ارائهٔ یک روش مبتنی بر مدل زبانی برای واحدسازی پیکرهٔ فارسی

مسعود قیومی[1]
پژوهشگاه علوم انسانی و مطالعات فرهنگی
تاریخ دریافت: ۱۳۹۶/۰۴/۱۳ تاریخ پذیرش: ۱۳۹۷/۰۲/۳۰

چکیده

متن نگاشته‌شدهٔ فارسی دو مشکل ساده ولی مهم دارد. مشکل اول واژه‌های چندواحدی هستند که از اتصال یک واژه به واژه‌های بعدی حاصل می‌شوند. مشکل دیگر واحدهای چندواژه‌ای هستند که از جداشدگی واژه‌هایی که با هم یک واحد واژگانی تشکیل می‌دهند حاصل می‌گردند. این مقاله الگوریتمی را معرفی می‌کند که بتواند به‌طور خودکار این دو مشکل را در متن نوشتاریِ فارسی بکاهد و یک متن معیار را به‌دست آورد. الگوریتمِ معرفی‌شده سه مرحله دارد. در مرحلهٔ اول، واژه‌های چندواحدی از هم جدا می‌شوند و واحدهای چندواژه‌ای به یکدیگر متصل می‌شوند. برای این مرحله، یک الگوریتم پایهٔ مبتنی‌بر مدل زبانی معرفی شده‌است که کار تفکیک واژه‌های چندواحدی به واژه‌های مستقل را انجام می‌دهد. این الگوریتم باتوجه‌به چالش‌های پیش‌آمده بهبود می‌یابد تا کارایی آن افزایش یابد. همچنین این مرحله از یک تحلیل‌گرِ صرفی برای بررسی وندِ تصریفی و اشتقاقی و روش انطباق فهرست واژه برای رفع مشکل واحدهای چندواژه‌ای استفاده می‌کند. در مرحلهٔ دوم، از روش انطباق برای بررسیِ چندواژگیِ افعال استفاده می‌شود. مرحلهٔ سوم تکرار مرحلهٔ اول است تا مشکلات جدید ایجادشده در متن بعداز اجرای مرحلهٔ دوم مرتفع شود. الگوریتم معرفی‌شده برای واحدسازی دادهٔ زبانیِ پایگاه داده‌های زبان فارسی استفاده شده‌است. با استفاده از این الگوریتم، ۷۲/۰۴ درصد خطای نگارشی واژه‌های دادهٔ آزمون تصحیح شده‌است. دقت این تصحیح در دادهٔ آزمون

1. M.Ghayoomi@ihcs.ac.ir

۹۷/۸۰ درصد و خطای نگارشی ایجادشده توسط این الگوریتم در دادهٔ آزمون ۰/۰۲ درصد است.

کلیدواژه‌ها: پردازش زبان طبیعی، واحدسازی داده، مدل‌سازی زبانی آماری، زبان‌شناسی پیکره‌ای.

۱- مقدمه

امروزه گسترش فناوری‌های مرتبط با زبان طبیعی، مانند غلط‌یاب‌های املایی و دستوری، ماشین ترجمه و غیره، و تنیده‌شدن این نوع فناوری با تاروپود زندگی، به تهیهٔ سامانه‌هایی منجر شده‌است که از زبان طبیعی به‌عنوان ابزار ارتباطی برای یک هدف مشخص استفاده می‌کند. افزایش پذیرشِ عمومیِ این سامانه‌ها در گروِ کاراییِ هرچه بیشتر آنهاست و برای داشتن کارایی بالاتر، نیاز است که این سامانه به اندازهٔ کافی با دادهٔ زبانی آموزش ببیند. هرچقدر نوفهٔ دادهٔ آموزش[1] کمتر باشد، مدل دقیق‌تری به دست می آید.

اساساً تهیهٔ داده‌ای که نوفه نداشته باشد بسیار سخت است و سبب می‌شود مدل زبانیِ تهیه‌شده فقط به این حجم دادهٔ آموزش محدود شود و کارایی خود را به هنگامِ استفادهٔ عملی ازدست بدهد. برای دسترسی به حجم بیشتر داده، نیاز است سامانه‌ای تهیه شود که بتواند به‌طور خودکار نوفهٔ داده را به حداقل برساند. هدف این مقاله، معرفی الگوریتمی است که با کمک مدل زبانیِ آماری تلاش می‌کند دادهٔ غیرمعیار پرنوفه را به داده‌ای نزدیک به دادهٔ معیار با حداقل نوفه تبدیل کند.

ساختار مقالهٔ حاضر به این شرح است: پس از مقدمه، در بخش ۲ عمدهٔ مشکلاتی که در پردازش دادهٔ فارسی با آنها مواجه می‌شویم معرفی می‌شود. در بخش ۳ به مطالعات و اقدامات عملیِ انجام‌شده برای معیارسازی و واحدسازیِ پیکرهٔ فارسی پرداخته می‌شود. در بخش ۴ دادهٔ فارسی که برای کاربرد عملی و واحدسازیِ داده به‌کار رفته‌است توصیف می‌شود. بخش ۵ به معرفی سامانهٔ واحدساز دادهٔ فارسی و بخش‌های آن می‌پردازد. در بخش ۶ به ارزیابی مدل معرفی‌شده پرداخته شده و مقاله با نتیجه‌گیری در بخش ۷ به پایان می‌رسد.

1. training data

۲- چالش‌های پردازش خط فارسی

در پردازش خط فارسی، چالش‌هایی وجود دارد که با فائق‌آمدن بر آنها می‌توان به دادهٔ معیار دست یافت. قیومی و ممتازی (۲۰۰۹)، قیومی و همکاران (۲۰۱۰)، و شمس‌فرد (۲۰۱۱) عمدهٔ عوامل چالش‌زای پردازش خط فارسی را معرفی کرده‌اند که از میان آنها به دو مشکل اصلی اشاره می‌کنیم.

۲-۱- مرز واژه

شریفی‌آتشگاه و بی‌جن‌خان (۲۰۰۹) دو دسته مشکل در تعیین مرز واژه را با عناوین واحدهای چندواژه‌ای[1] و واژه‌های چندواحدی[2] مطرح کرده‌اند. دستهٔ اول واژه‌هایی هستند که به‌دلیل ویژگی‌های نوشتاریِ بعضی حروف در خط فارسی و همچنین ویژگی زیباشناختیِ خط فارسی منفصل نوشته می‌شوند و این واژه‌های منفصل باید با هم ترکیب و به‌صورت یک واحد درنظر گرفته شوند؛ مانند «به‌نظررسیدن» که معمولاً به‌صورت «به نظر رسیدن» نوشته می‌شود. واژه‌های دستهٔ دوم واژه‌های چندواحدی هستند که به‌دلیل ویژگی‌های خط فارسی، به‌اشتباه به یکدیگر متصل می‌شوند و در اصل چند واژهٔ مستقل‌اند، مانند «و یا بهتر است» که ممکن است به‌دلیل عدم درج فاصله بین واژه‌ها به‌صورت «ویابهتراست» نوشته شود. دلیل اصلیِ ایجادِ مشکل در دستهٔ دوم، مربوط به مجموعه‌ای از حروف مانند «آ»، «ا»، «د»، «ذ»، «ر»، «ز»، «ژ»، و «و» است که شکل چسبیده به حرف مجاورِ بعدی را ندارند؛ بنابراین، به‌هنگام تحریر این حروف در رایانه، درج فاصله نادیده گرفته می‌شود.

۲-۲- تنوع در نگارش واژه

رعایت معیارنویسی یکی از ویژگی‌های زبان نوشتاری است که تخطی از آن، اشتباه نگارشی محسوب می‌شود. به‌رغمِ تلاشِ فرهنگستان زبان و ادب فارسی (۱۳۸۹) برای معیارسازیِ نوشتارِ واژه‌ها در زبان فارسی، وجود مشکلات یا محدودیت‌ها در پشتیبانی خط فارسی در فناوری‌های امروزی سبب شده‌است تا گویشوران این زبان به‌دلخواه خودشان املای واژه‌ها را تغییر دهند که

1. multi-token unit
2. multi-unit token

نتیجهٔ آن ایجاد مشکل تنوع نگارشی برای یک واژه است. انواع نمونه‌های این تنوع‌ها در قیومی و همکاران (۲۰۱۰) ذکر شده‌است که باتوجه‌به هدف این مقاله، می‌توان به مواردی چون «علاقه‌مند»، «علاقه مند» و «علاقمند» اشاره کرد. قیومی و همکاران (۱۳۹۴) تلاش کرده‌اند به‌طور خودکار تنوع نگارشیِ واژه‌هایی که به هم مرتبط هستند را تشخیص دهند.

۳- پیشینهٔ مطالعاتی

معیارسازیِ خط فارسی ازجمله وظایف فرهنگستان زبان و ادب فارسی است که در همین راستا این سازمان دستور خط فارسی (۱۳۸۹) را تهیه کرده‌است. این دستور خط شامل قواعد نگارشی و فهرستی از استثناهاست. بلندبودن فهرست استثناها و وجود محدودیت‌ها در پشتیبانی خط فارسی در فناوری‌های جدید امروز، سبب کمرنگ‌شدنِ معیارنویسی شده‌است. اهمیت و تأثیرِ نحوهٔ نگارش واژه در پردازش داده‌های زبانی توجه برخی پژوهشگران را به خود جلب کرده‌است. در ادامه به بعضی از پژوهش‌های صورت‌گرفته در این زمینه اشاره می‌شود.

شریفی‌آتشگاه (۱۳۸۸) در رسالهٔ دکتری خود تلاش کرده‌است با تهیهٔ فهرست و الگو، واژه‌های فارسی را از دو جنبهٔ ایستایی یا پویایی معیارسازی و واحدسازی کند. واژه‌های چندجزئیِ ایستا بسیط و غیرزایا هستند که برای این دسته از واژه‌ها از فهرست واژه و روش انطباق استفاده می‌شود؛ درحالی‌که واژه‌های چندجزئیِ پویا باز و زایا بوده و برای تشخیص این دسته از واژه‌ها از الگوهای قاعده‌مند استفاده می‌شود.

شمس‌فرد و همکاران (۲۰۱۰) مجموعهٔ ابزارهایی را تهیه کرده‌اند که با استفاده از آنها می‌توان متن را باتوجه‌به دستور خط فرهنگستان زبان و ادب فارسی معیارسازی کرد. این ابزار متن‌باز[1] نبوده و به‌صورت رایگان نبز در دسترس نیست. این ابزار سه کار تصحیح و تقطیع واژگانی، تحلیل واژگانی و برچسب‌گذاریِ مقولهٔ دستوری را انجام می‌دهد.

سراجی و همکاران (۲۰۱۲) و سرابی و همکاران (۲۰۱۳) مجموعهٔ ابزارهایی را تهیه کرده‌اند که پس‌از پیش‌پردازشِ دادهٔ ورودی، کار برچسب‌گذاریِ مقولات دستوری و همچنین تجزیهٔ

1. open source

نحویِ جملات در چارچوب دستور وابستگی را انجام می‌دهد. هضم[1] نیز مجموعهٔ ابزارهای متن‌باز به زبان پایتون[2] است که مانند مطالعهٔ سراجی و همکاران (۲۰۱۲) و سرابی و همکاران (۲۰۱۳) عمل می‌کند.

کاشفی (۱۳۹۰) ابزار متن‌بازی به‌نام «ویراستیار» را تهیه کرده‌است که به‌عنوان یک غلط‌یابِ املایی، کار واحدسازیِ واژه‌های فارسی را انجام می‌دهد. همچنین، فیلی و همکاران (۲۰۱۶) ابزاری به‌نام «وفا» را تهیه کرده‌اند که به‌عنوان یک غلط‌یابِ املایی، با بهره‌گیری از روش‌های قاعده‌بنیان و آماری متن فارسی را به متن معیار نزدیک می‌کند.

وزیرنژاد و همکاران (۱۳۹۴) ویرایش‌گرِ متن «شریف» را تهیه کرده‌اند که نوشتار فارسی را منطبق بر دستور خط فرهنگستان زبان و ادب فارسی معیارسازی می‌کند.

احمدیان و فیلی (۱۳۹۵) از درخت تصمیم[3] به‌عنوان یکی‌از روش‌های یادگیری ماشین[4] برای واحدسازی واژه‌های فارسی استفاده کرده‌اند. آنها نتیجهٔ به‌دست‌آمده را با واحدسازیِ قاعده‌بنیان مقایسه کرده و نتیجهٔ نزدیک به مدل قاعده‌بنیان را به‌دست آورده‌اند.

طباطبایی و صراف (۱۳۹۶) با استفاده از ترکیبِ دو شیوهٔ پیکره‌محور و قاعده‌محور تلاش کرده‌اند مشکل معیارسازی و واحدسازی در پیکرهٔ فارسی را رفع کنند. در شیوهٔ پیکره‌محور، از روش انطباق و فهرست واژگان حاصل از پیکره متنی «پیکره» (بی‌جن‌خان و همکاران، ۲۰۱۱) استفاده شده‌است.

قیومی (۱۳۹۶) برای فائق‌آمدن بر مشکل چندواژگی در حوزهٔ پردازش نحوی، سه الگوریتم معرفی کرده‌است. در یک الگوریتم، واژه‌بست به‌صورت خودکار از ابتدا یا انتهای واژهٔ میزبان جدا شده و از خروجی آن در تحلیلِ نحویِ عمیقِ جمله استفاده می‌گردد. در دو الگوریتم دیگر، تفکیک واژه‌های چندواحدی از هم برای تشخیص واژه‌های مستقل و همچنین ترکیب واحدهای چندواژه‌ای برای تشکیل یک واحد واژگانی مورد توجه بوده‌است. در پژوهش حاضر تلاش می‌شود کاستی‌های الگوریتمِ جداسازیِ واژه‌های چندواحدی پوشش داده و سبب افزایش کارایی الگوریتم شود.

1. https://github.com/sobhe/hazm/
2. python
3. decision tree
4. machine learning

معیارسازی و واحدسازیِ متنِ ورودی، یکی‌از قسمت‌های مهم پیش‌پردازش است؛ و مطالعات انجام‌شده مبیّن اهمیت جایگاه معیارسازی و واحدسازیِ دادهٔ فارسی و پردازش خودکار آن است. ازجمله مطالعات انجام‌شده در استفاده از مدل زبانی برای معیارسازی سایر زبان‌ها، می‌توان به مدل آماریِ تهیه‌شده توسط آدا[۱] و همکاران (۱۹۹۷) اشاره کرد که در آن برای معیارسازیِ دادهٔ مورد نیاز در سامانهٔ پردازش گفتار از سرگشتگی[۲] استفاده شده‌است. یانگ[۳] و آیزن‌اشتاین[۴] (۲۰۱۳) تلاش کرده‌اند از مدل لگاریتم خطی[۵] به روش بی‌مربی[۶] برای ساخت مدل زبانی استفاده کنند تا از آن برای معیارسازیِ دادهٔ زبانیِ رسانه‌های مجازی بهره ببرند. در یک بررسی دیگر، لی[۷] و لیو[۸] (۲۰۱۴) هم تلاش کرده‌اند به روش بی‌مربی داده‌های زبانیِ رسانه‌های مجازی مانند فیس‌بوک را که توسط کاربران مختلف تولید شده‌اند را معیارسازی کنند، تا امکان استفاده از روش‌های معیار پردازش زبان میسر شود. اسکانل[۹] (۲۰۱۴) نیز از مدل زبانی سه‌نگاشتی[۱۰] به‌همراه «تخفیف مطلق»[۱۱] برای معیارسازیِ خط در ترجمهٔ ماشینیِ زبانِ مقصد استفاده کرده‌است.

۴- دادهٔ زبانی

در انجام این مطالعه، دو نوع دادهٔ زبانی مورد نیاز است. دادهٔ اول، دادهٔ معیار[۱۲] است که برای تهیهٔ مدل زبانیِ معرفی‌شده در بخش ۵ کاربرد دارد. برای این منظور، از پیکرهٔ بی‌جن‌خان (۱۳۸۳) با حدود ۲/۵ میلیون واژه استفاده می‌شود. ویژگی این پیکره این است که مقولهٔ دستوریِ واژه‌ها به‌صورت نیمه‌خودکار مشخص شده‌است، بنابراین این پیکره توسط انسان بازبینی شده‌است، اشکالات نوشتاریِ مربوط به خط در آن به حداقل رسیده‌است و می‌تواند به‌عنوان دادهٔ معیار این مقاله مورد استفاده قرار گیرد.

1. G. Adda
2. Perplexity
3. Y. Yang
4. J. Eisenstein
5. log-linear
6. Unsupervised
7. C. Li
8. Y. Liu
9. K. Scannell
10. trigram
11. absolute discounting
12. gold standard data
13. target data

دادهٔ دوم، دادهٔ هدف[۱۳] است به این معنا که اشکالات نوشتاری مربوط به خط در آن تصحیح نشده‌است؛ و نیاز است با کمک مدلِ معرفی‌شده، اشکالات خطیِ پیکرهٔ هدف باتوجه‌به پیکرهٔ معیار برطرف شود. در انجام این پژوهش، پایگاه داده‌های زبان فارسی (عاصی، ۱۳۸۴) به‌عنوان دادهٔ هدف استفاده می‌شود. این پیکره نسبتاً متعادل[1] است و متون نوشتاری کتاب، مجله، روزنامه، نمایشنامه، داستان کودکان، و گفتار رسمی و غیررسمی را دربرمی‌گیرد. این پیکره با حدود ۶۰ میلیون واژه، از پیکرهٔ بی‌جن‌خان بزرگتر است و علاوه‌بر فارسی معاصر، داده‌های زبانی از قرن‌های ۵ تا ۷ هجری شمسی را نیز دربر دارد (عاصی و قندی، ۱۳۹۴). بخش کوچکی از داده‌های این پیکره شامل اطلاعات زبان‌شناختی، مانند برچسب آوایی، بن‌واژه و مقولهٔ دستوری، است که عمدهٔ برچسب‌ها به‌صورت دستی مشخص شده‌است.

۵- معرفی روش واحدسازی داده‌های فارسی

در این بخش، به معرفی مدل زبانی تهیه‌شده برای واحدسازیِ متن فارسی می‌پردازیم. با درنظرداشتنِ بررسیِ شریفی‌آتشگاه و بی‌جن‌خان (۲۰۰۹)، در مدل ارائه‌شده تلاش شده‌است بررسی واژه‌های چندواحدی و واحدهای چندواژه‌ای در سه مرحله انجام پذیرد. بررسی واژه‌های چندواحدی نسبت‌به واحدهای چندواژه‌ای دارای اولویت است تا ابتدا واژه‌های به‌اشتباه به هم متصل‌شده از هم منفصل شوند. سپس، وندهای تصریفی یا اشتقاقیِ منفصل‌شده از واژهٔ پایه به هم متصل شوند و یک واحد واژگانی را ایجاد کنند. لازم به ذکر است که در این پژوهش، انجام واحدسازی بیشتر بر روی واژه‌های ساده و نَه واحدهای چندواژه‌ای، مانند حرف اضافهٔ مرکب یا حرف ربط مرکب، متمرکز است. در ادامه، بخش‌های مرتبط با این سه مرحله توصیف می‌شود. در بخش ۵-۱، بخش‌های مختلف روش پیشنهادی و چالش‌های پیش‌رو که باید در طراحی الگوریتم در نظر گرفته شود تشریح می‌شود؛ و سپس در بخش۵-۲، الگوریتم مورد نظر به صورت منسجم توصیف می‌شود.

1. balanced

۵-۱- کلیات و چالش‌ها

۵-۱-۱- استخراج اطلاعات آماری n-نگاشت[1]

اساساً برای تهیهٔ مدل زبانی آماری به اطلاعات n-نگاشت نیاز است که این اطلاعات از پیکرهٔ زبانی استخراج می‌شود. در بخش ۴، ویژگی‌های پیکره‌های استفاده‌شده توضیح داده شده‌است. برای تهیهٔ مدل، دو نوع اطلاعات n-نگاشتی استخراج می‌شود: الف) استخراج اطلاعات آماری تک‌نگاشتی[2] و دونگاشتیِ[3] واژه‌ها از پیکرهٔ معیار. به‌هنگام استخراجِ اطلاعات، علایم سجاوندی در این پیکره نادیده گرفته می‌شود؛ ب) استخراج اطلاعات دونگاشتی واژه‌ها از پیکرهٔ هدف.

۵-۱-۲- الگوریتم پایهٔ تفکیک

برای حل مشکل واژه‌های چندواحدی، از یک الگوریتم بازگشتی[4] استفاده می‌شود. در این الگوریتم تنها به یک نوع اطلاعات نیاز است که عبارت است از فهرست واژگان موجود در اطلاعات آماریِ تک‌نگاشتی واژه‌ها از پیکرهٔ معیار. این اطلاعات از این پس «واژگان معیار» نامیده می‌شود.

در این الگوریتم، ابتدا وجود یا نبودِ زنجیرهٔ ورودی در این فهرست واژه کنترل می‌شود. درصورت نیافتنِ واژه در این فهرست، بررسی حرف‌به‌حرف زنجیره انجام می‌شود. چنانچه به‌هنگام خواندن آن زنجیره از ابتدا، واژه‌ای از توالی این حروف به دست آید که در واژگان معیار وجود دارد، آن واژه از ابتدای زنجیره به‌طور موقت جدا می‌شود و بقیهٔ زنجیره به‌صورت بازگشتی برطبق همین الگوریتم بررسی می‌شود. اگر اجرای الگوریتم بر روی بقیهٔ زنجیره پاسخ مثبتی دهد، تفکیک انجام‌شده مورد تأیید قرار می‌گیرد و الگوریتم به کل این زنجیرهٔ تفکیک‌شده پاسخ مثبت می‌دهد. این بدین مفهوم است که بقیهٔ زنجیره نیز در فهرست واژگان وجود دارد و یا به همین ترتیب در مراحل بعد قابل‌تفکیک است. همان‌طور که گفته شد، درصورت یافتن زنجیره‌ای از حروفِ ابتدای زنجیرهٔ ورودی در فهرست واژگان، جداسازی این

1. n-gram
2. unigram
3. bigram
4. recursive algorithm

حروف به‌صورت موقت انجام می‌پذیرد. این به آن معناست که علاوه بر این جداسازی، الگوریتم همچنان به‌دنبال سایر واژه‌هایی که در ابتدای زنجیرهٔ ورودی قابل تشخیص است نیز می‌گردد و به این ترتیب تمام حالات ممکن تفکیک زنجیرهٔ ورودی را مد نظر قرار می‌دهد. اگرچه خروجی این الگوریتم پیوستاری از واژه‌های تفکیک‌شده به‌صورت کاندید است، الگوریتم فوق قابلیت تشخیص بهترین تفکیک از میان تفکیک‌های کاندید را ندارد.

برای واضح‌شدن کار الگوریتم پایهٔ تفکیک مثالی می‌زنیم. تصور می‌کنیم زنجیرهٔ ورودی «مادراوست» باشد. این زنجیره در واژگان معیار یافت نمی‌شود. بنابراین از ابتدای زنجیره، حرف‌به‌حرف خوانده می‌شود و با واژگان معیار مقایسه می‌شود. چنانچه الگوریتم از توالی حروف این زنجیره، واژه یا واژه‌هایی را از میان واژگان معیار بیابد، پاسخ الگوریتم برای تمام زنجیره مثبت خواهد بود و موجب می‌شود عبارت ورودی به دنباله‌های «ما در اوست» و «مادر اوست» تفکیک گردد.

۵-۱-۳- الگوریتم بهبودیافتهٔ تفکیک

روش توضیح‌داده‌شده در بخش ۵-۱-۲ قادر است تمام تفکیک‌های ممکنِ یک عبارت را بیابد، اما اولویت‌دهی برای این تفکیک‌ها ندارد. می‌توان برای یافتنِ بهترین دنبالهٔ تفکیکی واژگان، از اطلاعات آماریِ تک‌نگاشتی و دونگاشتیِ واژه‌ها از پیکرهٔ معیار بهره برد؛ به این صورت که با کمک این اطلاعات زبانی، به‌هنگام اجرای الگوریتمِ پایه، احتمال دونگاشتیِ هریک از واژه‌های یافت‌شده نیز محاسبه شود. در این روش، خروجی الگوریتم شامل تمام دنباله‌های ممکن از واژه‌های تفکیک‌شده به‌همراه احتمال آنها خواهد بود. درنتیجه، تفکیکی که بالاترین احتمال را دارد، به‌عنوان کاندید و بهترین تفکیک انتخاب می‌شود.

بدیهی است همانند تمامی محاسبات مبتنی بر مدل زبانی، در اینجا نیز نیاز به هموارسازی[1] مدل زبانی احساس می‌شود که برای این منظور از اطلاعات آماریِ تک‌نگاشتی استفاده می‌شود تا درصورتی‌که توالیِ دونگاشتیِ مورد نظر، در اطلاعاتِ استخراج‌شده از پیکرهٔ معیار یافت نشود، بتوان با کمک اطلاعات تک‌نگاشتی، این احتمال را محاسبه کرد و از صفرشدن نتیجه جلوگیری

1. smoothing

شود. لازم به ذکر است که در اینجا بحث تنک‌بودنِ داده[1] مطرح است که برای رفع آن تمهیداتی اندیشیده شده‌است که در بخش ۵-۱-۵ توضیح داده می‌شود.

۵-۱-۴ جلوگیری از تفکیک‌های نادرست

اگرچه الگوریتمِ توصیف‌شده قادر خواهد بود دنبالهٔ تفکیک‌شده‌ای از واژه‌ها را برای هر عبارتِ ورودی به دست آورد، باتوجه‌به وجود تعداد زیادی از واژه‌های کوتاهِ دو یا سه‌حرفی در واژگان معیار و باتوجه‌به بالابودنِ بسامد این واژه‌ها، این امکان وجود دارد که عبارت ورودی به دنبالهٔ نادرستی از واژه‌های بسیارکوتاه تفکیک شود. برای جلوگیری از این مشکل، پس‌از اجرای الگوریتم، در یک مرحلهٔ تکمیلی، دنبالهٔ خروجی مورد بررسی مجدد قرار می‌گیرد. به این صورت که به‌ازای هر دونگاشتِ موجود در دنبالهٔ خروجی، چنانچه طول یکی از دو واژه سه حرف یا کمتر باشد، وجود دنبالهٔ دونگاشتی در اطلاعات دونگاشتیِ داده‌ها مورد بررسی قرار می‌گیرد. از آنجا که میزانِ تنکی داده‌ها در سطح دونگاشتی در پیکرهٔ معیار نسبتاً بالاست، برای این منظور، از اطلاعات دونگاشتیِ واژه‌ها که از پیکرهٔ هدف استخراج شده‌است استفاده می‌شود. همان‌طور که قبلاً گفته شد، مشکل اصلی پیکرهٔ هدف این است که کمتر از پیکرهٔ معیار قابل اعتماد است. برای کاهش تأثیر منفیِ استفاده از این پیکره، هنگام بررسیِ وجود دونگاشت در اطلاعاتِ دونگاشتیِ واژه‌ها که از پیکرهٔ هدف استخراج شده‌است، باید بسامد دونگاشتِ مورد جستجو بیش‌از ۵ باشد. درصورتِ غیبتِ دونگاشتِ مورد نظر یا بسامد کمتر از ۵، تفکیک انجام‌شده مورد تأیید قرار نخواهد گرفت و زنجیرهٔ ورودی بدون تغییر در خروجی نوشته می‌شود.

برخورد با واژه‌های ناشناخته امری اجتناب‌ناپذیر در پردازش خودکار زبان است. نحوهٔ رفتار با واژه‌های ناشناخته همانند واژه‌های چندواحدی است. بنابراین واژه‌های ناشناخته نیز تاحد امکان تفکیک می‌شود. چنانچه الگوریتم تفکیک‌سازی نتواند اطلاعات دونگاشتیِ مربوط به تفکیک واژه‌های ناشناخته را به‌دست آورد، از تفکیک آن خودداری می‌کند و واژهٔ ناشناخته به همان صورت اولیه، دست‌نخورده باقی می‌ماند.

1. data sparsity

۵-۱-۵- رفع مشکل تنک‌بودن داده

۵-۱-۵-۱- تنوع نگارشی واژه‌ها

یکی از عمده‌ترین مشکلاتی که در داده‌های فارسی وجود دارد، وجود تنوع در نحوهٔ نگارش واژه‌هاست که در بخش ۲-۲ به‌عنوان یکی‌از مشکلات و چالش‌های پردازش خط فارسی توضیح داده شد. البته این به این مفهوم نیست که همهٔ واژه‌ها دارای تنوع نگارشی هستند. حتی دستور خط فرهنگستان نیز این تنوع را تاحدی پذیرفته است، مانند واژهٔ «جرأت» که به‌صورت «جرئت» نیز می‌تواند نوشته شود. علاوه‌بر همزه، فاصلهٔ بین تکواژهای یک واژه نیز سبب ایجاد تنوع می‌شود، مانند: «میگوید» که بدون درج فاصله نوشته شده‌است، «می‌گوید» که با درج فاصلهٔ مجازی نوشته شده‌است، و «می گوید» که با درج فاصلهٔ کامل نوشته شده‌است.

دلیلِ درنظرگرفتنِ تنوع نگارشی این است که ممکن است داده‌ای که در پیکرهٔ معیار وجود دارد، به‌صورت دیگری در پیکرهٔ هدف نگارش شده باشد. بنابراین محدودکردنِ مدل به دادهٔ معیار سبب تنک‌شدن می‌شود و مدل جامعی از زبان به دست نمی‌آید. با درنظرگرفتنِ تنوع نگارشی می‌توان تاحدی از این مشکل کاست.

در این پژوهش، برای یافتنِ خودکارِ تنوعِ نگارشیِ واژه‌ها، از مدل معرفی‌شده توسط قیومی و همکاران (۱۳۹۴) استفاده می‌شود. مدل معرفی‌شدهٔ آنها، شکل توسعه‌یافتهٔ «فاصلهٔ لونشتاین»[1] (لونشتاین[2]، ۱۹۹۶) است که برای یافتنِ تنوع نگارشیِ واژه‌ها به‌کارمی‌رود. در تصویر ۱، شبه‌کد الگوریتمِ معرفی‌شده توسط قیومی و همکاران (۱۳۹۴) ارائه شده‌است. در الگوریتم سه حالت جایگزینی و حذف، درج و انطباق مورد توجه است و امکان وزن‌دهی به هریک از حالات وجود دارد.

ورودی: دو واژه (W_1, W_2)

محاسبهٔ فاصلهٔ لونشتاین:

- محاسبهٔ طول واژه‌های W_2 و W_1: N و M
- ساخت ماتریس دوبعدی $(N+1 \times M+1)$

1 Levenshtein distance
2. V. I. Levenshtein

- پرکردن سطر صفر ماتریس با نمایهٔ ستون مرتبط با آن
- پرکردن سطر صفر ماتریس با نمایهٔ سطر مرتبط با آن

شروع حلقه

- شروع از سطر یک و ستون یک تا سطر N و ستون M به‌صورت سطربه‌سطر
- محاسبهٔ امتیاز سلول [i,j] براساس حالات ممکن
- درج در واژهٔ اول:

اگر درج «ء»، «ئ»، یا فاصلهٔ مجازی، یا تنوین «ً»: [i,j]=[i1,j]+0.1

درغیراین‌صورت: [i,j]=[i-1,j]+1

- درج در واژهٔ دوم:

اگر درج «ء»، «ئ»، یا فاصلهٔ مجازی، یا تنوین «ً»: [i,j]=[i,j-1]+ 0.1

درغیراین‌صورت: [i,j]=[i,j-1]+1

- انطباق در دو واژه: [i,j]=[i-1,j-1]
- جایگزینی یک حرف در دو واژه:

اگر جایگزینی «آ» و «ا»: [i,j]=[i-1,j-1]+ 0.1

اگر جایگزینی «ا» و «أ»: [i,j]=[i-1,j-1]+ 0.1

اگر جایگزینی «أ» و «ئ»: [i,j]=[i-1,j-1]+0.1

اگر جایگزینی «ا» و «إ»: [i,j]=[i-1,j-1]+ 0.1

اگر جایگزینی «و» و «ؤ»: [i,j]=[i-1,j-1]+ 0.1

اگر جایگزینی «و» و «ئ»: [i,j]=[i-1,j-1]+0.1

اگر جایگزینی «ی» و «ئ»: [i,j]=[i-1,j-1]+0.1

اگر جایگزینی شناسه «د» و «ه» در آخر فعل: [i,j]=[i-1,j-1]+ 0.1

اگر جایگزینی «ا» و «و» در واژه، قبل از حروف «م» یا «ن»:

[i,j]=[i-1,j-1]+ 0.1

درغیراین‌صورت: [i,j]=[i-1,j-1]+1

- انتخاب یکی از سه حالت با کمترین امتیاز

پایان حلقه

خروجی: یک جفت واژه به‌همراه امتیاز محاسبه‌شده فاصلهٔ لونشتاین در سلول [N,M]

تصویر ۱- الگوریتم معرفی‌شدهٔ قیومی و همکاران (۱۳۹۴) برای یافتن تنوع نگارشی

۵-۱-۵-۲- استفاده از گنجینهٔ واژه[1]

همان‌طور که در بخش ۵-۱-۲ مطرح شد، در الگوریتمِ پیشنهادی، درصورت نبودِ واژهٔ ورودی در واژگان معیار، آن واژه به‌عنوان واژهٔ ناشناخته تلقی و تفکیک می‌شود. درنتیجه، فراهم‌کردن فهرستی خارج از واژه‌های معیار اهمیت به‌سزایی در کیفیت این روش پیشنهادی دارد. همان‌طور که می‌دانیم، گستردگیِ زبان در حوزه‌ها، ژانرها و سیاق کلام مختلف به‌گونه‌ای است که حتی با تهیهٔ یک پیکرهٔ بزرگ نمی‌توان به‌طورکامل مشکل تنک‌بودنِ داده را رفع کرد. یکی دیگر از روش‌های کاهشِ این مشکل، استفاده از گنجینه‌های واژه، شامل فهرستی از واژه‌های خاص مانند اسامی افراد، کشورها، شهرها، گل‌ها، کوه‌ها، دریاها، رودها و رودخانه‌ها و غیره، است. برای تهیه این فهرست از مجموعهٔ مستندات موجود در ویکی‌پدیای فارسی[2] استفاده شده‌است.

واژه‌های مصوب فرهنگستان و همچنین فهرستی از واژه‌های غیرفارسی که با خط فارسی نگارش شده‌اند، مانند دانلود و پرینت، به گنجینهٔ واژه اضافه شده و می‌توانند در کاهش مشکل تنک‌بودنِ داده مؤثر باشند.

۵-۱-۵-۳ خودآموزی از دادهٔ هدف

وجود تفاوت در سیاق کلامِ دو پیکرهٔ معیار و هدف، مشکلِ تنک‌بودنِ داده را پررنگ‌تر می‌کند. ترکیب اطلاعات آماریِ این دو پیکره (هم تک‌نگاشتی و هم دونگاشتی)، این مشکل را تاحدی تعدیل می‌بخشد. برای آمیختن اطلاعات، یک حد مرزی تعریف شده‌است تا بتوان تاحدودی اطلاعات غلطِ پیکرهٔ هدف را فیلتر کرد و از ورود این اطلاعات به داده‌های استخراج‌شدهٔ پیکرهٔ معیار جلوگیری کرد. این مرز، بسامد واژه‌های تک‌نگاشتی و دونگاشتی است.

۵-۱-۵-۴- استفاده از وندهای تصریفی و اشتقاقی

واژه‌سازی فرایندی زایاست که می‌تواند با اضافه‌شدنِ پیشوندها و پسوندهای تصریفی و اشتقاقی محقق شود. این ویژگیِ زاییِ زبان، سببِ افزایشِ مشکلِ تنک‌بودنِ داده می‌شود. برای کاهش

1. gazetteer
2. https://dumps.wikimedia.org/fawiki/

این مشکل باید از تحلیلگرِ صرفی استفاده شود. بر همین اساس، فهرستی از وندهای تصریفی و اشتقاقی تهیه شده‌است تا کار واژه‌سازی را انجام دهد. ویژگیِ استفاده از تحلیلگرِ صرفی این است که چنانچه صورت‌واژه‌ای از پیکرهٔ هدف در پیکرهٔ معیار یافت نشد ولی صورتِ دیگری از واژهٔ مورد نظر که محصول وندافزایی است در پیکرهٔ معیار یافت شود، الگوریتم با کمک این روش بتواند مشکل «خارج از واژگان»[1] را بکاهد.

۵-۲ - الگوریتمِ مدلِ معرفی‌شده

ساختار الگوریتمِ مرحلهٔ اولِ مدل معرفی‌شده برای تفکیک واژه‌های چندواحدی در تصویر ۲ آمده‌است. همان‌گونه که در این تصویر مشاهده می‌شود، ابتدا نیاز است اطلاعات n-نگاشتِ موردنظر از پیکرهٔ معیار و هدف استخراج شود. خوانش پیکرهٔ هدف به‌صورت جمله‌به‌جمله است. سپس این زنجیرهٔ ورودی با درنظرگرفتن «فاصلهٔ کامل»[2] به‌عنوان مرز واژه، به عناصر تشکیل‌دهندهٔ آن تقطیع می‌شود. هریک از این عناصر، بدون درنظرگرفتن این که آن واژه واقعی است یا خیر، به‌عنوان یک واژه تلقی می‌شود. هریک از واژه‌های زنجیرهٔ ورودی، به‌صورت حرف‌به‌حرف بررسی می‌شود. لازم به ذکر است که طول این واژه نباید زنجیره‌ای بیش از ۱۰۰ حرف باشد. یافتن انواع واژه‌های صحیحِ ممکن از این زنجیره و تأیید تقطیع آنها همگی در حافظه موقت رایانه انجام می‌پذیرد؛ لذا طولانی‌شدنِ این زنجیره، الگوریتم را با مشکل کمبود حافظه و افزایش زمان پردازش داده مواجه می‌کند که برای جلوگیری از مواجه‌شدن با این مشکل، این نوع زنجیره‌های طولانی بدون بررسی، به‌صورت دست‌نخورده در فایل خروجی نوشته می‌شود. سایر زنجیره‌های با طول کمتر از ۱۰۰ حرف باید از نظر تفکیک‌پذیری مورد بررسی قرار گیرند. بنابراین، واژه‌های این زنجیره با واژه‌های تک‌نگاشتِ استخراج‌شده از پیکرهٔ معیار مقایسه می‌شود. چنانچه واژه‌ای از پیکرهٔ هدف با واژه‌ای در فهرست تک‌نگاشت منطبق شود، آن واژه در فایل خروجی نوشته می‌شود.

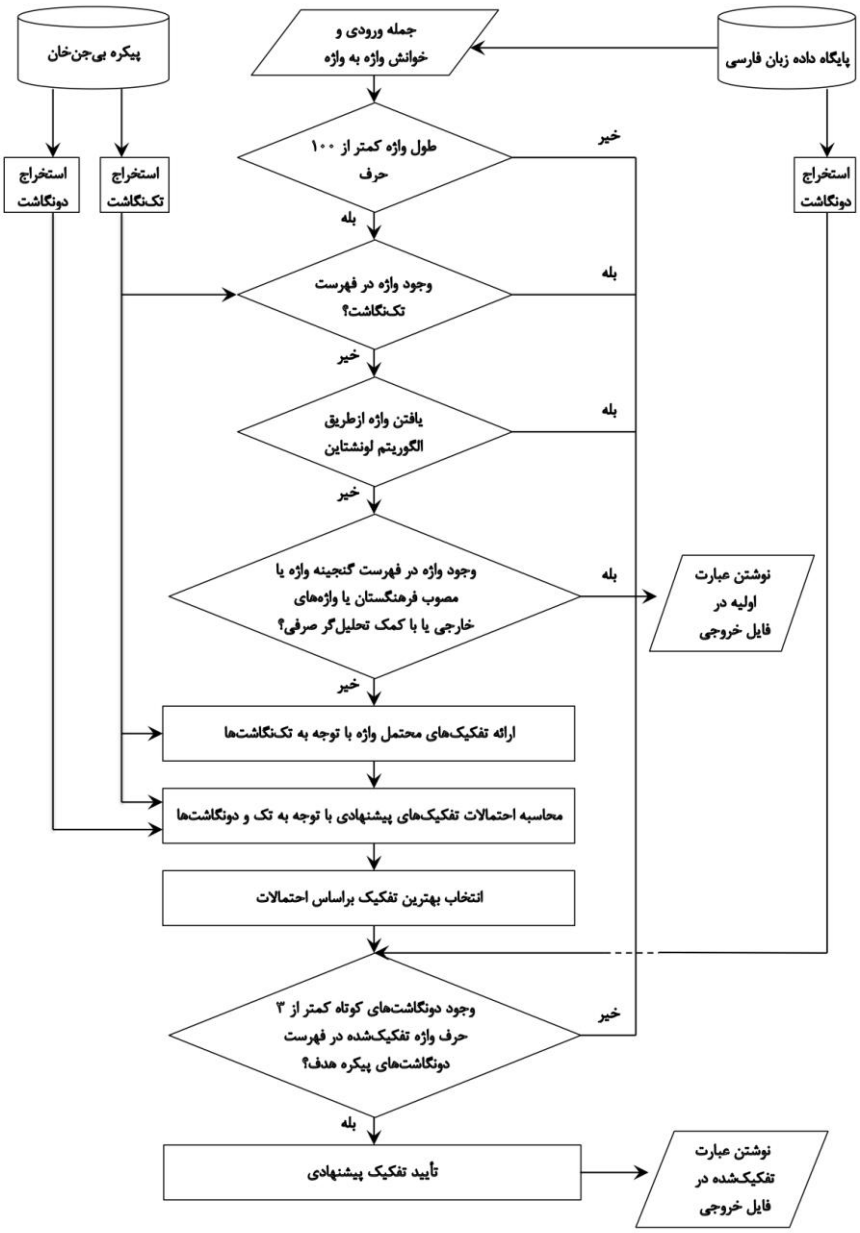
1. out of vocabulary
2. white space

در این مرحله، با چند چالش روبرو می‌شویم. یکی از این چالش‌ها مربوط به خط فارسی است. شکل نوشتاریِ حروف «آ»، «ا»، «د»، «ذ»، «ر»، «ز»، «ژ»، و «و» در چسبندگی به عنصر بعدی تغییر نمی‌کند. این ویژگی به‌همراه عدم رعایت فاصله‌گذاری، ممکن است سبب ابهام در تشخیص واژه شود. برای مثال، واژه‌هایی چون «مادر» و «وبا» نمونه‌هایی از این نوع ابهام هستند. در این ابهام مشخص نیست که نگارش صحیح باید یک واژهٔ بسیط «مادر» باشد یا دو واژهٔ «ما» و «در». با نگاهی به پیکره دریافتیم که حجم این موارد بسیار اندک است و در این مرحله از تهیهٔ مدل، رفع اینگونه ابهام‌ها در اولویت قرار ندارد.

چالش بعدی این است که آیا نیافتن واژه‌ای از پیکرهٔ هدف در تک‌نگاشت به این مفهوم است که این واژه باید تفکیک شود؟ پاسخ منفی است، چراکه ممکن است دلیل نیافتنِ واژهٔ پیکرهٔ هدف در تک‌نگاشت، تنک‌بودنِ دادهٔ معیار باشد. برای رفع تنک‌بودنِ داده و تشخیص حداکثریِ واژهٔ پیکرهٔ هدف در تک‌نگاشت، در چندین مرحله واژهٔ پیکرهٔ هدف در تک‌نگاشتِ دادهٔ معیار چک می‌شود.

ممکن است یکی از دلایل عدم انطباقِ واژهٔ پیکرهٔ هدف در فهرستِ تک‌نگاشت‌ها، تنوع نگارشیِ واژه باشد. الگوریتمِ معرفی‌شده توسط قیومی و همکاران (۱۳۹۴) که در قسمت ۵-۱-۵ توضیح داده شد، برای تشخیصِ تنوعِ نگارشی استفاده می‌شود تا بین تنوع نگارشیِ واژه‌ها ارتباط ایجاد کند و الگوریتمِ معرفی‌شده در این مقاله بتواند این واژه را تشخیص دهد. برای مثال، «علاقه‌مند» می‌تواند با درج فاصلهٔ مجازی (نیم‌فاصله) نگارش شود. علاوه‌بر آن، این واژه می‌تواند با درج فاصلهٔ کامل یا بدون درج فاصله، به‌صورت «علاقه مند» و «علاقمند» نیز نوشته شود. الگوریتمِ تنوعِ نگارشی سبب می‌شود این سه واژه یک واژه تلقی شود و از تنک‌بودنِ داده‌ها بکاهد. البته در تنوع نگارشی فقط فاصلهٔ بین عناصر مطرح نیست. مواردی چون همزه در «جرئت» و «جرأت» یا جایگزینی حرف «آ» با «ا» در «قرآن» و «قران» نیز تشخیص داده می‌شود.

همچنین با افزودنِ فهرست واژه به مدل می‌توان تاحدودی مشکلِ تنک‌بودنِ داده را برطرف کرد. ازجمله فهرست‌های واژهٔ افزوده‌شده به مدل، واژه‌های مصوب فرهنگستان است. این واژه‌ها نسبتاً جدیدند و معمولاً متداول نیستند. بنابراین، با



تصویر ۲- ساختارِ الگوریتمِ مرحلهٔ اولِ مدلِ معرفی‌شده

آگاهی از این واژه‌ها می‌توان از تفکیک آنها جلوگیری کرد. فهرست واژهٔ دیگری که اضافه شده‌است، گنجینهٔ واژه، شامل فهرستی از اسامی افراد، شهرها، کشورها، گل‌ها، رودها، رودخانه‌ها، کوه‌ها و موارد اینچنینی است که برای تهیهٔ این فهرست، عناوین مستندات موجود در ویکی‌پدیای فارسی به‌عنوان گنجینهٔ واژه به فهرست‌های واژهٔ افزوده‌شده به مدل اضافه می‌شود. برای کاربردی‌ترشدنِ مدلِ معرفی‌شده تلاش کرده‌ایم این مدل به‌گونه‌ای تهیه شود که قابلیت کاربرد مجدد بر روی یک مجموعهٔ دادهٔ دیگر را داشته باشد. با نگاهی به منبع، داده‌های دیگری مانند وبلاگ، کاربرد واژه‌های لاتین با خط فارسی، مانند «دانلود» یا «آنلاین»، جلب توجه می‌کند. برای رفع مشکل تنک‌بودنِ این نوع واژه‌ها، فهرستی از واژه‌های غیرفارسیِ نگارش‌شده با خط فارسی تهیه شده‌است تا بتوان از تفکیک‌پذیری آنها جلوگیری کرد.

روش دیگر کاستن تنک‌بودن داده، استفاده از تحلیل‌گر واژه است. این تحلیل‌گر می‌کوشد با تحلیل ساختارِ واژه و درنظرگرفتنِ وندهای تصریفی و اشتقاقی، بین واژهٔ پیکرهٔ هدف و تک‌نگاشتِ پیکرهٔ معیار ارتباط برقرار کند. برای مثال، اگر واژهٔ پیکرهٔ هدف «فروشگاه» باشد و در فهرست تک‌نگاشت «فروشگاه‌هایی» وجود داشته باشد، با کمک تحلیل‌گر صرفی واژهٔ «فروشگاه‌هایی» از واژهٔ «فروشگاه» ساخته می‌شود و سپس انطباق صورت می‌گیرد. لازم به ذکر است که این تحلیل صرفی دوسویه است؛ به این معنی که اگر در پیکرهٔ هدف واژهٔ «فروشگاه‌هایی» و در فهرست تک‌نگاشت واژهٔ «فروشگاه» وجود داشته باشد، می‌توان با کمک تحلیل‌گر صرفی از واژهٔ «فروشگاه‌هایی» به واژهٔ «فروشگاه» رسید و آنها را بر هم منطبق کرد.

اگر واژهٔ پیکرهٔ هدف به روش‌های فوق در فهرست تک‌نگاشتِ پیکرهٔ معیار یافت نشد، آن واژه باید تفکیک شود. روش تفکیک در بخش ۵-۱-۲ توضیح داده شده‌است و در اینجا به ذکر مثال بسنده می‌کنیم. تصور کنید ورودی مدل، زنجیرهٔ «ویابهتراست» باشد. این زنجیره، با کمک روش‌های گفته‌شده در بالا برای رفع تنک‌بودنِ داده، در فهرست تک‌نگاشت یافت نشده است و باید تفکیک شود. بنابراین، از ابتدای زنجیره، حرف‌به‌حرف خوانده می‌شود. چنانچه بتوان از توالی حروف، واژه‌ای را در واژگان معیار یافت، آن توالی حروف به‌عنوان یک واژه به‌طور موقت از زنجیره جدا می‌شود و بقیهٔ زنجیره به‌صورت بازگشتی مجدداً برطبق همین روش بررسی می‌شود. درنهایت، تفکیک پیشنهادی «و یا بهتر است» ارائه داده می‌شود. از آنجا که تک‌تک

واژه‌های تفکیک‌شدهٔ پیشنهادی، در فهرست تک‌نگاشتِ دادهٔ معیار وجود دارد، پاسخ الگوریتم برای اعمال این تفکیک مثبت است.

گاهی به‌دلیل ابهام، ممکن است چندین تفکیک پیشنهاد شود، مانند «مادراوست» که درمورد تفکیک‌پذیری «مادر» به «ما» و «در» و عدم تفکیک‌پذیری آن به‌صورت واژهٔ بسیط «مادر» ابهام وجود دارد. در این موارد احتمال دونگاشتی‌بودنِ هریک از واژه‌های یافت‌شده محاسبه می‌شود. در مرحلهٔ بعد، تفکیکی که بالاترین عدد احتمالات برمبنای مدلِ زبانیِ آماری را دارد، به‌عنوان کاندید و بهترین تفکیک انتخاب می‌شود. تاکنون جواب مثبت از الگوریتم گرفته شده‌است، ولی به‌دلیلِ تنک‌بودنِ داده و محدودبودن فهرستِ تک‌نگاشت، ممکن است واژه‌ای به‌اشتباه تفکیک شده باشد. برای کاهش این نوع خطا، از دونگاشتِ پیکرهٔ هدف برای واژه‌های با طول بیش‌از ۳ حرف استفاده می‌شود. چنانچه پاسخ الگوریتم پس‌از مقایسه‌کردن با دونگاشتِ پیکرهٔ هدف منفی باشد، از تفکیک آن صرف‌نظر می‌شود و آن زنجیره بدون تغییر در فایل خروجی نوشته می‌شود. چنانچه پاسخ الگوریتم مثبت باشد، تفکیک تأیید می‌شود و در فایل خروجی نوشته می‌شود.

توضیحات بالا، مربوط به پردازشِ مرحلهٔ اول در مدلِ معرفی‌شده است. در این مرحله، فقط کار تفکیک انجام نمی‌پذیرد و امکان ترکیبِ وندهای تصریفی و اشتقاقی با پایه نیز وجود دارد. لازم به ذکر است درحال‌حاضر الگوریتم معرفی‌شده توانایی واحدسازی واحدهای چندواژه‌ای، مانند حرف اضافهٔ مرکب، حرف ربط مرکب، قید مرکب و غیره، را ندارد که برای افزایش کارایی این الگوریتم در پژوهش‌های آتی به این موضوع پرداخته خواهد شد.

پردازشِ مرحلهٔ دوم ساده‌تر است. این مرحله فقط بر روی فعل متمرکز است و خروجیِ مرحلهٔ اول، ورودیِ این مرحله است. در این مرحله، پیشوندهای استمراریِ «می» یا «نمی» و واژه‌بست‌های تصریفیِ «ـام، ـای، ـاست، ـایم، ـاید، ـاند» که از فعل منفک شده‌اند و نقش یک واژهٔ مستقل را بازی می‌کنند، به فعل ملحق می‌شوند تا یک واژه را بسازند، مانند: «می خورده ام» که به «می‌خورده‌ام» تبدیل می‌شود.

مرحلهٔ سومِ پردازش، تکرار مرحلهٔ اول است و ورودی آن، خروجیِ مرحلهٔ دوم است. دلیل تکرارِ مرحلهٔ اول این است که ممکن است در مرحلهٔ اول یک زنجیره به واحدهایی تقسیم شود

که بعضی از این واحدها باید با واحدهای دیگر ترکیب شوند و یک واحد را بسازند، مانند «دردارو سازی» که در مرحلهٔ اول به «در دارو سازی» تبدیل می‌شود. برای فراهم‌آوردنِ امکانِ ترکیبِ واحدهای منفصل‌شده، پردازش مرحلهٔ اول مجدداً تکرار می‌شود تا بعضی از واحدها به یکدیگر متصل شوند و یک واژه را بسازند. برای مثال در مرحله سوم، «در دارو سازی» به «در داروسازی» تبدیل می‌شود.

۶- آزمایش‌ها و ارزیابی

مدلِ زبانیِ معرفی‌شده در بخش ۵ با زبان برنامه‌نویسی جاوا پیاده‌سازی شده‌است. در این مدل، ابتدا نیاز است که اطلاعاتِ آماریِ تک‌نگاشتی و دونگاشتی از دو پیکرهٔ معیار و هدف استخراج شود. با اجرای الگوریتمِ پیاده‌سازی‌شده، پیکرهٔ هدف به‌عنوان دادهٔ ورودی به نرم‌افزار داده می‌شود تا مشکل چندواژگی و واحدسازی را کاهش دهد.

برای ارزیابیِ کارایی الگوریتم، به دادهٔ آزمون نیاز است. متأسفانه تاکنون برای زبان فارسی مجموعهٔ داده‌ای که حاوی دادهٔ بدون واحدسازی معیار و معادلِ واحدسازیِ معیار آن باشد فراهم نشده‌است. علاوه‌بر این کاستی، باتوجه‌به این که الگوریتم پژوهش حاضر سه مرحله‌ای است و می‌خواهیم بر اساس هر مرحله، کارایی الگوریتم را محاسبه کنیم به دادهٔ معیار برای هر سه مرحله نیاز داریم. این داده با ویژگی‌هایی که مد نظر است به‌صورت آماده وجود ندارد. به همین دلیل برای ارزیابی الگوریتم معرفی شده نیاز است تا خروجی این الگوریتم به‌صورت دستی مورد ارزیابی قرار گیرد.

برای این منظور، به‌شکل تصادفی ۲۰۰ جمله از تمام داده‌های تحلیل‌شده انتخاب شده‌است و باتوجه‌به معیارهای ارزیابی که در ادامه توضیح داده می‌شود، کار ارزیابی به‌صورت دستی برای هریک از سه مرحلهٔ پردازش انجام می‌پذیرد. ممکن است تعداد ۲۰۰ جملهٔ آزمون برای ارزیابی کم به‌نظر بیاید و کافی نباشد و بتوان با داشتن حجم بیشتری دادهٔ آزمون نتایج واقع‌گرایانه‌تری به‌دست آورد. ازآنجاکه تهیهٔ دادهٔ معیار به‌صورت دستی کاری زمان‌بر و طاقت‌فرساست، در این پژوهش این سختی کار دوچندان است. دلیل آن این است که باتوجه‌به چندمرحله‌ای‌بودن الگوریتم معرفی‌شده، بررسی جداگانهٔ خروجی هر مرحله مورد نیاز است تا ضمن بررسی کارایی الگوریتم در هر مرحله، امکان مقایسهٔ بین مراحل مختلف وجود داشته باشد. بر این اساس،

۲۰۰ جمله مذکور، سه بار به‌طور مجزا مورد بررسی قرار گرفته‌است. برای ارزیابیِ عملکردِ مدلِ معرفی‌شده می‌توان از ماتریسِ درهم‌ریختگی[1] که در جدول ۱ آمده است استفاده کرد.

جدول ۱- ماتریس درهم‌ریختگی

		دادهٔ پردازش‌شده	
		صحیح	**غلط**
دادهٔ ورودی	**صحیح**	صحیح ← صحیح	صحیح ← غلط
	غلط	غلط ← صحیح	غلط ← غلط

منظور از «دادهٔ ورودی» در جدول ۱ داده‌ای است که با احتسابِ فاصلهٔ کامل به‌عنوان مرزنمای واژه بتوان واژه‌های مستقل یا واژه‌های چندواحدی را یافت که باید توسط الگوریتمِ معرفی‌شده به واژه‌های مستقل صحیح تبدیل شوند. این تبدیل ممکن است به‌صورت صحیح یا غلط انجام پذیرد. «دادهٔ پردازش‌شده» مربوط به خروجی و عملکرد الگوریتم بر روی داده است که نتیجهٔ به‌دست‌آمده از کار پردازشی الگوریتم معرفی‌شده می‌تواند صحیح یا غلط باشد. از درهم‌ریختگیِ این دو، چهار وضعیت را فراهم می‌آورد:

(الف) **صحیح ← صحیح:** در این وضعیت، یک واژهٔ صحیح در دادهٔ ورودی بدون تغییر و به همان صورت در خروجیِ الگوریتم دیده می‌شود.

(ب) **غلط ← صحیح:** در این وضعیت، یک واژهٔ غلط در دادهٔ ورودی پس از اعمالِ الگوریتمِ معرفی‌شده بر روی آن و انجامِ کارِ پردازشی، به یک واژهٔ صحیح تبدیل می‌شود.

(ج) **صحیح ← غلط:** در این وضعیت، یک واژهٔ صحیح در دادهٔ ورودی پس از اعمال الگوریتمِ معرفی‌شده بر روی آن و انجام کارِ پردازشی به یک واژهٔ غلط تبدیل می‌شود.

(د) **غلط ← غلط:** در این وضعیت، یک واژهٔ غلط در دادهٔ ورودی بدون تغییر و به همان صورتِ غلط در خروجیِ دیده می‌شود و الگوریتمِ معرفی‌شده هیچ‌گونه کارِ پردازشی بر روی آن انجام نمی‌دهد.

1. confusion matrix

با درنظرداشتنِ این چهار وضعیت، می‌توان فرمول‌های (۱) تا (۳) را استخراج کرد. فرمول (۱) برای محاسبهٔ میزان تصحیح خطاهای نوشتاری در معیارسازیِ متن به‌کار می‌رود. در این فرمول، نسبتِ غلط‌های تصحیح‌شده توسطِ الگوریتم به تعدادِ کلِ غلط‌هایِ ورودی، اعم از این که تصحیح شده باشد یا خیر، سنجیده می‌شود. بالابودن این عدد بیانگرِ توانایی الگوریتم برای تصحیحِ غلطِ دادهٔ ورودی است. فاصلهٔ میزانِ درصدِ تصحیحِ خطای نوشتاری تا ۱۰۰درصد که همان دادهٔ معیار است، بیانگر میزان کاری است که توسط الگوریتمِ معرفی‌شده انجام نمی‌گیرد و برای رسیدن به دادهٔ معیار نیاز به کارِ دستی است.

(۱)

$$\text{تصحیح خطای نوشتاری} = \frac{\text{غلط} \leftarrow \text{صحیح}}{\text{غلط} \leftarrow \text{غلط} \ + \ \text{غلط} \leftarrow \text{صحیح}}$$

فرمول (۲) برای محاسبهٔ میزانِ خطاهای ایجادشده توسطِ الگوریتمِ معرفی‌شده در مواردی که ابهام وجود دارد به‌کار می‌رود. در این فرمول، نسبت صحیح‌هایی که غلط شده‌اند به تمام صحیح‌ها، اعم از این که صحیح‌ها غلط شده باشند یا بدون تغییر باقی مانده باشند، سنجیده می‌شود. پایین‌بودنِ این عدد بیانگرِ توانایی الگوریتمِ معرفی‌شده در ابهام‌زدایی و هوشمندیِ آن در عدم اعمالِ هرگونه تغییری است.

(۲)

$$\text{ایجاد خطای نوشتاری} = \frac{\text{صحیح} \leftarrow \text{غلط}}{\text{صحیح} \leftarrow \text{صحیح} \ + \ \text{صحیح} \leftarrow \text{غلط}}$$

فرمول (۳) میزان دقتِ الگوریتمِ معرفی‌شده را محاسبه می‌کند. در این فرمول نسبتِ تمام صحیح‌ها در خروجیِ الگوریتم به تمام حالات در داده‌ها سنجیده می‌شود. بالابودنِ این عدد بیانگرِ میزانِ دقت در دادهٔ خروجی است. فاصلهٔ میزانِ درصدِ دقت تا ۱۰۰درصد بیانگر میزانِ غلط در دادهٔ خروجی است و نیاز آن به اصلاح دستی برای دستیابی به دقت ۱۰۰درصد است.

(۳)

$$\text{دقت} = \frac{\text{غلط} \leftarrow \text{صحیح} \ + \ \text{صحیح} \leftarrow \text{صحیح}}{\text{صحیح} \leftarrow \text{صحیح} \ + \ \text{غلط} \leftarrow \text{صحیح} \ + \ \text{صحیح} \leftarrow \text{غلط} \ + \ \text{غلط} \leftarrow \text{غلط}}$$

در جدول ۲، تعداد واژه‌های ۲۰۰ جملهٔ انتخاب‌شدهٔ آزمون به‌طور تصادفی، برای دادهٔ ورودی و خروجی آمده‌است. کاهش تعدادِ واژه‌ها در جملاتِ خروجیِ نهایی بیانگر این است که واحدهای چندواژه‌ایِ فراوان در دادهٔ ورودی وجود دارد که با رفع این مشکلات می‌توان به دادهٔ معیار و آمار دقیق‌تری از واژه‌های موجود در آن دست یافت.

جدول ۲- نتایج به‌دست‌آمده از داده‌های ورودی و خروجی نهایی

تعداد جملات	تعداد واژه‌های دادهٔ ورودی	تعداد واژه‌های دادهٔ خروجی نهایی	طول متوسط جملات دادهٔ خروجی نهایی
۲۰۰	۵۷۰۹	۵۴۰۳	۲۷

همان‌طور که در قسمت ۵-۲ توضیح داده شد، الگوریتمِ معرفی‌شده سه مرحلهٔ پردازش دارد که در ادامه، ارزیابی هریک از مراحل توضیح داده خواهد شد. مرحلهٔ اول بیشترین و مهم‌ترین نقش را در الگوریتمِ معرفی‌شده بازی می‌کند. در این مرحله تفکیک چندواحدی‌ها به واژه‌های مستقل یا ترکیب وندها به پایه‌ها صورت می‌پذیرد. مطابق با جدول۱، ارزیابیِ این مرحله در جدول ۳ نمایش داده شده‌است.

جدول ۳- ارزیابیِ مرحلهٔ اولِ الگوریتمِ معرفی‌شده

		دادهٔ پردازش‌شده	
		صحیح	غلط
دادهٔ ورودی	صحیح	۵۴۰۷	۱۰
	غلط	۱۷۹	۳۱

خروجیِ اجرایِ مرحلهٔ اولِ الگوریتم سبب می‌شود تعداد واژه‌های پیکره از ۵۷۰۹ واژه به ۵۶۲۷ واژه کاهش یابد. نتایج به‌دست‌آمده از اجرایِ مرحلهٔ اولِ الگوریتم در جدول ۳ گزارش شده‌است که به کمک آنها می‌توان میزان تغییرات را نسبت‌به تمام واژه‌های این مرحله سنجید. در این

مرحله، ۵۴۰۷ واژه از مجموع ۵۶۲۷ واژهٔ دادهٔ ورودی (۹۶/۰۹ درصد) با دادهٔ خروجی منطبق است و الگوریتمِ معرفی‌شده هیچ‌گونه تغییری روی این موارد اعمال نکرده‌است. ۱۷۹ واژه از مجموع ۵۶۲۷ واژهٔ ورودی (۳/۱۸ درصد) غلط بوده و به‌طور صحیح توسط ماشین تغییر کرده‌است. ۳۱ واژهٔ صحیح از مجموع ۵۶۲۷ واژهٔ ورودی (۰/۱۸ درصد) توسط ماشین غلط شده‌است؛ و ۱۰ واژه از مجموع ۵۶۲۷ واژهٔ ورودی که غلط است (۰/۵۵ درصد) بدون اعمال هرگونه تغییری توسط الگوریتم معرفی‌شده، در دادهٔ خروجی غلط باقی مانده‌است.

باتوجه‌به اطلاعات موجود در جدول ۳، می‌توان کارایی مدل در مرحلهٔ اول را طبق فرمول‌های (۱) تا (۳) محاسبه و نتایج را در جدول ۴ مشاهده کرد. براساسِ نتایجِ به‌دست‌آمده، از میان داده‌های غلطی که به‌عنوان دادهٔ ورودی به مدل داده شده‌است، ۸۵/۲۴ درصد آن به‌طور صحیح توسط ماشین تغییر کرده‌است. همچنین، از میان داده‌های صحیحی که به‌عنوان دادهٔ ورودی به مدل داده شده‌است، ۰/۱۹ درصد آن به‌اشتباه توسط ماشین تغییر کرده‌است. این عدد کمتر از یک درصد بوده و بسیار ناچیز است. دقت این الگوریتم که در اصل، خروجیِ دادهٔ صحیح است ۹۹/۲۷ درصد است. مابقی ۰/۷۳ درصد مربوط به غلط‌هایی است که یا توسط ماشین ایجاد شده‌است، یا توسط الگوریتم تصحیح نشده‌است.

جدول ۴- درصدِ کاراییِ الگوریتمِ معرفی‌شده در مرحلهٔ اول

تصحیح خطای نوشتاری	ایجاد خطای نوشتاری	دقت
۸۵/۲۴	۰/۱۹	۹۹/۲۷

پردازش داده در مرحلهٔ دوم بسیار ساده است. در این مرحله، عناصرِ متعلق به صورت‌های صرفیِ فعل، با یکدیگر ترکیب می‌شوند. جدول ۵، ارزیابیِ این مرحله را نمایش می‌دهد. خروجیِ اجرایِ مرحلهٔ دومِ الگوریتم سبب می‌شود تعداد واژه‌های پیکره از ۵۶۲۷ واژه به ۵۵۱۳ واژه کاهش یابد. نتایج به‌دست‌آمده از اجرایِ مرحلهٔ دومِ الگوریتم در جدول ۵ گزارش شده‌است که می‌توان به کمک آنها میزان تغییرات را نسبت‌به تمام واژه‌های این مرحله سنجید. در مرحلهٔ دوم، ۵۳۹۴ واژه از ۵۵۱۳ واژهٔ دادهٔ ورودی (۹۷/۸۴ درصد) با دادهٔ خروجی منطبق است و تغییری روی این داده اعمال نشده‌است. ۱۱۸ واژهٔ غلط از ۵۵۱۳ واژهٔ ورودی (۲/۱۴

درصد) توسط الگوریتم به‌طور صحیح تغییر کرده‌است. ۱ واژهٔ صحیح از ۵۵۱۳ واژهٔ ورودی (۰/۰۲ درصد) توسط الگوریتمِ معرفی‌شده غلط شده‌است؛ و مدل هیچ دادهٔ ورودی غلطی را بدون تغییر باقی نگذاشته‌است.

جدول ۵- ارزیابی مرحلهٔ دومِ الگوریتمِ معرفی‌شده

		دادهٔ پردازش‌شده	
		صحیح	غلط
دادهٔ ورودی	صحیح	۵۳۹۴	۱
	غلط	۱۱۸	۰

باتوجه‌به اطلاعات موجود در جدول ۵، می‌توان کاراییِ الگوریتم در مرحلهٔ دوم را محاسبه و نتیجه را در جدول ۶ مشاهده کرد. براساس نتایجِ به‌دست‌آمده، تمام داده‌های غلطی که به‌عنوان دادهٔ ورودی به الگوریتم داده می‌شود، به‌طور صحیح توسط ماشین تغییر کرده‌اند. از میان داده‌های صحیحی که به‌عنوان دادهٔ ورودی به مدل داده شده‌است، ۰/۰۲ درصد آن به‌اشتباه توسط ماشین تغییر کرده‌است که بسیار ناچیز است. دقت این الگوریتم برای دادهٔ خروجی صحیح ۹۹/۹۸ درصد است.

جدول ۶- درصدِ کاراییِ الگوریتمِ معرفی‌شده در مرحلهٔ دوم

تصحیح خطای نوشتاری	ایجاد خطای نوشتاری	دقت
۱۰۰	۰/۰۲	۹۹/۹۸

مرحلهٔ سومِ الگوریتمِ معرفی‌شده، درحقیقت تکرارِ مرحلهٔ اول است. هدف از این تکرار این است که چنانچه زنجیرهٔ متصلی از هم منفک شده و لازم باشد بعضی‌از عناصرِ این زنجیرهٔ منفک‌شده به عناصر دیگر متصل شود، امکان اتصال این عناصر فراهم آید. برای واضح‌تر شدن این هدف مثالی زده می‌شود. زنجیرهٔ «وعلاقه مند» را در نظر بگیرید. در مرحلهٔ یک، زنجیرهٔ «وعلاقه» از هم منفک شده و به «و» و «علاقه» تبدیل می‌شود. در این مرحله وند «مند» به پایهٔ خود

متصل نمی‌شود؛ بلکه در مرحلهٔ سوم، دو عنصر «علاقه» و «مند» به یکدیگر متصل می‌شوند. خروجیِ اجرایِ مرحلهٔ سومِ الگوریتم سبب می‌شود تعداد واژه‌های پیکره از ۵۵۱۳ واژه به ۵۵۰۶ واژه کاهش یابد. نتایج به‌دست‌آمده از اجرایِ مرحلهٔ سومِ الگوریتم در جدول ۷ گزارش شده‌است که می‌توان به کمک آنها میزان تغییرات نسبت‌به تمام واژه‌های این مرحله را سنجید.

در مرحلهٔ سوم، ۵۴۷۰ واژه از مجموع ۵۵۰۶ واژهٔ دادهٔ ورودی (۹۹/۳۵ درصد) با دادهٔ خروجی منطبق است و تغییری روی این داده اعمال نشده‌است. بنابراین در این مرحله روی حجمِ بسیار کمِ باقیمانده (کمتر از یک درصد)، کار پردازشی انجام می‌شود. ۷ واژه از مجموع ۵۵۰۶ واژهٔ دادهٔ ورودی (۰/۱۳ درصد) غلط بوده و توسط الگوریتم به‌طور صحیح تغییر کرده‌است. ۵ واژهٔ صحیح از مجموع ۵۵۰۶ واژهٔ دادهٔ ورودی (۰/۰۹ درصد) توسط الگوریتم غلط شده‌است؛ و روی ۲۴ واژهٔ غلط از مجموع ۵۵۰۶ واژهٔ دادهٔ ورودی (۰/۴۴ درصد) تغییری اعمال نشده‌است.

جدول ۷- ارزیابی مرحلهٔ سومِ الگوریتمِ معرفی‌شده

		داده پردازش‌شده	
		صحیح	غلط
دادهٔ ورودی	صحیح	۵۴۷۰	۵
	غلط	۷	۲۴

براساس جدول ۷، نتیجهٔ ارزیابیِ مرحلهٔ سومِ الگوریتمِ معرفی‌شده در جدول ۸ آمده است. باتوجه‌به نتایجِ به‌دست‌آمده می‌توان مشاهده کرد که ۲۲/۵۸ درصد خطاهای نوشتاری توسط ماشین تصحیح شده و ۰/۰۹ درصد خطاها توسط الگوریتم ایجاد شده‌است. دقت الگوریتم در مرحلهٔ سوم ۹۹/۴۷ درصد است.

جدول ۸- میزان درصدِ کاراییِ الگوریتمِ معرفی‌شده در مرحلهٔ سوم

تصحیح خطای نوشتاری	ایجاد خطای نوشتاری	دقت
۲۲/۵۸	۰/۰۹	۹۹/۴۷

پس‌از بررسی جملات خروجی در مرحلهٔ سوم، اشکالات باقیمانده به‌صورت دستی تصحیح شد و جمله معیار به‌دست آمد و منجر شد تعداد واژه‌های پیکره از ۵۵۰۶ واژه به ۵۴۰۳ واژه کاهش یابد. در جدول ۹، میزان تغییرات واژه‌های این مرحله نسبت‌به دادهٔ ورودی ارزیابی شده‌است. در تهیهٔ این داده‌ها، ۴۹۸۰ واژه از مجموع ۵۴۰۳ واژهٔ دادهٔ ورودی (۹۲/۱۷ درصد) با دادهٔ خروجی منطبق است. ۳۰۴ واژهٔ غلط از مجموع ۵۴۰۳ واژهٔ دادهٔ ورودی (۵/۶۳ درصد) صحیح شده‌است. ۱ واژهٔ صحیح از مجموع ۵۴۰۳ واژهٔ دادهٔ ورودی (۰/۰۲ درصد) غلط شده‌است؛ و ۱۱۸ واژهٔ غلط از مجموع ۵۴۰۳ واژهٔ دادهٔ ورودی (۲/۱۸ درصد) بدون تغییر باقی مانده‌است.

جدول ۹- ارزیابی تهیهٔ دادهٔ معیار نسبت به دادهٔ ورودی اولیه

		دادهٔ پردازش‌شده	
		صحیح	غلط
دادهٔ ورودی	صحیح	۴۹۸۰	۱
	غلط	۳۰۴	۱۱۸

در جدول ۱۰، درصدِ کاراییِ کلیِ الگوریتمِ معرفی‌شده گزارش شده‌است. براساسِ نتایجِ به‌دست‌آمده، از میان داده‌های غلطی که به‌عنوان دادهٔ ورودی تصحیح شده‌است، ۷۲/۰۴ درصد آن صحیح بوده‌است. همچنین، از میان داده‌های صحیحی که به‌عنوان دادهٔ ورودی به مدل داده شده‌است، ۰/۰۲ درصد به‌اشتباه تغییر کرده‌است. این عدد کمتر از یک درصد و بسیار ناچیز است. دقت کل این سامانه، ۹۷/۸۰ درصد است. مابقی مربوط به غلط‌هایی است که توسط الگوریتم ایجاد شده‌است یا غلط‌های موجود در داده‌هاست که توسط الگوریتم تصحیح نشده‌است.

جدول ۱۰- میزان درصد کارایی برای تهیهٔ دادهٔ معیار

تصحیح خطای نوشتاری	ایجاد خطای نوشتاری	دقت
۷۲/۰۴	۰/۰۲	۹۷/۸۰

۷- نتیجه‌گیری

آنچه در این مقاله مطرح شد، معرفی روشی مبتنی‌بر یک مدل زبانیِ آماری برای کاربرد در واحدسازیِ پیکرهٔ فارسی بود. اساساً متنِ نگاشته‌شدهٔ فارسی با دو مشکل ساده ولی فراگیر مواجه است. مشکل اول اتصالِ یک واژه به واژه‌های بعدی و تشکیل واژه‌های چندواحدی است؛ و مشکل بعدی واحدهای چندواژه‌ای است که از جداشدگی واژه‌هایی به دست می‌آیند که با هم یک واحد واژگانی را تشکیل می‌دهند.

اجرای الگوریتم این مدل، سه‌مرحله‌ای بود. در مرحلهٔ اول، واژه‌های چندواحدی از هم منفک و واحدهای چندواژه‌ایِ حاصل از وندافزایی به هم متصل می‌شود. برای رسیدن به هدف مرحلهٔ اول، یک الگوریتم پایه معرفی شد. سپس باتوجه‌به چالش‌های پیش‌آمده، این الگوریتم بهبود داده شد. در مرحلهٔ دوم، از روش انطباق برای بررسی چندواژگی در فعل استفاده شد. مرحلهٔ سوم در حقیقت تکرار مرحلهٔ اول است که بر روی موارد جاافتاده اعمال می‌شود. در انتها از الگوریتمِ معرفی‌شده برای واحدسازیِ داده‌های زبانیِ پایگاه داده‌های زبان فارسی استفاده شد. براساس ۲۰۰ جملهٔ دادهٔ آزمون، نتیجهٔ ۷۲/۰۴ درصدی در تصحیح خطای نگارشی دادهٔ ورودی با ۹۷/۸۰ درصد دقت و ۰/۰۲ درصد ایجاد خطای نگارشی توسط الگوریتم به‌دست آمد. این میزان دقت بیانگر این است که دادهٔ خروجیِ الگوریتمِ معرفی‌شده چقدر به دادهٔ معیار نزدیک است و چقدر به تلاش نیروی انسانی برای تکمیل واحدسازی داده‌ها نیاز است. برای رسیدن به این میزان دقت، ۷۲/۰۴ درصد از خطای نگارشی در متن به‌طور خودکار تصحیح شده‌است که البته می‌توان این انتظار را داشت که رفع مسائلی که سبب می‌شود الگوریتمِ معرفی‌شده نتواند به‌خوبی کار تصحیح را انجام دهد، به افزایش دقت الگوریتم معرفی‌شده بینجامد و این موضوعی است که در پژوهش‌های آتی به آن پرداخته خواهد شد.

منابع

احمدیان، ا. ح. و ه. فیلی (۱۳۹۵) «روش مبتنی بر یادگیری برای تعیین مرز بین کلمات در متن فارسی» در *مجموعه مقالات کنفرانس ملی سالانه انجمن کامپیوتر ایران*، تهران، ایران.

بی‌جن‌خان، م. (۱۳۸۳). «نقش پیکرهٔ زبانی در نوشتن دستور زبان: معرفی یک نرم‌افزار رایانه‌ای». *مجلهٔ زبان‌شناسی*. ۱۹ (۲): ۴۸-۶۷.

شریفی‌آتشگاه، م. (۱۳۸۸). *تولید نیمه‌خودکار درخت‌بانک گروه‌های نحوی در متون فارسی*. رسالهٔ دکتری، دانشگاه تهران.

طباطبایی سیفی، ش. و ا. صراف (۱۳۹۶) «سازه‌ساز: واژه‌بندی و یکسان‌سازی متون فارسی با رویکرد پیکره‌محور». در *مجموعه مقالات دومین کنفرانس بین‌المللی پژوهش‌های دانش‌بنیان در مهندسی کامپیوتر و فناوری اطلاعات*، تهران، ۱-۱۱.

عاصی، م. (۱۳۸۴). «پایگاه داده زبان فارسی در اینترنت». *پژوهشگران*. ش۲، تهران: پژوهشگاه علوم انسانی و مطالعات فرهنگی، ۱۳-۱۶.

عاصی، م. و س. قندی (۱۳۹۴). «پایگاه داده‌های زبان فارسی و پیکرهٔ تاریخی آن». در *مجموعه مقالات نخستین همایش ملی زبان‌شناسی پیکره‌ای*، به‌کوشش آ. میرزایی، تهران: نشر نویسه پارسی، ۱۹۳-۲۲۰.

فرهنگستان زبان و ادب فارسی (۱۳۸۹). *دستور خط فارسی*. تهران: فرهنگستان زبان و ادب فارسی.

قیومی، م. (۱۳۹۶). «مسئله چندواژگی در پردازش نحو رایانشی زبان فارسی». *مجموعه مقالات چهارمین همایش ملی زبانشناسی رایانشی*. به‌کوشش م. قیومی و آ. شهریاری‌فرد، تهران: نشر نویسه پارسی، ۱۱-۴۰.

قیومی، م.، س. شریفی و م. صناعتی (۱۳۹۴). «تنوع نگارشی در زبان فارسی و تهیهٔ خودکار دادگان املایی از پیکرهٔ زبانی مبتنی بر وب». *مجموعه مقالات اولین کنفرانس بین‌المللی وب‌پژوهی*. تهران: دانشگاه علم و فرهنگ.

کاشفی، ا. (۱۳۹۰). «ویراستیار: مطالعهٔ تطبیقی یک فعالیت پردازشی متن‌باز در زبان فارسی». *رهآورد نور*، ۳۴: ۹۶-۱۰۱.

وزیرنژاد، ب.، ف. سلطانزاده، م. مهدوی، و م. مرادی (۱۳۹۴). «ویرایش‌گر متن شریف: سامانه ویرایش و خطایابی املایی زبان فارسی». *پردازش علائم و داده‌ها*. ۱۲ (۴): ۴۳-۵۲.

Adda, G., M. Adda-Decker, J.Luc Gauvain, & L. Lamel (1997). "Text normalization and speech recognition in French". *Proceedings of 5th*

European Conference on Speech Communication and Technology (EUROSPEECH). Rhodes, Greece: 2711-2714.

Bijankhan, M., J. Sheykhzadegan, M. Bahrani, & M. Ghayoomi (2011). "Lessons from building a Persian written corpus: Peykare". *Language Resources and Evaluation*, 45(2):143–164.

Faili, H., N. Ehsan, M. Montazery, & M. T. Pilehvar (2016). "Vafa spell-checker for detecting spelling, grammatical, and real-word errors of Persian language". *Digital Scholarship in the Humanities*, 31 (1): 95–117.

Ghayoomi, M. & S. Momtazi (2009). "Challenges in developing Persian corpora from on-line resources". *Proceedings of 2009 IEEE International Conference on Asian Language Processing*. Singapore: 108-113.

Ghayoomi, M., S. Momtazi, & M. Bijankhan (2010). "A study of corpus development for Persian". *International Journal on Asian Language Processing.* 20 (1): 17-33.

Levenshtein, V. I. (1996). "Binary codes capable of correcting deletions, insertions, and reversals". *Soviet Physics Doklady*. 10 (8): 707–710.

Li, C., & Y. Liu (2014). "Improving text normalization via unsupervised model and discriminative reranking". *Proceedings of the ACL 2014 Student Research Workshop*. Baltimore, Maryland, USA: 86–93.

Scannell, K. (2014). "Statistical models for text normalization and machine translation". *Proceedings of the First Celtic Language Technology Workshop*. Dublin, Ireland: 33-40.

Sarabi, Z., H. Mahyar, & M. Farhoodi (2013). "ParsiPardaz: Persian language processing toolkit". *Proceedings of IEEE 3rd International eConference on Computer and Knowledge Engineering*. Mashad Ferdowsi University: 73-79.

Seraji, M., B. Megyesi, & J. Nivre (2012). "A basic language resource kit for Persian". *Proceedings of the 8th International Conference on Language Resources and Evaluation*. Istanbul, Turkey: 2245-2252.

Shamsfard, M. (2011). "Challenges and open problems in Persian text processing". *Proceedings of the 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*. Poznań, Poland: 65-69.

Shamsfard, M., H. Jafari, & M. Ilbeygi (2010). “STeP-1: A Set of fundamental tools for Persian Text Processing”. *Proceedings of the 7th International Conference on Language Resources and Evaluation*. Valletta, Malta: 859-865.

SharifiAtashgah, M., & M. Bijankhan (2009). “Corpus-based analysis for multi-token units in Persian”. *International Journal of Information and Communication Technology*. Tehran: Iran Telecom Research Center, 1 (3): 15-26.

Yang, Y., & J. Eisenstein (2013). “A log-linear model for unsupervised text normalization”. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: 61-72.