
بسم الله الرحمن الرحیم

وزارت علوم، تحقیقات و فناوری

پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)

طرح پژوهشی

طراحی یک روش هوشمند تجزیه رشته‌های مرجع در زبان فارسی

مجری

نصراله پاک‌نیت

همکاران

جلال‌الدین نصیری

عمار جلالی‌منش

بهمن ۱۳۹۷

به نام خدا

وزارت علوم تحقیقات و فناوری
پژوهشگاه علوم و فناوری اطلاعات ایران
ایرانداک

شماره: ۳۳/۱۴۱۰۹
تاریخ: ۱۳۹۷/۱۲/۰۱
پیوست: ندارد

پژوهش، مدیریت دانش، آموزش

گواهی

انجام طرح پژوهشی:

- **عنوان:** طراحی یک روش هوشمند تجزیه رشته‌های مرجع در زبان فارسی
- **مجری:** دکتر نصراله پاک‌نیت
- **همکاران:** دکتر جلال‌الدین نصیری، دکتر عمار جلالی‌منش
- **شمارهٔ قرارداد:** ۳۳/۱۳۱۲۶
- **تاریخ قرارداد:** ۹۶/۱۰/۴
- **تاریخ شروع:** ۹۶/۱۰/۴
- **تاریخ پایان:** ۹۷/۱۱/۱۴
- **ناظر:** دکتر آزاده محبی

در پژوهشگاه علوم و فناوری اطلاعات ایران گواهی می‌شود. گزارش پایانی این طرح، بر پایهٔ داوری در نشست ۳۲۱ (تاریخ ۹۷/۱۱/۳۰) شورای پژوهشی پژوهشگاه به تصویب رسیده است. لازم می‌دانم مراتب قدردانی خود را به مجری، همکاران، و ناظر این طرح تقدیم دارم.

با آرزوی بهروزی
پرویز شهریاری
معاون پژوهش و آموزش

تهران . خیابان انقلاب اسلامی . چهارراه فلسطین . شمارهٔ ۱۰۹۰ صندوق پستی ۱۳۷۱-۱۳۱۸۵
تلفن: ۶۶۴۹۴۹۸۰ - ۶۶۹۵۱۴۳۰ دورنگار: ۶۶۴۶۲۲۵۴ تلفن گویا: ۶۶۴۹۴۹۵۴
irandoc.ac.ir
info@irandoc.ac.ir

طراحی یک روش هوشمند تجزیه رشته‌های مرجع در زبان فارسی

مدیر طرح پژوهشی: نصراله پاک‌نیت، پژوهشگاه علوم و فناوری اطلاعات ایران.

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

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

وبگاه: http://irandoc.ac.ir/u/n-pakniat

رایانامه: pakniat@irandoc.ac.ir

همکاران: جلال‌الدین نصیری، عمار جلالی‌منش

چکیده

یک رشته مرجع را می‌توان به عنوان مجموعه‌ای از مولفه‌ها مانند نام نویسندگان، عنوان، محل نشر، سال نشر، شماره صفحات و ... در نظر گرفت. در حالیکه تجزیه رشته‌های مرجع موجود در انتهای یک مدرک علمی توسط کاربر انسانی به راحتی انجام‌پذیر است، تنوع موجود در فرمت‌های نگارش رشته‌های مرجع در کنار اشتباهات رخ داده توسط نویسندگان در نگارش این رشته‌ها، خودکارسازی انجام این عملیات را سخت کرده است. روش‌های زیادی برای خودکارسازی تجزیه رشته‌های مرجع ارائه شده است اما این روش‌ها وابسته به زبان بوده و امکان استفاده از یک روش ارائه شده برای یک زبان در زبانی دیگر منجر به نتایجی اشتباه می‌شود. تحقیقات صورت گرفته بیان‌گر این است که تاکنون هیچ روشی برای خودکارسازی تجزیه رشته‌های مرجع در زبان فارسی ارائه نشده است. با توجه به این مهم و نقش گسترده این مساله در ساخت خودکار شبکه‌های استنادی مدارک علمی و فرایندهای بازیابی اطلاعات، در این طرح پژوهشی به این مساله پرداخته شده است. در این راستا، پس از بررسی پیشینه مساله، از روش یادگیری ماشین بردار پشتیبان به عنوان یک دسته‌بند چند دسته‌ای استفاده شده و یک روش هوشمند برای مساله تجزیه رشته‌های مرجع در زبان فارسی ارائه شده است. با توجه به اهمیت انتخاب ویژگی‌های مناسب برای استفاده در دسته‌بند ماشین بردار پشتیبان، در این پژوهش این مهم با توجه به ویژگی‌های استفاده شده در زبان انگلیسی و ویژگی‌های زبان فارسی و ارجاع‌دهی در این زبان انجام شده است. روش ارائه شده پیاده‌سازی شده و با استفاده از مجموعه داده‌ای شامل ۹۲۲ رشته مرجع فارسی ایجاد شده در این پژوهش آموزش داده شده و مورد آزمایش قرار گرفته است. نتایج به دست آمده نشانگر مقدار ۹۲٪ برای پارامترهای دقت، فراخوانی و اف-۱ می‌باشد.

کلیدواژه‌ها: تجزیه رشته‌های مرجع، دسته‌بندی، دسته‌بندی چند دسته‌ای، ماشین بردار پشتیبان، ساخت خودکار شبکه‌های استنادی.

فهرست مطالب

عنوان **شماره صفحه**

فهرست جدول‌ها

عنوان **شماره صفحه**

فهرست شکل‌ها

عنوان **شماره صفحه**

قدردانی

اکنون‌که به یاری خدا طرح "طراحی یک روش هوشمند تجزیه رشته‌های مرجع در زبان فارسی" به پایان رسیده است، بر خود لازم می‌دانم:

از سرکار خانم دکتر آزاده محبی که علاوه بر پیشنهاد موضوع طرح، زحمت نظارت بر این طرح را نیز بر عهده داشته‌اند،

و همچنین سایر همکاران پژوهشگاه من جمله جناب آقای دکتر سیروس علیدوستی، ریاست محترم پژوهشگاه، و جناب آقای دکتر سید مهدی سمائی، ریاست محترم پژوهشکده علوم اطلاعات، به پاس تمام پشتیبانی‌های صورت گرفته،

کمال تشکر را داشته باشم.

فصل اول

مقدمه و اهداف

۱-۱- مقدمه

ارجاعات، که در مدارک علمی به وسیله رشته‌های مرجع (معمولا) در انتهای مدارک آورده می‌شوند، نقش مهمی در کتابخانه‌های دیجیتالی انتشار علمی مانند CiteSeer، arXiv e-Print، DBLP، Google Scholar و Scopus ایفا می‌کنند. علاوه بر استفاده از ارجاعات به عنوان ابزاری برای یافتن اطلاعات مورد علاقه، از این داده‌ها به عنوان معیاری برای بررسی تأثیر و اهمیت یک مقاله مشخص نیز استفاده می‌شود. در سطحی بالاتر، از میزان ارجاعات انجام شده به تحقیقات پژوهشگران به عنوان معیاری برای بررسی صلاحیت آن‌ها برای ارتقا، اعطای پژوهانه و پروژه‌های تحقیقاتی استفاده می‌شود. علاوه بر موارد فوق، از ارجاعات به عنوان ابزاری کمکی در فرایندهای بازیابی اطلاعات مانند خوشه‌بندی خودکار مدارک و نمایه‌سازی نیز استفاده می‌شود. مسأله اصلی در تمام کاربردهای فوق، ارائه روشی جهت تجزیه رشته‌های مرجع به اقلام اطلاعاتی اصلی تشکیل‌دهنده آن‌ها مانند نام نویسندگان، عنوان، محل نشر، تاریخ و شماره صفحات است که این مساله "تجزیه رشته‌های مرجع" نامیده می‌شود. اگرچه تجزیه رشته‌های مرجع توسط کاربر انسانی به سادگی انجام‌پذیر است اما به دلایلی مانند اشتباهات صورت گرفته در وارد کردن داده‌ها، فرمت‌های متفاوت به کار رفته در نگارش رشته‌های مرجع و عدم وجود استانداردی یکتا بدین منظور، اشکالات موجود در نرم‌افزارهای جمع‌آوری مراجع، به کار بردن اختصارات در نوشتن اسامی و محل‌های نشر، حذف برخی از قسمت‌های نام نویسندگان در اسامی چند بخشی و حجم زیاد داده‌ها، خودکارسازی این مسأله به سادگی امکان‌پذیر نیست. برای درک بیشتر مسأله و هدف آن، رشته مرجع زیر را در نظر بگیرید:

Pakniat, N., Noroozi, M., Eslami, Z.: ‘Secret image sharing scheme with hierarchical threshold access structure’, J. Vis. Commun. Image Represent., 2014, 25, (5), pp. 1093–1101

همان‌طور که مشخص است بخش‌های اصلی در این رشته مرجع به صورت زیر هستند:

Authors: Pakniat, N., Noroozi, M., Eslami, Z.

Title: Secret image sharing scheme with hierarchical threshold access structure

Publication Venue: J. Vis. Commun. Image Represent

Year of Publication: 2014

Volume Number: 25

Issue Number: 5

Page Numbers: 1093—1101.

ارجاع به این منبع همیشه یکسان نبوده و برای مثال به این منبع به فرمت‌های زیر نیز ارجاع داده شده است:

Pakniat N, Noroozi M, Eslami Z (2014) Secret image sharing scheme with hierarchical threshold access structure. J Visual Commun Image Represent 25(5):1093–1101

N. Pakniat, M. Noroozi, Z. Eslami, Secret image sharing scheme with hierarchical threshold access structure, Journal of Visual Communication and Image Representation 25 (5) (2014) 1093 – 1101.

N. Pakniat, M. Noroozi, Z. Eslami, Secret image sharing scheme with hierarchical threshold access structure, J. Vis. Commun. Image Represent. 25 (2014) 1093–1101.

N. Pakniat, M. Noroozi, and Z. Eslami, “Secret image sharing scheme with hierarchical threshold access structure,” Journal of Visual Communication & Image Representation, vol. 25, no. 5, pp. 1093–1101, 2014.

Pakniat, N., Noroozi, M., Eslami, Z.: Secret image sharing scheme with hierarchical threshold access structure. J. Vis. Commun. Image represent. 25(5), 1093–1101 (2014(

N. Pakniat, M. Noroozi, and Z. Eslami, “Secret image sharing scheme with hierarchical threshold access structure,” Journal of Visual Communication and Image Representation, vol. 25, no. 5, pp. 1093 – 1101, 2014.

Pakniat N, Noroozi M, Eslami Z. Secret image sharing scheme with hierarchical threshold access structure. Journal of Visual Communication and Image Representation 2014; 25(5):1093 – 1101.

Pakniat, N., M. Noroozi and Z. Eslami, 2014. Secret Image Sharing Scheme with Hierarchical Threshold Access Structure, Journal of Visual Communication and Image Representation, 5(5): 1093-1101.

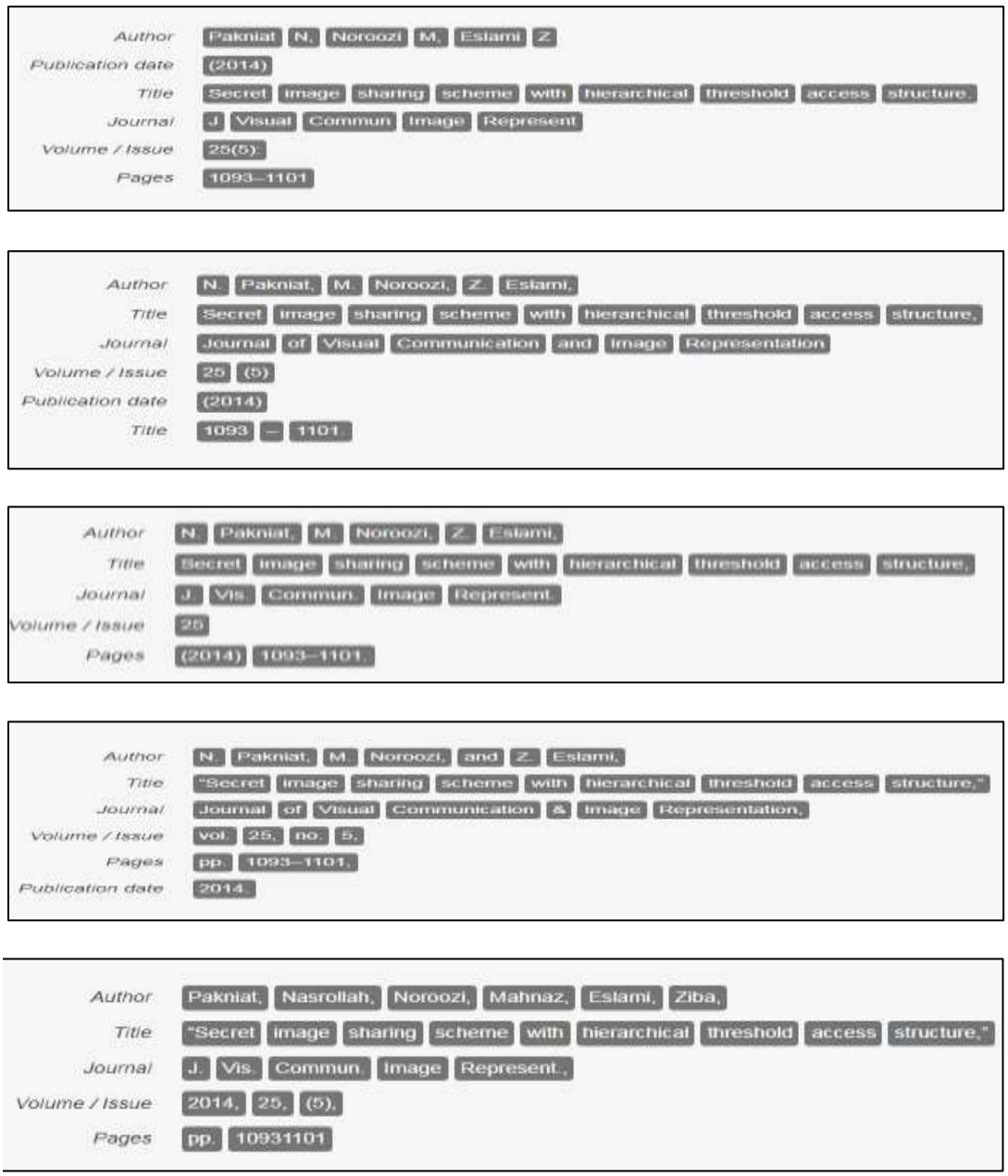
Pakniat, Nasrollah, Noroozi, Mahnaz, Eslami, Ziba, “Secret image sharing scheme with hierarchical threshold access structure,” J. Vis. Commun. Image Represent., 2014, 25, (5), pp. 10931101

N. Pakniat, M. Noroozi, and Z. Eslami, “Secret image sharing scheme with hierarchical threshold access structure,” Journal of Visual Communication and Image Representation, vol. 25, no. 5, pp. 1093–1101, 2014

N. Pakniat, M. Noroozi, Z. Eslami, “Secret image sharing scheme with hierarchical threshold access structure,” Journal of Visual Communication and Image Representation, 25, pp. 1093-1101, 2014.

Pakniat N, Noroozi M, Eslami Z (2014) Secret image sharing scheme with hierarchical threshold access structure. J Vis Commun Image Represent 25(5):1093–1101

الگوریتم‌ها و سامانه‌های زیادی برای تجزیه یک رشته مرجع در زبان انگلیسی ارائه شده که هر یک به نحوی قابل قبول قادر به تجزیه یک رشته مرجع ورودی هستند. برای مثال، در شکل ۱ برخی از رشته‌های مرجع فوق را به سامانه Freecite داده و نتایج به دست آمده را مشاهده خواهیم کرد:



شکل ۱. تجزیه به دست آمده برای برخی فرمت‌های ارجاع به رشته مرجع موردنظر با استفاده از سامانه Freecite.

همان‌طور که مشاهده می‌شود، در تمام موارد، این سامانه تقریبا به طور صحیح قادر به تشخیص اجزای این رشته مرجع است.

روش‌های تجزیه رشته‌های مرجع، وابسته به زبان بوده و استفاده از یک روش ارائه شده برای یک زبان در زبانی دیگر منجر به نتایج غالبا اشتباه می‌شود. در نتیجه، نمی‌توان از روش‌های موجود برای تجزیه

رشته‌های مرجع در سایر زبان‌ها، در زبان فارسی استفاده نمود. در ادامه با ذکر دو مثال این مسأله را نشان می‌دهیم. دو رشته مرجع زیر را در نظر بگیرید:

ایجیان اصلی،مهرداد،(۱۳۸۳)،رهیافتی نو به بنیانهای نظری پیشگیری از جرم،مجله حقوقی دادگستری،شماره۴۸-۴۹

حیدرنیا،نیلوفر؛ رحیمی،افسانه؛ احمدی موحد، اصغر، (۱۳۹۱ (، مجازات زندان و تاثیر آن بر پیشگیری از تکـرار جـرم ، حقوق جزا و جرمشناسی، پاییز و زمستان ۱۳۹۱ ، شماره ۲- : ۷۵ ۹۲.

نتایج تجزیه دو رشته مرجع فوق توسط سامانه FreeCite در شکل ۲ آورده شده است.

شکل ۲. تجزیه به دست آمده با استفاده از سامانه Freecite برای رشته‌های مرجع موردنظر.

همان‌طور که مشاهده می‌شود نتایج به دست آمده کاملا نادرست است. از جمله دلایل به وجود آمدن این خطاها می‌توان به موارد زیر اشاره کرد:

- برخی ویژگی‌های قابل استفاده در تجزیه یک رشته مرجع در یک زبان در زبانی دیگر معادل ندارند. برای مثال یکی از ویژگی‌های استفاده شده در اغلب روش‌های تجزیه خودکار رشته‌های مرجع زبان انگلیسی استفاده از حروف بزرگ و کوچک می‌باشد. در حالیکه چنین چیزی در زبان فارسی وجود نداشته و در نتیجه نمی‌توان از آن استفاده کرد.

- برخی از ویژگی‌های به کار رفته برای تجزیه رشته‌های مرجع در یک زبان، برای استفاده در زبانی دیگر باید بومی‌سازی شوند. برای مثال، یکی از ویژگی‌های در نظر گرفته شده در اغلب روش‌های تجزیه رشته‌های مرجع در زبان انگلیسی استفاده از کلماتی متداول مانند .j، journal، و ... برای تشخیص قلم اطلاعاتی محل نشر است. با این حال، استفاده از این کلمات در زبان فارسی مناسب نبوده و برای این زبان باید از کلماتی مثل مجله، فصلنامه، و ... بدین منظور استفاده کرد.

با توجه به اهمیت تجزیه خودکار رشته‌های مرجع در کاربردهایی مانند نمایه‌سازی خودکار و تحلیل استنادی و همچنین نقش پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک) در نگهداری، نمایه‌سازی و ارائه دانش افزوده از مدارک علمی کشور، در این طرح پژوهشی به مسأله تجزیه خودکار رشته‌های مرجع در زبان فارسی می‌پردازیم. در این راستا، در این طرح پژوهشی به بررسی روش‌های ارائه شده برای تجزیه رشته‌های مرجع در زبان انگلیسی پرداخته شده و سپس، با در نظر گرفتن ویژگی‌های به کار رفته برای تشخیص اقلام اطلاعاتی مختلف تشکیل‌دهنده رشته‌های مرجع در زبان انگلیسی و همچنین ویژگی‌های زبان فارسی و مراجع در زبان فارسی، این ویژگی‌ها را بومی‌سازی کرده و با استفاده از روش‌های یادگیری ماشین، روشی هوشمند جهت تجزیه خودکار رشته‌های مرجع نوشته شده به زبان فارسی و استخراج اقلام اطلاعاتی مختلف آن‌ها ارائه می‌کنیم. در ادامه روش ارائه شده را پیاده‌سازی کرده و کیفیت آن را بررسی خواهیم کرد. برای استفاده از روش‌های یادگیری ماشین در حل مسأله تجزیه رشته‌های مرجع به مجموعه‌ای برچسب‌گذاری شده از رشته‌های مرجع به زبان فارسی، نیاز داریم که با توجه به عدم وجود چنین مجموعه‌ای، به عنوان بخشی دیگر از مشارکت این طرح، چنین مجموعه‌ای را ایجاد کرده و از آن در پیاده‌سازی روش پیشنهادی استفاده خواهیم کرد.

۱-۲- اهداف طرح

هدف این طرح پژوهشی عبارت است از ارائه روشی هوشمند جهت تجزیه خودکار رشته‌های مرجع در زبان فارسی که می‌توان آن را به هدف‌های جزئی زیر تقسیم‌بندی نمود:

- بررسی روش‌های ارائه شده برای تجزیه رشته‌های مرجع در زبان انگلیسی.
- تعیین ویژگی‌های نوشتاری قابل استفاده در تجزیه رشته‌های مرجع در زبان فارسی.

- ارائه یک روش هوشمند جهت تجزیه خودکار رشته‌های مرجع در زبان فارسی با استفاده از روش‌های یادگیری ماشین.
- ایجاد یک مجموعه داده جهت استفاده به عنوان داده آموزش و آزمایش در روش پیشنهادی.
- پیاده‌سازی روش پیشنهادی و انجام آزمایشات لازم جهت بررسی کیفیت روش پیشنهادی.

۱-۳- ساختار این طرح پژوهشی

ادامه این طرح پژوهشی به شرح زیر می‌باشد: در فصل دوم به بررسی روش‌های ارائه شده برای تجزیه رشته‌های مرجع در زبان انگلیسی پرداخته خواهد شد. با توجه به نتایج به دست آمده از این فصل، استفاده از روش‌های یادگیری ماشین من‌جمله روش ماشین بردار پشتیبان منجر به دستیابی به نتایج بسیار خوبی در حل این مساله شده است. با توجه به این مهم، در فصل سوم، روش ماشین بردار پشتیبان مورد مطالعه قرار خواهد گرفت. در ادامه، جزئیات روش ارائه شده برای مساله تجزیه رشته‌های مرجع در زبان فارسی و ارزیابی آن در فصل چهارم ارائه شده است. در نهایت، نتیجه‌گیری و کارهای آینده در فصل پنجم ذکر شده است.

فصل دوم

بررسی روش‌های ارائه شده برای تجزیه رشته‌های مرجع در زبان انگلیسی

در این فصل به بررسی روش‌های موجود برای تجزیه رشته‌های مرجع در زبان انگلیسی پرداخته می‌شود. با توجه به ادبیات مسأله، روش‌های ارائه شده برای این مسأله را می‌توان به دو دسته زیر تقسیم‌بندی کرد:

- روش‌های مبتنی بر قاعده[۱]،
- روش‌های مبتنی بر یادگیری ماشین[۲].

تحقیقات صورت گرفته نشان‌دهنده این است که استفاده از روش‌های مبتنی بر یادگیری ماشین در حالت کلی منجر به نتایج بسیار بهتری شده و روش‌های مبتنی بر قاعده تنها در موقعیت‌هایی که بدانیم رشته‌های مرجع مورد آزمایش از مجموعه‌ای مشخص از قالب[۳]ها تبعیت کرده و با استفاده از این قالب‌ها ایجاد شده‌اند، کارا هستند. در ادامه این فصل، روش‌های ارائه شده در هر یک از دو دسته مورد نظر را بررسی خواهیم کرد.

۲-۱- روش‌های هوشمند تجزیه رشته‌های مرجع مبتنی بر قاعده

در این بخش، به طور خلاصه، به بررسی روش‌هایی که با استفاده از قواعد (اکتشافی) به تجزیه رشته‌های مرجع می‌پردازند، خواهیم پرداخت.

روش ارائه شده توسط لاورنس[۴] و همکاران

مسأله بررسی شده در (Lawrence et al., 1999a) و (Lawrence et al., 1999b) تطبیق خودکار رشته‌های مرجع بوده که کلی‌تر از مسأله تجزیه رشته‌های مرجع است اما (Lawrence et al., 1999a)

[1] Rule based methods
[2] Machine learning based methods
[3] Template
[4] Lawrence

و (Lawrence et al., 1999b) از اولین مقالاتی هستند که در آن به مسأله تجزیه خودکار رشته‌های مرجع اشاره شده است.

در این مقاله بیان شده که به طور کلی می‌توان از ۴ دسته روش زیر برای تطبیق خودکار رشته‌های مرجع استفاده کرد:

- استفاده از معیارهای فاصله ویرایش[۱] مانند فاصله لون‌اشتاین[۲] یا لایک‌ایت[۳].
- استفاده از معیارهای فراوانی کلمات[۴] مانند TF-IDF[۵].
- استفاده از اقلام اطلاعاتی[۶] مانند عنوان، سال انتشار و ... .
- آموزش مدل‌هایی احتمالاتی[۷] با استفاده از اطلاعات کتاب‌شناختی[۸] جهت تشخیص خودکار اقلام اطلاعاتی یا به عبارتی تجزیه خودکار رشته‌های مرجع و در ادامه استفاده از روش‌های قبل برای تطبیق رشته‌های مرجع.

لازم به ذکر است که در (Lawrence et al., 1999a) و (Lawrence et al., 1999b) ، تنها الگوریتم‌هایی برای سه دسته اول ارائه شده که با توجه به خارج بودن از حیطه این طرح پژوهشی، در اینجا به آن‌ها پرداخته نخواهد شد. خوانندگان گرامی برای جزئیات بیشتر در مورد این الگوریتم‌ها می‌توانند به (Lawrence et al., 1999a) و (Lawrence et al., 1999b) مراجعه کنند.

روش ارائه شده توسط بساگنی[۹] و همکاران

در (Besagni et al., 2003)، نویسندگان به بررسی مسأله تجزیه رشته‌های مرجع در مقالات اسکن شده پرداخته‌اند. بدین منظور، در ابتدا از روش‌های تشخیص کاراکتر نوری[۱۰] (OCR) استفاده شده و تصاویر مقاله دریافتی به متن تبدیل شده و سپس تجزیه رشته‌های مرجع در فایل‌های به دست آمده انجام شده است. با توجه به اضافه شدن خطاهای ناشی از OCR به سایر سختی‌های موجود برای مسأله،

---

[1] Edit distance measures
[2] Levenshtein
[3] LikeIt
[4] Word frequency measures
[5] Term Frequency–Inverse Document Frequency
[6] Field
[7] Probabilistic models
[8] Bibliographical information
[9] Besagni
[10] Optical character recognition

تجزیه رشته‌های مرجع در مقالات اسکن شده سخت‌تر از مقالات با فرمت متنی است. جزئیات روش ارائه شده در (Besagni et al., 2003)، که از سه فاز برچسب‌گذاری اولیه، تحلیل نحوی و تحلیل ساختاری تشکیل شده، به شرح زیر می‌باشد:

- برچسب‌گذاری اولیه: در ابتدا هر رشته مرجع به مجموعه‌ای از واحدهای متنی تقسیم‌بندی می‌شود. با توجه به ویژگی‌های متنی هر واحد مانند تنها شامل عدد یا حرف بودن واحد، آغاز واحد با حروف بزرگ یا کوچک، وجود کلمات مشخص مانند .et al، .In در واحد، وجود کلمات مشخص‌کننده نام نشریه، علائم نشان‌گذاری در واحد و ...، از مجموعه‌ای از برچسب‌ها استفاده شده و به واحدهای متن برچسب‌های ممکن اختصاص داده می‌شود.
- تحلیل نحوی: در این فاز، با انجام دو فرایند زیر بر روی هر رشته مرجع، برچسب‌های اختصاص‌یافته به واحدهای مختلف متن مورد بازنگری قرار گرفته و اصلاحات لازم انجام می‌شود:
  - با توجه به برچسب‌های اختصاص‌یافته، جستجو در میان مراجع یک مقاله برای وجود الگوهای منظم در مورد برخی اقلام اطلاعاتی مانند مشخص‌کننده رشته مرجع[1]، تاریخ و شماره صفحه انجام می‌شود. در نتیجه این فرایند، موقعیت برخی اقلام اطلاعاتی در رشته‌های مرجع به راحتی مشخص می‌شود و برخی اصلاحات مورد نیاز انجام می‌شود.
  - با استفاده از قواعدی اکتشافی (حاصل از سعی و خطا)، واحدهای مجاور و برچسب‌های اختصاص یافته به آن‌ها با یکدیگر تجمیع می‌شوند. تجمیع دو قسمت متوالی و آغازین نام به عنوان یک قسمت، تجمیع یک قسمت آغازین نام و یک نام احتمالی به عنوان یک نام، تجمیع نام نویسندگان و .et al به عنوان نام نویسندگان از جمله قواعد استفاده شده در این فرایند هستند.
- تحلیل ساختاری: فاز قبل یعنی تحلیل نحوی به دلایلی مانند ناشناخته ماندن برچسب برخی واحدها، ساختار پیچیده عناوین و تشابهات موجود مابین تاریخ و شماره صفحه، دارای محدودیت‌هایی در جداسازی اقلام اطلاعاتی است. ایده این فاز استفاده از تشخیص‌های درست به عنوان مدلی برای تشخیص برچسب بقیه واحدها است. در این مرحله، از دو مدل استفاده شده است که عبارتند از:

[1] Reference identifier

- مدل بین اقلام اطلاعاتی: در این قسمت با محاسبه فراوانی تکرار حضور متوالی برچسب‌ها، محتمل‌ترین برچسب برای واحدهایی که برچسب آن‌ها نامشخص است در نظر گرفته می‌شود. علاوه‌براین، برای واحدهایی که اختصاص هیچ برچسبی به آن‌ها ممکن نیست از برچسب واحد سمت راست یا سمت چپ آن‌ها استفاده می‌شود.
- مدل درون قلم اطلاعاتی: در این قسمت، ساختار اقلام اطلاعاتی و واحدهای موجود در اقلام اطلاعاتی مشخص می‌شود.

برای بررسی کیفیت روش پیشنهادی، نویسندگان مجموعه داده‌ای جهت آزمایش روش پیشنهادی ایجاد کرده و با استفاده از آن روش ارائه شده را مورد ارزیابی قرار داده‌اند. نتایج ارزیابی نویسندگان بدین صورت است که روش ارائه شده در ۷۵٫۹٪ به درستی، در ۱۸٫۸٪ به صورت ناقص و در ۵٫۳٪ کاملا اشتباه برچسب اقلام اطلاعاتی را تشخیص داده است.

روش ارائه شده توسط دینگ[1] و همکاران

هدف نویسندگان در (Ding et al., 2003)، استخراج اطلاعات ارجاعی، چه در مورد مقالات ارجاع‌شده[2] و چه در مورد مقاله ارجاع‌دهنده[3] است. در این مقاله، از مجموعه‌ای از مقالات چند نشریه که هم به صورت برخط[4] و هم به صورت چاپی[5] منتشر می‌شوند و چند نشریه که تنها به صورت برخط منتشر می‌شوند به عنوان داده مورد بررسی برای استخراج قالب‌ها و بررسی کیفیت روش نهایی استفاده شده است.

برای به دست آوردن اطلاعات مقاله ارجاع‌دهنده جدولی از سرنخ‌ها برای جستجوی اقلام اطلاعاتی موردنظر در مقالات تهیه شده که بر اساس این جدول، اقلام اطلاعاتی در مقالات جدید استخراج می‌شوند.

برای تجزیه رشته‌های مرجع موجود در یک مقاله نیز قواعدی کلی استخراج شده که مهم‌ترین آن‌ها به شرح زیر است:

---

[1] Ding
[2] Cited papers
[3] Citing paper
[4] Online
[5] Print

- نام نویسندگان به شکل‌های مختلف بیان می‌شود. اما در بررسی‌های صورت گرفته، در عمده ارجاعات نام یا نام خانوادگی یک نویسنده در ابتدای رشته آمده است.
- زمان انتشار ممکن است در قسمت‌های مختلفی از یک رشته مرجع واقع شده باشد. اما معمولا یک عدد چهاررقمی که با ۱۹ یا ۲۰ شروع شده، نشان‌دهنده زمان انتشار یک مرجع است.
- محل نشر به طور معمول همراه با کلیدواژه‌هایی مشخص که در این قلم اطلاعاتی استفاده می‌شود، می‌آید. در نتیجه با استفاده از کلیدواژه‌هایی مانند journal، proceeding، conference، report، .ed، و ... می‌توان با دقت خوبی این قلم اطلاعاتی را در رشته‌های مرجع مشخص کرد. درباره قلم اطلاعاتی محل نشر، با استناد به اینکه بیش از ۵۵٪ از ارجاعات به تنها ۱۰٪ از عناوین محل‌های نشر انجام می‌شود (Vinkler, 1994)، نویسندگان بیان کرده‌اند که با تهیه لیستی از نشریات و همایش‌های معروف می‌توان دقت را در تشخیص اطلاعات این قلم اطلاعاتی در یک رشته مرجع افزایش داد.

آزمایشات انجام شده بر روی داده‌های آزمایش، نشان داده که روش ارائه شده توسط نویسندگان در تجزیه رشته‌های مرجع نشریات با چاپ کاغذی و برخط نتایج خوبی به دست آورده اما برای نشریاتی که تنها به صورت برخط چاپ می‌شوند، نتایج ضعیفی حاصل شده است. نویسندگان ضمن بیان چندقالبی بودن ارجاعات به عنوان دلیلی بر ضعیف بودن نتایج آزمایشات در نشریات برخط، بیان کرده‌اند که با افزایش داده‌های مورد بررسی این مشکل برطرف خواهد شد.

روش ارائه شده توسط هوانگ[1] و همکاران

در (Huang et al., 2004) روشی جهت تجزیه رشته‌های مرجع ارائه شده که در آن در ابتدا، پایگاه داده‌ای شامل قالب‌های شناخته شده از رشته‌های مرجع ایجاد شده و سپس تجزیه یک رشته مرجع با استفاده از این پایگاه داده و یکی از روش‌های هم‌ترازسازی توالی[2] به نام BLAST انجام می‌شود. جزئیات دقیق‌تر این الگوریتم که از دو بخش "فاز تولید قالب" و "فاز تجزیه رشته مرجع" تشکیل شده، به شرح زیر می‌باشد:

[1] Huang
[2] Sequence alignment

۱- فاز تولید قالب: در این بخش، در ابتدا با استفاده از یک خزنده وب[1]، مجموعه‌ای از فایل‌های bibtex به دست آمده و سپس، با استفاده از عنوان مقاله (که به عنوان یکی از اقلام اطلاعاتی فایل Bibtex وجود دارد) و جستجو در وب‌سایت citeseerx.ist.psu.edu رشته مرجع نیمه ساختار یافته متناظر با هر فایل bibtex به دست می‌آید. در ادامه با دسترسی به این دو شکل از هر رشته مرجع، قالب متناظر با هر رشته مرجع نیمه ساختار یافته از طریق فرایند زیر ایجاد می‌شود.

a. با استفاده از علائم فاصله‌گذاری و نشان‌گذاری، رشته مرجع نیمه ساختار یافته به دنباله‌ای از واحد[2]ها تقسیم‌بندی می‌شود. در ادامه، با توجه به دانستن اطلاعات کتاب‌شناختی این رشته مرجع با توجه به وجود فایل bibtex متناظر با آن، از مجموعه‌ای از قواعد استفاده شده، برچسب هر واحد مشخص شده، رشته مرجع مورد نظر به یک دنباله پروتئینی[3] تبدیل شده و در پایگاه داده ذخیره می‌شود.

۲- فاز تجزیه رشته مرجع: همانند آن‌چه در فاز قبل داشتیم، در این فاز نیز در ابتدا رشته مرجع ورودی به مجموعه‌ای از واحدها تقسیم‌بندی می‌شود. سپس از عبارات منظم[4] استفاده شده و به هر واحد برچسبی از میان انواع برچسب‌های ممکن اختصاص یافته و رشته مرجع به یک دنباله پروتئینی تبدیل می‌شود. در ادامه، از الگوریتم BLAST استفاده شده و شبیه‌ترین قالب از بین قالب‌های موجود در پایگاه داده به این دنباله پروتئینی به دست می‌آید. پس از یافتن شبیه‌ترین قالب، با استفاده از برچسب‌های موجود در قالب مشخص شده، برچسب نهایی هر واحد در رشته مرجع ورودی مشخص شده و عمل تجزیه انجام می‌شود.

برای بررسی کیفیت روش، نویسندگان پایگاه داده‌ای شامل ۲۵۰۰ قالب ایجاد کرده و توانسته‌اند دقت[5] ۸۹٪ را در تجزیه رشته‌های مرجع مورد آزمایش به دست آورند که پارامتر دقت بیان‌گر نسبت برچسب‌های به درستی اختصاص‌یافته در مقایسه با کل برچسب‌های اختصاص‌یافته است.

روش ارائه شده توسط چن[6] و همکاران

[1] Web crawler
[2] Token
[3] Protein sequence
[4] Regular expression
[5] precision
[6] Chen

در (Chen et al., 2008)، از هم‌ترازسازی توالی برای تجزیه رشته‌های مرجع استفاده شده است. روش ارائه شده در (Chen et al., 2008)، بهبودی از روش ارائه شده توسط نویسندگان در (Huang et al., 2004) است. در این روش، در ابتدا سعی می‌شود تا جای ممکن اطلاعات متنی حذف و اطلاعات ساختاری ذخیره شود و از الگوریتمی به نام الگوریتم کانونی‌سازی[1] برای تبدیل رشته‌های مرجع ورودی به یک شکل کانونی استفاده می‌شود که این الگوریتم از مراحل زیر تشکیل شده است:

- از یک جدول کدگذاری برای نگاشت واحدهای رشته‌های مرجع به نمادهایی در دنباله پروتئینی استفاده می‌شود. کدگذاری موردنظر به شکلی انجام می‌شود که ویژگی‌های ساختاری رشته مرجع ورودی حفظ شود.
- با استفاده از روش tf-idf مجموعه‌ای از کلمات رزرو شده برای اقلام اطلاعاتی مختلف رشته‌های مرجع مانند عنوان، محل چاپ، شماره صفحه و ... ایجاد می‌شود.
- با توجه به فراوانی حضور چندتایی‌ها و همچنین استفاده از دانش یک کاربر انسانی، الگوهای پرکاربرد در دنباله ایجاد شده برای رشته‌های مرجع یافته شده و از آن برای ساده‌سازی نمایش‌ها استفاده می‌شود.

روش ارائه شده برای تجزیه رشته‌های مرجع توسط چن و همکاران از ۲ فاز "ساخت پایگاه داده قالب‌ها" و "پردازش رشته‌های مرجع جدید" تشکیل شده است که در ادامه جزئیات این فازها بیان خواهد شد.

- ساخت پایگاه داده قالب‌ها: این فاز به مجموعه‌ای از رشته‌های مرجع برچسب‌گذاری شده نیاز دارد که برای ایجاد آن، نویسندگان از فایل‌های bibtex موجود برای ارجاع به مقالات انگلیسی و رشته مرجع نیمه ساختار یافته متناظر با آن‌ها (به دست آمده با استفاده از موتورهای جستجو) استفاده کرده‌اند. در ادامه، با توجه به وجود فایل‌های bibtex متناظر با رشته‌های مرجع و الگوریتم کانونی‌سازی، دو صورت متناظر با هر رشته مرجع موجود در پایگاه داده به نام صورت سبک نگارش[2] (با استفاده از فایل‌های bibtex) و صورت اندیس[3] (با استفاده از الگوریتم کانونی‌سازی) ایجاد و ذخیره می‌شود.

[1] Canonicalization algorithm
[2] Style form
[3] Index form

- پردازش رشته‌های مرجع جدید: در این فاز، با ورودی یک رشته مرجع جدید، در ابتدا از الگوریتم کانونی‌سازی استفاده شده و رشته مرجع ورودی به صورت اندیس تبدیل می‌شود. سپس از الگوریتم BLAST استفاده شده و در میان مجموعه صورت‌های اندیس موجود در پایگاه داده، نزدیک‌ترین صورت‌ها به صورت ورودی محاسبه می‌شود. در ادامه، برای جستجوی دقیق‌تر، از صورت سبک نگارش معادل صورت اندیس‌های به دست آمده استفاده شده و با استفاده از یک الگوریتم هم‌ترازسازی توالی دیگر به نام Needleman-Wunsch که به صورت دقیق‌تر و با جزئیات بیشتر هم‌ترازسازی را انجام می‌دهد، صورت سبک نگارش معادل رشته مرجع ورودی به دست می‌آید. سپس، با دسترسی به صورت سبک نگارش رشته مرجع ورودی، اقلام اطلاعاتی مختلف رشته مرجع به راحتی به دست خواهد آمد. لازم به ذکر است که در (Chen et al., 2008)، پس از محاسبه اقلام اطلاعاتی مختلف یک رشته مرجع، از پردازشی نهایی برای تقسیم هر قلم اطلاعاتی به اجزای سازنده آن استفاده می‌شود تا برای مثال زمان به ماه و سال تقسیم شده یا اسامی نویسندگان به صورت مجزا به دست آید.

- برای بررسی کیفیت، آزمایشاتی توسط نویسندگان بر روی روش ارائه شده انجام گرفته که نتایج حاصل از استفاده از روش اعتبارسنجی متقابل ۵تایی[1] روی مجموعه داده cora (Seymore et al., 1999) نشان‌دهنده مقادیر ۹۱٫۸۹ ٪ و ۹۳٫۸۱٪ برای پارامترهای دقت و فراخوانی در سطح واحد و دقت ۸۷٫۵۷٪ در سطح اقلام اطلاعاتی است که پارامتر فراخوانی بیان‌گر نسبت برچسب‌های اختصاص‌یافته به کل واحدهایی است که باید به آن‌ها برچسب اختصاص می‌یافت.

روش ارائه شده توسط گوپتا[2] و همکاران

در (Gupta et al., 2009)، در ابتدا با استفاده از قواعدی اکتشافی مانند وجود کلیدواژه‌های نمایشگر بخش مراجع، وجود پاراگراف‌های با طول کم متوالی و وجود عدد سال در هر پاراگراف، شروع بخش مراجع مشخص می‌شود. در ادامه، هر پاراگراف از باقیمانده متن با استفاده از معیارهای زیر (که می‌توانند اهمیت یکسان نداشته باشند) جهت تعیین رشته‌های مرجع مورد بررسی قرار می‌گیرد:

- وجود عدد مشخص‌کننده سال در پاراگراف.

[1] Five fold cross validation
[2] Gupta

- وجود کلیدواژه‌های مشخص در پاراگراف.
- همبستگی بین طول پاراگراف در دست بررسی با متوسط طول رشته‌های مرجع.
- تعداد رشته‌های مرجع تشخیص داده شده در بالای پاراگراف در دست بررسی.
- تعداد رشته‌های مرجع تشخیص داده شده در زیر پاراگراف در دست بررسی.
- وجود همبستگی بین قالب رشته‌های مرجع با پاراگراف در دست بررسی.

انجام روند فوق از این جهت مهم است که همیشه بخش مراجع در یک جای مشخص در یک مقاله قرار نگرفته و به دلایلی مانند وجود ضمیمه[1]، جدول و شکل ممکن است بخش مراجع، آخرین بخش مقاله نباشد.

پس از مشخص کردن رشته‌های مرجع موجود در یک مقاله، عمل تجزیه هر رشته مرجع با استفاده از مراحل زیر انجام می‌شود:

- استفاده از عبارات منظم برای اختصاص برخی اقلام اطلاعاتی به برخی قسمت‌های مختلف از هر رشته مرجع.
- استفاده از اطلاعاتی پایه‌ای در زمینه مورد نظر (مانند استفاده از لیستی از عناوین نشریات برای برچسب‌زنی به یک قسمت به عنوان نشریه) برای دسته‌بندی بخش‌های به جا مانده از مرحله قبل.

روش ارائه شده توسط کنستانتین[2] و همکاران

در (Constantin et al., 2013)، نویسندگان یک روش مبتنی بر قاعده برای بازسازی ساختار منطقی یک مقاله با دریافت فایل pdf آن به عنوان ورودی ارائه کرده‌اند. در این روش، ساختار منطقی مقاله شامل تشخیص اقلام اطلاعاتی عنوان، نویسنده، چکیده، کلیدواژه‌ها، بخش‌ها و زیربخش‌ها، رشته‌های مرجع و اقلام اطلاعاتی هر رشته مرجع با استفاده از قواعد به دست آمده از مجموعه‌ای از قالب‌ها، تغییرات فونت و طرح‌بندی[3] فایل pdf مقاله مشخص می‌شود. از آن‌جا که موضوع این مقاله کلی‌تر از مسأله تجزیه رشته‌های مرجع است، نویسندگان در مورد قواعد به کار برده شده در مورد مسأله مورد بحث در این طرح پژوهشی اطلاعاتی ارائه نکرده‌اند.

---

1 Appendix
2 Constantin
3 Layout

۲-۲- روش‌های هوشمند تجزیه رشته‌های مرجع مبتنی بر یادگیری ماشین

مسأله تجزیه رشته‌های مرجع را می‌توان به عنوان یک مسأله برچسب‌گذاری دنباله‌ای[۱] یا یک مسأله دسته‌بندی چند دسته‌ای[۲] در نظر گرفت. با توجه به کارایی مناسب روش‌های یادگیری ماشین در حل مسائل برچسب‌گذاری دنباله‌ای و دسته‌بندی چند دسته‌ای، در سالیان گذشته به طور گسترده از روش‌های یادگیری ماشین برای حل مسأله تجزیه رشته‌های مرجع نیز استفاده شده است. روش‌های یادگیری ماشین را می‌توان به دو دسته روش‌های با نظارت[۳] و روش‌های بدون نظارت[۴] تقسیم‌بندی نمود. بررسی پیشینه مسأله بیان‌گر این است که تنها یک روش تجزیه رشته مرجع با استفاده از روش‌های یادگیری بدون نظارت ارائه شده است و سایر روش‌های تجزیه رشته مرجع مبتنی بر یادگیری ماشین از نوع با نظارت این روش‌ها استفاده می‌کنند. علاوه بر این، بر اساس ابزار به کار برده شده، روش‌های تجزیه رشته‌های مرجع مبتنی بر یادگیری با نظارت را می‌توان به سه زیر دسته روش‌های مبتنی بر مدل مخفی مارکوف[۵]، روش‌های مبتنی بر میدان‌های تصادفی شرطی[۶]، و روش‌های مبتنی بر ماشین‌های بردار پشتیبان[۷] تقسیم‌بندی کرد. در ادامه این بخش، در ابتدا روش‌های ارائه شده در این دسته‌ها و سپس تنها روش ارائه شده با استفاده از روش‌های یادگیری ماشین بدون نظارت را بررسی خواهیم کرد. لازم به ذکر است که نتایج بررسی‌های انجام شده در این بخش نشان‌دهنده این است که استفاده از یادگیری با نظارت برای حل مسأله تجزیه رشته‌های مرجع منجر به نتایج بسیار بهتری شده و در استفاده از روش‌های یادگیری ماشین با نظارت نیز هر سه ابزار مورد استفاده نتایجی تقریبا مشابه و قابل قبول ارائه کرده‌اند.

۲-۲-۱- تجزیه رشته‌های مرجع با استفاده از مدل مخفی مارکوف[۸] (HMM)

یک مدل مخفی مارکوف از چهارتایی (S, A, V, B) تشکیل شده است که $S = \{S_1, \dots, S_N\}$ مجموعه حالت‌ها و $V = \{V_1, \dots, V_M\}$ مجموعه نمادهای emission، A ماتریس احتمالات انتقالی و B ماتریس

[1] Sequence labling
[2] Multi class classifier
[3] Supervised
[4] Unsupervised
[5] Hidden Markov Model
[6] Conditional Random Field
[7] Supported Vector Machine
[8] Hidden Markov Model

احتمالات emission بوده و حل یک مسأله با استفاده از HMM به معنی مشخص کردن این پارامترها است.

در سالیان گذشته سه روش مبتنی بر مدل مخفی مارکوف برای حل مسأله تجزیه رشته‌های مرجع ارائه شده است. این روش‌ها، مسأله تجزیه رشته مرجع را به عنوان یک مساله برچسب‌زنی دنباله‌ای در نظر گرفته و از HMM برای حل آن استفاده کرده‌اند. با توجه به بررسی‌های صورت گرفته، تفاوت روش‌های موجود در این دسته، در نوع مدل مخفی مارکوف به کار برده شده برای حل مسأله می‌باشد و نتایج نشان می‌دهد که استفاده از مدل‌های مخفی مرتبه بالاتر منجر به دستیابی به نتایجی بهتر شده است. در ادامه این بخش، روش‌های موجود در این دسته را بررسی خواهیم کرد.

روش ارائه شده توسط هتزنر[1]

در یک مدل HMM اولیه منطقی برای حل مساله تجزیه رشته مرجع، یک حالت به ازای هر قلم اطلاعاتی در مجموعه حالت‌های مدل در نظر گرفته می‌شود و اجازه انتقال از هر حالتی به هر حالت دیگری داده می‌شود. اما این مدل در همه حالت‌ها بهینه نبوده و در زمانی که مسأله‌ای که قصد حل آن را داریم دارای یک ساختار دنباله‌ای مخفی باشد، مدلی که به ازای هر قلم اطلاعاتی چند برچسب در نظر گرفته و در آن تنها تعداد کمی انتقال ممکن از هر حالت وجود داشته باشد، بهتر عمل می‌کند. در (Hetzner, 2008) (همانند دو روش دیگر بررسی شده در این بخش)، نویسندگان بیان کرده‌اند که در نظر گرفتن دو حالت به ازای هر قلم اطلاعاتی در مجموعه حالت‌ها منجر به دستیابی به نتایج بهتری در مقایسه با سایر تقسیم‌بندی‌ها شده است. علاوه بر حالات ذکر شده، در این روش (همانند دیگر روش‌های بازبینی شده در این بخش)، دو حالت شروع و پایان نیز در مجموعه حالات در نظر گرفته شده است.

در این روش، از علائم نشان‌گذاری مانند"،"، "."، ":" و ... ، کلمات خاص مانند proceeding، proc.، press، university و ...، نمادهایی مشخص برای نشان دادن مجموعه‌هایی از کلمات مرتبط به یکدیگر مثل ماه‌های سال و مجموعه‌ای از نمادها برای نمایش ویژگی‌های کلماتی که در دسته‌های قبل جای نگرفته‌اند به عنوان مجموعه الفبای مساله V استفاده شده است. از جمله ویژگی‌های مورد نظر

[1] Hetzner

می‌توان به مواردی مانند کلمه‌ای با حروف کوچک، کلمه‌ای با حروف بزرگ، عددی چهار رقمی و ... اشاره کرد.

برای محاسبه احتمالات انتقال و emission، در این روش، ابتدا HMMای ساخته شده که همه رشته‌های مرجع موجود در مجموعه آموزش را تولید کند. به عبارت دیگر، در این روش، حالت شروع به ازای هر رشته موجود در مجموعه آموزش دارای یک خروجی بوده و به ازای هر رشته یک مسیر خروجی به حالت پایان وجود دارد. احتمال رفتن از حالت شروع به هر یک از مسیرها مساوی در نظر گرفته شده و احتمال emission از سایر حالت‌ها برابر با یک در نظر گرفته می‌شود. به این مدل، مدل maximum likelihood گفته می‌شود. در ادامه با استفاده از دو تکنیک Neighbor merging و V merging برخی از حالت‌های با برچسب یکسان ادغام شده تا مدلی به غایت مشخص[1] ایجاد شود. Neighbor merging هر دو حالتی با برچسب اقلام اطلاعاتی یکسان را که لینکی بین آن‌ها موجود باشد و V merging هر دو حالت با برچسب اقلام اطلاعاتی یکسان را که از یک حالت دیگر به آن‌ها لینک وجود داشته یا به یک حالت دیگر لینک داشته باشند، با یکدیگر ادغام می‌کند.

در ادامه احتمالات انتقال $P(q \to q')$ و احتمالات emission $P(q \uparrow \sigma)$ به صورت زیر محاسبه می‌شوند:

$$P(q \to q') = \frac{c(q \to q')}{\sum_{s \in Q} c(q \to s)} \qquad (۱-۲)$$

و

$$P(q \uparrow \sigma) = \frac{c(q \uparrow \sigma)}{\sum_{\rho \in \Sigma} c(q \uparrow \rho)}، \qquad (۲-۲)$$

که $q \to q'$ انتقال از حالت q به حالت 'q، و $q \uparrow \sigma$ نشان‌دهنده این است که حالت q نماد σ را emit کند و $c(x)$ نشان‌دهنده تعداد وقوع رخداد x در مجموعه داده آموزش است.

پس از به دست آوردن پارامترهای HMM، برای تجزیه یک رشته مرجع جدید، در ابتدا رشته ورودی به دنباله‌ای از واحدها تقسیم شده، سپس دنباله نمادهای متناظر با دنباله واحدها به دست آمده و از HMM طراحی شده برای به دست آوردن قلم اطلاعاتی (برچسب) متناظر با هر واحد استفاده می‌شود.

[1] Maximally specific model

برای تجزیه کارای رشته مورد نظر یا به عبارتی دیگر به دست آوردن کارای قلم اطلاعاتی متناظر با هر واحد در دنباله واحدهای رشته مرجع موردنظر، در روش‌های مبتنی بر HMM از الگوریتمی به نام Viterbi استفاده می‌شود که دارای پیچیدگی محاسباتی $O(TN^2)$ بوده که N تعداد حالت‌های HMM مورد نظر و T طول دنباله خروجی است.

برای بررسی کیفیت روش ارائه شده، نویسندگان آن را با استفاده از ۳۵۰ رشته مرجع موجود در نسخه‌ای تغییریافته از مجموعه داده cora (Seymore et al., 1999) آموزش داده و با استفاده از ۱۴۲ رشته مرجع از آن آزمایش کرده‌اند. نتایج آزمایشات نشان داده که در استفاده از این روش مقادیر ۸۴,۹٪ و ۸۴,۱٪ برای پارامترهای دقت و فراخوانی در سطح قلم اطلاعاتی حاصل شده است.

روش ارائه شده توسط یین[1] و همکاران

HMM ارائه شده در بخش قبل فراوانی لغات و اطلاعات مکانی[2] لغات در اقلام اطلاعاتی را در نظر می‌گیرد (اینکه یک لغت، لغت ابتدایی یک قلم اطلاعاتی باشد یا از جمله سایر لغات آن). با این حال، این مدل ترتیب حضور لغات در اقلام اطلاعاتی را در نظر نمی‌گیرد؛ درحالیکه در واقعیت چنین نبوده و از ترتیب حضور لغات نیز برای تجزیه یک رشته مرجع می‌توان استفاده کرد. برای مثال در HMM استفاده شده در روش قبل احتمال حضور عباراتی مانند technical report و report technical در یک قلم اطلاعاتی از رشته‌های مرجع یکسان است در حالیکه با توجه به واقعیت نباید چنین باشد. برای در نظر گرفتن این مهم و بهبود نتایج ، در (Yin et al., 2004)، از HMM دوتایی برای حل مساله تجزیه رشته‌های مرجع استفاده شده است که علاوه بر فراوانی و اطلاعات مکانی لغات، روابط دنباله‌های دوتایی لغات[3] را نیز در نظر می‌گیرد.

مجموعه حالت‌ها و نمادها و احتمالات انتقال در این روش، همانند روش قبل محاسبه شده اما احتمالات emission به شکلی جدید محاسبه می‌شود. در این روش، احتمال emission درونی با استفاده از یک مدل دوتایی محاسبه شده و بدین طریق روابط دنباله‌های دوتایی از لغات درون یک قلم اطلاعاتی در نظر گرفته می‌شود. احتمال emission در این مدل به صورت زیر نوشته می‌شود:

---

[1] Yin
[2] Position information
[3] Bigram sequentioal relation

$$P(\sigma|q) = \begin{cases} P(q \underline{\uparrow} \sigma) \\ P(q \overline{\uparrow} \sigma) = P(\sigma|\sigma_{-1}, q) \end{cases} \quad (۳-۲)$$

که $P(q \underline{\uparrow} \sigma)$ احتمال حضور σ در آغاز q است و $P(q \overline{\uparrow} \sigma)$ احتمال حضور σ درون q و حضور σ_{-1} به عنوان کلمه قبل از آن است و $P(\sigma|\sigma_{-1}, q) = \frac{c(q\uparrow\sigma_{-1}\sigma)}{c(q\uparrow\sigma_{-1})}$.

علاوه بر این، لازم به ذکر است که در روش ارائه شده در این مقاله از یک روش هموارسازی[1] به نام back off-shrinkage (Bikel et al., 1997) برای هموارسازی کردن احتمالات emission به دست آمده استفاده شده است. این رویکرد زمانی مناسب است که داده‌های آموزش ناکافی بوده و در نتیجه در بسیاری از موارد احتمال emission صفر به دست آمده باشد.

پس از محاسبه ماتریس‌های مورد نیاز، HMM دوتایی برای حل مسأله تجزیه رشته‌های مرجع به دست می‌آید. در ادامه همانند روش قبل از الگوریتم Viterbi (نسخه‌ای تغییریافته از آن) برای تجزیه رشته‌های مرجع جدید استفاده می‌شود.

برای آموزش و انجام آزمایشات با استفاده از روش پیشنهادی، در (Yin et al. 2004)، از ۷۱۳ رشته مرجع برچسب‌گذاری شده (به صورت دستی) و روش اعتبارسنجی متقابل ۴ سطحی[2] استفاده شده است. نتایج به دست آمده نشان‌دهنده مقداری تقریبا برابر با ۹۰٪ برای هر دو پارامتر دقت و بازخوانی بوده که این مهم نشان‌دهنده کیفیت بهتر روش ارائه شده در این مقاله به نسبت روش‌های تجزیه رشته مرجع مبتنی بر قاعده و همچنین روش تجزیه رشته مرجع مبتنی بر HMM ارائه شده در بخش قبل است.

روش ارائه شده توسط اوجوکوه[3] و همکاران

با توجه به نتایج به دست آمده حاصل از استفاده از HMM دوتایی در (Yin et al., 2004)، در (Ojokoh et al., 2011) نویسندگان از مدل HMM سه‌تایی[4] برای حل مسأله تجزیه رشته‌های مرجع استفاده کرده‌اند. در روش جدید از یک ماتریس انتقال سه‌بعدی استفاده شده که در نتیجه آن، انتقال

[1] Smoothing
[2] ۴- level cross validation
[3] Ojokoh
[4] Trigram HMM

به یک حالت نه تنها به حالت قبل، بلکه به حالت ماقبل‌تر نیز وابسته است. علاوه بر این، در این روش از رویکردی جدید برای هموارسازی احتمالات انتقال و emission استفاده شده است.

مجموعه حالت‌ها و نمادها در این روش، همانند سایر روش‌های ارائه شده در این بخش بوده اما احتمالات انتقال و emission به شکلی جدید در آن محاسبه می‌شود. برای محاسبه احتمالات انتقال و emission در اینجا نیز همانند سایر روش‌های ارائه شده در این بخش، در ابتدا HMMای که همه رشته‌های موجود در مجموعه داده آموزش را تولید کند، ایجاد شده و سپس به روشی مشابه با استفاده از Neighbor merging و V-merging برخی از حالت‌های با برچسب یکسان ادغام شده تا مدلی به غایت مشخص ایجاد شود. در ادامه با استفاده از این مدل، ماتریس سه‌بعدی احتمال انتقال $A = a_{ijk}$ به صورت $a_{ijk} = a_{ij} \cdot a_{jk}$ محاسبه می‌شود که $a_{ij} = P(S_i \to S_j)$ احتمال انتقال از حالت iام به حالت jام و $a_{jk} = P(S_j \to S_k)$ احتمال انتقال از حالت jام به حالت kام است. ماتریس احتمال emission یعنی B نیز به صورت زیر محاسبه می‌شود:

$$B = b_j(V_m) * b_k(V_m) \qquad (۴-۲)$$

که ماتریس‌های $b_j(V_m)$ و $b_k(V_m)$ به صورت زیر محاسبه می‌شوند:

$$b_j(V_m) = \begin{cases} P(V_{m-1}V_m, S_i \to S_j) \text{ if } i = j \\ P(V_m, S_i \to S_j) \quad \text{otherwise} \end{cases} \qquad (۵-۲)$$

و

$$.b_k(V_m) = \begin{cases} P(V_{m-1}V_m, S_j \to S_k) \text{ if } j = k \\ P(V_m, S_j \to S_k) \quad \text{otherwise} \end{cases} \qquad (۶-۲)$$

علاوه بر این، در این مقاله برای هموارسازی احتمالات محاسبه شده از نسخه‌ای تغییر یافته از back off-shrinkage استفاده شده است که بنا به گفته نویسندگان، با استفاده از آن اهمیت بیشتری به کلمات emit شده در انتقال به حالت مشابه داده می‌شود. نویسندگان بیان کرده‌اند که با استفاده از این روش توانسته‌اند به طور محسوسی دقت و فراخوانی روش ارائه شده را افزایش دهند.

پس از محاسبه ماتریس‌های مورد نیاز، زمانی که HMM سه‌تایی برای حل مسأله تجزیه رشته‌های مرجع به دست آمد، همانند روش‌های قبل، می‌توان از الگوریتم Viterbi (نسخه‌ای تغییر یافته از آن) برای تجزیه رشته‌های مرجع جدید استفاده کرد.

برای آموزش و انجام آزمایشات با استفاده از مدل پیشنهادی، نویسندگان از ۳ مجموعه داده استفاده کرده‌اند که در اینجا تنها نتایج گزارش شده در مورد مجموعه داده cora (Seymore et al., 1999) را بیان می‌کنیم. لازم به ذکر است که در استفاده از cora برای آموزش و آزمایش روش پیشنهادی، نویسندگان از روش اعتبارسنجی متقابل ۴سطحی استفاده کرده‌اند. نتایج گزارش شده توسط نویسندگان نشان‌دهنده مقداری بالاتر از ۹۵٪ برای هر دو پارامتر دقت و فراخوانی بوده که این مهم نشان‌دهنده کیفیت بهتر روش ارائه شده در مقایسه با روش‌های تجزیه رشته‌های مرجع مبتنی بر قاعده و همچنین سایر روش‌های تجزیه رشته‌های مرجع مبتنی بر HMM ارائه شده در این بخش است.

۲-۲-۲- تجزیه رشته‌های مرجع با استفاده از روش‌های میدان‌های تصادفی[1] (CRF)

با توجه به دستیابی ساده‌تر به نتایجی تقریبا بهتر، CRF ابزاری است که بیش از سایر روش‌های یادگیری ماشین برای حل مسأله تجزیه رشته‌های مرجع مورد استفاده قرار گرفته است. در استفاده از این ابزار، مسأله تجزیه رشته‌های مرجع به عنوان یک مسأله برچسب‌زنی دنباله‌ای در نظر گرفته شده و سپس از CRF برای حل آن استفاده می‌شود. با توجه به بررسی‌های صورت گرفته، CRF استفاده شده در کارهای انجام گرفته در این راستا مشابه بوده و تفاوت کارهای انجام گرفته در این زمینه در ویژگی[2]های به کار رفته در آن‌ها و همچنین زمینه علمی که در آن مسأله تجزیه رشته‌های مرجع مورد بررسی قرار گرفته، است. در ادامه این بخش، در ابتدا توصیفی از نحوه استفاده از CRF برای حل یک مسأله برچسب‌زنی دنباله‌ای بیان شده و سپس روش‌های موجود در این دسته را بررسی خواهیم کرد.

[1] Conditional Random fields
[2] Feature

میدان‌های تصادفی شرطی

میدان‌های تصادفی شرطی مدل‌های گرافیکی فاقد جهتی هستند که از آن‌ها برای بیشینه کردن یک احتمال شرطی استفاده می‌شود (Lafferty et al., 2001). یک CRF زنجیره خطی با پارامترهای $\Lambda = \{\lambda, \dots\}$ احتمال شرطی را برای دنباله حالت‌های $y = y_1 \dots y_T$ با ورودی $x = x_1 \dots x_T$ به صورت زیر تعریف می‌کند:

$$P_\lambda(y|x) = \frac{1}{Z_x} \exp\left(\sum_{t=1}^{T} \sum_k \lambda_k f_k(y_{t-1}, y_t, x, t)\right), \qquad (۲-۷)$$

که Z_x ثابت نرمال‌سازی بوده و باعث می‌شود مجموع احتمالات برای همه دنباله‌های حالات برابر با یک شود، $f_k(y_{t-1}, y_t, x, t)$ یک تابع ویژگی (معمولا دودویی) بوده و λ_k ضریب یادگرفته شده متناظر با f_k است. توابع ویژگی می‌توانند جنبه‌ای از یک انتقال حالت $y_{t-1} \to y_t$ و دنباله مشاهدات با مرکزیت زمان t باشند. مقادیر مثبت بزرگ برای λ_k نشان‌دهنده احتمال وقوع بالای یک اتفاق و مقادیر منفی نشان‌دهنده بعید بودن یک اتفاق است.

با ورودی مدل تعریف شده در رابطه فوق، ممکن‌ترین دنباله برچسب با ورودی x، یعنی $y^* = \arg\max_y P_\Lambda(y|x)$ را می‌توان به صورت کارا با برنامه‌نویسی پویا و با استفاده از الگوریتم Viterbi به دست آورد. پارامترهای مورد نیاز مسأله با استفاده از روش maximum likelihood به دست می‌آید.

روش ارائه شده توسط کانسیل[1] و همکاران

در (Councill et al., 2008)، نویسندگان با استفاده از CRF روشی جدید برای تجزیه رشته‌های مرجع ارائه کرده‌اند. بنابر ادعای نویسندگان، نوآوری آن‌ها در به کارگیری ویژگی‌های جدید برای حل مسأله، بسته‌بندی نتایج به عنوان یک ماژول نرم افزاری[2] قابل استفاده هم به صورت جداگانه و هم به صورت یک برنامه کاربردی وب[3]، و متن-باز[4] کردن برنامه ایجاد شده است.

دسته‌بندی کلی ویژگی‌های استفاده شده در این روش به شرح زیر است:

---

[1] Councill
[2] Software module
[3] Web application
[4] Open source

- شناسه واحد: این دسته شامل سه ویژگی است: ۱- خود واحد به شکلی که هست، ۲- نوشتار واحد با حروف کوچک و ۳- نوشتار واحد با حروف کوچک و حذف علائم نشان‌گذاری.
- پیشوند/پسوندهای چندتایی: این دسته شامل ۹ ویژگی است: ۱- اولین حرف واحد، ۲- دومین حرف واحد، ۳- سومین حرف واحد، ۴، چهارمین حرف واحد، ۵- آخرین حرف واحد، ۶- دومین حرف واحد از انتها، ۷- سومین حرف واحد از انتها، ۸- چهارمین حرف واحد از انتها و ۹- ویژگی دیگری برای بررسی اینکه آیا آخرین حرف واحد بزرگ، کوچک یا عدد است.
- حالت املایی[1]: شامل یک ویژگی برای بررسی تنها بزرگ بودن حرف اول واحد، درهم بودن حروف کوچک و بزرگ در واحد، بزرگ بودن تمام حروف واحد و ... .
- نشان‌گذاری[2]: شامل یک ویژگی در سطح ریزدانه[3] که بیان‌گر نوع نشان‌گذاری استفاده شده در واحد بوده و می‌تواند مقادیر زیر را اختیار کند: leading quotes، ending quotes، multiple hyphens، continuous punctuation (مانند "،" و "ک")، stop punctuation (مانند نقطه[4]، نقل‌قول‌های دوگانه[5]، paired braces، possible volume (مانند ۳(۴)) و غیره.
- عدد[6]: یک ویژگی برای بررسی ویژگی‌هایی عددی واحد که می‌تواند مقادیری مشخص مانند سال[7] (شامل 19xx تا 20xx)، شماره صفحه احتمالی[8] (شامل [0-9]-[0-9])، ترتیبی[9] (شامل عددی که پس از آن پسوندی مانند th آمده باشد)، یا دسته‌هایی کلی مانند ۴رقمی، ۳ رقمی، ۲ رقمی، تک رقمی، دارای رقم و فاقد رقم اختیار کند.

---

[1] Orthographic case
[2] Punctuation
[3] Fine grained
[4] Period
[5] Double quotes
[6] Number
[7] Year
[8] Possible page range
[9] Ordinal

- لغت‌نامه[1]: این دسته شامل ۶ ویژگی بوده که با استفاده از آن‌ها وجود واحد در لغت‌نامه‌هایی از نام انتشارات، نام مکان‌ها، نام‌های خانوادگی، نام‌های مذکر و مونث، و اسامی ماه‌ها بررسی می‌شود.
- محل[2]: از یک ویژگی برای ذخیره موقعیت نسبی یک واحد در یک رشته مرجع که به n قسمت مساوی تقسیم شده است (بنا بر آزمایشات مقدار n برابر با ۱۲ در نظر گرفته شده است) استفاده شده است. نویسندگان بیان کرده‌اند که با استفاده از این ویژگی تلاش کرده‌اند نظم موجود در قالب‌های ارجاع‌دهی را در نظر بگیرند.
- ویراستار احتمالی[3]: یک ویژگی برای بررسی وجود واحدی مانند ".eds" در رشته مرجع.
- پس از تجزیه رشته مرجع، برخی اقلام اطلاعاتی مانند نام، سال، شماره، شماره صفحه و ... می‌توانند به طرق مختلف بیان شوند. برای حل این مشکل، در این روش، پس از تجزیه، پردازش پسینی بر روی اقلام مختلف اطلاعاتی انجام می‌شود که حاصل آن یک شکل نرمال شده برای این اقلام اطلاعاتی است. لازم به ذکر است که در انجام آزمایشات برای بررسی کیفیت روش ارائه شده، نویسندگان در راستای مقایسه عادلانه با سایر روش‌ها، این مزیت را از کار خود حذف کرده‌اند. برای انجام مقایسات، از مجموعه داده‌های cora و CiteSeer استفاده شده که مجموعه داده سوم توسط خود نویسندگان ایجاد شده و شامل ۲۰۰ رشته مرجع تصادفا انتخاب شده از بین حدود ۱۴ میلیون رشته مرجع موجود در سیستم CiteSeerx است. نتایج آزمایشات صورت گرفته با توجه به مجموعه داده cora نشان‌دهنده مقدار 95.7٪ برای پارامترهای دقت و فراخوانی است. نتایج آزمایش روش با استفاده از مجموعه داده CiteSeer نیز قابل قبول اما ضعیف‌تر از نتایج به دست آمده روی cora بوده است که با توجه به آن، نویسندگان استدلال کرده‌اند که مجموعه داده cora شامل فرمت‌های متنوع از رشته‌های مرجع نبوده و در نتیجه کامل نیست.

روش ارائه شده توسط لوپز

[1] Dictionary
[2] Location
[3] Possible editor

- در (Lopez, 2009)، نویسندگان با هدف نشان دادن دقت روش‌های یادگیری ماشین در استخراج اطلاعات مقالات دانشگاهی، نتایج روش ارائه و پیاده‌سازی شده توسط خود را گزارش داده‌اند. روش یادگیری ماشین استفاده شده در (Lopez 2009)، CRF بوده و در آن از ۱۲۰۰ رشته مرجع به عنوان داده آموزش استفاده شده است. نتایج آزمایشات صورت گرفته توسط نویسندگان نشان‌دهنده میزان صحت ۹۵,۷٪ در سطح اقلام اطلاعاتی رشته‌های مرجع و ۷۸,۹٪ در سطح خود رشته مرجع است. لازم به ذکر است که روش مورد بحث در این مقاله، پیاده‌سازی شده و برنامه کاربردی تحت وب آن نیز قابل دسترس است.

روش ارائه شده توسط زو[1] و همکاران

- در (Zou et al., 2010)، نویسندگان از روش‌های یادگیری ماشین CRF و SVM برای مکان‌یابی و تجزیه رشته‌های مرجع موجود در فایل‌های HTML مقالات پزشکی استفاده کرده‌اند. برای مکان‌یابی مراجع، نویسندگان در ابتدا فایل HTML مقالات را رندر[2] کرده، روی مرورگر بارگذاری کرده، آن را به چندین منطقه[3] تقسیم کرده، ویژگی‌های هندسی[4] هر منطقه را به دست آورده و با استفاده از یک دسته‌بند دو دسته‌ای[5] SVM مناطق به دست آمده را به مناطق مرجع و مناطق غیرمرجع تقسیم‌بندی کرده‌اند. علاوه بر این نویسندگان با استفاده از ویژگی‌های بخش مراجع مانند اینکه مراجع همیشه در کنار هم بوده و با شکست خط[6] از هم جدا می‌شوند، روند فوق را تسریع داده و قابلیت اطمینان نتایج روش را ارتقا داده‌اند.
- پس از مکان‌یابی بخش مراجع و تجزیه این بخش به رشته‌های مرجع مجزا با استفاده از شکست خط‌های موجود، نویسندگان از CRF برای تجزیه هر یک از رشته‌های مرجع استفاده کرده‌اند.
- در این مقاله، یک رشته مرجع به دنباله‌ای از لغات تقسیم‌بندی شده و مشاهدات در محل t علاوه بر خود لغت، شامل ۱۴ ویژگی دودویی به شرح زیر است: ۳ ویژگی برای بررسی

---

[1] Zou
[2] Render
[3] Zone
[4] Geometric
[5] Two class classifier
[6] Line break

وجود لغت در لغت‌نامه‌هایی از اسامی نویسندگان، عنوان مقالات و عنوان نشریات (ایجاد شده از داده‌های ۱۰ سال MEDLINE)، ۱ ویژگی برای بررسی مشابهت لغت با فرمت صفحه، ۱ ویژگی برای بررسی شباهت لغت با فرمت نگارش ابتدای یک نام نویسنده، (مانند .N و .H.-N)، ۱ ویژگی برای بررسی عددی ۴ رقمی و کمتر از سال فعلی بودن لغت، ۱ ویژگی برای بررسی بودن لغت به صورت‌های et، .al، et یا .al، ۱ ویژگی برای بررسی بودن لغت به صورت مشخص‌کننده صفحه (.p، .pp، p یا pp)، ۱ ویژگی برای بررسی خاتمه یافتن لغت با "."، ۱ ویژگی برای بررسی بزرگ یا کوچک بودن حرف اول لغت، ۱ ویژگی برای بررسی وجود کاراکتری غیر از حرف در لغت، ۱ ویژگی برای بررسی تنها شامل رقم بودن لغت، ۱ ویژگی برای بررسی وجود همزمان عدد و حرف در لغت، و ۱ ویژگی برای بررسی وجود تنها عدد و حرف در لغت.

نویسندگان، برای آموزش و آزمایش مدل، مجموعه داده‌ای شامل ۲۴۰۰ رشته مرجع برچسب‌گذاری شده ایجاد کرده‌اند که از این میان، از ۶۰۰ رشته برای آموزش و از ۱۸۰۰ رشته برای آزمایش استفاده کرده‌اند. نویسندگان دقت روش را در دو سطح دقت در تشخیص صحیح برچسب لغت و قلم اطلاعاتی بررسی کرده‌اند که نتایج حاصل بیان‌گر دقت ۹۹,۰۴٪ در سطح لغت و ۹۷,۳٪ در سطح قلم اطلاعاتی می‌باشد.

روش ارائه شده توسط ژانگ[1] و همکاران

در (Zhang et al., 2011)، نویسندگان با استفاده از مجموعه مقالات با دسترسی آزاد PMC که در زمینه پزشکی است مجموعه داده‌ای شامل ۲۷۶۰۶ رشته مرجع برچسب‌گذاری شده (بعضا شامل خطا) ایجاد کرده و از آن برای آموزش و آزمایش یک مدل CRF برای حل مسأله تجزیه رشته‌های مرجع استفاده کرده‌اند. در این روش، برای آموزش و آزمایش، رشته‌های مرجع به دنباله‌هایی از واحدها شامل لغات و علائم نشان‌گذاری تقسیم شده و با استفاده از ویژگی‌های در نظر گرفته شده در (Settles, 2004) مدل پیشنهادی آموزش داده و آزمایش شده است.

[1] Zhang

نتایج آزمایشات انجام شده روی مجموعه داده ایجاد شده با استفاده از روش اعتبارسنجی متقابل ۱۰تایی نشان‌دهنده مقداری برابر با ۹۷,۹۴٪ برای پارامتر دقت و ۹۷,۹۶٪ برای پارامتر فراخوانی در سطح اقلام اطلاعاتی است.

روش ارائه شده توسط کیم[1] و همکاران

در (Kim et al., 2012)، نویسندگان با هدف بررسی اثرگذاری مدل CRF در تجزیه رشته‌های مرجع علوم اجتماعی و انسانی، با استفاده از مجموعه مقالات موجود در Revues.org مجموعه داده‌ای شامل بیش از ۳۰۰۰ رشته مرجع برچسب‌گذاری ایجاد کرده‌اند. علاوه‌براین، با تمرکز بر روی به دست آوردن روش‌های کلی برای واحدبندی کردن و استخراج ویژگی‌ها، نویسندگان سعی در ایجاد سیستمی غیر وابسته به فرمت رشته‌های مرجع و حتی زبان آن‌ها کرده‌اند. همچنین، برای بررسی عدم وابستگی مدل پیشنهادی به زمینه‌ای خاص از علوم، پس از آموزش مدل پیشنهادی با مجموعه داده علوم اجتماعی و انسانی ایجاد شده، روش پیشنهادی با استفاده از رشته‌های مرجع علوم دیگر مانند شیمی و فناوری اطلاعات نیز مورد آزمایش قرار گرفته است. نتایج مقایسات صورت گرفته با سیستم‌های تجزیه رشته مرجع ParsCit، Biblo، Freecite، و Gorbid، نشان‌دهنده بهتر عمل کردن این روش است. علاوه‌براین یکی دیگر از نتایج به دست آمده از آزمایشات این است که در حالیکه اغلب داده‌های موجود در مجموعه داده آموزش به زبان فرانسوی بوده، نتایج حاصل از اعمال روش روی داده‌های آزمایش به زبان انگلیسی نیز قابل قبول و مناسب بوده است.

روش ارائه شده توسط پنگ[2] و مک‌کالوم[3]:

در (Peng and McCallum, 2013)، از CRF برای به دست آوردن اطلاعات کتابخانه‌ای یک مقاله شامل تجزیه رشته‌های مرجع استفاده شده است. تمرکز اصلی نویسندگان در این‌جا بر روی به دست

[1] Kim
[2] Peng
[3] McCallum

آوردن اطلاعات مقاله ارجاع‌دهنده می‌باشد. با توجه به این مهم (یعنی عدم تمرکز نویسندگان بر روی مسأله تجزیه رشته‌های مرجع)، در (Peng and McCallum, 2013) اطلاعاتی درباره ویژگی‌های به کار برده شده در استفاده از CRF برای حل مسأله تجزیه رشته‌های مرجع بیان نشده است. برای بررسی کارایی روش، روش پیشنهادی با استفاده از مجموعه داده cora (Seymore et al., 1999) مورد آزمایش قرار گرفته که نتایج به دست آمده نشان‌دهنده مقدار ۹۱,۵٪ برای پارامتر دقت است.

روش ارائه شده توسط تکاسزکیک[1] و همکاران

در (Tkaczyk et al., 2014) و (Tkaczyk et al., 2015)، نویسندگان روشی ارائه کرده‌اند که با ورودی فایل pdf یک مقاله، فراداده[2]های آن، تمام متن ساختاریافته آن شامل بخش‌ها، زیربخش‌ها، و مجموعه مراجع و اقلام اطلاعاتی متناظر با هر رشته مرجع را خروجی می‌دهد. در این مقاله، با توجه به عدم نیاز به فاز یادگیری و آموزش روش‌های یادگیری ماشین بدون نظارت و دستیابی به نتایج خوب، برای دستیابی به رشته‌های مرجع مقاله از این روش‌ها استفاده شده است. برای تجزیه یک رشته مرجع، از مدل CRF استفاده شده و در آن، در ابتدا پس از واحدبندی یک رشته ، از ۴۳ ویژگی برای توصیف هر واحد استفاده شده است. ویژگی‌های در نظر گرفته شده علاوه بر خود واحد که به عنوان ویژگی اصلی است، شامل تعدادی ویژگی برای بررسی حضور کلاس‌های خاصی از کاراکتر ها مانند حضور عدد، حروف بزرگ، حروف کوچک و ... در واحد، تعدادی ویژگی برای بررسی وجود کارکترهای خاص مانند "،" یا "." یا "-" یا کلماتی خاص در واحد، و تعدادی ویژگی برای بررسی وجود واحد موردنظر در مجموعه‌های خاصی از داده‌ها مانند اسامی شهرها و کلمات به وفور یافت شده در عنوان نشریات و ... استفاده شده است. علاوه‌براین، لازم به ذکر است که در مدل CRF استفاده شده در (Tkaczyk et al., 2014) و (Tkaczyk et al., 2015)، برای تأثیر بردار ویژگی‌های واحدهای همسایه در تعیین برچسب یک واحد، بردار ویژگی متناظر با یک واحد به صورت مجموع بردار ویژگی‌های خود واحد و بردار ویژگی‌های واحدهای قبل و بعد از آن در نظر گرفته شده است.

[1] Tkaczyk
[2] Metadata

برای ارزیابی روش، نویسندگان از مجموعه داده‌های cora و CiteSeer و روش اعتبارسنجی متقابل ۵تایی استفاده کرده‌اند. نتایج به دست آمده نشان‌دهنده مقدار ۹۲٫۹٪ برای پارامتر دقت و ۹۳٫۸٪ برای پارامتر فراخوانی در سطح اقلام اطلاعاتی است.

۲-۲-۳- استفاده از SVM برای تجزیه رشته‌های مرجع

برای حل مسأله تجزیه رشته‌های مرجع دو روش با استفاده از SVM پیشنهاد شده که یکی مبتنی بر دسته‌بند چند دسته‌ای SVM و دیگری مبتنی بر SVM ساختاری است. با توجه به بررسی روش SVM با جزئیات دقیق در فصل بعد این پژوهش، در این قسمت از ذکر جزئیات روش SVM خودداری کرده و خوانندگان را به فصل بعد این گزارش ارجاع می‌دهیم. در ادامه روش‌های تجزیه رشته‌های مرجع مبتنی بر SVM را به طور خلاصه بررسی خواهیم کرد.

روش ارائه شده توسط زو و همکاران

در روش دوم ارائه شده در (Zou et al., 2010)، نویسندگان مسأله تجزیه رشته‌های مرجع را به عنوان یک مسأله دسته‌بند چند دسته‌ای در نظر گرفته و قلم اطلاعاتی متناظر با هر لغت موجود در رشته مرجع را با استفاده از دسته‌بند چنددسته‌ای SVM مشخص می‌کنند. علاوه‌براین، نویسندگان توجه کرده‌اند که قواعدی اکتشافی وجود دارند که همیشه در مورد رشته‌های مرجع صحیح هستند. با توجه به این مهم، نویسندگان پس از دسته‌بندی هر لغت موجود در دنباله لغات یک رشته مرجع مورد آزمایش، از این قواعد اکتشافی استفاده کرده و وجود تناقض با این قواعد را بررسی می‌کنند. در صورت نقض این قواعد، روش ایجاد شده از یک الگوریتم جستجو برای یافتن دنباله برچسب‌هایی با بیشترین احتمال که با این قواعد در تناقض نباشد، استفاده خواهد کرد.

برای طراحی دسته‌بند مورد نظر، در (Zou et al., 2010) از ۱۵ ویژگی زیر استفاده شده است: ۳ ویژگی برای بررسی وجود لغت در لغت‌نامه‌هایی از اسامی نویسندگان، عنوان مقالات و عنوان نشریات (ایجاد شده از داده‌های ۱۰ سال MEDLINE)، یک ویژگی برای بررسی مشابهت لغت با فرمت شماره صفحه، یک ویژگی برای بررسی شباهت لغت با فرمت نگارش نام یک نویسنده، (مانند .N و .H.-N)، یک ویژگی برای بررسی عددی ۴ رقمی و کمتر از سال فعلی بودن لغت، یک ویژگی

برای بررسی بودن لغت به صورت‌های et، al. et یا al.، یک ویژگی برای بررسی بودن لغت به صورت مشخص‌کننده شماره صفحه (p.، pp.، p یا pp)، یک ویژگی برای بررسی خاتمه یافتن لغت با "."، یک ویژگی برای بررسی بزرگ یا کوچک بودن حرف اول لغت، یک ویژگی برای بررسی وجود کاراکتری غیر از حرف در لغت، یک ویژگی برای بررسی وجود تنها رقم در لغت، یک ویژگی برای بررسی وجود همزمان عدد و حرف در لغت، یک ویژگی برای بررسی وجود تنها عدد و حرف در لغت و یک ویژگی برای نمایش موقعیت نرمال شده لغت در رشته مرجع (نرمال شده با توجه به تعداد لغات موجود در رشته مرجع). علاوه بر این، از آن‌جا که در مسأله تجزیه رشته‌های مرجع، لغات همسایه با احتمال بالاتری برچسب یکسانی خواهند داشت، در اینجا، در دسته‌بندی هر لغت، ویژگی‌های لغات قبل و بعد آن نیز در نظر گرفته می‌شوند.

نتایج آزمایشات انجام شده روی روش پیشنهادی و مجموعه داده ایجاد شده توسط نویسندگان نشان‌دهنده دقت ۹۷٫۳٪ در سطح لغت و دقت ۹۸٫۹۸٪ در سطح قلم اطلاعاتی است.

روش ارائه شده توسط ژنگ و همکاران

SVM ساختاری یک روش یادگیری ماشین با نظارت است که برای پیش‌بینی خروجی‌های با ساختار پیچیده مانند دنباله‌ها و گراف‌ها طراحی شده است. این ابزار به مسأله یافتن نگاشت کلی $f: X \to Y$ با ورودی نمونه‌هایی از زوج‌های ورودی خروجی آموزش $(x_1, y_1) \dots (x_n, y_n) \in X \times Y$ می‌پردازد به طوریکه خطای پیش‌بینی کمی وجود داشته باشد. ایده یادگیری SVM ساختاری، یادگیری یک تابع تفکیک‌کننده[1] F است که با استفاده از آن بتوانیم یک پیش‌بینی را با بیشینه کردن F روی Y و با ورودی مشخص x به دست آوریم: $f(x, w) = \arg\max F(x, y, w)$. $F(x, y, w) = w^T \psi(x, y)$ ترکیبی خطی از توصیفات مشترک ویژگی‌های ورودی و خروجی است که w برداری پارامتری و ψ یک بردار ویژگی است که x و y را به یکدیگر وابسته می‌کند.

[1] Discriminant

در (Zhang et al., 2011)، نویسندگان بیان کرده‌اند که با توجه به ساختار منظم موجود در رشته‌های مرجع، مسأله تجزیه مرجع را می‌توان به عنوان یک مسأله یادگیری دنباله‌ای[1] در نظر گرفت و در ادامه از SVM ساختاری برای حل این مسأله استفاده کرد.

در روش ارائه شده در (Zhang et al., 2011)، در ابتدا یک رشته مرجع پیش‌پردازش شده و با توجه به فاصله‌ها و علائم نشان‌گذاری به دنباله‌ای از واحدها تقسیم می‌شود. ویژگی‌های در نظر گرفته شده برای هر واحد نیز همانند مقاله ارائه شده در بخش قبل در نظر گرفته شده است. با توجه به اینکه در مسأله تجزیه رشته‌های مرجع، SVM ساختاری به عنوان یک الگوریتم یادگیرنده دنباله‌ای پیاده‌سازی می‌شود، تابع ψ در اینجا دربرگیرنده دو نوع ویژگی خواهد بود: ویژگی‌های انتقال حالت و ویژگی‌های مشاهده شده که برای واحدهای مستقل به دست آمده است. ویژگی‌های انتقال حالت برچسب‌های اختصاص‌یافته به واحدهای همسایه برای مدل کردن وابستگی بین برچسب‌های واحدهای همسایه در نظر گرفته می‌شود.

برای آموزش و انجام آزمایش روی روش پیشنهادی، نویسندگان با استفاده از مجموعه مقالات انتشارات MEDLINE مجموعه داده‌ای شامل ۶۰۰ رشته مرجع برای آموزش و ۱۸۰۰ رشته مرجع برای آزمایش ایجاد کرده و با پیاده‌سازی روش پیشنهادی و روش‌های مبتنی بر SVM معمولی و CRF نتایج حاصله را با یکدیگر مقایسه کرده‌اند. نتایج آزمایشات انجام شده نشان‌دهنده این است که در صورت در نظر نگرفتن ویژگی‌های واحدهای همسایه در بردار ویژگی‌های یک واحد، روش مبتنی بر SVM ساختاری در مقایسه با سایر روش‌ها به طور چشم‌گیری بهتر عمل می‌کند؛ اما در صورت در نظر گرفتن ویژگی‌های واحدهای همسایه تفاوت مورد نظر تقریبا از بین می‌رود. برای مثال با در نظر گرفتن ویژگی‌های چهار واحد همسایه هر واحد در محاسبات، دقت به دست آمده برای روش‌های مبتنی بر SVM، CRF و SVM ساختاری به ترتیب برابر با ۹۷٫۸۴٪، ۹۸٫۹۱٪، و ۹۸٫۹۹ ٪است.

[1] Sequence learning

۲-۲-٤- تجزیه رشته‌های مرجع با استفاده از روش‌های یادگیری ماشین بدون نظارت

بررسی پیشینه مساله بیان‌گر این است که تنها یک روش تجزیه رشته‌های مرجع مبتنی بر یادگیری ماشین بدون نظارت وجود دارد که نتایج آن نیز در مقایسه با سایر روش‌ها، ازجمله روش‌های مبتنی بر یادگیری ماشین با نظارت، ضعیف‌تر است. در ادامه این روش را به طور خلاصه بررسی خواهیم کرد.

روش ارائه شده توسط آجیچتین[1] و گانتی[2]

در (Agichtein and Ganti, 2004)، نویسندگان فرض کرده‌اند که به مجموعه‌ای از رشته‌های مرجع برچسب‌گذاری شده دسترسی نداشته بلکه تنها مجموعه داده‌ای تجزیه شده و ذخیره شده در تعدادی جدول وجود دارد. به عبارت دیگر در روش در نظر گرفته شده در (Agichtein and Ganti, 2004)، رشته‌های مرجع نیمه ساختار یافته وجود نداشته و در نتیجه در مجموعه داده در دسترس، خطاهای رایج وجود نداشته، ترتیب فیلدها مشخص نبوده و برچسب فیلدها (ستون‌های جدول) نیز نامشخص است. با توجه به این موارد، نویسندگان استفاده از روش‌های یادگیری ماشین بدون نظارت را برای حل این مسأله پیشنهاد کرده‌اند. با توجه به ضعیف بودن نتایج روش ارائه شده در (Agichtein and Ganti, 2004) و نیازمندی‌های آن، در این طرح پژوهشی از بررسی بیشتر آن خودداری کرده و علاقه‌مندان را به مرجع موردنظر ارجاع می‌دهیم.

[1] Agichtein
[2] Ganti

فصل سوم

دسته‌بند ماشین بردار پشتیبان (SVM)

همانطور که در فصل پیش مشخص گردید، در مقایسه با روش‌های مبتنی بر قاعده، استفاده از روش‌های مبتنی بر یادگیری ماشین نتایج بسیار بهتری را برای حل مساله تجزیه رشته‌های مرجع ارائه می‌دهند. علاوه بر این، بررسی پیشینه نشان داد که از میان روش‌های یادگیری ماشین با نظارت استفاده شده برای حل مساله مورد نظر (HMM، CRF و SVM)، SVM و CRF نتایج بهتری را ارائه کرده‌اند. با توجه به محدودیت‌های زمانی، روند استفاده و برتری کلی SVM نسبت به CRF، در این پژوهش روش SVM را انتخاب کرده و از آن برای حل مساله تجزیه رشته‌های مرجع در زبان فارسی استفاده خواهیم کرد. با توجه به این مهم، در ابتدای این فصل به مساله دسته‌بندی و معیارهای ارزیابی روش‌های دسته‌بندی پرداخته و سپس، جزئیات روش دسته‌بندی ماشین بردار پشتیبان SVM ارائه خواهد شد. لازم به ذکر است که اغلب مطالب ذکر شده در این فصل با استفاده از (کارگر ۱۳۹۰ و نصیری ۱۳۹۴) نگاشته شده‌اند.

۳-۱- 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.

در اغلب مسائل دسته‌بندی، برای ارزیابی روش‌ها از پارامترهای دقت[1]، فراخوانی[2] و F1 استفاده می‌شود که در ادامه آن‌ها را تعریف خواهیم کرد:

[1] Precision
[2] Recall

پارامتر دقت برای هر دسته به صورت $\frac{tp}{tp+fp}$ محاسبه می‌شود که tp و fp به ترتیب برابر با تعداد نمونه‌هایی است که به درستی و اشتباها در دسته موردنظر قرار گرفته‌اند. پارامتر فراخوانی برای هر دسته به صورت $\frac{tp}{tp+fn}$ محاسبه می‌شود که tp برابر با تعداد نمونه‌هایی است که به درستی در دسته موردنظر قرار گرفته و fn تعداد نمونه‌هایی است که درحالیکه باید در این دسته قرار می‌گرفتند ، در دسته دیگری قرار گرفته‌اند. پارامتر F1 نیز برابر با میانگین دو پارامتر دقت و فراخوانی در نظر گرفته می‌شود.

۳-۲- دسته‌بند ماشین بردار پشتیبان (SVM)[1]

روش ماشین بردار پشتیبان (SVM) از کارامدترین روش‌های دسته‌بندیست که در سال ۱۹۹۵ معرفی شده است (وپنیک[2] ۱۹۹۵). 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 (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.

۳-۲-۱- 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 (وپنیک ۱۹۹۵، وپنیک ۱۹۹۸). 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

[1] Support Vector Machine
[2] Vapnik

طوری که اگر x_i متعلق به دسته ۱ باشد y_i مقدار ۱ و چنانچه متعلق به دسته ۲ باشد مقدار 1- 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 $s = \{(x_i, y_i)\}_{i=1}^{M}$ به وسیله یک ابرصفحه خطی $\bar{w}^T x + \bar{b} = 0$ که $\bar{w}^T \in R^M$ و $\bar{b} \in R$ به نحوی از یکدیگر جدا شوند که رابطه زیر برقرار باشد:

$$\begin{cases} \bar{w}x_i + \bar{b} > 0 & y_i = 1 \\ \bar{w}x_i + \bar{b} < 0 & y_i = -1 \end{cases} \quad i = 1,2,,\dots,M. \qquad (۱-۳)$$

رابطه فوق را می‌توان به صورت زیر بازنویسی کرد:

$$\exists \varepsilon > 0 \; s.t.: \begin{cases} \bar{w}x_i + \bar{b} \geq \varepsilon & y_i = 1 \\ \bar{w}x_i + \bar{b} \leq \varepsilon & y_i = -1 \end{cases} \quad i = 1,2,,\dots,M \qquad (۲-۳)$$

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:

$$\begin{cases} w^T x_i + b \geq 1 & y_i = 1 \\ w^T x_i + b \leq -1 & y_i = -1 \end{cases} \quad i = 1,2,,\dots,M. \qquad (۳-۳)$$

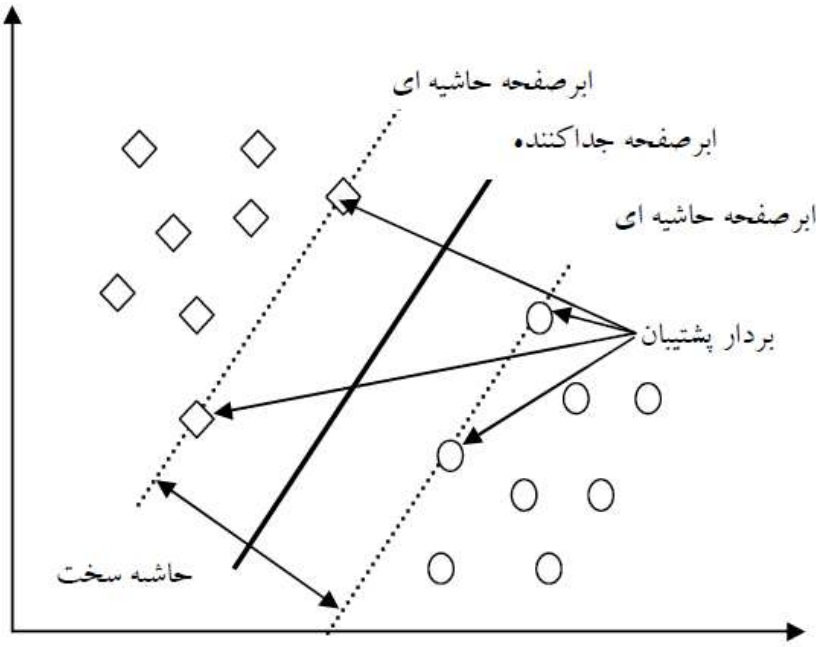
یا به عبارت دیگر

$$y_i(w^T x_i + b) \geq 1 \qquad i = 1,2,,\dots,M \qquad (۴-۳)$$

تعریف. مجموعه داده‌های $s = \{(x_i, y_i)\}_{i=1}^{M}$ را جداپذیر خطی[1] 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[2]، ناحیه بین این دو ابر صفحه حاشیه سخت[3]، فاصله بین این دو ابرصفحه پهنای حاشیه سخت و بردار x_i که در یکی از ابرصفحات حاشیه‌ای صدق کند بردار پشتیبان[4] نامیده می‌شوند.

---

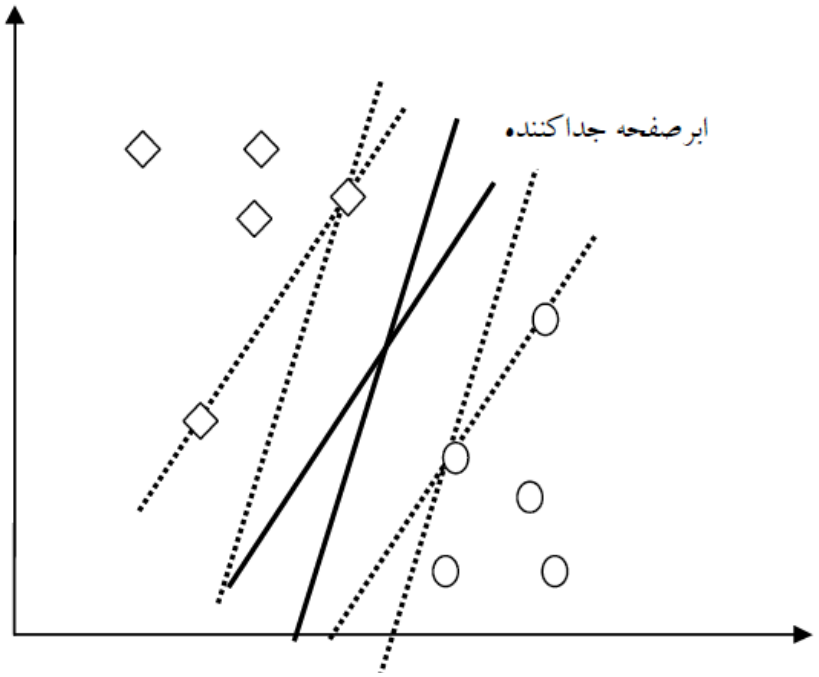
[1] Linearly Seperable
[2] Marginal Hyperplane
[3] Hard Margin
[4] Support Vector



شکل ۳. نمایش ابرصفحات حاشیه‌ای و ابرصفحه جداکننده برای تعدادی داده (کارگر ۱۳۹۰)

در شکل ۳ 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 نیز نشان داده شده است برای هر مجموعه داده جداپذیر خطی ابرصفحه‌های جدا کننده 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. 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

$$y_i(w^T x_i + b) \geq 1 \qquad i = 1,2,,\dots,M, w \in R^m, b \in R \qquad (۵-۳)$$

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

تعریف. تابع $D(x) = w^T x + b$ را تابع تصمیم[2] 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

[1] Optimal Sperating Hyperplane
[2] Decision Function

شده در تعریف فوق همان مدل مورد بحث است. تشخیص دسته داده‌ها با استفاده از این تابع تصمیم به صورت زیر انجام می‌شود:

$$x \in \begin{cases} 1 \text{ دسته } D(x) > 0 \\ 2 \text{ دسته } D(x) < 0 \end{cases} \qquad (۶-۳)$$

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:

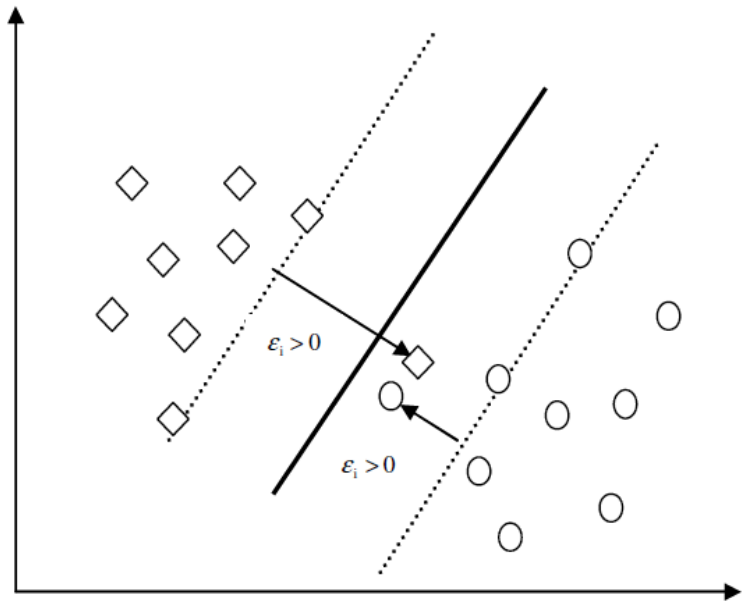
- داده‌هایی که بتوان با قبول درجه‌ای از خطا آن‌ها را با استفاده از یک ابرصفحه جداسازی کرد. این بدین معنی است که ممکن است در این دسته‌بندی دسته اشتباه به تعداد اندکی از داده‌ها اختصاص یابد.
- داده‌هایی که در صورت جداسازی خطی آن‌ها، خطای زیادی حاصل شود به طوری که مدل به دست آمده از این جداسازی فاقد اعتبار 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.

۳-۲-۲- 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 (۳-۴) اجازه هیچ گونه خطایی در دسته‌بندی نمی‌دهد. برای مجاز کردن خطا رابطه مورد نظر به صورت زیر تغییر داده می‌شود:

$$y_i(w^T x_i + b) \geq 1 - \varepsilon_i \qquad i = 1,2,,\dots,M \quad - \qquad (۳-۷)$$

که ε_i یک متغیر کمبود نامنفی است. شکل ۵ 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 ۵ 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 ۵. 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[1] گویند. در شکل 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:

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(۳-۷) 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.

۳-۲-۳- SVM برای دسته‌بندی داده‌های چند دسته‌ای

در بخش قبل، روش SVM برای دسته‌بندی داده‌های دو دسته‌ای مورد بررسی قرار گرفت. اما، اغلب مسائل در دنیای واقعی، مانند مساله مورد بررسی در این پژوهش، چند دسته‌ای بوده و استفاده از روش‌های دو دسته‌ای به خودی خود برای حل این مسائل مفید نیست. چندین رویکرد برای گسترش روش SVM با هدف استفاده از آن در مسائل چند دسته‌ای ارائه شده که در این بخش، دو نوع

[1] Soft marigin

متداول‌تر آن‌ها مورد بررسی قرار خواهد گرفت. برای کسب اطلاعات بیشتر در مورد سایر روش‌های دسته‌بندی SVM چند دسته‌ای به (کارگر ۱۳۹۰) مراجعه شود:

- روش SVM چند دسته‌ای با رویکرد یک دسته در مقابل سایر دسته‌ها[1] که به اختصار OSVM نامیده می‌شود.
- روش SVM چند دسته‌ای با رویکرد یک دسته در مقابل هر دسته[2] دیگر که به اختصار PSVM نامیده می‌شود.

بنابر فرمول‌بندی ارائه شده توسط وپنیک (Vapnik 1995)، در OSVM تبدیل یک مساله k دسته‌ای به k مساله دو دسته‌ای، از ابتدایی‌ترین روش‌های دسته‌بندی داده‌های چند دسته‌ای بوده از کارایی پایینی برخوردار است. در این روش برخی از نواحی در فضای مساله غیر دسته‌بندی شده باقی خواهد ماند. برای بهبود این مشکل، در روش PSVM ارائه شده توسط کربل (Krebel 1999) یک مساله k دسته‌ای به k(k-1)/2 مساله دو دسته‌ای که شامل همه ترکیبات دو دسته‌ای از مجموع k دسته است، تبدیل می‌شود. در ادامه جزئیات این روش‌ها را مورد بررسی قرار خواهیم داد.

روش OSVM برای دسته‌بندی داده‌های چند دسته‌ای

مجموعه داده‌های $s = \{(x_i, \bar{y}_i)\}_{i=1}^{M}$ که $x_i \in R^m (i = 1,2, \dots, M)$، $\bar{y}_i \in \{1,2, \dots, k\}$، $k > 2$ تعداد دسته‌ها و M تعداد داده‌ها است، را در نظر بگیرید. در روش OSVM، هر بار یکی از دسته‌ها از سایر دسته‌ها جداسازی می‌شود. برای این کار با فرض اینکه جداسازی دسته jام از سایر دسته‌ها مدنظر باشد، داده‌ها به صورت مساله‌ای دو دسته‌ای که در آن دسته اول شامل داده‌های دسته jام (متناظر با $y_i = 1$) و دسته دوم شامل داده‌های سایر دسته‌ها (متناظر با $y_i = -1$) در نظر گرفته می‌شوند. به عبارت دیگر:

$$y_i = \begin{cases} 1 & \bar{y}_i = j \\ -1 & \bar{y}_i \neq j \end{cases} \qquad i = 1, \dots, M \qquad (۸-۳)$$

[1] One- Against- all SVM
[2] Pairwise SVM

در ادامه، جداسازی به وسیله روش‌های SVM دو دسته‌ای انجام می‌شود. این عمل برای تمام دسته‌ها، یعنی برای $j = 1,2, \dots, k$، انجام می‌شود.

تعریف. مجموعه داده‌های k دسته‌ای $s = \{(x_i, \bar{y}_i)\}_{i=1}^{M}$ را در نظر بگیرید. فرض کنید برای $j = 1, \dots, k$، در روش OSVM تابع تصمیم $D_j(x) = w_j{}^T x + b_j$ برای جداسازی داده‌های دسته j ام از داده‌های سایر دسته‌ها به دست آمده باشد. در این صورت، در صورت وجود j–ای یکتا که $D_j(x) > 0$، آنگاه داده x متعلق به دسته jام است. در صورتی که رابطه فوق به ازای چند j برقرار باشد و یا این رابطه به ازای هیچ مقداری از j برقرار نباشد، داده x قابل دسته‌بندی نیست.

در روش OSVM ممکن است بسیاری از داده‌ها قابل دسته‌بندی نباشند. دلیل این امر این است که در این روش برخی از نواحی فضای دسته‌بندی در هیچ یک از دسته‌ها قرار نمی‌گیرند. به عبارت دیگر، وجود نواحی دسته‌بندی نشده در این روش از مشکلات آن محسوب می‌شود. برای بهبود این مشکل، راه‌حل‌های متفاوتی ارائه شده که یکی از آن‌ها روش PSVM است که در ادامه به آن پرداخته خواهد شد. لازم به ذکر است که روش SVM نیز مشکل نواحی دسته‌بندی نشده را به طور کامل حل نمی‌کند و راه‌حل‌های دیگری نیز برای این مساله ارائه شده که با توجه به عدم تمرکز این پژوهش بر روی این مساله، در این‌جا به آن‌ها پرداخته نشده و خوانندگان گرامی به (Amari 1999، LeCun and Bottou 1998، Masulli and Valentini 2000، Vapnik 1998 و Bradley and Mangasarian 1998) ارجاع داده می‌شوند.

روش PSVM برای دسته‌بندی داده‌های چند دسته‌ای

در روش PSVM یافتن ابرصفحه جداکننده بهینه برای هر ترکیب دوتایی از k دسته موجود انجام می‌شود. به عبارت دیگر، در این روش k(k-1)/2 دسته‌بندی انجام می‌شود. برای جداسازی دسته i ام از دسته j ام، تنها از داده‌های مربوط به این دو دسته استفاده می‌شود. داده‌های دسته i ام متناظر با $y_t = 1$ و داده‌های مربوط به دسته j ام متناظر با $y_t = -1$ در نظر گرفته می‌شود. به عبارت دیگر برای داده متعلق به یکی از دو دسته i یا j، رابطه زیر تعریف می‌شود:

$$y_t = \begin{cases} 1 & \bar{y}_t = i \\ -1 & \bar{y}_t = j \end{cases} \qquad t = 1, \dots, M \qquad (۹-۳)$$

سپس جداسازی به وسیله یکی از روش‌های SVM انجام می‌شود. این کار برای هر ترکیب دوتایی از k دسته انجام می‌شود.

تعریف. فرض کنید $w_{ij} \in R^m$ و $b_{ij} \in R$ پارامترهای ابرصفحه بهینه جداکننده دو دسته i و j برای مجموعه داده‌های k دسته‌ای $(k > 2)$ $s = \{(x_i, \bar{y}_i)\}_{i=1}^{M}$ باشند که به وسیله یکی از روش‌های دسته‌بندی SVM دو دسته‌ای به دست آمده‌اند و $x_i \in R^m (i = 1,2, \dots, M)$ و $\bar{y}_i \in \{1,2, \dots, k\}$. در این صورت تابع تصمیم جداسازی دو دسته i و j از یکدیگر به صورت زیر در نظر گرفته می‌شود:

$$D_{ij}(x) = {w_{ij}}^T x + b_{ij} \qquad i, j = 1,2, \dots, k, i \neq j \qquad (۱۰-۳)$$

با توجه به تعریف تابع تصمیم همواره $D_{ij}(x) = -D_{ji}(x)$ خواهد بود.

بنابراین، در روش PSVM برای یک مجموعه داده k دسته‌ای تعداد k(k-1)/2 تابع تصمیم به دست آمده که دسته‌ها را دو به دو از یکدیگر جداسازی می‌کند به طوری که اگر $D_{ij}(x) > 0$، داده x نسبت به دسته jام در دسته iام قرار دارد. هدف مساله یافتن دسته یک داده نسبت به کل دسته‌هاست.

تابع تصمیم در روش PSVM به صورت زیر است:

$$D_i(x) = \sum_{i=1, i \neq j}^{k} \operatorname{sgn}(D_{ij}(x)) \qquad (۱۱-۳)$$

که

$$\operatorname{sgn}(z) = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases} \qquad (۱۲-۳)$$

در این صورت اگر r در رابطه زیر دارای جواب یکتا i باشد، x متعلق به دسته i خواهد بود. در غیر این صورت، x با این روش قابل دسته‌بندی نیست:

$$\max_{r=1,2,\dots,k} D_r(x). \qquad (۱۳-۳)$$

فصل چهارم

دسته‌بند پیشنهادی برای تجزیه رشته‌های مرجع در زبان فارسی

در این فصل، جزئیات روش پیشنهادی را مورد بررسی قرار خواهیم داد. در این راستا، در ابتدا، جزئیات مجموعه داده ایجاد شده جهت آموزش و آزمایش دسته‌بند پیشنهادی توصیف خواهد شد. در ادامه، مجموعه ویژگی‌های در نظر گرفته شده در دسته‌بند نهایی بیان شده و سپس، نحوه استفاده از دسته‌بند پیشنهادی ارائه خواهد شد. در نهایت، نتایج ارزیابی روش پیشنهادی و نمونه‌هایی از خروجی‌های آن مشاهده خواهد شد.

۴-۱- ایجاد مجموعه‌ای برچسب‌گذاری شده از رشته‌های مرجع در زبان فارسی

همانطور که در فصل قبل مشاهده کردیم، وجود یک مجموعه داده برچسب‌گذاری شده از ابزارهای ضروری در طراحی روش‌های تجزیه رشته‌های مرجع است. علاوه بر استفاده از این مجموعه داده برای بررسی کیفیت تمام روش‌ها، در روش‌های مبتنی بر یادگیری ماشین از آن برای آموزش روش نیز استفاده می‌شود. با توجه به عدم وجود چنین مجموعه داده‌ای برای زبان فارسی و لزوم دسترسی به چنین داده‌ای برای انجام این پژوهش، به عنوان بخشی از مشارکت‌های این پژوهش مجموعه داده‌ای از رشته‌های مرجع برچسب‌گذاری شده برای زبان فارسی ایجاد شده است. برای انجام این کار، از مقالات ۵ سال اخیر نشریات علمی پژوهشی نمایه شده در پایگاه سیویلیکا در زمینه‌های علوم اجتماعی و انسانی استفاده شده و با استخراج تقریبی ۳ رشته مرجع تصادفا انتخاب شده از مقاله‌ای تصادفا انتخاب شده از هر سال هر یک از این نشریات، مجموعه داده‌ای شامل ۹۲۲ رشته مرجع ایجاد شده است. رشته‌های مرجع برچسب‌گذاری شده معادل هر یک از این رشته‌های مرجع استخراج شده نیز به صورت دستی توسط خبره ایجاد و در یک فایل bibtex ذخیره شده است. لازم به ذکر است که رشته‌های مرجع استخراج شده بدون هیچ تغییری در مجموعه داده ایجاد شده ذخیره شده و بعضا دارای اشتباهات نوشتاری نیز می‌باشند. علاوه بر این، با توجه به وجود منابع عربی در مراجع مقالات فارسی، مجموعه داده ایجاد شده شامل مراجعی به زبان عربی نیز می‌باشد.

در ادامه، انواع و تعداد موجود از هر نوع رشته مرجع موجود در مجموعه داده ایجاد شده را بررسی خواهیم کرد. انواع رشته‌های مرجع موجود در مجموعه داده ایجاد شده عبارت است از:

- مقاله چاپ شده در نشریه (۴۰۹ مورد)

- مقاله چاپ شده در همایش (۶۱ مورد)
- پایان‌نامه یا رساله دانشگاهی (۸۴)
- کتاب (۳۶۸ مورد)

در ادامه نمونه‌هایی از رشته‌های مرجع برچسب‌گذاری شده را خواهیم دید.

رجبی، م.، منصوریان، ع.، طالعی، م.، ۱۳۹۰، مقایسهٔ روش‌های تصمیم‌گیری چندمعیارهٔAHP ، AHP_OWAو AHP_OWA Fuzzey برای مکان‌یابی مجتمع‌های مسکونی در شهر تبریز، محیط‌شناسی، دوره ۳۷، شماره ۵۷، بهار ۱۳۹۰، صص. ۹۲- ۷۷.

```
@article{article,
author = {رجبی، م.، منصوریان، ع.، طالعی، م.},
title = {مقایسهٔ روش‌های تصمیم‌گیری چندمعیارهٔ AHP_OWA Fuzzey و AHP_OWA، AHP برای مکان‌یابی مجتمع‌های مسکونی در شهر تبریز},
journal = {محیط‌شناسی},
year = 1390,
number = 57,
pages = {92- 77},
month = بهار,
volume = 37
}
```

موحد، علی. ۱۳۸۶. گردشگری شهری، انتشارات دانشگاه شهید چمران اهواز، چاپ اول.

```
@book{book,
author = {موحد، علی},
year = ۱۳۸۶,
title = {گردشگری شهری},
publisher = {انتشارات دانشگاه شهید چمران اهواز},
note = {سری اول}
}
```

یاری، ارسطو. ۱۳۹۰. سنجش و ارزیابی سکونت-گاه‌های روستایی حوزه-ی کلان‌شهری و ارائه- مدل استراتژیک توسعه-ی پایدار (مطالعه موردی: حوزه-ی روستایی کلان‌شهر تهران). رساله-ی دکتری جغرافیا و برنامه-ریزی روستایی، استاد راهنما دکتر بدری، دانشگاه تهران، دانشکده جغرافیا

@phdthesis{phdthesis,
author = {یاری، ارسطو},
title = {مدل استراتژیک سنجش و ارزیابی سکونت‌گاه‌های روستایی حوزه‌ی کلان‌شهری و ارائه توسعه‌ی پایدار (مطالعه موردی: حوزه‌ی روستایی کلان‌شهر تهران)},
school = {دانشگاه تهران، دانشکده جغرافیا},
year = 1390,
note = {رساله‌ی دکتری جغرافیا و برنامه‌ریزی روستایی، استاد راهنما دکتر بدری}
}

گلوی، میترا. نجفی، رامین (۱۳۹۳). انگیزش، موتور محرک یادگیری. نخستین همایش ملی علوم تربیتی و روان‌شناسی.

@conference{conference,
author = {گلوی، میترا. نجفی، رامین},
title = {انگیزش، موتور محرک یادگیری},
booktitle = {نخستین همایش ملی علوم تربیتی و روان‌شناسی}
year = 1393,
}

دشتی برمکی، مجید؛ محسن رضایی؛ جواد اشجاری (۱۳۹۴) پنانسیل‌یابی منابع اب کارست کوههای دوان و شاپور براساس تصمیم‌گیری چندمعیاره.مجله‌پژوهش اب ایران. شماره ۱. پیاپی ۱۶. صفحات ۸۹-۱۰۰.

@article{article,
author = {دشتی برمکی، مجید؛ محسن رضایی؛ جواد اشجاری},
title = {پنانسیل‌یابی منابع اب کارست کوههای دوان و شاپور براساس تصمیم‌گیری چندمعیاره},
journal = {مجله‌پژوهش اب ایران},
year = 1394,
number = 16,
pages = {100-89},

volume = 1

}

۲. الوسی، سید محمود؛ روح المعانی فی تفسیر القران العظیم، بیروت، دار احیا التراث العربی، ۱۴۱۵ق.

@book{book,
author = {الوسی، سید محمود},

year = 1415,

title = {روح المعانی فی تفسیر القران العظیم},

publisher = {دار احیا التراث العربی},

address = {بیروت},

}

عیسی‌زاده، سهید؛ بلالی، اسماعیل و علی‌محمد قدسی (۱۳۸۹)، «تحلیل اقتصادی طلاق: بررسی ارتباط بیکاری و طلاق در ایران طی دوره ۱۳۸۵-۱۳۴۵»، فصلنامه مطالعات راهبردی زنان، سال سیزدهم، شماره ۵۰، صفحات ۲۸-۷.

@article{article,
author = {عیسی‌زاده، سهید؛ بلالی، اسماعیل و علی‌محمد قدسی},

title = {تحلیل اقتصادی طلاق: بررسی ارتباط بیکاری و طلاق در ایران طی دوره ۱۳۸۵-۱۳۴۵},

journal = {فصلنامه مطالعات راهبردی زنان},

year = 1389,

number = 50,

pages = {28-7},
volume = سیزدهم
}

جابری، محمدعابد (۱۳۷۶-۱۳۷۷). «بازگشت به اخلاق»، نقد و نظر، ترجمه حسین محبوب، س ۴، ش ۱۳-۱۴.

@article{article,

author = {جابری، محمدعابد},

title = {بازگشت به اخلاق},

journal = {نقد و نظر},

year = 1376-1377,

number = 13-14,
note = {ترجمه حسین محبوب},
volume = 4
}

۴-۲- استخراج ویژگی‌ها

در این قسمت مجموعه ویژگی‌های در نظر گرفته شده در دسته‌بند پیشنهادی توصیف خواهد شد. ویژگی‌های در نظر گرفته شده با توجه به کارهای پیشین انجام شده در زمینه تجزیه رشته‌های مرجع در زبان انگلیسی، بررسی مجموعه رشته‌های مرجع آماده شده در بخش اول این فصل و ویژگی‌های زبان فارسی به دست آمده است.

ویژگی‌های در نظر گرفته شده عبارتند از:

۱- موقعیت واحد در رشته مرجع: برای مثال در رشته مرجع زیر مقدار این ویژگی برای واحدهای "نصراله" و "رمزگذاری" به ترتیب برابر با ۳ و ۸ می‌باشد:

مهناز نوروزی، نصراله پاک نیت، زیبا اسلامی، رمزگذاری کلید عمومی با قابلیت جستجوی کلیدواژه: ارایه یک ساخت کلی امن در برابر حملات حدس کلیدواژه برخط و غیربرخط، چهاردهمین کنفرانس بین المللی انجمن رمز ایران، شهریور ۱۳۹۶، شیراز، ایران.

۲- آیا یکی از علائم نقطه‌گذاری {"،"، "."، "؛"، "("، ")"، "\"، "»"، "«"، "؟"، "?"، ":"} قبل از واحد وجود دارد؟

۳- آیا یکی از علائم نقطه‌گذاری فوق پس از واحدوجود دارد؟

۴- آیا واحد در مجموعه مشخص‌کننده‌های شماره صفحه {"ص"، "صص"، "صفحات"، "صفحه"، "صفحه‌های} است؟

برای در نظر گرفتن حالاتی که در آن بین مشخص‌کننده شماره صفحه و شماره صفحه فاصله‌ای وجود ندارد، هر تعداد عدد و کاراکتر "–" در انتهای واحدوجود داشته باشند حذف شده و سپس عضویت در مجموعه فوق بررسی می‌شود.

۵- آیا واحد در قالب شماره صفحه است؟ بدین منظور در صورتی که واحد با مشخص‌کننده‌های شماره صفحه آغاز شده باشد، مشخص‌کننده از ابتدای آن حذف شده و در ادامه در صورتی که واحد شامل "-" و تعدادی عدد باشد، به عنوان واحدی در قالب شماره صفحه در نظر گرفته می‌شود.

۶- آیا واحد یک عدد است؟

۷- تعداد واحدهای عددی مشاهده شده قبل از واحد فعلی.

۸- آیا واحد نمایانگر سال است؟ در اینجا عددی به عنوان سال در نظر گرفته می‌شود که ۴ رقمی بوده و بین ۱۰۰۰ تا ۲۰۲۰ باشد.

۹- آیا واحد نمایانگر نام یک نویسنده (مخفف نام نویسنده) است؟ بدین منظور حضور واحد در مجموعه {"ا."، "آ."، "ب."، "پ."، "ت."، "ث."، "ج."، "چ."، "ح."، "خ."، "د."، "ذ."، "ر."، "ز."، "ژ."، "س."، "ش."، "ص."، "ض."، "ط."، "ظ."، "ع."، "غ."، "ف."، "ق."، "ک."، "گ."، "ل."، "م."، "ن."، "و."، "ه."، "ی."} مورد بررسی قرار می‌گیرد.

۱۰-آیا واحد مختوم به یکی از پسوندهای نام خانوادگی رایج در زبان فارسی است؟ مجموعه پسوندهای در نظر گرفته شده عبارتند از {"پور"، "فرد"، "نیا"، "زاده"، "نژاد"، "راد"، "فر"، "ان"، "کیا"، "پناه"، "منش"}.

۱۱-آیا واحد شامل کاراکتری غیر از کاراکترهای متناظر با حروف فارسی است؟

۱۲-آیا به طور همزمان هم حروف الفبای فارسی و هم عدد در واحد موجود می‌باشد؟

۱۳-آیا واحد تنها شامل حروف الفبای فارسی و عدد است؟

۱۴-آیا واحد شامل مشخص‌کننده‌های دوره و شماره است؟ مجموعه مشخص‌کننده‌های دوره در نظر گرفته شده عبارت‌اند از {"دوره"، "دوره‌ی"، "سال"، "س"، "د"، "شماره"، "شماره‌ی"، "ش"}. برای در نظر گرفتن حالاتی که در آن بین مشخص‌کننده دوره و شماره

دوره فاصله‌ای وجود ندارد، در صورت وجود عدد در انتهای واحد آن را حذف کرده و سپس عضویت در مجموعه فوق بررسی می‌شود.

۱۵-آیا واحد شامل مشخص‌کننده شماره چاپ است؟ بدین منظور، در ابتدا در صورت وجود عدد در انتهای واحد آن را حذف کرده و سپس وجود واحد در مجموعه {"چاپ"، "چ"، "چ."} بررسی می‌شود.

۱۶-آیا واحد شامل مشخص‌کننده شماره جلد است؟ بدین منظور، در ابتدا در صورت وجود عدد در انتهای واحد آن را حذف کرده و سپس وجود واحد در مجموعه زیر {"جلد"، "ج"، "ج."} بررسی می‌شود.

۱۷-آیا واحد نام یک ماه یا فصل است؟ مجموعه ماه‌ها و فصل‌های یک سال، به طور واضح، به صورت {"فروردین"، "اردیبهشت"، "خرداد"، "تیر"، "مرداد"، "شهریور"، "مهر"، "آبان"، "ابان"، "آذر"، "دی"، "بهمن"، "اسفند"، "بهار"، "تابستان"، "پاییز"، "زمستان" در نظر گرفته شده است.

۱۸-آیا واحد عددی نوشته شده است؟ بدین منظور از نگارش فارسی اعداد کمتر از صد استفاده شده است.

۱۹-آیا واحد شامل حداقل یک حرف از زبان انگلیسی (چه بزرگ، چه کوچک) است؟

۲۰-آیا واحد در مجموعه {"دیگران" و "همکاران"} است؟

۲۱-آیا واحد در مجموعه {"دانشگاه"، "پژوهشگاه"، "دانشکده"، "پژوهشکده"، "پیام"، "نور"، "آزاد"، "گروه"، "استان"، "ایران"، "شهید"} است؟

۲۲-آیا واحد در مجموعه {"ترجمه"، "ترجمه‌ی"، "مترجم"، "به‌کوشش"، "بامقدمه"، "بامقدمه‌ی"، "تصحیح"} است؟

۲۳-آیا واحد در مجموعه {"استاد"، "راهنما"، "پایان‌نامه"، "رساله"، "کارشناسی"، "کارشناسی‌ارشد"، "دکتری"، "دکترا"، "رشته"، "رساله‌ی"، "راهنمایی"} است؟

۲۴-آیا واحد در مجموعه {"انتشارات"، "نشر"، "ناشر"، "الانتشار"، "چاپ"، "جهاد"} است؟

۲۵-آیا واحد در مجموعه {"تهران"، "اصفهان"، "شیراز"، "تبریز"، "یزد"، "بیروت"، "نجف"، "قم"} است؟

۲۶-آیا واحد در مجموعه {"فصلنامه"، "پژوهشنامه"، "مجله"، "دوفصلنامه"، "نشریه"، "نامه"، "دوماهنامه"، "ماهنامه"، "پژوهش"، "تحقیقات"، "تازه‌های"، "مطالعات"} است؟

۲۷-آیا واحد در مجموعه { "همایش"، "کنفرانس"، "ایران"، "ملی"، "بین‌المللی"، "کنگره"، "کنگره‌ی"، "سمینار"، "انجمن"، "اولین"، "نخستین"، "یکمین"، "دومین"، "سومین"، "چهارمین"، "پنجمین"، "ششمین"، "هفتمین"، "هشتمین"، "نهمین"، "دهمین"، "یازدهمین"، "دوازدهمین"، "سیزدهمین"، "چهاردهمین"، "پانزدهمین"، "شانزدهمین"، "هفدهمین"، "هجدهمین"، "نوزدهمین"، "خلاصه"، "مقالات"} قرار دارد؟

۲۸-آیا واحد در مجموعه {"و"، "الی"، "تا"} است؟

۲۹-آیا واحد در مجموعه اسامی نویسندگان (استخراج شده با خزش[1] از پایگاه سیویلیکا) قرار دارد؟

۳۰-امتیاز اختصاص داده شده به واحد (بین یک تا ۱۰) با توجه به فراوانی حضور واحد در عنوان نشریات (به دست آمده با خزش در پایگاه‌های https://www.civilica.com و http://isc.gov.ir).

۳۱-امتیاز اختصاص داده شده به واحد (بین یک تا ۱۰) با توجه به فراوانی حضور واحد در عنوان همایش‌ها (به دست آمده با خزش در پایگاه‌های https://www.civilica.com و http://isc.gov.ir).

۳۲-امتیاز اختصاص داده شده به واحد (بین یک تا ۱۰) با توجه به فراوانی حضور واحد در عنوان مقالات چاپ شده تا به امروز (به دست آمده با خزش در پایگاه https://www.civilica.com).

۳۳-آیا پس از حذف مشخص‌کننده‌های دوره، شماره نشر، شماره جلد و شماره چاپ از ابتدای واحد (در صورت وجود) تنها عددی کمتر از ۱۰۰ باقی می‌ماند؟

۳۴-برچسب سومین واحد قبل از واحد فعلی. در صورت عدم وجود چنین واحدی در رشته مرجع از null استفاده می‌شود.

[1] Crawl

۳۵-برچسب دومین واحد قبل از واحد فعلی. در صورت عدم وجود چنین واحدی در رشته مرجع از null استفاده می‌شود.

۳۶-برچسب واحد قبل از واحد فعلی. در صورت عدم وجود چنین واحدی در رشته مرجع از null استفاده می‌شود.

۳۷-تا ۶۶- ویژگی‌های شماره ۴ تا ۳۳ بیان شده در بالا برای واحد بعدی در دنباله واحدهای رشته مرجع موردنظر. در صورتی که واحد موردنظر آخرین واحد در دنباله واحدها باشد از null برای تمام ویژگی‌ها استفاده خواهد شد.

لازم به ذکر است که از بین ویژگی‌های ذکر شده در بالا، ویژگی‌های شماره ۱۰، ۱۱، ۱۸، ۱۹، ۲۵ و ۲۸ با توجه به زبان فارسی در این پژوهش معرفی و مورد استفاده قرار گرفته‌اند. ویژگی‌های شماره ۳۴، ۳۵ و ۳۶ نیز برای اولین بار در این پژوهش معرفی شده‌اند اما برای تجزیه رشته‌های مرجع در هر زبانی می‌توانند مورد استفاده قرار گیرند.

۳-۴- جزئیات روش پیشنهادی

در روش ارائه شده، در ابتدا مجموعه‌ای از ویرایش‌ها روی رشته‌های مرجع منبع انجام می‌شود.مهم‌ترین ویرایش‌ها انجام شده عبارتند از:

- اصلاح نویسه‌های فارسی: با توجه به وجود چندین نویسه با ظاهر یکسان برای برخی از حروف فارسی، یکی از پیش‌پردازش‌های لازم برای متن‌کاوی در زبان فارسی، اصلاح و یکسان‌سازی نویسه‌ها است که این مهم در روش ارائه شده توسط الگوریتم Virayesh(.) صورت گرفته است.
- تبدیل برخی فاصله‌ها به نیم‌فاصله: از دیگر عوامل تاثیرگذار برای متن‌کاوی در زبان فارسی، فاصله‌ها و نیم فاصله‌ها هستند و استفاده نادرست از آن‌ها منجر به کاهش کیفیت الگوریتم‌های متن‌کاوی خواهد شد. با توجه به این مهم، در الگوریتم Virayesh(.)، علاوه بر اصلاح نویسه‌ها، فاصله و نیم‌فاصله‌ها را در کلمات تاثیرگذار نیز اصلاح می‌کنیم.

در ادامه، با ذکر یک مثال عملکرد الگوریتم ویرایش روی یک رشته مرجع ورودی را بررسی خواهیم کرد. در مثال ذکر شده، با توجه به عدم امکان نمایش تصویری اصلاح نویسه‌ها، تغییرات انجام شده در این راستا، با کشیدن خطی به زیر آن مشخص شده است.

- رشته ورودی: ملکی، سعید. ۱۳۹۰. سنجش توسعه-ی پایدار در نواحی شهری با استفاده از تکنیک‌های برنامه‌ریزی (مطالعه-ی موردی: شهر ایلام)، مجله-ی جغرافیا و برنامه‌ریزی، شماره ۲۱، بهار، صفحات ۱۳۶-۱۱۷.
- رشته خروجی: ملکی، سعید. ۱۳۹۰. سنجش توسعه‌ی پایدار در نواحی شهری با استفاده از تکنیک‌های برنامه‌ریزی (مطالعه‌ی موردی: شهر ایلام)، مجله‌ی جغرافیا و برنامه‌ریزی، شماره ۲۱، بهار، صفحات ۱۳۶-۱۱۷.

پس از ویرایش رشته‌های مرجع، هر رشته مرجع به دنباله‌ای از واحدها تقسیم‌بندی می‌شود که در این پژوهش از هر لغت به عنوان یک واحد استفاده شده است. سپس، بردار ویژگی‌های متناظر با هر واحد محاسبه می‌شود. در ادامه، پس از استخراج بردار ویژگی متناظر با هر واحد هر رشته مرجع موجود در مجموعه داده آموزش، دسته‌بند SVM با استفاده از این داده‌ها آموزش داده می‌شود. در ادامه برای تجزیه یک رشته مرجع جدید به صورت زیر عمل می‌شود:

- برای هر واحد موجود در رشته مرجع (به ترتیب از ابتدا تا انتها):
  - همه مولفه‌های بردار ویژگی متناظر با آن به جز مولفه‌های ۳۴، ۳۵ و ۳۶ محاسبه می‌شود.
  - در صورتی که مقدار مولفه اول بردار ویژگی واحد مورد بررسی برابر با ۱ باشد، مقادیر مولفه‌های ۳۴، ۳۵ و ۳۶ آن برابر با null قرار داده می‌شود.
  - در غیر این صورت، مقدار مولفه ۳۴ واحد فعلی برابر با مقدار مولفه ۳۵ بردار ویژگی واحد ماقبل، مقدار مولفه ۳۵ واحد فعلی برابر با مقدار مولفه ۳۶ بردار ویژگی واحد ماقبل و مقدار مولفه ۳۶ واحد برابر با برچسب اختصاص یافته به واحد ماقبل قرار می‌گیرد.
  - با استفاده از دسته‌بند آموزش دیده، برچسب واحد فعلی به دست آمده و علاوه بر ذخیره، در بردارهای ویژگی واحدهای بعد مورد محاسبه قرار می‌گیرد.

- با استفاده از مجموعه‌ای از قواعد اکتشافی برچسب‌های اختصاص یافته به واحدها بهبود می‌یابند. قواعد استفاده شده به شرح زیر می‌باشد:
  - در صورت وجود یکی از واژگان "پایان‌نامه"، "رساله"، "رساله‌ی" در رشته مرجع، برچسب اختصاص یافته به واحدهایی که دسته‌بند به آن‌ها برچسب‌های انتشارات و همایش اختصاص داده است را به موسسه تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد برابر با موسسه باشد و واحد مورد نظر و واحدهای قبل از آن در رشته مرجع در مجموعه { "دانشگاه"، "پژوهشگاه"، "دانشکده"، "پژوهشکده"، "گروه"، "موسسه"} نباشند، برچسب واحد موردنظر برابر با برچسب واحد قبلی قرار داده می‌شود.
  - در صورتی که برچسب یک واحد عنوان و برچسب واحد ماقبل آن برابر همایش باشد، برچسب این واحد نیز به همایش تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد انتشارات یا غیره و برچسب واحد قبل و بعد آن برابر با یادداشت باشد، برچسب این واحد نیز به یادداشت تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد انتشارات یا یادداشت و برچسب واحد قبل و بعد آن برابر با موسسه باشد، برچسب این واحد نیز به موسسه تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد انتشارات یا همایش و برچسب واحد قبل و بعد آن برابر با یادداشت باشد، برچسب این واحد نیز به یادداشت تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد موسسه و برچسب واحد قبل و بعد آن برابر با همایش باشد، برچسب این واحد نیز به همایش تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد یادداشت و برچسب واحد قبل و بعد آن برابر با انتشارات باشد، برچسب این واحد نیز به انتشارات تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد نشریه، عنوان یا نام نویسنده و برچسب واحد قبل و بعد آن برابر با شماره باشد، برچسب این واحد نیز به شماره تغییر داده می‌شود.
  - در صورتی که برچسب یک واحد همایش و برچسب واحد قبل آن برابر با موسسه باشد، برچسب این واحد نیز به موسسه تغییر داده می‌شود.

- در صورتی که برچسب یک واحد موسسه و برچسب واحد قبل آن برابر با همایش باشد، برچسب این واحد نیز به همایش تغییر داده می‌شود.

۴-۴- ارزیابی روش پیشنهادی

برای ارزیابی، روش پیشنهادی با استفاده از زبان برنامه‌نویسی پایتون پیاده‌سازی شده است. در پیاده‌سازی انجام شده از کتابخانه sklearn برای استفاده از دسته‌بند SVM چند دسته‌ای خطی و با پارامترهای پیش‌فرض استفاده شده است. در این حالت، دسته‌بند خطی دو دسته‌ای از رویکرد L2-SVM استفاده کرده و با رویکرد یک دسته در مقابل سایر دسته‌ها به چند دسته‌ای گسترش می‌یابد. در ابتدا لازم به ذکر است که انتخاب این نوع از دسته‌بند SVM به دلایل زیر صورت گرفته است:

- مساله نیازمند به حل، مساله‌ای چند دسته‌ایست و ارزیابی‌های صورت گرفته نشان می‌دهد که نتایج بهتری در استفاده از رویکرد یک دسته در مقابل سایر دسته‌ها برای توسیع دسته‌بند دو دسته‌ای به چند دسته‌ای به دست می‌آید.
- کارایی دسته‌بند SVM خطی در مقایسه با نوع غیرخطی آن بهتر بوده و دسته‌بندی غیرخطی در صورتی استفاده می‌شود که استفاده از دسته‌بند خطی منجر به نتایج مناسبی نشود. با توجه به مناسب بودن نتایج در استفاده از نوع خطی (و عدم بهبود نتایج در استفاده از نوع غیرخطی)، در این پژوهش از دسته‌بند خطی با رویکرد L2-SVM استفاده شده است.

برای جزئیات بیشتر درباره نحوه پیاده‌سازی روش پیشنهادی، خوانندگان گرامی به پیوست که در برگیرنده کدهای نوشته شده برای روش پیشنهادی به زبان پایتون است، ارجاع داده می‌شوند.

در ادامه، در ابتدا نمونه‌هایی از نتایج اعمال روش پیشنهادی بر روی چند رشته مرجع آورده شده و سپس نتایج کلی ارزیابی روش پیشنهادی ارائه خواهد شد.

٤-٤-١- نمونه‌هایی از نتایج به دست آمده با استفاده از روش پیشنهادی

در ادامه، نمونه‌هایی از نتایج حاصل از اعمال دسته‌بند آموزش دیده پیشنهادی روی تعدادی رشته مرجع بیان خواهد شد. نمونه‌های ذکر شده شامل ارجاعاتی به مقالات چاپ شده در نشریه و همایش، پایان‌نامه دانشگاهی و کتاب است. در این نمونه‌ها، موارد اشتباها دسته‌بندی شده به صورت برجسته و

با کشیدن خط روی برچسب اشتباه مشخص شده‌اند. علاوه‌براین، برای خوانایی بیشتر، از نمادگذاری‌های زیر برای برچسب‌ها استفاده شده است:

A: نام نویسنده	T: عنوان
Y: سال	J: نشریه
N: شماره	PP: شماره صفحه
Pub: انتشارات	Add: آدرس
Oth: غیره	No: یادداشت
Sch: موسسه	C: همایش
M/S: ماه/فصل	

۱۶-نوغانی(A)، محسن. (A)، اصغرپور (A) ماسوله (A)، احمدرضا (A)، صفا (A)، شیما (A). و (A) کرمانی (A)، مهدی (A). ۱۳۸۸ (Y). کیفیت (T) زندگی (T) شهروندان (T) و (T) رابطه‌ی (T) ان (T) با (T) سرمایه (T) اجتماعی (T) در (T) شهر (T) مشهد (T)، مجله (J) علوم (J) اجتماعی (J) دانشکده (J) ادبیات (J) و (J) علوم (J) انسانی (J)، دانشگاه (Add) فردوسی (Pub) مشهد (Add)، بهار (M/S) و (M/S) تابستان (M/S).

۲.اسکندری (A)، سعیده (A) و (A) دیگران (A). ۱۳۹۲ (Y). ارزیابی (T) کارایی (T) مدل (T) دانگ (T) برای (T) تعیین (T) قابلیت (T) خطر (T) اتش‌سوزی (T) در (T) جنگل‌های (T)زرین‌اباد (T) نکا (T)، استان مازندران، فصل‌نامه (J) تحقیقات (J) جنگل (J) و (J) صنوبر (J) ایران (J) ، جلد (Oth) ۲۱ (N)، شماره (Oth) ۳ (N)، صص (Oth) ۴۳۹-(PP) ۴۵۱

مهناز (A) نوروزی (A)، نصراله (A) پاک (A) نیت(A) ، زیبا (A) اسلامی (A)، رمزگذاری (T) کلید (T) عمومی (T) با (T) قابلیت (T) جستجوی (T) کلیدواژه (T): ارایه (T) یک (T) ساخت (T) کلی (T) امن (T) در (T) برابر (T) حملات (T) حدس (T) کلیدواژه (T) برخط (T) و ~~(J)~~ غیربرخط ~~(C)~~، چهاردهمین (C) کنفرانس (C) بین المللی (C) انجمن (C) رمز (C) ایران (C)، شهریور (M/S) ۱۳۹۶ (Y)، شیراز (Add)، ایران (Add)

اصفهانی (A)، محمدمهدی (A). ۱۳۸۴ (Y). ایین (T) تندرستی (T). چاپ (No) اول (No). تهران (Add): اداره (Pub) ارتباطات (Pub) و (Pub) اموزش (Pub) سلامت (Pub) معاونت (Pub) سلامت (Pub) وزارت (Pub) بهداشت (Pub)، درمان (Pub) و (Pub) اموزش (Pub) پزشکی (Pub).

[۱۲] عدالت‌پناه (A)،ف. (A) و (A) رضازاده (A)،پ (A)، "پیش‌بینی (T) پارامترهای (T) موج (T) با (T) استفاده (T) از (T) مدل (T) SWAN(T)"، دوازدهمین (C) کنفرانس (C) دینامیک (C) شاره‌ها(C)، دانشگاه ~~(C)~~ ~~(C)~~نوشیروانی ~~(C)~~ بابل ~~(C)~~، ۱۳۸۸ (Y).

گایینی (A)، عباسعلی (A)، ابراهیم (A) علیدوست (A) قهفرخی (A)، علی (A) احمدی (A). (۱۳۸۸ (Y))، «تاثیر (T) مصرف (T) کوتاه‌مدت (T) مکمل (T) کراتین (T) بر (T) عملکردهای (T) سرعتی (T) و (T) قدرت (T) عضلانی (T)کشتی‌گیران (T)»، علوم ~~(T)~~ زیستی ~~(T)~~ و ~~(T)~~ ورزشی ~~(T)~~، شماره (Oth)۳(N): ص (Oth)۷۷ (PP) تا (PP) ۹۲ (PP).

[۳۱] ساسانیان (A)، سعید (A)؛ بررسی (T) اثر (T) تغییرات (T) PH (T) شیرابه (T) بر (T) خصوصیات (T) مکانیکی (T) پوششهای (T) رسی(T) (CCLها) (T)، پایان نامه (No)کارشناسی (No) ارشد (No)، دانشکده (Sch) عمران (Sch)، دانشگاه (Sch) صنعتی (Sch) شریف (Sch)،۱۳۸۳ (Y)

مارکس (A)، کارل (A). ۱۳۸۸ (Y). سرمایه (T): نقدی (T) بر (T) اقتصاد (T) سیاسی (T). ترجمه (No): حسن (No) مرتضوی (No)، تهران (Add) : نشر (Pub) اگاه (Pub).

٤-٤-۲- ارزیابی

برای ارزیابی روش ارائه شده 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 در این پژوهش شامل ۹۲۲ رشته مرجع می‌باشد که به هر واحد در هر رشته از این مجموعه داده یک برچسب از مجموعه برچسب‌های {نام نویسنده، عنوان، سال، نشریه، دوره، شماره، شماره صفحه، انتشارات، آدرس، غیره، یادداشت، موسسه، همایش، ماه/فصل} اختصاص یافته است. با توجه به آزمایش‌های صورت گرفته، استفاده از مجموعه برچسب فوق منجر به ایجاد اشتباهاتی در دسته‌های دوره و شماره می‌شود. ویژگی‌های مشترک دو برچسب دوره و شماره و استفاده از کلیدواژه‌های شماره برای مشخص کردن دوره در زبان فارسی دلیل این مساله است. با توجه به سهولت تفکیک این دو دسته از یکدیگر پس از دسته‌بندی، در دسته‌بند ارائه شده دسته‌های دوره و شماره با هم ترکیب و تحت عنوان شماره قرار گرفته‌اند.

با توجه به این تغییر، نتایج پیاده‌سازی دسته‌بند پیشنهادی روی مجموعه داده‌های موردنظر نشان می‌دهد که میانگین دقت، فراخوانی و F1 در روش ارائه شده برابر با ۹۲٪ است. این نتایج به صورت جزئی و برای هر برچسب به صورت مجزا در جدول شماره ۱ گزارش شده‌اند.

جدول ۱. نتایج ارزیابی روش پیشنهادی

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۷	۰,۹۸	۰,۹۸	۳۷۶۵
عنوان	۰,۹۲	۰,۹۶	۰,۹۴	۸۳۴۰
سال	۰,۹۸	۰,۹۸	۰,۹۸	۹۲۸
نشریه	۰,۸۷	۰,۸۱	۰,۸۴	۱۵۱۱
شماره	۰,۹۱	۰,۹۳	۰,۹۲	۶۶۲
شماره صفحه	۰,۹۴	۰,۹۶	۰,۹۵	۳۲۸
انتشارات	۰,۸۵	۰,۷۸	۰,۸۱	۱۱۷۲
آدرس	۰,۸۴	۰,۷۴	۰,۷۹	۴۷۰
غیره	۰,۹۸	۰,۹۲	۰,۹۵	۶۵۲
یادداشت	۰,۸۲	۰,۷۸	۰,۸۰	۱۴۲۷
موسسه	۰,۹۶	۰,۸۵	۰,۹۰	۴۹۹
همایش	۰,۷۴	۰,۸۱	۰,۷۷	۴۵۱
ماه/فصل	۰,۹۵	۰,۸۶	۰,۹۰	۸۰
میانگین	۰,۹۲	۰,۹۲	۰,۹۲	۲۰۲۸۵

برای بررسی میزان اثربخشی روش پیشنهادی بر روی انواع مختلف رشته‌های مرجع شامل مقالات چاپ شده در نشریات، مقالات چاپ شده در همایش‌ها، کتاب‌ها و پایان‌نامه‌ها و رساله‌های دانشگاهی، آزمایش‌هایی تنها با استفاده از رشته‌های مرجع با نوع مشخص به عنوان داده آزمایش صورت گرفته است. نتایج این آزمایشات که به ترتیب در جدول‌های شماره ۲ تا ۵ ارائه شده بیانگر این است که روش ارائه شده بهترین عملکرد را به ترتیب در تجزیه رشته‌های مرجع از نوع ارجاع به پایان‌نامه‌ها و رساله‌ها، ارجاع به مقالات چاپ شده در نشریات، ارجاع به مقالات چاپ شده در همایش‌ها و ارجاع به کتاب‌ها داشته است.

جدول ۲. نتایج ارزیابی روش پیشنهادی در حالتی که داده آزمایش تنها شامل ارجاع به مقالات چاپ شده در نشریات باشد.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۹	۰,۹۹	۰,۹۹	۲۱۴۳
عنوان	۰,۹۴	۰,۹۷	۰,۹۵	۴۵۸۱
سال	۰,۹۸	۰,۹۷	۰,۹۸	۴۱۴
نشریه	۰,۹۲	۰,۸۱	۰,۸۶	۱۵۰۹
شماره	۰,۹۶	۰,۹۴	۰,۹۵	۶۵۵
شماره صفحه	۰,۹۶	۰,۹۸	۰,۹۷	۲۹۵
آدرس	۰,۴۲	۰,۶۹	۰,۵۲	۱۶
غیره	۰,۹۹	۰,۹۷	۰,۹۸	۵۸۹
یادداشت	۰,۴۱	۰,۴۰	۰,۴۱	۱۱۷
ماه/فصل	۰,۹۶	۰,۹۰	۰,۹۳	۵۹
میانگین	۰,۹۵	۰,۹۴	۰,۹۴	۱۰۳۶۸

جدول ۳. نتایج ارزیابی روش پیشنهادی در حالتی که داده آزمایش تنها شامل ارجاع به مقالات منتشر شده در همایش‌ها باشد.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۱,۰۰	۱,۰۰	۱,۰۰	۳۶۴
عنوان	۰,۹۲	۰,۹۸	۰,۹۵	۷۲۷
سال	۰,۹۹	۱,۰۰	۰,۹۹	۶۶
شماره صفحه	۰,۷۱	۰,۹۴	۰,۸۱	۱۶
انتشارات	۰,۲۲	۰,۵۰	۰,۳۰	۱۴
آدرس	۰,۸۷	۰,۳۷	۰,۵۲	۱۱۰
غیره	۰,۹۰	۰,۵۴	۰,۶۸	۳۵
یادداشت	۰,۳۴	۰,۴۴	۰,۳۸	۳۲
همایش	۰,۸۶	۰,۸۱	۰,۸۳	۴۵۱
ماه/فصل	۰,۹۴	۰,۸۴	۰,۸۹	۱۹

میانگین	۰,۹۰	۰,۸۸	۰,۸۸	۱۸۳۷

جدول ٤. نتایج ارزیابی روش پیشنهادی در حالتی که داده آزمایش تنها شامل ارجاع به کتاب‌ها باشد.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۲	۰,۹۶	۰,۹۴	۱۰۶۸
عنوان	۰,۸۵	۰,۹۳	۰,۸۹	۱۸۷۴
سال	۰,۹۹	۰,۹۸	۰,۹۹	۳۶۶
شماره صفحه	۰,۸۶	۰,۷۱	۰,۷۷	۱۷
انتشارات	۰,۹۰	۰,۷۸	۰,۸۴	۱۱۵۸
آدرس	۰,۹۲	۰,۸۷	۰,۸۹	۳۲۲
غیره	۰,۷۱	۰,۳۷	۰,۴۹	۲۷
یادداشت	۰,۸۹	۰,۷۷	۰,۸۲	۹۳۰
میانگین	۰,۸۹	۰,۸۸	۰,۸۸	۵۷۷۰

جدول ٥. نتایج ارزیابی روش پیشنهادی در حالتی که داده آزمایش تنها شامل ارجاع به پارساها باشد.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۷	۰,۹۹	۰,۹۸	۱۹۰
عنوان	۰,۹۹	۰,۹۷	۰,۹۸	۱۱۵۸
سال	۰,۹۶	۱,۰۰	۰,۹۸	۸۲
آدرس	۰,۴۵	۰,۸۶	۰,۵۹	۲۲
یادداشت	۰,۸۷	۰,۹۵	۰,۹۱	۳۴۸
موسسه	۰,۹۹	۰,۸۵	۰,۹۲	۴۹۹
میانگین	۰,۹۶	۰,۹۴	۰,۹۵	۲۳۰۰

نتایج گزارش شده در جدول ۱، نتایج استفاده از تمام ویژگی‌های استخراج شده است. یکی از موارد تاثیرگذار در ویژگی‌های در نظر گرفته شده، برچسب واحدهای قبلی و بردار ویژگی متناظر با واحد

بعدی است. برای بررسی اثر این ویژگی‌ها در نتیجه نهایی، آزمایشاتی ۱- با حذف ویژگی‌های متناظر با واحد بعدی ۲- با حذف برچسب واحد سوم ماقبل، ۳- با حذف برچسب‌های واحدهای دوم و سوم ماقبل، ۴- با حذف برچسب‌های هر سه واحد ماقبل، از بردار ویژگی در نظر گرفته شده در روش پیشنهادی انجام شده که نتایج حاصل به ترتیب در جدول‌های ۶ تا ۹ نمایش داده شده‌اند.

نتایج به دست آمده از جدول ۶ بیانگر این است که ویژگی‌های متناظر با واحد بعدی تاثیر واضحی در تعیین برچسب یک واحد دارد. نتایج به دست آمده از جدول‌های ۷ و ۸ نشان‌دهنده این است که با حذف برچسب واحدهای سوم و یا برچسب واحدهای دوم و سوم از ویژگی‌های واحد فعلی در میانگین پارامترهای محاسبه شده تغییری ایجاد نشده اما در صورت استفاده از آن‌ها، مقدار پارامترهای ارزیابی برای برچسب‌های متناظر با فیلدهای طولانی (مانند نشریه، موسسه، همایش) حدود ۱٪ بهتر به دست می‌آید. با توجه به اهمیت این برچسب‌ها در نتایج، به نظر استفاده از این برچسب‌ها مفیدتر خواهد بود. نتایج به دست آمده از جدول ۵ نشان‌دهنده این است که در نظر نگرفتن برچسب واحدهای قبل در بردار ویژگی واحد منجر به کاهش ۱ درصدی در میانگین پارامتر دقت خواهد شد.

لازم به ذکر است که با افزایش تعداد برچسب‌های قبلی به ۴ نتایج شدیدا دچار افت شده و پارامترهای نهایی حدود ۵٪ کاهش داشته‌اند.

جدول ٦. نتایج ارزیابی روش پیشنهادی در صورت در نظر نگرفتن ویژگی‌های واحد بعدی در بردار ویژگی واحدها.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۴	۰,۹۹	۰,۹۶	۳۷۶۵
عنوان	۰,۹۲	۰,۹۳	۰,۹۳	۸۳۴۰
سال	۰,۹۹	۰,۹۸	۰,۹۸	۹۲۸
نشریه	۰,۷۸	۰,۸۴	۰,۸۲	۱۵۱۱
شماره	۰,۸۷	۰,۸۹	۰,۸۸	۶۶۲
شماره صفحه	۰,۹۳	۰,۹۰	۰,۹۲	۳۲۸
انتشارات	۰,۸۴	۰,۷۸	۰,۸۱	۱۱۷۲
آدرس	۰,۸۱	۰,۷۱	۰,۷۶	۴۷۰

غیره	۰,۹۸	۰,۹۰	۰,۹۴	۶۵۲
یادداشت	۰,۸۰	۰,۷۷	۰,۷۹	۱۴۲۷
موسسه	۰,۹۶	۰,۸۴	۰,۸۸	۴۹۹
همایش	۰,۶۹	۰,۶۹	۰,۶۹	۴۵۱
ماه/فصل	۰,۹۳	۰,۹۴	۰,۹۳	۸۰
میانگین	۰,۹۰	۰,۹۰	۰,۹۰	۲۰۲۸۵

جدول ۷. نتایج ارزیابی روش پیشنهادی در صورت در نظر نگرفتن برچسب سومین واحد ماقبل واحد فعلی در بردار ویژگی واحدها.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۷	۰,۹۸	۰,۹۸	۳۷۶۵
عنوان	۰,۹۴	۰,۹۴	۰,۹۴	۸۳۴۰
سال	۰,۹۹	۰,۹۸	۰,۹۹	۹۲۸
نشریه	۰,۸۰	۰,۸۸	۰,۸۴	۱۵۱۱
شماره	۰,۹۲	۰,۹۳	۰,۹۲	۶۶۲
شماره صفحه	۰,۹۴	۰,۹۵	۰,۹۵	۳۲۸
انتشارات	۰,۸۵	۰,۷۷	۰,۸۱	۱۱۷۲
آدرس	۰,۸۳	۰,۷۴	۰,۷۸	۴۷۰
غیره	۰,۹۸	۰,۹۳	۰,۹۵	۶۵۲
یادداشت	۰,۸۳	۰,۷۷	۰,۸۰	۱۴۲۷
موسسه	۰,۹۵	۰,۸۶	۰,۹۱	۴۹۹
همایش	۰,۷۶	۰,۸۱	۰,۷۸	۴۵۱
ماه/فصل	۰,۹۲	۰,۸۹	۰,۹۰	۸۰
میانگین	۰,۹۲	۰,۹۲	۰,۹۲	۲۰۲۸۵

جدول ۸. نتایج ارزیابی روش پیشنهادی در صورت در نظر نگرفتن برچسب دومین و سومین واحد ماقبل واحد فعلی در بردار ویژگی واحدها.

برچسب	دقت	فراخوانی	F1	تعداد

نام نویسنده	۰,۹۶	۰,۹۹	۰,۹۷	۳۷۶۵
عنوان	۰,۹۴	۰,۹۴	۰,۹۴	۸۳۴۰
سال	۰,۹۹	۰,۹۸	۰,۹۸	۹۲۸
نشریه	۰,۷۹	۰,۸۸	۰,۸۳	۱۵۱۱
شماره	۰,۹۱	۰,۹۳	۰,۹۲	۶۶۲
شماره صفحه	۰,۹۵	۰,۹۴	۰,۹۵	۳۲۸
انتشارات	۰,۸۸	۰,۷۸	۰,۸۲	۱۱۷۲
آدرس	۰,۸۳	۰,۷۴	۰,۷۸	۴۷۰
غیره	۰,۹۸	۰,۹۳	۰,۹۵	۶۵۲
یادداشت	۰,۸۴	۰,۷۸	۰,۸۱	۱۴۲۷
موسسه	۰,۹۵	۰,۸۷	۰,۹۱	۴۹۹
همایش	۰,۷۴	۰,۷۹	۰,۷۶	۴۵۱
ماه/فصل	۰,۹۲	۰,۸۹	۰,۹۰	۸۰
میانگین	۰,۹۲	۰,۹۲	۰,۹۲	۲۰۲۸۵

جدول ۹. نتایج ارزیابی روش پیشنهادی در صورت در نظر نگرفتن برچسب واحدهای قبل در بردار ویژگی واحدها.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۷	۰,۹۹	۰,۹۸	۳۷۶۵
عنوان	۰,۹۴	۰,۹۴	۰,۹۴	۸۳۴۰
سال	۰,۹۹	۰,۹۸	۰,۹۸	۹۲۸
نشریه	۰,۸۰	۰,۸۴	۰,۸۲	۱۵۱۱
شماره	۰,۹۱	۰,۹۳	۰,۹۲	۶۶۲
شماره صفحه	۰,۹۴	۰,۹۵	۰,۹۴	۳۲۸
انتشارات	۰,۸۵	۰,۷۷	۰,۸۱	۱۱۷۲
آدرس	۰,۸۰	۰,۷۴	۰,۷۶	۴۷۰
غیره	۰,۹۷	۰,۹۳	۰,۹۵	۶۵۲

یادداشت	۰,۸۳	۰,۷۸	۰,۸۱	۱۴۲۷
موسسه	۰,۹۴	۰,۸۳	۰,۸۸	۴۹۹
همایش	۰,۷۳	۰,۷۳	۰,۷۳	۴۵۱
ماه/فصل	۰,۹۱	۰,۹۰	۰,۹۱	۸۰
میانگین	۰,۹۱	۰,۹۱	۰,۹۱	۲۰۲۸۵

علاوه‌براین، برای بررسی میزان اثرگذاری پردازش‌های پسین انجام شده در روش ارائه شده، آزمایش یکسانی بدون استفاده از این پردازش‌های پسین بر روی داده‌ها انجام شد. نتایج به دست آمده، که در جدول ۱۰ گزارش شده است، نشان‌دهنده اثرگذاری زیاد این پردازش‌ها بر روی برچسب‌های موسسه و عنوان همایش است و در حالت کلی انجام این پردازش‌ها منجر به افزایش ۱ درصدی پارامترها شده است.

جدول ۱۰. نتایج ارزیابی روش پیشنهادی در صورت در نظر نگرفتن پردازش‌های پسین بر روی برچسب‌های اختصاص یافته به واحدها.

برچسب	دقت	فراخوانی	F1	تعداد
نام نویسنده	۰,۹۷	۰,۹۸	۰,۹۸	۳۷۶۵
عنوان	۰,۹۲	۰,۹۶	۰,۹۴	۸۳۴۰
سال	۰,۹۸	۰,۹۸	۰,۹۸	۹۲۸
نشریه	۰,۸۶	۰,۸۱	۰,۸۴	۱۵۱۱
شماره	۰,۹۲	۰,۹۳	۰,۹۲	۶۶۲
شماره صفحه	۰,۹۴	۰,۹۶	۰,۹۵	۳۲۸
انتشارات	۰,۸۳	۰,۷۸	۰,۸۱	۱۱۷۲
آدرس	۰,۸۶	۰,۷۴	۰,۸۰	۴۷۰
غیره	۰,۹۸	۰,۹۲	۰,۹۵	۶۵۲
یادداشت	۰,۸۳	۰,۷۵	۰,۷۹	۱۴۲۷
موسسه	۰,۶۵	۰,۷۲	۰,۶۸	۴۹۹
همایش	۰,۶۹	۰,۶۴	۰,۶۷	۴۵۱

ماه/فصل	۰,۹۴	۰,۸۵	۰,۹۱	۸۰
میانگین	۰,۹۱	۰,۹۱	۰,۹۱	۲۰۲۸۵

نتیجه‌گیری و کارهای آینده

در این پژوهش، در ابتدا به بررسی کارهای صورت گرفته در زمینه تجزیه رشته‌های مرجع در زبان انگلیسی پرداخته شد. بررسی‌های انجام شده نشان داد که روش‌های ارائه شده برای تجزیه رشته‌های مرجع را می‌توان به دو دسته کلی ۱- روش‌های مبتنی بر قاعده و ۲- روش‌های مبتنی بر یادگیری ماشین تقسیم‌بندی کرد که دسته دوم نتایج بسیار بهتری را ارائه کرده است. علاوه‌براین، از میان روش‌های یادگیری ماشین موجود، روش‌های HMM، CRF و SVM برای حل این مساله مورد استفاده قرار گرفته که استفاده از SVM و CRF منجر به دستیابی به نتایج بهتری شده است.

پس از بررسی پیشینه مساله، در این پژوهش، روشی مبتنی بر SVM برای تجزیه رشته‌های مرجع در زبان فارسی ارائه شد. برای طراحی روش ارائه شده در ابتدا مجموعه داده‌ای از رشته‌های مرجع برچسب‌گذاری شده در زبان فارسی تهیه گردید. سپس، بردارهای ویژگی متناظر با رشته‌های مرجع موجود در مجموعه داده تهیه شده ایجاد و دسته‌بند SVM با استفاده از آن آموزش داده شد. نتایج بررسی آزمایشات بر روی دسته‌بند به دست آمده نشان‌دهنده ۹۲٪ برای هر سه پارامتر دقت، فراخوانی و F1 می‌باشد.

از جمله کارهای قابل انجام برای ارتقا این پژوهش می‌توان به استفاده از دسته‌بندهای دیگر مانند ماشین بردار پشتیبان دوقلو[1]، CRF و یادگیری عمیق برای حل این مساله و مقایسه نتایج با نتایج به دست آمده در این پژوهش اشاره کرد. علاوه‌براین، همانطور که در مقدمه این پژوهش نیز ذکر شد، مساله

[1] Twin SVM

تجزیه رشته‌های مرجع مساله‌ای پایه‌ای برای بسیاری مسائل دیگر است که از آن جمله می‌توان به تطبیق رشته‌های مرجع در یک سطح بالاتر و ساخت خودکار شبکه‌های استنادی در حالت کلی اشاره کرد. با توجه به کیفیت مناسب روش ارائه شده، می‌توان از آن به عنوان ابزاری پایه‌ای برای حل این مسائل استفاده کرد که انشاالله در طرح‌های آتی به آن پرداخته خواهد شد.

مراجع

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

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

Lawrence, S., C. Lee Giles, and K. Bollacker (1999). Digital libraries and autonomous citation indexing. Computer 32(6); 67-71,

Lawrence, S., C. L. Giles, and K. D. Bollacker (1999). Autonomous citation matching. In Proceedings of the third annual conference on Autonomous Agents; 392-393,

Huang, I. A., J. M. Ho, H. Y. Kao, and W. C. Lin (2004). Extracting citation metadata from online publication lists using BLAST. In Pacific-Asia Conference on Knowledge Discovery and Data Mining; 539-548,

Gupta, D., B. Morris, T. Catapano, and G. Sautter (2009). A new approach towards bibliographic reference identification, parsing and inline citation matching. In International Conference on Contemporary Computing; 93-102,

Chen, C. C., K. H. Yang, H. Y. Kao, and J. M. Ho (2008). BibPro: A Citation parser based on sequence alignment techniques. In Advanced Information Networking and Applications-Workshops, 22nd International Conference on; 1175-1180,

Seymore, K., A. McCallum, R. Rosenfeld, (1999). Learning hidden Markov model structure for information extraction. In AAAI-99 workshop on machine learning for information extraction; 37-42,

Besagni, D., A. Belaïd, N. Benet (2003). A segmentation method for bibliographic references by contextual tagging of fields. In Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on; 384-388,

Ding, Y., G. Chowdhury, and S. Foo (1999). Template mining for the extraction of citation from digital documents. In Proceedings of the Second Asian Digital Library Conference, Taiwan; 47-62,

Vinkler, P. (1994). The origin and features of information referenced in pharmaceutical patents. Scientometrics, 30(1); 283-302,

Constantin, A., S. Pettifer, and A. Voronkov (2013). PDFX: fully-automated PDF-to-XML conversion of scientific literature. In Proceedings of the 2013 ACM symposium on Document engineering; 177-180,

Hetzner, E. (2008). A simple method for citation metadata extraction using hidden markov models. In Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries; 280-284,

Yin, P., M. Zhang, Z. Deng, and D. Yang (2004). Metadata extraction from bibliographies using bigram HMM. In International Conference on Asian Digital Libraries; 310-319,

Bikel, D. M., S. Miller, R. Schwartz, and R. Weischedel (1997). Nymble: a high-performance learning name-finder. In Proceedings of the fifth conference on applied natural language processing; 194-201,

Ojokoh, B., M. Zhang, and J. Tang (2011). A trigram hidden Markov model for metadata extraction from heterogeneous references. Information Sciences, 181(9); 1538-1551,

Lafferty, J., A. McCallum, and F. C. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data,

Peng, F., and A. McCallum (2013). Accurate information extraction from research papers using conditional random fields. Retrieved on April 13,

Councill, I. G., C. L. Giles, and M. Y. Kan (2008). ParsCit: an Open-source CRF Reference String Parsing Package. In LREC, 8; 661-667,

Zou, J., D. Le, and G. R. Thoma (2010). Locating and parsing bibliographic references in HTML medical articles. International Journal on Document Analysis and Recognition, 13(2); 107-119,

Zhang, Q., Y. G. Cao, and H. Yu (2011). Parsing citations in biomedical articles using conditional random fields. Computers in biology and medicine, 41(4); 190-194,

Kim, Y. M., P. Bellot, J. Tavernier, E. Faath, and M. Dacos (2012). Evaluation of BILBO reference parsing in digital humanities via a comparison of different tools. In Proceedings of the 2012 ACM symposium on Document engineering; 209-212,

Tkaczyk, D., P. Szostek, P. J. Dendek, M. Fedoryszak, and L. Bolikowski (2014). Cermine--automatic extraction of metadata and references from scientific

literature. In Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on; 217-221,

Tkaczyk, D., P. Szostek, M. Fedoryszak, P. J. Dendek, and L. Bolikowski, (2015). CERMINE: automatic extraction of structured metadata from scientific literature. International Journal on Document Analysis and Recognition (IJDAR), 18(4); 317-335,

Zhang, X., J. Zou, D. X. Le, and G. R. Thoma (2011). A structural SVM approach for reference parsing. BMC bioinformatics, 12(3); S7,

Agichtein, E., and V. Ganti (2004). Mining reference tables for automatic text segmentation. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining; 20-29,

Lopez, P. (2009). GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In International Conference on Theory and Practice of Digital Libraries; 473-474,

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In Proceedings of the international joint workshop on natural language processing in biomedicine and its applications; 104-107,

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

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

Liu, B. (1998). Minimax chance constrained programming models for fuzzy decision system. Information Science, 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), Brighton, UK, volume 2; 788–791,

Krebel, U. H. G. (1999). Pairwise classification and support vector machines. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods: Support Vector Learning; 255–68,

Amari, S., S. Wu (1999). An information-geometrical method for improving the performance of support vector machine classifiers. In Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN ’99), Edinburgh, UK, volume 1; 85–90,

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11); 2278–324,

Bradley, P. S., and O. L. Mangasarian (1998). Feature selection via concave minimization and support vector machines. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML ’98), Madison; 82–90,

پیوست

در این قسمت، بخش‌های مهم کدهای نوشته شده در پیاده‌سازی روش پیشنهادی آورده شده است. کدهای استفاده شده را می‌توان در ۳ بخش تقسیم‌بندی کرد که این ۳ بخش عبارتند از:

- کد مربوط به خزش در پایگاه داده‌های https://www.civilica.com و http://isc.gov.ir جهت استخراج مجموعه‌هایی از عناوین مقالات، اسامی نویسندگان و لیست نشریات و همایش‌های موجود در این پایگاه داده‌ها.
- کد مربوط به استخراج ویژگی‌های واحدهای هر رشته مرجع موجود در مجموعه داده آماده شده در فصل چهارم این پژوهش.
- کد مربوط به آموزش و آزمایش دسته‌بند SVM با استفاده از بردارهای ویژگی استخراج شده در مرحله قبل.

در ادامه و به ترتیب این کدها فراهم شده‌اند:

کد مربوط به خزش در پایگاه داده‌های https://www.civilica.com و http://isc.gov.ir جهت استخراج مجموعه‌هایی از عناوین مقالات، اسامی نویسندگان و لیست نشریات و همایش‌های موجود در این پایگاه داده‌ها.

```
import codecs
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import urllib.request  as urllib2
from bs4 import BeautifulSoup
def Ext_ISC_Conf():
    ints=list(range(1,226))
    ISC_Confs=""
    for i in ints:
        url="http://mcl.isc.gov.ir/confirmedConferences.aspx?p="+str(i)
        page = urllib2.urlopen(url)
        soup = BeautifulSoup(page)
        confs=soup.findAll('a',  href=True)
        for con in confs:
            xxx=con['href']
            if xxx.startswith("http://conf.isc.gov.ir"):
                try:
                    ISC_Confs=ISC_Confs+con.string+"---"
                except TypeError:
                    continue
        with codecs.open("E:/CN.txt", 'w', encoding = 'utf8') as file:
            file.write(ISC_Confs)
def Ext_ISC_jour():
    unichars=unichars=['%d8%a2','%d8%a7','%d8%a8','%d9%be','%d8%aa','%d8%ab','%d8%ac','%da%86','%d8%ad','%d8%ae','%d8%af','%d8%b0','%d8%b1','%d8%b2','%da%98','%d8%b3','%d8%b4','%d8%b5','%d8%b6','%d8%b7','%d8%b8','%d8%b9','%d8%ba','%d9%81','%d9%82','%da%a9','%da%af','%d9%84','%d9%85','%d9%86','%d9%88','%d9%87','%db%8c']
    ISC_Journals=""
    for c in unichars:
        url="http://ecc.isc.gov.ir/searchEjournalsByName/"+c
        page = urllib2.urlopen(url)
        soup = BeautifulSoup(page)
        journals=soup.findAll('a', style="color: #326598")
        for j in journals:
            ISC_Journals=ISC_Journals+j.string+"\n"
    with codecs.open("E:/CN.txt", 'w', encoding = 'utf8') as file:
        file.write(ISC_Confs)
def Ext_Civil_conf():
    ints=list(range(0,215))
    Civil_C=""
    for i in ints:
        url="https://www.civilica.com/Conferences-"+str(i*20)+"-20-Date-DESC-AI-.html"
        req=urllib2.Request(url)
        req.add_header('User-agent', 'Mozilla 5.0')
        page = urllib2.urlopen(req)
        soup = BeautifulSoup(page)
        Jnames=soup.findAll('h4')
        for Jname in Jnames:
            try:
                x=Jname.find('a')
                Civil_C=Civil_C+x['title']+"---"
            except Exception:
```

```
                pass
    with codecs.open("E:/CNCivil.txt", 'w', encoding = 'utf8') as file:
        file.write(Civil_C)
def Ext_Civil_jour():
    ints=list(range(0,16))
    Civil_J=""
    for i in ints:
        url="https://www.civilica.com/Journals---HS-"+str(i*20)+"-20-
FaName-DESC-AI-.html"
        req=urllib2.Request(url)
        req.add_header('User-agent', 'Mozilla 5.0')
        page = urllib2.urlopen(req)
        soup = BeautifulSoup(page)
        Jnames=soup.findAll('h4')
        for Jname in Jnames:
            try:
                x=Jname.find('a').find('b')
                Civil_J=Civil_J+x.string+"---"
            except Exception:
                pass
    with codecs.open("E:/JNCivil.txt", 'w', encoding = 'utf8') as file:
        file.write(Civil_J)
def Auth_Jour_Names():
    ints=list(range(0,32))
    for i in ints:
        url="https://www.civilica.com/Journals----"+str(i*20)+"-20-FaName-
DESC-AI-.html"
        req=urllib2.Request(url)
        req.add_header('User-agent', 'Mozilla 5.0')
        page = urllib2.urlopen(req)
        soup = BeautifulSoup(page)
        Civil_Jour_string=""
        Civil_Jour_Auth_Names=""
        Jnames=soup.findAll('h4')
        for Jname in Jnames:
            try:
                x=Jname.find('a')
                newurl="https://www.civilica.com/"+x['href']
                newurl=urllib2.quote(newurl)
                newurl=newurl.replace("%3A",":")
                newurl=newurl.replace("%3D","=")
                newreq=urllib2.Request(newurl)
                newreq.add_header('User-agent', 'Mozilla 5.0')
                newpage = urllib2.urlopen(newreq)
                newsoup = BeautifulSoup(newpage)
                Issuenames=newsoup.findAll('a')
                for Issuename in Issuenames:
                    try:
                        if "دوره" in Issuename.text and "شماره" in Issuename.t
ext:
                            nnewurl ="https://www.civilica.com/"+Issuename[
'href']
                            nnewurl=urllib2.quote(nnewurl)
                                nnewurl=nnewurl.replace("%3A",":")
                                nnewurl=nnewurl.replace("%3D","=")
                                nnewreq=urllib2.Request(nnewurl)
                                nnewreq.add_header('User-
agent', 'Mozilla 5.0')
                                nnewpage = urllib2.urlopen(nnewreq)
                                nnewsoup = BeautifulSoup(nnewpage)
                                paperlinks=nnewsoup.findAll('a')
                                for paperlink in paperlinks:
                                    try:
```

```
                                                if "Paper-
    " in paperlink['href']:
                                                    nnnewurl ="https://www.civil
    ica.com/"+paperlink['href']
                                                    nnnewurl=urllib2.quote(nnnew
    url)
                                                    nnnewurl=nnnewurl.replace("%
    3A",":")
                                                    nnnewurl=nnnewurl.replace("%
    3D","=")
                                                    nnnewreq=urllib2.Request(nnn
    ewurl)
                                                    nnnewreq.add_header('User-
    agent', 'Mozilla 5.0')
                                                    nnnewpage = urllib2.urlopen(
    nnnewreq)
                                                    nnnewsoup = BeautifulSoup(nn
    newpage)
                                                    titles=nnnewsoup.findAll('h1
    ')
                                                    for title in titles:
                                                        try:
                                                            if numberofEL(title.
    string)<10:
                                                                Civil_Jour_strin
    g=Civil_Jour_string+title.string+"---"
                                                        except Exception:
                                                            pass
                                                    authors=nnnewsoup.findAll('a
    ')
                                                    for author in authors:
                                                        try:
                                                            if "name=PaperSearch
    " in author['href'] and numberofEL(author.string)<2:
                                                                Civil_Jour_Auth_
    Names=Civil_Jour_Auth_Names+author.string+"---"
                                                        except Exception:
                                                            pass
                                            except Exception:
                                                pass
                                    except Exception:
                                        pass
                                except Exception:
                                    pass
                    try:
                        with codecs.open("E:/CivilJString.txt", 'r', encoding = 'utf
    8') as file:
                            text = file.read()
                    except Exception:
                        text=""
                        pass
                    Civil_Jour_string=Civil_Jour_string+"---"+text
                    with codecs.open("E:/CivilJString.txt", 'w', encoding = 'utf8')
    as file:
                        file.write(Civil_Jour_string)
                    try:
                        with codecs.open("E:/CivilJAuthNames.txt", 'r', encoding = '
    utf8') as file:
                            text = file.read()
                    except Exception:
                        pass
                    Civil_Jour_Auth_Names=Civil_Jour_Auth_Names+"---"+text
```

```
153.            with codecs.open("E:/CivilJAuthNames.txt", 'w', encoding = 'utf8
    ') as file:
154.                file.write(Civil_Jour_Auth_Names)
155.        def Auth_Conf_Names():
156.            ints=list(range(0,215))
157.            for i in ints:
158.                url="https://www.civilica.com/Conferences-"+str(i*20)+"-20-
    Date-DESC-AI-.html"
159.                req=urllib2.Request(url)
160.                req.add_header('User-agent', 'Mozilla 5.0')
161.                page = urllib2.urlopen(req)
162.                soup = BeautifulSoup(page)
163.                Civil_Conf_string=""
164.                Civil_Auth_Names=""
165.                Cnames=soup.findAll('h4')
166.                for Cname in Cnames:
167.                    try:
168.                        x=Cname.find('a')
169.                        newurl="https://www.civilica.com/"+x['href']
170.                        newurl=urllib2.quote(newurl)
171.                        newurl=newurl.replace("%3A",":")
172.                        newurl=newurl.replace("%3D","=")
173.                        newreq=urllib2.Request(newurl)
174.                        newreq.add_header('User-agent', 'Mozilla 5.0')
175.                        newpage = urllib2.urlopen(newreq)
176.                        newsoup = BeautifulSoup(newpage)
177.                        divs=newsoup.findAll('div',class_='item')
178.                        for div in divs:
179.                            try:
180.                                y=div.find('h4').find('a').find('b')

181.                                if numberofEL(y.string)<10:
182.                                    Civil_Conf_string=Civil_Conf_string+y.st
    ring+"---"
183.                            except Exception:
184.                                pass
185.                        links=newsoup.findAll('a')
186.                        for link in links:
187.                            try:
188.                                if "name=PaperSearch" in link['href'] and nu
    mberofEL(link.string)<2:
189.                                    Civil_Auth_Names=Civil_Auth_Names+link.s
    tring+"---"
190.                            except Exception:
191.                                pass
192.                    except Exception:
193.                        pass
194.            try:
195.                with codecs.open("E:/CivilConfString.txt", 'r', encoding = '
    utf8') as file:
196.                    text = file.read()
197.            except Exception:
198.                text=""
199.                pass
200.            Civil_Conf_string=Civil_Conf_string+"---"+text
201.            with codecs.open("E:/CivilConfString.txt", 'w', encoding = 'utf8
    ') as file:
202.                file.write(Civil_Conf_string)
203.            try:
204.                with codecs.open("E:/CivilAuthNames.txt", 'r', encoding = 'u
    tf8') as file:
205.                    text = file.read()
206.            except Exception:
```

```
207.                pass
208.            Civil_Auth_Names=Civil_Auth_Names+"---"+text
209.            with codecs.open("E:/CivilAuthNames.txt", 'w', encoding = 'utf8'
    ) as file:
210.                file.write(Civil_Auth_Names)
211.        def Names():
212.            Ext_ISC_Conf()
213.            print(1)
214.            Ext_ISC_jour()
215.            print(2)
216.            Ext_Civil_conf()
217.            print(3)
218.            Ext_Civil_jour()
219.            print(4)
220.            Auth_Conf_Names()
221.            Auth_Jour_Names()
222.        Names()
```

کد مربوط به استخراج ویژگی‌های واحدهای هر رشته مرجع موجود در مجموعه داده آماده شده در فصل چهارم این پژوهش

```
import logging
import os
import codecs
from random import randint
import math
def Extract_features(text,bibitem,jnames_scores,names,tw_scores,cnames_scor
es):
    templist=[]
    for i in list(range(3)):
        templist.append("0")
    rand_int=randint(0,9)
    text=Virayesh(text)
    while not text[0] in Alphabets:
        text=text[1:]
    tokens=text.split(" ")
    ttext=""
    for i in list(range(len(tokens))):
        tmp=tokens[i][:-1]
        if not is_f_name_indicator(tmp,Fnameindicators):
            tmp=replace_sep_with_space(tmp,seperators)
        if not hasNumbers(tmp):
            tmp=tmp.replace("-","")
        while tmp.startswith(" "):
            tmp=tmp[1:]
        while tmp.endswith(" "):
            tmp=tmp[:-1]
        ttext=ttext+tmp+tokens[i][-1:]+" "
    ttext=ttext.replace("  "," ")
    tokens=ttext.split(" ")
    currindex=1
    fvectors=[]
    flag=False
    numnumbers=0
    for i in list(range(len(tokens)-1)):
        token=tokens[i]
        if token in ["\r\n","\r","\n"]:
            continue
        previousindex=0
        fvector=[]
        fvector.append(format(currindex,'02d'))
        fvector.append(rand_int)
        currindex=currindex+1
        fvector.append(int(flag))
        fvector.append(int(is_ended_with_seperator(token,seperators)))
        if is_ended_with_seperator(token,seperators) and not is_f_name_indi
cator(token,Fnameindicators):
            flag=True
            token=token[:-1]
        else:
            flag=False
        fvector.append(int(contains_page_ind(token)))
        fvector.append(int(is_page_number(token,Alphabets)))
        fvector.append(int(is_number(token) or is_number(token[:-1])))
        if is_number(token) or is_number(token[:-1]):
            numnumbers=numnumbers+1
        fvector.append(int(is_number(token) or is_number(token[:-
1]))*numnumbers)
```

```
        fvector.append(int(is_year_number(token)))
        fvector.append(int(is_f_name_indicator(token,Fnameindicators)))
        fvector.append(int(contain_l_name_indicator(token,lnameindicators))
)
        fvector.append(int(is_contain_nonalphabetic(token, Alphabets)))
        fvector.append(int(is_contain_num_alph(token,Alphabets,numbers)))
        fvector.append(int(is_only_contain_num_alph(token,Alphabets,numbers
)))
        fvector.append(int(contains_volume_ind(token) or contains_num_ind(t
oken)))
        #fvector.append(int(is_it_series_indicator(token)))
        #fvector.append(int(is_it_part_indicator(token)))
        fvector.append(int(contains_series_ind(token)))
        fvector.append(int(contains_parts_ind(token)))
        fvector.append(int(is_mon_seas_name(token)))
        fvector.append(int(is_a_written_number(token)))
        fvector.append(int(is_containing_Eng_alphabet(token,engalph)))
        fvector.append(int(is_et_al(token)))
        fvector.append(int(token in ["دانشگاه","پژوهشگاه","دانشکده","پژوهشکده","پیام","
نور","آزاد","گروه","استان","ایران","شهید"]))
        fvector.append(int(token in ["ترجمه","ترجمه‌ی","مترجم","به‌کوشش","بامقدمه","
بامقدمه‌ی","تصحیح"] or token.startswith("ترجمه") or token.startswith("مترجم")))
        fvector.append(int(token in ["استاد","راهنما","پایان‌نامه","پایان‌نامه","رساله","
کارشناسی","کارشناسی‌ارشد","دکتری","دکترا","رشته","رساله‌ی","راهنمایی"]))
        fvector.append(int(token in ["انتشارات","نشر","ناشر","الانتشار","چاپ","جهاد"]))

        fvector.append(int(token in ["تهران","اصفهان","شیراز","تبریز","یزد","بیروت","
نجف","قم"]))
        fvector.append(int(is_j_ind(token)))
        fvector.append(int(is_c_ind(token)))
        fvector.append(int(token in ["و","تا", "الی"]))
        fvector.append(int(is_author_name(token,names)))
        fvector.append(format(j_name_score(token,jnames_scores),'04d'))
        fvector.append(format(c_name_score(token,cnames_scores),'04d'))
        fvector.append(format(title_name_score(token,tw_scores),'04d'))
        for i in list(range(len(templist))):
            fvector.append(0)
        fvector.append("{:<10}".format(token_Label(token,bibitem)))
        fvector.append(token)
        fvectors.append(fvector)
    return fvectors
def is_page_number(token,Alphabets):
      indicators=["ص","صص","صفحات","صفحه","صفحه‌های"]
      numbers=['0','1','2','3','4','5','6','7','8','9']
      tmp=token
      if not hasNumbers(tmp):
            return False
      if contains_page_ind(tmp):
            while not is_number(tmp[:1]):
                  tmp=tmp[1:]
      pnum=list(tmp)
      for pn in pnum:
            if not str(pn) in numbers+['']:
                        return False
             if '-' in pnum:
                   return True
             else:
                   return False
```

```
105.        Fnameindicators=["ا.",  "آ.", "ب.", "پ.", "ت.", "ث.", "ج.", "چ.", "
    ح.",
106.                         "خ.", "د.", "ذ.", "ر.", "ز.", "ژ.", "س.", "ش.", "ص.
    ",
107.                         "ض.", "ط.", "ظ.", "ع.", "غ.", "ف.", "ق.", "ک.", "
    گ.",
108.                         "ل.", "م.", "ن.", "و.", "ه.", "ی."]
109.        def Virayesh(text):
110.                          text=special_year_editting(text)
111.                          text=convertnumbs(text)
112.                          text=text.replace("ط","ط")
113.                          text=text.replace("ﭽ","ب")
114.                          text=text.replace("ﺎ","ا")
115.                          text=text.replace("ﻲ","ی")
116.                          text=text.replace("ی","ی")
117.                          text=text.replace("ئ","ی")
118.                          text=text.replace("ی","ی")
119.                          text=text.replace("ی","ی")
120.                          text=text.replace("ی","ی")
121.                          text=text.replace("ى","ی")
122.                          text=text.replace("ێ","ی")
123.                          text=text.replace("ي","ی")
124.                          text=text.replace("ﻱ","ی")
125.                          text=text.replace("أ","ا")
126.                          text=text.replace("أ","ا")
127.                          text=text.replace("إ","ا")
128.                          text=text.replace("آ","ا")
129.                          text=text.replace("ا","ا")
130.                          text=text.replace("ﹶ","ا")
131.                          text=text.replace("ﹷ","ا")
132.                          text=text.replace("ﺇ","ا")
133.                          text=text.replace("ﹰ","ا")
134.                          text=text.replace("ﺏ","ب")
135.                          text=text.replace("ﺑ","ب")
136.                          text=text.replace("ﺒ","ب")
137.                          text=text.replace("ﺐ","ب")
138.                          text=text.replace("ﭗ","پ")
139.                          text=text.replace("ۀ","ه")
140.                          text=text.replace("ۀ","ه")
141.                          text=text.replace("ه","ه")
142.                          text=text.replace("ۀ","ه")
143.                          text=text.replace("ه","ه")
144.                          text=text.replace("ۀ","ه")
145.                          text=text.replace("ۀ","ه")
146.                          text=text.replace("ﻫ","ه")
147.                          text=text.replace("ﻩ","ه")
```

```
148.                    text=text.replace("ت","ت")
149.                    text=text.replace("ټ","ت")
150.                    text=text.replace("ټ","ت")
151.                    text=text.replace("ټ","ت")
152.                    text=text.replace("ث","ث")
153.                    text=text.replace("چ","چ")
154.                    text=text.replace("چ","چ")
155.                    text=text.replace("ح","ح")
156.                    text=text.replace("ځ","ح")
157.                    text=text.replace("خ","خ")
158.                    text=text.replace("د","د")
159.                    text=text.replace("ډ","د")
160.                    text=text.replace("ډ","د")
161.                    text=text.replace("ډ","د")
162.                    text=text.replace("ډ","د")
163.                    text=text.replace("ذ","ذ")
164.                    text=text.replace("ر","ر")
165.                    text=text.replace("ړ","ی")
166.                    text=text.replace("م","م")
167.                    text=text.replace("ح","ح")
168.                    text=text.replace("ۀ","م")
169.                    text=text.replace("ۍ","د")
170.                    text=text.replace("ض","ض")
171.                    text=text.replace("ع","ع")
172.                    text=text.replace("ۍ","د")
173.                    text=text.replace("ر","ر")
174.                    text=text.replace("س","س")
175.                    text=text.replace("و","و")
176.                    text=text.replace("ل","ل")
177.                    text=text.replace("ک","ک")
178.                    text=text.replace("ن","ن")
179.                    text=text.replace("ح","ح")
180.                    text=text.replace("غ","غ")
181.                    text=text.replace("ف","ف")
182.                    text=text.replace("ع","ع")
183.                    text=text.replace("ڤ","ف")
184.                    text=text.replace("ت","ت")
185.                    text=text.replace("ع","ع")
186.                    text=text.replace("ې","ی")
187.                    text=text.replace("ج","ج")
188.                    text=text.replace("۱","1")
189.                    text=text.replace("ء","")
190.                    text=text.replace("۲","2")
191.                    text=text.replace("۳","3")
```

```
                text=text.replace("۴","4")
                text=text.replace("۵","5")
                text=text.replace("۸","8")
                text=text.replace("۷","7")
                text=text.replace("۶","6")
                text=text.replace("ی","ی")
                text=text.replace("و","و")
                text=text.replace("\ufeff","")
                text=text.replace("ی","ی")
                text=text.replace("5","5")
                text=text.replace("","-")
                text=text.replace(" ،","،")
                text=text.replace(" .",".")
                text=text.replace("","")
                text=text.replace("  "," ")
                text=text.replace("ۀ","ت")
                text=text.replace("ـ","")
                text=text.replace("ہ","ه")
                text=text.replace("ۀ","ت")
                text=text.replace("گ","گ")
                text=text.replace("ﻟ","ل")
                text=text.replace("ﺷ","ش")
                text=text.replace("ﻫ","ه")
                text=text.replace("ﻧ","ن")
                text=text.replace("خ","خ")
                text=text.replace("ط","ط")
                text=text.replace("م","م")
                text=text.replace("ی","ی")
                text=text.replace("ﺳ","س")
                text=text.replace("ـ","-")
                text=text.replace(" های "," های ")
                text=text.replace(" ها "," ها ")
                text=text.replace(" ی "," ی ")
                text=text.replace(" ای "," ای ")
                text=text.replace("–","-")
                text=text.replace("ﻻ","لا")
                text=text.replace(" -","-")
                text=text.replace("- ","-")
                text=text.replace("با مقدمه","بامقدمه")
                text=text.replace("به کوشش","به‌کوشش")
                text=text.replace("پژوهش نامه","پژوهش‌نامه")
                text=text.replace("فصل نامه","فصل‌نامه")
                text=text.replace("بین المللی","بین‌المللی")
                text=text.replace("پایان نامه","پایان‌نامه")
                text=text.replace("-","")
                text=sep_text_from_number(text)
                return text
        def all_extract():
```

```
240.             finall_f_l=[]
241.             with codecs.open("E:/Names/jn.txt", 'r',encoding='utf-
    8') as file:
242.                  rtext=file.read()
243.             rtext=Virayesh(rtext)
244.             jnames_scores=rtext.split("\r\n")
245.             with codecs.open("E:/Names/AuthNames.txt", 'r',encoding='utf-
    8') as file:
246.                  rtext=file.read()
247.             rtext=Virayesh(rtext)
248.             names=rtext.split("\r\n")
249.             with codecs.open("E:/Names/TitWords.txt", 'r',encoding='utf-
    8') as file:
250.                  rtext=file.read()
251.             rtext=Virayesh(rtext)
252.             tw_scores=rtext.split("\r\n")
253.             with codecs.open("E:/Names/cn.txt", 'r',encoding='utf-
    8') as file:
254.                  rtext=file.read()
255.             rtext=Virayesh(rtext)
256.             cnames_scores=rtext.split("\r\n")
257.             with codecs.open("E:/t1.txt", 'r',encoding='utf-8') as file:
258.                  ttttext=file.read()
259.             texts=ttttext.split("\r\n")
260.             with codecs.open("E:/t2.txt", 'r',encoding='utf-8') as file:
261.                  bbbtext=file.read()
262.             bibs=bbbtext.split("@")
263.             for i in list(range(900)):
264.                  fvectors=[]
265.                  fvectors=Extract_features(texts[i],bibs[i+1],jnames_scor
    es,names,tw_scores,cnames_scores)
266.                  for fv in fvectors:
267.                       finall_f_l.append(fv)
268.             tempstr=""
269.             for i in list(range(1,len(finall_f_l)-1)):
270.                  if finall_f_l[i-1][len(finall_f_l[0])-
    3]==finall_f_l[i+1][len(finall_f_l[0])-
    3] and finall_f_l[i][len(finall_f_l[0])-
    3]!=finall_f_l[i+1][len(finall_f_l[0])-
    3] and finall_f_l[i+1][len(finall_f_l[0])-
    3] in ["{:<10}".format("journal"),"{:<10}".format("title"),"{:<10}".format(
    "booktitle"),"{:<10}".format("publisher"),"{:<10}".format("note")]:
271.                       finall_f_l[i][len(finall_f_l[0])-3]=finall_f_l[i-
    1][len(finall_f_l[0])-3]
272.             for l in finall_f_l:
273.                  for e in l:
274.                       tempstr=tempstr+str(e)+" "
275.                  tempstr=tempstr+"\r\n"
276.             with codecs.open("E:/FFF.txt", 'w',encoding='utf-
    8') as file:
277.                  file.write(tempstr)
278.        def is_number(s):
279.            try:
280.                int(s)
281.                return True
282.            except ValueError:
283.                return False
284.        def special_year_editting(ref):
285.              ref=ref.replace("  "," ")
286.              numbers=[0,1,2,3,4,5,6,7,8,9]
287.              yearindicators=["ش","ق","م"]
288.              for n in numbers:
289.                   for c in yearindicators:
```

```
                        ref=ref.replace(str(n)+" "+c,str(n)+c)

            return ref
      def is_contain_nonalphabetic(token, Alphabets):
            for c in token:
                  if c not in Alphabets:
                        return True
            return False
      numbers=["0","1","2","3","4","5","6","7","8","9"]
      def is_contain_num_alph(token,Alphabets,numbers):
            flag1=False
            flag2=False
            for c in token:
                  if c in Alphabets:
                        flag1=True
            for c in token:
                  if c in numbers:
                        flag2=True
            return flag1 and flag2
      def is_only_contain_num_alph(token,Alphabets,numbers):
            for c in token:
                  if not c in Alphabets+numbers:
                        return False
            return True
      def is_a_written_number(token):
            yekan=["یک","دو","سه","چهار","پنج","شش","هفت","هشت","نه"]
            specials=["ده","یازده","دوازده","سیزده","چهارده","پانزده","شانزده",
                        "هفده","هجده","نوزده"]
            dahgan=["بیست","سی","چهل","پنجاه","شصت","هفتاد","هشتاد","نود"]
            if token in yekan+specials+dahgan:
                  return True
            else:
                  return False
      engalph=["a","b","c","d","e","f","g","h","i","j","k","l","m","n","o"
,
                  "p","q","r","s","t","u","v","w","x","y","z","A","B","
C","D",
                  "E","F","G","H","I","J","K","L","M","N","O","P","Q","
R","S",
                  "T","U","V","W","X","Y","Z"]
      def is_containing_Eng_alphabet(token,engalph):
            for c in token:
                  if c in engalph:
                        return True
            return False
      def is_year_number(token):
              yearindicators=["ش","ق","م"]
              if is_number(token):
                      if 1000<int(token)<2020:
                              return True
                      else:
                              return False
              elif is_number(token[:-1]) and token[-
1:] in yearindicators:
                      if 1000<int(token[:-1])<2020:
                              return True
                      else:
                              return False
              else:
                      return False
      def is_f_name_indicator(token,Fnameindicators):
```

```
            if token in Fnameindicators:
                return True
            else:
                return False
        def is_ended_with_seperator(token,seperators):
                if isendedwith(token,seperators):
                        return True
                else:
                        return False
        def contain_l_name_indicator(token,lnameindicators):
              if token in lnameindicators:
                    return True
              else:
                    return False
        def hasNumbers(inputString):
              return any(char.isdigit() for char in inputString)
        def token_Label(token,bibitem):
              lines=bibitem.split("\r\n")
              for l in lines:
                    parts=l.split("=")
                    if (len(parts)>1):
                          if token in parts[1]:
                                return parts[0].replace(" ","")
              return "BiKhiaL"
        def is_author_name(token,names):
          if token in names:
            return True
          return False
        def title_name_score(token,tw_scores):
          n1=0
          n2=0
          n3=0
          n4=0
          if "" in token:
            partss=token.split("")
          for w in tw_scores:
            ws=w.split(": ")
            if len(ws)==2 and ws[0]==token:
              n3=int(ws[1])
            if "" in token:
              if len(ws)==2 and ws[0]==partss[0]:
                n1=int(ws[1])
              if len(ws)==2 and ws[0]==partss[1]:
                n2=int(ws[1])
              n4=min(n1,n2)
          return math.ceil(math.log(max(n4,n3)+1))
        def dash_editor(text):
          numbs=[0,1,2,3,4,5,6,7,8,9]
          for n in numbs:
            text=text.replace(str(n)+" -",str(n)+"-")
            text=text.replace(str(n)+"- ",str(n)+"-")
          return text
        def replace_sep_with_space(tmp,seperators):
              for s in seperators:
                    tmp=tmp.replace(str(s), " ")
              return tmp
        def j_name_score(token,jnames_scores):
              for js in jnames_scores:
                    jss=js.split(": ")
                    if jss[0]==token and len(jss)>1:
                          return math.ceil(math.log(int(jss[1])))
              return 0
        def c_name_score(token,cnames_scores):
```

```
409.            for cs in cnames_scores:
410.                  css=cs.split(": ")
411.                  if css[0]==token and len(css)>1:
412.                        return math.ceil(math.log(int(css[1])))
413.            return 0
414.      def is_j_ind(token):
415.            inds=["فصلنامه","پژوهشنامه","مجله","دوفصلنامه","نشریه","دوماهنامه","ماهنامه","
    مطالعات","تازه‌های","تحقیقات","پژوهش","ماه‌نامه","دوفصل‌نامه","فصل‌نامه","پژوهش‌نامه"]
416.            tmp=token
417.            if token[-2:]=="ی":
418.                tmp=tmp[:-2]
419.            for ind in inds:
420.                if (tmp in inds) or (tmp.startswith(ind)):
421.                  return True
422.            else:
423.                  return False
424.      def is_c_ind(token):
425.            inds=["اولین","انجمن","سمینار","کنگره","بین‌المللی","ملی","ایران","کنفرانس","همایش",
    "دهمین","نهمین","هشتمین","هفتمین","ششمین","پنجمین","چهارمین","سومین","دومین","یکمین","نخستین",
    "یازدهمین","دوازدهمین","سیزدهمین","چهاردهمین","پانزدهمین","شانزدهمین",
426.                        "هفدهمین","هجدهمین","نوزدهمین","خلاصه","مقالات"]
427.            tmp=""
428.            token2=token
429.            if token2[-2:]=="ی":
430.                token2=token2[:-2]
431.            if "" in token2:
432.                  parts=token2.split("")
433.                  t1=parts[0]
434.                  t2=parts[1]
435.                  if len(t1)<len(t2):
436.                        tmp=t2
437.                  else:
438.                        tmp=t1
439.            else:
440.                  tmp=token2
441.            if tmp in inds:
442.                  return True
443.            else:
444.                  return False
445.      def contains_volume_ind(token):
446.            tmp=token
447.            indicators=["دوره","دوره‌ی","سال","س","د"]
448.            while is_number(tmp[-1:]):
449.                  tmp=tmp[:-1]
450.            if tmp in indicators:
451.                  return True
452.            else:
453.                  return False
454.      def contains_num_ind(token):
455.            tmp=token
456.            indicators=["شماره","شماره‌ی","شماره‌های","ش"]
457.            while is_number(tmp[-1:]):
458.                  tmp=tmp[:-1]
459.            if tmp in indicators:
460.                  return True
461.            else:
462.                  return False
463.      def contains_series_ind(token):
464.            tmp=token
```

```
            indicators=["چاپ","چ","چ."]
            while is_number(tmp[-1:]):
                  tmp=tmp[:-1]
            if tmp in indicators:
                  return True
            else:
                  return False
      def contains_parts_ind(token):
            tmp=token
            indicators=["جلد","ج","ج."]
            while is_number(tmp[-1:]):
                  tmp=tmp[:-1]
            if tmp in indicators:
                  return True
            else:
                  return False

      def contains_page_ind(token):
            indicators=["ص","صص","صفحات","صفحه","صفحه‌های"]
            tmp=token
            while is_number(tmp[-1:]) or tmp[-1:]=="-":
                  tmp=tmp[:-1]
            if tmp in indicators:
                  return True
            else:
                  return False
      def is_et_al(token):
            etalforms=["همکاران","دیگران"]
            if token in etalforms:
                  return True
            else:
                  return False
      def is_mon_seas_name(token):
            months1=["فروردین","اردیبهشت","خرداد","تیر","مرداد","شهریور",
                        "مهر","آبان","آذر","دی","بهمن","اسفند"]
            months2=["فروردین‌ماه","اردیبهشت‌ماه","خردادماه","تیرماه","مردادماه","شهریورماه",
                        "مهرماه","آبان‌ماه","آذرماه","دی‌ماه","بهمن‌ماه","اسفندماه"]
            seasons=["بهار","تابستان","پاییز","زمستان"]
            months3=[]
            for i in list(range(len(months1))):
                for j in list(range(i+1, len(months1))):
                    months3.append(months1[i]+""+months1[j])
            months=months1+months2+months3
            if token in months+seasons:
                  return True
            else:
                  return False
      def is_it_part_indicator(token):
            tmp=token
            inds=["جلد","ج","ج."]
            while is_number(tmp[-1:]):
                  tmp=tmp[:-1]
            if tmp in inds:
                  return True
            else:
                  return False
      def is_it_series_indicator(token):
            tmp=token
```

```
523.              inds=["چاپ","چ","چ."]
524.              while is_number(tmp[-1:]):
525.                    tmp=tmp[:-1]
526.              if tmp in inds:
527.                    return True
528.              else:
529.                    return False
530.        def is_it_vol_indicator(token):
531.              tmp=token
532.              inds=["دوره","دوره‌ی","سال","س","د",".د",".س"]
533.              while is_number(tmp[-1:]):
534.                    tmp=tmp[:-1]
535.              if tmp in inds:
536.                    return True
537.              else:
538.                    return False
539.        def is_it_num_indicator(token):
540.              tmp=token
541.              inds=["شماره","شماره‌ی","شماره‌های","ش"]
542.              while is_number(tmp[-1:]):
543.                    tmp=tmp[:-1]
544.              if tmp in inds:
545.                    return True
546.              else:
547.                    return False
548.        def isendedwith(temp, somelist):
549.              for item in somelist:
550.                    if temp.endswith(item):
551.                          return True
552.              return False
553.        Alphabets=["ا",  "آ", "ب", "پ", "ت", "ث", "ج", "چ", "ح",
554.                         "خ", "د", "ذ", "ر", "ز", "ژ", "س", "ش", "ص",
555.                         "ض", "ط", "ظ", "ع", "غ", "ف", "ق", "ک", "گ",
556.                         "ل", "م", "ن", "و", "ه", "ی",""]
557.        def is_school_name(token,names):
558.          if token in names:
559.            return True
560.          return False
561.        def sep_text_from_number(token):
562.              tmp=token
563.              inds=["ش","ش.","شماره","شماره‌ی","شماره‌های","د",".د","س",".س","سال",
    "دوره‌ی","جلد","چاپ","ج",".ج","چ",".چ","دوره",".ص","ص","صص",".صص","صفحات","
    آبان","مهر","شهریور","مرداد","تیر","خرداد","اردیبهشت","فروردین","صفحه های","صفحه‌های","صفحه
    ابان","آذر","دی","بهمن","اسفند","بهار","تابستان","پاییز","زمستان"]
564.              numbers=["0","1","2","3","4","5","6","7","8","9"]
565.              for ind in inds:
566.                    for n in numbers:
567.                          tmp=tmp.replace(ind+n,ind+" "+n)
568.              return tmp
569.        def convertnumbs(ref):
570.              newref=""
571.              for c in ref:
572.                    if c=="۱":
573.                          newref=newref+unidecode(str(1))
574.                    elif c=="۲":
575.                          newref=newref+unidecode(str(2))
576.                    elif c=="۳":
```

```
                        newref=newref+unidecode(str(3))
                elif c=="۴":
                        newref=newref+unidecode(str(4))
                elif c=="۵":
                        newref=newref+unidecode(str(5))
                elif c=="۶":
                        newref=newref+unidecode(str(6))
                elif c=="۷":
                        newref=newref+unidecode(str(7))
                elif c=="۸":
                        newref=newref+unidecode(str(8))
                elif c=="۹":
                        newref=newref+unidecode(str(9))
                elif c=="۰":
                        newref=newref+unidecode(str(0))
                else:
                        newref=newref+c
          return newref
    seperators=["،",".",":","؛","(",")","\"","»","«","؟","?"]
    lnameindicators=["پور","فرد","نیا","زاده","نژاد","راد","فر","ان","کیا","پناه","منش"
]
    all_extract()
```

کد مربوط به آموزش و آزمایش دسته‌بند SVM با استفاده از بردارهای ویژگی استخراج شده در مرحله قبل

```
import logging
import os
import codecs
from random import randint
import pandas as pd
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
import math
from xlrd import *
from openpyxl import load_workbook
svclassifier = SVC(C=4,kernel='linear')
def add_next_token_features(table):
    newtable=[]
    for i in list(range(len(table)-1)):
        temp=[]
        for j in list(range(len(table[0]))):
            temp.append(table[i][j])
        for j in list(range(3, len(table[0])-4)):
            if table[i+1][0]!=1:
                temp.append(table[i+1][j])
            else:
                temp.append("0")
        newtable.append(temp)
    temp=[]
    for j in list(range(len(table[0]))):
        temp.append(table[len(table)-1][j])
    for j in list(range(3,len(table[0])-4)):
```

```
            temp.append("0")
      newtable.append(temp)
      return newtable
def strfeatures_to_intfeatures(table):
      Labels=["0","author","title","year","journal","volume","number","page
s","publisher","address","BiKhiaL","translator","tyear","note","school","bo
oktitle","series","month"]
      for i in list(range(len(table))):
            for j in list(range(len(table[i]))):
                  if table[i][j] in Labels:
                        table[i][j]=Labels.index(table[i][j])
      return table
def add_previous_labels(table):
      finallist=[]
      temparray=[]
      templist=[]
      for j in list(range(3)):
            templist.append("0")
      for i in list(range(len(table))):
            temparray=table[i]
            if table[i][0]==1:
                  for j in list(range(3)):
                        templist[j]="0"
            for k in list(range(3)):
                  temparray[len(temparray)-1-(3-k)]=templist[k]
            templist=shift_left(templist)
            templist[2]=temparray[len(temparray)-1]
            finallist.append(temparray)
      return finallist
def build_test_data(table,ind):
      newtable=[]
      for i in list(range(len(table))):
            if table[i][1]==ind:
                  newtable.append(table[i])
      return newtable
def build_train_data(table,ind):
      newtable=[]
      for i in list(range(len(table))):
            if table[i][1]!=ind:
                  newtable.append(table[i])
      return newtable
def shift_left(somelist):
      for i in list(range(len(somelist)-1)):
            somelist[i]=somelist[i+1]
      somelist[len(somelist)-1]=0
      return somelist
def xlsx_to_list(wb):
      tempp=[]
      sheet=wb.active
      max_row=sheet.max_row
      max_column=sheet.max_column
      for i in range(1,max_row+1):
            temparr=[]
            for j in range(1,max_column+1):
                  temparr.append(sheet.cell(row=i,column=j).value)
            tempp.append(temparr)
      return tempp
def test(test_list):
      predicted=[]
      test_item=[]
      temp=[]
      for i in list(range(3)):
            temp.append(0)
```

```
89.       for i in list(range(len(test_list))):
90.             if test_list[i][0]==1:
91.                   for j in list(range(3)):
92.                         temp[j]=0
93.             for j in list(range(3)):
94.                   test_list[i][36-(3-j)]=temp[j]
95.             temp=shift_left(temp)
96.             test_item=[]
97.             test_item.append(test_list[i])
98.             df2=pd.DataFrame(test_item)
99.             x2=df2.drop(df2.columns[[1,36]],axis=1)
100.                  y2=df2[df2.columns[36]]
101.                  y_pred = svclassifier.predict(x2)
102.                  temp[2]=y_pred[0]
103.                  predicted.append(y_pred[0])
104.           return predicted
105.      def main_test():
106.            strr=""
107.            l_pred=[]
108.            l2=[]
109.            wb=load_workbook('E:/Features3.xlsx')
110.            mylist=xlsx_to_list(wb)
111.            myflist=add_previous_labels(mylist)
112.            mytflist=add_next_token_features(myflist)
113.            myfflist=strfeatures_to_intfeatures(mytflist)
114.            myffflist=[]
115.            for i in list(range(10)):
116.                  train_list=build_train_data(myfflist,i)
117.                  test_list=build_test_data(myfflist,i)
118.                  myffflist=myffflist+test_list
119.                  df1=pd.DataFrame(train_list)
120.                  x1=df1.drop(df1.columns[[1,36]],axis=1)
121.                  y1=df1[df1.columns[36]]
122.                  df2=pd.DataFrame(test_list)
123.                  y2=df2[df2.columns[36]]
124.                  for item in y2:
125.                        l2.append(item)
126.                  svclassifier.fit(x1, y1)
127.                  y_pred=test(test_list)
128.                  for item in y_pred:
129.                        l_pred.append(item)
130.            c=0
131.            l3=[]
132.            wb=load_workbook('E:/Book12.xlsx')
133.            slist=xlsx_to_list(wb)
134.            l3=[]
135.            flag=False
136.            flag2=False
137.            l4=[]
138.            for j in list(range(10)):
139.                  for i in list(range(len(slist))):
140.                        if slist[i][0]==j:
141.                              l3.append(slist[i][1])
142.                              l4.append(slist[i][2])
143.            for j in list(range(1,len(l_pred)-1)):
144.                  if l4[j]==1:
145.                        flag=False
146.                        flag2=False
147.                  if l3[j] in ["پایان‌نامه","رساله","رساله‌ی"]:
148.                        flag=True
149.                  if l3[j] in ["چاپ","چ","چ."]:
150.                        flag2=True
```

```
                    if flag==True and l_pred[j] in [8,15]:
                        l_pred[j]=14
                    if flag2==True and l_pred[j] in [5]:
                        l_pred[j]=16
                    if l_pred[j-1]==4 and l_pred[j]==2:
                        l_pred[j]=4
                    if l_pred[j]==14 and not l3[j] in ["دانشگاه","پژوهشگاه","دانشکده","پژوهشکده","گروه","موسسه"]:
                        l_pred[j]=l_pred[j-1]
                    if l_pred[j-1]==15 and l_pred[j]==2:
                        l_pred[j]=15
                    if l_pred[j-1]==13 and l_pred[j+1]==13 and l_pred[j]==8:
                        l_pred[j]=13
                    if l_pred[j-1]==14 and l_pred[j+1]==14 and l_pred[j]==8:
                        l_pred[j]=14
                    if l_pred[j-1]==14 and l_pred[j+1]==14 and l_pred[j]==15:
                        l_pred[j]=14
                    if l_pred[j-1]==15 and l_pred[j+1]==15 and l_pred[j]==14:
                        l_pred[j]=15
                    if l_pred[j-1]==8 and l_pred[j+1]==8 and l_pred[j]==13:
                        l_pred[j]=8
                    if l_pred[j-1]==14 and l_pred[j+1]==14 and l_pred[j]==13:
                        l_pred[j]=14
                    if l_pred[j-1]==13 and l_pred[j+1]==13 and l_pred[j]==10:
                        l_pred[j]=13
                    if l_pred[j-1]==5 and l_pred[j+1]==5 and l_pred[j]==4:
                        l_pred[j]=5
                    if l_pred[j-1]==5 and l_pred[j+1]==5 and l_pred[j]==1:
                        l_pred[j]=5
                    if l_pred[j-1]==5 and l_pred[j+1]==5 and l_pred[j]==2:
                        l_pred[j]=5
                    if l_pred[j-1]==14 and l_pred[j]==15:
                        l_pred[j]=14
                    if l_pred[j-1]==15 and l_pred[j]==14:
                        l_pred[j]=15
            print(confusion_matrix(l2,l_pred))
            print("----------------------")
            print(classification_report(l2,l_pred))
            for i in list(range(len(l_pred))):
                strr=strr+"{:<12}".format(Labels[l2[i]])+"      "+"{:<12}".format(Labels[l_pred[i]])+"    "+str(l3[i])+"\r\n"
            with codecs.open("E:/predicted.txt", 'w',encoding='utf-8') as file:
                file.write(strr)
    Labels=["0","author","title","year","journal","volume","number","pages","publisher","address","BiKhiaL","translator","tyear","note","school","booktitle","series","month"]
    main_test()
```

Abstract

Bibliographic references, provided usually at the end of scientific reports, are composed of a set of fields including authors, title, journal, year, school, etc. While parsing a bibliographic reference to its constructive fields can easily be done by human users, it is difficult to make this task autonomous due to the variations in formats. There exist many solutions in the literature for this problem however, the existing ones are highly language-sensitive in such a way that results of parcing a bibliographic reference in some language by an existing method for another language would be catastrophe. To the best of our knowledge, there doesn't exist any solution to this problem in Persian language. In this research project, based on the importance of this problem in document retrieval and autonomous citation indexing, we first provide a comprehensive literature review. Then, we propose an intelligent method for citation parsing in Persian language by using the support vector machine (SVM) classification method. We have also implemented the proposed method using Python and trained and tested it by using a dataset designed to this end in this project. The results show 92% in average for precision, recall and F1 measures.

Keywords: Citation parsing, classification, multi-class classification, support vector machine.

Iranian Research Institute for Information Science and Technology (Irandoc)

Research Project

Designing an intelligent citation parsing method for Persian language

By:

Nasrollah Pakniat

Co-authors:

Jalal A. Nasiri

Ammar Jalalimanesh

Feburary 2019