بررسی تأثیرات ریشه‌یابی در بازیابی اطّلاعات در زبان فارسی

نوا احسان و هشام فیلی
دانشکدهٔ مهندسی برق و کامپیوتر، پردیس دانشکده‌های فنی، دانشگاه تهران

چکیده:

یکی از مهم‌ترین موضوعات در پردازش زبان طبیعی و بازیابی اطّلاعات، یافتن ریشهٔ کلمات است. ریشهٔ کلمه، جزئی از کلمه است که پس از حذف وندهای کلمه (پیشوند، پسوند و میانوند) باقی می‌ماند. یکی از روش‌های افزایش کارایی سیستم‌های بازیابی اطّلاعات، استفاده از ریشه‌یابی کلمات است. زیرا اشتقاقات مختلف یک کلمه به ریشهٔ آن کلمه تبدیل می‌شوند. درنتیجه جستجو بر اساس ریشهٔ کلمه انجام خواهد شد و اندازهٔ ساختار ایندکس کاهش می‌یابد. در این مقاله الگوریتمی برای به‌دست آوردن ریشهٔ کلمات در زبان فارسی ارائه شده است و سپس نتیجهٔ آن در بازیابی اطّلاعات با الگوریتم‌های متفاوت رتبه‌بندی، مورد ارزیابی قرار گرفته است. الگوریتم ارائه شده با استفاده از قواعد ساخت‌واژی زبان فارسی و استفاده از مجموعه لغات برای جلوگیری از ایجاد ریشه‌های نادرست، به ریشه‌یابی کلمات می‌پردازد. تعداد قواعد استفاده شده ۴۳ قانون است. با استفاده از الگوریتم ارائه شده، اندازهٔ ساختار ایندکس پنج درصد کاهش یافته است و همچنین میزان میانگین متوسّط دقّت (mean average precision) در سیستم بازیابی اطّلاعات حدود پنج درصد افزایش یافته است.

واژگان کلیدی: ریشه‌یابی، پردازش زبان طبیعی و بازیابی اطّلاعات

۱- مقدّمه

واژهٔ ریشه‌یابی به معنای حذف پسوندها، پیشوندها و میانوندهای کلمه و به‌دست آوردن ریشهٔ کلمه است و یکی از اهداف آن در بازیابی اطّلاعات، جستجوی کلمه بر اساس ریشهٔ آن می‌باشد. در این صورت به‌طور مثال هر دو کلمهٔ «*دانشجو*» و «*دانشجویان*» در سیستم بازیابی اطّلاعات به‌صورت «*دانشجو*» ذخیره می‌شوند و اگر کاربری در درخواست[1] خود از کلمهٔ «دانشجو» استفاده کند، سندهایی که در آن واژهٔ «دانشجویان» وجود دارد نیز می‌توانند به‌عنوان پاسخ به کاربر نشان داده شوند. همچنین اندازهٔ ساختاری که برای ذخیره ایندکس در نظر گرفته می‌شود، کاهش خواهد یافت.

الگوریتم پیشنهاد شده برای به‌دست آوردن ریشهٔ کلمات در زبان فارسی همانند ریشه‌یاب پورتر[2] در زبان انگلیسی (*Porter, 1980*) بر اساس قواعد ساختواژی[3] کلمات است. همچنین برای جلوگیری از ایجاد ریشه‌های ناصحیح از پیکره‌ای از کلمات فارسی نیز استفاده شده است. اساس روش پورتر حذف پسوندهای کلمات است. با این فرض که لغت‌نامه‌ای وجود ندارد و برای هر پسوند، شرایطی وجود دارد که نشان می‌دهد حذف آن از یک کلمه در چه شرایطی ریشهٔ معتبر ایجاد می‌کند (*Porter, 1980*). این روش در زبان فارسی استفاده شده است و پسوندهای موجود در زبان فارسی که طبق شرایطی می‌توانند ریشهٔ معتبر ایجاد کنند، از کلمه حذف می‌شوند.

الگوریتم‌های رتبه‌بندی از بخش‌های اصلی سیستم‌های بازیابی اطّلاعات به‌شمار می‌آیند. بنابراین روش پیشنهادشده برای ریشه‌یابی با سه روش متفاوت رتبه‌بندی، مورد ارزیابی قرار گرفته است.

[1] query
[2] Porter
[3] Morphological rules

۲- کارهای مشابه

از رایج‌ترین الگوریتم‌های ریشه‌یابی موجود در زبان انگلیسی الگوریتم پورتر است. در این روش با حذف پسوندهای کلمات، از تعداد واژه‌های منحصربه‌فرد در بازیابی اطّلاعات کاسته می‌شود. درنتیجه موجب بالا رفتن کارایی سیستم خواهد شد. جزئیات روش در (*Porter, 1980*) شرح داده شده است. از روش‌های دیگری که در زبان انگلیسی به‌کار می‌رود الگوریتم Lovins است (*Lovins, 1968*) که این الگوریتم شامل ۲۵۰ پسوند است که طولانی‌ترین پسوند متصل به کلمه را حذف می‌کند؛ با این شرط که کلمهٔ باقی‌مانده حداکثر سه نویسه داشته باشد. الگوریتم Bacchin در سال ۲۰۰۲ توسط Bacchin ارائه شد(*Bacchin et al., 2005*). این الگوریتم یک روش غیرساختاری است و بر اساس تحلیل آماری بر روی کلمات یک زبان کار می‌کند. این روش مستقل از زبان است و در (*MohammadiNasiri et al., 2006*) پیاده‌سازی و تحلیل آن در زبان فارسی مورد بررسی قرار گرفته است. از آنجا که ریشه‌یابی در کارایی بازیابی اطّلاعات اهمیت فراوان دارد، در زبان فارسی نیز الگوریتم‌هایی برای ریشه‌یابی ارائه شده است. یکی از این روش‌ها با استفاده از ماشین حالت متناهی[1] پیاده‌سازی‌شده است. این روش در سال ۲۰۰۵ ارائه شد. شرح کامل این روش در (*Taghva et al., 2005*) آمده است.

شریف‌لو و *شمس‌فرد* (*Sharifloo and Shamsfard, 2008*) در مقابل روش حذف پسوندها، روش پایین به بالا را به‌کار گرفته‌اند. در این روش ابتدا ریشه‌های ممکن از کلمه استخراج و سپس بررسی می‌شود کدام ریشه با قواعد موجود همخوانی دارد. بستهٔ نرم‌افزاری Step-1 نیز شامل ریشه‌یاب در زبان فارسی می‌باشد (*Shamsfard et al., 2010*) که این ریشه‌یاب قادر به ریشه‌یابی کلمات تصریفی، تعدادی از کلمات اشتقاقی و تحلیل ساخت‌واژی آن‌ها است. روش دیگری بر اساس حذف حروف در (*Tashakori et al., 2003*) ارائه شده است. به این ترتیب که حروف را تا حدّ امکان از انتهای کلمه حذف می‌کند. این روش به‌دلیل وجود استثناهای فراوان در زبان فارسی، می‌تواند منجر به ایجاد ریشه‌های نادرست شود. روش دیگری بر اساس حذف پسوندها در (*Dolamic et al., 2009*) ارائه شده است که به بهبود ۲/۱ درصد در بازیابی اطلاعات با استفاده از مدل DFR_PL2 (*Dolamic et al., 2009*) انجامیده است. روشی مبتنی بر قاعده نیز در (*Rahimtoroghi et al., 2010*) ارائه شده است که نحوهٔ عملکرد این ریشه‌یاب در بازیابی اطّلاعات با روش BM25 مورد ارزیابی قرار گرفته است. آماری از نحوهٔ عملکرد ریشه‌یاب به‌صورت مستقل در دست نمی‌باشد.

روش ارائه شده در این مقاله یک روش مبتنی بر قاعده و بر پایه حذف پسوندها و پیشوندهای کلمات است که برای جلوگیری از ایجاد پسوندهای نادرست از پیکره‌ای از مجموعه لغات زبان نیز استفاده شده است. در این مقاله علاوه‌بر ارزیابی ریشه‌یاب در یافتن ریشه‌های صحیح، عملکرد آن در بازیابی اطلاعات با سه مدل متفاوت رتبه‌بندی بررسی شده است.

در ابتدا به شرح الگوریتم ارائه شده در فارسی پرداخته می‌شود و سپس کارایی آن در سه مدل متفاوت بازیابی اطلاعات مورد ارزیابی قرار خواهد گرفت.

۳- الگوریتم ریشه‌یابی در زبان فارسی

روش ارائه شده تا حدّی شبیه به الگوریتم پورتر در زبان انگلیسی است. روش پورتر بر پایهٔ این فرض است که پسوندها در زبان انگلیسی از ترکیب پسوندهای کوچک‌تر تشکیل شده‌اند که چنین فرضی برای فارسی می‌تواند درست باشد. روند جداسازی پسوندها در **پنج** مرحله صورت می‌گیرد که هر کدام قواعد ریشه‌یابی مخصوص به خود دارند. قواعد جایگزینی نیز بر اساس محدودیت‌های موجود در قواعد اعمال می‌شود؛ به‌طور مثال یکی از این محدودیت‌ها می‌تواند طول ریشهٔ ایجاد شده باشد. این طول برابر است با تعداد واج‌های مصوّت[2] و صامت[3] که پشت سر هم قرار می‌گیرند. همچنین از محدودیت‌های دیگر می‌تواند چنین باشد که یک ریشه به واج مصوّت یا به واج صامت ختم شود. هنگامی که همهٔ شرایط یک قاعده ارضا شوند، آن قاعده بر روی کلمه اعمال می‌شود و نتیجهٔ آن حذف پسوند مورد نظر است و الگوریتم به مرحلهٔ بعد می‌رود. اگر شرایط یک قاعده ارضا نشود قانون بعدی در همان مرحله بررسی می‌شود، تا زمانی که شرایط یکی از قوانین ارضا شوند یا قوانین یک مرحله به اتمام برسند. این کار در هر **پنج** مرحله از الگوریتم انجام می‌شود. ترتیب قواعد بر اساس حذف طولانی‌ترین پسوند است.

[1] Deterministic finite state automata

[2] Vowel
[3] Consonant

۳-۱ - قوانین ریشه‌یابی در اسم‌ها

در زبان فارسی قواعد مشخّصی برای جمع، مالکیت و مقایسه اسامی وجود دارد. اسامی ممکن است در ترکیب اضافه قرار گیرند و یا شامل پسوند نکره یا ضمایر ملکی باشند. این قوانین در (*Megerdoomian, 2004*) به‌تفصیل بیان شده‌اند، که بعضی به شرح زیر می‌باشند.

۱.[اسم + نشانهٔ جمع + ی وصل]گروه اسمی

۲.[اسم + نشانهٔ جمع + نشانهٔ نکره/نشانهٔ مالکیت]گروه اسمی

پسوندهای جمع عبارتند از *«ها، ان، ون، ین و ات»*.

ترتیب قوانین طوری قرار گرفته است که در ابتدا طولانی‌ترین پسوند حذف شود. طولانی‌ترین پسوندی که از قانون (۱) به‌دست می‌آید عبارت است از *«های»* و طولانی‌ترین پسوندهایی که از قانون (۲) به‌دست می‌آیند عبارتند از *«های، هایم، هایت، هایش،...»*. اگر این پسوندها وجود نداشته باشند، الگوریتم به سراغ پسوندهای کوتاه‌تر می‌رود.

از بین پسوندهای ایجاد شده، آنهایی که باعث ایجاد ریشه‌های نامعتبر نمی‌شوند یا می‌توان شرایطی تعیین کرد که از ایجاد ریشه‌های نامعتبر جلوگیری شود از انتهای کلمات حذف می‌شوند. به‌طور مثال *«ان»* در بسیاری کلمات متعلق به خود کلمه است؛ مانند *«فراوان، سخنران»*. بنابراین حذف آن می‌تواند ریشهٔ نامعتبر ایجاد کند. از محدودیت‌های دیگری که روی همهٔ قوانین اعمال می‌شود، این است که طول ریشهٔ ایجاد شده نباید کمتر از سه باشد. بررسی کلمات موجود در پیکرهٔ فارسی (*Bijankhan, 2006*) نشان داده است که ۸۴ درصد از کلماتی که طول کمتر از سه دارند، حروف اضافه و ربط می‌باشند. همچنین محدودیت طول کلمه در ریشه‌یاب (*Taghva et al., 2005*) نیز استفاده شده است. همچنین تا حدّ امکان کلمات استثنا در نظر گرفته شده‌اند. به‌طور مثال پسوندهای «ها و *های و هایی*» از کلمه *«انتها»* برداشته نمی‌شوند. چند نمونه از اعمال این قانون در ادامه آمده است.

(۱) تابلوهایی: تابلو +هایی

(۲) رنگ‌هایی: رنگ +هایی

(۳) فضای: فضا + ی

کلماتی که *به «الف»* یا *«واو»* ختم می‌شوند برای گرفتن پسوند مالکیت باید یک *«ی»* اضافه شود؛ مانند *«دانشجویت، صدایت»*. بنابراین *«یت»* باید از انتهای آنها حذف شود. امّا به دلیل آنکه در بسیاری کلمات، مانند *«حمایت، تقویت، روایت و...» «یت»* پسوند محسوب نمی‌شود پس از حذف این پسوند، کلمهٔ ایجاد شده در مجموعهٔ لغات جستجو می‌شود و در صورت یافت نشدن کلمهٔ ایجادشده، پسوند از انتهای کلمه حذف نخواهد شد. همچنین این بررسی برای پسوند *«یش»* نیز انجام می‌شود تا کلماتی که به *«ایش»* ختم می‌شوند مانند *«آزمایش، گرایش و افزایش و ...»* پسوند نامعتبر ایجاد نکنند. در این مقاله از کلمات موجود در پیکرهٔ فارسی بی‌جن‌خان (*Bijankhan, 2006*) به‌عنوان مجموعه لغات استفاده شده است.

کلماتی که *به «یی»* ختم می‌شوند *با «ان»* جمع بسته می‌شوند که در این موارد کل پسوند *«یان»* حذف می‌شود.

(۳) روستاییان: روستا + یان

(۴) اروپاییان: اروپا + یان

کلمات مختوم به *«ی»* با *«ون»* جمع بسته می‌شوند؛ مانند *«انقلابیون»* اما به‌دلیل ایجاد پسوندهای نامعتبر در کلماتی مانند *«تلویزیون، میلیون و...»* این قانون نیز توسط مجموعه لغات مورد بررسی قرار می‌گیرد.

در اسم‌هایی که به *«الف یا واو»* ختم می‌شوند، علامت نکره به *«یی»* تبدیل می‌شود و به جای *«ی»*، *«یی»* از انتهای آنها حذف می‌شود.

(۵) صدایی: صدا + یی

برای پیونددادن اسم‌هایی که به *«الف یا واو»* ختم می‌شوند از یای وصل استفاده می‌شود. بنابراین اگر کلمه‌ای به *«ای»* یا *«وی»* ختم شده باشد، *«ی»* از انتهای آن حذف و کلمه ایجاد شده، در مجموعهٔ لغات جستجو می‌شود. در این صورت کلماتی مانند *«سماوی، تساوی، معنوی»* نیز ریشهٔ نامعتبر ایجاد نمی‌کنند.

(۶) هوای: هوا + ی

پسوند *«جات»* نیز از پسوندهایی است که از حذف آن خودداری شده است. زیرا قاعدهٔ مشخّصی برای حذف آن نمی‌توان یافت. در کلمه *«کارخانجات»* باید پس از حذف *«جات»* یک *«ه»* به انتهای کلمه اضافه شود. در کلمهٔ *سبزیجات* تنها حذف *«جات»* کفایت می‌کند و در کلمهٔ *«احتیاجات»* تنها *«ات»* باید حذف شود.

در مورد کلماتی که پسوند *«گان»* دارند، با حذف «گان» ممکن است ریشه‌های نامعتبر ایجاد شود مانند «رایگان، بزرگان». بنابراین حذف آن با بررسی در مجموعهٔ لغات انجام می‌گیرد. پس از حذف پسوند *«گان»* حرف *«ه»* به انتهای کلمه اضافه می‌شود، مانند:

(۷) رانندگان: راننده + گان

در زبان فارسی صفات در حالت تفضیلی و عالی به‌ترتیب پسوند «تر» و «ترین» می‌گیرند. این پسوندها به‌جز کلماتی که به «متر» ختم می‌شوند در بقیهٔ موارد حذف می‌شوند. پسوند«گاه» نیز در همهٔ موارد حذف می‌شود.

(۸) بزرگتر: بزرگ + تر

(۹) جالب‌ترین: جالب + ترین

(۱۰) گردش‌گاه: گردش + گاه

۳-۲- ریشه‌یابی افعال

افعال در زبان فارسی به تنهایی یا به کمک وابسته‌هایی بر چهار مفهوم دلالت می‌کنند، مفهوم مثبت و منفی، مفهوم شخص، مفهوم مفرد و جمع بودن و مفهوم زمان (*Givi and Anvari, 2006*). افعال ساده با توجه به زمان فعل، به افعال ماضی و مضارع تقسیم می‌شوند. افعال مضارع از بن مضارع و افعال ماضی از بن ماضی تشکیل شده‌اند. (جدول ۱) ساخت‌های افعال زمان حال و امر را و (جدول ۲) ساخت‌های افعال ماضی و مستقبل را نشان می‌دهد.

(جدول ۱): ساخت‌های زمان حال و امر
(Givi and Anvari, 2006)

اجزایی که پیش از بن می‌آیند	بن مضارع	اجزایی که پس از بن می‌آیند
می	خور	م
می	خور	ی
می	خور	د
ب	خور	ید
ب	خور	ند

(جدول ۲): ساخت‌های زمان گذشته و مستقبل
(Givi and Anvari, 2006)

اجزای پیش از بن	بن ماضی	اجزای پس از بن
	خورد	م
	خورد	ه ام
می	خورد	ه ای
می	خورد	م
	خورد	ه بودیم
	خورد	ه باشید
	خورد	ه بوده ایم
خواهی	خورد	

افعال در زبان فارسی به‌همراه نشانه‌های شخص و زمان می‌باشند. از آنجا که حذف شناسه‌ها می‌تواند منجر به ایجاد ریشه‌های نادرست شود، در این خطایاب از حذف این نشانه‌ها خودداری شده است. در مورد افعال، پیشوندهای «می» و «نمی» که نشانه، استمرار می‌باشند، حذف می‌شوند.

۴- ارزیابی درونی ریشه‌یاب

به‌منظور ارزیابی درونی ریشه‌یاب، ۱۰۰ کلمه به‌صورت تصادفی انتخاب شده‌اند. از کلّ ۱۰۰ واژهٔ مجموعه آزمون، ریشه‌یاب شرح داده شده ۹۲ ریشه را به‌صورت صحیح ایجاد کرده است. ارزیابی مشابه (*Sharifloo and Shamsfard, 2008*) برای زبان فارسی، که در آن ریشه‌یاب مبتنی بر قاعده، شامل ۲۵۲ قانون و ۲۰ قانون جبرانی است، ۹۰/۱٪ دقّت در ریشه‌یابی داشته است.

۵- ارزیابی ریشه‌یاب در بازیابی اطلاعات

از آنجا که در سیستم بازیابی اطّلاعات، روش‌های مختلفی برای رتبه‌بندی سندها وجود دارد، الگوریتم ارائه شده با سه روش متفاوت از سه مدل رتبه‌بندی ارزیابی شده است. روش اوّل استفاده از مدل بُرداری[1] است. در این روش سندها و درخواست‌ها، به‌صورت بُردار تعریف می‌شوند و معیار شباهت دو بُردار نیز توسط فاصلهٔ ۱÷ ض کسینوسی بین دو بُردار تعریف می‌شود. روش Lnu-ltu (*Singhal at al., 1996*) نوعی روش بُرداری است که از ترکیبی از طول سند و میانگین طول اسناد برای ارزیابی شباهت دو بُردار استفاده می‌کند و نتایج گزارش شده آن در مجموعه TREC (*Hiemstra et al., 2000*) بهتر از فاصله کسینوسی است. روش دوم مدل زبانی[2] است که معیار آن شباهت مدل زبانی سند و درخواست[3] می‌باشد. روش Hiemstra (*Hiemstra, 2001*) روشی است که یک مدل رتبه‌بندی با استفاده از مدل زبان ارائه کرده است. روش سوم مدل آماری است که احتمال حضور کلمات درخواست در سند را تخمین می‌زند. روش‌های فراوانی در مدل آماری پیشنهاد شده‌اند. در این مقاله روش BM25 (*Jones et al., 2000*) انتخاب شده است. برای ارزیابی الگوریتم نیاز به مجموعه‌ای از مستندات فارسی است و یک مجموعه از درخواست‌ها که باید توسط افرادی

[1] Vector Model
[2] Language model
[3] Query

ارزیابی شوند و قضاوت‌های[1] افراد را به‌عنوان معیاری برای ارزیابی الگوریتم در نظر می‌گیریم. برای این کار از مجموعه مستندات همشهری[2] استفاده شده است. این مجموعه یک پیکرهٔ استاندارد، شامل حدود ۱۶۵۰۰۰ سند و ۵۲ درخواست و قضاوت برای آن تعریف شده است. همان‌طور که در (شکل ۱) نشان داده شده است، هر درخواست شامل یک موضوع[3] و یک توصیف[4] است.. طول موضوعات کوتاه‌تر و میانگین آن ۲/۸۴ کلمه است. در ادامهٔ مقاله نتایج حاصل از استفادهٔ توصیف، به‌عنوان درخواست سند نشان داده شده است. در این مجموعه هر سند توسط سه برچسب[5] مشخّص شده است، DID (شمارهٔ سند)، DATE (تاریخ) و CAT (موضوع). این پیکره در گروه تحقیقاتی پایگاه داده‌های[6] دانشگاه تهران تهیه شده است. جزییات آماری این پیکره در (*Aleahmad at al., 2009*) شرح داده شده است.

```
<QID>
2
</QID>
<title>
        قوانین انرژی هسته ای
</title>
<description>
قوانین جهانی حاکم بر استفاده از انرژی هسته ای کدامند؟
</description>
```

(شکل ۱): نمونه‌ای از درخواست در مورد قوانین انرژی هسته‌ای

برای ساختن ایندکس و به‌دست آوردن معیارهای دقّت[7] و فراخوانی[8] بر روی این پیکره از ابزار[9] Terrier استفاده شده است. الگوریتم ریشه‌یابی بر روی پیکره و درخواست‌ها اعمال شده است. تعدادی از کلمات ریشه‌یابی شده، با این روش در (جدول ۳) نشان داده شده است.

(جدول ۳): تعدادی از کلمات ریشه‌یابی شده

کلمه	ریشه	پسوند
زندگی	زنده	گی
نمایشگاه	نمایش	گاه
مورچگان	مورچه	گان
تابلوهای	تابلو	های
مزایای	مزایا	ی
میلیونها	میلیون	ها
روبروی	روبرو	ی
وفایی	وفا	یی
گردشگاه	گردش	گاه
زمین‌های	زمین	های
فعالیت‌های	فعالیت	های
آشنایی	آشنا	یی
محل‌های	محل	های
جالب‌ترین	جالب	ترین
بزرگتر	بزرگ	تر
دانشجویان	دانشجو	یان

تعداد کلّ کلمات، بدون کلمات پرتکرار[10] در حالت بدون ریشه‌یابی در این پیکره، برابر با ۴۹،۰۲۰،۳۸۲ کلمه است و تعداد کلمات یکتا[11] برابر با ۴۷۹،۵۵۴ کلمه است. پس از ریشه‌یابی تعداد کلّ کلمات به ۴۶،۵۳۹،۹۹۵ و کلمات یکتا به ۴۵۵،۸۴۹ کلمه کاهش یافت که این خود باعث کم شدن ایندکس‌ها و بالا رفتن سرعت بازیابی می‌شود.

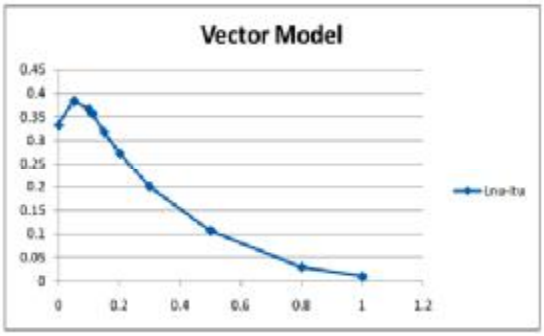
۵-۱- مدل برداری

در مدل بُرداری، سندها به‌صورت یک بُردار تعریف می‌شوند که هر بُعد متعلّق به یک کلمه است. اگر کلمه در سند موجود باشد، بُعد متناظر آن غیر صفر است. روش Lnu-Itu یک روش بُرداری است و از فرمول زیر برای رتبه‌بندی استفاده می‌کند.

$$S(d)=\frac{((1+\log(tf))*(1+\log(qtf))*\log((1+N)/n))}{((1+\log(average(tf)))*((1-s)+s*N.U.W/averageN.U.W)^2} \quad (1)$$

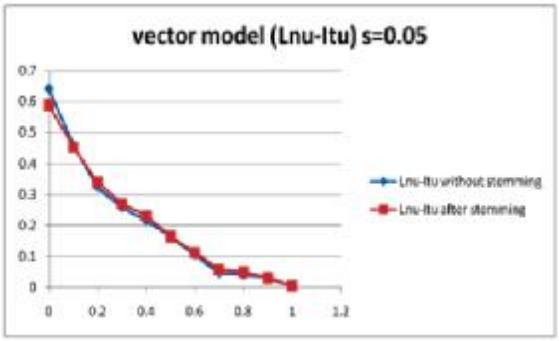
در فرمول (۱) d نماد یک سند است و S(d) امتیاز مربوط به سند d است. tf فرکانس تکرار کلمه در سند d، qtf فرکانس تکرار کلمه در درخواست، N تعداد سندها، n تعداد کلمات، N.U.W تعداد کلمات یکتا در سند d می‌باشد. s پارامتر ثابتی در وزن‌دهی است که مقدار آن با آموزش از یک مجموعه داده تنظیم می‌گردد. در (شکل ۲)، میانگین متوسط دقّت[12] بر اساس مقادیر مختلف s به‌دست آمده است و نشان می‌دهد که بالاترین دقّت در s برابر با ۰/۰۵ است و مقدار آن برابر با ۰/۳۸۳۹ می‌باشد.

[1] Judgment
[2] Hamshahri Corpus, http://ece. ut. ac.ir/dbrg/hamshahri/
[3] Title
[4] Description
[5] Tag
[6] Database research group
[7] Precision
[8] Recall
[9] Terrier Information Retrieval Platform, http: //ir. dcs. gla. ac.uk/terrier
[10] Stopword
[11] Unique terms
[12] Mean average precision



(شکل ۲): نمودار میانگین متوسط دقّت برحسب مقادیر مختلف s

(شکل ۳)، مقایسهٔ روش بدون ریشه‌یابی و پس از ریشه‌یابی را نشان می‌دهد. در هر دو حالت، کلمات پر تکرار حذف شده‌اند. میانگین متوسط دقّت در حالت بدون ریشه‌یابی، برابر با ۰/۱۹۰۱ و در حالت ریشه‌یابی شده برابر با ۰/۱۹۲۳ است.



(شکل ۳): نمودار دقّت - فراخوانی برای دو اجرای ریشه‌یابی نشده وریشه‌یابی شده

۵-۲- استفاده از مدل زبان

از مهم‌ترین مسائل در استفاده از مدل زبان، روش‌هایی است که برای به‌دست آوردن مدل زبانی سند استفاده می‌شود؛ مانند روش‌های هموارسازی[1]. در این مقاله از روش Hiemstra استفاده شده است. جزییات این روش در (*Hiemstra, 2001*) بیان شده است. این روش چهار فرمول دارد. معادلات زیر روش‌های رتبه‌بندی را در این روش نشان می‌دهند (*Hiemstra, 2001*).

$$S1(d)=\sum_{i=1}^{n}\log(1+\frac{\lambda_i tf(t_i,d)(\sum_t cf(t))}{(1-\lambda_i)cf(t_i)(\sum_t tf(t,d))}) \quad (۲)$$

$$S2(d)=\sum_{i=1}^{n}\log(1+\frac{\lambda_i tf(t_i,d)(\sum_t df(t))}{(1-\lambda_i)df(t_i)(\sum_t tf(t,d))}) \quad (۳)$$

$$S3(d)=\log(\sum_t tf(t,d))+\sum_{i=1}^{n}\log(1+\frac{\lambda_i tf(t_i,d)(\sum_t cf(t))}{(1-\lambda_i)cf(t_i)(\sum_t tf(t,d))}) \quad (۴)$$

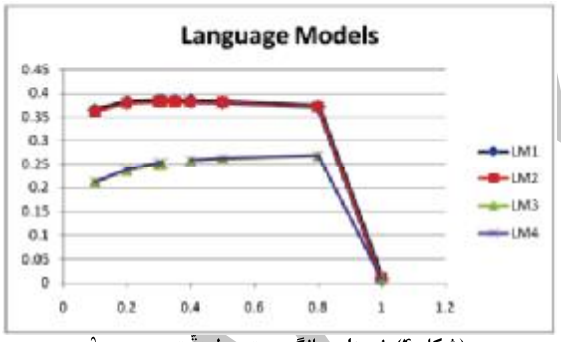
[1] Smoothing

$$S4(d)=\log(\sum_t tf(t,d))+\sum_{i=1}^{n}\log(1+\frac{\lambda_i tf(t_i,d)(\sum_t df(t))}{(1-\lambda_i)df(t_i)(\sum_t tf(t,d))}) \quad (۵)$$

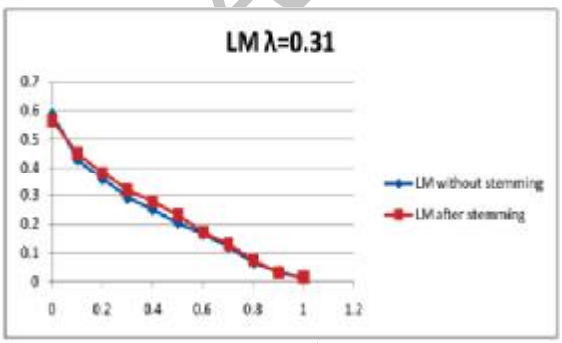
در هر چهار فرمول S امتیازی است که به سند تعلّق می‌گیرد. tf بیانگر فرکانس تکرار کلمه در سند d است. در فرمول (۲) برای هموارکردن[2] میزان تکرار کلمات از cf که میزان تکرار کلمه در کل مجموعه[3] است، استفاده شده و در فرمول سه به جای cf از df استفاده شده است که تعداد سندهایی را نشان می‌دهد که کلمه در آنها آمده است. در فرمول‌های (۳) و (۴) مقداری برای نرمال‌کردن طول سند اضافه شده است. پارامتر λ پارامتری است که مقدار آن باید با یادگیری تنظیم شود.

در (شکل ۴) میانگین متوسط دقت بر اساس مقادیر مختلف λ در هر چهار روش به‌دست آمده است و نشان می‌دهد که بالاترین دقّت در روش اوّل با مقدار λ برابر با ۰/۳۱ است و مقدار آن برابر با ۰/۳۸۴۲ می‌باشد.

(شکل ۵) مقایسهٔ روش بدون ریشه‌یابی و پس از ریشه‌یابی را نشان می‌دهد. میانگین متوسط دقّت در حالت بدون ریشه‌یابی برابر با ۰/۲۱۰۱ و در حالت ریشه‌یابی شده برابر با ۰/۲۲۱۷ است. بنابراین میزان میانگین متوسط دقّت پنج درصد افزایش یافته است.



(شکل ۴): نمودار میانگین متوسط دقّت برحسب λ در چهار روش مدل زبانی



(شکل ۵): نمودار دقّت - فراخوانی برای دو اجرای ریشه‌یابی نشده و ریشه‌یابی شده

[2] Smoothing
[3] Collection frequency

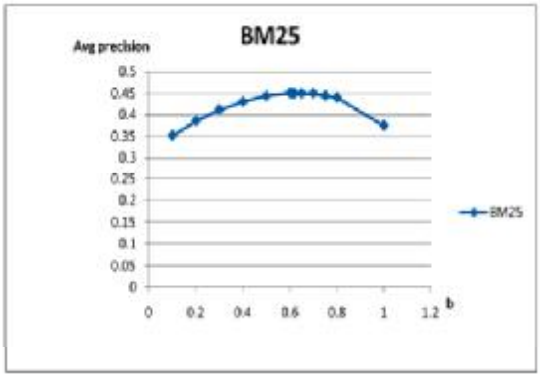
۵-۳ مدل آماری

از بین روش‌های آماری، روش BM25 (*Hiemstra, 2001*) انتخاب شده است. در این روش اسناد براساس حضور کلمات درخواستی در متن بدون توجه به ارتباط کلمات درخواست رتبه‌بندی می‌شوند. در واقع معادله رتبه‌بندی آن از مجموعه‌ای از توابع رتبه‌بندی استفاده می‌کند. فرمول رتبه بندی آن به صورت زیر است.

$$S(d) = \sum_{r \in Q,D} \ln \frac{N - df + 0.5}{df + o.5} \cdot \frac{(k_1 + 1)tf}{(k_1((1-b) + b\frac{dl}{avdl})) + tf} \cdot \frac{(k_3 + 1)qtf}{k_3 + qtf} \quad (۶)$$

در معادلهٔ (۶) tf، تعداد تکرار کلمه در سند و qtf، تعداد تکرار کلمه در درخواست، N، تعداد کل سندها، df، تعداد سندهایی که شامل کلمه مورد نظر می‌باشند، dl، اندازه سندها (بایت) و avdl، میانگین اندازهٔ کل سندها است. پارامتر b پارامتری است که باید مقدار آن تنظیم شود.

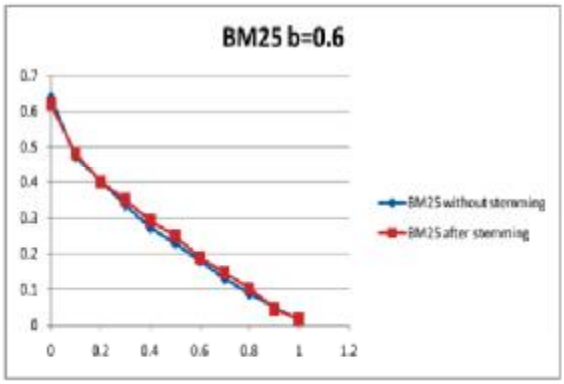
(شکل ۶) میانگین متوسط دقّت بر اساس مقادیر مختلف b را نشان می‌دهد. بالاترین دقّت در b برابر با ۰/۶ و مقدار آن برابر با ۰/۴۵۰۹، که از مقادیر به‌دست آمده در دو روش قبلی بیشتر است.



(شکل ۶): نمودار میانگین متوسط دقّت برحسب b

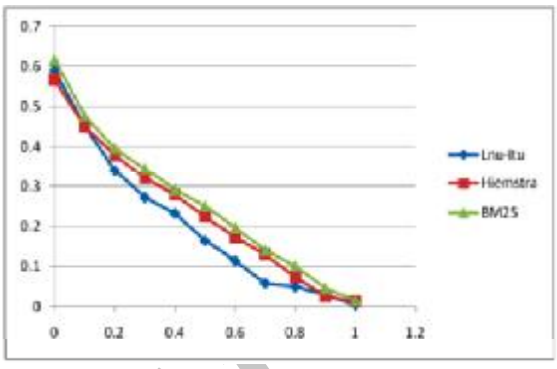
(شکل ۷) مقایسهٔ روش بدون ریشه‌یابی و پس از ریشه‌یابی را نشان می‌دهد. میانگین متوسط دقّت در حالت بدون ریشه‌یابی برابر با ۰/۲۳۲۱ و در حالت ریشه‌یابی شده برابر با ۰/۲۴۴۷ است. بنابراین استفاده از ریشه‌یاب پنج درصد در بهبود میانگین متوسط دقت موثر بوده است.

از آنجا که نتایج به‌دست آمده در نمودارها بسیار نزدیک می‌باشند برای ارزیابی اهمیت نتایج به‌دست آمده، از آزمون Wilcoxon[1] استفاده شده است. این آزمون برای مقادیر به‌دست آمده در روش BM25 به‌دست آمده است. مقدار ۱/۹۳ برای z-ration به‌دست آمده و درجهٔ اهمیت برای آزمون directional، ۰/۰۲۶۸ و برای non-directional، ۰/۰۵۳۶ به‌دست آمده است.



(شکل ۷): نمودار دقّت - فراخوانی برای دو اجرای ریشه‌یابی نشده و ریشه‌یابی شده

مقایسهٔ هر سه روش فوق در حالت ریشه‌یابی‌شده در (شکل ۸) نشان داده شده است که دیده می‌شود روش BM25 از روش‌های دیگر عملکرد بهتری دارد.



(شکل ۸): مقایسه سه روش فوق در حالت ریشه‌یابی شده

۶- نتیجه‌گیری و ادامهٔ کار

در این مقاله روشی مشابه روش پورتر برای ریشه‌یابی کلمات فارسی ارائه شده است. از آنجا که پسوندها در زبان فارسی به‌خوبی قابل تشخیص نمی‌باشند، به‌عنوان مثال «ان» در بعضی کلمات علامت جمع و در بعضی متعلق به خود کلمه است؛ درنتیجه در هر قانون یا شرایطی برای حذف پسوند در نظر گرفته شده است و یا کلمه ایجاد شده در مجموعه لغات جستجو می‌گردد تا از ایجاد ریشه‌های نادرست جلوگیری شود. طبق نتایج به‌دست آمده دیده می‌شود که ریشه‌یاب

[1] Wilcoxon signed rank test (Wilocoxon signed rank test, http: //faculty. vassar. edu/lowry/ch12a.htm)

پیاده‌سازی شده، نتایج قابل قبولی از نظر دقّت بازیابی به‌دست می‌دهد. نتایج به‌دست آمده بر اساس آزمایش برروی پیکرهٔ همشهری می‌باشد که شامل حدود ۱۶۵،۰۰۰ سند است. می‌توان برای نتیجه‌گیری بهتر، آزمایش‌ها را بر روی پیکره‌های دیگری با تعداد بیشتری سند تکرار کرد. همچنین می‌توان روش‌های دیگری را برای رتبه‌بندی انتخاب و مورد آزمایش قرار داد.

۷- مراجع

Aleahmad, A., Amiri, H., Oroumchian, F., and Rahgozar, M., 2009. "Hamshahri: A standard Persian text collection", Knowledge-Based systems, Vol. 22 No. 5, pp. 382-387.

Bacchin, M., Ferro, N. and Melucci, M., 2005. "A probabilistic model for stemmer generation" Information Processing & Management, Vol. 41, No. 1, pp. 121–137.

Bijankhan, M., 2006. "Naghshe Peykarehaye Zabani dar Neveshtane Dasture Zaban: Mo'arrefiye yek Narmafzare Rayane'i [theRole of Corpus in generating grammar: Presenting a computational software and Corpus] ", Iranian Linguistic Journal, Vol. 19 pp. 48-67.

Dolamic, L., Fautsch, C. and Savoy, J., 2009. "UniNE at CLEF 2009: Persian ad hoc retrieval and IP", in CLEF Workshop, Part I. LNCS, Vol. 6241, Springer, Heideberg.

Givi, H.A, Anvari, H., 2006. "Persian Language", 27th ed., Fatemi, Tehran.

Hiemstra, D. and Vries, A.P., 2000. "Relating the new language models of information retrieval to the traditional retrieval models", University of Twente, Centre for Telematics and Information Technology Technical Report.

Hiemstra, D., 2001. "Using Language Models for Information Retrieval. " PhD thesis, University of Twente.

Jones, K.S, Walker, S. and Robertson, S.E., 2000. "A Probabilistic Model of Information Retrieval: Development and Comparative Experiments (parts1 and 2) ", Information Processing and Management, Vol. 36, No. 6, pp. 779-840.

Lovins, J.B., 1968. "Development of a Stemming Algorithm", Mechanical Translation and computation Linguistics, pp 23-31.

MohammadiNasiri, M., SheykhEsmaili, K. and Abolhassani, H., 2006. "A Statistical Stemmer for Persian Language", 11th International CSI Computer Conference, Tehran, Iran, Jan.

Megerdoomian, K., 2004. "Finite-state morphological analysis of Persian, " in Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages, pp. 35 – 41.

Porter, M.F., 1980. "An algorithm for suffix stripping", Program, Vol. 14 No. 3, pp. 130-137, July.

Rahimtoroghi, E., Faili, H. and Shakery, A., 2010. "A Structural Rule-based Stemmer for Persian", International Symposium on telecommunications.

Sharifloo, A. and Shamsfard, M., 2008. "A Bottom up Approach to Persian Stemming", In Proceedings of the Third International Joint Conference on Natural Language Processing.

Shamsfard, M., Jafari, H.S. and Ilbeygi, M., 2010. "STeP-1: A Set of Fundamental Tools for Persian Text Processing. " In 8th Language Resources and Evaluation Conference.

Singhal, A., Buckley, C. and Mitra, M., 1996. "Pivoted document length normalization", In Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 21 – 29.

Taghva, K., Beckley, R. and Sadeh, M., 2005. "A Stemming Algorithm for the Farsi Language", International Conference on Information Technology: Coding and Computing.

Tashakori, M., Meybodi, M.R and Oroumchian, F., 2003. "Bon: The Persian stemmer", in Proc. 1st EurAsian Conference. on Information.

نوا احسان مدرک کارشناسی مهندسی کامپیوتر (نرم‌افزار) را در سال ۱۳۸۴ از دانشگاه صنعتی شریف و مدرک کارشناسی ارشد را در سال ۱۳۹۰ در رشتهٔ مهندسی کامپیوتر (نرم‌افزار) از دانشگاه تهران اخذ نموده است. از سال ۱۳۹۰ دانشجوی دکتری رشتهٔ مهندسی کامپیوتر (فناوری اطّلاعات) دانشگاه تهران می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: پردازش زبان طبیعی، بازیابی اطّلاعات و داده کاوی.

نشانی رایانامک ایشان عبارت است از:

n.ehsan@ece.ut.ac.ir

هشام فیلی تحصیلات خود را در مقطع کارشناسی نرم‌افزار در دانشکدهٔ مهندسی کامپیوتر دانشگاه صنعتی شریف با رتبهٔ یک در سال ۱۳۷۶ به پایان رساند. سپس مقاطع کارشناسی ارشد نرم‌افزار و دکتری هوش مصنوعی را به‌ترتیب در سال‌های ۱۳۷۸ و ۱۳۸۵ در همان دانشکده تکمیل کرد. از سال ۱۳۸۷ تاکنون عضو هیئت علمی دانشکدهٔ مهندسی برق و کامپیوتر دانشکدهٔ فنی دانشگاه تهران است. زمینه‌های تحقیقاتی مورد علاقهٔ ایشان عبارتند از: پردازش هوشمند متن و زبان طبیعی، مترجم ماشینی، داده کاوی، بازیابی اطّلاعات و شبکه‌های اجتماعی می‌باشد.

نشانی رایانامک ایشان عبارت است از:

hfaili@ut.ac.ir