررسی کرده‌اند که نتایج حاصله بیانگر این است که جملات درباره پیشنهادهای پژوهش از نوع روش به طور دقیق‌تر قابل شناسایی هستند.

در (Qian et al. 2021)، نویسندگان از جملات درباره پیشنهادهای پژوهش برای بررسی روند پژوهش در موضوعات پژوهشی متفاوت در حوزه پردازش زبان طبیعی استفاده کرده‌اند. در این راستا، نویسندگان مجموعه‌ای از مقالات ارائه شده در همایش‌ها در زمینه پردازش زبان طبیعی را در نظر گرفته، پیشنهادهای پژوهش آن‌ها را استخراج کرده و در دسته‌های روش، منابع، ارزیابی، کاربرد، مساله و سایر دسته‌بندی نموده‌اند. نویسندگان همچنین ۲۹ موضوع پژوهشی، شامل ترجمه ماشین، استخراج اطلاعات و ...، را به عنوان موضوع مقاله‌های موردنظر در نظر گرفته‌اند. نتایج حاصله بیانگر این است که پیشنهادهای پژوهش ارائه شده در مقالات با موضوعات پژوهشی مختلف، در دسته‌های متفاوتی قرار دارد. پیشنهادهای پژوهش ارائه شده در مقالات با موضوعات پژوهشی نوظهور بیشتر در دسته منابع قرار داشته درحالیکه موضوعات پژوهشی بالغ و قدیمی‌تر بیشتر شامل پیشنهادهای پژوهش از نوع روش و کاربرد هستند.

---

[۱] n-grams

[۲] part of speech tagging

[۳] naïve bayes

[۴] logistic regression

[۵] support vector machine

[۶] random forest

در (Wainer and Valle 2013) به این مساله پرداخته شده که پس از چاپ، چه اتفاقی برای پیشنهادی پژوهش مندرج در مقاله‌های چاپ شده در زمینه علوم کامپیوتر رخ می‌دهد. در این راستا، نویسندگان با در نظر گرفتن ۱۰۰ مقاله چاپ شده در نشریات و ۱۰۰ مقاله ارائه شده در همایش‌ها در زمینه علوم کامپیوتر، به بررسی این موضوع پرداخته‌اند که چند درصد از پیشنهادهای پژوهش ارائه شده در سال‌های بعد انجام شده‌اند. نکته قابل توجه اینکه نتایج آن‌ها نشان داده که ۷۰٪ مقالات هیچ‌گاه ادامه نیافته‌اند. با توجه به نتایج این پژوهش، می‌توان نتیجه گرفت که پیشنهادهای پژوهش مطرح شده در انتهای مقالات همواره ارزشمند نبوده و در بسیاری از موارد انجام نمی‌شوند.

۲-۳. استخراج تقدیر و تشکرها

در (Cronin et al. 1993) و (Cronin et al. 2003)، به ترتیب مجموعه مقالات علمی چاپ شده در مهم‌ترین نشریات جامعه‌شناسی در یک بازه زمان ۱۰ ساله و مهم‌ترین نشریات روانشناسی و فلسفه در یک بازه زمانی ۱۰۰ ساله، مورد بررسی قرار گرفته و «تقدیر و تشکر»[۱]های موجود در آن‌ها به صورت دستی استخراج شده است. با توجه به نتایج؛ نویسندگان به این نتیجه رسیده‌اند که تقدیر و تشکرها در یکی از ۶ دسته زیر قرار دارند: ۱) پشتیبانی معنوی[۲]، پشتیبانی مادی[۳]، پشتیبانی ویراستاری[۴]، پشتیبانی ارائه[۵]، پشتیبانی ابزاری/فنی[۶]، و پشتیبانی مفهومی[۷] یا ارتباط تعاملی همتایان[۸].

در (Councill et al. 2005) و (Giles et al. 2004)، مساله استخراج خودکار تقدیر و تشکرها از اسناد علمی به عنوان حالت خاصی از مساله استخراج خودکار فراداده‌ها یا استخراج اطلاعات از اسناد علمی در نظر گرفته شده و نویسندگان از عبارات منظم و روش‌های یادگیری ماشینی بدین منظور استفاده کرده‌اند. نتایج بررسی‌ها نشان‌دهنده این است که با استفاده از عبارات منظم می‌توان به طور قابل قبول به شناسایی و استخراج خودکار تقدیر و تشکرها از اسناد علمی پرداخت.

در (Khabsa et al. 2012)، سیستمی با نام Ackseer برای استخراج خودکار تقدیر و تشکرها از اسناد علمی و جستجو در آن‌ها تشکیل شده است. سیستم Ackseer تماما خودکار بوده و اسناد علمی

---

۱ acknowledgement

۲ moral support

۳ finantial support

۴ editorial support

۵ presentational support

۶ instrumenal/technical support

۷ conceptual support

۸ peer interactive communication

موجود در کتابخانه‌های دیجیتال (مانند مقالات نشریات، مقالات همایش‌ها و کتاب‌ها) را بررسی کرده، بخش‌های تقدیر و تشکر را از آن‌ها استخراج کرده و موجودیت‌های مورد تقدیر واقع شده در آن‌ها را شناسایی می‌کند.

۳-۳. استخراج پیشینه پژوهش[۱]

پیشینه پژوهش بخشی از یک مقاله علمی است که کارهای مرتبط دیگر نویسندگان را معرفی کرده و مقایسه‌ای با کار حاضر نویسندگان مقاله فعلی نیز انجام می‌دهد. تولید خودکار بخش کارهای مرتبط برای یک مقاله جدید، ابزاری را در اختیار محققان قرار می‌دهد تا بدون از دست دادن کارهای مرتبط بتوانند به طور کارا این بخش از مقاله را ایجاد کنند. هدف مساله ایجاد خودکار پیشینه پژوهش عبارت است از ایجاد بخش مرور ادبیات یا پیشینه تحقیق با توجه به سایر بخش‌های مقاله و مجموعه‌ای از منابع مورد استناد قرار گرفته در آن.

در (Li and Ouyang 2022)، نویسندگان از جهاتی مانند گونه‌های متفاوت در نظر گرفته شده برای مساله، جمع‌آوری مجموعه داده، روش به کار رفته برای حل مساله و ارزیابی کارایی به طور جامع به بررسی مساله ایجاد خودکار بخش پیشینه پژوهش پرداخته و مسائل و حوزه‌های نیازمند مطالعه بیشتر در این زمینه را مشخص نموده‌اند.

در (Chen and Zhuge 2017)، نویسندگان رویکردی جهت ایجاد خودکار بخش کارهای مرتبط برای یک مقاله جدید ارائه کرده‌اند. رویکرد ارائه شده ابتدا مقالاتی که به مراجع مقاله جدید ارجاع داده‌اند، را جمع‌آوری کرده و در ادامه، جملاتی که در آن‌ها به مرجع موردنظر اشاره شده را استخراج می‌کند. رویکرد ارائه شده سپس کلیدواژه‌های مقالات جمع‌آوری شده و مقاله در حال نوشته شدن را استخراج کرده و گرافی از کلیدواژه‌ها ایجاد می‌کند. در ادامه درخت اشتاینر مینیمم[۲] ایجاد می‌شود که کلیدواژه‌های متمایزکننده و کلیدواژه‌های عنوان را پوشش می‌دهد. در نهایت خلاصه موردنظر با استخراج جملات پوشش‌دهنده درخت اشتاینر ایجاد می‌شود.

(Hoang and Khan 2010) یک سیستم خلاصه‌ساز بخش کارهای مرتبط ارائه کرده است. این سیستم، با دریافت چندین مقاله به عنوان ورودی، خلاصه‌ای از کارهای مرتبط مربوط به مقاله هدف را با توجه به موضوع مقاله ایجاد می‌کند. در روش پیشنهادی توسط هوانگ[۳] و خان[۴] مجموعه‌ای از

---

[۱] related works

[۲] minimum Steiner tree

[۳] Hoang

[۴] Khan

کلیدواژه‌ها که به صورت سلسله‌مراتبی مرتب شده‌اند و توصیف‌گر موضوعات مقاله هدف هستند، به عنوان ورودی دریافت شده و در ادامه، با توجه به مجموعه‌ای از قواعد، بخش کارهای مرتبط مقاله هدف ایجاد می‌شود.

در (Hu and Wan 2014) با استفاده از یادگیری با نظارت و یک چارچوب بهینه‌سازی، یک سیستم دیگر برای ایجاد بخش کارهای مرتبط یک مقاله هدف با توجه به منابع ورودی ایجاد شده است.

٤-٣. نتیجه‌گیری

با توجه به بررسی‌های صورت گرفته، پژوهش‌های محدودی در زمینه استخراج پیشنهادهای پژوهش از اسناد مدارک علمی صورت گرفته است. در این پژوهش‌ها، از ابزارهای عبارات منظم و الگوریتم‌های یادگیری ماشینی[1] استفاده شده که با توجه به نتایج، به نظر می‌رسد استفاده از الگوریتم‌های یادگیری ماشینی منجر به نتایج بهتری شده است. با توجه به این مهم، در این پژوهش نیز از رویکرد یادگیری ماشینی برای استخراج پیشنهادهای پژوهش استفاده خواهیم کرد.

[1] machine learning

فصل چهارم
پیش‌نیازهای نظری

در این فصل، پیش‌نیازهای نظری لازم برای درک ادامه این پژوهش ارائه خواهد شد. در این راستا، در ابتدا به بررسی روش‌های دسته‌بندی و معیارهای ارزیابی آن‌ها خواهیم پرداخت. سپس، روش‌های دسته‌بندی ماشین بردار پشتیبان (SVM)، رگرسیون لجستیک و بیز ساده که ابزارهای مورد استفاده در این پژوهش هستند، را مورد بررسی قرار خواهیم داد. لازم به ذکر است که مطالب ارائه شده در بخش‌های ۴-۱ و ۴-۲ برگرفته از (نصیری ۱۳۹۰)، (کارگر ۱۳۹۰)، (نصیری ۱۳۹۶) و (پاک‌نیت و همکاران ۱۳۹۷)، مطالب ارائه شده در بخش ۴-۳ برگرفته از (اصلان‌پور، ۱۳۹۱) و مطالب ارائه شده در بخش ۴-۴ برگرفته از (میرزایی ۱۳۹۵) هستند.

٤-١. روش‌های دسته‌بندی و معیارهای ارزیابی آن‌ها

دسته‌بندی به فرایند نسبت دادن یک شی یا نمونه ناشناخته به یک رده یا دسته بر اساس ویژگی‌های آن گفته می‌شود. برای این منظور دسته‌بندی‌کننده یک مدل از داده‌های آموزشی در مرحله آموزش ایجاد کرده و با استفاده از این مدل داده‌های جدید را در مرحله آزمون مورد آزمایش قرار می‌دهد (پاک‌نیت و نصیری، ۱۳۹۹).

در اغلب مسائل دسته‌بندی، برای ارزیابی روش‌ها از پارامترهای دقت، صحت، فراخوانی و F1 استفاده می‌شود که در ادامه آن‌ها را تعریف خواهیم کرد.

پارامتر دقت: پارامتر دقت برای هر دسته به صورت

$$\frac{tp}{tp + fp} \qquad (۴-۱)$$

محاسبه می‌شود که tp و fp به ترتیب برابر با تعداد مثبت صادق[۱] و مثبت کاذب[۲] هستند.

پارامتر فراخوانی: پارامتر فراخوانی برای هر دسته به صورت

$$\frac{tp}{tp + fn} \qquad (۴-۲)$$

محاسبه می‌شود که tp همانند بالا و fn برابر با تعداد منفی کاذب[۱] است.

---

[۱] true positive

[۲] false Positive

پارامتر F1: پارامتر F1 برابر با میانگین دو پارامتر دقت و فراخوانی در نظر گرفته می‌شود.

پارامتر صحت برای هر دسته به صورت

$$\frac{\mathrm{tp} + tn}{\mathrm{tp} + \mathrm{fp} + \mathrm{fn} + \mathrm{tn}} \qquad (۴-۳)$$

محاسبه می‌شود که tp ، fp و fn همانند بالا و tn برابر با تعداد منفی صادق[۲] است.

۴-۲. دسته‌بند ماشین بردار پشتیبان (SVM)

روش ماشین بردار پشتیبان (SVM) از کارآمدترین روش‌های دسته‌بندیست که در سال ۱۹۹۵ معرفی شده است (Vapnik 1995). در این روش، مسأله همواره به صورت یک مساله برنامه‌ریزی ریاضی مدل‌سازی شده و با حل این مدل، جواب تعیین می‌گردد. هدف نهایی در این روش یافتن یک یا چند ابرصفحه برای جداسازی داده‌ها از یکدیگر است. لازم به ذکر است که در برخی موارد ممکن است جداسازی داده‌ها با ابرصفحه‌ها از یکدیگر میسر نباشد و استفاده از ابرصفحه‌ها برای جداسازی منجر به خطا در دسته‌بندی شود. برای حل این معضل می‌توان از منحنی‌ها و یا توابع غیرخطی برای جداسازی استفاده کرد. پس از تعیین ابرصفحه‌های موردنیاز برای جداسازی دسته‌ها، از آن‌ها به عنوان معیاری برای تعیین وضعیت داده‌هایی که وضعیت آن‌ها نامشخص است استفاده خواهد شد. در این حالت، هر ابرصفحه‌ای برای دسته‌بندی مناسب نبوده و در صورتی که دقت حاصل از مقداری قابل قبول بیشتر باشد از آن ابرصفحه برای جداسازی استفاده می‌شود. در غیر این صورت باید برای یافتن یک ابرصفحه دیگر با دقت بالاتر تلاش شود.

۴-۲-۱. روش SVM کلاسیک برای داده‌های دو دسته‌ای جدایی‌پذیر خطی

ساده‌ترین نوع روش SVM برای دسته‌بندی داده‌های دو دسته‌ای جدایی‌پذیر خطی روش کلاسیک SVM است که مبنای بسیاری از سایر انواع روش‌های SVM نیز به شمار می‌رود (Vapnik 1995، Vapnik 1998). فرض کنید تعداد M داده عددی به صورت بردارهای ستونی m بعدی x_1 ، ...، x_M متعلق به دسته‌های ۱ یا ۲ در اختیار باشند. متناظر با هر x_i می‌توان یک اسکالر y_i در نظر گرفت به طوری که اگر x_i متعلق به دسته ۱ باشد y_i مقدار ۱ و چنانچه متعلق به دسته ۲ باشد مقدار ۱- دریافت کند. با توجه به این دسته‌بندی برای مجموعه داده‌ها، می‌توان نمایشی به صورت $s = \{(x_i, y_i)\}_{i=1}^{M}$ برای آن‌ها در نظر گرفت. در ادامه، در روش SVM هدف این است که مجموعه داده‌های $s =$

---

[۱] false negetive

[۲] true negetive

$\{(x_i, y_i)\}_{i=1}^{M}$ به وسیله یک ابرصفحه خطی $\bar{w}^T x + \bar{b} = 0$ که در آن $\bar{w}^T \in R^M$ و $\bar{b} \in R$ است به نحوی از یکدیگر جدا شوند که رابطه زیر برقرار باشد:

$$\begin{cases} \bar{w}x_i + \bar{b} > 0 & y_i = 1 \\ \bar{w}x_i + \bar{b} < 0 & y_i = -1 \end{cases} \quad i = 1,2,\dots,M. \qquad (۴-۴)$$

رابطه فوق را می‌توان به صورت زیر بازنویسی کرد:

$$\exists \varepsilon > 0 \text{ s.t.}: \begin{cases} \bar{w}x_i + \bar{b} \geq \varepsilon & y_i = 1 \\ \bar{w}x_i + \bar{b} \leq \varepsilon & y_i = -1 \end{cases} \quad i = 1,2,\dots,M \qquad (۴-۵)$$

با ضرب طرفین نامعادلات موجود در رابطه فوق در عدد $\frac{1}{\varepsilon} > 0$ و جایگذاری $w = \frac{1}{\varepsilon}\bar{w}$ و $b = \frac{1}{\varepsilon}\bar{b}$ خواهیم داشت:

$$\begin{cases} w^T x_i + b \geq 1 & y_i = 1 \\ w^T x_i + b \leq -1 & y_i = -1 \end{cases} \quad i = 1,2,\dots,M. \qquad (۴-۶)$$

یا به عبارت دیگر

$$y_i(w^T x_i + b) \geq 1 \qquad i = 1,2,\dots,M \qquad (۴-۷)$$

تعریف. مجموعه داده‌های $s = \{(x_i, y_i)\}_{i=1}^{M}$ را جداپذیر خطی[۱] گویند هرگاه $w^T \in R^m$ و $b \in R$ موجود باشند به‌طوریکه با استفاده از این پارامترها رابطه فوق برای مجموعه داده s برقرار باشد. در این صورت، ابرصفحه $w^T x + b = 0$، ابرصفحه جداکننده داده‌های s نامیده می‌شود. علاوه بر این، دو ابرصفحه $w^T x + b = 1$ و $w^T x + b = -1$ ابرصفحات حاشیه‌ای[۲]، ناحیه بین این دو ابر صفحه حاشیه سخت[۳]، فاصله بین این دو ابرصفحه پهنای حاشیه سخت و بردار x_i که در یکی از ابرصفحات حاشیه‌ای صدق کند بردار پشتیبان[۴] نامیده می‌شوند.

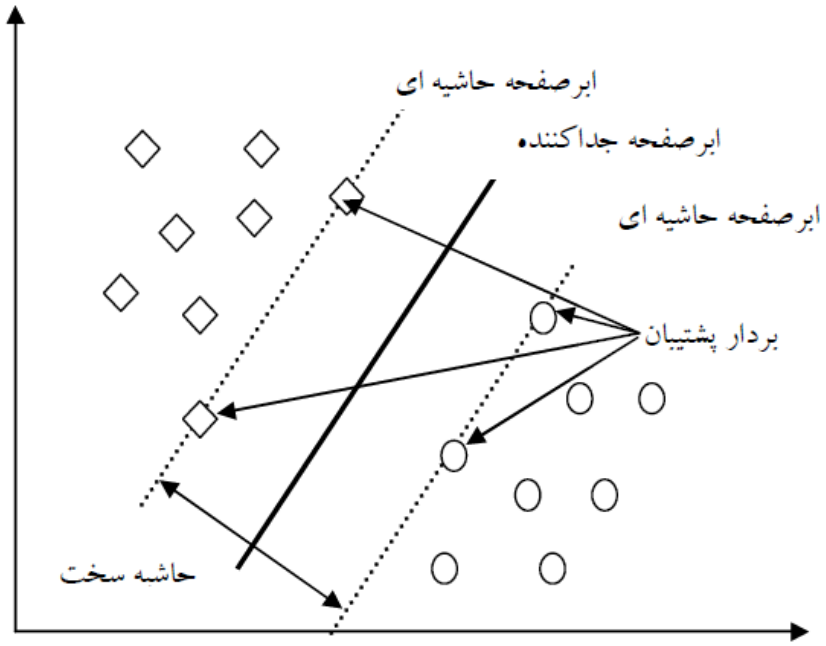
---

[۱] linearly seperable

[۲] marginal hyperplane

[۳] hard margin

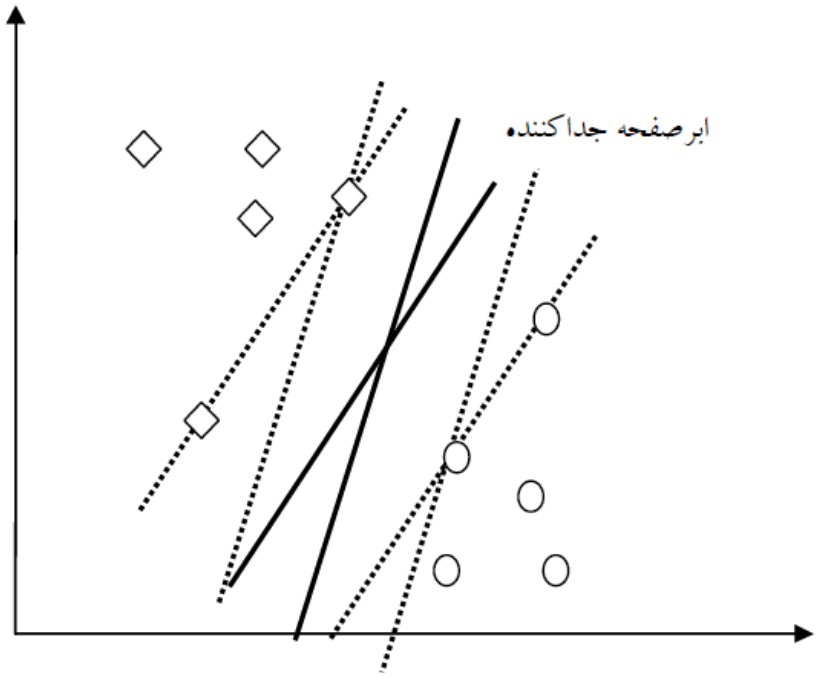
[۴] support vector



شکل ٤-١. نمایش ابرصفحات حاشیه‌ای و ابرصفحه جداکننده برای تعدادی داده (کارگر ۱۳۹۰)

در شکل ۴-۱ مثالی از ابرصفحات حاشیه‌ای، ابرصفحه جداکننده، حاشیه سخت و بردار پشتیبان برای مجموعه‌ای فرضی از داده‌ها در فضای دو بعدی نشان داده شده است.

همانطور که در شکل ۴-۲ نیز نشان داده شده است برای هر مجموعه داده جداپذیر خطی ابرصفحه‌های جداکننده زیادی می‌توان پیدا کرد. پهنای حاشیه سخت برخی از این ابرصفحه‌ها کوچک و برخی دیگر بزرگ‌تر است. در روش SVM ترجیح بر این است که ابرصفحه جداکننده به نحوی محاسبه شود که پهنای حاشیه سخت متناظر بیشترین مقدار ممکن را داشته باشد.



شکل ٤-۲. وجود ابرصفحات جداکننده متعدد برای یک مجموعه داده دو دسته‌ای (کارگر ۱۳۹۰)

با توجه به این‌که ابرصفحات حاشیه‌ای با یکدیگر موازی هستند، بیشترین پهنای حاشیه سخت مربوط به ابرصفحات حاشیه‌ایست که بیشترین فاصله را از یکدیگر داشته باشند. با در نظر گرفتن $\frac{2}{\|w\|}$ به عنوان فاصله بین دو ابرصفحه حاشیه‌ای، هدف در استفاده از SVM عبارت است از کمینه کردن $\frac{1}{2}\|w\|^2$ به‌طوریکه

$$y_i(w^T x_i + b) \geq 1 \qquad i = 1,2,\dots,M, w \in R^m, b \in R \qquad (۴-۸)$$

تعریف. ابرصفحه جداکننده با بیشترین حاشیه سخت مربوط به مجموعه داده‌های $s = \{(x_i, y_i)\}_{i=1}^{M}$ را ابرصفحه جداکننده بهینه[۱] یا به اختصار ابرصفحه بهینه نامند.

تعریف. تابع $D(x) = w^T x + b$ را تابع تصمیم[۲] مجموعه داده‌های $s = \{(x_i, y_i)\}_{i=1}^{M}$ گویند هرگاه $w^T x + b = 0$ ابرصفحه بهینه جداکننده این مجموعه باشند.

همانطور که قبلا نیز بیان شد، هدف در استفاده از SVM عبارت است از استفاده از داده‌هایی با دسته مشخص برای یافتن مدلی جهت تعیین دسته داده‌هایی که دسته آن‌ها مشخص نیست. تابع تصمیم ذکر شده در تعریف فوق همان مدل مورد بحث است. تشخیص دسته داده‌ها با استفاده از این تابع تصمیم به صورت زیر انجام می‌شود:

---

۱ optimal sperating hyperplane

۲ decision function

$$x \in \begin{cases} 1 \text{ دسته } D(x) > 0 \\ 2 \text{ دسته } D(x) < 0 \end{cases} \qquad (۴-۹)$$

اگر $D(x) = 0$، در این صورت x روی ابرصفحه جداکننده بهینه قرار داشته و مدل حاصل قادر به تشخیص دسته x نخواهد بود. در این حالت ممکن است بتوان با تغییر داده‌های اولیه ساخت مدل، ابرصفحه بهینه دیگری به دست آورد تا تعیین کننده وضعیت دسته x باشد.

روش کلاسیک SVM بررسی شده برای داده‌هایی مناسب است که جداپذیر خطی باشند. اما، در اکثر مسائل واقعی چنین چیزی برقرار نبوده و در نتیجه روش ارائه شده در بخش قبل برای این مسائل بدون استفاده خواهد بود. در ادامه، به بررسی راه‌حل روش SVM برای داده‌هایی که به صورت خطی جداپذیر نیستند، پرداخته خواهد شد.

داده‌هایی که جداپذیر خطی نیستند را می‌توان در یکی از دو حالت زیر در نظر گرفت:

- داده‌هایی که بتوان با قبول درجه‌ای از خطا آن‌ها را با استفاده از یک ابرصفحه جداسازی کرد. این بدین معنی است که ممکن است در این دسته‌بندی دسته اشتباه به تعداد اندکی از داده‌ها اختصاص یابد.
- داده‌هایی که در صورت جداسازی خطی آن‌ها، خطای زیادی حاصل شود به طوری که مدل به دست آمده از این جداسازی فاقد اعتبار باشد. در این حالت ممکن است بتوان جداسازی موردنظر را به صورت غیرخطی (یعنی با استفاده از توابعی غیرخطی) انجام داد. با توجه به عدم استفاده از SVM در این حالت در این پژوهش، جزئیات بیشتر در مورد این حالت بیان نشده و خوانندگان محترم برای اطلاع در مورد این حالت به منابع (Liu 1998، Masulli and Valentini 2000) ارجاع داده می‌شوند.

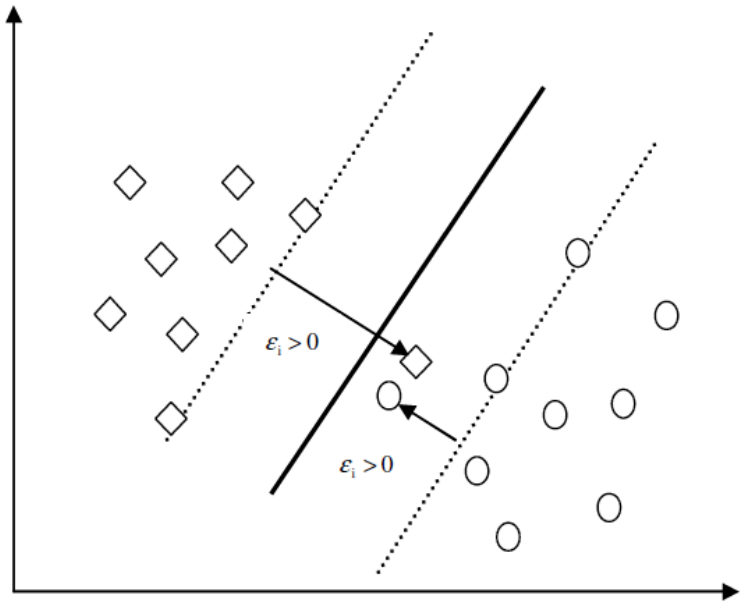
۴-۲-۲. روش SVM بر اساس نرم‌های $L1$ و $L2$

همانطور که بیان شد جداسازی مجموعه داده‌های دو دسته‌ای $s = \{(x_i, y_i)\}_{i=1}^{M}$ با استفاده از رابطه (۴-۴) اجازه هیچ گونه خطایی در دسته‌بندی نمی‌دهد. برای مجاز کردن خطا رابطه مورد نظر به صورت زیر تغییر داده می‌شود:

$$y_i(w^T x_i + b) \geq 1 - \varepsilon_i \qquad i = 1,2,\dots,M \qquad (۴-۱۰)$$

که ε_i یک متغیر کمبود نامنفی است. شکل ۴-۴ مفهوم ε_i را برای جداسازی داده‌هایی که به صورت خطی جداپذیر نیستند نمایش می‌دهد. در این صورت، اگر $0 < \varepsilon_i < 1$، همانطور که در شکل ۴-۳ نیز دیده می‌شود، داده x_i در درون ناحیه بین دو ابرصفحه حاشیه‌ای قرار گرفته است. اگر $\varepsilon_i \geq 1$، x_i به وسیله ابرصفحه بهینه به دست آمده اشتباها دسته‌بندی شده و در صورتی که

$\varepsilon_i = 0$، داده موردنظر به درستی دسته‌بندی شده است. در صورتی که داده اشتباه دسته‌بندی شده باشد مقدار ε_i برابر با فاصله x_i تا ابرصفحه حاشیه‌ای مربوط به دسته درست داده است.



شکل ٤-۳. یک دسته‌بندی خطی همراه با خطا (کارگر ۱۳۹۰)

حال فرض کنید مجموعه داده‌های دو دسته‌ای $s = \{(x_i, y_i)\}_{i=1}^{M}$ به کمک ابرصفحه جداکننده به نحوی دسته‌بندی شوند که برخی از آن‌ها در دسته اشتباه قرار گیرند. در این صورت، در این‌جا برخی از داده‌ها در ناحیه بین دو ابرصفحه حاشیه‌ای قرار خواهند گرفت و ناحیه بین دو ابرصفحه حاشیه‌ای را حاشیه نرم[۱] گویند. در شکل ۴-۳ مثالی از یک حاشیه نرم در دسته‌بندی مجموعه‌ای از داده‌های فرضی در یک فضای دو بعدی نشان داده شده است. در این نوع دسته‌بندی، اهداف زیر مدنظر است:

۱- حاشیه نرم متعلق به ابرصفحه جداکننده بهینه بیشترین پهنای ممکن را داشته باشد.

۲- تعداد داده‌های اشتباه دسته‌بندی شده به حداقل ممکن برسد.

هدف اول به دنبال بیشینه کردن فاصله بین دو ابرصفحه حاشیه‌ای است، در حالیکه هدف دوم به دنبال کمینه کردن تعداد داده‌های اشتباها دسته‌بندی شده است. با توجه به عدم وجود سنخیت بین این دو هدف، هدف دوم به عنوان کمینه کردن خطای دسته‌بندی در نظر گرفته می‌شود که خطای دسته‌بندی به صورت زیر تعریف می‌شود:

[۱] soft marigin

تعریف: فرض کنید دسته‌بندی مجموعه داده‌های دو دسته‌ای $s = \{(x_i, y_i)\}_{i=1}^{M}$ با استفاده از یک ابرصفحه جداکننده به نحوی انجام شود که رابطه (۴-۷) برقرار باشد. در این صورت، $\|\varepsilon\|_p^p$ به عنوان خطای دسته‌بندی تعریف شده که $p \in N$ و $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_M)$.

بسته به اینکه از چه نرمی برای خطای دسته‌بندی داده‌ها در تعریف فوق استفاده شود، حالت‌های مختلفی از روش SVM به دست می‌آید. دو نرم اصلی مورد استفاده عبارتند از $p = 1$ (استفاده از نرم L1) که در این صورت مدل به دست آمده، مدل SVM بر اساس L1 یا به اختصار L1-SVM نامیده شده و $p = 2$ (استفاده از نرم L2) که در این صورت مدل حاصل را مدل SVM بر اساس L2 یا به اختصار L2-SVM گویند.

۴-۳. رگرسیون لجستیک

رگرسیون لجستیک در ابتدا فقط در پژوهش‌های اپیدمیولوژی به‌کار گرفته می‌شد اما در ادامه به دیگر علوم از جمله اقتصاد، تجارت، زبان‌شناسی و ... نیز راه یافت. پژوهشگران در مقاله‌ها و کتب بسیاری به بررسی این ابزار پرداخته‌اند که از مهم‌ترین پژوهش‌های صورت گرفته در این زمینه می‌توان به (Cox and Snell 1989)، (Collett 1991)، (Kleinbaum 1994) و (Hosmer and Lemeshow 2000) اشاره کرد.

اغلب، در مباحث مربوط به رگرسیون بحث بر سر یافتن رابطه‌ای بین متغیر پاسخ y و مجموعه‌ای از متغیرهای پیش‌بینی‌کننده $x_1, x_2, \dots x_k$ است که می‌توان آن را به صورت $y = f(x_1, x_2, \dots x_k)$ بیان کرد. به عنوان نمونه می‌دانیم که در سوانح رانندگی، عواملی همچون زمان، وضعیت هوا، وضعیت راننده و استحکام خودرو بر متغیر پاسخ یعنی میزان خسارت اثرگذار هستند. چنین رابطه‌هایی را در ساده‌ترین حالت می‌توان به صورت یک رابطه خطی نوشت:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \qquad (۴-۱۱)$$

رابطه بالا را رگرسیون خطی[۱] می‌نامند. در این رابطه ضرایب β_0، β_1، ...، β_k به کمک نمونه‌های تصادفی و معمولا به روش کمترین مربعات[۲] خطا برآورد می‌شوند. برآورد فوق مستلزم فرضیات زیر است:

---

[۱] linear regression

[۲] least square

- خطی بودن الگو
- مستقل بودن مشاهدات
- نرمال بودن توزیع متغیر پاسخ
- ثابت بودن واریانس متغیر پاسخ

اما این مفروضات همیشه برقرار نبوده و در نتیجه الگوی خطی در بسیاری از مسائل کاربردی کارساز نخواهد بود. به عنوان مثال، در بسیاری از پژوهش‌ها لازم است تا از الگوهای غیرخطی استفاده کرد؛ یا در برخی موارد ممکن است متغیر پاسخ مورد مطالعه، یک متغیر گسسته باشد که در این حالت باید از رگرسیون‌های کیفی استفاده کرد. رگرسیون‌هایی با متغیرهای وابسته گسسته دارای انواع مختلفی هستند که با توجه به ماهیت متغیر وابسته تعیین می‌شوند.

در بسیاری از پژوهش‌ها متغیر پاسخ دو حالتی است. یعنی پاسخ‌ها تنها شامل دو حالت مانند وجود یا عدم وجود، خرید یا عدم خرید، بهبود یا عدم بهبود و ... هستند که آن‌ها را با ۰ یا ۱ نشان می‌دهیم. در این حالت اگر از روش‌های معمول رگرسیون خطی استفاده کنیم با مشکلاتی مانند موارد زیر روبرو خواهیم بود:

- ممکن است مقادیر پیش‌بینی‌شده فاقد معنا بوده یا در عمل قابل تفسیر نباشند.
- ممکن است امکان مقایسه مقادیر پیش‌بینی‌شده با یکدیگر وجود نداشته باشد.
- ممکن است فرض‌های اولیه رگرسیون برقرار نباشند.

در چنین شرایطی از رگرسیون لجستیک استفاده می‌شود.

۱-۳-٤. مدل‌بندی

متغیرهای پیش‌بینی‌کننده که می‌توانند بر مقدار متغیر پاسخ تأثیر بگذارند، ممکن است متغیرهایی کمی باشند. در چنین حالتی اگر به الگوی خطی ذکر شده توجه کنیم می‌بینیم که سمت راست تساوی از نظر تئوری می‌تواند مقادیری از منفی بی‌نهایت تا مثبت بی‌نهایت را اختیار کند در حالی که سمت چپ تساوی فقط مقادیر صفر یا یک را می‌گیرد. یک راه‌حل این مشکل این است که سمت چپ تساوی را به یک متغیر پیوسته با دامنه $(-\infty, +\infty)$ تبدیل کنیم. این کار را در سه مرحله انجام می‌دهیم:

۱. استفاده از احتمال y به جای آن. در این صورت اگر p احتمال $y = 1$ باشد، $1 - p$ احتمال $y = 0$ است، پس مقادیر سمت چپ تساوی بین صفر تا مثبت یک خواهد بود.

۲. استفاده از احتمال «نسبت بخت»[۱] به جای استفاده مستقیم به صورت

$$OR = \frac{p}{1-p} \qquad (۴-۱۲)$$

در این صورت مقدار حاصل از صفر تا مثبت بی‌نهایت تغییر خواهد کرد.

۳. گرفتن لگاریتم طبیعی از متغیر جدید یعنی نسبت بخت:

$$ln\frac{p}{1-p} \qquad (۴-۱۳)$$

که در این صورت، مقادیر آن مانند سمت راست تساوی بین منفی بی‌نهایت تا مثبت بی‌نهایت تغییر می‌کند. این عبارت را $logit(p)$ یعنی مقدار تابع لجیت در نقطه p می‌گویند.

با این تغییرات رابطه خطی قبلی به صورت زیر تغییر می‌کند که به الگوی لجیت معروف است:

$$ln\frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \qquad (۴-۱۴)$$

الگوهای رگرسیون لجستیک برای بیان پیش‌بینی متغیرهای دو حالتی یا در وضعیت‌هایی که هدف آن برآورد مفاهیمی چون رخ دادن یا رخ ندادن یک اتفاق است، الگوهای مناسبی هستند. استفاده از رگرسیون لجستیک به ویژه زمانی که با پایگاه داده‌های بسیار بزرگ مواجه هستیم یا در وضعیت‌هایی که متغیرهای مستقل، از یک قاعده و نظم کلی پیروی نمی‌کنند و مفروضات مدل‌های عمومی را نقض می‌کنند بسیار مفید است.

در رگرسیون لجستیک پیش‌شرط‌های انجام رگرسیون خطی موردنیاز نیست. همچنین برای برآورد پارامترهای β_0، β_1، ...، β_k به جای استفاده از روش حداقل مربعات مرسوم در رگرسیون خطی از روش حداکثر درست‌نمایی[۲] استفاده می‌شود.

در رگرسیون لجستیک:

۱. احتمال پیش‌بینی رخداد $y = 1$ را می‌توان بر اساس رابطه زیر بیان کرد:

$$\pi(\boldsymbol{X}) = E(y|\boldsymbol{X}) = p = P(y = 1) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}}{1 + e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}} \qquad (۴-۱۵)$$

۲. هر یک از ضرایب β_iها میزان تغییرات لگاریتم «نسبت بخت» را در متغیر پاسخ به ازای افزایش یک واحد در عامل i–ام نشان می‌دهند (به شرط ثابت بودن سایر متغیرها).

---

[۱] odd ratio

[۲] maximum likelihood

۳. از روی مقادیر β_iها می‌توان «نسبت بخت» مربوط به عامل i–ام را به صورت e^{β_i} به دست آورد.
۴. با استفاده از مقادیر برآورد شده پارامترهای β_0، β_1، ...، β_k و داشتن یک نمونه تصادفی از متغیرهای پیش‌بینی‌کننده و قرار دادن آن‌ها در رابطه (۴–۱۱) مقدار پیش‌بینی شده برای احتمال وقوع متغیر پاسخ به دست می‌آید (Hosmer and Lemeshow 2000).

۲–۳–٤. برآورد پارامترها

در رگرسیون لجستیک به جای استفاده از روش حداقل مربعات که در رگرسیون خطی معمولی مرسوم است، برای برآورد β_iها از روش ماکزیمم درست‌نمایی استفاده می‌شود.

۳–۳–٤. رگرسیون لجستیک ساده

برای به دست آوردن برآوردها ابتدا رگرسیون ساده با یک متغیر مستقل را در نظر گرفته و تعریف می‌کنیم:

$$g(x) = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_0 + \beta_1 x \qquad (۴–۱۶)$$

$$\pi(x) = p = P(y=1|x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \qquad (۴–۱۷)$$

$$1-\pi(x) = 1-p = P(y=0|x) \qquad (۴–۱۸)$$

حال تابع درست‌نمایی برای جفت مشاهده (y_i, x_i) به صورت زیر است:

$$l(\beta_0, \beta_1) = \pi(x_i)^{y_i}[1-\pi(x_i)]^{(1-y_i)} \qquad (۴–۱۹)$$

چون فرض می‌کنیم مشاهدات مستقل هستند، تابع درست‌نمایی برای کل نمونه به صورت حاصل‌ضرب تک تک عبارات خواهد بود:

$$l(\beta_0, \beta_1) = \prod_{i=1}^{n} \pi(x_i)^{y_i}[1-\pi(x_i)]^{(1-y_i)} \qquad (۴–۲۰)$$

اکنون باید β_iهایی را به دست آوریم که تابع درست‌نمایی را بیشینه سازند، در این راستا و برای سادگی، ابتدا از عبارت فوق لگاریتم گرفته می‌شود:

$$L(\beta_0,\beta_1)=\ln[l(\beta_0,\beta_1)]=\sum_{i=1}^{n}\{y_i\ln[\pi(x_i)]+(1-y_i)\ln[1-\pi(x_i)]\} \quad (۴-۲۱)$$

برای یافتن مقادیر β_iهایی که $L(\beta_0,\beta_1)$ را بیشینه می‌کنند باید از عبارت فوق نسبت به β_iها مشتق گرفته و نتیجه را برابر صفر قرار دهیم. در نتیجه این کار معادلات موسوم به معادلات درست‌نمایی به شکل زیر حاصل می‌شوند:

$$\sum_{i=1}^{n}[y_i-\pi(x_i)]=0 \quad (۴-۲۲)$$

$$\sum_{i=1}^{n}x_i[y_i-\pi(x_i)]=0 \quad (۴-۲۳)$$

حل این معادلات در رگرسیون خطی به راحتی امکان پذیر است اما در رگرسیون لجستیک این تابع یک تابع خطی از β_iها نیست و برای به دست آوردن جواب باید از روش‌های مخصوصی استفاده کرد. این روش‌ها بر پایه تکرارهای متوالی استوار هستند و برنامه‌های کامپیوتری و نرم‌افزارها با سرعت خوبی قادر به انجام آن‌ها هستند.

در کل با استفاده از معادلات درست‌نمایی می‌دانیم که

$$\sum_{i=1}^{n}y_i=\sum_{i=1}^{n}\hat{\pi}(x_i) \quad (۴-۲۴)$$

یعنی مجموع مقادیر مشاهده شده برابر با مجموع مقادیر برآورد شده است، و $\hat{\pi}(x_i)$ برآورد ماکزیمم درست‌نمایی $\pi(x_i)$ است.

بعد از به دست آوردن مقادیر برآورد شده برای پارامترها و قرار دادن آن‌ها در رابطه (۴-۱۴) می‌توان $\pi(\hat{x}_i)$، احتمال اینکه مقدار متغیر پاسخ برابر یک باشد، را محاسبه کرد.

٤-۳-٤. رگرسیون لجستیک چندگانه

فرض کنید برای پیش‌بینی متغیر وابسته از k متغیر مستقل استفاده شود که مقیاس عددی یا فاصله‌ای دارند. متغیرها را با $\boldsymbol{X}=(X_1,\dots,X_k)$ نشان می‌دهیم، پس باید با استفاده از نمونه تصادفی $(\boldsymbol{X}_i,y_i),i=1,2,\dots n$ ، پارامترهای $\beta=(\beta_0,\dots,\beta_k)$ را برآورد نماییم.

$$g(\boldsymbol{X})=\ln\left[\frac{\pi(\boldsymbol{X})}{1-\pi(\boldsymbol{X})}\right]=\beta_0+\beta_1x_1+\cdots+\beta_kx_k \quad (۴-۲۵)$$

$$\pi(\boldsymbol{X}) = p = P(Y = 1|\boldsymbol{X}) = \frac{e^{g(\boldsymbol{X})}}{1 + e^{g(\boldsymbol{X})}} \quad (۴-۲۶)$$

روش برآورد همان روش ماکزیمم درست‌نمایی است. تابع ماکزیمم درست‌نمایی را تشکیل می‌دهیم و از آن لگاریتم می‌گیریم.

$$L(\beta_0, \dots, \beta_k) = \sum_{i=1}^{n} \{y_i \ln[\pi(\boldsymbol{X}_i)] + (1 - y_i)\ln[1 - \pi(\boldsymbol{X}_i)]\} \quad (۴-۲۷)$$

در این حالت چون تعداد متغیرهای مستقل k عدد است پس $k + 1$ عدد معادله درست‌نمایی خواهیم داشت:

$$\sum_{i=1}^{n} [y_i - \pi(\boldsymbol{X}_i)] = 0 \quad (۴-۲۸)$$

$$\sum_{i=1}^{n} x_{ij}[y_i - \pi(\boldsymbol{X}_i)] = 0, \qquad j = 1, \dots, k \quad (۴-۲۹)$$

مانند حالت قبل با استفاده از روش‌های عددی مبتنی بر تکرار و استفاده از نرم‌افزارها می‌توان مقادیر پارامترها را برآورد کرد.

بعد از به دست آوردن مقادیر برآورد شده برای پارامترها و قرار دادن آن‌ها در رابطه (۴-۲۲) می‌توان $\pi(\hat{x}_i)$ احتمال اینکه مقدار متغیر پاسخ برابر یک باشد را محاسبه کرد.

٤-٤. شبکه‌های بیزی و مدل بیز ساده

٤-٤-١. قضیه بیز

قضیه بیز از قضایای مهم و پرکاربرد نظریه احتمالات و روشی برای دسته‌بندی پدیده‌ها بر پایه احتمال وقوع یا عدم وقوع آن‌ها است. اگر بتوانیم افراز مناسبی را برای فضای نمونه‌ای مفروضی انتخاب کنیم، با دانستن اینکه کدام یک از پیش‌آمدهای افراز شده رخ داده است، بخش مهمی از عدم قطعیت تقلیل می‌یابد.

این قضیه از آن جهت مفید است که می‌توان از طریق آن احتمال یک پیش‌آمد را با مشروط کردن نسبت به وقوع یا عدم وقوع یک پیش‌آمد دیگر محاسبه کرد. در بسیاری از حالت‌ها، محاسبه احتمال یک پیش‌آمد به صورت مستقیم کاری دشوار است. با استفاده از این قضیه و مشروط کردن پیش‌آمد موردنظر نسبت به پیش‌آمدهای دیگر، می‌توان احتمال موردنظر را محاسبه کرد.

معادله اصلی

فرض می‌کنیم $B_1, \dots, B_k$ یک افراز برای فضای S نمونه‌ای تشکیل دهد. طوری که به ازای هر $j = 1, \dots, k$ داشته باشیم:

$$P(B_j) > 0 \qquad (۴-۳۰)$$

و فرض کنید A پیش‌آمدی با فرض $P(A) > 0$ باشد، در این صورت به ازای $i = 1, \dots, k$ داریم:

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^{k} P(B_j)P(A|B_j)} \qquad (۴-۳۱)$$

٤,١-٤. دسته‌بند بیز ساده

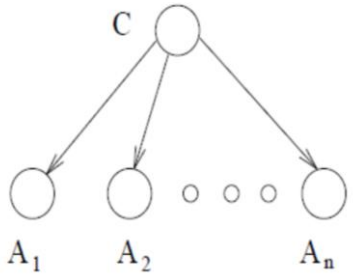
یکی از دسته‌بندهای موثر در این زمینه که در عملیات پیش‌بینی با دسته‌بندهای بسیار قوی رقابت می‌کند، دسته‌بند بیز ساده است. این دسته‌بند برای یادگیری از داده‌های آموزش موقعیت احتمالاتی هر ویژگی A_i با در نظر گرفتن برچسب دسته C استفاده می‌کند. سپس، دسته‌بندی با به‌کارگیری قوانین بیز برای محاسبه احتمال وقوع C با در نظر گرفتن ثابت‌های $A_1, \dots, A_n$ و سپس پیش‌بینی دسته با بیشترین احتمال صورت می‌گیرد. این محاسبات، با در نظر گرفتن یک استقلال قوی، عملی خواهد بود: همه ویژگی‌های A_i با وجود دسته C از هم مستقلند. استقلال در اینجا به معنی استقلال احتمالاتی است. به این معنی که A از B زمانی که دسته C وجود دارد، در صورت برقراری

$$\Pr(A|B, C) = \Pr(A|C) \qquad (۴-۳۱)$$

برای همه مقادیر ممکن A، B و C در صورتی که

$$\Pr(C) > 0 \qquad (۴-۳۲)$$

مستقل است.



شکل ۳-٤. نمونه گراف مدل بیزی (میرزایی، ۱۳۹۵)

در عملکرد بیز ساده فرض فوق به وضوح غیرواقعی است. برای روبرو شدن با این مشکل به بهترین شکل، به یک زبان و ماشین مناسب برای کار و تغییر فرضیه مذکور نیاز داریم، که هر دو در شبکه‌های بیزی موجود هستند.

زمانی که صحبت از شبکه‌های بیزی به میان می‌آید، بیز ساده دارای ساختار ساده‌ای خواهد بود که در شکل ۴-۴ نشان داده شده است. این شبکه، فرض اصلی دسته‌بند ساده بیز را در خود دارد که عبارت است از اینکه، هر ویژگی (هر برگ در شبکه)، با در نظر گرفتن حالت متغیر دسته (ریشه در شبکه)، از ویژگی‌های دیگر مستقل است. با توجه به در دست داشتن ابزار مناسب برای کار و تغییر فرضیه استقلال، این سوال واضح به ذهن می‌رسد که آیا می‌توان دسته‌بندهای بهتری را با استفاده از یادگیری نامحدود شبکه‌های بیزی استنتاج کرد؟ در این راستا، در بخش بعدی به صورت جزئی‌تر به شبکه‌های بیز پرداخته خواهد شد.

۲-٤-٤. یادگیری شبکه‌های بیزی

این شبکه‌ها، گراف‌های بدون دور جهت‌دار هستند که شرایط را برای توزیع احتمالاتی مشترک[۱] در مجموعه‌ای از متغیرهای تصادفی فراهم می‌کنند. هر راس گراف یک متغیر تصادفی و یال‌ها، ارتباط مستقیم بین متغیرهای تصادفی را نشان می‌دهند. به طور دقیق‌تر، شبکه حالات استقلال موقعیتی زیر را بیان می‌کند: هر متغیر، از متغیرهای غیر از والدین خود مستقل است. این استقلال، در ادامه، باعث کاهش پارامترها در هنگام انجام محاسبات توزیع احتمالاتی می‌شود. پارامترهای احتمالاتی در جدول‌هایی ارائه می‌شوندکه هر کدام مختص یکی از متغیرها هستند. با استفاده از استقلال بیان شده در شبکه، توزیع مشترک با استفاده از این توزیع‌های موقعیتی محلی، به طور خاص تعیین می‌شوند.

مجموعه متناهی $U = \{X_1, \dots X_n\}$ را در نظر بگیرید که از متغیرهای تصادفی گسسته تشکیل شده است که هر متغیر X_i ممکن است مقدار خود را از یک مجموعه متناهی دیگر بگیرد و با $Val(X_i)$ نشان داده شود. در این بخش از حروف بزرگ برای نشان دادن نام متغیرها و از حروف کوچک برای نشان دادن مقادیر اختصاص یافته به متغیرها استفاده شده است. مجموعه متغیرها با حروف پررنگ بزرگ و مقادیر اختصاص یافته به آن‌ها با حروف پررنگ کوچک نمایش داده شده‌اند. P نشان‌دهنده توزیع احتمالاتی مشترک متغیرهای U و $\boldsymbol{X}$، $\boldsymbol{Y}$ و $\boldsymbol{Z}$ زیرمجموعه‌های U هستند. گوییم $\boldsymbol{X}$ و $\boldsymbol{Y}$ بر اساس موقعیت، با وجود $\boldsymbol{Z}$، از یکدیگر مستقل هستند اگر برای هر $\boldsymbol{x} \in Val(\boldsymbol{X})$، $\boldsymbol{y} \in Val(\boldsymbol{Y})$ و $\boldsymbol{z} \in Val(\boldsymbol{Z})$ زمانیکه $P\left(\boldsymbol{y} \text{ و } \boldsymbol{z}\right) > 0$ داشته باشیم که $P\left(\boldsymbol{x} \middle| \boldsymbol{y} \text{ و } \boldsymbol{z}\right) = P(\boldsymbol{x}|\boldsymbol{z})$.

یک شبکه بیزی یک گراف بدون دور جهت‌دار تفسیرپذیر است که یک توزیع احتمالاتی مشترک را بین مجموعه‌ای از متغیرهای تصادفی U بیان می‌کند. به عبارت دیگر، یک شبکه بیزی برای U یک زوج $B = (G, \theta)$ است که G یک گراف غیرچرخشی مستقیم است که رئوس آن نشان‌دهنده

[۱] joint probability distribution

متغیرهای تصادفی $X_1, \dots, X_n$ و یال‌های آن نشان‌دهنده ارتباط بین این متغیرهاست. گراف G فرضیات استقلالی را بیان می‌کند: هر متغیر X_i از دیگر متغیرهای G به جز والدین خود مستقل است. عضو دوم زوج مرتب، θ، مجموعه‌ای از پارامترها را که توصیف‌کننده کمیت شبکه هستند، بیان می‌کند. این مجموعه شامل پارامتر $\theta_{X_i|\pi_{x_i}} = P_B(X_i|\pi_{x_i})$ برای هر متغیر x_i از X_i، و π_{x_i} از π_{X_i} می‌شود که در آن π_{X_i} نشان‌دهنده مجموعه والدین X_i در G است. یک شبکه بیزی B یک توزیع احتمالاتی مشترک یکتا در مجموعه U را با استفاده از رابطه زیر تعریف می‌کند:

$$P_B(X_1, \dots, X_n) = \prod_{i=1}^{n} P_B(X_i|\pi_{X_i}) = \prod_{i=1}^{n} \theta_{X_i|\pi_{X_i}} \quad (۴-۳۳)$$

اگر مجموعه آموزش $D = \{\boldsymbol{u_1}, \dots, \boldsymbol{u_N}\}$ را به عنوان زیرمجموعه‌ای از $\boldsymbol{U}$ داشته باشیم، مساله اصلی در یادگیری شبکه‌های بیزی می‌تواند یافتن شبکه B به‌گونه‌ای باشد که بیشترین هم‌خوانی را با D داشته باشد. رویکرد متداول در برخورد با این مساله معرفی یک تابع امتیاز است که هر شبکه را با توجه به داده‌های آموزش برآورد می‌کند و سپس به دنبال بهترین شبکه با توجه به این تابع می‌گردد.

فصل پنجم

روش پیشنهادی

در این فصل، جزئیات روش پیشنهادی برای استخراج پیشنهادهای پژوهش از پایان‌نامه‌ها و رساله‌های فارسی زبان را مورد بررسی قرار می‌دهیم. در این راستا، در ابتدا، جزئیات مجموعه داده ایجاد شده جهت آموزش و آزمایش روش پیشنهادی توصیف خواهد شد. در ادامه، مجموعه ویژگی‌های در نظر گرفته شده در دسته‌بند پیشنهادی بیان شده و سپس، نحوه استفاده از دسته‌بند پیشنهادی ارائه خواهد شد. نتایج ارزیابی روش پیشنهادی و نمونه‌هایی از خروجی‌های آن در بخش بعدی مشاهده خواهد شد و در نهایت، نتیجه به‌کارگیری روش پیشنهادی در سامانه «گنجینه پیشنهادهای پژوهش پایان‌نامه‌ها و رساله‌ها» ارائه خواهد شد.

۱-۵. ایجاد مجموعه‌ای برچسب‌گذاری شده از پیشنهادهای پژوهش در زبان فارسی

وجود یک مجموعه داده برچسب‌گذاری شده از ابزارهای ضروری برای طراحی روش‌های مبتنی بر یادگیری ماشینی با نظارت است. با توجه به این مهم و عدم وجود چنین مجموعه داده‌ای برای مساله استخراج پیشنهادهای پژوهش در زبان فارسی، در این پژوهش مجموعه داده‌ای بدین منظور ایجاد شده است. مجموعه داده موردنظر با استفاده از تعداد ۳۰۰ پارسا تهیه شده است که به صورت تصادفی، از سال‌ها، دانشگاه‌ها و رشته‌های مختلف انتخاب شده‌اند. ایجاد مجموعه داده از پارساها به صورت زیر صورت گرفته است:

۱- در ابتدا مرتبط‌ترین بخش به پیشنهادهای پژوهش از هر پارسا استخراج شده است.

۲- در ادامه، جملات بخش‌های استخراج شده به دست آمده و به هر جمله برچسب «پیشنهاد پژوهش» یا «غیر پیشنهاد پژوهش» توسط کاربر انسانی اختصاص یافته است.

از بین ۳۰۰ پارسای بررسی شده در این پژوهش، تنها ۱۷۵ عدد از آن‌ها کامل بوده و سایر موارد تنها شامل بخش‌های ابتدایی بوده و در نتیجه حذف گردیدند. بخش‌های استخراج شده از ۱۷۵ پارسای باقیمانده، دربردارنده ۲۱۳۱ جمله بوده که از این بین، ۶۳۱ جمله حاوی پیشنهادهای پژوهش و ۱۵۰۰ جمله فاقد پیشنهاد پژوهش بودند.

در ادامه، ابتدا نمونه‌هایی از عناوین بخش‌های استخراج شده و سپس نمونه‌هایی از جملات این بخش‌ها و برچسب‌های اختصاص یافته به آن‌ها آورده شده است.

عناوین پرتکرار بخش‌های استخراج شده:

- پیشنهادات پایان‌نامه
- پیشنهادها
- پیشنهادهای پژوهشی و کاربردی
- پیشنهاد برای تحقیقات آتی
- پیشنهادات پژوهشی مستخرج از رساله
- نتیجه‌گیری
- پیشنهادات اکتشافی
- جمع بندی و نتیجه گیری نهایی
- پیشنهادهایی برای تحقیق بیشتر
- نتیجه گیری و پیشنهادات
- پیشنهادات
- پیشنهادات پژوهشی
- نتیجه نهایی
- نتایج و پیشنهادها
- توصیه‌هایی برای آینده
- خلاصه و نتیجه‌گیری
- بحث و نتیجه گیری
- نتیجه‌گیری نهایی
- مطالعات آتی

در ادامه نمونه‌هایی از جملات و برچسب اختصاص‌یافته به آن‌ها ارائه شده است:

- پیشنهاد می‌شود تا پژوهش‌های آتی به بررسی نقش متغیرهایی چون سبک‌های مقابله با استرس بپردازند که نقش تعدیل‌کننده را در رابطه با فشارروانی و فرسودگی تحصیلی می‌توانند ایفا می‌کنند.(پیشنهاد پژوهش)

- ۱. مطالعات مشابه بر روی سایر ماهیان منطقه انجام شود و نتایج مقایسه گردد. (پیشنهاد پژوهش)
- ۷. همچنین می‌توان مطالعات نانواکوتوکسیکولوژی را جهت بررسی اثرات سمی نانوذرات بر گونه‌های آبزی مورد توجه قرار داد. (پیشنهاد پژوهش)
- در پایان پیشنهادهای زیر برای مطالعات بیشتر در جهت افزایش بازدهی گرمایی گلخانه ارائه می‌گردد: (غیر پیشنهاد پژوهش)
- ۱. پیشنهاد می‌شود در تحقیقی مشابه چهار گروه سایتوکاینی Th_1، TH_2، Th_{17} و T_{reg} سنجیده شوند. (پیشنهاد پژوهش)
- مهم‌ترین موضوعات نیازمندِ تحقیق بدین شرح است: (غیر پیشنهاد پژوهش)
- با توجه به پذیرش ضرورت تعامل و همکاری قوای سه گانه ، نیاز به پیش بینی شیوه ها و نهادهای نظارتی برای تعدیل این همکاری ، بر کسی پوشیده نیست. (غیر پیشنهاد پژوهش)
- ۱- پیشنهاد می‌شود که چنین طرحی در مورد دیگر گروه‌های سنی معتادین نیز به اجرا درآید تا اثربخشی آن در مورد سنین مختلف آشکار شود. (پیشنهاد پژوهش)
- همانگونه که در ادبیات بررسی شد خودتنظیمی و خودکارآمدی پرهیز از مواد افراد به عنوان متغیرهای مهم در موفقیت های درمانی اعتیاد و تداوم تغییر در پرهیز از مواد ذکر شده است. (غیر پیشنهاد پژوهش)
- با توجه به نقش حساس و خطیر گمرک در چرخه اقتصادی کشور و در جهت کنترل و مقابله با جرایم و تخلفات گمرکی، موارد زیر را به عنوان پیشنهاد مطرح میکنیم: (غیر پیشنهاد پژوهش)
- در پاسخ به پرسش دوم و با توجه به نتایج این پژوهش و پژوهش های پیشین، باید اذعان داشت که روشهای مقداری میتوانند در تمام مراحل طراحی همچون فرم‌یابی، نماسازی، منظرسازی، طراحی پلان و بهینه سازی بر مبنای کاهش مصرف انرژی و بهره‌وری کمک طراح بوده و او را در روند پاسخ یابی یاری نماید. (غیر پیشنهاد پژوهش)
- ۱۲. تعیین حق عضویت و پرداخت حداقل سرمایه به نسبت اراضی آبی، باغات و اراضی دیم (غیر پیشنهاد پژوهش)
- ۱- مطالعه جهت شناسایی دقیق علت رویداد زمین لرزه اهر – ورزقان (پیشنهاد پژوهش)
- – بررسی و مقایسه شدت ارتباط بین هزینه سرمایه و ارزش بازار شرکت ها در دوره های رونق و رکود. (پیشنهاد پژوهش)

- ۲. تاثیر اسانس گیاه *S. khuzestanica* بر سایر آبزیان (پیشنهاد پژوهش)
- – پیشنهاد می شود از میوه های *R- Conglomeratus* در طب سنتی ایران به عنوان ماده رنگ موی گیاهی و همچنین به عنوان عامل بند آورنده خونریزی، التیام دهنده و کاهش دهنده فشار و چربی خون استفاده شود. (غیر پیشنهاد پژوهش)
- در این راستا، بهتر است مباحثی نظیر حمل کالاهای خطرناک و مسؤولیت ناشی از حمل آن ها، بیمه مسؤولیت و شیوه رسیدگی به این دعاوی نیز مورد بازبینی و تجدیدنظر قرار گیرد. (غیر پیشنهاد پژوهش)
- همچنین محققین در قالب پژوهش‌های بعدی می‌توانند با توجه به محدودیت دسترسی به داده‌ها اثر سایر متغیرها با مصرف انرژی، اثرات غیرخطی انرژی و ... را بررشد اقتصادی کشورهای مختلف بررسی کرده و یا با بکارگیری تحلیل‌های اقتصاد سنجی فضایی بیزینی اثرات منطقه ای متغیرهای فوق را بر رشد اقتصادی کشورها تبیین نمایند. (پیشنهاد پژوهش)

۲-۵. استخراج ویژگی‌ها

در این قسمت مجموعه ویژگی‌های در نظر گرفته شده در روش پیشنهادی توصیف خواهد شد. ویژگی‌های در نظر گرفته شده با توجه به کارهای پیشین انجام شده در زمینه استخراج پیشنهادهای پژوهش در زبان انگلیسی، و بررسی مجموعه داده آماده شده در بخش اول این فصل به دست آمده است.

ویژگی‌های در نظر گرفته شده عبارتند از:

۱. جمله در چندمین پاراگراف از متن ورودی است.
۲. جمله چندمین جمله از پاراگراف است.
۳. طول پاراگراف.
۴. »پاراگراف دربردارنده جمله» یک آیتم از یک لیست است.
۵. وجود یکی از واژگان {«بیشتر»، «خواه»، «آتی»، «آینده»} در جمله.
۶. شروع جمله با یکی از واژگان {«بررسی»، «مطالعه»، «توسعه»، «تعمیم»، «استفاده»، «اجرا»، «مطالعات»}.
۷. وجود یکی از واژگان {«بررسی»، «مطالعه»، «توسعه»، «تعمیم»، «استفاده»، «اجرا»، «مطالعات»} در جمله.
۸. وجود یکی از واژگان {«بود»، «شد»} در جمله.
۹. وجود یکی از واژگان {«پژوهش»، «تحقیق»، «محقق»، «دانشجو»، «پژوهشگر»} در جمله.

۱۰. وجود یکی از واژگان {«پیشنهاد»، «توصیه»} در جمله.
۱۱. وجود یکی از واژگان {«شود»، «گیرد»} در جمله.
۱۲. وجود یکی از واژگان {«واگذار»، «موکول»} در جمله.
۱۳. شروع جمله با یکی از عبارات {«اما»، «ولی»، «بنابراین»، «در نتیجه»، «زیرا»، «لذا»، «همچنین»، «به عنوان مثال»، «برای مثال»}.
۱۴. ختم جمله به «.».
۱۵. ختم جمله به «:».
۱۶. تا ۳۰. ویژگی‌های ۱ تا ۱۵ ذکر شده در بالا برای جمله قبل از جمله فعلی. در صورتی که جمله فعلی، اولین جمله از بخش احتمالا دربردارنده پیشنهادهای پژوهش باشد، از null برای تمام ویژگی‌ها استفاده خواهد شد.
۳۱. تا ۴۵. ویژگی‌های ۱ تا ۱۵ ذکر شده در بالا برای جمله بعد از جمله فعلی. در صورتی که جمله فعلی، آخرین جمله از بخش احتمالا دربردارنده پیشنهادهای پژوهش باشد، از null برای تمام ویژگی‌ها استفاده خواهد شد.

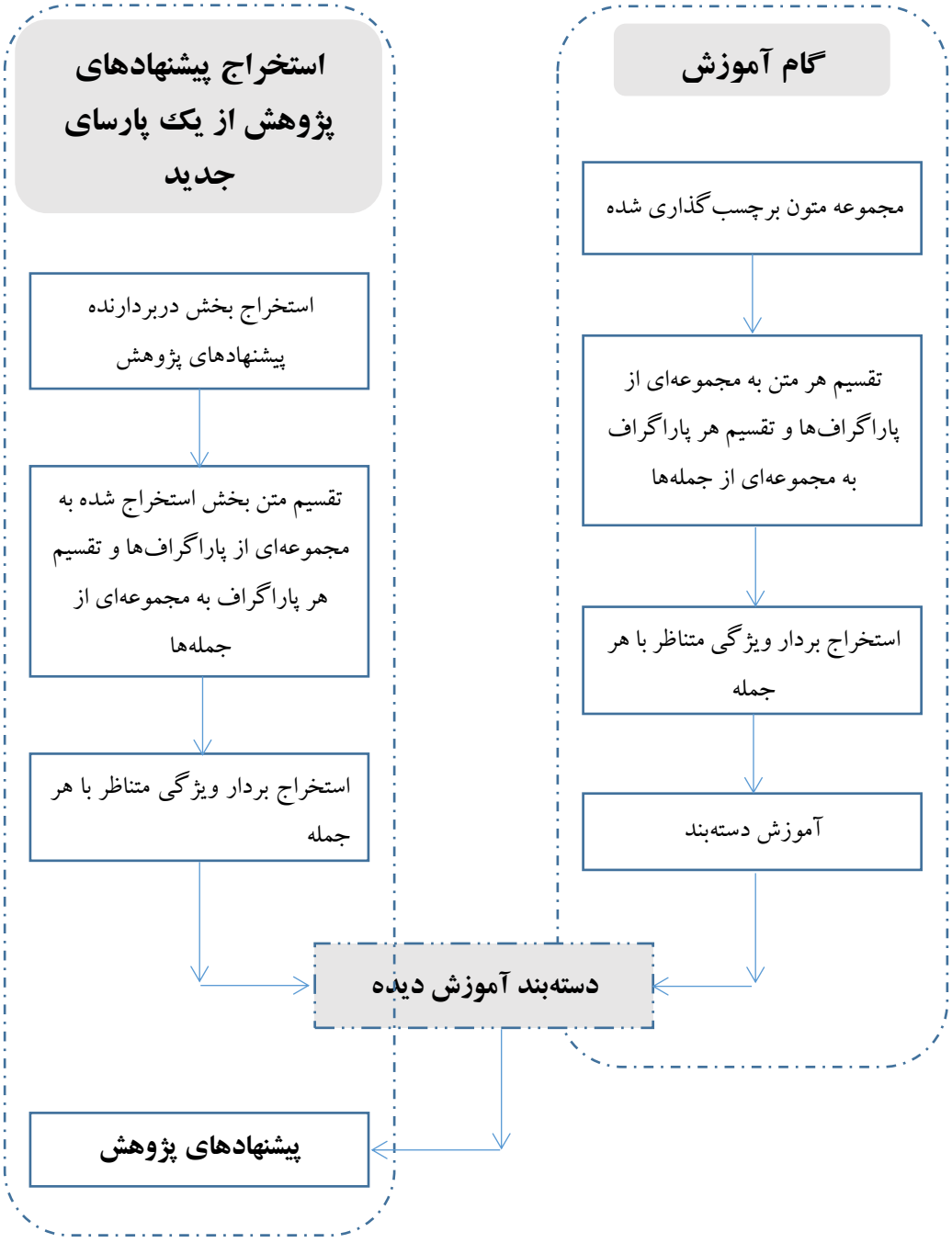
۳-۵. جزئیات روش پیشنهادی

روش پیشنهادی، که جزئیات آن در شکل ۵-۱ نمایش داده شده است، از دو فاز تشکیل شده است: فاز آموزش و فاز استخراج پیشنهادهای پژوهش از یک پارسای جدید.

آموزش

۱. در ابتدا یکسان‌سازی نویسه‌ها بر روی داده‌های ورودی انجام می‌شود. این گام از آن جهت مهم است که برای برخی از حروف فارسی، چندین نویسه با ظاهر یکسان وجود داشته و در نتیجه برای دستیابی به نتیجه مطلوب، یکی از پیش‌پردازش‌های لازم برای متن‌کاوی در زبان فارسی، اصلاح و یکسان‌سازی نویسه‌ها است که این مهم در روش ارائه شده نیز انجام می‌شود. الگوریتم edit_text در پیوست الف دربردارنده یکسان‌سازی‌های صورت گرفته در این زمینه است.
۲. تقسیم متون به واحدهای سازنده: در ادامه، هر متن موجود در مجموعه داده تهیه شده به دنباله‌ای از واحدها تقسیم‌بندی می‌شود که در این پژوهش از جمله به عنوان واحد استفاده شده است.
۳. محاسبه بردار ویژگی: در ادامه بردار ویژگی متناظر با هر واحد محاسبه می‌شود.

۴. در ادامه، پس از استخراج بردار ویژگی متناظر با هر واحد موجود در مجموعه داده آموزش، دسته‌بند مورد استفاده با استفاده از این داده‌ها آموزش داده می‌شود.


گام آموزش
مجموعه متون برچسب‌گذاری شده
تقسیم هر متن به مجموعه‌ای از پاراگراف‌ها و تقسیم هر پاراگراف به مجموعه‌ای از جمله‌ها
استخراج بردار ویژگی متناظر با هر جمله
آموزش دسته‌بند
دسته‌بند آموزش دیده
استخراج پیشنهادهای پژوهش از یک پارسای جدید
استخراج بخش دربردارنده پیشنهادهای پژوهش
تقسیم متن بخش استخراج شده به مجموعه‌ای از پاراگراف‌ها و تقسیم هر پاراگراف به مجموعه‌ای از جمله‌ها
استخراج بردار ویژگی متناظر با هر جمله
پیشنهادهای پژوهش

استخراج پیشنهادهای پژوهش از یک پارسای جدید

۱. شناسایی بخشی از یک پارسا که دربردارنده پیشنهادهای پژوهش است.

روش پیشنهادی در این گام، از یک الگوریتم اکتشافی جهت شناسایی بخش موردنظر در یک پارسا استفاده می‌کند. الگوریتم اکتشافی به‌کار رفته در این گام مراحل زیر را دنبال می‌کند:

- **پیش‌پردازش**: در ابتدا نرمال‌سازی متن شامل یکسان‌سازی نویسه‌ها انجام می‌شود.
- **شناسایی عناوین بخش‌هایی که احتمالا حاوی پیشنهادهای پژوهش هستند**: یکی از راه‌کارهای ممکن برای این مرحله استفاده از فهرست مطالب پارساها است. با این حال، از آن‌جا که بسیاری از دانش‌آموختگان از سبک[۱] یکسان با سایر قسمت‌های متن پارسا برای نگارش عناوین فصل‌ها و بخش‌ها و ایجاد فهرست مطالب استفاده کرده‌اند، به‌کارگیری چنین رویکردی منجر به دستیابی به نتایج مناسب نگردید. با توجه به این مساله، در این مرحله از روش پیشنهادی، در ابتدا تمام‌متن یک پارسای ورودی به دنباله‌ای از پاراگراف‌ها تقسیم شده و سپس پاراگراف‌هایی با اندازه کوتاه (بین ۵ تا ۴۰ کاراکتر) به عنوان پاراگراف‌های احتمالا نشان‌دهنده عناوین بخش‌ها و فصل‌های پارسای ورودی در نظر گرفته شدند. همچنین، پاراگراف‌های مابین دو عنوان احتمالی تشخیص داده شده نیز به عنوان بدنه آن بخش در نظر گرفته شد. در ادامه، بخش‌هایی که یکی از کلمات {«نتیجه»،«پیشنهاد»،«بحث»،«آتی»،«آینده»} در عنوان آن‌ها وجود دارد به عنوان بخش‌هایی که احتمالا حاوی پیشنهادهای پژوهش هستند در نظر گرفته می‌شوند.[۲]
- **حذف کلمات بی‌معنی:** در این مرحله، هر یک از کلمات {«و»،«از»،«برای»،«ها»،«های»} در صورت وجود در عنوان از آن حذف می‌شوند. این گام بدین دلیل انجام شده که در روش پیشنهادی، تعداد کلمات

---

[۱] style

[۲] لازم به ذکر است که این مرحله تنها در صورتی لازم به انجام است که فایل پارسا با فرمت Word باشد. در صورتی که فایل پارسا با فرمت Tex باشد، با توجه به ساخت‌یافته بودن این نوع فایل‌ها، تنها لازم است عبارات موجود در تگ‌های \section، \subsection و \subsubsection را به عنوان خروجی این مرحله در نظر بگیریم. جزئیات بیشتر، در پیوست الف، که دربردارنده کد پایتون روش پیشنهادی است، قابل مشاهده است.

عنوان در شناسایی عنوان بخش حاوی پیشنهادهای پژوهش موثر است. لازم به ذکر است که در اینجا منظور از کلمه عبارت است از هر دنباله‌ای از کاراکترها که بین دو کاراکتر فاصله محصور شده باشد.

- **شناسایی یکتای بخش دربرگیرنده پیشنهادهای پژوهش**: در این قسمت از مجموعه کلمات {«پیشنهاد»، «بیشتر»، «پژوهش»، «آتی»، «آینده»، «تحقیق»، «بعدی»، «مطالعات»، «مطالعه»} به عنوان کلمات مثبت و مجموعه کلمات {«اجرایی»، «کاربردی»، «نتیجه»} به عنوان کلمات منفی استفاده شده و با توجه به وجود کلمات فوق و همچنین تعداد واژگان موجود در عنوان موردنظر، یک امتیاز به هر یک از عناوین مشخص شده در مرحله قبل داده شد. در ادامه، عنوانی که بالاترین امتیاز را به دست آورده و پاراگراف‌های مابین آن تا عنوان بعدی شناسایی شده در تمام‌متن پارسا، به عنوان بخش دربردارنده پیشنهادهای پژوهش در نظر گرفته شده و برای بررسی در سطح جمله به مرحله بعد داده می‌شود.

۲. شناسایی پیشنهادهای پژوهش

در این گام از دسته‌بند آموزش دیده در بخش قبل برای تمایز پیشنهادهای پژوهش از دیگر جملات متن ورودی استفاده می‌شود. جزئیات این گام به شرح زیر است:

- متن ورودی به مجموعه پاراگراف‌ها و در ادامه به مجموعه جملات تقسیم می‌گردد.
- بردار ویژگی متناظر با هر جمله از متن ورودی محاسبه می‌شود.
- از دسته‌بند آموزش دیده، استفاده شده و به هر جمله برچسب «پیشنهاد پژوهش» یا «غیر پیشنهاد پژوهش» داده می‌شود.

٤-٥. ارزیابی روش پیشنهادی

برای ارزیابی، روش پیشنهادی با استفاده از زبان برنامه‌نویسی پایتون و دسته‌بندهای بیز ساده، رگرسیون لجستیک و SVM پیاده‌سازی شده است. برای استفاده از دسته‌بندهای بیز ساده، رگرسیون لجستیک و SVM در پیاده‌سازی انجام شده از کتابخانه scikit-learn (Pedregosa et al. 2011) استفاده شده است.

در ادامه، ابتدا نمونه‌هایی از نتایج اعمال روش پیشنهادی برای استخراج جملات درباره پیشنهادهای پژوهش ارائه شده؛ سپس، ارزیابی الگوریتم ابتکاری پیشنهادی برای استخراج بخش احتمالا حاوی پیشنهادهای پژوهش از پارساهای فارسی‌زبان؛ و در نهایت، نتایج ارزیابی کلی روش برچسب‌دهی پیشنهادی با به‌کارگیری دسته‌بندهای مختلف ارائه خواهد شد.

۱-٤-۵. نمونه‌هایی از نتایج به دست آمده با استفاده از روش پیشنهادی

در ادامه، نمونه‌هایی از نتایج حاصل از اعمال روش پیشنهادی با به‌کارگیری دسته‌بندهای مختلف روی چند بخش نتیجه‌گیری و پیشنهادهای پژوهش ارائه شده است.

جدول ۵-۱. نتایج اعمال دسته‌بندهای آموزش دیده بر روی متن‌های ورودی نمونه. در نتایج ارائه شده، متون مربوط به پارساهای مختلف با خط حاشیه پررنگ از یکدیگر جدا شده‌اند. همچنین، موارد اشتباه تشخیص داده شده با پس‌زمینه رنگی مشخص شده‌اند.

جملات متن ورودی	Naïve bayes	Logestic regression	SVM
۴-۶-پیشنهاداتی برای مطالعات اینده	×	×	×
در این پژوهش فرض شده که میان دو تصمیم تغییر وسیله شخصی و انتخاب وسیله نقلیه جایگزین همبستگی وجود دارد و تنها به براورد یک ضریب همبستگی ان هم برای نشان دادن میزان همبستگی میان مدل اول (تغییر وسیله نقلیه از سواری شخصی) و مدل دوم (انتخاب وسیله نقلیه جایگزین) استفاده شد.	✓	×	×
در پژوهش‌های دیگر می‌توان با فرض وجود همبستگی میان تصمیم تغییر وسیله نقلیه شخصی با هر یک از شیوه‌های جایگزین پرداخت و به ازای هر شیوه نقلیه یک ضریب همبستگی براورد کرد.	✓	✓	✓
مدل‌های توامان قابل مقایسه با مدل‌های اشیانه‌ای می‌باشند.	✓	×	×
برای درک بیشتر کارایی مدل همزمان ارایه شده در این پایان‌نامه می‌توان نتایج به دست امده را با یک مدل اشیانه‌ای دو سطحی که در سطح اول تغییر و یا عدم تغییر وسیله نقلیه شخصی، و در سطح دوم انتخاب سایر شیوه‌های سفر وجود دارد مورد بررسی قرار داد.	✓	×	✓
پیشنهادات کاربردی و پژوهشی	✓	×	×
یکی از حرکات بسیار مؤثر اموزش و پرورش در راستای تقویت روابط ایران و عراق، می‌تواند اصلاح تصویر عراق در ذهن نوجوانان و کودکان ایرانی باشد.	✓	✓	✓
اساساً تصویر و نگرش نسبت به عراق در ذهن کودکان ایرانی لازم است دستخوش تغییراتی شود.	✓	✓	✓
برای اینکه نسل خردسال بتواند نگرش درستی نسبت به همسایه هم‌ایین غربی‌مان داشته باشد و تعاملات مؤثر و موفق چهره‌به‌چهره با انها تجربه نماید، لازم است سیاست‌گذاری جامعی در خصوص نظام اموزش و	✓	✓	✓

برنامه‌های درسی دانش‌اموزان در این حیطه انجام گیرد.			
طبعاً این سیاست‌گذاری نیاز به کارهای پژوهشی دارد.	✓	×	×
به عنوان مثال، وارد شدن خاطرات اربعین یا خاطراتی از مهمان‌نوازی‌های میزبانان عراقی در کتاب‌های درسی می‌تواند یکی از سیاست‌های مؤثر باشد.	✓	✓	✓
۵-۸ پیشنهادهای پژوهشی:	×	×	×
- بررسی استلزام های یافته های پژوهش حاضر برای تدوین برنامه های درسی فبک در پایه های مختلف تحصیلی	✓	✓	✓
- بررسی مبانی جامعه شناسی برنامه فبک	✓	✓	✓
- بررسی سوابق تربیت فکری در اسلام و آرای فلاسفه مسلمان	✓	✓	✓
- بررسی موانع اجرایی برنامه فبک در ایران و ارائه راهکار هایی برای آن.	✓	✓	✓
- طراحی الگوی برنامه فلسفه برای کودکان متناسب با تعلیم و تربیت اسلامی برای سایر دوره های تحصیلی.	✓	✓	✓
- بررسی دیدگاه های علمان نسبت به مولفه های این برنامه از مطلوب واجرایی بودن	✓	✓	✓
پیشنهادات جهت انجام تحقیقات آتی	✓	×	×
۱. به پژوهشگرانی که تصمیم دارند در آینده روی این موضوع مطالعه کنند توصیه می شود از بررسی روشهای پژوهش نظیر مشاهده مصاحبه نیز استفاده کنند در صورت امکان گذراندن مدت کوتاهی در جامعه مورد مطالعه بسیار مفید است.	✓	✓	✓
۲. این موضوع در زمان و مکان های متفاوت صورت می گیرد تا همبستگی بین این متغیرها در هر زمان و هر مکان مشخص شود و اهمیت رابطه این دو متغیر، بیشتر معلوم گردد.	✓	✓	✓
تحقیق درباره عوامل و موانع موثر بر همه متغیرهای مذکور در سایر مکان ها و زمان های دیگر سال صورت گیرد و بصورت تحقیقی جداگانه مورد بررسی قرار گیرد.	✓	✓	✓
۳. اندازه گیری متغیرهای اصلی در این تحقیق بصورت محلی صورت گرفته و قابل تعمیم و مقایسه با سایر شهر ها و منطقه ها نیست اما چنانچه این اندازه گیری به صورت عمومی برای شهرها و روستاهای دیگر انجام پذیرد در این صورت امکان مقایسه و در نتیجه الگوبرداری ممکن خواهد بود.	✓	✓	✓
۴. انجام مطالعات به صورت طولی، مقایسه ای در افزایش سطح اطمینان به داده ها موثر است.	✓	✓	✓
۵. انتخاب یک نمونه بزرگتر از جوامع برای کاهش مشکل تعمیم نتایج.	✓	✓	✓

پیشنهادات	✓	×	×
مطالعه انجام شده را می توان به عنوان یکی از اولین مطالعات انجام شـده در این سطح بر روی داده¬های لرزه ای در منطقه اهر – ورزقان دانست.	✓	✓	✓
لذا قطعا با کاستی هایی همراه خواهد بود.	✓	✓	✓
اما بعنوان پیشنهاد جهت مطالعات آتی این موارد قابل بررسی خواهند بود:	✓	×	×
۱- تمرکز مطالعات زلزله شناسی بر نـواحی شـمال غربـی بـه عنـوان یکی از نواحی فعال	✓	✓	✓
۲- مطالعه جهـت شناسـایی دقیـق علـت رویـداد زمـین لـرزه اهـر – ورزقان	✓	✓	✓
۳- ایجاد شبکه متراکم جهت شناسایی گسل های موجود در منطقه	✓	✓	✓
۴-سعی در شناسایی بهتر پس¬لرزه ها و مدل سازی آنها	✓	✓	✓

۲-٤-۵. ارزیابی

در این بخش به ارزیابی گام‌های روش پیشنهادی پرداخته می‌شود. در ابتدا، ارزیابی الگوریتم ابتکاری پیشنهادی ارائه شده و در ادامه ارزیابی دسته‌بند پیشنهادی برای تشخیص پیشنهادهای پژوهش از غیرپیشنهادهای پژوهش ارائه می‌شود.

ارزیابی الگوریتم ابتکاری پیشنهادی برای شناسایی بخش در بردارنده پیشنهادهای پژوهش

برای ارزیابی الگوریتم ابتکاری پیشنهادی برای شناسایی بخش در بردارنده پیشنهادهای پژوهش، از پارساهای استفاده شده برای ایجاد مجموعه داده پیشنهادهای پژوهش استفاده شده است. همانطور که در بخش ۵-۱ بیان شد، برای ایجاد مجموعه داده موردنظر از ۱۷۵ پارسای فارسی زبان استفاده شده است. برای ارزیابی الگوریتم ابتکاری پیشنهادی، آن را بر روی هر یک از ۱۷۵ پارسای موردنظر اعمال کرده و در نهایت درصد مواردی که الگوریتم پیشنهادی به درستی بخش موردنظر را شناسایی کرده است، محاسبه شد. نتایج حاصل بیانگر این است که روش پیشنهادی در حدود ۹۲ درصد موارد به درستی بخش دربردارنده پیشنهادهای پژوهش را شناسایی می‌کند.

ارزیابی روش پیشنهادی برای تشخیص پیشنهادهای پژوهش از غیرپیشنهادهای پژوهش

برای ارزیابی روش ارائه شده برای تشخیص پیشنهادهای پژوهش از غیرپیشنهادهای پژوهش از مجموعه داده طراحی شده در ابتدای این فصل و روش اعتبارسنجی متقابل ۱۰ سطحی استفاده شده است. در این روش، داده‌ها به ۱۰ بخش تقسیم‌بندی شده و هر بخش یک بار به عنوان داده آزمایش به دسته‌بند آموزش دیده با استفاده از داده‌های ۹ بخش دیگر داده شده و نتایج ارزیابی محاسبه می‌شود.

مجموعه داده ایجاد شده شامل ۲۱۳۱ جمله می‌باشد که به هر یک برچسب «پیشنهاد پژوهش» یا «غیر پیشنهاد پژوهش» اختصاص یافته است. نتایج پیاده‌سازی روش پیشنهادی با دسته‌بندهای مختلف روی مجموعه داده‌های موردنظر در جداول ۵-۲ تا ۵-۴ ارائه شده است. با توجه به خروجی به دست آمده می‌توان نتیجه گرفت که به‌کارگیری دسته‌بندهای رگرسیون لجستیک و ماشین بردار پشتیبان منجر به دستیابی به نتایج بهتری برای مساله استخراج پیشنهادهای پژوهش می‌شود.

جدول ۵-۲. نتایج پیاده‌سازی با دسته‌بند بیز ساده

	دقت	فراخوانی	F1	تعداد
پیشنهاد پژوهش	۰/۵۱	۰/۹۷	۰/۶۷	۶۳۱
غیر پیشنهاد پژوهش	۰/۹۸	۰/۶۱	۰/۷۵	۱۵۰۰
میانگین	۰/۸۴	۰/۷۲	۰/۷۳	۲۱۳۱

جدول ۵-۳. نتایج پیاده‌سازی با دسته‌بند رگرسیون لجستیک

	دقت	فراخوانی	F1	تعداد
پیشنهاد پژوهش	۰/۷۹	۰/۷۷	۰/۷۸	۶۳۱
غیر پیشنهاد پژوهش	۰/۹۰	۰/۹۱	۰/۹۱	۱۵۰۰
میانگین	۰/۸۷	۰/۸۷	۰/۸۷	۲۱۳۱

جدول ۵-٤. نتایج پیاده‌سازی با دسته‌بند SVM

	دقت	فراخوانی	F1	تعداد
پیشنهاد پژوهش	۰/۸۰	۰/۷۸	۰/۷۹	۶۳۱
غیر پیشنهاد پژوهش	۰/۹۱	۰/۹۲	۰/۹۱	۱۵۰۰
میانگین	۰/۸۸	۰/۸۸	۰/۸۸	۲۱۳۱

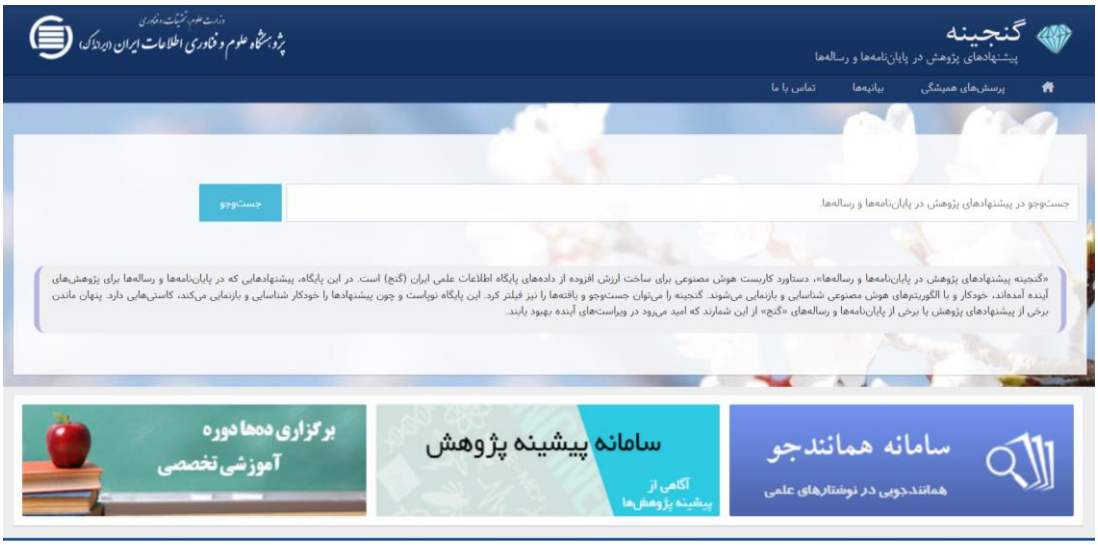
همچنین، مقدار صحت به دست آمده از دسته‌بندهای مختلف در جدول ۵-۵ نمایش داده شده است.

جدول ۵-۵. محاسبه پارامتر صحت در استفاده از دسته‌بندهای مختلف

	بیز ساده	رگرسیون لجستیک	SVM
صحت	۰/۷۰	۰/۸۸	۰/۸۸

۵-۵. به‌کارگیری روش پیشنهادی

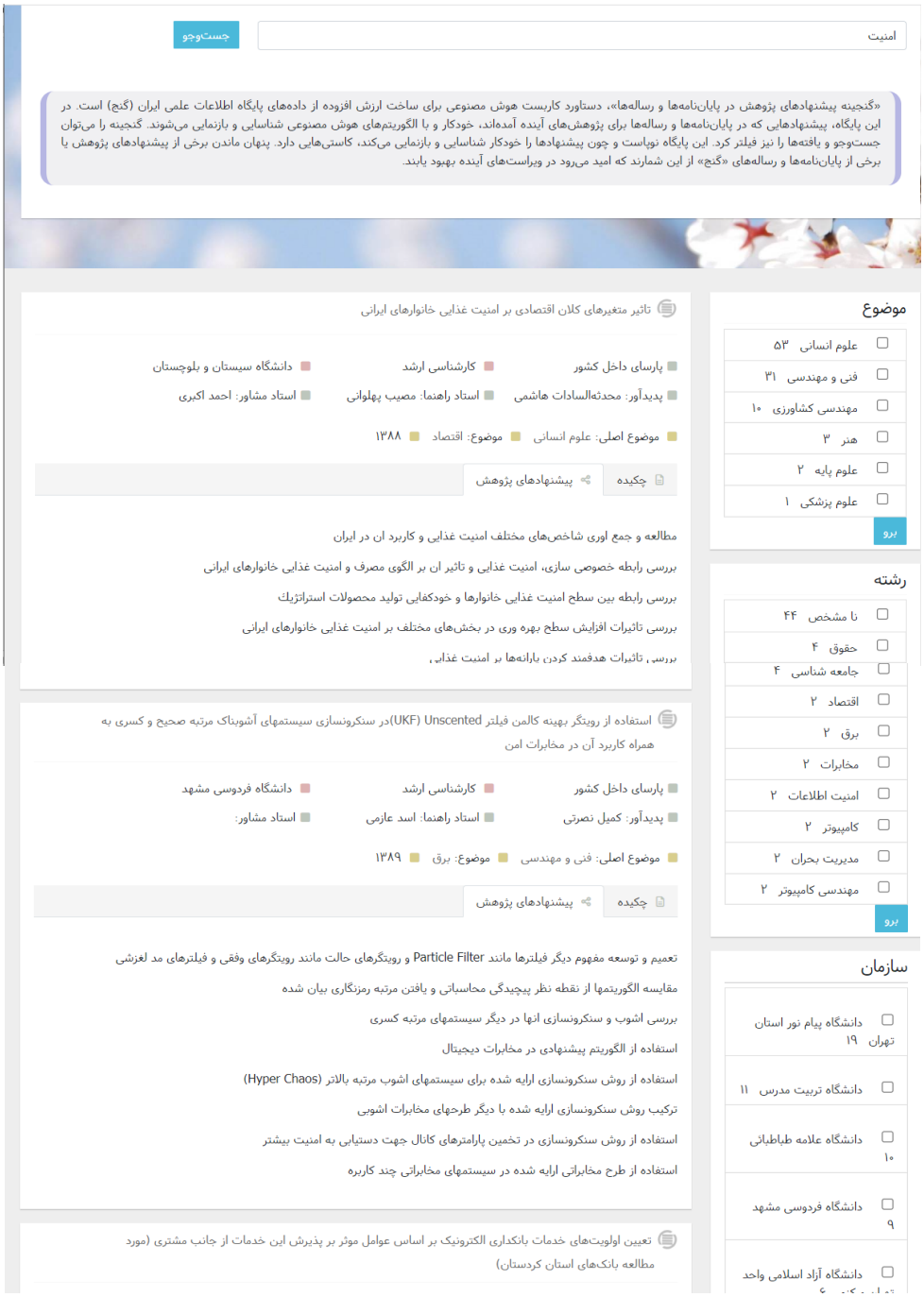
با توجه به محدودیت‌های موجود در توسعه و تغییر سامانه گنج، و طبق توافقات صورت گرفته مابین ریاست پژوهشگاه و مرکز فناوری اطلاعات و امنیت فضای مجازی پژوهشگاه، سرویس‌های ارزش افزوده می‌توانند روی پایگاه گنج اما در قالب یک زیردامنه جدا از گنج خدمت‌دهی کنند. این تصمیم با توجه به شرایط خاص توسعه گنج و محافظت از آن در برابر رخدادهای خدمات ارزش افزوده در پیش گرفته شده است. با توجه به این محدودیت‌ها، برای به کارگیری روش پیشنهادی، سامانه «گنجینه پیشنهادهای پژوهش پایان‌نامه‌ها و رساله‌ها» در زیردامنه‌ای مشابه با سامانه گنج ایجاد شده است. این سامانه از طریق آدرس https://ganjpro.irandoc.ac.ir قابل دسترس است و در آن کاربران قادر به جستجو در پیشنهادهای پژوهش استخراج شده از پایان‌نامه‌ها و رساله‌های موجود در پایگاه گنج هستند (شکل ۵-۲).



شکل ۵-۲. صفحه اصلی سامانه گنجینه پیشنهادهای پژوهش پایان‌نامه‌ها و رساله‌ها.

در این سامانه، کاربر می‌تواند با وارد کردن عبارت موردنظر به جستجو در میان پیشنهادهای پژوهش استخراج شده از پایان‌نامه‌ها و رساله‌ها، و جستجو بر روی نتایج و محدودسازی آن‌ها بپردازد. به عنوان مثال، نتیجه جستجوی واژه امنیت در این سامانه در شکل ۵-۳ ارائه شده است. همانطور که می‌توان در این شکل دید، در این سامانه پیشنهادهای پژوهش به صورت یک Tab جدید (همانند آن‌چه در مورد چکیده، کلیدواژه‌ها، فهرست نوشته‌ها و فهرست‌های منابع در سامانه گنج وجود داشت) به هر پارسا اضافه شده و هر پیشنهاد پژوهش در یک پاراگراف مجزا نمایش داده می‌شود. در این سامانه، دسترسی به اطلاعات کامل و در صورت امکان، تمام‌متن نتایج جستجو، از طریق کلیک بر روی عنوان هر یک از نتایج جستجو و هدایت به سامانه گنج فراهم شده است. همچنین، در سامانه گنجینه

پیشنهادهای پژوهش در پایان‌نامه‌ها و رساله‌ها، کاربران می‌توانند از نوار سمت راست استفاده کرده و به محدودسازی جستجوی خود با توجه به اقلامی همچون موضوع، رشته، سازمان، سال و مقطع بپردازند.



فصل ششم
نتیجه‌گیری و پیشنهادهای پژوهش

۱-٦. نتیجه‌گیری

بخش دربردارنده پیشنهادهای پژوهش یکی از بخش‌های مهم در اسناد و مدارک علمی است. این بخش ارزش زیادی برای پژوهشگران داشته و سرنخ‌هایی در رابطه با موضوعات پژوهشی و ایده‌های مناسب برای حل آن‌ها در اختیار پژوهشگران قرار می‌دهد. پژوهشگران به طور معمول از اطلاعات ارائه شده در این بخش برای انتخاب مساله پژوهش مناسب استفاده می‌کنند. با توجه به زمان انتشار اسناد علمی، می‌توان با بررسی بخش کارهای آتی، مساله‌های پژوهشی به‌روز یا به عبارتی عامیانه‌تر، مساله‌هایی که جای کار دارند، را شناسایی کرد و در راستای حل آن‌ها تلاش نمود. با این حال، با توجه به رشد روزافزون مدارک علمی، مطالعه اسناد موجود و استخراج پیشنهادهای پژوهش برای پژوهشگران امری دشوار بوده و روزبه‌روز نیز بر دشواری آن افزوده می‌شود. با توجه به سختی استخراج دستی پیشنهادهای پژوهش از اسناد علمی، در این پژوهش مساله استخراج خودکار این اقلام اطلاعاتی از پارساهای فارسی‌زبان موجود در پایگاه گنج پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)، در نظر گرفته شده و با بررسی روش‌های ارائه شده بدین منظور در زبان انگلیسی و همچنین ویژگی‌های زبان فارسی، یک روش خودکار بدین منظور ارائه شده است. روش ارائه شده، مبتنی بر روش‌های یادگیری ماشینی با نظارت است. با توجه به نیاز روش‌های یادگیری ماشینی با نظارت به مجموعه داده‌ای برچسب‌دار برای آموزش و آزمایش، و همچنین عدم وجود چنین مجموعه داده‌ای برای مساله استخراج پیشنهادهای پژوهش در زبان فارسی، به عنوان یکی از دست‌آوردهای این پژوهش چنین مجموعه داده‌ای ایجاد گردید. در ادامه، روش پیشنهادی با استفاده از زبان برنامه‌نویسی پایتون پیاده‌سازی شد. همچنین، کتابخانه scikit-learn برای استفاده از دسته‌بندهای مختلف در پیاده‌سازی انجام شده به کار رفته است.

نتایج پیاده‌سازی روش پیشنهادی با دسته‌بندهای بیز ساده، رگرسیون لجستیک و SVM روی مجموعه داده‌های ایجاد شده بیانگر این است که به‌کارگیری دسته‌بندهای رگرسیون لجستیک و ماشین بردار پشتیبان منجر به دستیابی به نتایج بهتری برای مساله استخراج پیشنهادهای پژوهش می‌شود.

پس از بررسی نتایج آزمایشی، در نهایت، روش پیشنهادی تحت عنوان یک ماژول نرم‌افزاری بر روی داده‌های موجود در پایگاه گنج پیاده و نتایج حاصل در سامانه «گنجینه پیشنهادهای پژوهش پایان‌نامه‌ها و رساله‌ها» از طریق آدرس اینترنتی https://ganjpro.irandoc.ac.ir ارائه شده و به‌روزرسانی می‌شود.

۶-۲. پیشنهادهای پژوهش

در این پژوهش به مساله استخراج پیشنهادهای پژوهش از پارساهای فارسی‌زبان پرداخته شد و با به‌کارگیری روش‌های یادگیری ماشینی با نظارت، یک روش برای استخراج پیشنهادهای پژوهش ارائه گردید. به عنوان یکی از کارهای قابل انجام برای ارتقا این پژوهش می‌توان به استفاده از دیگر روش‌های یادگیری ماشینی مانند ماشین بردار پشتیبان دوقلو[۱]، CRF، یادگیری عمیق و روش‌های ترکیبی برای استخراج پیشنهادهای پژوهش و مقایسه نتایج با نتایج به دست آمده در این پژوهش اشاره کرد.

استفاده از روش ارائه شده در این پژوهش برای اهداف علم‌سنجی از دیگر کارهایی است که می‌توان در ادامه این پژوهش انجام داد. به عنوان مثال می‌توان با به‌کارگیری روش پیشنهادی در سطوح مختلفی مانند رشته، دانشگاه و کشور به ارزیابی میزان انجام پیشنهادهای پژوهش ارائه شده در پارساها پرداخت و از نتایج حاصل برای سیاست‌گذاری استفاده کرد.

---

[۱] Twin SVM

فهرست منابع

اصلان‌پور، خالد. ۱۳۹۱. داده‌کاوی با رگرسیون لجستیک. پایان‌نامه کارشناسی ارشد دانشگاه تبریز، دانشکده علوم ریاضی، گروه آمار.

پاک‌نیت، نصراله، جلال‌الدین نصیری، و عمار جلالی‌منش. ۱۳۹۷. طراحی یک روش هوشمند تجزیه رشته‌های مرجع در زبان فارسی. پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک).

کارگر، مرتضی. ۱۳۹۰. دسته‌بندی داده‌ها با استفاده از روش SVM. پایان‌نامه کارشناسی دانشگاه شهید باهنر کرمان.

میرزایی، الهام. ۱۳۹۵. پیش‌بینی متاستاز سرطان با استفاده از رویکرد مدل بیز ساده و الگوریتم ژنتیک. پایان‌نامه کارشناسی ارشد دانشگاه تربیت مدرس.

نصیری، جلال‌الدین. ۱۳۹۰. بازشناسی اعمال انسان با رویکرد مقاوم سازی دسته‌بند تفکیکی. رساله دکتری دانشگاه تربیت مدرس، تهران.

Chen, J., and H. Zhuge. 2019. Automatic generation of related work through summarizing citations. Concurrency and Computation: Practice and Experience. 31(3): e4261.

Chen, L., and H. Fang. 2019. KO KNOWLEDGE ORGANIZATION. 46(3): 171-186.

Collett, D. 1991. Modelling binary data. London: Chapman and Hall.

Cox, D., and E. Snell. 1989. Analysis of binary data. London: Chapman and Hall.

Cronin, B., D. Shaw, and K. La Barre. 2003. A cast of thousands: Coauthorship and subauthorship collaboration in the 20th century as manifested in the scholarly journal literature of psychology and philosophy. Journal of the American Society for information Science and Technology. 54(9): 855-871.

Cronin, B., G. McKenzie, L. Rubio, and S. Weaverozniak. 1993. Accounting for influence: Acknowledgments in contemporary sociology. Journal of the American Society for Information Science. 44(7): 406-412.

Councill, I.G., C.L. Giles, H. Han, and E. Manavoglu. 2005. Automatic acknowledgement indexing: expanding the semantics of contribution in the CiteSeer digital library. Proceedings of the 3rd international conference on Knowledge capture, pp. 19-26.

Giles, C.L., and I.G. Councill. 2004. Who gets acknowledged: Measuring scientific contributions through automatic acknowledgment indexing. Proceedings of the National Academy of Sciences of the United States of America 101(51): 17599-17604.

Hao, W., Z. Li, Y. Qian, Y., Wang, and C. Zhang. 2020. The ACL FWS-RC: A Dataset for Recognition and Classification of Sentence about Future Works. Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, pp. 261-269.

Hosmer, D.W., and S. Lemeshow. 2000. Applied Logestic Regression. Second edition. New York: John Wiley Sons.

Hoang, C.D.V., and M. Y. Kan. 2010. Towards automated related work summarization. Proceedings of the 23rd International Conference on Computational Linguistics: Posters. pp. 427-435.

Hu, Y., and X. Wan. 2015. Mining and analyzing the future works in scientific articles. arXiv preprint arXiv:1507.02140.

Hu, Y., and X. Wan. 2014. Automatic Generation of Related Work Sections in Scientific Papers: An Optimization Approach. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1624-1633.

Khabsa, M., P. Treeratpituk, and C.L. Giles. 2012. Ackseer: a repository and search engine for automatically extracted acknowledgments from digital libraries. Proceedings of the 12th ACM/IEEE-CS joint conference on Digital Libraries. pp. 185-194.

Kleinbaum, D. 1994. Logistic regression: a self- learning text. New York: Springer–Verlag.

Li, K., and E. Yan. 2019. Using a keyword extraction pipeline to understand concepts in future work sections of research papers. ISSI. pp. 87-98.

Liu, B. 1998. Minimax chance constrained programming models for fuzzy decision system. Information Sciences. 112: 25-38.

Masulli, F., and G. Valentini. 2000. Comparing decomposition methods for classification. Proceedings of the Fourth International Conference on Knowledge- Based Intelligent Engineering Systems and Allied Technologies (KES 2000). pp. 788–791.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel et al. 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research. 12: 2825-2830.

Qian, Y., Z. Li, W. Hao, Y. Wang, and C. Zhang. 2021. Using Future Work Sentences to Explore Research Trends of Different Tasks in a Special Domain. Proceedings of the Association for Information Science and Technology. 58(1): 532-536.

Teufel, S. 2017. Do "Future Work" sections have a purpose? Citation links and entailment for global scientometric questions. In BIRNDL@ SIGIR (1).

Vapnik, V. N. 1995. The Nature of Statistical Learning Theory. New York: Springer-Verlag.

Vapnik, V. N. 1998. Statistical Learning Theory. New York: John Wiley & Sons.

Wainer, J., and E. Valle. 2013. What happens to computer science research after it is published? Tracking CS research lines. Journal of the American Society for Information Science and Technology. 64(6): 1104-1111.

Zhu, Z., D. Wang, and S. Shen. 2019. Recognizing sentences concerning future research from the full text of JASIST. Proceedings of the Association for Information Science and Technology. 56(1): 858-859.

پیوست ۱: کد پایتون روش پیشنهادی

در این بخش، در ابتدا کتابخانه‌های مورداستفاده در کد پایتون معرفی شده و در ادامه، کد پایتون روش پیشنهادی ارائه می‌شود.

کتابخانه‌ها

در این قسمت به معرفی کتابخانه‌های استفاده شده در کد پایتون تهیه شده پرداخته می‌شود.

- **کتابخانه __future__**: این کتابخانه به برنامه‌نویسان این امکان را می‌دهد تا بتوانند ماژول‌ها، توابع و کتابخانه‌های موردنظر خود را به‌گونه‌ای تعریف کنند تا در نسخه‌های آیندهٔ این زبان امکان اجرای برنامه روی ویژگی‌های جدید ارائه‌شده برای ماژول‌ها، توابع و کتابخانه‌های بیان شده فراهم شود. به‌کارگیری این ماژول منجر بدین خواهد شد تا برنامه موردنظر تا حدود زیادی با قابلیت‌های نسخهٔ‌های جدید این زبان سازگار باشد.
- **کتابخانه re**: این کتابخانه یک مجموعه ابزار برای کار با عبارات منظم در پایتون فراهم آورده است.
- **کتابخانه prickle**: از این کتابخانه serialize کردن استفاده می‌شود. فرایند serialization روشی برای تبدیل ساختار داده به فرم خطی است که می‌تواند از طریق شبکه ذخیره یا منتقل شود.
- **کتابخانه docx2txt**: از این کتابخانه برای استخراج متون از فایل‌های با پسوند docx استفاده می‌شود.
- **کتابخانه logging**: از این کتابخانه برای رویدادنگاری استفاده می‌شود.
- **کتابخانه matplotlib**: از این کتابخانه برای رسم نمودار استفاده می‌شود.
- **کتابخانه os**: این کتابخانه امکان تعامل برنامه با سیستم را فراهم و امکان اجرا و پیاده‌سازی فرآیندهای متنوع سیستمی را فراهم می‌کند
- **کتابخانه codecs**: این کتابخانه کلاس‌های پایه را برای کدک‌های استاندارد پایتون (کدگذارها و کدگشاها) تعریف می‌کند.
- **کتابخانه numpy**: با استفاده از این کتابخانه امکان استفاده از آرایه‌ها و ماتریس‌های بزرگ چند بعدی فراهم می‌شود. هم‌چنین با استفاده از این کتابخانه، می‌توان توابع ریاضیاتی سطح بالا را بر روی آرایه‌ها و ماتریس‌ها به کار گرفت.
- **کتابخانه docx**: از این کتابخانه برای ایجاد، خواندن، نوشتن و به‌روزرسانی فایل‌های ورد در پایتون استفاده می‌شود.

- **کتابخانه unidecode**: از این کتابخانه برای تبدیل اعداد فارسی به انگلیسی استفاده شده است.
- **کتابخانه random**: از این کتابخانه برای ایجاد اعداد تصادفی استفاده شده است.
- **کتابخانه pandas**: از این کتابخانه عمدتاً برای تجزیه و تحلیل داده‌ها و دستکاری داده‌های جدولی مربوطه در Dataframeها استفاده می‌شود
- **کتابخانه sklearn**: یک کتابخانه رایگان برای یادگیری ماشین در زبان برنامه‌نویسی پایتون است. این کتابخانه دربردارنده الگوریتم‌های زیادی از جمله ماشین بردار پشتیبان (SVM)، درخت تصمیم و k-نزدیک‌ترین همسایه است. همچنین این کتابخانه از کتابخانه‌های عددی و آماری پایتون مانند NumPy و SciPy پشتیبانی می‌کند.
- **کتابخانه math**: از این کتابخانه برای انجام عملیات‌های ریاضی استفاده شده است.

نصب کتابخانه‌ها

برای نصب کتابخانه‌های ذکر شده بر روی پایتون، لازم است تا مراحل زیر انجام شود:

- رابط خط فرمان (cmd) را باز می‌کنیم.
- رابط خط فرمان را به محل نصب کتابخانه‌های پایتون هدایت می‌کنیم.
- با استفاده از دستور pip install package_name کتابخانه موردنظر را نصب می‌نماییم.

کد پایتون تهیه شده

```
1. from __future__ import unicode_literals
2. import re
3. import pickle
4. import time
5. import docx2txt
6. import logging
7. import matplotlib.pyplot as plt
8. import os
9. import codecs
10.      import numpy as np
11.      from docx import Document
12.      from docx.shared import Inches
13.      from unidecode import unidecode
14.      from random import randint
15.      import pandas as pd
16.      from sklearn.naive_bayes import GaussianNB
```

```
17.      from sklearn.linear_model import LogisticRegression
18.      from sklearn.svm import SVC
19.      from sklearn.metrics import classification_report,
   confusion_matrix
20.      import math
21.      clssfr = GaussianNB()
22.      #clssfr = LogisticRegression(random_state=3)
23.      #clssfr = SVC(C=4,kernel='linear',probability=True)
24.
25.      current_dir=os.getcwd()
26.      files_dir=current_dir+"\Data"
27.
28.      numbers=["0","1","2","3","4","5","6","7","8","9"]
29.
30.      def convertnumbs(txt):
31.          '''
32.          Converts the Persian numbers to English numbers.
33.
34.          Parameters:
35.              txt (string): input text.
36.
37.          Returns:
38.              newtxt: output text.
39.          '''
40.          newtxt=""
41.          for c in txt:
42.              if c=="۱":
43.                  newtxt=newtxt+unidecode(str(1))
44.              elif c=="۲":
45.                  newtxt=newtxt+unidecode(str(2))
46.              elif c=="۳":
47.                  newtxt=newtxt+unidecode(str(3))
48.              elif c=="۴":
49.                  newtxt=newtxt+unidecode(str(4))
50.              elif c=="۵":
51.                  newtxt=newtxt+unidecode(str(5))
52.              elif c=="۶":
53.                  newtxt=newtxt+unidecode(str(6))
54.              elif c=="۷":
55.                  newtxt=newtxt+unidecode(str(7))
56.              elif c=="۸":
57.                  newtxt=newtxt+unidecode(str(8))
58.              elif c=="۹":
59.                  newtxt=newtxt+unidecode(str(9))
60.              elif c=="۰":
61.                  newtxt=newtxt+unidecode(str(0))
62.              else:
63.                  newtxt=newtxt+c
64.          return newtxt
```

```
65.
66.     def edit_text(text):
67.         '''
68.         Unifies the characters in Persian Language.
69.
70.         Parameters:
71.             text (string): input text.
72.
73.         Returns:
74.             text: output text.
75.         '''
76.         text=convertnumbs(text)
77.         text=text.replace(" "," ")
78.         text=text.replace(" "," ")
79.         text=text.replace("”"," ")
80.         text=text.replace("”"," ")
81.         text=text.replace(" "," ")
82.         text=text.replace(" "," ")
83.         text=text.replace("–","-")
84.         text=text.replace("-","-")
85.         text=text.replace(" "," ")
86.         text=text.replace("‘"," ")
87.         text=text.replace("’"," ")
88.         text=text.replace("&"," & ")
89.         text=text.replace("ط","ط")
90.         text=text.replace("ﺑ","ب")
91.         text=text.replace("ﺎ","ا")
92.         text=text.replace("ﻳ","ي")
93.         text=text.replace("ﯽ","ي")
94.         text=text.replace("ئ","ي")
95.         text=text.replace("ى","ي")
96.         text=text.replace("ي","ي")
97.         text=text.replace("ى","ي")
98.         text=text.replace("ۍ","ي")
99.         text=text.replace("ێ","ي")
100.        text=text.replace("ې","ي")
101.        text=text.replace("ې","ي")
102.        text=text.replace("أ","ا")
103.        text=text.replace("أ","ا")
104.        text=text.replace("إ","ا")
105.        text=text.replace("آ","ا")
106.        text=text.replace("ا","ا")
107.        text=text.replace("ٱ","ا")
108.        text=text.replace("أ","ا")
109.        text=text.replace("إ","ا")
110.        text=text.replace("ﺍ","ا")
111.        text=text.replace("ب","ب")
112.        text=text.replace("ﺒ","ب")
```

```
113.        text=text.replace("پ","ب")
114.        text=text.replace("ب","ب")
115.        text=text.replace("پ","پ")
116.        text=text.replace("ة","ه")
117.        text=text.replace("ۀ","ه")
118.        text=text.replace("ه","ه")
119.        text=text.replace("ۀ","ه")
120.        text=text.replace("ه","ه")
121.        text=text.replace("ۀ","ه")
122.        text=text.replace("ة","ه")
123.        text=text.replace("ه","ه")
124.        text=text.replace("ھ","ه")
125.        text=text.replace("ت","ت")
126.        text=text.replace("ٹ","ت")
127.        text=text.replace("ټ","ت")
128.        text=text.replace("ټ","ت")
129.        text=text.replace("ث","ث")
130.        text=text.replace("ج","ج")
131.        text=text.replace("چ","چ")
132.        text=text.replace("ح","ح")
133.        text=text.replace("ځ","ح")
134.        text=text.replace("خ","خ")
135.        text=text.replace("د","د")
136.        text=text.replace("ډ","د")
137.        text=text.replace("ډ","د")
138.        text=text.replace("ڈ","د")
139.        text=text.replace("ډ","د")
140.        text=text.replace("ذ","ذ")
141.        text=text.replace("ر","ر")
142.        text=text.replace("ۍ","ي")
143.        text=text.replace("ۀ","م")
144.        text=text.replace("ۂ","ح")
145.        text=text.replace("ۀ","م")
146.        text=text.replace("ۮ","د")
147.        text=text.replace("ض","ض")
148.        text=text.replace("ع","ع")
149.        text=text.replace("ۮ","د")
150.        text=text.replace("ر","ر")
151.        text=text.replace("س","س")
152.        text=text.replace("و","و")
153.        text=text.replace("ـل","ل")
154.        text=text.replace("ک","ک")
155.        text=text.replace("ن","ن")
156.        text=text.replace("ح","ح")
157.        text=text.replace("غ","غ")
158.        text=text.replace("ف","ف")
159.        text=text.replace("ۼ","ع")
```

```
160.        text=text.replace("ﻓ","ف")
161.        text=text.replace("ﺗ","ت")
162.        text=text.replace("ﻋ","ع")
163.        text=text.replace("ﻳ","ي")
164.        text=text.replace("ﺟ","ج")
165.        text=text.replace("ﻲ","ي")
166.        text=text.replace("ى","ي")
167.        text=text.replace("١","1")
168.        text=text.replace("ء","")
169.        text=text.replace("٢","2")
170.        text=text.replace("٣","3")
171.        text=text.replace("٤","4")
172.        text=text.replace("٥","5")
173.        text=text.replace("٨","8")
174.        text=text.replace("٧","7")
175.        text=text.replace("٦","6")
176.        text=text.replace("ي","ي")
177.        text=text.replace("و","و")
178.        text=text.replace("\ufeff","")
179.        text=text.replace("ي","ي")
180.        text=text.replace("5","5")
181.        text=text.replace("¬","-")
182.        text=text.replace(" ،","،")
183.        text=text.replace(" .",".")
184.        text=text.replace("‌","")
185.        text=text.replace("  "," ")
186.        text=text.replace("ﺗ","ت")
187.        text=text.replace("ـ","")
188.        text=text.replace("ﻪ","ه")
189.        text=text.replace("ﺗ","ت")
190.        text=text.replace("ﮔ","گ")
191.        text=text.replace("ﻟ","ل")
192.        text=text.replace("ﺷ","ش")
193.        text=text.replace("ﻫ","ه")
194.        text=text.replace("ﻧ","ن")
195.        text=text.replace("ﺧ","خ")
196.        text=text.replace("ط","ط")
197.        text=text.replace("م","م")
198.        text=text.replace("ي","ي")
199.        text=text.replace("ﺳ","س")
200.        text=text.replace("ـ","-")
201.        text=text.replace(" هاي ","هاي ")
202.        text=text.replace(" ها ","ها ")
203.        text=text.replace(" ي ","ي ")
204.        text=text.replace(" اي ","اي ")
205.        text=text.replace("–","-")
206.        text=text.replace("ﻻ","لا")
207.        text=text.replace(" -","-")
```

```
208.         text=text.replace("- ","-")
209.         text=text.replace("  "," ")
210.         return text
211.
212.
213.     sctn_inds2=["نتیجه","پیشنهاد","بحث","آتي","آینده"]
214.     sctn_inds=[]
215.     for word in sctn_inds2:
216.         sctn_inds.append(edit_text(word))
217.
218.     positive_words2=["پیشنهاد","بیشتر","پژوهش","
    آتی","آینده","تحقیق", "بعدي","مطالعات","مطالعه","توصیه"]
219.     positive_words=[]
220.     for word in positive_words2:
221.         positive_words.append(edit_text(word))
222.
223.     negetive_words2=["اجرایي","کاربردي","نتیجه"]
224.     negetive_words=[]
225.     for word in negetive_words2:
226.         negetive_words.append(edit_text(word))
227.
228.     stop_words2=["و","از","براي","ها","هاي"]
229.     stop_words=[]
230.     for w in stop_words2:
231.         stop_words.append(edit_text(w))
232.
233.     def section_score(par):### ekhtesas emtiaz be bakhsh
    haye kandid az payan name baraye moshakhas kardan bakhsh
    dar bar darandeh pishnahad haye pazhuhesh.
234.         '''
235.         Assigns some score to a candidate section title
    to determine whether it could be be the section containing
    future work suggestions or not.
236.
237.         Parameters:
238.             par (string): A title in a text
239.
240.         Returns:
241.             score (float): How much the received title
    could possibly be the title of the section containing
    future work suggestions.
242.         '''
243.         text=edit_text(par.text)
244.         text=text.replace("\r\n","")
245.         text=text.replace("\n","")
246.         text=text.replace("\r","")
247.         for ch in numbers+[".","-"]:
248.             text=text.replace(ch," ")
249.         text=text.replace("  "," ")
```

```
250.         if edit_text("پیشنهاد") in text:
251.             if edit_text("تحقیق") in text:
252.                 if edit_text("آینده") in text or
   edit_text("آتی") in text:
253.                     return 1
254.         for w in stop_words:
255.             text=text.replace(" "+w+" "," ")
256.         while text[:1] in
   ["0","1","2","3","4","5","6","7","8","9","-","."]:
257.             text=text[1:]
258.         while text[:1]==" ":
259.             text=text[1:]
260.         while text[-1:]==" ":
261.             text=text[:-1]
262.         text=text.replace("  "," ")
263.         score=-10
264.         count=0
265.         if 4<len(text)<40:
266.             score=0
267.             for word in positive_words:
268.                 if word in text:
269.                     count+=1
270.                     score+=1
271.             for word in negetive_words:
272.                 if word in text:
273.                     count+=1
274.                     score-=0.25
275.             score=score-(len(text.split(" "))-count)*0.5
276.             score=float(score/len(text.split(" ")))
277.         return score
278.
279.     def is_sctn_match(text):
280.         '''
281.         Checks whether a section title contains at least
   one of the "future works section" indicators or not.
282.
283.         Parameters:
284.             txt (string): input section title.
285.
286.         Returns:
287.             True/False.
288.         '''
289.         for ind in sctn_inds:
290.             if ind in text:
291.                 return True
292.         return False
293.
294.     def ext_rsrch_drctn_txt(doc_address):
295.         '''
```

```
296.         Extracts the section that contains future work
   suggestions.
297.
298.         Parameters:
299.             doc_address(string): The address of a
   document.
300.
301.         Returns:
302.             rsrch_drctn_text(string): The text of the
   section (if exists) that contains future work suggestions.
303.         '''
304.         document = Document(doc_address)
305.         pars = document.paragraphs
306.         pssbl_trgt_pars=[]
307.         num_pars=len(pars)
308.         i=0
309.         flag=False
310.         while i<num_pars:
311.             pair=[]
312.             par=pars[i]
313.             txt=edit_text(par.text)
314.             while txt.startswith(" "):
315.                 txt=txt[1:]
316.             while txt.endswith(" "):
317.                 txt=txt[:-1]
318.             if 3<len(txt)<40:
319.                 if is_sctn_match(txt) and i>len(pars)/2
   and len(txt.replace(" ",""))>4:
320.                     flag=True
321.                 else:
322.                     flag=False
323.             if flag:
324.                 pair.append(i)
325.                 pair.append(par)
326.                 pair.append(section_score(par))
327.                 pssbl_trgt_pars.append(pair)
328.             i=i+1
329.         pssbl_sctn_ttls=[]
330.         for pair in pssbl_trgt_pars:
331.             if 3<len(pair[1].text)<40:
332.                 pssbl_sctn_ttls.append(pair)
333.         future_work_section=[]
334.         ind=-1
335.         score=-10
336.         for i in list(range(len(pssbl_sctn_ttls))):
337.             pair=pssbl_sctn_ttls[i]
338.             if 4<len(pair[1].text)<40:
339.                 if pair[2]>=score:
340.                     ind=i
```

```
                    score=pair[2]
        rsrch_drctn_text=pars[pssbl_sctn_ttls[ind][0]].te
   xt+"\r\n"
        for i in list(range(pssbl_sctn_ttls[ind][0]+1,
   num_pars)):
            if 3<len(pars[i].text)<40:
                break
            else:
                rsrch_drctn_text+=pars[i].text+"\r\n"
        return rsrch_drctn_text

    def divide_section_2_sntnc(text):
        '''
        Divides a text to a set of sentences.

        Parameters:
            text: input text.

        Returns:
            sntncs (string[]): sentences of the input
   text.
        '''
        sntncs=[]
        pars=text.split("\r\n")
        for i in list(range(len(pars))):
            par=pars[i]
            if "." in par[:5]:
                par=par[:5].replace(".","-")+par[5:]
            par_sntncs=par.split(".")
            for j in list(range(len(par_sntncs))):
                sntncs.append(par_sntncs[j])
        return sntncs

    research_words2=["پژوهش","تحقیق","محقق","دانشجو"]
    research_words=[]
    for word in research_words2:
        research_words.append(edit_text(word))
    suggesting_words2=["پیشنهاد","توصیه"]
    suggesting_words=[]
    for word in suggesting_words2:
        suggesting_words.append(edit_text(word))
    suggest_words2=["شود", "گیرد"]
    suggest_words=[]
    for word in suggest_words2:
        suggest_words.append(edit_text(word))
    leftopen_words2=["واگذار","موکول"]
    leftopen_words=[]
    for word in leftopen_words2:
        leftopen_words.append(edit_text(word))
```

```
387.     future_words2=["ایـنده","اتـی","خواه","بـیشتر"]
388.     future_words=[]
389.     for word in future_words2:
390.         future_words.append(edit_text(word))
391.     other_words2=["بـررسي","مطـالـعه","تـوسعه","تـعمیم","
   اسـتفـاده","اجـرا","مطـالـعات"]
392.     other_words=[]
393.     for word in other_words2:
394.         other_words.append(edit_text(word))
395.     psv_words2=["بـود","شد"]
396.     psv_words=[]
397.     for word in psv_words2:
398.         psv_words.append(edit_text(word))
399.     rabt_words2=["امـا","ولـي","بـنـابـرايـن"," در
   بـراي "," بـه عنوان مـثـال","همچنین","لـذا","زیـرا","نـتیجه
   مـثـال"]
400.     rabt_words=[]
401.     for word in rabt_words2:
402.         rabt_words.append(edit_text(word))
403.
404.     def add_neighbor(data_ftrs):### afzoodan vizhegi haye
   motenazer ba jomalat hamsaye be jomle dar haal barresi.
405.         '''
406.         Adds features corresponding to the previous and
   next sentences to vector features of the current sentence.
407.
408.         Parameters:
409.             data_ftrs(Array): The set of feature vectors
   corresponding to sentences of a text.
410.
411.         Returns:
412.             new_data_ftrs(Array): The new set of feature
   vectors corresponding to the sentences.
413.         '''
414.         new_data_ftrs=[]
415.         for i in list(range(len(data_ftrs))):
416.             new=[]
417.             new.append(data_ftrs[i][0])
418.             new.append(data_ftrs[i][3])
419.             if data_ftrs[i][1]==True and
   data_ftrs[i][2]==True:
420.                 for j in list(range(4,len(data_ftrs[i])-
   1)):
421.                     new.append(False)
422.             else:
423.                 for j in list(range(4,len(data_ftrs[i])-
   1)):
424.                     new.append(data_ftrs[i-1][j])
425.             try:
```

```
                for j in list(range(4,len(data_ftrs[i])-1)):
                    new.append(data_ftrs[i+1][j])
            except:
                for j in list(range(4,len(data_ftrs[i])-1)):
                    new.append(False)
            for k in list(range(4,len(data_ftrs[i]))):
                new.append(data_ftrs[i][k])
            new_data_ftrs.append(new)
        return new_data_ftrs

    def train():
        '''
        Trains the classifier.
        '''
        text=""
        f_y=[]
        f_p=[]
        document = Document('Data/trn&tst.docx')
        for par in document.paragraphs:
            if par.style.name!= 'List Paragraph':
                text=text+edit_text(par.text)+"\r\n"
            else:
                text=text+"- "+edit_text(par.text)+"\r\n"
        docs_text=text.split("@@@")
        f_docs=[]
        f_docs.append(docs_text[0])
        for i in list(range(1,len(docs_text)-1)):
            f_docs.append(docs_text[i][3:])
        data_ftrs=[]
        for doc in f_docs:
            doc_snts_ftrs=Extract_features(doc)
            for row in doc_snts_ftrs:
                data_ftrs.append(row)
        new_data_ftrs=add_neighbor(data_ftrs)
        df1=pd.DataFrame(new_data_ftrs)
        x1=df1.drop(df1.columns[[0,1,len(new_data_ftrs[0])-1]],axis=1)
        y1=df1[df1.columns[len(new_data_ftrs[0])-1]]
        clssfr.fit(x1, y1)

    def Extract_features(doc_text):
        '''
        Extracts the feature vector corresponding to each sentence of the text

        Parameters:
            doc_text (string): The input text
```

```
471.
472.         Returns:
473.             sntncs_ftrs: The feature vectors
   corresponding to the sentences of the received text.
474.         '''
475.         doc_text=doc_text.replace("!!!","$$$")
476.         t_pars=doc_text.split("\r\n")
477.         pars=[]
478.         rand=randint(0,9)
479.         for par in t_pars:
480.             if len(par.replace(" ",""))>5:
481.                 temp=par
482.                 while temp[-1:]==" ":
483.                     temp=temp[:-1]
484.                 while temp[:1]==" ":
485.                     temp=temp[1:]
486.                 pars.append(temp)
487.         pre_lbl=0
488.         title=""
489.         try:
490.             title=pars[0]
491.         except:
492.             pass
493.         l1=False
494.         l2=False
495.         section_labels1=["پیشنهاد"]
496.         for w in section_labels1:
497.             if w in title:
498.                 l1=True
499.         section_labels2=["پژوهش","آتي","آینده"]
500.         for w in section_labels2:
501.             if w in title:
502.                 l2=True
503.         sntncs_ftrs=[]
504.         for i in list(range(len(pars))):
505.             par=pars[i]
506.             par=par.replace("...","---")
507.             for num in numbers:
508.                 if str(num)+". " in par[:5]:
509.                     par=par[:5].replace(str(num)+". ",
   str(num)+"- ")+par[5:]
510.             par_is_started_with_number_or_dash=False
511.             if par[:1] in numbers+["-"] or ("(" in
   par[:5] and ")" in par[:5]):
512.                 par_is_started_with_number_or_dash=True
513.             par_sntncs=par.split(". ")
514.             for j in list(range(len(par_sntncs))):
515.                 sntnc_ftrs=[]
516.                 lbl=0
```

```
517.                 if len(par_sntncs[j])>15:
518.                     sntnc=par_sntncs[j]
519.                     sntnc_ftrs.append(sntnc)
520.                     if i==0:
521.                         sntnc_ftrs.append(True)
522.                     else:
523.                         sntnc_ftrs.append(False)
524.                     if j==0:
525.                         sntnc_ftrs.append(True)
526.                     else:
527.                         sntnc_ftrs.append(False)
528.                     sntnc_ftrs.append(rand)
529.                     if "$$$" in sntnc:
530.                         lbl=True
531.                         sntnc=sntnc.replace("$$$","")
532.                     else:
533.                         lbl=False
534.                     while sntnc[-1:]==" ":
535.                         sntnc=sntnc[:-1]
536.                     sntnc_ftrs.append(int(len(pars[i])/30
   ))
537.                     sntnc_ftrs.append(par_is_started_with
   _number_or_dash)
538.                     sntnc_ftrs.append(l1)
539.                     sntnc_ftrs.append(l2)
540.                     sntnc_contains_future_words=False
541.                     for word in future_words:
542.                         if word in sntnc:
543.                             sntnc_contains_future_words=T
   rue
544.                             break
545.                     sntnc_ftrs.append(sntnc_contains_futu
   re_words)
546.
547.                     sntnc_is_started_with_other_words=Fal
   se
548.                     for word in other_words:
549.                         if word in sntnc[:10]:
550.                             sntnc_is_started_with_other_w
   ords=True
551.                             break
552.                     sntnc_ftrs.append(sntnc_is_started_wi
   th_other_words)
553.                     sntnc_contains_other_words=False
554.                     for word in other_words:
555.                         if word in sntnc[4:]:
556.                             sntnc_contains_other_words=Tr
   ue
557.                             break
```

```
                    sntnc_ftrs.append(sntnc_contains_othe
   r_words)
                    sntnc_contains_psvwords=False
                    for word in psv_words:
                        if word in sntnc[4:]:
                            sntnc_contains_psvwords=True
                            break
                    sntnc_ftrs.append(sntnc_contains_psvw
   ords)
                    sntnc_contains_research_words=False
                    for word in research_words:
                        if word in sntnc:
                            sntnc_contains_research_words
   =True
                            break
                    sntnc_ftrs.append(sntnc_contains_rese
   arch_words)
                    sntnc_contains_suggesting_words=False
                    for word in suggesting_words:
                        if word in sntnc:
                            sntnc_contains_suggesting_wor
   ds=True
                            break
                    sntnc_ftrs.append(sntnc_contains_sugg
   esting_words)
                    sntnc_contains_suggest_words=False
                    for word in suggest_words:
                        if word in sntnc:
                            sntnc_contains_suggest_words=
   True
                            break
                    sntnc_ftrs.append(sntnc_contains_sugg
   est_words)
                    sntnc_contains_leftopen_words=False
                    for word in leftopen_words:
                        if word in sntnc:
                            sntnc_contains_leftopen_words
   =True
                            break
                    sntnc_ftrs.append(sntnc_contains_left
   open_words)
                    sntnc_started_with_rabt_words=False
                    for word in rabt_words:
                        if word in sntnc[:10]:
                            sntnc_started_with_rabt_words
   =True
                            break
                    sntnc_ftrs.append(sntnc_started_with_
   rabt_words)
```

```
595.                        sntnc_is_ended_with_dot=False
596.                        if sntnc[-1:]==".":
597.                            sntnc_is_ended_with_dot=True

598.                        sntnc_ftrs.append(sntnc_is_ended_with
    _dot)
599.                        sntnc_is_ended_with_2dot=False
600.                        if sntnc[-1:]==":":
601.                            sntnc_is_ended_with_2dot=True

602.                        sntnc_ftrs.append(sntnc_is_ended_with
    _2dot)
603.                        sntnc_ftrs.append(edit_text("می‌شود")
    in sntnc or edit_text("میشود") in sntnc or edit_text("می
    شود") in sntnc)
604.                        sntnc_ftrs.append(lbl)
605.                        sntncs_ftrs.append(sntnc_ftrs)
606.            return sntncs_ftrs
607.
608.        def ext_rsrch_drctn_frm_nw_doc(doc_address):###
    estekhraj pishnahad haye pazhuhesh az yek payan name jadid
609.            '''
610.            Extracts future works from a Document.
611.
612.            Parameters:
613.                doc_address (string): The address of a
    document.
614.
615.            Returns:
616.                rsrch_drctns (string): future research
    suggestions of the received document (if it contains any).
617.            '''
618.            rsrch_drctns=""
619.            text=edit_text(ext_rsrch_drctn_txt(doc_address))
620.            print(text)
621.            print("+++++++")
622.            ftrs=Extract_features(text)
623.            nw_dt_ftrs=add_neighbor(ftrs)
624.            for i in list(range(len(nw_dt_ftrs))):
625.                r=nw_dt_ftrs[i]
626.                temp_r=[]
627.                temp_r.append(r)
628.                df=pd.DataFrame(temp_r)
629.                x=df.drop(df.columns[[0,1,len(r)-1]],axis=1)
630.                y=clssfr.predict(x)
631.                if y[0]==True:
632.                    rsrch_drctns+="- "+ftrs[i][0]+"\r\n"
633.            return rsrch_drctns
634.    train()
```

```
635.     with open ('Clssfr_NB_Pickle', 'wb') as f:
636.         pickle.dump(clssfr,f)
637.     doc_address='Data/0.docx'
638.     print(ext_rsrch_drctn_frm_nw_doc(doc_address))
```

پیوست ۲: مستندات فنی ماژول نرم‌افزاری تهیه شده

در این بخش، مستندات فنی متناظر با برنامه نهایی تهیه و ارائه شده ارائه می‌شود.

راهنمای کد نرم‌افزار:

زبان برنامه‌نویسی و بسته‌های استفاده شده

زبان برنامه نویسی: C# .netcore6

پایگاه داده: SQL Server

الگوی طراحی[۱] مورد استفاده: UnitOfWork & Repository

فرانت‌اند: Bootstrap و JQuery

زبان اصلی برنامه‌نویسی استفاده شده برای تهیه سامانه ارائه شده #C است که از فریم‌ورک .Net Core6 استفاده می‌نماید. برنامه ارائه شده یک Web application است که خدمت جستجو روی داده‌های استخراج شده از بخش پیشنهادهای پژوهش پارساها را ارائه می‌کند. برای ساخت پایگاه داده برنامه از Entity Framework Code-First استفاده شده است. برای بخش موتور جستجو از قابلیت Fulltext-Search پایگاه داده SQLServer استفاده شده است. داده‌های موردنظر توسط این سرویس نمایه شده تا در جستجوها مورد استفاده قرار گیرد. همچنین از بسته ReflectionIT برای صفحه‌بندی استفاده شده است.

کلاس‌های موجود در برنامه تهیه شده

کلاس Repository: لایه دسترسی به داده با الگوی طراحی UnitOfWork ایجاد شده است. براین اساس، متدهای موردنیاز در کلاس Repository قرار دارد که عبارتند از:

- کلاس GenericRepository: این کلاس شامل متدهای اصلی برای دسترسی به داده است و همان‌طور که از نام آن مشخص است به صورت Generic است. متدهای اصلی این کلاس شامل موارد زیر است:
  - Find(Expression<Func<T, bool>> expression): برای دسترسی به یک یا چند رکورد مشخص.
  - GetAll(): برای واکشی همه رکوردها
  - GetById(): دسترسی به یک رکورد مشخص با Id آن

این متدها از طریق رابط کاربری[۲] IGenericRepository در دسترس هستند.

---

[۱] Design pattern

[۲] Interface

کلاس FutureWorkRepository: این کلاس برای دسترسی به جدول FutureWorks ایجاد شده که خود از کلاس GenericRepository ارث‌بری می‌کند. متد اصلی این کلاس GetFutureWorks است. این متد دربرگیرنده بخش جستجو بوده و خروجی موردنظر را بر اساس QueryString داده شده، تهیه می‌کند. این جستجو با دستور EF.Functions.FreeText انجام می‌شود و همان‌طور که مشخص است از سرویس Fulltext-Search پایگاه داده بهره می‌برد. دسترسی به این کلاس از طریق رابط کاربری IFutureWorks خواهد بود.

کلاس ResultViewModel: این کلاس یک ساختار داده است که در لایه نمایش مورد استفاده قرار می‌گیرد. به طور مشخص، خروجی متد GetFutureWorks یک لیست از این کلاس است.

کلاس Article: این کلاس ساختار داده پارساها است که یک نمای خلاصه از جدول Articles در پایگاه داده اصلی ایرانداک است.

کلاس FutureWork: این کلاس یک ساختار داده برای متون استخراج شده از تمام‌متن‌ها است و رابطه‌ای یک به یک با کلاس Article دارد.

پایگاه داده و جدول‌ها

در این پژوهش از یک ماژول استخراج پیشنهادهای آینده با زبان پایتون استفاده شده است که با اجرای این ماژول اطلاعات لازم از فایل تمام‌متن مدارک واکشی شده و در جدول FutureWorks در پایگاه داده ذخیره می‌شوند. این جدول شامل اقلام اطلاعاتی زیر است:

- ArticleId
- TextExtract

که ArticleId به عنوان کلید خارجی با جدول Article در پایگاه داده ارتباط یک به یک دارد.

توابع

در این بخش به توصیف توابع به کار رفته در برنامه تهیه شده پرداخته شده است.

تابع Convertnumbs: این تابع یکسان‌نویسی اعداد را انجام می‌دهد.

تابع edit_text: این تابع ویرایش‌های موردنیاز را روی متون فارسی زبان انجام می‌دهد.

تابع section_score: این تابع به اختصاص امتیاز به بخش‌های کاندید به عنوان بخش دربردارنده پیشنهادهای پژوهش یک پارسا می‌پردازد.

تابع is_sctn_match: این تابع به بررسی یک بخش از پارسای ورودی می‌پردازد و تعیین می‌کند که آیا این بخش می‌تواند یک بخش کاندید برای بخش دربردارنده پیشنهادهای پژوهش باشد یا خیر.

تابع ext_rsrch_drctn_txt: این تابع به طور یکتا بخش دربردارنده پیشنهادهای پژوهش را از میان بخش‌های کاندید شناسایی می‌کند.

تابع ext_rsrch_drctn_txt_TEX: این تابع همانند تابع ext_rsrch_drctn_txt است با این تفاوت که بر روی فایل‌های با فرمت tex. عمل می‌کند.

تابع divide_section_2_sntnc: این تابع متن متناظر با یک بخش را به مجموعه‌ای از جملات تقسیم می‌کند.

تابع add_neighbor: در روش ارائه شده برای شناسایی جملات دربردارنده پیشنهادهای پژوهش، علاوه بر ویژگی‌های متناظر با یک جمله، از ویژگی‌های جملات قبل و بعد از آن نیز برای تعیین دربردارنده پیشنهاد پژوهش بودن یا نبودن جمله فعلی استفاده می‌شود. این کار در برنامه ارائه شده توسط این تابع صورت می‌گیرد.

تابع train: آموزش دسته‌بندهای مورد استفاده در این پژوهش توسط این تابع صورت می‌گیرد.

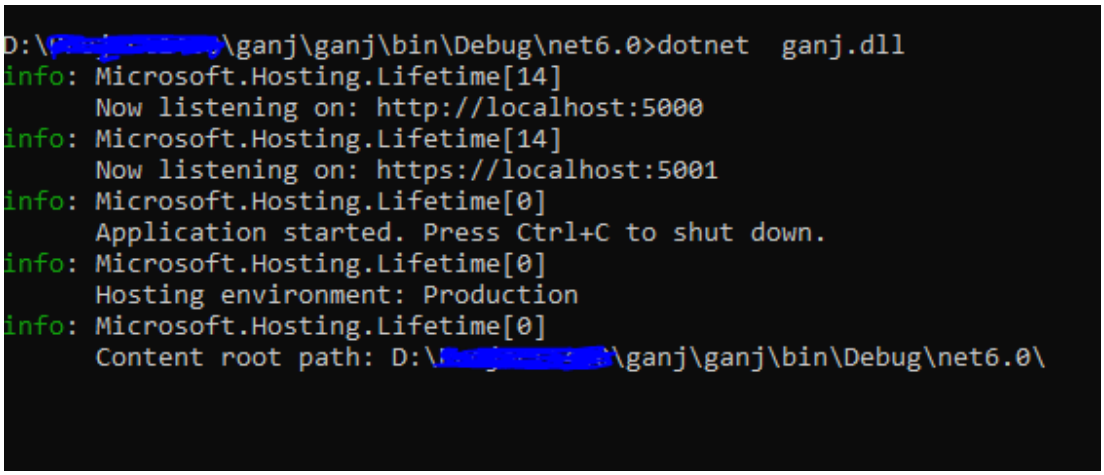
تابع Extract_features: این تابع، با ورودی یک جمله، بردار ویژگی متناظر با آن را جهت تعیین دربردارنده پیشنهاد پژوهش بودن یا نبودن آن جمله استخراج می‌نماید.

تابع ext_rsrch_drctn_frm_nw_doc: این تابع، با ورودی یک پارسا، پیشنهادهای پژوهش آن را استخراج کرده و به خروجی می‌برد.

راهنمای نصب و استقرار نرم‌افزار و راهنمای کاربران نرم‌افزار

برای اجرای برنامه تهیه شده در این پژوهش کافی است با استفاده از cmd یا powershell به پوشه مشخص شده در مسیر برنامه رفته و دستور زیر، فراخوانی شود:

dotnet Ganj2.dll



همانطور که در شکل مشخص است سامانه روی پورت ۵۰۰۰ با پروتکل http و روی پورت ۵۰۰۱ با پروتکل https در دسترس است. در صورت عمومی بودن آی‌پی سرور، سایت از بیرون نیز در دسترس خواهد بود.

فایل پیکربندی‌های سامانه با نام appsettings در root اصلی پوشه برنامه قرار دارد و مقادیری مانند Connection String در این فایل مقداردهی شده است.

پایگاه داده: این سامانه برای دسترسی به اطلاعات مدارک می‌تواند به یکی از پایگاه‌های Edit و یا Ganj متصل شود. به صورت پیش‌فرض، سامانه به پایگاه داده Edit متصل است.

Abstract

Future work suggestions, mentioned in the scientific documents, are very important and contain valuable information that can provide researchers new research topics or directions. Studying documents is the main way to identify these open problems. However, due to the ever-increasing growth of scientific documents, it is difficult to study all published documents and extract open problems. Considering this difficulty, especially for new researchers, the problem of automatic extraction of future work suggestions from the existing theses of Iran Scientific Information Database (Ganj) has been addressed in this research. To this end, at first, the existing solutions for this problem were reviewed, and then, by applying supervised machine learning algorithms, an automatic method for extracting future work suggestions from Persian theses is presented. In order to use the presented approach, a labeled dataset is needed which, due to the lack of such a data set for the Persian language, such a dataset is also created and used for training and testing the proposed method in this research. To implement the proposed method, various machine learning tools such as Naive Bayes, logistic regression, and support vector machine (SVM) have been employed. The experiment results show that the use of logistic regression and support vector machine tools has led to better results compared to the Naive Bayes.
At the end, the presented method has been implemented in the form of a software module and implemented as a new system called "Ganjineh". Ganjineh uses the data of "Iran Scientific Information Database" (Ganj) and provides the extracted future research directions to the users.

Keywords: Text mining; Open problems; Automatic extraction; Future works; Iranian Scientific Information Database.

Iranian Research Institute
for Information Science and Technology
(I r a n D o c)

Research Project

Design and implementation of a software module for automatic extraction of future research suggestions from theses of Iran Scientific Information Database (Ganj)

By:
Nasrollah Pakniat

Ali Asghar Hojjatpanah

Fall 2022