
بسم الله الرحمن الرحيم

خطایابی املایی خودکار در زبان فارسی

امید کاشفی، میترا نصری و کامیار کنعانی

همراه با ضمیمه‌های:
مبدل تقویم و مبدل عدد، سینا ایروانیان
مبدل پینگلیش، مهرداد صنوبری
اصلاح علائم نشانه‌گذاری، کامیار کنعانی

دبیرخانه شورای عالی اطلاع‌رسانی
پاییز ۱۳۸۹

سرشناسه : امید کاشفی، ۱۳۶۳-
عنوان و نام پدیدآور : خطایابی املایی خودکار در زبان فارسی/ امید کاشفی، میترا نصری، کامیار کنعانی.
مشخصات نشر : تهران: شورای عالی اطلاع‌رسانی، دبیرخانه، ۱۳۸۹.
مشخصات ظاهری : س، ۱۸۹ ص.: مصور، جدول، نمودار.
شابک : ۸-۳۴-۸۸۴۶-۹۶۴-۹۷۸
وضعیت فهرست نویسی : فیپا
موضوع : فارسی -- داده‌پردازی
موضوع : خط فارسی -- داده‌پردازی
موضوع : ویراستاری-- داده‌پردازی
شناسه افزوده : نصری، میترا، ۱۳۶۱ -
شناسه افزوده : کنعانی، کامیار، ۱۳۵۷ -
شناسه افزوده : شورای عالی اطلاع رسانی. دبیرخانه
رده بندی کنگره : ۱۳۸۹ ۶خ۲ک/PIR۲۶۴۳
رده بندی دیویی : ۰/۷ فا۴
شماره کتاب‌شناسی ملی : ۲۱۵۰۶۷۱

خطایابی املایی خودکار در زبان فارسی

تالیف: امید کاشفی kashefi@ieee.org

میترا نصری mitra.nasri@ut.ac.ir

کامیار کنعانی kanani@ce.sharif.edu

طرح روی جلد: هوتن زنگنه‌پور hootanzangeneh@yahoo.com

نوبت چاپ: اول

شمارگان: ۱۰۰۰ نسخه

ISBN: 978-964-8846-34-8

شابک: ۸-۳۴-۸۸۴۶-۹۶۴-۹۷۸

نشانی: تهران، خیابان شهید مطهری، بین خیابان سنایی و خیابان قائم مقام فراهانی، روبروی اداره امور مالیاتی شمال تهران، شماره ۳۵۸. کد پستی: ۱۵۸۶۹۹۴۳۱۱

تلفن: ۸۸۸۳۹۸۹۵، نمابر: ۸۸۸۳۹۸۹۴ ، صندوق پستی: ۳۴۹۹-۱۵۸۷۵

نشانی وبگاه: www.scict.ir

فهرست مطالب

عنوان **صفحه**

عنوان **صفحه**

عنوان **صفحه**

عنوان **صفحه**

عنوان **صفحه**

فهرست شکل‌ها

عنوان **صفحه**

فهرست جدول‌ها

عنوان **صفحه**

عنوان **صفحه**

عنوان **صفحه**

پیشگفتار

باسمه تعالی

بقا و توسعه‌ی فرهنگ ملّی درگرو رشد مدام مؤلفه‌های آن است و در این میان افزون بر مؤلفه‌های اخلاقی و عقیدتی، خط و زبان از مهمترین مؤلفه‌های فرهنگ ملّی تلقی می‌شود و یکی از عناصر کلیدی و هویت بخش در میان ما ایرانیان است. ادیبان، دانشمندان و فضلای بسیاری در حلقه‌های علمی و فکری این دیار به این زبان سخن گفته و قلم زده‌اند. در این عصر هم روزگاری بود که راهیابی این عنصر هویت بخش در فرهنگ ملی به محیط دیجیتال (اعم از برخط و برون خط) بیشتر به یک آرزو می‌ماند. اما امروز تا حدّی از این مرحله گذشته‌ایم و باید به غِنا و توانمندی محتوای فارسی موجود در محیط رایانه در رقابت با سایر زبان‌ها بیندیشیم. لذا ضروری است که در برنامه‌ای دراز مدت با دو سویه‌ی «پژوهشی» و «کاربردی»، زبان فارسی را به بستری زایا و پذیرا برای مفاهمه‌ی علمی و فنّی بدل سازیم و جامعه‌ی فارسی زبان را از این نهر جاری سیراب کنیم. از این رهگذر است که فرهنگ ایرانی- اسلامی ما توانمندتر خواهد شد و مزیّت رقابتی خود را در میان فرهنگ‌های دیگر نشان خواهد داد.

فعالیت‌های پردازشی در حوزه‌ی خط و زبان فارسی نمونه‌ای بارز و در عین حال مهم در این راستاست که تا به بار نشستن آن در همه‌ی حوزه‌ها راه پر پیچ و خمی در پیش داریم. دبیرخانه‌ی شورای عالی اطلاع‌رسانی، طی سه سال گذشته، علاوه بر فعالیت‌های جاری خود در حوزه‌ی توسعه‌ی خط و زبان فارسی در فضای دیجیتال، اهتمام خاصی به شناسایی، امکان سنجی و تحلیل مقدّماتی پروژه‌های پردازشی این حوزه داشته است. در این مسیر هم از نهادهای علمی و اشخاص حقیقی برای طراحی و پیاده‌سازی این قبیل پروژه‌ها به طور مستقیم و غیر مستقیم حمایت کرده است و این روند هم ادامه خواهد

یافت. اما طبیعی است که به دلایل مختلف فنی، اجرایی و گاهی هم سیاست‌گذارانه، برخی از این پروژه‌ها جز با ورود مستقیم این شورا به سرانجام نخواهد رسید یا آینده‌ی روشنی برای اجرا و به ثمر نشستن آن متصوّر نیست. پروژه‌ی پژوهشی- اجرایی خطایاب فارسی (ویراستیار) یکی از این قبیل پروژه‌های مهم است که با اقدام و حمایت مستقیم این شورا و همکاری جمعی از محققان به بار نشسته است و اکنون نسخه‌ی اول آن با رویکرد غلط‌یابی صَرفی زبان فارسی عرضه می‌شود.

ضمن پیشکش نرم‌افزار کاربردی «ویراستیار۱»، پژوهشنامه‌ها و پیوست‌های فنی آن به محضر علاقه‌مندان و فارسی‌زبانان سراسر جهان، از ارباب نظر خواستاریم نقدهای عالمانه و راهگشای خود را به دبیرخانه‌ی شورای عالی اطلاع‌رسانی منعکس فرمایند تا در هنگام تجدید نسخه‌ی این نرم‌افزار و ارتقای آن استفاده شود.

مطالعه، طراحی و اجرای این پروژه حاصل تلاش مشترک تعدادی از نهادها، مدیران و پژوهشگران پیشرو در حوزه‌ی «پردازش خط و زبان فارسی» است که مراتب سپاس و قدرشناسی خود را به محضرشان اعلام می‌دارم. امید است با فعالیت هم‌افزای نهادهای علمی و دانشگاهی، شاهد باروری روزافزون فعالیت‌های صورت گرفته در این بخش باشیم.
بعون الله تعالی

دکتر حمید شهریاری

دبیر شورای عالی اطلاع‌رسانی

فصل اول

مقدمه

۱-۱ مقدمه

امروزه با گسترش کاربرد رایانه، نیاز به استفاده از توانایی‌های غیر قابل چشم‌پوشی آن در حوزه‌ی زبان‌شناسی نیز به شدت احساس می‌شود. حوزه‌های پردازش زبان طبیعی[۱] و زبان‌شناسی رایانه‌ای[۲] به تلاش برای ماشینی کردن فرایند زبان‌شناسی سنتی می‌پردازند. منشأ پیدایش زبان‌شناسی رایانه‌ای را می‌توان هم‌زمان با شکل‌گیری تلاش‌هایی برای تولید ماشین ترجمه‌ی خودکار در دهه‌ی ۵۰ میلادی در ایالات متحده آمریکا دانست [1]. این ماشین ترجمه‌ی خودکار، قرار بود مجلات علمیِ روسی را به انگلیسی ترجمه کند اما با شکست این پروژه، مشخص شد که پردازش خودکار زبان طبیعی بسیار پیچیده‌تر از آن است که پیش‌تر تصور می‌شد. پردازش زبان طبیعی می‌تواند در سطوح مختلفِ زبان صورت پذیرد. پردازش زبان در هر سطح، نیازمند دانش، منابع و دادگان آن سطح و سطوح پایین‌تر است. در ادامه سطوح مختلف زبان به اختصار آمده‌اند [2, 3].

- سطح صَرف[۳] زبان، شامل آواشناسی[۴]، واج‌شناسی[۵] و واژک‌شناسی[۶] که به نحوه‌ی ساخته شدن واژه‌ها و صداها می‌پردازد.
- سطح نحو[۷] زبان که به چیدمان و ارتباط واژه‌ها به یکدیگر و مباحث دستوری آن‌ها در جملات می‌پردازد.

---

۱ معادل فارسی عبارت انگلیسی Natural Language Processing (NLP)
۲ معادل فارسی عبارت انگلیسی Computational Linguistics
۳ معادل فارسی واژه‌ی انگلیسی Inflection
۴ معادل فارسی واژه‌ی انگلیسی Phonetics
۵ معادل فارسی واژه‌ی انگلیسی Phonology
۶ معادل فارسی واژه‌ی انگلیسی Morphology
۷ معادل فارسی واژه‌ی انگلیسی Syntax

- سطح معناشناسی[۱] که به ارتباط معنایی واژه و عبارات می‌پردازد.
- سطح کاربردشناسی[۲] که به کاربردها، نگرش‌ها یا منظور اصلی از واژه‌ها، جملات و عبارات می‌پردازد.
- گفتمان[۳] که به تفسیر و تبیین متن با در نظر گرفتن بافت[۴]، موقعیت، کاربردشناسی، و چرایی تولید چنین متنی از میان امکانات مجاز موجود در آن زبان می‌پردازد [۴].

یکی از کاربردهای پردازش زبان طبیعی خطایابی و اشکال‌زدایی از متون است. با افزایش کاربرد رایانه در کشور، امروز حجم بسیار زیادی از اطلاعات و متون فارسی توسط رایانه تولید می‌شوند. فرایند تولید و ورود اطلاعات، به ویژه متن، هیچ گاه عاری از خطا نبوده و هزینه‌های بسیاری برای یافتن و برطرف ساختن این خطاها صرف می‌شود.

امروزه روند پرشتاب فن‌آوری، زمان را به سرمایه‌ای بسیار با ارزش تبدیل نموده است. سطح وسیعی از نگارندگانِ زبان فارسی شامل نویسندگان، ویراستاران، ناشران، تولیدکنندگان متون فارسی، دانشجویان، و حتی کاربرانی که به خط و زبان فارسی آشنایی دارند و از رایانه برای پردازش متون فارسی استفاده می‌کنند، نیازمند سامانه‌های خطایابی و تصحیح خودکارِ خطا هستند تا هم در هزینه و زمان صرفه‌جویی شود، و هم دقت و صحت متون افزایش یابد.

ویژگی‌های خاص خط و زبان فارسی موجب شده تا برخی چالش‌های خاص پیرامون خطایابی و تصحیح خطا به وجود آیند که در زبان‌های دیگر اصلاً مطرح نیستند. اشکالات و چالش‌های خط و زبان فارسی موجب شده تا تولید سامانه‌های خطایابی و تصحیح خودکار به کندی پیش‌رفته و نتایج به دست آمده رضایت‌بخش نباشد.

یکی از این اشکالات، نداشتن دستور خطِ جامعی متناسب با نیازهای سامانه‌های پردازش متون است. فرهنگستان زبان و ادب فارسی از سال ۱۳۷۲ شروع به بررسی، گردآوری و تدوین دستورِ خطِ فارسی نمود، اما هدف اصلی از نگارشِ آن، یکسان‌سازی چهره‌ی خط جهت کاربردِ در رایانه نبود و در بسیاری از موارد دستِ نگارندگان برای

---

۱ معادل فارسی واژه‌ی انگلیسی Semantics
۲ معادل فارسی واژه‌ی انگلیسی Pragmatics
۳ معادل فارسی واژه‌ی انگلیسی Discourse
۴ معادل فارسی واژه‌ی انگلیسی Context

انتخاب شکل نوشتار واژه باز گذاشته شد که نتیجه‌ی آن چیزی جز پیدایش ابهام در تشخیص رایانه‌ای واژه‌ها نیست.

از دیگر چالش‌های زبان فارسی، واژک‌شناسیِ غنی و پیچیده‌ی آن است. کلمات در زبان فارسی می‌توانند با ترکیب‌های بسیار زیادی از پسوندها[۱] صَرف[۲] شوند. کلمات اشتقاقی[۳] بسیار زیادی نیز در زبان فارسی وجود دارند؛ اما قوانین اشتقاق، تصریف و ترکیب[۴]، دقیق و جامع نیست. ترکیب وندها[۵] با اسامی در زبان فارسی، به علت تعدد وندها، یکی از اشکالات جدیِ واژک‌شناسیِ زبان فارسی است که هیچ گاه به طور جدی و بایسته به آن پرداخته نشده است. نداشتن قواعد تعریف شده‌ی مشخص برای ساخت و صَرف فعل‌های فارسی[۶]، وجود فعل‌های بسیار پیچیده، وجود فعل‌های چندجزئی و مرکب و نیز موارد خاص و استثناء که از قواعد صَرف فعلی تبعیت نمی‌کنند نیز از دیگر چالش‌های واژک‌شناسی زبان فارسی هستند. زبان فارسی علاوه بر فاصله‌گذاری معمول در دیگر زبان‌ها، فاصله‌ی درون واژه‌ای[۷] نیز دارد که قوانین مشخص و دقیقی پیرامون نحوه‌ی فاصله‌گذاری موجود نیست.

در این کتاب سعی شده تا ابعاد، پیچیدگی‌ها و چالش‌های پردازشی زبان فارسی، خصوصاً با رویکرد خطایابی املایی مورد بررسی قرار گیرند، راهکارهای مواجهه و مرتفع ساختن این چالش‌ها مطرح شده، در نهایت روشی برای خطایابی و اصلاح خطاهای املایی خودکار ارائه شود. امید است این کتاب، هر چند کوچک، سرآغازی برای تحول در پردازش زبان فارسی باشد که این مهم جز با یاری مخاطبان اندیشمند این کتاب و پژوهش‌گران فرزانه‌ی ایران عزیز، امکان‌پذیر نخواهد بود.

---

۱ معادل فارسی واژه‌ی انگلیسی Suffix

۲ معادل فارسی واژه‌ی انگلیسی Declension

۳ معادل فارسی واژه‌ی انگلیسی Derivation

۴ معادل فارسی واژه‌ی انگلیسی Composition

۵ معادل فارسی واژه‌ی انگلیسی Affix

۶ معادل فارسی واژه‌ی انگلیسی Conjugation

۷ معمولاً این فاصله‌ی درون واژه‌ای به نیم‌فاصله یا شبه‌فاصله (Pseudo-space) تعبیر می‌شود که نویسه‌ی صحیح نشان‌دهنده‌ی آن در کدگذاری یونی‌کد، Zero Width Non-Joiner (ZWNJ) با کد U+200C است.

۱-۲ نحوه‌ی سازمان‌دهی مطالب

در ادامه و در فصل دوم، چالش‌ها و اشکالات دستورِ خطِ فارسی مصوب فرهنگستان زبان و ادب فارسی مورد بررسی قرار خواهد گرفت و پیشنهاداتی جهت رفع اشکالات و ابهامات موجود در آن، با رویکرد زبان‌شناسیِ محاسباتی و پردازش زبان ارائه خواهد شد. در فصل سوم، ویژگی‌ها و چالش‌های پردازش زبان فارسی، خصوصاً با رویکرد خطایابی املایی مورد بررسی قرار خواهند گرفت. در این فصل سعی شده تا با نگاهی قانون محور، یک مدلِ منطقی و قابل پردازش از سطوحِ مورد نیاز زبان فارسی در خطایابی املایی (صَرف، واژک‌شناسی و واج‌شناسی) ارائه شود تا کمترین ابهام ممکن را دارا بوده، قابل پیاده‌سازی و بهره‌برداریِ پردازشی باشد. فصل چهارم، به ارائه‌ی روش‌هایی جهت خطایابی و تصحیح خطای خودکار در زبان فارسی پرداخته است. در این فصل سعی شده تا حد امکان چالش‌ها و ویژگی‌های خاص زبان فارسی که خطایابیِ املایی را تحت تاثیر قرار می‌دهند، مورد بررسی و پوشش قرار گیرند.

این کتاب همچنین شامل چهار ضمیمه‌ی فنی زیر است: ۱) مبدل اعداد، که به نحوه‌ی تشخیصِ گونه‌های مختلف نوشتار اعداد در متن و نحوه‌ی تبدیل آن‌ها به گونه‌های دیگر می‌پردازد، ۲) مبدل تقویم، که به نحوه‌ی تشخیص گونه‌های مختلف نوشتارِ تقویم در انواع مختلف نظامِ تقویم در متن (شمسی، میلادی و هجری)، و نحوه‌ی تبدیل آن‌ها به گونه‌های نوشتاری و نظام‌های تقویم دیگر می‌پردازد، ۳) مبدل نوشتار فارسی با حروف انگلیسی (پینگلیش) به فارسی، که به ارائه‌ی روشی جهت تبدیلِ متون پینگلیش به متونی با خط فارسی می‌پردازد، و ۴) اصلاح علائم نشانه‌گذاری، که به ارائه‌ی قواعد نشانه‌گذاری در زبان فارسی و روشی جهت تشخیص و اصلاح خطاهای نشانه‌گذاری می‌پردازد.

فصل دوم

چالش‌ها و اشکالاتِ پردازشیِ دستورخط فارسی
(مصوب فرهنگستان)

۲-۱ مقدمه

با توجه به سرعت رشد فن‌آوری اطلاعات در کشورهای پیشرفته‌ی جهان، باید تلاش نمود تا موانع رشد این صنعت در ایران به سرعت بر طرف شود. یکی از این موانع، نداشتن دستور خطِ جامعی متناسب با نیازهای سامانه‌های پردازش متون است. مهم‌ترین اقدام رسمی برای حفظ چهره‌ی خط فارسی را فرهنگستان زبان و ادب فارسی از سال ۱۳۷۲ شروع کرد که تلاشی تحسین برانگیز بود، اما هدف اصلی از نگارشِ آن، یکسان‌سازی چهره‌ی خط جهت کاربرد در رایانه نبود. در نتیجه، در بسیاری از مواردِ دست نگارندگان برای انتخاب شکل نوشتار واژه باز گذاشته شد که نتیجه‌ی آن چیزی جز پیدایش ابهام در تشخیص رایانه‌ای واژه‌ها نبود.

وجود ارتباط متقابل میان زبان‌شناسان و فن‌آورانِ حوزه‌ی رایانه یکی از نیازهای اصلی جامعه‌ی اطلاعاتی امروز و رشد صنعت پردازش الکترونیکی در کشور عزیزمان است. اگر در خط فارسی تغییری برای همگام شدن با این رشد روی ندهد، میزان عقب‌ماندگی ما در فن‌آوری اطلاعات از سایر کشورها جبران ناپذیر خواهد شد. زمانی فرا خواهد رسید که روزنامه‌ها و مقالاتِ خارجی با سرعت بسیار تولید می‌شوند و خلاصه‌ی آن‌ها در کسری از ثانیه استخراج می‌شود، در حالی که ما در ایران حتی کار خطایابی املایی را نیز با خطای بالا و به کُندی انجام می‌دهیم. زمانی که موتورهای جستجوی خارجی می‌توانند هنگام جستجوی یک واژه، مترادف‌ها، ریشه‌ها و سایر مشتقات آن را نیز بازیابی کنند، ما همچنان در حال رفع مشکلِ چندگانگی کدِ نویسه‌ی «ی» و یا مشکل اجزای واژه هستیم و یک جستجوی ساده را نیز نمی‌توانیم در وب‌گاه‌های رسمی کشور به درستی انجام دهیم. از این رو اهمیت موضوعِ یکسان‌سازی نحوه‌ی نگارش واژه‌ها و نیز موارد ابهام‌زای موجود در دستورِ خطِ فارسی دوچندان می‌شود و باید تصمیمی جدی جهت رفع اشکالات و چالش‌های این بخش اندیشید.

۲-۲ دستور خطِ فارسی و رایانه

۲-۲-۱ ویژگی‌های خط فارسی در پردازش رایانه‌ای

خط فارسی نیز مانند سایر خط‌های دنیا ویژگی‌هایی دارد که پردازش آن را توسط رایانه مشکل می‌نمایند. این ویژگی‌های ذاتی از آنجا ناشی می‌شوند که در زبان‌های طبیعی بسیاری از واژگان از ترکیب با واژگان دیگر ساخته می‌شوند. اگر در نگارش واژگانی که از این ترکیب‌ها به وجود آمده‌اند قواعد مشخصی رعایت نشود، واژگان حاصل ممکن است معنایی متفاوت از آنچه در آغاز مورد انتظار بوده است پیدا کنند. از دگر سو، تفاوت‌هایی نیز میان زبان فارسی و سایر زبان‌های طبیعی دنیا وجود دارد. در واقع، نحوه‌ی ساخت واژه‌ها و اتصال آن‌ها در فارسی، دسته‌ی دیگری از مشکلاتِ مربوط به پردازش متون را در این زبان پدید می‌آورند. در ادامه این ویژگی‌های خاصِ زبان فارسی مورد بررسی قرار خواهند گرفت.

۲-۲-۱-۱ *ویژگی‌های عمومی*

پردازش واژگانیِ کلیه‌ی زبان‌های طبیعی امری دشوار است. ترکیب واژگان، منجر به تشکیل واژگانی می‌شود که ممکن است در اثر بی‌دقتیِ کاربران، از دید رایانه به دو یا چند شکل مختلف خوانده شوند. مثلاً در جایی که منظور نویسنده «سیب‌زمینی» است، اگر در اثر بی‌دقتی «سیب زمینی» نوشته شود، رایانه قادر به تشخیص واژه‌ی اصلی نخواهد بود.

دومین دلیل پیچیدگی پردازش واژگانی زبان‌های طبیعی، ترکیب واژها با یکدیگر و تولیدِ واژه‌هایی است که حاوی اطلاعاتی مانند مالکیت، جمع یا مفرد بودن واژه هستند (به عنوان نمونه «کتاب‌هایشان»). این واژه‌های جدید در واژه‌نامه‌ها وجود ندارد اما معنای آن‌ها همان معنایی است که در واژه‌ی اولیه نهفته بوده است. از دید رایانه تنها در صورتی دو واژه با هم یکسان هستند که به یک شکل نوشته شده باشند.

سومین دلیل مشکل بودن تفسیر واژه از دید رایانه، آن است که برخی از قاعده‌های تولید واژه در زبان‌های طبیعی، می‌توانند واژه‌هایی به وجود آورند که در واژه‌نامه‌ها وجود ندارند مانند «بازنگریسته». از سوی دیگر، اگر رایانه بتواند تمام قاعده‌های ساخت واژه‌ها را در خود جای دهد، در آن صورت واژگانی که امکان تولید آن‌ها در زبان وجود دارد اما گویش‌وران زبان تاکنون آن‌ها را به کار نبرده‌اند نیز در زمره‌ی واژه‌های مورد تایید رایانه قرار خواهند گرفت. بنابراین، پردازش واژه‌های یک متن به خودی خود رایانه را با مشکلاتی در تشخیص واژه‌ها مواجه می‌کند. این‌ها همگی در صورتی هستند که اشتباهات

املایی کاربران در نظر گرفته نشود و نیز تمام کاربران قواعد و اصول نسبتاً یکسانی را در نگارش خود به کار برند.

۲-۲-۱-۲ *ویژگی‌های اختصاصی زبان فارسی*

در این بخش ویژگی‌های زبان فارسی که منجر به پیچیدگیِ کار پردازش رایانه‌ای می‌شوند بررسی خواهند شد. گفتنی است برخی از مشکلاتی که این ویژگی‌ها ایجاد می‌کنند مربوط به ماهیت زبان فارسی است و حتی با تغییر خط فارسی نیز حل نمی‌شود.

ساخت واژه‌ها در زبان فارسی از قواعدی پیروی می‌کند که متفاوت با زبان نسبتاً فراگیرترِ انگلیسی است. در زبان انگلیسی (برای نمونه)، اضافه شدن پسوندهای متعدد به یک واژه، کاربرد بسیار محدودی دارد اما بسیاری از ترکیب‌های زبان فارسی از طریق اتصال تعدادی وند یا واژه‌ی مستقل به واژهای دیگر ساخته می‌شوند مانند «شیرینی‌خوران»، «خداشناس»، «حقیقت‌جو»، «آب‌سردکن». واژه‌ها در بسیاری از موارد از ترکیب چند واژه‌ی مستقل و با معنی در کنار یکدیگر به وجود آمده‌اند در حالی که اگر میان این اجزا فاصله ایجاد شود، به دو واژه‌ی مستقل تبدیل خواهند شد. این مشکل در زبان انگلیسی بسیار کم‌تر رخ می‌دهد.

در زبان فارسی، ضمایر مفعولی و نشانه‌های جمع به واژه‌ها متصل می‌شوند. این امر موجب پیچیدگیِ تفکیکِ رایانه‌ای واژه‌ها می‌شود زیرا حروف نشان‌دهنده‌ی ضمایر مفعولی و نشانه‌های جمع، با تعدادی از وندها و نیز واژگان فارسی مشابهت دارند. در نتیجه، رایانه نمی‌تواند به سادگی در مورد بخشِ اصلیِ واژه قضاوت کند. برای نمونه، «ان» در «درختان» نشانه‌ی جمع است در حالی که «ان» در «خوران» نشانه‌ی صفت فاعلی است. بنابراین، در کاربردهای رایانه‌ای برای آن که بتوان واژه‌های حاصل از بن مضارع یا بن ماضی را تعریف کرد (مثلاً برای آن که بتوان مشخص کرد که «ان» در «خوران» نشانه‌ی جمع است یا نشانه‌ی صفت فاعلی)، هیچ امکانی وجود ندارد. در بسیاری از کاربردهای پردازشی زبان، مثلاً در نرم‌افزار موتورهای جستجو (مانند Google)، رایانه‌ها باید تمام واژه‌های متن را استخراج نمایند. در این برنامه‌ها، نشانه‌های جمع یا ضمایر مفعولی باید از واژه حذف شوند. از سوی دیگر، داشتن کارایی بالا (سرعت بالا) یکی از ملزومات این نرم‌افزارهاست. تشخیص این که یک نشانه (مانند «ان») در یک واژه، یک نشانه‌ی قابل حذف است (مانند نشانه‌ی جمع) یا خیر (مانند نشانه‌ی صفت فاعلی)، نیازمند جستجو در واژه‌نامه‌ای است که باید حاوی بن‌های مضارع و ماضیِ تمام فعل‌ها باشد. همچنین، رایانه

باید حالت‌های ممکن را بررسی کند تا مطمئن شود که یک واژه را به درستی از نشانه‌های اضافه‌اش جدا نموده است. حال، اگر یک نشانه‌ی جمع در کنار نشانه‌ی دیگری مانند فعل‌های اسنادی که به صورت اختصاری نوشته شده است یا ضمایر مفعولی قرار گیرد (مانند «درختانند»، «شیرینی‌خورانشان» یا «درختانمان»)، تشخیص آن، زمان بیشتری خواهد برد و در کارایی موتورهای جستجو تأثیر خواهد داشت و در مواردی برای آن‌ها راه حلی وجود ندارد. در زبان انگلیسی چنین مشکلاتی به مراتب کم‌تر رخ می‌دهند.

در زبان فارسی، *اِعراب* با وجود آن که تلفظ می‌شود، اکثراً نوشته نمی‌شود. اگر واژه‌ای با اِعراب نوشته شود از دید رایانه با حالت بی‌اعراب آن متفاوت خواهد بود. در نتیجه، لازم است که همواره برای واژه‌های با اعراب روش جداگانه‌ای در نظر گرفته شود. از طرفی، اکثر فارسی‌نویسان از گذاشتن برخی از حرکت‌های الزامی مانند تشدیدهای لازم و تنوین خودداری می‌کنند. نتیجه‌ی این امر، متفاوت شدن صورت ظاهری واژه در رایانه با آن چه در واژه‌نامه‌ها وجود دارد است.

به دلیل آموزش‌های متفاوتی که از طریق نظام آموزشِ کشور به فارسی‌نویسان داده شده است، حفظ یکپارچگی خط فارسی برای نویسندگانِ مختلف که آموزش‌های متفاوتی دیده‌اند مشکل است. در زبان‌های دیگر این اتفاق (تغییر دستور خط) بسیار به ندرت رخ می‌دهد؛ در نتیجه، تمام مردم در سنین مختلف از طرز صحیح نگارش واژه‌ها و دستور خطِ خود، آگاهی دارند. البته آگاهی داشتن دلیل بر رعایت کردن اصول نیست اما وجود توافق کلی در نگارش واژه‌ها امری است که موجب تصحیح متونِ رسمی و یک‌دستی خط می‌شود. تغییرات خط فارسی در نظام آموزش کشور در هر دوره مانند تحوّلی برای کاربران قدیم این خط به حساب آمده است به نحوی که شکل این خط زمانی از جدانویسی به پیوسته‌نویسی و بعد مجدداً به جدانویسی و در نهایت به روشی تلفیقی تبدیل شده است.

تفاوت‌هایی که در زبان محاوره و زبان نوشتارِ فارسی وجود دارد عامل دیگری برای ناهم‌خوانیِ نگارش‌های مختلف یک متن در زبانِ فارسی است. امروزه بسیاری از کاربرانِ وب‌نوشت‌های فارسی، زبان محاوره را مستقیماً مکتوب می‌نمایند. یعنی به جای نوشتن «به او گفتم خانه هستم»، نگارش‌های «به او گفتم خونم» و «بهش گفتم خونم» جانشین می‌شود. این تفاوت میان زبان گفتار و نوشتار در سایر زبان‌های طبیعی تاثیر کم‌تری دارد و حداکثر ساختار جمله را تغییر می‌دهد در حالی که در فارسی هم ساختار اجزای جمله تغییر می‌کند و هم تلفظ متفاوت واژه‌ها، املای آن‌ها را تغییر می‌دهد.

۲-۲-۲ نویسندگان متون رایانه‌ای

همان طور که گفته شد، به دلیل تغییر دستورِ خط در نظام آموزش کشور، اکثر نویسندگان عادی زبان فارسی، در نحوه‌ی نگارش واژه توافق ندارند. در نتیجه متونی که این گروه تولید می‌کنند (که عمدتاً شامل متون غیر رسمی وب‌نوشت‌های فارسی است)، از هیچ سبک و قاعده‌ی خاصی پیروی نمی‌کنند. از این دسته از کاربران که صرف نظر کنیم، کاربرانی را خواهیم دید که در مکان‌های غیر رسمی مشغول حروف‌چینی متون با رایانه هستند. این کاربران اکثراً دستورِ خطِ فارسی را در اختیار دارند اما متنی که هر یک از آن‌ها تولید می‌کند از لحاظ کیفیت ویرایشی، با دیگری متفاوت است. در واقع بسیاری از آن‌ها پاره‌ای از اصول اصلی ویرایش را با وجود آگاهی از آن رعایت نمی‌کنند. ساده‌ترین مثال، قاعده‌ی استفاده از تنوین است. البته گستردگیِ بیش از حد در رعایت نکردن این قاعده، شاید دلیلی بر لزوم بازنگری در آن باشد. در این زمینه در بخش‌های بعدی توضیحات کامل‌تری ذکر خواهد شد.

ویراستاران حرفه‌ای نیز با وجود آشنایی با دستورِ خطِ فارسی و اصول ویرایش، توافق کلی در نگارش واژه‌ها ندارند. مثلاً ممکن است که یک ویراستار خاص واژه «آن‌ها» را به صورت «آنها» بنویسد (یا به عکس) زیرا دستورِ خطِ فارسی در این زمینه قواعد مشخص، قاطع و بدون ابهامی ندارد. این کار باعث می‌شود تا نتیجه‌ی تحلیل متنی که یک ویراستار حرفه‌ای به رایانه داده است، مانند نتیجه‌ی تحلیل متن مشابهی که ویراستار حرفه‌ای دیگری آن را ویرایش کرده است نباشد.

۲-۲-۳ دستورِ خطِ فارسی مصوب فرهنگستان

دستورِ خطِ فارسی مصوب فرهنگستان زبان و ادب فارسی، تنها مرجع رسمی برای خط فارسی است که آخرین نسخه‌ی آن در سال ۱۳۸۴ منتشر شده است. این مرجع کوچک با ملاحظه‌ی تنوع کاربران آن و به زبانی قابل درک برای تمامی آن‌ها (و به دور از اصطلاحات تخصصی زبان‌شناسان) نوشته شده است و برای بیشتر کاربران قابل درک و فهم، و نیز کاربردی است که از این جهت، گامی مؤثر در یکسان‌سازی خط فارسی به حساب می‌آید. این شیوه‌نامه شامل بندهایی در زمینه‌ی نشانه‌های خط فارسی، املای برخی از واژگان خاص، پیشوندها و پسوندها، و نیز قاعده نگارش ترکیب‌ها است. سر فصل‌های انتخاب شده برای دستورِ خطِ فارسی مصوب فرهنگستان زبان و ادب فارسی، نسبتاً کامل هستند و مسائل اساسیِ نوشتار فارسی را شامل می‌شود. سادگی مطالعه و درک آن و

مثال‌های ذکر شده برای هر بند از قواعد نیز قابل تقدیرند. با این حال، دستورِ خطِ فارسی مصوب فرهنگستان زبان و ادب فارسی، برای کاربرد در سامانه‌های رایانه‌ای طراحی نشده است. به این معنی که مخاطب این جزوه نویسندگان عادی متون فارسی‌اند و با وجود تلاشی که برای ساماندهی خط برای نویسندگان متون رقمی (رایانه‌ای) شده است، نتایج به دست آمده قابل قبول نیستند.

مهم‌ترین چالش در حوزه متون رقمی، مسئله‌ی فاصله‌گذاری و پس از آن اعراب است. تا زمانی که ابهام این دو بخش به طور دقیق برای کاربران رایانه حل نشود، متون تولیدی این کاربران برای مراحل بعدی پردازش در رایانه مناسب نخواهد بود و در نتیجه تنها کاربردشان این است که توسط کاربر دیگری خوانده شوند. در این حالت، جستجو، محاسبه‌ی میزان پیچیدگی، خلاصه‌سازیِ خودکار، یا تشخیص و تصحیح خودکارِ خطاهای املایی در این متون، مشکلات بسیاری را در بر خواهد داشت. تا زمانی که دو کاربر ورزیده و دانش‌آموخته‌ی زبان فارسی وجود داشته باشند که املای یک واژه را متفاوت بنویسند، سامانه‌های رایانه‌ای مشکل پردازشی در متون فارسی خواهند داشت.

تعدد واژه‌هایی که مستعدِ چندگونگی در نوشتار هستند، ملاک تعیین اهمیت اشکالات دستورِ خطِ فارسی در یکسان‌سازی چهره‌ی خط برای کاربران رایانه است. در زمینه قواعدِ مشخصی که دست کاربران را برای نگارش آزادانه‌ی واژه‌ها می‌بندد، در بسیاری از محافل ادبی بحث‌های فراوانی شده است، همان‌طور که نگرش‌های جدانویسیِ و پیوسته‌نویسیِ افراطی در برهه‌های زمانی متفاوت، رد یا تایید شدند. اما نباید از یاد برد که پیشرفت کشور در حوزه‌ی فن‌آوریِ اطلاعات، در گرو اهمیت دادن به مسئله‌ی یکسان‌سازی چهره‌ی خط فارسی است. نبود تحول در این حوزه و نداشتن تمایل سازمان‌های رسمی و غیر رسمی به تقبّل پروژه‌هایی که خاص زبان فارسی باشند، ناشی از اطلاع آن‌ها از به‌هم‌ریختگی قواعد دستورِ خطِ فارسی است. در واقع، اگر خط فارسی و قواعد نگارش تا این حد ابهام‌زا نبود، ما نیز کمی پس از دهه‌ی ۱۹۸۰ که اولین خطایابِ املاییِ تجاریِ زبان انگلیسی به بازار عرضه گردید، می‌توانستیم این سامانه را در ایران تولید نماییم.

۲-۲-۳-۱ *اشکالات عمده‌ی دستورِ خطِ فارسی*

به طور خلاصه از دید رایانه‌ای، می‌توان ایرادهایی را که به دستورِ خطِ فارسی فرهنگستان

زبان و ادب فارسی وارد است چنین دسته‌بندی کرد:

- باز گذاشتن دست نویسندگان در فاصله‌گذاری میان واژه
- نداشتن دستورالعمل قطعی برای استفاده از نیم‌فاصله
- نبود قواعد ثابت برای فاصله‌گذاری ترکیب‌ها؛ استفاده از دستورالعمل مبتنی بر واژه (مانند تک‌هجایی بودن، بسیط‌گونه بودن)

البته حتی اگر تمام این اشکالات حل شوند و ابهام‌ها برطرف شوند، همچنان در زبان فارسی کاربرانی خواهند بود که واژه‌ها را خارج از این استاندارد می‌نویسند. اما، همان طور که در بخش‌های قبل کاربران و اشکالات دسته‌بندی شدند، بدون تحول بنیادین در خط و زبان فارسی مشکلات عدم هم‌خوانیِ واژه‌ها همواره وجود خواهند داشت.

هدف از بیان چالش‌های دستورِ خطِ فارسی آن است که بتوان برای زبان فارسی شیوه‌ی خطی به وجود آورد که اگر کاربران حرفه‌ای آن‌ها را به کار بَرند، میزان ابهام‌های تولیدی برای تحلیل رایانه‌ای متون به حداقل برسد. این موضوع برای حفظ حیات الکترونیکی زبان فارسی امری ضروری است.

۲-۲-۳-۲ *رویکردهای موجود در فاصله‌گذاری*

پیش از آغاز این بخش، قسمت‌هایی از سطرهای آغازین کتاب دستورِ خطِ فارسی (ص۲) مرور می‌شود:

«در باب دستورِ خطِ فارسی، همواره اختلاف سلیقه و مشرب وجود داشته است؛ بعضی طرفدار باز گذاشتن دست نویسنده در انتخاب شیوهٔ نگارش بوده و حداکثر جواز و رخصت را تجویز می‌کرده‌اند و بعضی دیگر، برعکس، گرایش به وضع قوانینی عام و قطعی و تخلّف‌ناپذیر داشته و آرزو می‌کرده‌اند که در عالم خط و کتابت نیز قوانینی شبیه قوانین حاکم بر علائم ریاضیات حاکم باشد. از جهتی دیگر، برخی از اهل فن معایب و مشکلات موجود در خطّ فارسی را تا آن اندازه فراوان و جدّی دانسته‌اند که رفع آن‌ها را جز با افزودن و در کار آوردن حروف و علائم جدید میسّر نمی‌شمرده‌اند، و گروهی دیگر کم‌ترین تحوّل و تبدّلی را در خطّ فعلی نپذیرفته و آن را به زیان زبان می‌دانسته‌اند».

در مورد راهبرد اصلی فرهنگستان زبان و ادب فارسی در باب تغییرات اساسی در زبان و یا افزودن نشانه‌های جدید به آن اختیار با استادان زبان فارسی و صاحب نظران است اما در زمینه‌ی فاصله‌گذاری میان واژه‌ها که یکی از بزرگ‌ترین چالش‌های کنونی متون

فارسی است، می‌توان دلایل گروه‌های موافق با پیوسته‌نویسی و جدانویسی را به شکل زیر دسته‌بندی نمود:

– پیوسته‌نویسی کامل

- ♦ درج فاصله برای کاربرانی که می‌خواهند واژه‌ای مانند «میشود» را به صورت «می شود» بنویسند باعث خواهد شد که «می» در مواردی در انتهای خط اول باقی بماند و «شود» به ابتدای سطر بعد منتقل گردد. این کار از خوانایی نوشته می‌کاهد.
- ♦ اگر برای جبران انتقال «می» به سطر بعد، از نیم‌فاصله استفاده شود مستلزم آشنایی کاربران با این کلید و موقعیت آن در صفحه کلید است. زدن این کلید تنها برای کاربرانی ساده است که با گذشت زمان به آن عادت کرده‌اند.
- ♦ اگر واژه‌ها پیوسته نوشته شوند و واژه‌نامه‌ها نیز بر اساس اصول پیوسته‌نویسی تدوین شده باشند، جستجوی واژه‌ها در واژه‌نامه بسیار آسان خواهد بود.

– جدانویسی کامل

- ♦ چشم انسان‌ها واژه‌هایِ پر حرف را به درستی نمی‌خواند. مثلاً اگر بخواهیم واژه «عافیت‌طلب» را به شکلِ «عافیتطلب» بنویسیم، خواندن آن برای هر خواننده‌ای مشکل است و نیاز به مکثِ اضافه دارد. در نتیجه جدانویسی منجر به سادگیِ خواندن واژه‌ها می‌گردد.
- ♦ اگر تمام واژه‌ها جدا نوشته شوند ابهامی در معنای کلام باقی نمی‌ماند. هر بخش از واژه به صورت مجزا نوشته می‌شود. در حالی که در پیوسته‌نویسی، مشخص نیست که تا کجا باید واژه‌ها را به هم متصل نمود و اجزای واژه دقیقاً کدام‌ها هستند.
- ♦ جدانویسی در صورتی که همراه با استفاده از نیم‌فاصله باشد، فرایند تفکیک واژه‌ها را ساده می‌کند. زبان‌های طبیعی که اجزای متصل کم‌تری دارند در پردازش رایانه‌ای ساده‌تر هستند. به عکس هر چه بخش‌های بیشتری از واژه به هم متصل شوند، تشخیص اجزا برای رایانه پیچیده‌تر خواهد شد. در نتیجه، مواردی همچون تشخیص و تصحیح املاء واژه و تشخیص نقش دستوریِ آن مشکل‌تر خواهد شد.

– راه حل بینابینی

- ♦ نه کاملاً پیوسته و نه کاملاً جدا. به این ترتیب ظاهر واژه‌ها واژه حفظ می‌شود و حداقلی از اصول اولیه نیز رعایت می‌گردد.
- ♦ دستورالعمل‌های مجزا برای جدانویسی و پیوسته‌نویسی ارائه می‌شود و سایر موارد مطابق با سلیقه‌ی نویسنده خواهد بود.

با وجود رویکردهای مختلف، نکاتی وجود دارند که در جمع‌بندی نهایی نباید فراموش شود. اگر حروف واژه زیاد شوند، پیوسته‌نویسی واژه آن را از شکل قابل خواندن خارج می‌نماید. خواندن واژه‌هایی که بیشتر از ۷ الی ۸ حرف در بخش اصلی خود دارند، برای بیشتر خوانندگان سخت و نیازمند به مکث است.

چشم خوانندگان به الگوی واژه بیش از حروف آن‌ها عادت دارد. در واقع بسیاری از واژه قبل از آن که در ذهن انسان حرف به حرف پردازش شوند، به کمک شکلشان در ذهن تشخیص داده می‌شوند. مثلاً واژه «خدافاحظ» با آن که خطا نوشته شده در لحظه‌ی اول واژه‌ای «خداحافظ» را تداعی می‌نماید. در حالی که اگر حرف به حرف پردازش می‌شد، این فرایند بیشتر طول می‌کشید. با توجه به این واقعیت، هیچ دستورالعملی که بخواهد شکل بسیاری از واژه را تغییر دهد، در میان مردم فراگیر نخواهد شد.

مشکل واژه‌های واحد مرکبی که با فاصله از هم جدا می‌شوند (مانند «خوش‌خیالی»)، زمانی که به انتهای خط می‌رسند و در نتیجه‌ی آن یک بخش از واژه در انتهای سطر اول و بخش دیگر در ابتدای سطر دوم قرار می‌گیرد، به کمک نیم‌فاصله حل می‌شود. اما باید برای استفاده از نیم‌فاصله قواعد دقیق و ثابتی تعیین نمود. درست است که استعمال نیم‌فاصله برای کاربران مبتدی سخت است اما در صورت استفاده نکردن از این امکان باید خط فارسی را از اساس تغییر دهیم که امکان این تغییر وجود ندارد یا بسیار دشوار است. اگر استفاده از این نویسه در زمان تدریس نرم‌افزارهای ویرایشگر متن، به کاربران آموزش داده شود، مثلاً در سرفصل‌های دروس پرطرفدارِ گواهی‌نامه‌ی کاربریِ رایانه[1]، یک فصل با عنوان اصول اولیه نگارش گنجانده شود، برخی از این اشکالات برطَرف خواهند شد.

اگر از جدانویسی واژه‌ها بی‌رویه استفاده شود، منجر به اِشکال در خواندن واژه خواهد شد. مثلاً اگر «بدبختانه» به صورت «بدبخت‌انه» نوشته شود، خواندن آن دشوار و به چشم

[1] معادل فارسی عبارت انگلیسی International Computer Driving License (ICDL)

بیننده ناآشنا خواهد آمد. باید دقت داشت که برخی از اجزای واژه مانند بسیاری از پسوندها، مدت‌هاست که در واژگان زبان نفوذ کرده‌اند و جایگاه محکمی یافته‌اند. واژگان تولیدی از این راه، با وجود آن که در ذات خود مرکب هستند اما بدون پسوندِ خود، کاربردی متفاوت خواهند یافت و از دید خوانندگان هم پیچیده به نظر خواهند رسید.

با وجود آن که نباید شکل واژگان زبان را با یک قاعده زیر و رو نمود اما بسیاری از واژگان برای چشم‌های متفاوت، متفاوت به نظر می‌رسند. برای مثال، بسیاری از کاربران به واژه «میشود» عادت کرده‌اند در حالی که گروه عمده‌ی دیگری چنین نیستند. برای یکسان‌سازی چهره‌ی خط فارسی، در نهایت تغییر برخی از روش‌های نگارش امری ناگزیر خواهد بود. بزرگ‌ترین مشکلِ راه حل‌های بینابینی، ابهام آن‌ها در استفاده از قواعد است. زمانی که انتخاب شکل نگارش واژه رسماً به سلیقه‌ی نویسنده سپرده شود، اشکالات پردازشی زبان آغاز خواهند شد.

۲-۲-۴ نمونه‌هایی از کاربردهای نیازمند به یکسان‌سازی خط فارسی

نیازی که امروز در قالب این کتاب دیده می‌شود نیازی است که قابل اغماض نیست و رشد صنعتِ پردازش متون در ایران به آن وابسته است. برای روشن شدن عمق تاثیرِ خط فارسی، نمونه‌هایی از کاربردهایی که نیاز به قاعده‌مند شدن زبان فارسی دارند را از نظر می‌گذرانیم:

- خطایابی املایی
- خطایابی ویرایشی و دستوری
- موتور جستجوی فارسی
- بازشناسی خودکار حروف فارسی
- خلاصه‌سازی فارسی
- استخراج واژه‌های کلیدی متن
- شباهت‌سنجی میان متون
- پالایش متون
- ترجمه‌ی ماشینی
- دسته‌بندی و خوشه‌بندی متون
- انواع داده‌کاوی و متن کاوی
- نمایه‌گذاری

۲-۳ واژگان خاص

در این بخش به مرور بخش‌هایی از دستورِ خطِ فارسی می‌پردازیم که با عنوان «املای بعضی از واژه‌ها، پیشوندها و پسوندها» آورده شده‌اند. یادآوری می‌شود که این گزارش فقط مواردی را ذکر خواهد نمود که عملیات پردازش رایانه را با اختلال مواجه می‌نمایند. جدول (۲-۱) نشان‌دهنده‌ی موارد مبهم در رایانه است. در این جدول ابتدا واژه‌ای مورد بحث (مانند «بی») ذکر گردیده و به دنبال آن قاعده‌ی ابهام‌زا آورده شده. در نهایت نیز مشکل رایانه با این قاعده عنوان شده است. در ستون سمت راست در موارد معدودی، نکته یا پیشنهاد حل مشکل نیز ارائه شده است.

جدول (۲-۱) ابهام‌های قواعد دستورِ خطِ فارسی

واژه/ نشانه	قاعده‌ی جاری	مشکل
ابن	حذف یا حفظ همزه اگر در میان دو اسم عَلَم قرار گیرد. مثال: حسین‌بن‌راضی، حسین‌ابن‌علی	اشکال: اسامی عَلَم برای رایانه شناخته شده نیستند. راه حل: اگر «بن» یا «ابن» با نیم‌فاصله از دو اسم کنار خود جدا شوند مشکلی رخ نخواهد داد. با وجود آن که «حسین‌بن‌علی» با «حسین‌ابن‌علی» متفاوت است، اگر با نیم فاصله کنار هم نوشته شده باشد رایانه می‌تواند «بن» را در مواردی جانشین «ابن» و به عکس نماید. نکته: بهتر است دستور صریحی برای استفاده از فاصله یا نیم‌فاصله در استفاده از «بن» و «ابن» وجود داشته باشد.
هم	«هم» اگر واژه بسیط‌گونه بسازد، به شکل پیوسته نوشته می‌شود.	اشکال: رایانه نمی‌تواند بسیط‌گونه بودن واژه‌ها را تشخیص دهد مگر آن که این واژه‌ها در واژه‌نامه ضبط شده باشند. نکته: در صورتی که «هم» باید جدا از واژه‌ی بعد از خود نوشته شود، نیازمند دستور صریحی برای استفاده از فاصله یا نیم‌فاصله هستیم.
به	«به» هرگاه صفت بسازد پیوسته نوشته می‌شود.	اشکال: اگر صفتِ حاصل، در واژگان وجود نداشته باشد، رایانه نمی‌تواند صحت نگارش واژه را تأیید کند. راه حل: صفت‌هایی که با «ب» ساخته می‌شوند باید حتماً در واژه‌نامه وجود داشته باشند. تمام کاربران نیز باید از صفت بودنِ واژه‌ی حاصل اطمینان داشته باشند. نکته: در صورتی که «به» باید جدا از واژه‌ی بعد از خود نوشته شود، نیازمند دستور صریحی برای استفاده از فاصله یا نیم‌فاصله برای آن هستیم.

ادامه‌ی جدول (۲-۱)

واژه/ نشانه	قاعده‌ی جاری	مشکل
به	«بـه» هرگـاه پـیش از واژه‌ی عربـی قـرار گیـرد، پیوسـته نوشته می‌شود.	اشکال: واژه‌نامه‌ی جامعی بـرای واژه‌هـای عربـیِ بـه کـار رفتـه در متون فارسی وجود ندارد در نتیجه رایانه راه مشخصی در برخـورد با این واژه‌ها نخواهد داشت. از دید رایانـه ایـن واژه‌هـا مـی‌تواننـد صورت‌هایی از واژه‌ها فارسی با خطاهای املایی تلقی شوند.
بی	در صـــــورتی کـــــه واژه بسـیط‌گونـه باشـد، «بـی» بـه شکل پیوسته نوشته می‌شود.	اشکال: رایانه از بسیط‌گونه بودن واژه‌ها بی‌خبر است مگـر آن کـه این واژه‌ها در واژه‌نامه وجود داشته باشند. اشکال: اگر شُبه‌ای در مورد اینکه کاربران زبان، یـک واژه سـاخته شده با «بی» را بسیط‌گونه بدانند وجود دارد (ماننـد بـی‌راه)، در آن صـورت ممکـن اسـت بسـیاری از آنـان واژه‌هـای واحـد را بـه دو صورت متفاوت بنویسند. نکته: اگر باید «بی» را جدا از واژه‌ی بعـد از خـود نوشـت، نیازمنـد دسـتور صـریحی بـرای اسـتفاده از فاصـله یـا نـیم‌فاصـله بـرای آن هستیم.
هم	اگـــــر جـــــزء دوم واژه، تک‌هجـایی باشـد، «هـم» بـه صـــورت پیوســـته نوشـــته می‌شود.	اشکال: رایانه نمی‌تواند تک‌هجایی یا چندهجایی بودن واژگـان را تشخیص دهد. در هیچ مرجع فارسی نیز واژه‌نامه‌ی حرکـت‌دار (بـا اِعراب) تدوین نشده است. بنابراین، رایانه روشـی بـرای تشـخیص تعداد هجاهای واژه‌ها ندارد. تنها راه حل باقی مانده، آن است کـه صورت پیوسته‌ی این واژگان در واژه‌نامه وجود داشته باشد. اشکال: اگر کاربری واژه‌ای را که با «هم» شروع می‌شود به اشـتباه یا به دلیل از یاد بردن قاعده تک‌هجایی، به صـورت جـدا نوشـت، تشـخیص اینکـه ایـن واژه، همـان واژه‌ی اول اسـت بـرای رایانـه ناممکن خواهد بود.
هم	«هـــم» اگـــر جـــزء دوم بـــا مصـوت «آ» شـروع گـردد پیوسته نوشته مـی‌شـود. اگـر همزه قبل از «آ» ظاهر شـود، جدا نوشته می‌شود.	اشکال: برای کاربران اجرای ایـن دو بنـد سـاده نیسـت و منجـر بـه جدانویسی برخی از واژگان بند اول می‌شود. در نتیجه نگـارش دو کـاربر مختلـف ممکـن اسـت بـرای ایـن واژه یکسـان نباشـد. هـر شیوه‌ای که منجر به بروز ابهام در نوشتن واژه‌ها گـردد رایانـه را از مسیر تشخیص درست واژه منحرف می‌نماید.
هم	«هم» بر سر واژه‌ای که با «م» یا «الف» آغاز می‌شوند جـدا نوشته می‌شود.	اشـکال: ایـن بنـد نیـز بـر ابهـام جدانویسـی و پیوسـته‌نویسـی «هـم» می‌افزاید. نکته: واژه «هم» از جمله مواردی است کـه نیـاز جـدّی بـه تـدوین قواعد مشخص‌تر و ساده‌تری برای نگارش دارد. تدوین این قواعد برای اهداف یاد شده در فصل اول بسیار ضروری است.

ادامه‌ی جدول (۲-۱)

واژه/ نشانه	قاعده‌ی جاری	مشکل
می/ نمی	جدانویسی «می» و «نمی» از واژه‌ی بعد از خود.	اشکال: اگر ذکر نشود که این جدانویسی با نیم‌فاصله است یا فاصله، «می» در موارد زیادی می‌تواند «مِی» تلقی شود و «نمی» به صورت «نَمی» (اندکی رطوبت). در نتیجه تفسیر جمله کاملاً متفاوت خواهد شد.
ها	قاعده: موارد جدانویسی «ها».	اشکال: همان‌طور که موارد جدانویسی «ها» در دستورِ خطِ فارسی گویای آن است، تقریباً تمام این موارد برای رایانه ابهاماتی بسیار پیچیده خواهند داشت. حتی خود کاربران نیز به دلیل زیاد بودن این بندها، ترجیح می‌دهند «ها» را جدا بنویسند. به همین دلیل پیشنهاد می‌شود که جدانویسی «ها» برای یکی از قدم‌های بسیار موثر در کاهش ابهام واژه، به صورت ساده و صریح اعلام گردد. تشخیص جمع بودن یک واژه از روی مشاهده‌ی نشانه‌ی جمع «ها» برای رایانه، امکان پردازش‌های پیشرفته‌تری مانند تطابق دادن فعل با نهاد را در آینده فراهم می‌آورد. همچنین، تصحیح املای واژه‌ها را بسیار سریع و ساده می‌کند. نکته: استفاده از نیم‌فاصله باید به صورت صریح عنوان گردد.
«ام، ای، است» و ضمائر مفعولی	در مواردی که باید جدا نوشته شوند.	اشکال: فاصله‌گذاری میان اجزای واژه‌ای که به آن «ام» اضافه شده است حتماً باید مشخص شود که با فاصله است یا نیم‌فاصله. زیرا در صورت استفاده از فاصله ایجاد ابهام می‌نماید. برای فاصله‌گذاری فعل کمکی ماضی نقلی (ام، ای، است و ...) نیز باید قاعده‌ای تنظیم گردد.
کسره‌ی اضافه	نشانه‌ی کسره‌ی اضافه در واژه مختوم به «ه» غیر ملفوظ به شکل «ء» نوشته می‌شود.	اشکال: استفاده از «هٔ» یا «ی» هر دو به یک اندازه متداول است به ویژه که در زمان آموزش این نشانه در نظام آموزشی کشور، هر دو در دوره‌های متوالی به دانش‌آموزان تعلیم داده شده است و بی‌نظمی فراوانی در این مورد وجود دارد.
تنوین	نگارش تنوین	اشکال: همان‌طور که عنوان گردید، نوشتن تنوین الزامی است اما بسیاری از کاربران زبان به دلیل آن که نگارش آن نیازمند به فشردن یک کلید اضافه است که معمولاً جای آن را نمی‌دانند، تنها به درج‌کردن حرف «ا» قناعت می‌کنند. این مسئله باعث می‌شود که رایانه نتواند در موارد زیادی واژه تنوین‌دار را تشخیص دهد و در نتیجه متن حروف‌چینی شده توسط دو کاربر، مثل هم تعبیر نمی‌شود.
ترکیب‌های عربی		نکته: در نگارش ترکیب‌های چند واژه‌ایِ عربی، باید استفاده از فاصله یا نیم‌فاصله مشخص شود.

ادامه‌ی جدول (۲-۱)

واژه/ نشانه	قاعده‌ی جاری	مشکل
همزه	قواعد همزه	به خاطرسپاری قواعد نگارش همزه برای اکثر کاربران مسئله‌ای پیچیده است اگرچه این امر دلیل بر عدم لزوم به این قواعد نیست. تنها نکته‌ی موجود در مورد این قواعد آن است محدود شدن تنوع استفاده از همزه به دو یا سه گونه، کمک فراوانی به افزایش سرعت پردازش رایانه‌ای متون فارسی می‌نماید. اشکال: اگر کاربری کلاً به جای «أ»، «إ»، «ؤ»، «ء» و «ئ» از «ا»، «و» و «ی» استفاده کند (مثلاً به دلیل سختیِ به خاطرسپاری مکان حروف همزه در صفحه کلید)، آنگاه رایانه نمی‌تواند همزه‌دار بودن این واژه را حدس بزند و تصحیح کند. رفع این مشکل از رفع حالتی که کاربر شکل اشتباهی از همزه را به کار برده است پیچیده‌تر است.
تشدید	اگر عدم حضور تشدید منجر به بروز ابهام گردد، نوشتن آن الزامی است.	اشکال: بسیاری از کاربران در زمان حروف‌چینی، به ابهامِ به وجود آمده در معنای واژه توجه نمی‌کنند. از دیگر سو، وادار نمودن آن‌ها به درج نویسه‌ی تشدید مناسب نیست زیرا اکثراً جای صحیح آن را نمی‌دانند. البته خوشبختانه تشدید خیلی کم در متون ظاهر می‌شود.

جدول (۲-۱) شامل قواعدی بود که می‌توانستند برای رایانه ابهام ایجاد نمایند. گفتنی است که در تمام مواردی که جدانویسی مطرح می‌شود، باید حتماً عنوان گردد که این عمل با استفاده از فاصله است یا نیم‌فاصله.

تمام موارد فوق مربوط به تک‌واژه‌ها بود و منطقی است اگر فاصله‌گذاری میان اجزای آن‌ها با نیم‌فاصله انجام گردد. صراحت دستور استفاده از فاصله یا نیم‌فاصله به این دلیل اهمیت دارد که اگر میان دو واژه فاصله ظاهر شود، آن دو واژه می‌توانند در معنای مستقل خود تفسیر شوند. در این صورت معنایی که از کنار هم قرار گرفتن آن‌ها با نیم‌فاصله حاصل می‌شود از میان می‌رود. مثلاً اگر «می‌خورد» به صورت «می خورد» نوشته شود، معنای جمله‌ی زیر کاملاً متفاوت خواهد شد:

- «او می خورد» به این معنی که او قبلاً مِی (شراب) خورده است.
- «او می‌خورد.» به این معنی که او چیزی را می‌خورد یا هم اکنون در حال خوردن چیزی است.

۲-۴ ترکیب‌ها

آن‌چه تاکنون در مورد ابهامات قواعد نگارش واژه‌ها عنوان گردید، در مورد واژه‌های غیر مرکب بود در حالی که در این فصل قواعد ترکیب‌ها مرور خواهد شد. در این کتاب هیچ پیشنهاد قاطعی برای جدانویسی کامل یا پیوسته‌نویسی کامل ارائه نمی‌شود و تشخیص این مورد بر عهده‌ی استادان فرهنگستان زبان و ادب فارسی خواهد بود. آن‌چه در این متن به شدت مورد تأکید است، تدوین ***قواعد بدون ابهام*** در زمینه‌ی ترکیب‌ها است. قواعدی که چه دستور بر پیوسته‌نویسی دهند و چه جدانویسی، در حوزه‌ی خود ابهام‌زا نباشند. نویسندگان کتاب اذعان دارند که در مواردی، مشخص نمودن دستور دقیق فاصله‌گذاری بین واژگان ساده نخواهد بود. اما حتی در همان موارد نیز اگر بتوان حکمی قطعی صادر کرد که پردازش متن را ساده‌تر نماید کمک موثری به سامانه‌های رایانه‌ای شده است که آثار آن خیلی زود در صحنه‌ی فن‌آوری اطلاعات در کشور ظاهر خواهد شد.

۲-۴-۱ موارد مبهمِ پیوسته‌نویسی

جدول (۲-۲) نشان‌دهنده مواردی است که در پیوسته‌نویسی رایانه‌ای ابهام ایجاد می‌کنند.

جدول (۲-۲) موارد مبهمِ قواعد دستورِ خطِ فارسی در پیوسته‌نویسی

قاعده	مشکل
مرکّب‌هایی که بسیط‌گونه هستند مانند: آبرو، الفبا، آبشار، نیشکر، رختخواب، یکشنبه، پنجشنبه، سیصد، هفتصد، یکتا، بیستگانی	اشکال: رایانه نمی‌تواند بسیط‌گونه بودن یک واژه را تشخیص دهد. بنابراین، اگر این واژه‌ها به همین شکل در واژه‌نامه وجود نداشته باشند، رایانه نیز قادر به تشخیص خطا نخواهد بود. راه حل: تمام واژه‌های بسیط‌گونه‌ی مرکب، باید حتماً در واژه‌نامه ضبط شده باشد.
اگر جزء دوم واژه با «ا» شروع شود و تک‌هجایی باشد واژه پیوسته نوشته می‌شود.	اشکال: رایانه از تعداد هجاهای واژه‌ها بی‌اطلاع است. در نتیجه به هیچ‌وجه نمی‌تواند صحت واژه را تایید نماید مگر آن که عین واژه به صورت پیوسته در واژه‌نامه وجود داشته باشد. اشکال: اگر کاربران به اشتباه واژه را جدا نوشته باشند و اگر فاصله‌ی کامل میان اجزای آن باشد، رایانه هیچ راهی برای اصلاح آن نخواهد داشت. اگر نیم‌فاصله گذاشته باشند، رایانه قادر به تطبیق دادن این واژه با واژه‌ی درون واژه‌نامه نخواهد بود گرچه می‌تواند صحت آن را با نگارش جاری، تایید نماید.

ادامه جدول (۲-۲)

قاعده	مشکل
اگر جزء دوم با «ا» شروع شود و چند هجایی باشد، دست نویسنده باز است.	اشکال: در این صورت، رایانه تشخیص نخواهد داد که «دل‌آویز» همان «دلاویز» است که کاربری به صورت دوم نوشته است. این قاعده نیازمند بررسی مجدد و تصمیم‌گیری قطعی در مورد فاصله‌گذاری آن می‌باشد.
هرگاه در اجزای یک واژه‌ی مرکب، کاهش واجی روی داده باشد.	این نمونه نیز مانند واژه‌های بسیط‌گونه، نیازمند وجود واژه در واژه‌نامه است. با این تفاوت که به دلیل کاربرد واژه‌ی جدید، کم‌تر آن را به اشتباه جدا می‌نویسند یا حتی مرکب می‌دانند.
مرکبی که دست‌کم یک جزء آن کاربرد ندارد.	در این مورد نیز مانند مورد قبل اکثر کابران از پیوسته‌نویسی استفاده می‌کنند. اما توجه به این نکته ضروری است که چنین واژه‌هایی حتماً باید در واژه‌نامه وجود داشته باشند.
مرکب‌هایی که جدانوشتن آن‌ها ابهام به وجود آورد.	همان‌طور که عنوان شد، اگر تمام کاربران به این مسئله توجه کنند مشکلی پیش نخواهد آمد. اما اگر نویسنده‌ای اشتباهاً در این مورد جدانویسی را انتخاب کند، رایانه قادر به تشخیص حالت پیوسته نیست.
واژه مرکبی که جزء دوم آن‌ها تک‌هجایی باشد و جنبه‌ی سازمانی داشته باشند.	رایانه نمی‌تواند مفهوم واژه‌ها را تشخیص دهد. در نتیجه نمی‌تواند تشخیص دهد که یک واژه جنبه‌ی سازمانی دارد یا خیر. اگر این واژه به همین شکل در واژه‌نامه باشد، آن را تایید خواهد نمود. اما اگر کاربران زبان، رویکرد واحدی را در نگارش این واژه‌ها پیش نگیرند (که اکثراً نمی‌گیرند)، رایانه نمی‌تواند یکسان بودن دو واژه از این مجموعه را که یکی پیوسته و دیگری جدا نوشته شده است تشخیص دهد.

۲-۴-۲ موارد مبهمِ جدانویسی

جدول (۲-۳) نشان‌دهنده موارد مبهم در جدانویسی، در رایانه است.

جدول (۲-۳) ابهام‌های قواعد دستورِ خطِ فارسی در جدانویسی

قاعده	مشکل
واژه‌های پیشوندی همواره جدا نوشته می‌شود.	موارد استثناءای که برای «هم» و «به» و «بی» وجود دارد حتماً باید رفع ابهام شده باشند. نکته: نحوه‌ی فاصله‌گذاری این واژه‌ها باید مشخص شده باشد.

ادامه‌ی جدول (۲-۳)

قاعده	مشکل
واژه‌های پیشوندی همواره جدا نوشته می‌شود.	موارد استثناءای که برای «هم» و «به» و «بی» وجود دارد حتماً باید رفع ابهام شده باشند. نکته: نحوه‌ی فاصله‌گذاری این واژه‌ها باید مشخص شده باشد.
ترکیب‌های اضافی	اشکال: در این مورد که ترکیب‌های اضافی را اکثر کاربران جدا می‌نویسند مشکلی وجود ندارد. مشکل زمانی روی می‌دهد که باید از فاصله و نیم‌فاصله میان اجزای ترکیب استفاده نمود. از دید رایانه، «آب میوه» با «آب‌میوه» متفاوت است.
جزء دوم با الف آغاز شود.	همان‌طور که در یکی از بندهای پیوسته‌نویسی قاعده‌ای مشابه وجود داشت و دست کاربر در نگارش پیوسته و جدای ترکیب‌هایی که جزء دوم آن‌ها با الف آغاز می‌شود، آزاد گذاشته شده بود، این مورد نیز بسیار ابهام‌زاست. کاربران باید به طور قاطع بدانند که واژه‌ای مانند «دلاویز» یا «دل‌انگیز» باید جدا نوشته شوند یا پیوسته. پیشنهاد: در مورد این واژه‌ها، حالت جدا در نظر گرفته شود و بر استفاده از نیم فاصله نیز تأکید گردد. البته نظر استادان فرهنگستان در این زمینه حرف نهایی خواهد بود.
حرف پایانی جزء اول با حرف آغازین جزء دوم هم‌مخرج یا مشابه باشد.	اشکال: رایانه از هم‌مخرج بودن حروف بی‌اطلاع است. اگر کاربران بخواهند این واژگان را به هر روشی که مایل‌اند بنویسند، رایانه نه قادر به تصحیح و نه قادر به بازشناسی واژه‌های مشابه است. نکته: اگر فرض بر آن باشد که اکثر ترکیب‌ها جدا نوشته می‌شوند، این مشکل خود به خود مرتفع می‌گردد.
مرکب‌های اتباعی	اشکال: بسیاری از مرکب‌های اتباعی در واژه‌نامه وجود ندارند. در این صورت این قاعده باید به گونه‌ای جامع و بی‌ابهام باشد که هر کاربری که تصمیم به نگارش یک نمونه مرکب اتباعی گرفت، آن را مشابه با سایر کاربران بنویسد. نکته: استفاده از فاصله و نیم‌فاصله باید مشخص شده باشد.
هر یک از اجزای واژه‌ای مرکب، چند حرف مختوم داشته باشد.	اشکال: از دید رایانه، «پای‌برهنه» با «پابرهنه» متفاوت است و کاربرانی که یکی از این دو صورت را استفاده می‌کنند باید بپذیرند که ممکن است نتیجه‌ی پردازش متن آن‌ها تا حدی نادرست باشد.

ادامه‌ی جدول (۲-۳)

قاعده	مشکل
مرکب‌هایی که یک جزء آن واژه دخیل هستند.	اشکال: رایانه واژه‌های دخیل را نمی‌شناسد. در نتیجه دخیل بودن یا نبودن واژه‌ها باید برای تمام کاربران زبان محرز باشد تا همواره نگارش آن‌ها از چنین واژه‌هایی یکسان باشد. اشکال: تمام مرکب‌های اتباعی به ترکیب‌های دو واژه‌ای محدود نمی‌شوند. واژه‌هایی مانند «شِرّ و وِر»، «آش و لاش» نیز مرکب اتباعی هستند که درون خود از میانوند استفاده می‌نمایند. در این موارد وجود دستوری قاطع برای استفاده از نیم‌فاصله یا فاصله، امری مهم است زیرا به کار بردن فاصله، این مرکب‌های اتباعی را دچار ابهام در معنا می‌کند و منجر به آن می‌شود که جزء دوم (اتباع) در جمله یک خطای املایی در نظر گرفته شود. نکته: اگر فرض شود که پیش‌فرض نگارش ترکیب‌ها، جدانویسی است، این مورد نیز خود به خود مرتفع خواهد شد و تنها مسئله‌ی ابزار فاصله‌گذاری باقی خواهد ماند.
عبارت‌های عربی چند جزئی	استفاده از فاصله و نیم‌فاصله باید مشخص شده باشد. اشکال: کاربران فارسی زبان اکثراً واژه‌های عربی را به یک شکل نمی‌نویسند. به ویژه که عبارت‌های عربی می‌توانند در عین آن که یک مفهوم را می‌رسانند با حروف متفاوتی (ناشی از شرایط صَرفی متفاوت) نوشته شوند.
نگارش عدد یک (پیوسته و جدا)	اشکال: اگر «یک» توسط کاربران مختلف هم پیوسته و هم جدا نوشته شود، از دید رایانه دو واژه را تولید می‌نماید. ابهام میان این موارد باید بر طرف گردد زیرا در غیر این صورت تمام پردازش‌های بعدی روی این واژه‌ها دچار اختلال خواهد شد.
هرگاه با پیوسته‌نویسی اجزای ترکیب معلوم نشوند	اشکال: رایانه از اینکه در وضعیت خاصی ممکن است اجزای ترکیب معلوم نشوند بی‌خبر است. در نتیجه نمی‌تواند تشخیص دهد که «پاکنام» حالت خطادار واژه‌ی «پاک‌نام» است یا واژه‌ای است که خطا املایی دیگری دارد.
واژه با پیوسته‌نویسی نامأنوس شود.	اشکال: از دید خیلی از کاربران، واژه‌های «حقیقتجو» و ... نامأنوس نیستند در نتیجه بسیاری از آن‌ها این واژه‌ها را به یک حالت نمی‌نویسند. نکته: اگر پیش‌فرض نگارش واژه‌ها، جدانویسی باشد این مورد به سادگی مرتفع می‌گردد.

ادامه‌ی جدول (۲-۳)

قاعده	مشکل
یک جزء آن اسم خاص باشد	اشکال: همان‌طور که گفته شد، رایانه نمی‌تواند خاص بودن اسامی را تشخیص دهد مگر آن که در واژه‌نامه تعریف شده باشد. در غیر این صورت اگر کاربران توافقی برای نگارش واژه‌های مرکب با اسم خاص نداشته باشند، رایانه نیز قادر به تصحیح یا کشف یکسان بودن دو واژه نیست.
جزء آغازی یا پایانی آن بسیار پر بسامد باشد.	اشکال: رایانه نمی‌تواند بسامد واژه‌ها را تشخیص دهد. در نتیجه نمی‌تواند تشخیص دهد که «نیکبخت» همان «نیک‌بخت» بوده است. نکته: اگر پیش‌فرض نگارش واژه، جدانویسی باشد این مورد به سادگی مرتفع می‌گردد.

۲-۴-۳ ترکیب‌های اضافی

ترکیب‌های اضافی مواردی هستند که کاربران به دلیل آن که کسره‌ی میانی ترکیب را تلفظ می‌کنند، دو واژه را از هم جدا می‌نویسند. اما مشکل جایی ظاهر می‌شود که باید میان عناصر ترکیب فاصله‌ای درج نمود. اگر از فاصله استفاده شود (که در بسیاری از موارد چنین است)، باید قواعدی در مورد استفاده از نیم‌فاصله در مواردی مانند «سیب‌زمینی» وجود داشته باشد. این که به چه دلیل «سیب‌زمینی» به شکل «سیب زمینی» نوشته نمی‌شود باید برای تمام کاربران زبان کاملاً و بدون هیچ تردیدی مشخص شده باشد.

در بررسی‌های ساده‌ی اولیه چنین به نظر می‌آید که زمانی که کسره‌ی اضافه در ترکیب‌های اضافی تلفظ نمی‌شود (یعنی دو واژه به صورت متداول در کنار یکدیگر به کار رفته‌اند)، نیم‌فاصله به کار می‌رود. اگر این مورد صحیح است باید در دستور خط گنجانده شود تا کاربران این‌گونه واژه‌ها را به شکل‌های مختلف ننویسند و چهره‌ی خط همه جا یکسان باشد.

یکی از مسائل پیچیده در فاصله‌گذاری ترکیب‌ها، حالت‌هایی است که باید یک ترکیب، چندجزئی نوشته شود. این گونه تَرکیب‌ها اگر مانند یک واژه در نظر گرفته نشوند، معنایی متفاوت به وجود می‌آورند. مثلاً اگر «آب‌سردکن» به این صورت نوشته نشده باشد، خواننده باید چند لحظه درنگ کند تا تشخیص دهد که منظور از «آب سرد کن»، یک جمله‌ی امری نیست و یک دستگاه به نام آب‌سردکن است. ذهن انسان می‌تواند این تفاوت را تشخیص دهد زیرا به معنای جمله و نیز قواعد ساخت جمله آگاهی دارد. در

نتیجه می‌تواند بگوید «آب‌سردکن» را می‌توان با «ها» جمع بست. اما رایانه به هیچ روی نمی‌تواند تشخیص دهد که «آب سرد کن‌ها» یک عبارت درست است زیرا از دید قواعد پیاده‌سازی شده در رایانه، «ها» به فعل اضافه نمی‌شود. این در شرایطی است که رایانه بتواند تشخیص دهد «کن» یک فعل است که خود نیازمند پردازش پیچیده‌ی دیگری است.
اگر کاربری در موتور جستجوی گوگل، واژه «آب‌سردکن» را جستجو کند، نمی‌تواند صفحاتی راکه در آن‌ها واژه «آب سرد کن» نوشته شده است پیدا کند و به عکس. این مشکل در مورد تعداد زیادی از واژه‌ها مرکب فارسی رخ می‌دهد. برای یکسان‌سازی نگارش چنین واژه‌هایی سه رویکرد عمده وجود دارد:

- در رویکرد اول، در تمام موارد نیم‌فاصله استفاده می‌شود تا مرز واژه‌ها از یکدیگر مشخص باشد. اما در این حال، عبارت‌هایی مانند «پررفت‌وآمد» باید به چه شکلی نوشته شوند؟ آیا استفاده از نیم‌فاصله در این سطح جایز است؟

- در رویکرد دوم، در تمام موارد از فاصله استفاده می‌شود. اما در این روش، رایانه هیچ معیاری برای تعیین درستی «آب سرد کن ها» نخواهد داشت. این روش گرچه ساده است اما کیفیت پردازش‌های بعدی رایانه‌ای را به شدت کاهش می‌دهد.

- در رویکرد سوم، در مواردی که میان واژگان یک ترکیب، صدا (کسره‌ی اضافه) وجود ندارد، آن‌ها را با نیم‌فاصله و در غیر این صورت با فاصله می‌نویسیم. رویکرد سوم کاملاً نیازمند به قواعد دسته‌بندی شده برای بررسی دقیق‌تر انواع ترکیب‌ها خواهیم بود و این رویکرد احتمالاً به تدوین یک جدول برای بیان شرایط فاصله‌گذاری ترکیب‌های چند واژه‌ای (بیش از دو واژه) منجر می‌گردد.

در جدول (۲-۴) تعدادی از قواعد تولید واژه که منجر به تولید ترکیب‌های چندجزئی می‌شوند ذکر شده است. در ستون اول از این جدول قاعده تولید واژه عنوان شده است، در ستون دوم مثال ذکر گردیده و در ستون سوم نکته یا پیشنهادی مربوط به این قاعده آورده شده است.

جدول (۲-۴) برخی از قواعد تولید ترکیب‌های چندجزئی

قاعده	مثال	پیشنهاد
اسم/ضمیر + بن مضارع ← صفت فاعلی و نیز: اسم/ضمیر + بن مضارع + ی ← صفت	دل‌گشا، زودرنج، خویشتن‌دار، خاطره‌نویس	در تمام ترکیب‌هایی که در آن‌ها بن مضارع به کار رفته است، جزء قبل از بن مضارع باید با نیم‌فاصله در کنار بن مضارع قرار گیرد. در هیچ حالتی نیز پیوسته نوشته نمی‌شوند.
گروه وصفی ← صفت	آب‌سردکن	اگر بدون کسره اضافه به هم متصل شوند باید با نیم‌فاصله نوشته شوند.
صفت و موصوف مبهم + بن مضارع ← صفت فاعلی	همه‌چیزدان، هیچ‌چیزندان	اگر بدون کسره اضافه به هم متصل شوند باید با نیم‌فاصله نوشته شود. این گونه ترکیب‌ها در هیچ حالتی پیوسته نوشته نمی‌شوند.
صفت مفعولی + «شده» ← صفت مفعولی	گرفته شده، خوانده شده	چون در ساخت فعل‌های مجهول از این قاعده بسیار استفاده می‌شود و رسم نیست که اجزای گروه فعلی را با نیم‌فاصله کنار هم قرار دهند، بهتر است همه‌جا این مورد با فاصله‌ی تمام باشد.
عدد + و + عدد ← صفت شمارشی	بیست و یک	چون نوشتن ترکیب‌های عددی با نیم‌فاصله معمول نیست، بهتر است که این مورد همه‌جا با فاصله‌ی تمام نوشته شود.
صفت + ترکیب عطفی ← صفت	پررفت‌وآمد، با آب و رنگ، با شرم‌وحیا	کاملاً مبهم.
حرف اضافه + ضمیر + صفت ← صفت	از ما بهتران، از خود راضی، از خود بی‌خود	کاملاً مبهم.
حرف اضافه + متمم + صفت مفعولی ← صفت	به‌هم‌خورده، به‌هم‌ریخته	کاملاً مبهم.

ادامه‌ی جدول (۲-۴)

قاعده	مثال	پیشنهاد
ترکیب عطفی (اسم) + صفت ← صفت	دست‌ودل‌باز، دست و پا چلفتی	کاملاً مبهم.
ترکیب عطفی (اسم) + بن مضارع ← صفت	دست‌وپاگیر	کاملاً مبهم.
مرکب اتباعی که در آن‌ها تغییر در واج دوم و استفاده از میانوند وجود دارد.	مارچ‌ومورچ	اگر اجزای مرکب اتباعی با فاصله نوشته شوند، به دلیل بی‌معنی بودن جزء دوم، در رایانه برای یک خطا املایی تلقی خواهند شد. در نتیجه به هیچ وجه قابل بازیابی و تبدیل شدن به شکل اصلی آن نیستند. به همین دلیل بهتر است مرکب‌های اتباعی همواره با نیم‌فاصله از هم جدا شوند.
مرکب‌های اتباعی که بعد از حرف دوم یک «و» اضافه می‌شود.	هارت‌وهورت	مشابه مورد قبل (مرکب‌های اتباعی)
صفت منفی + و + صفت منفی ← صفت	بی‌بو و بی‌خاصیت	کاملاً مبهم.
فعل امر + و + فعل نهی ← صفت	کجداروم‌ریز، بخور و نمیر	کاملاً مبهم.
اسم + و + بن فعلِ هم‌معنی ← اسم مصدر	مرگ‌ومیر	کاملاً مبهم.
بن ماضی + و + بن مضارع ← اسم مصدر	گفت‌وگو، رفت‌وروب، جست‌وجو	کاملاً مبهم.

۲-۵ نتیجه‌گیری

با توجه به اهمیت زبان‌شناسی رایانه‌ای در دنیای امروز، نبود دستور خطِ جامعی متناسب با نیازهای سامانه‌های پردازش متن‌های رقمی مشکلات فراوانی به همراه خواهد داشت؛ بنا براین، توصیه می‌شود که ارتباطی متقابل میان زبان‌شناسان و فناوران حوزه‌ی رایانه برقرار گردد تا بتوان حوزه زبان‌شناسی رایانه‌ای را تقویت نمود. به طور خلاصه می‌توان اشکالات

دستورِ خطِ فارسی فرهنگستان زبان و ادب فارسی را چنین دسته‌بندی نمود:

- باز گذاشتن دست نویسندگان در فاصله‌گذاری میان واژه.
- نداشتن دستورالعمل قطعی برای استفاده از نیم‌فاصله.
- نداشتن قواعدی ثابت برای فاصله‌گذاری ترکیب‌ها؛ استفاده از دستورالعمل مبتنی بر واژه (مانند تک‌هجایی بودن، بسیط گونه بودن).

این کتاب با بیان برخی اشکالات اساسی دستورِ خطِ فارسی فرهنگستان زبان و ادب فارسی از دیدگاه زبان‌شناسی رایانه‌ای، پیشنهادهایی برای برطرف نمودن برخی از مشکلات دارد ولی حتی اگر تمام این اشکالات حل شوند و ابهام‌ها برطرف گردند، همچنان در زبان فارسی کاربرانی خواهند بود که واژه‌ها را خارج از این استاندارد می‌نویسند که بدون تحول‌های بنیادین در خط و زبان فارسی مشکلات عدم هم‌خوانی و چندگانگی در نوشتار همواره وجود خواهد داشت.

فصل سوم

چالش‌های خطایابی در زبان فارسی

۳-۱ مقدمه

هر زبانی ویژگی‌ها، محدودیت‌ها و توانایی‌های خاصی دارد. زبان فارسی واژک‌شناسی غنی و پیچیده‌ای دارد. کلمات در زبان فارسی می‌توانند با ترکیب‌های بسیار زیادی از پسوندها تصریف شوند. کلمات اشتقاقی بسیار زیادی نیز در زبان فارسی موجود هستند اما قوانین اشتقاق، تصریف و ترکیب دقیق و جامع نیست. بسیاری از حروف زبان فارسی هم‌آوا[۱] و هم‌شکل[۲] هستند و فرایند املاء و در نتیجه خطایابی را با مشکل مواجه می‌سازند.

چالش‌ها و اشکالات دستورِ خطِ فارسی در فصل دوم مورد بررسی قرار گرفتند، اما شاید مهم‌ترین چالش زبان فارسی در فاصله‌گذاری میان ترکیب‌ها باشد. زبان فارسی علاوه بر فاصله‌گذاری معمول در دیگر زبان‌ها، فاصله‌ی درون واژه‌ای نیز دارد که قوانین مشخص و دقیقی جهت نحوه‌ی فاصله‌گذاری موجود نیست. همه‌ی این خصوصیات بر خطایابی املایی در زبان فارسی تأثیر دارند و باید مورد توجه و مطالعه قرار گیرند.

در این فصل همچنین مروری بر نحوه‌ی اتصال تکواژها به یکدیگر در زبان فارسی خواهیم داشت و پیش از آن نیز عناصر تشکیل دهنده‌ی ترکیب‌ها در زبان فارسی را از نظر خواهیم گذراند. تاکید عمده در متن حاضر، اتصال انواع تکواژهایی است که به طور معمول به واژه متصل می‌گردند (تکواژهای تصریفی)، مانند نشانه‌ی جمعِ «ها». تا زمانی که نتوان بخش‌های غیر اصلی واژه را استخراج نمود، تشخیص درستی املای یک واژه ممکن نیست. در واقع اگر بدانیم چه تکواژهایی به واژه اضافه شده‌اند که در معنای آن تاثیری نداشته‌اند می‌توانیم با حذف آن‌ها اصل واژه را از نظر درستی مورد بررسی قرار دهیم. برای مثال، علائم جمع در معنای واژه تغییری به وجود نمی‌آورد اما باعث می‌شوند تا ظاهر واژه

---

۱ معادل فارسی واژه‌ی انگلیسی Homophone

۲ معادل فارسی واژه‌ی انگلیسی Homoshape

از حالتی که در واژه‌نامه تعریف شده متفاوت گردد. به جز نشانه‌های جمع، تکواژهای دیگری مانند «ی» نکره و کسره‌ی اضافه وجود دارد که به دلیل ساختار جمله، به واژه متصل شده‌اند. به جز موارد ذکر شده، تکواژهای مستقل (تکواژهای اشتقاقی) نیز از دیگر اجزای متصل شونده به یک واژه هستند. این تکواژها در معنای واژه تغییر به وجود می‌آورند. تکواژهای اشتقاقی را به هر واژه‌ای نمی‌توان متصل نمود و برای انجام اتصال، باید شکل جدید واژه در زبان کاربرد داشته باشد. در نتیجه، با قواعد تولید واژه‌ی مرکب از طریق ترکیب با تکواژهای وابسته، نباید واژه‌ای به وجود آید که هنوز در زبان وارد نشده است. یعنی چون مثلاً تکواژ «گار» به بن مضارع متصل می‌شود، نمی‌توان ادعا کرد که «خورگار» یک واژه‌ی معتبر در زبان فارسی است.

بنابراین، به مرجعی جامع برای مجموعه‌ی واژه‌های معتبر فارسی در تشخیص صحت اتصال یک تکواژ وابسته به یک واژه، نیاز خواهد بود. اما از آن جایی که در واژه‌نامه‌هایی مانند لغت‌نامه‌ی دهخدا (که نسبتاً جامع هستند) نیز هنوز برخی از واژه‌های زبان وارد نشده‌اند (مانند واژه «نگریسته» و «بازنگری»)، در نتیجه، نمی‌توان برای تصحیح خطا، فقط بر واژه‌نامه متکی بود همچنان که نمی‌توان تأکید تام بر قواعد زبان داشت. در ادامه اجزاء تشکیل دهنده‌ی ترکیب‌ها در زبان فارسی مورد بررسی قرار خواهند گرفت.

۳-۱-۱ عناصر تشکیل‌دهنده‌ی ترکیب‌ها در فارسی

پیش از شرح واژه‌ها مرکب در فارسی لازم است عناصری که می‌توانند در ساخت واژه ظاهر شوند مرور شود. این عناصر در جدول (۳-۱) نشان داده شده‌اند.

جدول (۳-۱) عناصر تشکیل‌دهنده‌ی ترکیب‌ها در فارسی

عنصر	شرح	مثال
اسم	اسم، واژه‌ای است که می‌تواند به طور مستقیم برای نامیدن یک شخص، حیوان، چیز و یا مفهوم به کار رود.	کتاب، اسب، تهران
صفت	صفت، حالت و مقدار و شماره و یا یکی دیگر از ویژگی‌های اسم را می‌رساند.	خوب، کارگر، پسندیده، آموزگار
اسم	اسم، واژه‌ای است که می‌تواند به طور مستقیم برای نامیدن یک شخص، حیوان، چیز و یا مفهوم به کار رود.	کتاب، اسب، تهران
بن فعل	در اینجا منظور از بن فعل، بن‌های ماضی یا مضارع هستند.	خور، گفت، رفت

ادامه جدول (۳-۱)

عنصر	شرح	مثال
صفت	صفت، حالت و مقدار و شماره و یا یکی دیگر از ویژگی‌های اسم را می‌رساند.	خوب، کارگر، پسندیده، آموزگار
بن فعل	در اینجا منظور از بن فعل، بن‌های ماضی یا مضارع هستند.	خور، گفت، رفت
فعل	فعل، واژه یا گروه واژگانی است که به چهار مفهوم دلالت می‌کند: شخص، شمار، زمان و یکی از موارد زیر: «انجام دادن یا انجام گرفتن کاری»، «واقع شدن کار بر کسی یا چیزی»، «پذیرفتن حالتی یا صفتی»، «اسناد» (یعنی نسبت دادن صفتی یا حالتی بر چیزی)، «وجود داشتن»، «مالکیت و دارا بودن چیزی».	رفتم، خورده شده است، می‌گویم، دارم می‌آیم
مصدر	اسمی است که فعل از آن مشتق می‌شود.	گفتن، خوردن
فعل امر	فعل امرِ یک مصدر، دستوری برای انجام آن فعل است به صیغه دوم شخص مفرد. این فعل با «ب» شروع می‌شود.	بگو
بن ماضی	مصدر با حذف حرف «ن» ازپایان آن.	گفت
بن مضارع	فعل امر با حذف «ب» از اول آن.	خور، گو
قید	قید، واژه یا گروه واژگانی است که مفهومی به مفهوم فعل و نیز گاهی به مفهوم صفت یا مسند یا قید دیگر و یا مصدر می‌افزاید و توضیحی در مورد آن‌ها می‌دهد و آن‌ها را با آن «مفهوم جدید» مقید می‌کند.	هرگز، مثلاً، اتفاقاً، به طوری که
ضمیر	ضمیر واژه‌ای است که به جای اسم یا گروه اسمی می‌نشیند یا به شخصی یا چیزی در عالم خارج اشاره می‌کند.	من، خود، خویشتن، شما، تان
ضمیر مفعولی	ضمیرهای شخصی ضمیرهایی هستند که بر اشخاص دلالت می‌کنند و به صورت پیوسته نوشته می‌شوند مانند «من، تو، او، ما، شما، ایشان» و «م، ت، ش، مان، تان، شان».	م، ت، ش، مان، تان، شان
وند	تکواژی وابسته که بدون تکواژ اصلی، معنادار نیست. منظور از «وند» در متن حاضر هر تکواژی است که می‌تواند پیشوند، پسوند یا میانوند باشد.	گار، ه، ی، هم، بر، فرا
پسوند	تکواژ وابسته‌ای است که به انتهای یک تکواژ مستقل متصل می‌شود و معنای آن را عوض می‌کند.	گار، ه، ی
پیشوند	تکواژ وابسته‌ای است که در ابتدای یک تکواژ مستقل قرار می‌گیرد و معنای آن را عوض می‌کند.	هم، بر، فرا

ادامه جدول (۳-۱)

عنصر	شرح	مثال
میانوند	تکواژ وابسته‌ای که بین دو یا چند تکواژ مستقل واقع می‌شود و معنای آن‌ها را عوض می‌کند.	و، ا
علامت جمع	نشانه‌ای است که بر جمع بودن واژه دلالت می‌کند.	ان، ها، گان
نشانه نکره	«ی» نشانه‌ی نکره است و در موارد معدودی می‌تواند به صورت «ای» یا «ئی» نوشته شود.	ی، ای
کسره اضافه	صوتی است که برای اتصال دو واژه در ترکیب‌های اضافی یا وصفی میان آن‌ها قرار می‌گیرد و بعد از واژه اول تلفظ می‌شود گاهی هم به صورت «ی» در انتهای واژه نوشته می‌شود.	ی
فعل‌های اسنادی	فعل «استن» کاربردها و نیز استثنائات فراوانی در زبان فارسی دارد. در این متن منظور از این فعل،حالت مخفف شده‌ی آن است.	م، ی، ست، یم، ید، ند

۳-۲ واژک‌شناسی

زبان فارسی دارای واژک‌شناسی پیچیده و در برخی موارد مبهم در قواعد تصریف فعلی، تصریف اسمی و قواعد ترکیب و فاصله‌گذاری است. نبود مجموعه‌ی دقیق و قطعی از قوانین، همچنین وجود موارد استثناءِ بی‌شمار، صَرف و ریشه‌یابیِ فعل‌ها و واژه‌ها را مشکل می‌سازد. به عنوان مثال بخشی از یک فعل می‌تواند با کلمات غیر فعلی ترکیب شده و از جزء دیگر فاصله گیرد. به عنوان نمونه فعل «اجازه دادن» می‌توانند به صورت «اجازه‌ی صحبت دادن» به کار رود که میان دو جزء فعل در جمله فاصله افتاده است.

زبان فارسی شامل تعداد بسیار زیادی وند، خصوصاً پسوند، است که این وندها می‌توانند با یکدیگر نیز ترکیب شوند. وندهای زبان فارسی اشتقاقی یا تصریفی هستند. وندهای اشتقاقی آن دسته از وندها هستند که پس از ترکیب با ریشه[۱]، واژه‌ای با معنای متفاوت ایجاد می‌کنند. به عنوان نمونه، واژه‌های «دانشگاه»، «دانشجو»، «دانشمند» و «دانش‌آموز» همگی از اشتقاق ریشه‌ی «دانش» ساخته شده‌اند که معانیِ متفاوتی با یکدیگر، همچنین نسبت به واژه‌ی «دانش» دارند. وندهای تصریفی آن دسته از وندها هستند که پس از ترکیب با ریشه، معنی آن را عوض نمی‌کنند و تنها آن را از نظر شخص، شمار و مواردی

۱ معادل فارسی واژه‌ی انگلیسی Lemma

از این دست صرف می‌کنند. به عنوان نمونه، پسوند تصریفیِ نشانه‌ی جمع «ها» پس از ترکیب با واژه‌ی «کتاب»، واژه‌ی «کتاب‌ها» را با همان معنی می‌سازد. ترکیب پسوندها، در برخی موارد، توسط قوانین واج‌شناسیِ تکواژها[۱] نیز مورد تأثیر قرار می‌گیرد. به عنوان نمونه پسوند تصریفیِ ضمیر ملکیِ *اول شخص مفرد*، می‌تواند به صورت «م» در واژه‌ی «کتابم»، به صورت «ام» در واژه‌ی «خانه‌ام» و به صورت «یم» در واژه‌ی «خدایم» تغییر آوا و نوشتار دهد. زبان فارسی بیش از ۲۷۰۰ ترکیب مختلف از پسوندهای تصریفی دارد که این تعدادِ بسیار زیادِ پسوندها، محققان را ناگذیر به ریشه‌یابیِ[۲] تصریفی واژه‌ها می‌سازد.

۳-۲-۱ صَرف واژه‌های غیر فعلی

مبحث ترکیب وندها با اسامی در زبان فارسی، به علت تعدد وندها، یکی از اشکالات جدیِ واژک‌شناسیِ زبان فارسی است که هیچ گاه به طور جدی و بایسته به آن پرداخته نشده است. زبان فارسی شامل پیشوند[۳]، میانوند[۴] و پسوند است. در این میان، میانوندها نقش تصریفی ندارند و هنگام ترکیب با واژه‌ها، واژه‌های جدید با معانی متفاوتی می‌سازند که در چنین شرایطی ریشه‌یابی مطرح نخواهد شد. پیشوندهای تصریفی در زبان فارسی تعداد قابل توجهی نیستند، با یکدیگر نیز ترکیب نمی‌شوند، همچنین به صورت بی‌قاعده، تنها با برخی واژه‌های خاص ترکیب می‌شوند؛ از این رو بهتر است، با توجه به تعداد نسبتاً کم ترکیب‌ها، به جای نگهداری ریشه و تصریف آن (ریشه‌یابی واژه‌ی ترکیبی)، هر ترکیب به صورت یک واژه‌ی مستقل نگهداری شود. این حالت میزان ابهام و خطا را نیز کاهش می‌دهد. اما چالش جدی زبان فارسی در پسوندهای تصریفی، ترکیب داخلی آن‌ها و ترکیب آن‌ها با واژه‌ها است.

پسوندهای تصریفی زبان فارسی به اختصار شامل ۱) نشانه‌ی جمعِ «ها»، ۲) نشانه‌ی جمع «ان»، ۳) ضمایر مِلکی و مفعولی، ۴) فعل‌های اسنادی، ۵) «ی» نسبت، ۶) «ی» نکره، ۷) «ی» بدل از کسره، ۸) صفات تفصیلی، ۹) صفات ترتیبیِ شمارشی، و ۱۰) صفت ترتیبیِ مبهم هستند. این پسوندها می‌توانند توسط قوانین آواشناختی مورد تأثیر قرار گیرند، با

---

۱ معادل فارسی واژه‌ی انگلیسی Morphophonology

۲ معادل فارسی واژه‌ی انگلیسی Lemmatization

۳ معادل فارسی واژه‌ی انگلیسی Prefix

۴ معادل فارسی واژه‌ی انگلیسی Infix

پسوندهای دیگر ترکیب شوند و حتی شکل ریشه را نیز تحت تأثیر قرار دهند. مسأله‌ی تصریف واژه‌های غیر فعلی توسط پسوندها هنگامی پیچیده‌تر می‌شود که پسوندها، بسته به نقش ادات سخنِ[۱] واژه‌ها با آن‌ها ترکیب می‌شوند و به عنوان مثال کلیه‌ی واژه‌های غیر فعلی زبان فارسی پسوند نشانه‌ی جمعِ «ها» نمی‌پذیرند. گونه‌های ادات سخن در بالاترین سطح (سطحی که برای خطایابی مورد نیاز است) شامل ۱) اسم که واحدهای اندازه‌گیری[۲] را نیز شامل می‌شود، ۲) فعل، ۳) قید[۳]، ۴) صفت[۴]، ۵) حرف که حروف اضافه[۵]، حروف صوت[۶] و حروف ربط[۷] را در بر می‌گیرد، ۶) ضمیر[۸] که حروف تعریف[۹] را نیز در بر می‌گیرد، و ۷) عدد[۱۰] هستند. در ادامه به بررسی دقیق‌تر ویژگی‌های وندهای تصریفی (پسوندهای تصریفی) و قواعد ترکیب آن‌ها با یکدیگر می‌پردازیم، همچنین پسوندهای دیگر که عموماً اشتقاقی هستند، میانوندها و پیشوندها را نیز به طور خلاصه مرور خواهیم کرد.

۳-۲-۱-۱ *وندهای تصریفی*

این بخش به بررسی ویژگی‌های وندهای تصریفی، که عموماً پسوند هستند، شامل، ادات سخنِ واژه‌هایی که پسوند تصریفیِ مورد نظر را می‌پذیرند، تغییر شکل پسوند تصریفیِ مورد نظر بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت، و نوع فاصله‌گذاری پسوند تصریفیِ مورد نظر هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت اختصاص یافته است.

---

۱ معادل فارسی واژه‌ی انگلیسی Part of Speech
۲ معادل فارسی واژه‌ی انگلیسی Scale
۳ معادل فارسی واژه‌ی انگلیسی Adverb
۴ معادل فارسی واژه‌ی انگلیسی Adjective
۵ معادل فارسی واژه‌ی انگلیسی Preposition
۶ معادل فارسی واژه‌ی انگلیسی Interjection
۷ معادل فارسی واژه‌ی انگلیسی Conjunction
۸ معادل فارسی واژه‌ی انگلیسی Pronoun
۹ معادل فارسی واژه‌ی انگلیسی Determiner
۱۰ معادل فارسی واژه‌ی انگلیسی Number

۳-۲-۱-۱-۱ *پسوندهای تصریفی*

در این بخش ویژگی‌های پسوندهای تصریفیِ زبان فارسی شامل نشانه‌ی جمعِ «ها»، نشانه‌ی جمع «ان»، ضمایر مِلکی و مفعولی، فعل‌های اسنادی، «ی» نسبت، «ی» نکره، «ی» بدل از کسره، صفات تفصیلی، صفات ترتیبیِ شمارشی، و صفت ترتیبیِ مبهم مورد بررسی قرار خواهند گرفت.

p نشانه‌ی جمعِ «ها»

نشانه‌ی جمع «ها» تقریباً به تمام اسم‌ها و نیز صفت‌های جانشین اسم در فارسی متصل می‌شود. موارد خاص شامل ۱) اسم‌های معنی (اسم‌هایی که بار معنایی و نه فیزیکی دارند)، مانند هوش، عقل، پرهیزکاری، اراده، شجاعت، بخشش، و نفرت (گناه از جمله اسم‌های معنی است که با «ان» نیز جمع بسته می‌شود)، ۲) اسم جمادات، مانند میز و کتاب، ۳) واژه‌های دخیل، و ۴) اسم رستنی‌ها مانند «درخت»، با «ها» جمع بسته می‌شوند. جدول (۳-۲) ویژگی‌های پسوند تصریفیِ نشانه‌ی جمعِ «ها» را نشان می‌دهد.

جدول (۳-۲) ویژگی‌های پسوند تصریفیِ نشانه‌ی جمعِ «ها»

	ادات سخنِ واژه‌هایی که پسوند تصریفیِ نشانه‌ی جمعِ «ها» می‌پذیرند		تغییر شکل پسوند تصریفیِ نشانه‌ی جمعِ «ها» بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوند تصریفیِ نشانه‌ی جمعِ «ها» هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	بدون تغییر	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	بدون تغییر	**مختوم به «ی»**	نیم‌فاصله
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	بدون تغییر	**مختوم به «ه» غیر ملفوظ**	نیم‌فاصله
ضمیر	می‌پذیرد				
صفت	نمی‌پذیرد	**مختوم به «و» غیر ملفوظ**	بدون تغییر	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	نیم‌فاصله
توضیحات	پسوند تصریفیِ نشانه‌ی جمعِ «ها» برای جمع بستن جاندار و غیر جاندار به کار می‌رود. هنگام ترکیب با کلمات مختوم به صامتِ منفصل («د»، «ذ»، «ر»، «ز»، «ژ»، «و»)، بدون فاصله می‌آید.					

p نشانه‌ی جمعِ «ان»

موارد زیر از جمله کاربردهای این نشانه‌ی جمع است که البته از «ها» کم‌کاربردتر است. جدول (۳-۳) ویژگی‌های پسوند تصریفیِ نشانه‌ی جمعِ «ان» را نشان می‌دهد.

جدول (۳-۳) ویژگی‌های پسوند تصریفیِ نشانه‌ی جمعِ «ان»

	ادات سخنِ واژه‌هایی که پسوند تصریفیِ نشانه‌ی جمعِ «ان» می‌پذیرند		تغییر شکل پسوند تصریفیِ نشانه‌ی جمعِ «ان» بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوند تصریفیِ نشانه‌ی جمعِ «ان» هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	یان	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	بدون تغییر	**مختوم به «ی»**	پیوسته
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	گان	**مختوم به «ه» غیر ملفوظ**	پیوسته
ضمیر	می‌پذیرد				
صفت	نمی‌پذیرد	**مختوم به «و» غیر ملفوظ**	یان	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	پیوسته
توضیحات	پسوند تصریفیِ نشانه‌ی جمعِ «ان» برای جمع بستن جاندارها به کار می‌رود. هنگام تصریف واژه‌های مختوم به «ه» غیر ملفوظ، حرف «ه» از انتهای واژه حذف می‌شود، مانند «پرنده» + «ان» ß «پرندگان»					

p ضمایر مِلکی و مفعولی

پسوندهای تصریفیِ ضمایر مِلکی و مفعولی در شکل معمول، شامل «م»، «ت»، «ش»، «مان»، «تان»، «شان» هستند. این ضمایر به واژه‌هایی که نشانه‌ی نکره دارند اضافه نمی‌شوند زیرا واژه را معرفه می‌نمایند. جدول (۳-۴) ویژگی‌های پسوندهای تصریفیِ ضمایر مِلکی و مفعولی را نشان می‌دهد.

جدول (۳-۴) ویژگی‌های پسوندهای تصریفیِ ضمایر مِلکی و مفعولی

ادات سخنِ واژه‌هایی که پسوندهای تصریفیِ ضمایر مِلکی و مفعولی می‌پذیرند		تغییر شکل پسوندهای تصریفیِ ضمایر مِلکی و مفعولی بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوندهای تصریفیِ ضمایر مِلکی و مفعولی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	یم، یت، یش، یمان، یتان، یشان	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	ام، ات، اش، مان، تان، شان	**مختوم به «ی»**	پیوسته
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	ام، ات، اش، مان، تان، شان	**مختوم به «ه» غیر ملفوظ**	نیم‌فاصله
ضمیر	می‌پذیرد				
صفت	می‌پذیرد	**مختوم به «و» غیر ملفوظ**	یم، یی، یش، یمان، یان، یشان	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	پیوسته
توضیحات	هنگامی که «و» صدای /o/ می‌دهد، مانند واژه‌ی «جلو»، ضمایر مِلکی و مفعولی به صورت «ام، ات، اش، مان، تان، شان» مورد استفاده قرار خواهند گرفت و هنگامی که صدای /u/ می‌دهد، مانند واژه‌ی «دانشجو»، به صورت « یم، یی، یش، یمان، یان، یشان» مورد استفاده قرار خواهند گرفت.				

p فعل‌های اسنادی

پسوندهای تصریفیِ فعل‌های اسنادی در شکل معمول، شامل «م»، «ی»، «است»، «یم»، «ید»، «ند» هستند، جدول (۳-۵) ویژگی‌های پسوندهای تصریفیِ فعل‌های اسنادی را نشان می‌دهد.

جدول (۳-۵) ویژگی‌های پسوندهای تصریفیِ فعل‌های اسنادی

ادات سخنِ واژه‌هایی که پسوندهای تصریفیِ فعل‌های اسنادی می‌پذیرند		تغییر شکل پسوندهای تصریفیِ فعل‌های اسنادی بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوندهای تصریفیِ فعل‌های اسنادی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	یم، یی، ست، ییم، یید، یند	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	ام، ای، است، ایم، اید، اند	**مختوم به «ی»**	نیم‌فصله
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	ام، ای، است، ایم، اید، اند	**مختوم به «ه» غیر ملفوظ**	نیم‌فصله
ضمیر	می‌پذیرد				
صفت	می‌پذیرد	**مختوم به «و» غیر ملفوظ**	ام، ای، است، ایم، اید، اند	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	پیوسته
توضیحات	در مواردی که فعل اسنادی در صیغه‌ی سوم شخص مفرد، به صورت «است» می‌آید، پسوند باید با واژه‌ی قبلی خود فاصله‌ی کامل داشته باشد.				

p «ی» نسبت

جدول (۳-۶) ویژگی‌های پسوند تصریفیِ «ی» نسبت را نشان می‌دهد.

جدول (۳-۶) ویژگی‌های پسوند تصریفیِ «ی» نسبت

ادات سخنِ واژه‌هایی که پسوند تصریفیِ «ی» نسبت می‌پذیرند		تغییر شکل پسوند تصریفیِ «ی» نسبت بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوند تصریفیِ «ی» نسبت هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	یی	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	ای	**مختوم به «ی»**	نیم‌فصله
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	ای	**مختوم به «ه» غیر ملفوظ**	نیم‌فصله
ضمیر	می‌پذیرد				
صفت	می‌پذیرد	**مختوم به «و» غیر ملفوظ**	یی	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	پیوسته

p «ی» نکره

پسوند تصریفیِ «ی» نکره برای نکره کردن یک اسم به کار می‌رود. جدول (۳-۷) ویژگی‌های پسوند تصریفیِ «ی» نکره را نشان می‌دهد.

جدول (۳-۷) ویژگی‌های پسوند تصریفیِ «ی» نکره

ادات سخنِ واژه‌هایی که پسوند تصریفیِ «ی» نکره می‌پذیرند		تغییر شکل پسوند تصریفیِ «ی» نکره بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوند تصریفیِ «ی» نکره هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	یی	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	ای	**مختوم به «ی»**	نیم‌فصله
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	ای	**مختوم به «ه» غیر ملفوظ**	نیم‌فصله
ضمیر	می‌پذیرد				
صفت	می‌پذیرد	**مختوم به «و» غیر ملفوظ**	یی	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	پیوسته

p «ی» بدل از کسره

«ی» بدل از کسره‌ی اضافه در واژه‌هایی که به صدای «ـِ»، مانند «خانه» ختم می‌شوند، جایگزین کسره‌ی اضافه در حالت مضاف می‌شود، مانند «خانه‌ی کوچک». جدول (۳-۸) ویژگی‌های «ی» بدل از کسره را نشان می‌دهد. نکته‌ی قابل ذکر آن است که نشانه‌ی نکره و کسره‌ی اضافه در یک واژه هم‌زمان ظاهر نمی‌شوند زیرا کسره‌ی اضافه واژه را معرفه می‌کند.

جدول (۳-۸) ویژگی‌های «ی» بدل از کسره

ادات سخنِ واژه‌هایی که «ی» بدل از کسره می‌پذیرند		تغییر شکل «ی» بدل از کسره بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری «ی» بدل از کسره هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	می‌پذیرد	**مختوم به «ا»**	بدون تغییر	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	نمی‌آید	**مختوم به «ی»**	
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	بدون تغییر	**مختوم به «ه» غیر ملفوظ**	نیم‌فصله
ضمیر	می‌پذیرد				
صفت	می‌پذیرد	**مختوم به «و» غیر ملفوظ**	بدون تغییر	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	نمی‌آید	**مختوم به حروف صامت**	

p صفات تفصیلی

پسوندهای تصریفیِ صفات تفصیلی شامل «تر» و «ترین» هستند. جدول (۳-۹) ویژگی‌های پسوند صفات تفصیلی را نشان می‌دهد.

جدول (۳-۹) ویژگی‌های پسوندهای تصریفیِ صفات تفصیلی

ادات سخنِ واژه‌هایی که پسوندهای تصریفیِ صفات تفصیلی می‌پذیرند		تغییر شکل پسوندهای تصریفیِ صفات تفصیلی بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوندهای تصریفیِ صفات تفصیلی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	نمی‌پذیرد	**مختوم به «ا»**	بدون تغییر	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	بدون تغییر	**مختوم به «ی»**	نیم‌فصله
حرف	نمی‌پذیرد				
		مختوم به «ه» غیر ملفوظ	بدون تغییر	**مختوم به «ه» غیر ملفوظ**	نیم‌فصله
ضمیر	نمی‌پذیرد				
صفت	می‌پذیرد				
		مختوم به «و» غیر ملفوظ	بدون تغییر	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	نمی‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	نیم‌فصله
توضیحات	هنگام ترکیب با واژه‌های مختوم به صامتِ منفصل («د»، «ذ»، «ر»، «ز»، «ژ»، «و»)، بدون فاصله می‌آیند.				

p صفات ترتیبیِ شمارشی

پسوندهای تصریفیِ صفات ترتیبیِ شمارشی شامل «ـُم» و «ـُمین» هستند. جدول (۳-۱۰) ویژگی‌های پسوندهای تصریفیِ صفات ترتیبیِ شمارشی را نشان می‌دهد.

جدول (۳-۱۰) ویژگی‌های پسوندهای تصریفیِ ترتیبیِ شمارشی

ادات سخنِ واژه‌هایی که پسوندهای تصریفیِ صفات ترتیبیِ شمارشی می‌پذیرند		تغییر شکل پسوندهای تصریفیِ صفات ترتیبیِ شمارشی بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوندهای تصریفیِ صفات ترتیبیِ شمارشی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	نمی‌پذیرد	**مختوم به «ا»**	بدون تغییر	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	بدون تغییر	**مختوم به «ی»**	پیوسته
حرف	نمی‌پذیرد				
		مختوم به «ه» غیر ملفوظ	بدون تغییر	**مختوم به «ه» غیر ملفوظ**	پیوسته
ضمیر	نمی‌پذیرد				
صفت	نمی‌پذیرد				
		مختوم به «و» غیر ملفوظ	بدون تغییر	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	می‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	پیوسته

p صفت ترتیبیِ مبهم

پسوند تصریفیِ صفت ترتیبیِ مبهم به صورت «گانه» می‌آید. جدول (۳-۱۱) ویژگی‌های پسوند تصریفیِ صفت ترتیبیِ مبهم را نشان می‌دهد.

جدول (۳-۱۱) ویژگی‌های پسوند تصریفیِ ترتیبیِ شمارشی

ادات سخنِ واژه‌هایی که پسوند تصریفیِ صفات ترتیبیِ مبهم می‌پذیرند		تغییر شکل پسوند تصریفیِ صفات ترتیبیِ مبهم بر اساس قواعد آواشناختی هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت		نوع فاصله‌گذاری پسوند تصریفیِ صفات ترتیبیِ مبهم هنگام تصریف واژه‌های مختوم به حروف صامت و مصوت	
اسم	نمی‌پذیرد	**مختوم به «ا»**	بدون تغییر	**مختوم به «ا»**	پیوسته
فعل	نمی‌پذیرد	**مختوم به «ی»**	بدون تغییر	**مختوم به «ی»**	نیم‌فصله
حرف	نمی‌پذیرد	**مختوم به «ه» غیر ملفوظ**	بدون تغییر	**مختوم به «ه» غیر ملفوظ**	نیم‌فصله
ضمیر	نمی‌پذیرد				
صفت	نمی‌پذیرد	**مختوم به «و» غیر ملفوظ**	بدون تغییر	**مختوم به «و» غیر ملفوظ**	پیوسته
عدد	می‌پذیرد				
قید	نمی‌پذیرد	**مختوم به حروف صامت**	بدون تغییر	**مختوم به حروف صامت**	نیم‌فصله
توضیحات	هنگام ترکیب با واژه‌های مختوم به صامتِ منفصل («د»، «ذ»، «ر»، «ز»، «ژ»، «و»)، بدون فاصله می‌آید.				

۳-۲-۱-۱-۲ *ترکیب پسوندهای تصریفی با یکدیگر*

در این بخش چگونگی اتصال و ترکیب پسوندهای تصریفی با یکدیگر خواهیم مورد بررسی قرار خواهند گرفت. جهت مشخص کردن قواعد اتصال، از نشانه‌گذاری BNF[1] [5] استفاده شده است؛ در این نشانه‌گذاری عبارت سمت راست، امکان ترکیب شدن با عبارات سمت چپ را دارد.

<اسم> ::= <نشانه‌ی جمع «ها»> | <نشانه‌ی جمع «ان»> | <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نسبت> | <«ی» نکره> | <«ی» بدل از کسره>

<ضمیر> ::= <نشانه‌ی جمع «ها»> | <نشانه‌ی جمع «ان»> | <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نسبت> | <«ی» نکره> | <«ی» بدل از کسره>

<صفت> ::= <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نسبت> | <«ی» نکره> | <«ی» بدل از کسره> | <صفات تفصیلی>

<عدد> ::= <صفات ترتیبیِ شمارشی> | <صفت ترتیبیِ مبهم>

<ضمایر مِلکی و مفعولی> ::= <فعل‌های اسنادی>

<نشانه‌ی جمع «ها»> ::= <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نکره> | <«ی» بدل از کسره>

<نشانه‌ی جمع «ان»> ::= <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نکره>

<صفات تفصیلی> ::= <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نکره> | <نشانه‌ی جمع «ها»>

<صفات ترتیبی شمارشی> ::= <فعل‌های اسنادی> | <ضمایر مِلکی و مفعولی> | <«ی» نکره> | <نشانه‌ی جمع «ها»>

---

۱ مخفف عبارت انگلیسی Backus-Naur Form

<صفت ترتیبی مبهم>::=<فعل‌های اسنادی>|<ضمایر مِلکی و مفعولی>|<«ی» نکره>|<نشانه‌ی جمع «ها»>

<«ی» نسبت>::=<فعل‌های اسنادی>|<ضمایر مِلکی و مفعولی>|<«ی» نکره>|<نشانه‌ی جمع «ها»>|<نشانه‌ی جمع «ان»>|<صفات تفصیلی>

لازم با ذکر است که قواعد آواشناسی مطرح شده در بخش قبل برای هر پسوندِ تصریفی، هنگام ترکیب پسوندها با یکدیگر نیز صادق است. به عنوان نمونه ترکیب پسوند تصریفی نشانه‌ی جمعِ «ها» به ضمیر مِلکی و مفعولی دوم شخص مفرد، به صورت «هایَت» خواهد بود، نه «هات».

تعداد پسوندهای تصریفی ممکن، با در نظر گرفتن ترکیب‌های پسوندهای با یکدیگر به ۲۷۵۸ پسوند می‌رسد که می‌توان بر اساس گرامر و قواعد آوایی ارائه شده نسبت به ریشه‌یابی تصریفی ترکیب‌های زبان فارسی اقدام نمود. با در نظر گرفتن قواعد فاصله‌گذاری ذکر شده نیز می‌توان پس از تشخیص پسوند، نسبت به اصلاح فاصله‌گذاری‌ها نیز اقدام نمود که این مهم می‌تواند چالش‌های بسیاری را از زبان فارسی مرتفع نماید. جهت بهبود دقت ریشه‌یابی می‌توان از یک برچسب‌گذارِ ادات سخن جهت تشخیص ادات سخنِ ریشه و تطبیق آن با پسوند استخراج شده، همچنین از یک واژه‌نامه از ریشه‌های زبان جهت صحت‌سنجی ریشه‌های استخراج شده استفاده نمود.

۳-۲-۱-۲ *وندهای اشتقاقی*

در این بخش وندهای اشتقاقی شامل پسوندها، پیشوندها و میانوندها به طور مختصر مرور خواهند شد و در هر مورد عناصرِ ترکیب، قواعدِ ترکیب، اداتِ سخنِ ترکیب حاصل و نحوه‌ی فاصله‌گذاری صحیح مورد بررسی قرار خواهند گرفت.

۳-۲-۱-۲-۱ *پسوندهای اشتقاقی*

عبارت‌هایی که از ترکیب با پسوندهای اشتقاقی ساخته می‌شوند، می‌توانند صفت، اسم یا قید باشند. جدول (۳-۱۲)، به تفکیک هر پسوند، ادات سخنِ ترکیبِ تولید شده و نحوه‌ی فاصله‌گذاری صحیح را نشان می‌دهد.

جدول (۳-۱۲) عبارت‌های پسوندی

پسوند	قاعده	محصول	مثال	نگارش
ا	صفت + ا	اسم	درازا، پهنا	پیوسته
ا	بن مضارع + ا	صفت فاعلی	بینا، زیبا	پیوسته
ا	بن مضارع + ا	صفت لیاقت	خوانا، رها	پیوسته
ا	فعل دعا + ا	فعل لازم	مبادا، بادا	پیوسته
ار	بن ماضی + ار	اسم مصدر	کشتار، کردار	پیوسته
ار	بن فعل + ار	صفت فاعلی	خواستار، برخوردار، پرستار	پیوسته
ار	بن ماضی + ار	صفت مفعولی	گرفتار، مردار	پیوسته
ار	صفت + ار	صفت	پدیدار	پیوسته
آسا	اسم + آسا	صفت/قید	برق‌آسا، غول‌آسا	نیم‌فاصله
اک	بن مضارع + اک	اسم	پوشاک، خوراک	پیوسته
آگین	اسم + آگین	صفت مبالغه	عطرآگین، زهرآگین	تهی
ان	بن مضارع + ان	صفت فاعلی	لغزان، روان، خواهان	پیوسته
ان	برای نسبت نیایی	اسم خاص	بابکان، قبادان	پیوسته*
ان	برای قید زمان/ مکان	قید/اسم خاص	بامدادان، شامگاهان، توران	پیوسته*
ان	اسم + بن مضارع+ان	اسم مصدر	آینه‌بندان، شیرینی‌خوران	پیوسته*
ان	فعل امر + ان	اسم مصدر	آشتی‌کنان	پیوسته
اندر	مادر/پدر/دختر/پسر	صفت‌های مربوط به خویشاوندی ناتنی	مادراندر(نامادری)، پدراندر (پدرندر، پدندر)، پسراندر (پسندر)، دختراندر (دختندر)	پیوسته
انه	اسم + انه	اسم	عصرانه، صبحانه	پیوسته
انه	اسم/صفت + انه	صفت/قید لیاقت	جسورانه، شاهانه	پیوسته
انه	اسم + انه	صفت	ماهانه، روزانه، سالانه	پیوسته
انی	اسم + انی	صفت نسبی	طولانی، روحانی	پیوسته
بار	اسم + بار	اسم؛ به معنی کنار	رودبار، جویبار، دریابار	پیوسته
بان	اسم/صفت + بان	اسم	نگهبان، سایه‌بان	#
چه	اسم + چه	اسم مصغر	طاقچه، آلوچه، کوچه	پیوسته
چی ¥	اسم + چی	صفت فاعلی، شغلی	گاریچی، پستچی	پیوسته

ادامه جدول (۳-۱۲)

پسوند	قاعده	محصول	مثال	نگارش
دان	اسم + دان	اسم	گلدان	پیوسته
دیس	اسم + دیس	اسم	تاقدیس، ناودیس	پیوسته
زار	اسم + زار	اسم	گلزار، چمنزار	
سار	اسم + سار	اسم (مشابهت، کثرت در مکان، ناحیه، نسبت) و نیز به معنی سر	دیوسار، کوهسار، چشمه‌سار، رخسار، شرمسار، گاوسار، سبکسار، نگونسار	#
سان	اسم + سان	اسم	همسان، لاله‌سان، پیلسان	#
ستان	اسم/صفت + ستان	اسم	بوستان، ترکستان	پیوسته
سیر	صفت/اسم + سیر	اسم	سردسیر	پیوسته
ش	بن‌مضارع/صفت + ش	اسم مصدر	جهش، دانش، نرمش	پیوسته
ک	اسم/صفت + ک	اسم/صفت	مرغک، مردک، دلبرک، عروسک، دم‌پختک، سنگک	پیوسته
ک	قید + ک	قید	نرم‌نرمک، کم‌کمک	پیوسته
ک	صفت + ک	اسم	گرمک	پیوسته
کده	اسم/صفت + کده	اسم	میکده، دانشکده	پیوسته
که	صفت + که		مردکه، زنکه	پیوسته
کی	صفت/اسم + کی	صفت/قید	یواشکی، زورکی، دزدکی	پیوسته
گار	بن فعل + گار	صفت فاعلی	آموزگار، خداوندگار	پیوسته
گار	اسم + گار	اسم/صفت	کامگار، یادگار، روزگار	پیوسته
گار	بن ماضی + گار	صفت لیاقت	ماندگار	پیوسته
گان	اسم/صفت شمارشی+ گان	صفت نسبی	مهرگان، دهگان	پیوسته
گانه	صفت بیانی/شمارشی/مبهم+ گانه	صفت	جداگانه، چندگانه، سه‌گانه	پیوسته
گانی	اسم + گانی	صفت	خدایگانی	پیوسته
گاه	اسم + گاه	اسم مکان یا زمان	توقفگاه، دانشگاه، آزمایشگاه	پیوسته
گر	اسم + گر	صفت فاعلی	حیله‌گر، دادگر، ستمگر	#
گر	اسم + گر	صفت شغلی	کارگر، رفتگر، مسگر	#
گین	اسم + گین	صفت	خشمگین	پیوسته

ادامه جدول (۳-۱۲)

پسوند	قاعده	محصول	مثال	نگارش
لاخ	–	صفت/اسم	سنگلاخ، آتش‌لاخ، نمک‌لاخ	$
م	عدد/صفت مبهم+م	صفت‌شمارشی	دوم، سوم، چهارم، چندم	پیوسته
مان	صفت + مان	صفت	شادمان	پیوسته
مان	اسم + مان	اسم	خانمان، دودمان	پیوسته
مان	بن مضارع + مان	اسم معنی	زایمان، سازمان	پیوسته
مان	بن ماضی + مان	اسم ذات	ساختمان	پیوسته
مند	صفت/اسم + مند	صفت لیاقت	ارجمند	پیوسته
مین	–	صفت مبهم/ پرسشی/ شمارشی	دومین، چندمین، کدامین	پیوسته
نا	صفت + نا	اسم	درازنا، تنگنا	پیوسته
ناک	اسم + نا	صفت	خشمناک، ترسناک	پیوسته
نده	بن مضارع + نده	صفت فاعلی	آموزنده	پیوسته
و	–	صفت	ریشو، اخمو	پیوسته
وار	اسم/صفت + وار	صفت/قید/اسم	امیدوار، شاهوار	پیوسته
وار	اسم + وار	صفت/قید/اسم	بهشت‌وار، پلنگ‌وار، دیوانه‌وار	¥
واره	اسم + واره	اسم	ماهواره، دستواره، گاهواره	پیوسته
ور	اسم/صفت + ور	اسم/صفت	دانشور، کینه‌ور، بارور، بهره‌ور	#
ور	اسم + ور	اسم/صفت	رنجور، مزدور	پیوسته
وش	اسم/صفت + وش		تلخوش، شیروش	پیوسته
ومند	اسم + ومند	صفت	برومند	پیوسته
وند	–	–	فولادوند، پسوند	پیوسته
یّت	اسم + یّت	اسم/اسم مصدر	انانیّت، مسیحیّت	پیوسته
ین	اسم + ین	صفت نسبی	سیمین	پیوسته
ین	صفت + ین	صفت	دروغین، چرکین	پیوسته
ین	به/مه + ین	صفت برترین	بهین، مهین	پیوسته
ین	قید + ین	صفت	دیرین (تنها مورد)	پیوسته
ینه	اسم/قید + ینه	صفت	سیمینه، دیرینه	پیوسته

ادامه جدول (۳-۱۲)

پسوند	قاعده	محصول	مثال	نگارش
یّه	صفت/اسم + یّه	اسم خاص	ترکیّه	پیوسته
یّه	صفت/اسم + یّه	صفت نسبی	اصلیّه	پیوسته
ه	بن‌مضارع + ه	اسم‌مصدر/ اسم آلت	ناله، گریه، آویزه، ماله	پیوسته
ه	بن ماضی + ه	صفت مفعولی/ اسم آلت	گفته، دیده، دریده	پیوسته
ه	صفت مبهم/شمارشی + ه	صفت/قید	یک‌شنبه، دوروزه، سه‌ماهه	پیوسته
ه	اسم + ه	اسم	دندانه، گردنه	پیوسته
ه	اسم/صفت + ه	اسم معرفه (محاوره‌ای)	مرده، بقاله، دختره	پیوسته
ه	صفت + ه	صفت مبالغه	می‌خواره، خودکامه	پیوسته
ه	صفت بیانی/شمارشی + ه	اسم	زرده، دهه، سده	پیوسته
ه	واژه + ه	واژه	چهره، رخساره، کرانه	پیوسته
ه	اسم + ه	اسم منسوب	روزه، رویه، پشته	پیوسته
ی	واژه + ی	صفت نسبی	مرزبانی، این‌جوری، تهرانی، پرداختی	پیوسته
ی	قید + ی	قید	نهانی	پیوسته
ی	صفت + ی	صفت/قید	پنهانی، نهانی	پیوسته
ی	صفت/ضمیر + ی	اسم مصدر	مایی، خوبی، مردی	پیوسته
ی		نشانه وحدت	سیری	پیوسته
ی		قید زمان	عصری، عمری	پیوسته
ی		نشانه تحبیب	نورچشمی، استادی	پیوسته
ی		وصفی	کتابی که خریدم	پیوسته
ی		نشانه لیاقت	دیدنی، خوردنی	پیوسته
ی	قید + ی	صفت	ناگهانی	پیوسته

- توضیح *: نام مواردی همچون مکان، قبیله و زمان که با «ان» ساخته می‌شود، محدود است نه زایا. این موارد را می‌توان به کمک واژه‌نامه تشخیص داد.
- توضیح #: «بان» و «سار» و «سان» در تمام موارد به واژه قبل از خود متصل می‌شود مگر در شرایطی که حرف انتهای واژه، «ه» ملفوظ باشد. البته واژه «سایه‌بان» به صورت «سایبان» هم نوشته شده است.

- توضیح $: پسوند «لاخ» از آنجا که بسیار کم کاربرد است، گاهی جدا از واژه‌ی پیش از خود نوشته می‌شود. در این مورد، دستورِ خطِ فارسی تاکید بر پیوسته‌نویسی دارد زیرا این واژه یک پسوند است.
- توضیح ɣ: از آنجا که پیوسته‌نویسی «وار» با اسم‌هایی مانند «پلنگ» منجر به دشوارخوانی آن‌ها می‌شود، می‌توان آن‌ها را با استفاده از نیم‌فاصله جدا نوشت. در دستورِ خطِ فارسی نیز این بند مستثنی از قاعده‌ی پیوسته‌نویسی واژه ترکیب شده با پسوند می‌باشد.

۳-۲-۱-۲-۲ *میانوندهای اشتقاقی*

جدول (۳-۱۳) نمونه‌هایی از واژگان مرکب است که به کمک میانوندهای اشتقاقی ساخته می‌شوند.

جدول (۳-۱۳) عبارت‌های سازنده‌ی واژه‌ی مشتق-مرکب

میانوند	قاعده	محصول	مثال	نگارش
و	بن ماضی + و + بن ماضی	اسم مصدر	زدوخورد، رفت‌وآمد	*
و	بن مضارع + و + بن مضارع	اسم مصدر	سوزوگدازر، گیرودار	*
ا	بن مضارع + ا + بن مضارع	اسم مصدر	کشاکش، تکاپو	پیوسته
و	بن ماضی + و + بن مضارع	اسم مصدر	گفت‌وگو، جست‌وجو	*
و	اسم + و + بن‌فعل	اسم مصدر	مرگومیر	نیم‌فاصله
در			پی‌درپی	تهی#
تا			سرتاسر	نیم‌فاصله#
وا			جورواجور	نیم‌فاصله#
به			سربه‌سر	نیم‌فاصله#
ا	اسم + ا + اسم	اسم	سراپا، بناگوش، پیشاپیش	پیوسته
و	اسم + و + بن فعلِ هم‌معنی	اسم‌مصدر	مرگ‌ومیر	نیم‌فاصله
و	فعل امر + و + فعل نهی	صفت‌فاعلی	کجداروم‌مریز، بخور و نمیر	*
و	اسم + و + اسم	صفت	گرگ‌ومیش	*
و	صفت + و + صفت	صفت	سرخ‌وسفید، سفید و سیاه	*
و	صفت منفی + و + صفت منفی	صفت	بی‌بو و بی‌خاصیت	*
و	صفت/قید + و + صفت/قید	قید	افتان و خیزان	*

- توضیح *: در بسیاری از انواع ترکیب‌های معناداری که با میانوند «و» ساخته می‌شوند، این اشکال پیش می‌آید که به دلیل حرف انتهایی بخش اول یا حرف آغازین بخش دوم که یکی از حروف «ر»، «ز»، «ژ»، «و»، «ذ» و «د»، برای خوانایی واژه، آن را با فاصله‌ی کامل و نه نیم‌فاصله می‌نویسند. برای مثال، «رفت و روب» از «رفت‌وروب» خواناتر است. فرهنگستان نیز در این زمینه الزامی برای استفاده از نیم‌فاصله ندارد و تنها الزام آن برای جدانویسی این واژه‌هاست.
- توضیح #: برخی از حروف اضافه نیز به عنوان میانوند به کار می‌روند. در این حالت اگر به اشتباه جدا نوشته شده باشند، ممکن است تشخیص آن‌ها در جمله مشکل شود.

۳-۲-۱-۲-۳ *پیشوندهای اشتقاقی*

عبارت‌های پیشوندیِ اشتقاقی در جدول (۳-۱۴) است. همان‌طور که مشخص است، اکثر واژگان تولیدی از این دست، در واژه‌نامه یافت خواهند شد.

جدول (۳-۱۴) قواعد ساخت واژگان پیشوندی

پیشوند	قاعده	محصول	مثال	نگارش
ب	ب + اسم	صفت	بهنجار، بنام (مشهور)	پیوسته
با	با + اسم	صفت/قید	باهوش، باهنر	*
باز	باز + فعل	فعل مرکب	بازآمدن، بازیافتن	*
باز	باز + بن ماضی	اسم	بازخواست، بازدید	*
باز	باز + بن مضارع	اسم	بازجو، بازپرس	*
بر	بر + فعل	فعل مرکب	برداشتن، برخواستن	*
بر	بر + بن ماضی	اسم	برخاست، برآورد	*
بر	بر + بن مضارع	اسم	برچسب	*
بر	بر + اسم	صفت	برکنار، برقرار، بردوام	*
به	به + اسم	صفت	بهروز (روز نیک، اسم خاص)	پیوسته
ب	ب + اسم	صفت	بهنجار، بنام (مشهور)	پیوسته
با	با + اسم	صفت/قید	باهوش، باهنر	*

ادامه‌ی جدول (۳-۱۴)

پیشوند	قاعده	محصول	مثال	نگارش
باز	باز + فعل	فعل مرکب	بازآمدن، بازیافتن	*
باز	باز + بن ماضی	اسم	بازخواست، بازدید	*
باز	باز + بن مضارع	اسم	بازجو، بازپرس	*
بر	بر + فعل	فعل مرکب	برداشتن، برخواستن	*
بر	بر + بن ماضی	اسم	برخاست، برآورد	*
بر	بر + بن مضارع	اسم	برچسب	*
بر	بر + اسم	صفت	برکنار، برقرار، بردوام	*
به	به + اسم	صفت	بهروز (روز نیک، اسم خاص)	پیوسته
بی	بی + اسم	صفت	بی‌خرد، بی‌گناه، بی‌ادب	نیم‌فاصله
بی	بی+ بن فعل	صفت	بی‌ریخت، بی‌تاب	نیم‌فاصله
بی	بی+ ضمیر	صفت/قید	بی‌خود، بی‌خویشتن	نیم‌فاصله
بی	بی+ ترکیب‌های وصفی/ اضافی/عطفی	صفت	بی‌دست‌وپا، بی‌سروسامان، بی‌شیله‌پیله	نیم‌فاصله
در	در + فعل	فعل مرکب	درآمدن	*
در	در + بن فعل	اسم	درآمد، درخواست، درگیر	*
در	در + ضمیر مبهم		درهم	*
فر	فر+اسم/ صفت	اسم	فرخجسته، فرآیند، فرآورده	*
فراز	فراز+ فعل	فعل مرکب	فرازآمدن، فرازآوردن	نیم‌فاصله
فرا	فرا + فعل	فعل مرکب	فراگرفتن	*
فرا	فرا + بن مضارع	اسم/قید	فراخور	*
فرا	فرا + ضمیر مبهم		فراهم	*
فرو	فرو + فعل	فعل مرکب	فروبردن، فروگذاشتن	*

ادامه‌ی جدول (۳-۱۴)

پیشوند	قاعده	محصول	مثال	نگارش
فرو	فرو + بن فعل	اسم/قید	فروپاشی، فروگذار، فروکش	*
فرو	فرو + اسم	اسم/ قید	فروتن، فرومایه، فرودست	*
لا	لا + واژه عربی		لامذهب، لاعلاج	*
ن	ن وسط صفات بیانی	صفت	خدانشناس، زبان‌نفهم	پیوسته
ن	ن + اسم/صفت/بن	صفت/قید	نفهم، نسنجیده	پیوسته
نا	نا + صفت/اسم	صفت	نادرست، ناکام،	*
نا	نا وسط صفات بیانی	صفت	حق‌ناشناس	*
نا	نا + بن		ناشایست، ناسپاس، ناپسند	*
نا	نا + مصدر		نادیدن، نارسیدن	*
نا	نا + اسم/صفت	صفت/قید/اسم	ناسپاس، نادانسته، نادرستی	*
نا	نا + اسم ناتنی مفرد	اسم	نامادری، ناخواهری، نادختری	*
وا	وا + مصدر	فعل مرکب	واداشتن	*
وا	وا + بن مضارع	قید/اسم	واگیر، وادار، واریز	*
وا	وا + اسم مفرد	اسم/ صفت/ قید	وارو، واپس	*
ور	ور + مصدر	فعل مرکب	وررفتن، ورآمدن، ورچیدن	*
ور	ور + بن	اسم/ صفت	ورانداز، ورمال، ورشکست	*
ور	ور + اسم	اسم	وردست،	*
ور	ور + صفت	قید	ورپریده	*
هم	هم + اسم	صفت مفرد	همکار، هم‌اسم، همراه، همدم	#
هم	هم + بن ماضی	صفت مفرد	هم‌نشست، همزاد	#
هم	هم + صفت اشاره	اسم/ صفت/ قید	همین، همان، همچنین، همچنان	پیوسته
هم	هم + بن مضارع	اسم/ صفت	هم‌نشین، همساز، همگرا	#

ادامه‌ی جدول (۳-۱۴)

پیشوند	قاعده	محصول	مثال	نگارش
هم	هم + ضمیر مبهم	ضمیر	همدیگر، همدگر	پیوسته
هم	هم + پسوند	صفت	همسان، همگر	پیوسته
هم	هم + حرف اضافه		همچو، همچون	پیوسته

– توضیح *: بر اساس قواعد خط فارسی، کلیه‌ی واژه‌های پیشوندی باید جدا نوشته شوند. در مواردی که نگارش «تهی» پیشنهاد شده است، به دلیل حرف انتهایی پیشوند که غیر چسبان است (مانند «ا»، «ز» و «ر») خود به خود پیوسته‌نویسی روی نخواهد داد و نیز استفاده از نیم‌فاصله، منجر به سختی عمل نگارش خواهد شد، اما استفاده از فاصله‌ی تمام در همه‌ی این موارد نادرست است. بنابراین، تنها حالت خطا در نگارشِ واژه، زمانی است که با فاصله‌ی تمام نوشته شود.

– توضیح #: در مورد نگارش واژه‌هایی که با پیشوند «هم» به وجود آمده‌اند استثنائاتی وجود دارد که باید حتما پیوسته نوشته شوند. این موارد استثناء عبارتند از زمانی که واژه‌ی حاصل، بسیط گونه باشد (مانند همشهری)، یا بخش دوم از واژه‌ی حاصل، تک هجایی باشد و یا با «آ» شروع شده باشد. در سایر موارد، هم با یک نیم‌فاصله از واژه‌ی بعد از خود جدا می‌شود مانند «همکار» و «هماهنگ».

۳-۲-۲ صَرف فعل‌ها

نداشتن قواعد تعریف شده‌ی مشخص برای ساخت و صَرف فعل‌های فارسی، وجود فعل‌های بسیار پیچیده، وجود فعل‌های چندجزئی و مرکب و نیز موارد خاص و استثناء که از قواعد صَرف فعلی تبعیت نمی‌کنند، ریشه‌یابی فعل‌های فارسی را مشکل می‌سازد. از این رو، برای حفظ صحت و دقت کارکرد خطایابی، باید تمامی حالات صَرف کلیه‌ی فعل‌ها در فهرست واژگان زبان قرار گیرد. اما چون تعداد این فعل‌ها بسیار زیاد می‌شود این امر خود چالشی دیگر است که استفاده ترکیبی از ریشه‌یاب‌های جامع، دقیق و مطمئن برای بخش اعظمی از فعل‌ها و نیز قراردادن فعل‌های خاص در واژه‌نامه فعلاً بهترین راهکار به نظر می‌رسد.

۳-۲-۲-۱ *ساختارهای فعل فارسی*

فعل را در زبان فارسی، از جهت ساختمان، به شش گروه می‌توان تقسیم کرد:

- **فعل‌های ساده:** فعل‌های ساده فعل‌هایی هستند که بن مضارع آن‌ها یک تکواژ است؛ مانند: «رفتن»، «گفتن». این فعل‌ها در چند دسته قرار می‌گیرند.

- **فعل‌های پیشوندی:** فعل‌های پیشوندی فعل‌هایی هستند که از یک پیشوند و یک فعل ساده ساخته شده‌اند. مهمترین پیشوندهایی که فعل پیشوندی می‌سازند عبارتند از: «بر»، «در»، «فرو»، «فرا»، «باز»، «وا» و «ور»؛ نمونه‌های فعل‌های پیشوندی: «برداشتن»، «درافتادن»، «فرورفتن» و «بازداشتن».

- **فعل‌های مرکب:** فعل‌های مرکب فعل‌هایی هستند که از یک کلمه، که آن را *فعل‌یار* می‌نامند، با یک فعل ساده که همکرد نامیده می‌شود، ساخته می‌شوند و مجموعاً معنی واحدی را می‌رسانند؛ مانند: «آرایش کردن» و «تأسف خوردن».

- **فعل‌های پیشوندی مرکب:** فعل‌های پیشوندی گاهی با کلمه‌ای ترکیب می‌شوند و معنی واحدی را بیان می‌کنند. معنی مزبور نسبت به معنی لغوی کلمه‌های سازنده غالباً مجازی است؛ مانند: «دم در کشیدن»، «سر در آوردن» و «تن در دادن».

- **عبارت‌های فعلی:** عبارت فعلی به دسته‌ای از کلمات اطلاق می‌شود که از مجموع آن‌ها معنی واحدی حاصل می‌شود. عبارت‌های فعلی بیش از دو کلمه هستند که نخستین کلمه‌ی ترکیب، حرف اضافه است و مجموع عبارت نیز معمولاً معنی مجازی دارد؛ مانند: «از پای درآمدن» و «بر پا کردن».

- **فعل‌های لازم یک شخصه:** مراد از این اصطلاح، فعل‌هایی است که به صورت لازم و فقط با ساخت سوم شخص مفرد به کار می‌روند و به جای شناسه، ضمیر پیوسته‌ی مفعولی و اضافی، شخص و زمان فعل را نشان می‌دهد. مانند: «خوشم آمد». ضمیرِ «م» پیوسته به «خوش»، به جای شناسه، شخص فعل را معین می‌کند و «آمد» در همه‌ی شکل‌های فعل ماضی ساده به همان شکل باقی می‌ماند. مانند: «خوشم آمد»، «خوشت آمد»، «خوشش آمد».

۳-۲-۲-۱-۱ *فعل‌های خاص*

سایر ساختارهای فعلی در فارسی وجود دارند که دارای ساختار ساده و حالت‌های *صَرفی* محدودی هستند. این ساختارها در زیر آمده‌اند.

- **فعل ربطی (اسنادی)**: فعل‌های ربطی فعل‌هایی هستند که معنی کاملی ندارند و فقط برای اثبات یا نفی نسبت به کار می‌روند و معنای آن‌ها با آوردن صفت یا کلمه‌ای دیگر کامل می‌شود؛ مانند: «هوا روشن است». معروف‌ترین فعل‌های ربطی یا اسنادی عبارتند از: «استن»، «بودن» و «شدن». همچنین فعل‌های «گشتن» و «گردیدن» اگر به معنی «شدن» باشند، فعل ربطی به شمار می‌روند.
- **فعل‌های شبه کمکی**: فعل‌هایی مانند «بایستن»، «توانستن»، «شایستن»، «خواستن» و «شدن» که گاهی در جمله به پاره‌ای از فعل‌ها و بن ماضی فعل‌ها نوعی کمک معنایی و کاربردی می‌دهند، فعل شبه‌کمکی نامیده می شودند و فعل همراه آن‌ها را *فعل پیرو* می‌گویند؛ مثلاً در جمله‌ی «سیما نتوانست به مسافرت برود»، «نتوانست» فعل شبه‌کمکی، و «برود»، فعل پیرو است.
- **فعل غیر شخصی**: فعل‌های شبه‌کمکی «توانستن»، «بایستن» و «شایستن» گاهی فعلی می‌سازند که بر شخص معینی دلالت نمی‌کنند؛ مانند «نتوان رفت»، «نباید گفت»، «نشاید رفت». این فعل‌ها را فعل‌های غیرشخصی می‌گویند.
- **فعل وصفی**: فعل وصفی آن است که فعل را به صورت صفت مفعولی (= بن ماضی + ه) بیاورند. در این صورت، فعل دیگری در آخر جمله می‌آورند. ارزش فعل اخیر (از جهت شخص، شمار، زمان، و وجه) ارزش فعل وصفی را مشخص می‌کند؛ مانند «احمد به خانه **رفته** نهار **خورد**».

۳-۲-۲-۲ *صَرف فعل*

فعل در زبان فارسی در ۱۴ زمان و در ۶ صیغه‌ی شخصی *صَرف* می‌شود. در هر زمان، با توجه به *مجهول* یا *معلوم* بودن فعل، *منفی* یا مثبت بودن صیغه *صَرفی* و همچنین *طرز بیان*، حالت‌های مختلفی ایجاد می‌شود. در *صَرف* فعل، پیشوندها یا پسوندهایی به ریشه فعل افزوده می‌گردند. در برخی موارد هنگام افزایش پسوند یا پیشوند، حروف ریشه، قلب به سایر حروف می‌گردند یا حذف می‌شوند.

صَرف فعل در تمام ساختارهای فعلی یکسان است. لذا، یافتن الگوی *صَرف* فعل حائز

اهمیت است. در فعل‌های مرکب فقط یک بخشِ فعل صَرف می‌شود و بخش‌های دیگر ثابت می‌مانند. فعل‌ها در زبان فارسی در ۱۴ زمان زیر صَرف می‌شوند:

- ماضی ساده: «گفتم»، «گفتی»، «گفت»، ...
- ماضی استمراری: «می‌گفتم»، «می‌گفتی»، «می‌گفت»، ...
- ماضی بعید: «گفته بودم»، «گفته بودی»، «گفته بود»، ...
- ماضی مستمر: «داشتم می‌گفتم»، «داشتی می‌گفتی»، «داشت می‌گفت»، ...
- ماضی ساده نقلی: «گفته‌ام»، «گفته‌ای»، «گفته است»، ...
- ماضی استمراری نقلی: «می‌گفته‌ام»، «می‌گفته‌ای»، «می‌گفته است»، ...
- ماضی بعید نقلی: «گفته بوده‌ام»، «گفته بوده‌ای»، «گفته بوده است»، ...
- ماضی مستمر نقلی: «داشته‌ام می‌گفته‌ام»، «داشته‌ای می‌گفته‌ای»، «داشته است می‌گفته است»، ...
- ماضی التزامی: «گفته باشم»، «گفته باشی»، «گفته باشد»، ...
- مضارع اخباری: «می‌گویم»، «می‌گویی»، «می‌گوید»، ...
- مضارع مستمر: «دارم می‌گویم»، «داری می‌گویی»، «دارد می‌گوید»، ...
- مضارع التزامی: «بگویم»، «بگویی»، «بگوید»، ...
- آینده: «خواهم گفت»، «خواهی گفت»، «خواهد گفت»، ...
- امر: «بگو»، «بگویید»

برخی از این زمان‌ها یا صیغه‌های خاص ممکن است امروزه مورد استفاده نباشند. یا مثلاً برخی از صیغه‌های زمانی و شخصی برای برخی فعل‌های خاص استفاده نشوند. در هر ساختی از فعل، بخشی یا جزیی است که مفهوم کار، حالت، وجود، یا اسناد از آن معلوم می‌شود. این جز را «بن» گویند. تمام صیغه‌های فعل از *بن فعل* و *قاعده صَرف* آن به طور کاملاً قاعده‌مندی قابل استخراج هستند. هر فعلی دو بن دارد *بن ماضی* و *بن مضارع*. اگرچه قواعدی برای تبدیل بن ماضی به بن مضارع وجود دارد، اما استثنائاتی نیز وجود دارد. به منظور صحت، دقت و سادگی بهتر است برای هر فعل در واژه‌نامه دو بن موجود باشد. با داشتن بن فعل (ماضی و مضارع) می‌توان تمام صیغه‌های صَرفی آن را تولید نمود.

فعل‌ها یا لازم هستند یا متعدی (یا دو وجهی که می‌شود آن‌ها را معادل متعدی در نظر گرفت). فعل‌های متعدی در دو شکل ۱) معلوم (می‌گوید)، و ۲) مجهول (گفته می‌شود) صَرف می‌شوند. اما فعل‌های لازم فقط به صورت معلوم صَرف می‌شوند.

۳-۲-۲-۲-۱ تعداد صیغه‌ها

با توجه به تعداد صیغه‌ها و زمان‌ها، تعداد صیغه‌های صَرفی برای یک بن متعدی در ساختار صَرفی ساده برابر ۳۲۰ حالت متفاوت است. به این صورت که زمان‌ها به جز امر (یعنی ۱۳ زمان مختلف) در ۶ شخص صَرف می‌شوند. فعل امر دارای دو شخص است. علاوه بر این فعل‌ها از نظر معلوم یا مجهول بودن در دو حالت صَرف می‌شوند. همچنین هر صیغه‌ای دارای دو حالت منفی و مثبت است لذا تعداد کل صیغه‌ها برابر می‌شود با: ۳۲۰=۲×۲×(۱×۲+۶×۱۳). البته تعدد حالت‌های صَرفی به همین جا ختم نمی‌شود و برای هر کدام از ۳۲۰ صیغه فوق، ممکن است چندین گونه بیان داشته باشیم. مثلاً برای مضارع التزامی منفی بن «گوی» داریم: «نگویم» و «مگویم»، و برای مثبت آن داریم: «بگویم» و «گویم». لذا، اگر بخواهیم از نظر طرز بیان تعداد فعل‌ها را حساب کنیم بیش از مقدار ۳۲۰ خواهد شد. فرمول فوق علاوه بر فعل‌های ساده برای ساختارهای صَرفی سایر فعل‌ها شامل: فعل‌های پیشوندی، فعل‌های مرکب، فعل‌های مرکب پیشوندی، و فعل‌های عبارت فعلی نیز برقرار است؛ در واقع هنگام صَرف این گونه ساختارها فقط جزء فعلی آن‌ها صَرف می‌شود و سایر اجزاء (با تغییرات جزیی) ثابت باقی می‌مانند.

ساختارهای «فعل‌های ناگذر یک شخصه» و «فعل‌های اسنادی» و سایر ساختارهای خاص دارای زمان و صیغه‌های محدودی هستند و از فرمول فوق تبعیت نمی‌کنند و باید جداگانه به واژه‌نامه افزوده شوند که به علت سادگی در این بخش مورد بررسی قرار نخواهند گرفت. ضمایر مفعولی تنها ساختارهایی هستند که می‌توانند به جزء اصلی فعل و سایر اجزای آن بچسبند، که این موضوع در غلط‌یابی املایی بسیار حائز اهمیت است. به عنوان مثال: «گفتمش» یعنی «به او گفتم». البته ترکیب تمام اتصالات ضمایر مفعولی و شخص‌های فعل متداول نیست ولی از نظر دستوری درست است مانند «بگوییدتان» که یک ترکیب بدون استفاده اما درست است.

۳-۲-۲-۳ بیان الگوی صَرف فعل با عبارات منظم

الگوی صَرف فعل‌های فارسی یک زبان منظم است. همچنین عبارات منظم روشی بسیار گویا برای بیان الگوهای صَرف فعل هستند؛ لذا در اینجا از عبارات منظم برای بیان الگوی صَرف فعل‌ها استفاده شده است. به عنوان مثال با استفاده از عبارت منظم، الگوی صَرف فعل ماضی استمراری نقلی از مصدر «گفتن» به صورت زیر خواهد بود:

می à(گفت)ه(à ایم|à اید|à اند|à ام|à ای| است)؟

از علامت " " برای بیان فاصله و از "◊" برای بیان نیم‌فاصله استفاده شده است (الگو را از راست به چپ بخوانید). برای تشخیص سایر مصدرها می‌توان الگوی فوق را به صورت زیر تعمیم داد:

می à(گفت|خورد|زد|گرفت)ه(à ایم|à اید|à اند|à ام|à ای| است)؟

در بالا فقط ۴ بن به عنوان نمونه قرار داده شده‌اند. می‌توان تمامی بن‌های ماضی را در الگوی فوق قرار داد. با توجه به این که تعداد بن‌های ماضی فارسی محدود است (حدود ۳۵۰ بن)، ارائه‌ی چنین الگوهایی عملیاتی و دست‌یافتنی است.

۳-۲-۲-۴ *الگوی صَرف فعل‌ها در زبان فارسی*

در این بخش قوانین صَرف فعل در زبان فارسی برای تمام ساختارهای فعل به طور کامل فرمول‌بندی شده و در قلب عبارات منظم بیان می‌شود. همان طور که در بخش قبل گفته شد ۶ ساختار فعل وجود دارد. همه این ساختارها به جز «عبارت‌های فعلی» دارای الگوی یکسان برای تمام فعل‌های متعلق به آن ساختار هستند. الگوی صَرف فعل‌های ساده به عنوان پایه‌ای برای سایر ساختارهای فعل هستند که در آن‌ها نیز رفتاری یکسان دارند.

نحوه‌ی بیان الگوها به این صورت است که ابتدا تمام الگوهای صَرفی مربوط به یک زمان، مثلاً ماضی ساده، استخراج شده و به همراه مثال ارایه می‌شوند. سپس تمام این الگوها تجمیع شده و در قالب یک الگوی تصریف جامع، ارایه خواهند شد که به ترتیب به آن‌ها «زیرالگو» و «الگوی جامع» گفته می‌شود. به هر کدام از زیرالگوها یک کد داده شده است. هر کدام از این کدها شامل بخش‌های زیر است:

- زمان اصلی: گذشته (G)، حال (H)، آینده و امر (A) هستند.
- زمان فرعی: یک شماره است که مشخص کننده زمان است.
- معلوم یا مجهول: I برای معلوم و T برای مجهول است.
- منفی یا مثبت: با P و N که به ترتیب بر مثبت و منفی دلالت می‌کنند، مشخص می‌شود.
- حالت بیان: با شماره مشخص می‌شود.

علاوه بر زیرالگوهای ارایه شده، ممکن است زیرالگوهای دیگری هم وجود داشته باشند. در این بخش پر استفاده‌ترین زیرالگوها مطرح شده‌اند و زیرالگوهای کم استفاده لیست

نشده‌اند. لذا ممکن است مثال‌ها و زیرالگوها تمام حالت‌های صَرفی را پوشش ندهند اما این جامعیت در الگوی جامع رعایت می‌شود. الگوی جامع شامل حالت‌های غلطِ رایج نیز هست. این کار به دو دلیل صورت گرفته است: اول این که بیان الگو راحت‌تر می‌شود و دوم این که در مواردی که نیاز به شناسایی الگوهای غلط باشد، این کار مفید واقع می‌شود. علایم اختصاری که در بیان عبارات منظم استفاده شده‌اند در جدول (۳-۱۵) آمده‌اند.

جدول (۳-۱۵) علایم اختصاری به کار رفته در بیان الگوهای صَرف فعل‌ها

ردیف	علامت	توضیح
۱	[G]	بن ماضی
۲	[AG]	بن‌های ماضی که با الف شروع می‌شوند و «آ» به «ا» تبدیل شده است
۳	[BG]	بن ماضی که با الف شروع نمی‌شود
۴	[H]	بن مضارع
۵	[HA]	بن مضارعی که به الف ختم می‌شود
۶	[AH]	بن مضارعی که با الف شروع می‌شود
۷	[HB]	بن مضارعی که به الف ختم نمی‌شود
۸	[BH]	بن مضارعی که با الف شروع نمی‌شود
۹	[AHA]	بن مضارعی که با الف شروع و با الف خاتمه می‌یابد
۱۰	[AHB]	بن مضارعی که ابتدای آن الف است و انتهای آن الف نیست
۱۱	[BHA]	بن مضارعی که ابتدای آن الف نیست و انتهای آن الف است
۱۲	[BHB]	بن مضارعی که ابتدا و انتهای آن الف نیست
۱۳	[PFGS]	(م|ی|یم|ید|ند)؟
۱۴	[PFGN]	(ٰایم|ٰاید|ٰاند|ٰام|ٰای| است)؟
۱۵	[PFH]	(یم|ید|ند|م|ی|د)
۱۶	[Z]	(م|ت|ش|مان|تان|شان)؟
۱۷	[IMG]	بن‌های ماضی مجهول‌ساز: (شد|گشت|گردید)
۱۸	[IMH]	بن‌های مضارع مجهول‌ساز: (شو|گرد)
۱۹	{D}	بیانگر فاصله بین بخش‌های فعل

فعل‌های کمکیِ مجهول‌ساز، گاهی اوقات فعل‌های «گردیدن» و «گشتن» به جای «شدن» استفاده می‌شوند، مانند «گفته می‌گردد» به جای «گفته می‌شود». در مواردی بسیار نادر، فعل «رفتن» به جای «شدن» استفاده می‌شود، مانند «گفته رفت» به جای «گفته شد».

توجه کنید که الگوهای تصریفی مجهول فقط برای فعل‌های متعدی کاربرد دارند. در ادامه از مصدر «گفتن» برای بیان مثال‌ها استفاده شده که یک فعل متعدی است. در مورد فعل‌های لازم مانند «رفتن» صیغه‌های مجهول صَرف نمی‌شوند.

برای منفی‌سازی در فارسی می‌توان هم از «ن» استفاده کرد و با کاربرد کمتر از «م». همچنین گاهی برای تاکید در جملات مثبت از «ب» در ابتدای فعل استفاده می‌شود. با در نظر گرفتن این حالت‌ها ممکن است تعداد حالت‌های صَرفی زیاد شود؛ که همه‌ی این حالت‌ها در مثال‌ها و زیرالگوها لیست نشده‌اند ولی در الگوی جامع در نظر گرفته شده‌اند.

۳-۲-۲-۴-۱ *ماضی ساده معلوم*

الگوی صَرف فعل ماضی ساده معلوم در جدول (۳-۱۶) نشان داده شده است.

جدول (۳-۱۶) الگوی صَرف فعل ماضی ساده معلوم

کد	عبارت منظم و مثال صَرف	توضیحات
G1IP1	[G](م|ی|یم|ید|ند)؟(م|ت|ش|مان|تان|شان)؟ مثال: گفتم، گفتی، گفت، ...	ماضی ساده معلوم مثبت (یک)
G1IP2	(ب)([BG])(م|ی|یم|ید|ند)؟(م|ت|ش|مان|تان|شان)؟ مثال: بگفتم، بگفتی، بگفت، ... (ب)(ی[AG])(م|ی|یم|ید|ند)؟(م|ت|ش|مان|تان|شان)؟ مثال: بیامرزیدم، بیامرزیدی، بیامرزید، ...	ماضی ساده معلوم مثبت (دوم)
G1IN1	(ن|م)([BG])(م|ی|یم|ید|ند)؟(م|ت|ش|مان|تان|شان)؟ مثال: نگفتم، نگفتی، نگفت، ... (ن|م)(ی[AG])(م|ی|یم|ید|ند)؟(م|ت|ش|مان|تان|شان)؟ مثال: نیامرزیدم، نیامرزیدی، نیامرزید، ...	ماضی ساده معلوم منفی (یک)
G1I	(ب|ن|م)؟([BG])[Z][PFGS] ((ب|ن|م)ی)؟([AG])[Z][PFGS]	ماضی ساده معلوم

در G1IP2 و G1IN1 وقتی بن ماضی با الف شروع شود، «ی» به اول بن افزوده می‌شود و «آ» قلب به «ا» می‌شود. مثلاً از مصدر «آویخت» می‌گوییم «بیاویختم».

۳-۲-۲-۴-۲ *ماضی ساده مجهول*

الگوی صَرف فعل ماضی ساده مجهول در جدول (۳-۱۷) نشان داده شده است.

جدول (۳-۱۷) الگوی صَرف فعل ماضی ساده مجهول

کد	عبارت منظم	توضیحات
G1TP1	[G]ه شد(یم|ید|ند|م|ی)؟ مثال: گفته شدم، گفته شدی، گفته شد، ...	ماضی ساده مجهول مثبت (یک)
G1TP2	ب([BG])ه شد(یم|ید|ند|م|ی)؟ مثال: بگفته شدم، بگفته شدی، بگفته شد، ... ب(ی[AG])ه شد(یم|ید|ند|م|ی)؟ مثال: بیامرزیده شدم، بیامرزیده شدی، بیامرزیده شد، ...	ماضی ساده مجهول مثبت (دو)
G1TP3	[G]ه بشد(یم|ید|ند|م|ی)؟ مثال: گفته بشدم، گفته بشدی، گفته بشد، ...	ماضی ساده مجهول مثبت سوم
G1TN1	[G]ه (ن|م)شد(یم|ید|ند|م|ی)؟ مثال: گفته نشدم، گفته نشدی، گفته نشد، ...	ماضی ساده مجهول منفی یکم
G1TN2	(ن|م)([BG])ه شد(یم|ید|ند|م|ی)؟ مثال: نگفته شدم، نگفته شدی، نگفته شد، ... (ن|م)(ی[AG])ه شد(یم|ید|ند|م|ی)؟ مثال: نیامرزیده شدم، نیامرزیده شدی، نیامرزیده شد، ...	ماضی ساده مجهول منفی دوم
G1T	((ن|م|ب)؟([BG]))ه (ب|ن)؟[IMG][PFGS] ((ن|م|ب)ی)؟([AG])ه (ب|ن)؟[IMG][PFGS]	ماضی ساده مجهول

بین جزء مجهول ساز، «شدن»، و جزء صَرفی «بن فعل» فاصله نمی‌افتد (یعنی کلمه‌ی دیگری قرار نمی‌گیرد).

۳-۲-۲-۴-۳ *ماضی استمراری معلوم*

الگوی صَرف فعل ماضی استمراری معلوم در جدول (۳-۱۸) نشان داده شده است.

جدول (۳-۱۸) الگوی صَرف فعل ماضی استمراری معلوم

کد	عبارت منظم	توضیحات
G2IP1	می◊[G](یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: می گویم، می گویی، می گوید، ...	ماضی استمراری معلوم مثبت یکم
G2IN1	نمی◊[G](یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: نمی گویم، نمی گویی، نمی گوید، ...	ماضی استمراری معلوم مثبت یکم
G2I	(ن)؟می◊[G][PFGS][Z]	ماضی استمراری معلوم
جزء «می» به همراه «نیم فاصله» می‌آید. در مواقعی که مصدر با «الف» شروع می‌شود، تغییری در آن حاصل نمی‌شود.		

در این الگو گویش‌های قدیم لحاظ نشده‌اند. این گویش‌ها در جدول (۳-۱۹) نشان داده شده‌اند.

جدول (۳-۱۹) نمونه‌ی گویش‌های قدیم فعل ماضی استمراری معلوم

سوم شخص جمع	دوم شخص جمع	اول شخص جمع	سوم شخص مفرد	دوم شخص مفرد	اول شخص مفرد
می‌بگفتند	می‌بگفتید	می‌بگفتیم	می‌بگفت	می‌بگفتی	می‌بگفتم
همی گفتند	همی گفتید	همی گفتیم	همی گفت	همی گفتی	همی گفتم
همی‌بگفتند	همی‌بگفتید	همی‌بگفتیم	همی‌بگفت	همی‌بگفتی	همی‌بگفتم
می‌نگفتند	می‌نگفتید	می‌نگفتیم	می‌نگفت	می‌نگفتی	می‌نگفتم
همی‌نگفتند	همی‌نگفتید	همی‌نگفتیم	همی‌نگفت	همی‌نگفتی	همی‌نگفتم
همی‌نگفتند	همی‌نگفتید	همی‌نگفتیم	همی‌نگفت	همی‌نگفتی	همی‌نگفتم

۳-۲-۲-۴-۴ *ماضی استمراری مجهول*

الگوی صَرف فعل ماضی استمراری مجهول در جدول (۳-۲۰) نشان داده شده است.

جدول (۳-۲۰) الگوی صَرف فعل ماضی استمراری مجهول

کد	عبارت منظم	توضیحات
G2TP1	[G]ه می◊شد(یم|ید|ند|م|ی)؟ مثال: گفته می‌شدم، گفته می‌شدی، گفته می‌شد ...	ماضی استمراری مجهول مثبت یکم
G2TP2	ب([BG])ه می◊شد(یم|ید|ند|م|ی)؟ مثال: بگفته می‌شدم، بگفته می‌شدی، بگفته می‌شد، ... ب(ی[AG])ه می◊شد(یم|ید|ند|م|ی)؟ مثال: بیامرزیده می‌شدم، بیامرزیده می‌شدی، بیامرزیده می‌شد، ...	ماضی استمراری مجهول مثبت دوم
G2TN1	[G]ه نمی◊شد(یم|ید|ند|م|ی)؟ مثال: گفته نمی‌شدم، گفته نمی‌شدی، گفته نمی‌شد	ماضی استمراری مجهول منفی یکم
G2TN2	(ن|م)([BG])ه می◊شد(یم|ید|ند|م|ی)؟ مثال: نگفته می‌شدم، نگفته می‌شدی، نگفته می‌شد، ... (ن|م)(ی[AG])ه می◊شد(یم|ید|ند|م|ی)؟ مثال: نیامرزیده می‌شدم، نیامرزیده می‌شدی، نیامرزیده می‌شد، ...	ماضی استمراری مجهول منفی دوم
G2T	(ن|م|ب)؟[BG]ه (ن)؟می◊[IMG][PFGS] ((ن|م|ب)ی)؟[AG]ه (ن)؟می◊[IMG][PFGS]	ماضی استمراری مجهول

نمونه گویش‌ها قدیم ماضی استمراری مجهول در جدول (۳-۲۱) نشان داده شده‌اند.

جدول (۳-۲۱) نمونه‌ی گویش‌های قدیم فعل ماضی استمراری مجهول

سوم شخص جمع	دوم شخص جمع	اول شخص جمع	سوم شخص مفرد	دوم شخص مفرد	اول شخص مفرد
گفته همی‌شدند	گفته همی‌شدید	گفته همی‌شدیم	گفته همی‌شد	گفته همی‌شدی	گفته همی‌شدم
همی گفته می‌شدند	همی گفته می‌شدید	همی گفته می‌شدیم	همی گفته می‌شد	همی گفته می‌شدی	همی گفته می‌شدم
بگفته همی‌شدند	بگفته همی‌شدید	بگفته همی‌شدیم	بگفته همی‌شد	بگفته همی‌شدی	بگفته همی‌شدم
گفته همی‌نشدند	گفته همی‌نشدید	گفته همی‌نشدیم	گفته همی‌نشد	گفته همی‌نشدی	گفته همی‌نشدم
همی گفته نمی‌شدند	همی گفته نمی‌شدید	همی گفته نمی‌شدیم	همی گفته نمی‌شد	همی گفته نمی‌شدی	همی گفته نمی‌شدم

۳-۲-۲-۴-۵ *ماضی بعید معلوم*

الگوی صَرف فعل ماضی بعید معلوم در جدول (۳-۲۲) نشان داده شده است.

جدول (۳-۲۲) الگوی صَرف فعل ماضی بعید معلوم

کد	عبارت منظم	توضیحات
G3IP1	[G]ه بود(یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: گفته بودم، گفته بودی، گفته بود، ...	ماضی بعید معلوم مثبت یکم
G3IP2	ب([BG])ه بود(یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: بگفته بودم، بگفته بودی، بگفته بود، ... ب(ی[AG])ه بود(یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: بیامرزیده بودم، بیامرزیده بودم، بیامرزیده بود، ...	ماضی بعید معلوم مثبت دوم
G3IN1	(ن|م)([BG])ه بود(یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: بگفته بودم، بگفته بودی، بگفته بود، ... (ن|م)(ی[AG])ه بود(یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: بیامرزیده بودم، بیامرزیده بودم، بیامرزیده بود، ...	ماضی بعید معلوم منفی یکم
G3I	(ن|م|ب)؟[G]ه بود[PFGS][Z]	ماضی بعید معلوم

در ماضی بعید معلوم منفی چسباندن «ن» به ابتدای «بود»، مانند: «گفته نبودم»، غلط است.

برخی الگوهای صَرفی قدیمی در الگوی فوق لحاظ نشده است.

بین اجزای فعل کلمه دیگری قرار نمی‌گیرد

۳-۲-۲-۴-۶ *ماضی بعید مجهول*

الگوی صَرف فعل ماضی بعید مجهول در جدول (۳-۲۳) نشان داده شده است.

جدول (۳-۲۳) الگوی صَرف فعل ماضی بعید مجهول

کد	عبارت منظم	توضیحات
G3TP1	[G]ه شده بود(یم|ید|ند|م|ی)؟ مثال: گفته شده بودم، گفته شده بودی، گفته شده بود، ...	ماضی بعید مجهول مثبت یکم
G3TP2	ب[BG]ه شده بود(یم|ید|ند|م|ی)؟ مثال: بگفته شده بودم، بگفته شده بودی، بگفته شده بود، ... ب(ی[AG])ه شده بود(یم|ید|ند|م|ی)؟ مثال: بیامرزیده شده بودم، بیامرزیده شده بودم، بیامرزیده شده بودم، ...	ماضی بعید مجهول مثبت دوم
G3TN1	[G]ه نشده بود(یم|ید|ند|م|ی)؟ مثال: گفته نشده بودم، گفته نشده بودی، گفته نشده بود، ...	ماضی بعید مجهول منفی یکم
G3TN2	(ن|م)[BG]ه شده بود(یم|ید|ند|م|ی)؟ مثال: نگفته شده بودم، نگفته شده بودی، نگفته شده بود، ... (ن|م)(ی[AG])ه شده بود(یم|ید|ند|م|ی)؟ مثال: نیامرزیده شده بودم، نیامرزیده شده بودم، نیامرزیده شده بودم، ...	ماضی بعید مجهول منفی دوم
G3T	(ب|ن|م)؟[BG]ه (ب|ن)؟[IMG]ه بود[PFGS] ((ب|ن|م)ی)؟[AG]ه (ب|ن)؟[IMG]ه بود[PFGS]	ماضی بعید مجهول

چسباندن «ن» نفی به ابتدای «بود» غلط است.

چسباندن «ب» به ابتدای «شد» مجاز است و در الگوی کلی لحاظ شده است.

بین اجزای فعل کلمه دیگری قرار نمی‌گیرد.

۳-۲-۲-۴-۷ ماضی مستمر معلوم

الگوی صَرف فعل ماضی مستمر معلوم در جدول (۳-۲۴) نشان داده شده است.

جدول (۳-۲۴) الگوی صَرف فعل ماضی مستمر معلوم

کد	عبارت منظم	توضیحات
G4TP1	داشت(یم|ید|ند|م|ی)؟ {D}می◊[G](یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: داشتم می گفتم، داشتی می گفتی، داشت می گفت، ...	ماضی مستمر معلوم مثبت یکم
G4TN1	داشت(یم|ید|ند|م|ی)؟ {D}نمی◊[G](یم|ید|ند|م|ی)؟(م|ت|ش|مان|تان|شان)؟ مثال: داشتم نمی گفتم، داشتی نمی گفتی، داشت نمی گفت، ...	ماضی مستمر معلوم منفی یکم
G4T	داشت[PFGS]{D}(ن)؟می◊[G][PFGS][Z]	ماضی مستمر معلوم

بین اجزای فعل فاصله می‌افتد؛ مانند: «**داشتم** سیب را **می‌خوردم**». در الگوی فوق {D} معرف این فاصله است.
پسوند شخص باید در دو جزء فعل یکسان باشد.

۳-۲-۲-۴-۸ ماضی مستمر مجهول

الگوی صَرف فعل ماضی مستمر مجهول در جدول (۳-۲۵) نشان داده شده است.

جدول (۳-۲۵) الگوی صَرف فعل ماضی مستمر مجهول

کد	عبارت منظم	توضیحات
G5IP1	داشت(یم|ید|ند|م|ی)؟ {D}[G]ه می◊شد(یم|ید|ند|م|ی)؟ مثال: داشتم گفته می‌شدم، داشتی گفته می‌شدی، داشت گفته می‌شد، ...	ماضی مستمر مجهول مثبت یکم
G5IN1	داشت(یم|ید|ند|م|ی)؟ {D}[G]ه نمی◊شد(یم|ید|ند|م|ی)؟ مثال: داشتم گفته نمی‌شدم، داشتی گفته نمی‌شدی، داشت گفته نمی‌شد، ...	ماضی مستمر مجهول منفی یکم
G5I	داشت[PFGS]{D}[G]ه (ن)؟می◊[IMG][PFGS]	ماضی مستمر مجهول

چسباندن «ن» نفی به ابتدای بن فعل غلط است.
بین اجزای فعل کلمات دیگری قرار می گیرند. در الگو این فاصله با {D} نشان داده شده است.
پسوند شخص باید در دو جزء فعل یکسان باشد.

۳-۲-۲-۴-۹ *ماضی ساده نقلی معلوم*

الگوی صَرف فعل ماضی ساده نقلی معلوم در جدول (۳-۲۶) نشان داده شده است.

جدول (۳-۲۶) الگوی صَرف فعل ماضی ساده نقلی معلوم

کد	عبارت منظم	توضیحات
G5IP1	[G]ه(ا|ایم|اید|اند|ام|ای| است)؟(م|ات|اش|امان|اتان|اشان)؟ مثال: گفته‌ام، گفته‌ای، گفته است، ...	ماضی ساده نقلی معلوم مثبت یکم
G5IP2	ب[BG]ه(ا|ایم|اید|اند|ام|ای| است)؟(م|ات|اش|امان|اتان|اشان)؟ مثال: گفته‌ام، گفته‌ای، گفته است، ... ب(ی[AG])ه(ا|ایم|اید|اند|ام|ای| است)؟(م|ات|اش|امان|اتان|اشان)؟ مثال: بیامرزیده‌ام، بیامرزیده‌ای، بیامرزیده است، ...	ماضی ساده نقلی معلوم مثبت دوم
G5IN1	(ن|م)[BG]ه(ا|ایم|اید|اند|ام|ای| است)؟(م|ات|اش|امان|اتان|اشان)؟ مثال: نگفته‌ام، نگفته‌ای، نگفته است، ... (ن|م)(ی[AG])ه(ا|ایم|اید|اند|ام|ای| است)؟(م|ات|اش|امان|اتان|اشان)؟ مثال: نیامرزیده‌ام، نیامرزیده‌ای، نیامرزیده است، ...	ماضی ساده نقلی معلوم منفی یکم
G5I	(ن|م|ب)؟[BG]ه[PFGN][Z] ((ن|م|ب)ی)؟[AG]ه[PFGN][Z]	ماضی ساده نقلی معلوم

در سوم شخص مفرد «است» می‌تواند حذف شود؛ مثلاً: «گفته» به جای «گفته است».
حالت‌های صَرفی قدیمی در این الگو لحاظ نشده است.

۳-۲-۲-۴-۱۰ *ماضی ساده نقلی مجهول*

الگوی صَرف فعل ماضی ساده نقلی مجهول در جدول (۳-۲۷) نشان داده شده است.

جدول (۳-۲۷) الگوی صَرف فعل ماضی ساده نقلی مجهول

کد	عبارت منظم	توضیحات
G5TP1	[G]ه شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: گفته شده‌ام، گفته شده‌ای، گفته شده است، ...	ماضی ساده نقلی مجهول مثبت یکم
G5TP2	ب[BG]ه شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: گفته شده‌ام، گفته شده‌ای، گفته شده است، ... ب(ی[AG])ه شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: بیامرزیده شده‌ام، بیامرزیده شده‌ای، بیامرزیده شده است، ...	ماضی ساده نقلی مجهول مثبت دوم
G5TN1	[G]ه نشده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: گفته نشده‌ام، گفته نشده‌ای، گفته نشده است، ...	ماضی ساده نقلی مجهول منفی یکم
G5TN2	(ن|م)[BG]ه شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: نگفته شده‌ام، نگفته شده‌ای، نگفته شده است، ... (ن|م)(ی[AG])ه شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: نیامرزیده شده‌ام، نیامرزیده شده‌ای، نیامرزیده شده است، ...	ماضی ساده نقلی مجهول منفی دوم
G5T	(ن|م|ب)؟[BG]ه (ب|ن)؟[IMG]ه[PFGN] ((ن|م|ب)ی)؟[AG]ه (ب|ن)؟[IMG]ه[PFGN]	ماضی ساده نقلی مجهول

در سوم شخص مفرد «است» می‌تواند حذف شود.

چسباندن «ب» به ابتدای «شد» مجاز است که در الگوی کلی لحاظ شده است.

۳-۲-۲-۴-۱۱ *ماضی استمراری نقلی مجهول*

الگوی صَرف فعل ماضی استمراری نقلی مجهول در جدول (۳-۲۸) نشان داده شده است.

جدول (۳-۲۸) الگوی صَرف فعل ماضی استمراری نقلی مجهول

کد	عبارت منظم	توضیحات
G6TP1	[G]ه می◊شده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟ مثال: گفته می‌شده‌ام، گفته می‌شده‌ای، گفته می‌شده‌است، ...	ماضی استمراری نقلی مجهول مثبت یکم
G6TP2	ب[BG]ه می◊شده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟ مثال: بگفته می‌شده‌ام، بگفته می‌شده‌ای، بگفته می‌شده است، ... ب(ی[AG])ه می◊شده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟ مثال: بیامرزیده می‌شده‌ام، بیامرزیده می‌شده‌ای، بیامرزیده می‌شده است، ...	ماضی استمراری نقلی مجهول مثبت دوم
G6TN1	[G]ه نمی◊شده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟ مثال: گفته نمی‌شده‌ام، گفته نمی‌شده‌ای، گفته نمی‌شده است، ...	ماضی استمراری نقلی مجهول منفی یکم
G6TN2	(ن|م)[BG]ه می◊شده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟ مثال: نگفته می‌شده‌ام، نگفته می‌شده‌ای، نگفته می‌شده است، ... (ن|م)(ی[AG]ه) می◊شده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟ مثال: نیامرزیده می‌شده‌ام، نیامرزیده می‌شده‌ای، نیامرزیده می‌شده است، ...	ماضی استمراری نقلی مجهول منفی دوم
G6T	(ن|م|ب)؟[BG]ه (ن)؟می◊[IMG]ه[PFGN] ((ن|م|ب)؟ی)[AG]ه (ن)؟می◊[IMG]ه[PFGN]	ماضی استمراری نقلی مجهول

حالت‌های صَرفی قدیمی با پیشوند «همی» در این الگو لحاظ نشده است.
در سوم شخص مفرد «است» می‌تواند حذف شود.
بین اجزای فعل کلمه دیگری قرار نمی‌گیرد.

۳-۲-۲-۴-۱۲ *ماضی استمراری نقلی معلوم*

الگوی صَرف فعل ماضی استمراری نقلی معلوم در جدول (۳-۲۹) نشان داده شده است.

جدول (۳-۲۹) الگوی صَرف فعل ماضی استمراری نقلی معلوم

کد	عبارت منظم	توضیحات
G6IP1	می◊[G]ه(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: می گفته‌ام، می گفته‌ای، می گفته است، ...	ماضی استمراری نقلی معلوم مثبت یکم
G6IN1	نمی◊[G]ه(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: نمی گفته‌ام، نمی گفته‌ای، نمی گفته است، ...	ماضی استمراری نقلی معلوم منفی یکم
G6I	(ن)؟می◊[G]ه[PFGN][Z]	ماضی استمراری نقلی معلوم

حالت‌های صَرفی قدیمی با پیشوند «همی» در این الگو لحاظ نشده است.

در سوم شخص مفرد «است» می‌تواند حذف شود.

۳-۲-۲-۴-۱۳ *ماضی بعید نقلی معلوم*

الگوی صَرف فعل ماضی بعید نقلی معلوم در جدول (۳-۳۰) نشان داده شده است.

جدول (۳-۳۰) الگوی صَرف فعل ماضی بعید نقلی معلوم

کد	عبارت منظم	توضیحات
G7IP1	[G]ه بوده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: گفته بوده‌ام، گفته بوده‌ای، گفته بوده است، ...	ماضی بعید نقلی معلوم مثبت یکم
G7IP2	ب[BG]ه بوده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: بگفته بوده‌ام، بگفته بوده‌ای، بگفته بوده است، ... ب(ی[AG])ه بوده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: بیامرزیده بوده‌ام، بیامرزیده بوده‌ای، بیامرزیده بوده است، ...	ماضی بعید نقلی معلوم مثبت دوم
G7IN1	(ن|م)[BG]ه بوده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: نگفته بوده‌ام، نگفته بوده‌ای، نگفته بوده است، ... (ن|م)(ی[AG])ه بوده(◊ایم|◊اید|◊اند|◊ام|◊ای| است)؟(م|ت|ش|مان|تان|شان)؟ مثال: نیامرزیده بوده‌ام، نیامرزیده بوده‌ای، نیامرزیده بوده است، ...	ماضی بعید نقلی معلوم منفی یکم
G7I	(ن|م|ب)؟[BG]ه بوده[PFGN][Z] ((ن|م|ب)؟ی)[AG]ه بوده[PFGN][Z]	ماضی بعید نقلی معلوم

اتصال «ن» به «بود» برای منفی کردن، نادرست است.

بین اجزای فعل کلمه دیگری قرار نمی‌گیرد.

۳-۲-۲-۴-۱۴ *ماضی بعید نقلی مجهول*

الگوی صَرف فعل ماضی بعید نقلی مجهول در جدول (۳-۳۱) نشان داده شده است.

جدول (۳-۳۱) الگوی صَرف فعل ماضی بعید نقلی مجهول

کد	عبارت منظم	توضیحات
G7TP1	[G]ه شده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: گفته شده بوده‌ام، گفته شده بوده‌ای، گفته شده بوده است، ...	ماضی بعید نقلی مجهول مثبت یکم
G7TP2	ب|[BG]ه شده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: بگفته شده بوده‌ام، بگفته شده بوده‌ای، بگفته شده بوده است، ... ب(ی[AG])ه شده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: بیامرزیده شده بوده‌ام، بیامرزیده شده بوده‌ای، بیامرزیده شده بوده است، ...	ماضی بعید نقلی مجهول مثبت دوم
G7TP3	[G]ه بشده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: گفته بشده بوده‌ام، گفته بشده بوده‌ای، گفته بشده بوده است، ...	ماضی بعید نقلی مجهول مثبت سوم
G7TN1	[G]ه نشده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: گفته نشده بوده‌ام، گفته نشده بوده‌ای، گفته نشده بوده است، ...	ماضی بعید نقلی مجهول منفی یکم
G7TN2	(ن|م)[G]ه شده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ (ن|م)[BG]ه شده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: نگفته شده بوده‌ام، نگفته شده بوده‌ای، نگفته شده بوده است، ... (ن|م)(ی[AG])ه شده بوده(◌ٔایم|◌ٔاید|◌ٔاند|◌ٔام|◌ٔای| است)؟ مثال: نیامرزیده شده بوده‌ام، نیامرزیده شده بوده‌ای، نیامرزیده شده بوده است، ...	ماضی بعید نقلی مجهول منفی دوم
G7T	(ن|م|ب)؟[BG]ه (ب|ن)؟[IMG]ه بوده[PFGN] ((ن|م|ب)؟ی)[AG]ه (ب|ن)؟[IMG]ه بوده[PFGN]	ماضی بعید نقلی مجهول

بین اجزای فعل کلمه دیگری قرار نمی‌گیرد.

چسباندن «ن» به ابتدای «بود» برای منفی سازی غلط است.

در سوم شخص مفرد «است» می‌تواند حذف شود.

۳-۲-۲-۴-۱۵ *ماضی مستمر نقلی معلوم*

الگوی صَرف فعل ماضی مستمر نقلی معلوم در جدول (۳-۳۲) نشان داده شده است.

جدول (۳-۳۲) الگوی صَرف فعل ماضی مستمر نقلی معلوم

کد	عبارت منظم	توضیحات
G8IP1	داشته(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ {D}می‌ا[G]ه(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: داشه‌ام می‌گفته‌ام، داشته‌ای می‌گفته‌ای، داشته است می‌گفته است، ...	ماضی مستمر نقلی معلوم مثبت یکم
G8IN1	داشته(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ {D}نمی‌ا[G]ه(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: داشه‌ام نمی‌گفته‌ام، داشته‌ای نمی‌گفته‌ای، داشته است نمی‌گفته است، ...	ماضی مستمر نقلی معلوم منفی یکم
G8I	داشته[PFGN]{D}(ن)؟می‌ا[G]ه[PFGN]	ماضی مستمر نقلی معلوم

بین اجزای فعل کلمات دیگری قرار می‌گیرد.

در سوم شخص مفرد «است» می‌تواند حذف شود.

پسوند ضمیر در دو جزء فعل باید یکسان باشد. برای سادگی الگو را به صورت فوق نوشتیم.

۳-۲-۲-۴-۱۶ *ماضی مستمر نقلی مجهول*

الگوی صَرف فعل ماضی مستمر نقلی مجهول در جدول (۳-۳۳) نشان داده شده است.

جدول (۳-۳۳) الگوی صَرف فعل ماضی مستمر نقلی مجهول

کد	عبارت منظم	توضیحات
G8TP1	داشته(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ {D}[G]ه می‌شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: داشته‌ام گفته می‌شده‌ام، داشته‌ای گفته می‌شده‌ای، داشته است گفته می‌شده است، ...	ماضی مستمر نقلی مجهول مثبت یکم
G8TN1	داشته(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ {D}[G]ه نمی‌شده(ا‌یم|ا‌ید|ا‌ند|ام|ای| است)؟ مثال: داشته‌ام گفته نمی‌شده‌ام، داشته‌ای گفته نمی‌شده‌ای، داشته است گفته نمی‌شده است،	ماضی مستمر نقلی مجهول منفی یکم
G8T	داشته[PFGN]{D}[G]ه (ن)؟می‌ا[IMG]ه[PFGN]	ماضی مستمر نقلی مجهول

بین اجزای فعل کلمات دیگری قرار می‌گیرد.

در سوم شخص مفرد «است» می‌تواند حذف شود.

پسوند ضمیر در دو جزء فعل باید یکسان باشد. برای سادگی الگو را به صورت فوق نوشتیم.

این زمان خیلی کم مورد استفاده قرار می‌گیرد.

۳-۲-۲-۴-۱۷ *ماضی التزامی معلوم*

الگوی صَرف فعل ماضی التزامی معلوم در جدول (۳-۳۴) نشان داده شده است.

جدول (۳-۳۴) الگوی صَرف فعل ماضی التزامی معلوم

کد	عبارت منظم	توضیحات
G9IP1	[G]ه باش(یم|ید|ند|م|ای)؟(م|ات|اش|مان|تان|شان)؟ مثال: گفته باشم، گفته باشی، گفته باشد، ...	ماضی التزامی معلوم مثبت یکم
G9IP2	ب[BG]ه باش(یم|ید|ند|م|ای)؟(م|ات|اش|مان|تان|شان)؟ مثال: گفته باشم، گفته باشی، گفته باشد، ... ب(ی[AG])ه باش(یم|ید|ند|م|ای)(م|ات|اش|مان|تان|شان)؟ مثال: بیازرده باشم، بیازرده باشی، بیازرده باشد، ...	ماضی التزامی معلوم مثبت دوم
G9IN1	(ن|م)[BG]ه باش(یم|ید|ند|م|ای)؟(م|ات|اش|مان|تان|شان)؟ مثال: نگفته باشم، نگفته باشی، نگفته باشد، ... (ن|م)(ی[AG])ه باش(یم|ید|ند|م|ای)؟(م|ات|اش|مان|تان|شان)؟ مثال: نیازرده باشم، نیازرده باشی، نیازرده باشد، ...	ماضی التزامی معلوم منفی یکم
G9IN2	[G]ه نباش(یم|ید|ند|م|ای)؟(م|ات|اش|مان|تان|شان)؟ مثال: گفته نباشم، گفته نباشی، گفته نباشد، ...	ماضی التزامی معلوم منفی دوم
G9I	(ن|م|ب)؟[G]ه (ن)؟باش[PFGS][Z]	ماضی التزامی معلوم

بین اجزای فعل کلمه دیگری قرار نمی‌گیرد.

۳-۲-۲-۴-۱۸ *ماضی التزامی مجهول*

الگوی صَرف فعل ماضی التزامی مجهول در جدول (۳-۳۵) نشان داده شده است.

جدول (۳-۳۵) الگوی صَرف فعل ماضی التزامی مجهول

کد	عبارت منظم	توضیحات
G9TP1	[G]ه شده باش(یم|ید|ند|م|ی)؟ مثال: گفته شده باشم، گفته شده باشی، گفته شده باشد، ...	ماضی التزامی مجهول مثبت یکم
G9TP2	ب[BG]ه شده باش(یم|ید|ند|م|ی)؟ مثال: بگفته شده باشم، بگفته شده باشی، بگفته شده باشد، ... ب(ی[AG])ه شده باش(یم|ید|ند|م|ی)؟ مثال: بیامرزیده شده باشم، بیامرزیده شده باشی، بیامرزیده شده باشد، ...	ماضی التزامی مجهول مثبت دوم
G9TP3	[G]ه بشده باش(یم|ید|ند|م|ی)؟ مثال: گفته بشده باشم، گفته بشده باشی، گفته بشده باشد، ...	ماضی التزامی مجهول مثبت سوم
G9TN1	(ن|م)([BG]ه شده باش(یم|ید|ند|م|ی)؟ مثال: نگفته شده باشم، نگفته شده باشی، نگفته شده باشد، ... (ن|م)(ی[AG])ه شده باش(یم|ید|ند|م|ی)؟ مثال: نیامرزیده شده باشم، نیامرزیده شده باشی، نیامرزیده شده باشد، ...	ماضی التزامی مجهول منفی یکم
G9TN2	[G]ه نشده باش(یم|ید|ند|م|ی)؟ مثال: گفته نشده باشم، گفته نشده باشی، گفته نشده باشد، ...	ماضی التزامی مجهول منفی دوم
G9T	(ب|ن|م)؟[BG]ه (ب|ن|م)؟[IMG]ه باش[PFGS] ((ب|ن|م)؟ی)[AG]ه (ب|ن|م)؟[IMG]ه باش[PFGS]	ماضی التزامی مجهول

بین اجزای فعل کلمه دیگری قرار نمی‌گیرد.

«م» می‌تواند به جای «ن» برای منفی کردن استفاده شود که در الگوی کلی لحاظ شده است.

۳-۲-۲-۴-۱۹ *مضارع اخباری معلوم*

الگوی صَرف مضارع اخباری معلوم در جدول (۳-۳۶) نشان داده شده است.

جدول (۳-۳۶) الگوی صَرف فعل مضارع اخباری معلوم

کد	عبارت منظم	توضیحات
H1IP1	می(\[HA]ی|\[HB])(یم|ید|ند|م|ی|د)(م|ت|ش|مان|تان|شان)؟ مثال: می‌گویم، می‌گویی، می‌گوید، ...	مضارع اخباری معلوم مثبت یکم
H1IN1	نمی(\[HA]ی|\[HB])(یم|ید|ند|م|ی|د)(م|ت|ش|مان|تان|شان)؟ مثال: نمی‌گویم، نمی‌گویی، نمی‌گوید، ...	مضارع اخباری معلوم منفی یکم
H1I	(ن)؟می(\[HA]ی|\[HB])\[PFH]	مضارع اخباری معلوم

بن‌هایی که به الف ختم می‌شوند، به انتهای آن‌ها «ی» افزوده می‌گردد.
الگوهای صَرف قدیمی با «همی» در این الگو لحاظ نشده است.

۳-۲-۲-۴-۲۰ *مضارع اخباری مجهول*

الگوی صَرف فعل مضارع اخباری مجهول در جدول (۳-۳۷) نشان داده شده است.

جدول (۳-۳۷) الگوی صَرف فعل مضارع اخباری مجهول

کد	عبارت منظم	توضیحات
H1TP1	\[G]ه می◊شو(یم|ید|ند|م|ی|د) مثال: گفته می‌شوم، گفته می‌شوی، گفته می‌شود، ...	مضارع اخباری مجهول مثبت یکم
H1TN1	\[BG]ه نمی◊شو(یم|ید|ند|م|ی|د) مثال: گفته می‌شوم، گفته می‌شوی، گفته می‌شود، ...	مضارع اخباری مجهول منفی یکم
H1T	(ب|ن|م)؟\[BG]ه (ن)؟می◊\[IMH]\[PFH] ((ب|ن|م)؟ی)\[AG]ه (ن)؟می◊\[IMH]\[PFH]	مضارع اخباری مجهول

بین اجزای فعل فاصله نمی‌افتد.

۳-۲-۲-۴-۲۱ *مضارع مستمر معلوم*

الگوی صَرف فعل مضارع مستمر معلوم در جدول (۳-۳۸) نشان داده شده است.

جدول (۳-۳۸) الگوی صَرف فعل مضارع مستمر معلوم

کد	عبارت منظم	توضیحات
H2IP1	دار(یم|ید|ند|م|ی|د) {D}می([HB])(یم|ید|ند|م|ی|د)(م|ت|ش|مان|تان|شان)؟ مثال: دارم می گویم، داری می گویی، دارد می گوید، ... دار(یم|ید|ند|م|ی|د) {D}می([HA]ی)(یم|ید|ند|م|ی|د)(م|ت|ش|مان|تان|شان)؟ مثال: دارم می گشایم، داری می گشایی، دارد می گشاید، ...	مضارع مستمر معلوم مثبت یکم
H2IN1	دار(یم|ید|ند|م|ی|د) {D}نمی([HB])(یم|ید|ند|م|ی|د)(م|ت|ش|مان|تان|شان)؟ مثال: دارم نمی گویم، داری نمی گویی، دارد نمی گوید، ... دار(یم|ید|ند|م|ی|د) {D}نمی([HA]ی)(یم|ید|ند|م|ی|د)(م|ت|ش|مان|تان|شان)؟ مثال: دارم نمی گشایم، داری نمی گشایی، دارد نمی گشاید، ...	مضارع مستمر معلوم منفی یکم
H2I	دار[PFH]{D}ن؟می([HA]ی|[HB])[PFH][Z]	مضارع مستمر معلوم

پسوند شخص در دو جزء فعل باید یکسان باشد. برای سادگی الگوی آن را به صورت فوق نشان دادیم.

۳-۲-۲-۴-۲۲ *مضارع مستمر مجهول*

الگوی صَرف فعل مضارع مستمر مجهول در جدول (۳-۳۹) نشان داده شده است.

جدول (۳-۳۹) الگوی صَرف فعل مضارع مستمر مجهول

کد	عبارت منظم	توضیحات
H2TP1	دار(یم|ید|ند|م|ی|د) {D}[G]ه می◊شو(یم|ید|ند|م|ی|د)	مضارع مستمر مجهول مثبت یکم
H2TN1	دار(یم|ید|ند|م|ی|د) {D}[G]ه نمی◊شو(یم|ید|ند|م|ی|د)	مضارع مستمر مجهول منفی یکم
H2T	دار[PFH]{D}[G]ه (ن)؟می◊[IMH][PFH]	مضارع مستمر مجهول

بین اجزای فعل فاصله می‌افتد که توسط {D} مشخص شده است.

پسوند شخص در دو جزء فعل باید یکسان باشد. برای سادگی الگو آن را به صورت فوق نشان دادیم.

۳-۲-۲-۴-۲۳ *مضارع التزامی معلوم*

الگوی صَرف فعل مضارع التزامی معلوم در جدول (۳-۴۰) نشان داده شده است.

جدول (۳-۴۰) الگوی صَرف فعل مضارع التزامی معلوم

کد	عبارت منظم	توضیحات
H3IP1	ب(ی[AHA]ای|ی[AHB]||[BHA]ی||[BHB])(یم|ید|ند|م|ای|اد)(م|ات|ش|مان|تان|شان)؟	مضارع التزامی معلوم مثبت یکم
H3IP2	([BHA]ی||[BHB])(یم|ید|ند|م|ای|اد)(م|ات|ش|مان|تان|شان)؟	مضارع التزامی معلوم مثبت دوم
H3IN1	ن(ی[AHA]ای|ی[AHB]||[BHA]ی||[BHB])(یم|ید|ند|م|ای|اد)(م|ات|ش|مان|تان|شان)؟	مضارع التزامی معلوم منفی یکم
H3I	(ب|ن|م)؟(ی[AHA]ای|ی[AHB]||[BHA]ی||[BHB])[PFH][Z]	مضارع التزامی معلوم

«م» برای منفی سازی فعل به جای «ن» استفاده می‌شود که در الگوی کلی لحاظ شده است.

در بنهایی که با الف شروع می‌شوند، در صورت افزودن «ب» به ابتدا، «ی» به ابتدای آنها اضافه می‌شود؛ مانند: بن «آزار» که می‌شود «بیازار». در بنهایی که به الف ختم می‌شوند، «ی» به انتهای آنها افزوده می‌شود؛ مانند: بن «زدا» که می‌شود «بزدایم». فعل‌هایی که با الف شروع و خاتمه می‌یابند، در هر دو طرف «ی» می‌گیرند؛ مانند: بن «آسا» که می‌شود «بیاساید».

در بن‌هایی که با «آ» شروع می‌شوند وقتی «ب» افزوده می‌شود، «آ» به «ا» تبدیل می شود.

۳-۲-۲-۴-۲۴ *مضارع التزامی مجهول*

الگوی صَرف فعل مضارع التزامی مجهول در جدول (۳-۴۱) نشان داده شده است.

جدول (۳-۴۱) الگوی صَرف فعل مضارع التزامی مجهول

کد	عبارت منظم	توضیحات
H3TP1	[G]ه شو(یم|ید|ند|م|ی|د) مثال: گفته شوم، گفته شوی، گفته شود، ...	مضارع التزامی مجهول مثبت یکم
H3TP2	[G]ه بشو(یم|ید|ند|م|ی|د)	مضارع التزامی مجهول مثبت دوم
H3TP3	ب[BG]ه شو(یم|ید|ند|م|ی|د) ب(ی[AG])ه شو(یم|ید|ند|م|ی|د)	مضارع التزامی مجهول مثبت سوم
H3TN1	[G]ه نشو(یم|ید|ند|م|ی|د)	مضارع التزامی مجهول منفی یکم
H3TN2	(ن|م)[BG]ه شو(یم|ید|ند|م|ی|د) (ن|م)(ی[AG])ه شو(یم|ید|ند|م|ی|د)	مضارع التزامی مجهول منفی دوم
H3T	(ب|ن|م)؟[BG]ه (ب|ن|م)؟[IMH][PFH] ((ب|ن|م)؟ی)[AG]ه (ب|ن|م)؟[IMH][PFH]	مضارع التزامی مجهول

بین اجزای فعل فاصله نمی‌افتد.

۳-۲-۲-۴-۲۵ *آینده معلوم*

الگوی صَرف فعل آینده معلوم در جدول (۳-۴۲) نشان داده شده است.

جدول (۳-۴۲) الگوی صَرف فعل آینده معلوم

کد	عبارت منظم	توضیحات
A1IP1	خواه(یم|ید|ند|م|ی|د) [G](م|ت|ش|مان|تان|شان)؟ مثال: خواهم گفت، خواهی گفت، خواهد گفت، ...	آینده معلوم مثبت یکم
A1IP2	خواه(یم|ید|ند|م|ی|د) ب[G](م|ت|ش|مان|تان|شان)؟	آینده معلوم مثبت دوم
A1IP3	بخواه(یم|ید|ند|م|ی|د) [G](م|ت|ش|مان|تان|شان)؟	آینده معلوم مثبت سوم
A1IN1	نخواه(یم|ید|ند|م|ی|د) [G](م|ت|ش|مان|تان|شان)؟	آینده معلوم منفی یکم
A1I	(ب|ن|م)؟خواه[PFH] (ب|ن|م)؟[G][Z]	آینده معلوم

بین اجزای فعل فاصله نمی‌افتد. در صورت ایجاد فاصله، «خواستن» فعل کمکی حساب می‌شود.

۳-۲-۲-۴-۲۶ *آینده مجهول*

الگوی صَرف فعل آینده مجهول در جدول (۳-۴۳) نشان داده شده است.

جدول (۳-۴۳) الگوی صَرف فعل آینده مجهول

کد	عبارت منظم	توضیحات
A1TP1	[G]ه خواه(یم|ید|ند|م|ی|د) شد مثال: گفته خواهم شد، گفته خواهی شد، گفته خواهد شد، ...	آینده مجهول مثبت یکم
A1TP2	ب[G]ه خواه(یم|ید|ند|م|ی|د) شد	آینده مجهول مثبت دوم
A1TP3	[G]ه بخواه(یم|ید|ند|م|ی|د) شد	آینده مجهول مثبت سوم
A1TN1	[G]ه نخواه(یم|ید|ند|م|ی|د) شد	آینده مجهول منفی یکم
A1TN2	ن[G]ه خواه(یم|ید|ند|م|ی|د) شد	آینده مجهول منفی دوم
A1T	(ب|ن|م)؟[G]ه (ب|ن|م)؟خواه[PFH] [IMG]	آینده مجهول

اتصال «ن» به ابتدای «شد» برای منفی سازی اشتباه است.

از «م» به جای «ن» می‌توان برای منفی کردن استفاده کرد.

بین اجزای فعل فاصله نمی‌افتد.

۳-۲-۲-۴-۲۷ *امر معلوم*

الگوی صَرف فعل امر معلوم در جدول (۳-۴۴) نشان داده شده است.

جدول (۳-۴۴) الگوی صَرف فعل امر معلوم

کد	عبارت منظم	توضیحات
A2IP1	ب(ی[AHA]ی؟|ی[AHB]|[BHA]ی؟|[BHB]) مثال: بگو(ی)، بگویید	امر معلوم مثبت یکم
A2IN1	([AHB]ی|[BHB])	امر معلوم منفی یکم
A2IN1	(ن|م)(ی[AHA]ی؟|ی[AHB]|[BHA]ی؟|[BHB])	امر معلوم منفی یکم
A2I	(ب|ن|م)؟(ی[AHA]ی؟|ی[AHB]|[BHA]ی؟|[BHB])	امر معلوم

در بن‌هایی که با الف شروع می‌شوند هنگام اتصال «ب، ن، م» به ابتدای آن، «ی» به اول بن افزوده می‌شود و «آ» به «ا» تبدیل می‌شود؛ مانند: «بیاویز» از بن «آویز».

در بن‌هایی که به الف ختم می‌شود، می‌توان «ی» به انتهای آن اضافه کرد. مانند: «بزدای» از بن «زدا».

در بن‌هایی که به «ی» ختم می‌شوند می‌توان «ی» را حذف کرد (اختیاری است). مانند: «بگوی» و «بگو» از بن «گوی».

۳-۲-۲-۴-۲۸ *امر مجهول*

الگوی صَرف فعل امر مجهول در جدول (۳-۴۵) نشان داده شده است.

جدول (۳-۴۵) الگوی صَرف فعل مضارع امر مجهول

کد	عبارت منظم	توضیحات
A2TP1	[G]ه بشو(ید)؟ مثال: گفته بشو، گفته بشوید	امر مجهول مثبت یکم
A2TP2	[G]ه شو(ید)؟	امر مجهول مثبت دوم
A2TN1	[G]ه نشو(ید)؟	امر مجهول منفی یکم
A2T	[G]ه (ب|ن|م)شو(ید)؟	امر مجهول

می‌توان از «م» برای نهی استفاده کرد.

۳-۲-۳ فاصله‌گذاری

در زبان‌های هند و اروپایی، متن معمولاً با واژه‌ها که با فاصله از یکدیگر جدا شده‌اند شکل می‌گیرد. زبان فارسی علاوه بر فاصله‌ی میانِ واژه‌ها، نوع دیگری از فاصله که میان اجزای یک واژه قرار می‌گیرد را نیز شامل می‌شود. به نظر می‌رسد که استفاده از این فاصله‌ی میانِ اجزای یک واژه که اخیراً رواج یافته، ناشی از ناتوانی خط فارسی در انتقال کارای مفاهیم باشد. این فاصله‌ی میان اجزای واژه به *نیم‌فاصله* یا *شبه‌فاصله* موسوم است.

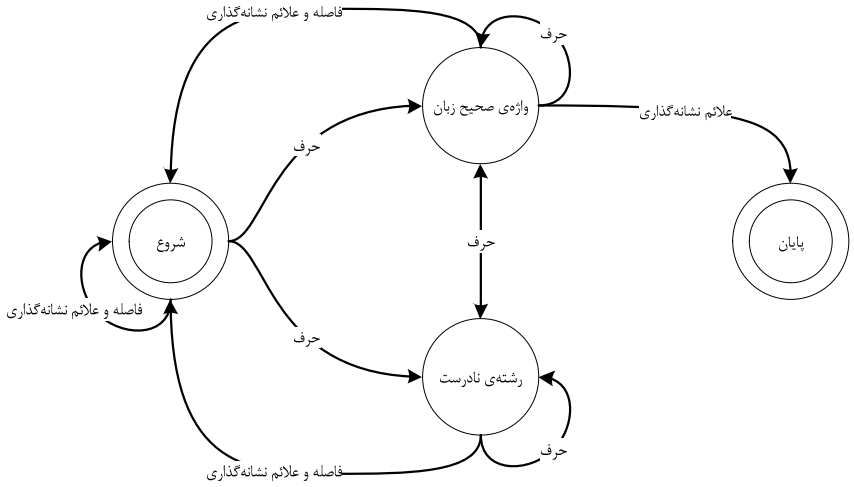
چالشی که ورود نیم‌فاصله به شیوه‌ی نگارش خط فارسی بیش از پیش بر آن دامن زده است، ابهام در پیوسته‌نویسی و جدانویسی واژه‌هایی است که ترکیبی از چند تکواژ هستند. جدای از اشکالات دستور خطِ رسمی و موارد ابهام آن که در فصل ۲ به آن‌ها اشاره شد، در حال حاضر نگارندگان به زبان فارسی، در هر سطحی از تحصیلات، به طور کاملاً سلیقه‌ای اقدام به پیوسته‌نویسی، استفاده از نیم‌فاصله، یا جدا نویسی می‌کنند. جهت احراز میزان اهمیت این چالش در زبان فارسی، بسامد گونه‌های مختلف نوشتار چند واژه بر اساس انواع فاصله‌گذاری در جدول (۳-۴۶) آورده شده است. این بسامدها از موتور جستجوی گوگل، به عنوان یک منبع دانش بسیار گسترده که می‌تواند نماینده‌ی معتبری برای بسامدِ کاربردِ واژه‌ها میان نگارنگان آن باشد، استخراج شده‌اند. بسامد بالا و نزدیک

انواع گونه‌های نوشتار واژه‌ها، خصوصاً بسامد بالاتر گونه‌های نوشتاری غلط، نشان دهنده‌ی کاربرد گسترده‌ی انواع گونه‌های نوشتاری و فاصله‌گذاری میان فارسی‌زبانان است. توجه به این نکته ضروری است که از دیدگاه پردازشی، گونه‌های مختلف نوشتار یک واژه، واژه‌هایی متفاوت هستند و از این رو تنوع در فاصله‌گذاری و نوشتار یک واژه‌ی یکسان، یک چالش جدی در پردازش زبان فارسی و خطایابی املایی آن است.

جدول (۳-۴۶) بسامد گونه‌های مختلف نوشتار واژه‌ها و فاصله‌گذاری میان اجزاء آن‌ها

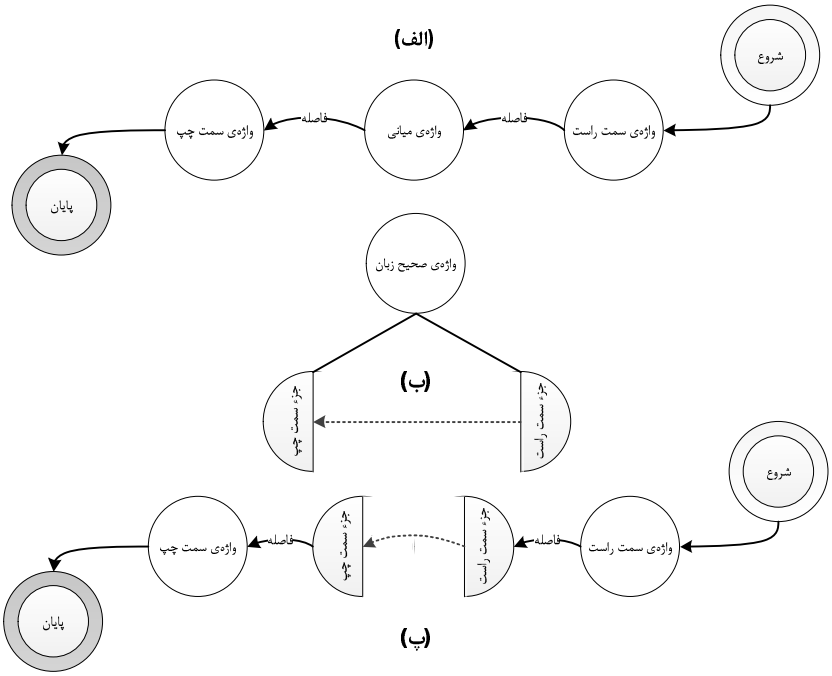
نوشتار صحیح	بسامد	پیوسته‌نویسی	بسامد	جدانویسی	بسامد
می‌شود	۲،۸۰۰،۰۰۰	میشود	۶،۹۰۰،۰۰۰	می شود	۱۰،۳۰۰،۰۰۰
شرکت‌ها	۱،۲۷۰،۰۰۰	شرکتها	۱،۱۹۰،۰۰۰	شرکت ها	۱،۰۸۰،۰۰۰
آب‌سردکن	۵،۶۰۰	آبسردکن	۲۰،۴۰۰	آبسرد کن، آب سرد کن	۱،۰۷۰،۰۰۰
کم‌تر	۳۵۱،۰۰۰	کمتر	۴۵،۴۰۰،۰۰۰	کم تر	۱۲۶،۰۰۰
به عنوان	۷،۷۳۰،۰۰۰	بعنوان	۶۴،۴۰۰،۰۰۰		
آن‌ها	۵۱،۲۰۰،۰۰۰	آنها	۱۳۰،۰۰۰،۰۰۰	آن ها	۱،۵۵۰،۰۰۰
واژک‌شناسی	۱	واژکشناسی	۲	واژک شناسی	۲۶

در نوشتار زبان فارسی از هم‌آیی حروف جهت ایجاد واژه، فاصله برای تفکیک واژه‌ها، و علائم نشانه‌گذاری پایان دهنده برای اتمام جمله‌ها استفاده می‌شود. در این میان معمولاً هم‌آیی اشتباهِ حروف، موجب ایجاد خطاهای املایی در واژه‌ها می‌شود و واژه‌ای خارج از واژه‌های زبان را ایجاد می‌کنند. اما علت دیگر خطاهای املایی، اشتباه در هم‌آیی فاصله و حروف است، به گونه‌ای که این هم‌آیی اشتباه، واژه صحیحی را دچار اشکال نماید. روش ساخت واژه‌ها، جمله‌ها و متن‌های زبان فارسی در شکل (۳-۱) نشان داده شده است.



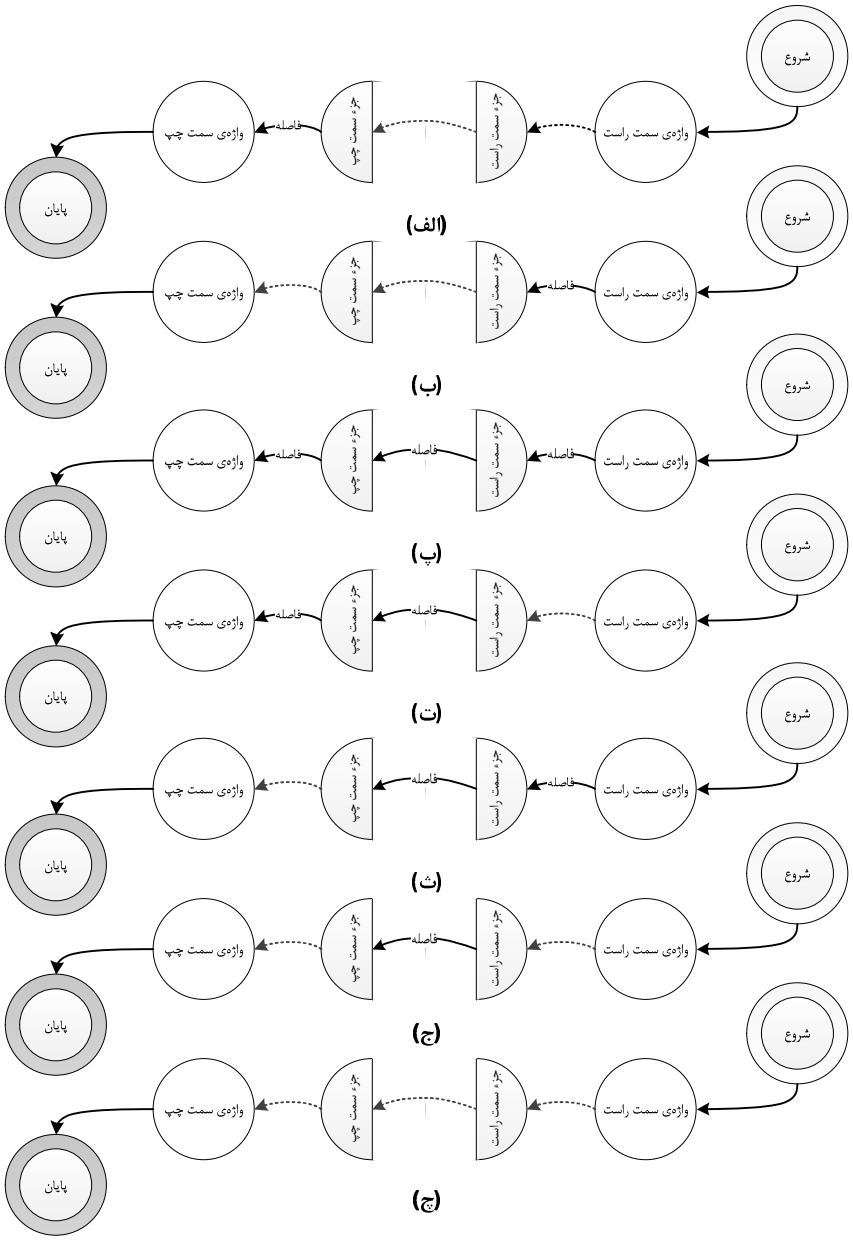
شکل (۳-۱) نمونه ساخت واژه‌ها و جمله‌ها در زبان فارسی

می‌توان فرض کرد که یک واژه از دو بخشِ ***جزء سمت چپ*** و ***جزء سمت راست*** همان گونه که در شکل (۳-۲) مشخص گردیده، تشکیل شده باشد. شکل (۳-۲) بخش الف، یک هم‌آیی صحیح از سه واژه را نشان می‌دهد که نماینده‌ی مدل نحوه‌ی شکل‌گیری متون زبان فارسی بر اساس مدل ارائه شده در شکل (۳-۱) است. در این حالت سه واژه‌ی ***واژه‌ی سمت راست***، ***واژه‌ی میانی***، و ***واژه‌ی سمت چپ*** یک قطعه‌ی متنی را تشکیل می‌دهند که می‌توان کلیه‌ی بخش‌های متن را مانند مدل هم‌آیی این سه واژه شبیه‌سازی نمود. همچنین می‌توان فرض کرد که یک واژه از دو بخشِ ***جزء سمت چپ*** و ***جزء سمت راست*** همان گونه که در شکل (۳-۲) بخش ب، مشخصِ گردیده، تشکیل شده باشد. با جایگزینی ***واژه‌ی میانی*** با ***جزء سمت چپ*** و ***جزء سمت راست***ِ تشکیل دهنده‌ی آن، مدل هم‌آیی سه واژه به صورت شکل (۳-۲) بخش پ، خواهد شد.



شکل (۳-۲) نمونه‌ی هم‌آیی واژه‌ها در زبان فارسی

مدل هم‌آیی سه واژه که در شکل (۳-۲) بخش پ نشان داده شده است، تنها حالت صحیح از هم‌آیی این سه واژه است. شکل (۳-۳) هفت حالت نادرست از هم‌آیی سه واژه بر اساس اشتباه در فاصله‌گذاری را نشان می‌دهد. این هفت حالت در اغلب موارد منجر به ایجاد خطای املایی در متن می‌شوند. اشکالات فاصله‌گذاریِ اشتباه و ترکیب آن‌ها با چالش‌هایی که فاصله‌ی درون واژه‌ای ایجاد می‌کند، عمده‌ی خطاهای املایی زبان فارسی را شامل می‌شود (حدود ۸۰ درصد) [6] که در روش‌های معمول خطایابی قابل تصحیح و بعضاً قابل تشخیص نیز نیستند. بنابراین، در نظر گرفتن چالشِ فاصله‌گذاری میان و درون واژه‌ها در زبان فارسی از نکات بسیار مهم در طراحی و ایجاد خطایاب‌های املایی صَرفی و حتی نحوی خواهد بود.



شکل (۳-۳) هفت گونه از خطاهای املاییِ ایجاد شده بر اثر فاصله‌گذاری نادرست

۳-۳ حروف هم‌آوا

هم‌آواها، واژه‌هایی هستند که مشابه یکدیگر تلفظ می‌شوند. این واژه‌ها می‌توانند یکسان نیز نوشته شوند، مانند «شیر» به مصداق حیوان و «شیر» به مصداق مایعی خوراکی، یا نوشتار یکسان نداشته باشند، مانند «قالب» و «غالب». هم‌آواها می‌توانند واژه‌های صحیحی از زبان یا واژه‌هایی دارای خطای املایی باشند که مورد اخیر، خصوصاً در زبان فارسی از چالش‌های خطایابی املایی است. خطاهای املاییِ رخ داده بر اساس هم‌آوایی واژه‌ها، معمولاً توسط افرادی که به زبانی غیر از زبان مادریِ خود نگارش می‌کنند یا تسلط کافی به زبان مورد نظر ندارند روی می‌دهد. اما در زبان فارسی به علت تعداد بسیار زیاد حروف هم‌آوا (نزدیک ۶۵ ٪ از حروف با در نظر گرفتن همزه)، خطاهای املاییِ رخ داده بر اساس هم‌آوایی واژه‌ها حتی میان نگارندگان با تحصیلات بالا نیز روی می‌دهد. به عنوان نمونه، واژه‌ی «دغدغه» /dæɣdæɣe/ می‌تواند به صورت‌های «دقدغه»، «دغدقه» و «دقدقه» نیز نوشته شود یا واژه‌ی «استثنایی» /estesnɒʔi/ می‌تواند به ۴۳۲ گونه‌یِ هم‌آوا نوشته شود. حروف هم‌آوای زبان فارسی در جدول (۳-۴۷) دسته‌بندی شده‌اند.

جدول (۳-۴۷) حروف هم‌آوای فارسی

ردیف	دسته	اعضاء	تلفظ رایج	تلفظ‌های دیگر
۱	الف	{«آ» ,«ا»}	/ɒ/	/æ/, /e/, /o/, /ʔ/
۲	همزه	{«ی» ,«ئ» ,«و» ,«ؤ» ,«ا» ,«إ» ,«أ» ,«ء»}	/ʔ/	/æ/, /e/, /o/, /ɒ/, /j/, /v/, /i/, /u/
۳	ت	{«ط» ,«ت»}	/t/	
۴	سین	{«ث» ,«ص» ,«س»}	/s/	
۵	ه	{«ح» ,«ه»}	/h/	/e/, /æ/
۶	ز	{«ذ» ,«ظ» ,«ض» ,«ز»}	/z/	
۷	قاف	{«ق» ,«غ»}	/ɣ/, /ɢ/	

۳-۴ حروف هم‌شکل

بسیاری از حروف زبان فارسی بسته به مکانِ قرارگیری در واژه، می‌توانند چهار حالتِ نوشتاری مختلف داشته باشند. این حالت‌ها عبارتند از حالت مجرد مانند «گ»، حالت آغازین مانند «گـ»، حالت میانی مانند «ـگـ»، و حالت پایانی مانند «ـگ». برخی از حروف زبان فارسی ظاهر مشابه با یکدیگر دارند. این حروف هم‌شکل، خصوصاً هنگامی که از قلم‌های نامناسب و در اندازه‌های کوچک استفاده می‌شود، به سختی قابل تمایز از یکدیگر هستند. از طرفی چون بسیاری از حروف هم‌شکل، در چیدمان[1] استاندارد صفحه‌کلید فارسی مجاور یکدیگر نیز هستند، احتمال این که اشتباهاً به جای یکدیگر درج شوند و منجر به پدید آمدن خطاهای املایی شوند، بالا است. خطاهای املایی ناشی از حروف هم‌شکل به طور خاص، در نویسه‌خوانی نوری متون فارسی اهمیت ویژه‌ای دارند و توجه خاصی را هنگام خطایابی می‌طلبند. حروف هم‌شکل زبان فارسی در جدول (۳-۴۸) دسته‌بندی شده‌اند.

جدول (۳-۴۸) حروف هم‌شکل فارسی

ردیف	اعضاء	حالت‌های هم‌شکل
۱	{«ا», «إ», «أ»}	همه‌ی حالت‌ها
۲	{«پ», «ب»}	همه‌ی حالت‌ها
۳	{«ث», «ت»}	همه‌ی حالت‌ها
۴	{«چ», «ج», «ح»}	همه‌ی حالت‌ها
۵	{«خ», «ح»}	همه‌ی حالت‌ها
۶	{«ذ», «د»}	همه‌ی حالت‌ها
۷	{«ژ», «ز», «ر»}	همه‌ی حالت‌ها
۸	{«ض», «ص»}	همه‌ی حالت‌ها
۹	{«ظ», «ط»}	همه‌ی حالت‌ها
۱۰	{«غ», «ع»}	همه‌ی حالت‌ها
۱۱	{«گ», «ک»}	همه‌ی حالت‌ها
۱۲	{«ﻗ», «ﻓ», «ﻘ», «ﻔ»}	حالت‌های آغازین و میانی

۱ معادل فارسی واژه‌ی انگلیسی Layout

فصل چهارم

خطایابی املایی خودکار در زبان فارسی

۴-۱ مقدمه

با آغاز استفاده‌ی عمومی از رایانه، چالش‌های بسیاری پیرامون پردازش زبان‌های طبیعی به وجود آمد. خطا در کار با زبان‌های طبیعی امری است ناگزیر که متخصصان را بر آن داشت تا برای رفع اشکالات موجود در این زمینه تلاش کنند. از جمله‌ی این مسائل خطاهای املایی و نگارشی کاربران رایانه‌ای است که می‌تواند دلایل مختلفی داشته باشد مانند اشتباه در تحریر و یا کم‌اطلاعی کاربران از زبان مورد استفاده. بسیاری از خطاها از دید کاربران پنهان می‌ماند و بسیاری دیگر خطاهایی هستند که کاربران به اشتباه گمان بر درست بودن آن‌ها دارند. انسان‌ها معمولا در هنگام گفتگو و یا نوشتن دچار اشتباهاتی در چهار سطحِ لغوی[۱]، نحوی[۲]، معنایی[۳]، و مبتنی بر بافت متن[۴] می‌شوند [7, 8]. خطاهای املایی در زمینه‌های مختلف و به دلایل مختلفی روی می‌دهند، به عنوان نمونه خطاهای املایی در زمینه‌هایی مانند خطاهای حروف‌چینی در زمینه نمایه‌سازی و سیستم‌های بازیابی اطلاعات [9]، تشخیص اشتباه متون نوشته شده [10, 11]، خطاهای املایی در متون علمی و دانشگاهی [12]، و خطاهای املایی ناخودآگاه در نوشتار کودکان [13, 14] مورد بررسی قرار گرفته‌اند.

خطایاب‌های املایی معمولا به تصحیح خطاهای حروف‌چینی[۵]، نگارشی[۶] و خطاهای ناشی از بازشناسی نوری نویسه‌ها[۷] می‌پردازند [15]. خطاهای حروف‌چینی معمولا از

---

۱ معادل فارسی واژه انگلیسی Lexical
۲ معادل فارسی واژه انگلیسی Syntactic
۳ معادل فارسی واژه انگلیسی Semantic
۴ معادل فارسی واژه انگلیسی Contextual
۵ معادل فارسی واژه انگلیسی Typographical
۶ معادل فارسی واژه انگلیسی Orthographical
۷ معادل فارسی عبارت انگلیسی Optical Character Recognition (OCR)

اشتباهات معمول حروف‌چینی ناشی می‌شوند. برای مثال، خطای املایی حروف‌چینی ممکن است به علت درج اشتباه حرف کناریِ نویسه‌ی مورد نظر در صفحه کلید روی داده باشد، مانند درج اشتباه «رایاته» به جای «رایانه» [9, 12, 15]. خطاهای نگارشی به علت ناآگاهی نگارنده از قواعد و واژگان زبان مانند حدس زدن املای واژه، نگارش واژه از روی تلفظ آن، و یا انتخاب واژه‌ی اشتباه، مانند استفاده از «همچنین» به جای «همچنان»، به وجود می‌آیند [15].

پژوهش‌های صورت گرفته بر روی پیکره‌های بسیار بزرگ [9, 16] نشان داد که ۸۰٪ تا ۹۰٪ از خطاهای املایی به دلیل چهار نوع خطای عمده رخ می‌دهند. این چهار نوع خطا عبارتند از (۱) حذف یک حرف، (۲) درج یک حرف، (۳) جایگزینی یک حرف با حرفی دیگر، و (۴) جابجایی دو حرف کنار هم از واژه‌ی صحیح. این نوع از خطاهای املایی به خطاهای تکی[۱] موسومند [9, 17]، در حالی که خطاهای املایی‌ای که بیش از یک خطای تکی داشته باشند، به خطاهای چندگانه[۲] موسومند [18]. مثالی از این چهار نوع خطای املایی برای واژه‌ی «صلح» در جدول (۴-۱) نشان داده شده‌اند. پژوهش‌گران همچنین دریافتند که حرف اول کلمات معمولا صحیح هستند و خطای املایی در حروف اول کلمات رخ نمی‌دهد [12, 14, 19].

جدول (۴-۱) نمونه‌ای از انواع خطاهای املایی

نوع خطا	انواع خطاهای املایی برای واژه‌ی «صلح»
جایگزینی	صح
حذف	صاح
درج	صلخح
جابجایی	صحل

فرایند خطایابی معمولا شامل سه مرحله‌ی (۱) تشخیص واژه‌ی غلط، (۲) پیدا کردن پیشنهاد یا پیشنهادات صحیح جهت جایگزینی، و (۳) رتبه‌بندی پیشنهادات هنگامی که بیش از یک پیشنهاد برای جایگزینی با واژه‌ی غلط وجود دارد؛ این رتبه‌بندی بر اساس میزان مناسب بودن هر پیشنهاد جهت جایگزینی با واژه‌ی غلط صورت می‌گیرد.

۱ معادل فارسی عبارت انگلیسی Single-error

۲ معادل فارسی عبارت انگلیسی Multi-error

روش سنتی خطایابی و اصلاح خطاهای املایی به این صورت است که واژه‌ها در یک واژه‌نامه از واژه‌های صحیح زبان جستجو می‌شوند، در صورت عدم یافتن یک واژه در واژه‌نامه، آن واژه به عنوان یک غلط املایی تشخیص داده می‌شود. در مرحله‌ی بعد با اعمال تغییرات (حذف و درج حروف، جایگزینی حروف با حروف دیگر و جابجایی حروف کنار هم) در واژه‌ی غلط، لیستی از واژه‌های محتمل زبان تولید می‌شود؛ با جستجوی این واژه‌های محتمل در واژه‌نامه، واژه‌های صحیح زبان به عنوان پیشنهادات جهت جایگزینی با واژه‌ی غلط استخراج می‌گردند [20] و در نهایت این پیشنهادات بر حسب میزان مناسب بودن جهت جایگزینی با واژه‌ی دارای خطای املایی، رتبه‌بندی می‌شوند [21, 22].

یک خطایاب باید پیشنهادهای مطلوب‌تر را از میان شاید صدها پیشنهاد گزینش کرده و فهرستی کوچک‌تر (معمولاً ۷ تایی) از پیشنهادها را ارائه کند. از طرفی، دقت و قدرت یک خطایاب به ارائه‌ی پیشنهاد مطلوب سر فهرست پیشنهادها یا هر چه نزدیک‌تر به ابتدای فهرست است [20, 23]. روش‌ها و سیاست‌های مختلفی جهت رتبه‌بندی پیشنهادات وجود دارد. از آن جمله می‌توان به (۱) روش‌های فاصله ویرایشی[۱] و شباهت رشته‌ای[۲] [9, 24, 25]، (۲) روش‌های آماری[۳] و احتمالی[۴] [10, 16, 26]، (۳) کلیدهای مشابهت[۵] [27-30]، و (۴) روش‌های مبتنی بر قوانین [19, 31]. پژوهش‌گران همچنین نشان داده‌اند که روش‌های رتبه‌بندی بر اساس فاصله‌ی ویرایشی و شباهت رشته‌ای، و روش‌های آماری و احتمالی مانند استفاده از بسامد کلمات[۶] بهترین نتایج را به دست می‌دهند [22, 32]. البته استفاده از

۱ معادل فارسی عبارت انگلیسی Edit Distance

۲ معادل فراسی عبارت انگلیسی String Distance

۳ معادل فارسی واژه‌ی انگلیسی Statistical

۴ معادل فارسی واژه‌ی انگلیسی Probabilistic

۵ معادل فارسی عبارت انگلیسی Similarity Key

۶ در این روش با استفاده از بسامد تکرار واژه‌ها در نمونه زبانی، پیشنهادهای جایگزینی‌ای که بسامد بیشتری دارند انتخاب می‌شوند. مشکل این روش محاسبه صحیح و معتبر بسامد واژه‌ها است که این امر بسیار وابسته به پیکره‌های متنی مورد استفاده است. از طرفی برخی واژه مانند حروف اضافه و برخی فعل‌ها، بسامد بسیار بالایی پیدا می‌کنند در حالی که احتمال این که اشتباه نوشته شوند پایین است و رتبه‌بندی پیشنهادات جایگزینی را مختل می‌سازد.

روش‌های ترکیبی[۱] و تطبیقی[۲] با استفاده از اطلاعات آماری و دانش زبان‌شناختی از زبان مقصد می‌تواند نتایج بهتری را ارائه دهد اما این اطلاعات و دانش برای کلیه‌ی زبان‌ها در دسترس نیست و یا تولید آن مستلزم صرف وقت و هزینه‌ی بسیار است [6].

۲-۴ پژوهش‌ها و کارهای انجام گرفته پیرامون خطایابی املایی

پژوهش‌های ارزشمند زیادی پیرامون تشخیص و تصحیح خطاهای املایی صورت گرفته است. این مطالعات شامل پژوهش‌های پیشگام از اوایل دهه‌ی ۶۰ میلادی [9, 12, 19] و پژوهش‌های جدیدتر مانند خطایابی با استفاده از روش‌های یادگیری ماشین[۳] [33-36]، خطایابی با استفاده از چند-وزنی‌ها[۴] [37-39]، خطایابی با استفاده از کلیدهای مشابهت و کلیدهای آوایی[۵] [27-30]، و روش‌های خطایابی بدون واژه‌نامه [40, 41] است.

نسیم و همکارانش [42]، یک روش رتبه‌بندی پیشنهاداتِ جایگزینیِ کارا با استفاده از کلیدها آوایی و شکلیِ[۶] ثابت و ایستا برای زبان اُردو ارائه کرده‌اند. پژوهش‌گران در این کار با ترکیب اطلاعات آماری از بسامد واژه‌ها، کلیدهای آوایی، کلیدهای شکلی، و فاصله‌ی رشته‌ای ساده یک روش ترکیبی رتبه‌بندی پیشنهاد داده‌اند که با ادعایشان مبنی بر «مناسب نبودن روش‌های آماری برای زبان‌هایی که از لحاظ آماری دچار نقصان اطلاعات هستند، مانند زبان اُردو» در تضاد است. آن‌ها همچنین تحلیلی مناسب از الگوهای خطاهای املایی در زبان اُردو ارائه کرده‌اند اما این الگوها را با الگوهای خطا در دیگر زبان‌ها مقایسه نکرده‌اند. همچنین استفاده از کلیدهای ثابت و ایستا، روش ارائه شده را وابسته به زبان می‌سازد و استفاده از این روش در زبان‌های دیگر، مستلزم انجام فرایند زمان‌بر، پرهزینه و غیرقطعیِ استخراج کلیدهای آوایی و شکلی زبان مورد نظر است.

شالان و همکارانش [43] خطاهای املایی سنتی، همچنین خطاهای املایی ناشی از

---

۱ معادل فارسی واژه‌ی انگلیسی Hybrid

۲ معادل فارسی واژه‌ی انگلیسی Adaptive

۳ معادل فارسی عبارت انگلیسی Machine Learning

۴ معادل فارسی عبارت انگلیسی N-Gram

۵ معادل فارسی عبارت انگلیسی Phonetic Key

۶ معادل فارسی واژه‌ی انگلیسی Shapex

هم‌آواها و هم‌شکل‌ها را در زبان عربی مطالعه نموده و یک خطایاب املایی ارائه کرده‌اند. خطایاب ارائه شده به تشخیص خطاهای املایی و ارائه پیشنهادات جهت جایگزینی می‌پردازد؛ اما بحثی راجع به روش رتبه‌بندی احتمالیِ مورد استفاده، همچنین ارزیابی خطایاب ارائه شده مطرح نگردیده است.

بَراری و همکارانش [44] یک خطایاب تطبیقی با استفاده از درخت سه‌برگچه‌ای[1] برای زبان فارسی ارائه کرده‌اند. آن‌ها به هر گره درخت یک هزینه‌ی گذر اختصاص داده‌اند و سعی در پیمایش درخت با کمترین هزینه‌ی گذر برای واژه‌های پیشنهادی دارند. آن‌ها همچنین از روش‌های یادگیری جهت به‌روزرسانی هزینه‌های گذر استفاده کرده‌اند. محققان در این روش تأثیرات چیدمان صفحه کلید، هم‌آوایی و هم‌شکلی حروف را در پیدایش خطاهای املایی در نظر نگرفته‌اند. در ادامه قاسمی‌زاده و همکارانش [45] در تکمیل روش فوق، یک روشی یادگیرنده با استفاده از ساختار درخت سه‌برگچه‌ای ارائه کرده‌اند. آن‌ها روش ارائه شده را با استفاده از دو مجموعه‌ی دادگان مَحَک از زبان فارسی و انگلیسی ارزیابی کرده و به نتایج نسبتا مناسبی نیز دست یافته‌اند اما باز هم به تأثیرات چیدمان صفحه کلید، هم‌آوایی و هم‌شکلی حروف را در پیدایش خطاهای املایی نپرداخته‌اند.

فَیلی و همکارانش [46, 47] یک خطایابِ زبان فارسی ارائه کرده‌اند که توانایی تشخیص و تصحیح خطاهای املایی، معنایی و نَحوی را داراست. خطایاب املایی ارائه شده از یک روش مرتب‌سازی ترکیبی با استفاده از شباهت رشته‌ای و بسامد واژه‌ها استفاده می‌کند. در این روش جدولی شامل شباهت میان حروف مختلف بر اساس چندین تابع ابتکاری تولید شده است که با استفاده از این جدول، شباهت میان رشته‌ها (واژه‌ی غلط و پیشنهادات جایگزینی) محاسبه می‌شود. توابع ابتکاری به کار رفته از اطلاعات مختلفی مانند بررسی‌های آماریِ خطاها و چیدمان صفحه کلید استفاده می‌کنند.

موسسه دانش‌گستران سپنتا نیز خطایابی تحت عنوان «ویرا» [48] ارائه نموده که به ادعای پدیدآورندگان آن از پایگاه واژه‌های ریشه‌یابی‌شده با بیش از صد هزار مدخل صحیح فارسی و بیش از ششصد بن فعل فارسی استفاده می‌کند، و توانایی تشخیص بیش از یک میلیون واژه‌ی *انشقاقی* با استفاده از موتور ریشه‌یاب را داراست. همچنین با افزودن

---

۱ معادل فارسی واژه‌ی انگلیسی Ternary

یک رابط خدمات تحت وب[۱]، این محصول امکان ارائه‌ی خدمات خطایابی و تصحیح خطاهای املایی را تحت وب داراست. این خطایاب از سرعت مناسبی در تشخیص و تصحیح خطا برخوردار است اما پشتیبانی مناسبی از واژک‌شناسی زبان فارسی ارائه نکرده، تنها فاصله‌ی ویرایشی یک را مورد پوشش قرار می‌دهد، و نتایج مثبتِ نادرستِ[۲] زیادی در واژه‌های ترکیبی دارد. به عنوان مثال واژه‌های «قاشق‌سواری»، «ریخت‌شنایی» و «اسب‌سوادی» توسط این خطایاب به عنوان واژه‌های صحیح زبان فارسی تشخیص داده می‌شوند.

همان طور که اشاره شد، روش‌های رتبه‌بندی بر اساس فاصله‌ی ویرایشی و شباهت رشته‌ای، و روش‌های آماری و احتمالی مانند استفاده از بسامد کلمات بهترین نتایج را به دست می‌دهند. از طرفی، اطلاعات آماری و زبان‌شناختی کافی برای زبان‌هایی مانند زبان فارسی، در دسترس نیست؛ از این رو، روش‌های رتبه‌بندی بر اساس فاصله‌ی ویرایشی و شباهت رشته‌ای می‌توانند نتایج اتکاپذیرتری را ارائه نمایید. روش‌های زیادی از رتبه‌بندی بر اساس فاصله‌ی ویرایشی و شباهت رشته‌ای وجود دارد که برخی از آن‌ها مانند هَمینگ[۳] [49] و لِوِنشتاین[۴] [25] بر روی دقت و مؤثر بودن روش تمرکز کرده‌اند و برخی دیگر مانند هیرشبِرگ[۵] [50]، اُکُنِن[۶] [51] و مَسِک[۷] [52] بر روی کاهش پیچیدگی محاسباتی، حافظه‌ی مورد نیاز و زمان اجرا متمرکز شده‌اند. در ادامه روش‌های رتبه‌بندی مطرح بر اساس شباهت رشته‌ای و فاصله‌ی ویرایشی را مورد بررسی قرار خواهیم داد.

۴-۲-۱ روش فاصله‌ی حروف

در روش فاصله‌ی حروف[۸] ابتدا حروف دو واژه به صورت متناظر بررسی می‌شود و در صورت تطابق نداشتن، یک امتیاز منفی محاسبه می‌شود؛ سپس حروف دو واژه دو به دو،

---

۱ معادل فارسی عبارت انگلیسی Web Service
۲ معادل فارسی عبارت انگلیسی False Positive
۳ نوشتار فارسی اسم انگلیسی Hamming
۴ نوشتار فارسی اسم انگلیسی Levneshtein
۵ نوشتار فارسی اسم انگلیسی Hirschberg
۶ نوشتار فارسی اسم انگلیسی Ukkonen
۷ نوشتار فارسی اسم انگلیسی Masek
۸ معادل فارسی عبارت انگلیسی Letter Distance

مقایسه می‌شوند و مانند مورد قبل در صورت عدم تطابق یک امتیاز منفی دیگر نیز محاسبه می‌شود. پس از آن، اگر حروف اول دو واژه نیز متفاوت بودند، بار دیگر امتیازی منفی محاسبه می‌گردد. برای مثال، برای دو واژه‌ی «تقلب» و «تفلاب» این امتیازها این گونه محاسبه می‌شوند:

- ۳ امتیاز منفی برای عدم هم‌خوانی حروف متقابل
- ۴ امتیاز منفی برای عدم هم‌خوانی دو به دوی حروف متقابل، مثلاً هم‌خوانی نداشتن «تق» با «تف» و «قل» با «فل»
- چون حروف اول یکسان هستند امتیاز منفی دیگری محاسبه نمی‌شود

بنابر این، امتیاز منفی محاسبه شده برای این دو واژه ۷ است. حال اگر همین امتیاز را در ادامه برای دو واژه‌ی «تالاب» و «تفلاب» محاسبه کنیم، خواهیم داشت:

- ۱ امتیاز منفی برای عدم هم‌خوانی حروف متقابل
- ۲ امتیاز منفی برای عدم هم‌خوانی دو به دوی حروف متقابل

در نهایت امتیاز منفی در این حالت معادل ۳ می‌شود که با مقایسه با حالت قبل می‌توان نتیجه گرفت «تالاب» به نسبتِ «تقلب»، پیشنهاد بهتری برای جایگزینی با «تفلاب» است.

۴-۲-۲ فاصله‌ی هَمینگ

فاصله هَمینگِ دو واژه با طول مشابه، شامل تعداد حروف متناظرِ نامشابه است [49]. برای مثال، فاصله هَمینگ دو واژه‌ی «امید» و «نماد»، به علت عدم مطابقت «ا» با «ن» (حروف اول از دو واژه) و «ی» با «ا» (حروف سوم از دو واژه) معادل ۲ است. برای تبدیل این فاصله به حالت نرمال شده در بازه‌ی ۰ تا ۱، این فاصله به طول واژه تقسیم می‌گردد. بنابراین فاصله‌ی دو واژه‌ی «امید» و «نماد» معادل ۰/۵ است. الگوریتم کلی فاصله‌ی هَمینگ در شکل (۴-۱) نشان داده شده است.

برای واژه‌ها یا رشته‌هایی که طول یکسان ندارند، نسخه‌ای از فاصله‌ی هَمینگ مورد استفاده قرار می‌گیرد که به ازای هر حرف اضافه، یک فاصله‌ی اضافه منظور می‌کند. به عنوان نمونه فاصله ی هَمینگِ میان دو واژه‌ی «امید» و «امنیت» به دلیل عدم مطابقت «ی» با «ن» (حروف سوم از دو واژه) و «د» با «ی» (حروف چهارم از دو واژه) و حرف «ت» از واژه‌ی «امنیت» که موجب شده طول آن بیشتر از واژه‌ی «امید» شود، معادل ۳ خواهد بود.

```
1: Hamming(q, l)
2: {
3:     if (q_i == l_i)
4:         f_h(i) = 0
5:     else
6:         f_h(i) = 1
7:     return AVG(f_h)
8: }
```

شکل (۴-۱) الگوریتمِ کلیِ فاصله‌ی هَمینگ

۴-۲-۳ فاصله‌ی لِوِنشتاین

فاصله‌ی لِوِنشتاین میان دو رشته از کمینه‌ی تغییرات لازم برای تبدیل یک رشته به دیگری محاسبه می‌شود. این تغییرات شامل، درج یک حرف اضافه، حذف یک حرف و یا جایگزینی دو حرف است [25]. برخلاف روش هَمینگ، این روش می‌تواند بر واژه‌های با طول متفاوت نیز اعمال شود که این تفاوت در طول می‌تواند توسط حذف و یا درج حروف ایجاد شده باشد. برای مثال، فاصله‌ی لِوِنشتاین میان دو واژه‌ی «کامپیوتر» و «کامپچیوت» معادل ۲ است (حذف «چ» و درج «ر»).

همان گونه که در شکل (۴-۲) نشان داده شده است، برای همه‌ی واژه‌های یک تابع $f_l(0,0)$ محاسبه و معادل صفر می‌گردد. سپس یک تابع $f_l(i,j)$ برای تمامی حروف به صورت مکرر محاسبه می‌شود. هر درج، حذف و یا جایگزینی یک امتیاز به تابع می‌افزاید.

```
1: Levenshtein (q , l)
2: {
3:     f_l(0 , 0) = 0
4:     if (q_i == l_j)
5:         d(q_i , l_j) = 0
6:     else
7:         d(q_i , l_j) = 1
8:     f_l(i , j) = min (( f_l(i – 1 , j) + 1), ( f_l(i , j - 1) + 1), ( f_l(i – 1 , j - 1) + d(q_i , l_j )))
9:     return f_l( |q| , |i| )
10: }
```

شکل (۴-۲) الگوریتم کلی فاصله‌ی لِوِنشتاین

۴-۲-۴ فاصله‌ی دَمِرا-لِوِنشتاین

دَمِرا[1] با مطالعه و بررسی پیکره‌های متنی بسیار، گونه‌ای دیگر از خطاهای املایی را برای جابه‌جایی دو حرف مجاور در یک واژه مشاهده نمود [9]. بنابراین، با افزودن این نوع خطا به روش لِوِنشتاین، روش دَمِرا-لِوِنشتاین از چهار نوع خطای درج، حذف، جایگزینی و جابجایی پشتیبانی می‌کند. الگوریتم کلی فاصله‌ی دَمِرا-لِوِنشتاین در شکل (۴-۳) آمده است.

```
1:  DamerauLevenshtein (q , l)
2:  {
3:      f_dl(0 , 0) = 0
4:      if (q_i == l_j)
5:          d(q_i , l_j) = 0
6:      else
7:          d(q_i , l_j) = 1
8:      if ((q_i == l_j-1) and (q_i-1 == l_j))
9:          t(q_i , l_j) = 0
10:     else
11:         t(q_i , l_j) = 2
12:     f_dl(i , j) = min ((f_dl(i – 1 , j) + 1), (f_dl(i , j - 1) + 1), (f_dl(i – 1, j - 1) + d(q_i , l_j)), (f_dl(i – 2 , j - 2) + t(q_i , l_j)))
13:     return f_dl(|q| , |i|)
14: }
```

شکل (۴-۳) الگوریتم کلی فاصله‌ی دَمِرا-لِوِنشتاین

۴-۲-۵ فاصله‌ی وَگنِر-فیشِر

روش وَگنِر-فیشِر، روش تغییر یافته‌ای از فاصله‌ی لِوِنشتاین است که در آن، بر خلاف روش لِوِنشتاین که هزینه‌ی مشابهی برای حذف، درج و جایگزینی و معادل یک در نظر گرفته شده است، هزینه‌های متفاوتی برای گونه‌های مختلف از خطاهای املایی در نظر گرفته شده است [24].

---

۱ نوشتار فارسی اسم انگلیسی Damerau

این روش، همچنین روش لِونِشتاین و دَمِرا-لِونِشتاین، پیچیدگی زمانی[۱] و فضایی‌ای[۲] معادل $O(mn)$ دارند که m طول واژه‌ی اول و n طول واژه‌ی دوم است. الگوریتم کلی فاصله‌ی وَگنِر-فیشِر در شکل (۴-۴) امده است که مشابه الگوریتم فاصله‌ی لِونِشتاین است با این تفاوت که هزینه‌ی هر خطا می‌تواند متفاوت باشد. در شکل (۴-۴) $d(q_i, \varepsilon)$ هزینه‌ی حذف، $d(\varepsilon, l_j)$هزینه‌ی درج و $d(q_i, \varepsilon)$ هزینه‌ی جایگزینی دو حرف است.

```
1: WagnerFischer (q , l)
2: {
3:     fwf(0 , 0) = 0
4:     d(qi , ε) = cost of deletion
5:     d(ε , lj) = cost of insertion
6:     d(qi , lj) = cost of substitution
7:     fwf(i , j) = min (( fwf(i – 1 , j) + d(qi , ε)), ( fwf(i , j - 1) + d(ε , lj)), ( fwf(i – 1 , j - 1) + d(qi , lj)))
8:     return fwf(|q| , |i|)
9: }
```

شکل (۴-۴) الگوریتم کلی فاصله‌ی وَگنِر-فیشِر

۴-۲-۶ فاصله‌ی جَرو-وینکلِر

روش جَرو-وینکلِر [53] گونه‌ای از روش فاصله‌ی جَرو است که پیش‌تر برای محاسبه میزان اتصال رکوردها[۳] استفاده می‌شد. الگوریتم محاسبه‌ی فاصله‌ی جرو بین دو رشته‌ی l و q در شکل (۴-۵) نشان داده شده است. معیار فاصله‌ی جَرو عددی نرمال ارائه می‌دهد که ۰ بیان‌گر عدم شباهت و ۱ بیان‌گر شباهت و تطابق کامل است. این روش توجهی خاص به جابجایی حروف مجاور دارد.

۱ معادل فارسی عبارت انگلیسی Time Complexity
۲ معادل فارسی عبارت انگلیسی Spatial Complexity
۳ معادل فارسی عبارت انگلیسی Record Linkage

```
1: Jaro(q , l)
2: {
3:     m = number of matching letters
4:     t = number of transpositions
5:     f_j = 1/3 ( m/|q| + m/|l| + (m−t)/m )
6:     return f_j
7: }
```

شکل (۴-۵) الگوریتم کلی فاصله‌ی جَرو

فاصله‌ی جَرو-وینکلِر [54, 55] با تغییر روش جَرو و افزودن معیار پیشوند مشترکِ آغازین روش محاسبه‌ی مشابهت رشته‌ای دیگری ارائه کرده که الگوریتم کلی این روش در شکل (۴-۶) نشان داده شده است. در این الگوریتم، l_c طول پیشوند مشترکِ آغازین در ابتدای دو واژه و حداکثر ۴ است و ρ ضریب میزان اهمیتِ پیشوند مشترکِ آغازین است که به طور پیش‌فرض معادل ۰/۱ در نظر گرفته می‌شود.

```
1: JaroWinkler(q , l)
2: {
3:     l_c = max(length of initial common prefix(q , l), 4)
4:     ρ = 0.1  /* scaling factor */
5:     f_jw = Jaro(q , l) + l_c × ρ × (1 – Jaro (q , l))
6:     return f_jw
7: }
```

شکل (۴-۶) الگوریتم کلی فاصله‌ی جَرو-وینکلِر

۴-۳ الگوهای خطاهای املایی

بررسی و تحلیل الگوهای خطاهای املاییِ، احتمال رخداد گونه‌های مختلف خطاهای حروف‌چینی (درج، حذف، جایگزینی و جابجایی) را مشخص می‌نماید. مطالعات زیادی بر روی تحلیل و بررسیِ الگوهای خطاهای املایی و حروف‌چینی در پیکره‌های متنی

بزرگ برای زبان انگلیسی صورت گرفته است [9, 12, 17]. جدول (۴-۲) الگوها و نرخ رخدادِ خطاهای املاییِ گزارش شده برای زبان انگلیسی را توسط پژوهشگران مختلف نشان می‌دهد.

جدول (۴-۲) الگوها و نرخ رخداد خطاهای املایی و حروف‌چینی در زبان انگلیسی

نوع خطا	دامِرا	پولاک	پِترسون	میتون
جایگزینی	٪ ۶۷	٪ ۳۳	٪ ۳۴	٪ ۴۲
حذف	٪ ۱۸	٪ ۳۲	٪ ۳۵	٪ ۳۳
درج	٪ ۱۲	٪ ۳۰	٪ ۲۱	٪ ۱۹
جابجایی	٪ ۳	٪ ۵	٪ ۱۰	٪ ۶

دامِرا، پولاک و پِترسون نتایج تقریباً مشابهی از الگوها و نرخ رخداد خطاهای املایی در زبان انگلیسی ارائه داده‌اند در حالی که نتایج گزارش شده‌ی میتون قدری متفاوت است با این حال، ترتیب رخداد خطاهای تکی در کلیه‌ی نتایج یکسان است. میتون از یک پیکره‌ی متنی که در آن به طور دستی خطاهای املایی ایجاد شده استفاده کرده بود، و پژوهشگران دیگر از پیکره‌هایی با خطاهای واقعی استفاده کرده بودند. الگوها و نرخ رخداد خطاهای املایی ارائه شده برای زبان انگلیسی تحلیل شده‌اند و شاید مشابه الگوهای زبان فارسی نباشند. از این رو، جهت تحلیل الگوها و نرخ رخداد خطاهای املایی در زبان فارسی از سه پیکره شامل پیکره‌ی همشهری [56]، پیکره‌ی مرکز تحقیقات کامپیوتری علوم اسلامی، و پیکره‌ی پایان‌نامه‌های و سمینارهای دانشجویان آموزش عالی دانشکده‌ی مهندسی کامپیوتر دانشگاه علم و صنعت ایران استفاده نمودیم.

پیکره‌ی همشهری از مقاله‌های چاپ شده بین سال‌های ۱۳۷۵ تا ۱۳۸۱در روزنامه‌ی همشهری استخراج گردیده است. این پیکره از ۱۶۰۰۰۰ مقاله در ۸۳ دسته با ۱۵ میلیون واژه و ۴۱۷۰۰۰ واژه‌ی غیر تکراری تشکیل شده است. پیکره‌ی مرکز تحقیقات کامپیوتری علوم اسلامی شامل بیش از ۸۰۰ کتاب در حوزه‌ی علوم انسانی و اسلامی است که از ۵۰ میلیون واژه و ۳۹۸۰۰۰ واژه‌ی غیر تکراری تشکیل شده است. پیکره‌ی علم و صنعت ایران نیز شامل ۲۱۰۰۰۰۰ واژه و ۲۳۱۰۰۰ واژه‌ی غیر تکراری است.

جدول (۴-۳) الگوها و نرخ رخداد خطاهای املایی در زبان فارسی را نشان می‌دهد. خطاهای وابسته به زمینه آن دسته از خطاها هستند که پیشنهاد مناسب جهت جایگزینی، تنها

با در نظر گرفتن واژه‌های مجاور در متن امکان پذیر است. به عنوان نمونه، واژه‌های «خدا» و «جدا» هر دو می‌توانند جایگزین واژه‌ی دارای خطای املاییِ «چدا» شوند و این که کدام پیشنهاد مناسب است تنها از معنای جمله و کلمات مجاور مشخص خواهد شد.

جدول (۴-۳) الگوها و نرخ رخداد خطاهای املایی و حروف‌چینی در زبان فارسی

خطاهای تکی	خطاهای چندگانه	هم‌آواها	هم‌شکل‌ها	خطا در حرف اول	خطاهای وابسته به بافت	خطاهای املایی
جایگزینی ٪ ۵۳	دو ٪ ۸۹	تکی ٪ ۹۳	تکی ٪ ۸۱	٪ ۲۱	٪ ۱۲	٪ ۰/۰۶
حذف ٪ ۲۷						
درج ٪ ۱۳	بیشتر ٪ ۱۱	چندگانه ٪ ۷	چندگانه ٪ ۱۹			
جابجایی ٪ ۷						
مجموع ٪ ۷۸	**٪ ۲۳**	**٪ ۵۸**	**٪ ۵۴**	**٪ ۲۱**	**٪ ۱۲**	**٪ ۰/۰۶**

۴-۴ خطایابی املایی در زبان فارسی

همان‌طور که پیش‌تر گذشت، روش سنتی خطایابی و اصلاح خطاهای املایی در سه مرحله انجام می‌گیرد. مرحله‌ی اول شامل تشخیص خطا است که در این مرحله واژه‌ها در یک واژه‌نامه از واژه‌های صحیح زبان جستجو می‌شوند، در صورت عدم یافتن یک واژه در واژه‌نامه، آن واژه به عنوان یک غلط املایی تشخیص داده می‌شود. مرحله‌ی بعد به تولید پیشنهادات برای واژه‌های دارای خطای املایی اختصاص می‌یابد. در این مرحله با اعمال تغییرات (حذف و درج حروف، جایگزینی حروف با حروف دیگر و جابجایی حروف کنار هم) در واژه‌ی دارای خطای املایی، لیستی از واژه‌های محتمل تولید می‌شود که با جستجوی این واژه‌های محتمل در واژه‌نامه، واژه‌های صحیح زبان به عنوان پیشنهادات جهت جایگزینی با واژه‌ی غلط استخراج می‌گردند. مرحله‌ی سوم مرحله‌ی رتبه‌بندی ترتیب ارائه‌ی پیشنهادات است. در این مرحله پیشنهادات بر حسب میزان مناسب بودن جهت جایگزینی با واژه‌ی دارای خطای املایی، رتبه‌بندی می‌شوند. در ادامه هر یک از این مراحل را در فرایند تشخیص و تصحیح خطاهای املایی برای زبان فارسی مورد بررسی قرار خواهیم داد.

۴-۴-۱ تشخیص خطا

تشخیص خودکار خطاهای املایی موجود در متن شاید مهم‌تر از تصحیح خودکار خطاها باشد. هدف از خطایابی این است که متن مورد نظر در نهایت عاری از خطا باشد. از طرفی تشخیص دستی خطاهای املایی موجود در متن هم دشوار و هم مستلزم صرف هزینه است. از این رو، تشخیص خودکار خطاهای موجود در متن بسیار پر اهمیت است و همین که یک ویراستار از خطاهای موجود در متن آگاهی یابد، شخصاً و به صورت دستی می‌تواند این خطاها را اصلاح نموده و در نهایت متنی بدون خطای املایی را ارائه نماید.

جهت تشخیص خطاهای املایی، به طور معمول، واژه‌نامه‌ای از واژه‌های صحیح زبان تهیه می‌شود و واژه‌های موجود در متن یک به یک در آن جستجو می‌گردند. در صورتی که واژه‌ای از متن در این واژه‌نامه یافت نشود، این واژه به عنوان یک خطای املایی در نظر گرفته می‌شود. نکاتی که در مرحله‌ی تشخیص خطا، خصوصاً در زبان فارسی باید مورد توجه قرار گیرند ویژگی‌های واژه‌نامه و اطلاعات مورد نیاز از هر واژه، ساختار داده و نحوه‌ی نگهداری و ارائه‌ی واژه‌نامه، و تعداد بسیار زیاد واژه‌های تصریفی است که نگهداری آن‌ها در واژه‌نامه چالش برانگیز است. در ادامه مسائل تأثیرگذار در تشخیص خطاهای املایی در زبان فارسی موررد بررسی قرار خواهند گرفت.

۴-۴-۱-۱ *واژه‌نامه*

از آن‌جا که تشخیص خطا با جستجوی واژه‌ها در واژه‌نامه صورت می‌گیرد و واژه‌ای که در واژه‌نامه موجود است به عنوان واژه‌ای صحیح و بدون خطای املایی و واژه‌ای که موجود نیست به عنوان یک خطای املایی در نظر گرفته می‌شود، کیفیت واژه‌های موجود در واژه اهمیت بسیاری پیدا می‌کند. اگر واژه‌نامه دارای واژه‌های کم کاربرد و مهجور زبان باشد، به گونه‌ای که این واژه‌ها در متون معمول و معیار عمدتاً کاربردی نداشته باشند، ممکن است یک واژه‌ی پر کاربرد زبان به گونه‌ای خطا نوشته شود که معادل یک واژه‌ی کم کاربرد شود که در واژه‌نامه موجود است و در این حالت این خطای املایی نادیده گرفته می‌شود. به عنوان مثال با مراجعه به واژه‌نامه‌ی دهخدا، واژه‌ی «کاب» واژه‌ای صحیح و «پیاله‌ای است دراز و هشت پهلو» اما با توجه به کاربرد کم آن در نوشتار معیار و معمول، اگر چنین واژه‌ای در واژه‌نامه موجود باشد آن گاه ممکن است نوشتار اشتباه واژه‌های پرکاربرد «کتاب» یا «کباب» به صورت «کاب»، تشخیص داده نشود. از این رو واژه‌نامه باید

جامع، یعنی دارای واژه‌های پرکاربرد زبان معیار و مانع، به معنای نداشتن واژه‌های غلط، کم کاربرد، غیر معیار و مهجور باشد.

اطلاعات دیگری از یک واژه که نگهداری آن در واژه‌نامه می‌تواند به خطایابی املایی کمک نماید، بسامد کاربرد واژه و ادات سخن آن است. بسامد تکرار و کاربرد یک واژه می‌تواند هنگام پیشنهاد واژه جهت جایگزینی با یک خطای املایی کاربرد داشته باشد و هر چه یک واژه پرکاربردتر باشد، احتمال این که پیشنهادی صحیح‌تر و مناسب‌تر جهت جایگزینی با واژه‌ی دارای خطای املایی باشد بیشتر است. ادات سخن یک واژه نیز می‌تواند هنگام بررسی قواعد تصریف و هم‌آیی واژه با پسوندها و واژه‌های دیگر کاربرد داشته باشد.

نکته‌ی دیگری که باید مورد توجه قرار گیرد این است که خطایاب املایی توانایی تصحیح و ارائه‌ی پیشنهاد در حوزه‌ی واژه‌های موجود در واژه‌نامه را دارد. از این رو، در صورتی که تصحیح واژه‌های تصریفی، به عنوان مثال تصحیح یک فعل یا یک اسم که جمع بسته شده، مد نظر باشد، این واژه‌ها نیز باید به نحوی در واژه‌نامه موجود باشند. یک راه حل افزودن کلیه‌ی تصریف‌های معتبر فعل‌ها و واژه‌های غیر فعلی به واژه‌نامه است که حجم واژه‌نامه بر روی دیسک و حافظه را به طور قابل ملاحضه‌ای افزایش می‌دهد. راه حل دیگر تصریف و تولید ماشینی واژه‌های تصریفی و افزودن آن‌ها هنگام بارگذاری است که به همان میزان حافظه نیاز دارد اما حجم واژه‌نامه بر روی دیسک بسیار کم‌تر خواهد بود.

روشی پیشنهادی در این کتاب، تنها تولید و تصریف فعل‌ها و افزودن آن‌ها هنگام بارگذاری و بررسی واژه‌های تصریفیِ غیر فعلی، که تعداد بسیار زیادی دارند (بیش از ۲۷۰۰ تصریف برای هر واژه‌ی غیر فعلی موجود در واژه‌نامه)، در زمان اجرا[1] و بدون صرف حافظه جهت نگهداری آن‌ها است. در ادامه به چگونگی تولید فعل‌ها و پس از آن چگونگی در نظر گرفتن صَرف واژه‌های غیر فعلی خواهیم پرداخت.

۴-۴-۱-۱-۱ *تصریف فعل‌ها*

جهت ایجاد تمامی تصریف‌های فعلی باید تمام صیغه‌های ممکن صَرفی برای یک بن تولید شوند؛ یعنی فعل باید برای تمام زمان‌ها، شخص‌ها، معلومیت، مثبت و منفی، صَرف شود؛

---

۱ معادل فارسی عبارت انگلیسی Run Time

علاوه بر آن، باید بیان‌های ممکن برای هر صیغه نیز تولید شود. در مرحله بعد باید قطعات هر کدام از صیغه‌ها تولید شده و پس از حذف قطعات تکراری، نتیجه به واژه‌نامه افزوده شود.

p تولید قطعات فعل

در این‌جا منظور از قطعات فعل، بخش‌هایی از فعل است که توسط فاصله یا توسط کلمات دیگر، از هم جدا می‌شوند. مثلاً فعل «خواهم گفت» که صیغه‌ی آینده اول شخص از بن ماضی «گفت» است، دو بخش «خواهم» و «گفت» را شامل می‌شود. با مطالعه‌ی الگوهای صَرف فعل که در بخش‌های گذشته ارایه شد درمی‌یابیم که قطعات فعل در الگوهای مختلف تکرار می‌شوند. می‌توان نشان داد که برای تولید تمام قطعات فعلی برای تمام زمان‌ها کافیست تا برای چند زمان محدود قطعات فعل را تولید کنیم. در واقع باید زیر مجموعه‌ای پوشا از زمان‌ها را یافت که تمام زمان‌ها را پوشش دهد. البته ممکن است تعداد زیادی از این زمان‌ها وجود داشته باشد. یک نمونه از مجموعه زمان‌های پوشا می‌تواند مجموعه‌ی {*ماضی ساده، ماضی استمراری، ماضی ساده نقلی، ماضی استمراری نقلی، مضارع اخباری، مضارع التزامی، امر*} باشد. به عنوان مثال تعداد قطعات مختلفی که از مصدر «گفتن» بدست می‌آید، شامل ۱۱۲ قطعه مختلف است که در شکل (۴-۷) بخش الف و قطعات فعلی از مصدر «آراستن» در شکل (۴-۷) بخش ب آمده‌اند.

دیگر قطعات فعلی در صَرف یک فعل مانند قطعه‌ی «خواهم» در «خواهم گفت»، هنگام تولید قطعات فعلی از مصدر «خواستن» تولید شده و به واژه‌نامه افزوده می‌شود و از این رو نیاز به تولید این گونه قطعات برای مصدرهای مختلف نیست. خطایابی املایی فعل‌های مرکب، پیشوندی مرکب، ربطی، شبه‌کمکی، غیر شخصی، وصفی، و عبارت‌های فعلی که از ترکیب اسامی، پیشوندها یا حروف اضافه با قطعات فعلی حاصل می‌شوند نیز با همین روش امکان‌پذیر خواهد بود، زیرا اسامی و حروف اضافه به طور جداگانه در واژه‌نامه موجود هستند.

گفتم، گفتی، گفت، گفتیم، گفتید، گفتند، نگفتم، نگفتی، نگفت، نگفتیم، نگفتید، نگفتند، بگفتم، بگفتی، بگفت، بگفتیم، بگفتید، بگفتند، مگفتم، مگفتی، مگفت، مگفتیم، مگفتید، مگفتند، می‌گفتم، می‌گفتی، می‌گفت، می‌گفتیم، می‌گفتید، می‌گفتند، نمی‌گفتم، نمی‌گفتی، نمی‌گفت، نمی‌گفتیم، نمی‌گفتید، نمی‌گفتند، گفته‌ام، گفته‌ای، گفته، گفته‌ایم، گفته‌اید، گفته‌اند، نگفته‌ام، نگفته‌ای، نگفته، نگفته‌ایم، نگفته‌اید، نگفته‌اند، بگفته‌ام، بگفته‌ای، بگفته است، بگفته‌ایم، بگفته‌اید، بگفته‌اند، مگفته‌ام، مگفته‌ای، مگفته است، مگفته‌ایم، مگفته‌اید، مگفته‌اند، می‌گفته‌ام، می‌گفته‌ای، می‌گفته، می‌گفته‌ایم، می‌گفته‌اید، می‌گفته‌اند، نمی‌گفته‌ام، نمی‌گفته‌ای، نمی‌گفته، نمی‌گفته‌ایم، نمی‌گفته‌اید، نمی‌گفته‌اند، می‌گویم، می‌گویی، می‌گوید، می‌گوییم، می‌گویید، می‌گویند، نمی‌گویم، نمی‌گویی، نمی‌گوید، نمی‌گوییم، نمی‌گویید، نمی‌گویند، بگویم، بگویی، بگوید، بگوییم، بگویید، بگویند، نگویم، نگویی، نگوید، نگوییم، نگویید، نگویند، گویم، گویی، گوید، گوییم، گویید، گویند، مگویم، مگویی، مگوید، مگوییم، مگویید، مگویند، بگو، نگو، گو، مگو

(الف)

آراستم، آراستی، آراست، آراستیم، آراستید، آراستند، نیاراستم، نیاراستی، نیاراست، نیاراستیم، نیاراستید، نیاراستند، بیاراستم، بیاراستی، بیاراست، بیاراستیم، بیاراستید، بیاراستند، میاراستم، میاراستی، میاراست، میاراستیم، میاراستید، میاراستند، می‌آراستم، می‌آراستی، می‌آراست، می‌آراستیم، می‌آراستید، می‌آراستند، نمی‌آراستم، نمی‌آراستی، نمی‌آراست، نمی‌آراستیم، نمی‌آراستید، نمی‌آراستند، آراسته‌ام، آراسته‌ای، آراسته است، آراسته‌ایم، آراسته‌اید، آراسته‌اند، نیاراسته‌ام، نیاراسته‌ای، نیاراسته، نیاراسته‌ایم، نیاراسته‌اید، نیاراسته‌اند، بیاراسته‌ام، بیاراسته‌ای، بیاراسته، بیاراسته‌ایم، بیاراسته‌اید، بیاراسته‌اند، میاراسته‌ام، میاراسته‌ای، میاراسته است، میاراسته‌ایم، میاراسته‌اید، میاراسته‌اند، می‌آراسته‌ام، می‌آراسته‌ای، می‌آراسته، می‌آراسته‌ایم، می‌آراسته‌اید، می‌آراسته‌اند، نمی‌آراسته‌ام، نمی‌آراسته‌ای، نمی‌آراسته، نمی‌آراسته‌ایم، نمی‌آراسته‌اید، نمی‌آراسته‌اند، می‌آرایم، می‌آرایی، می‌آراید، می‌آراییم، می‌آرایید، می‌آرایند، نمی‌آرایم، نمی‌آرایی، نمی‌آراید، نمی‌آراییم، نمی‌آرایید، نمی‌آرایند، بیارایم، بیارایی، بیاراید، بیاراییم، بیارایید، بیارایند، نیارایم، نیارایی، نیاراید، نیاراییم، نیارایید، نیارایند، آرایم، آرایی، آراید، آراییم، آرایید، آرایند، میارایم، میارایی، میاراید، میاراییم، میارایید، میارایند، بیارا، نیارا، آرا، میارا

(ب)

شکل (۴-۷) قطعات فعلیِ موجود از مصدر «گفتن» و «آراستن»

۴-۴-۱-۲ *نحوه‌ی ارائه‌ی واژه‌نامه*

یکی از نکات مهم در مرحله‌ی تشخیص خطای املایی، کارایی ساختارِ داده‌ی[۱] نگهداری واژه‌نامه است. ساختارِ نگهداری و ارائه‌ی واژه‌نامه باید از سرعت مناسبی در جستجوی واژه‌ها برخوردار باشد تا عمل تشخیص خطا هر چه سریع‌تر صورت گیرد. از این رو، این ساختار داده باید در حافظه نگهداری شود. از طرفی تعداد واژه‌های صحیح زبان عددی قابل توجه است و از این رو نگهداری تعداد زیادی واژه در حافظه مستلزم تخصیص حافظه‌ی بسیار خواهد بود که با توجه به محدود بودن منبعِ حافظه، ساختار نگهداری واژه‌نامه باید تا حد امکان از فضای بهینه جهت نگهداری واژه‌ها استفاده نماید.

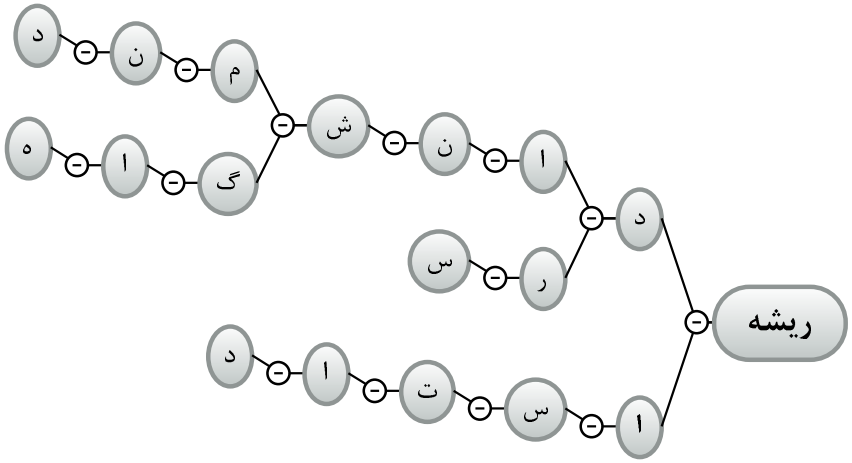
یک ساختار داده‌ی مناسب جهت نگهداری و ارائه‌ی واژه‌نامه می‌تواند استفاده از درخت[۲] الفبایی که حروف زبان گره‌های[۳] آن هستند باشد. نمونه‌ای از این درخت جهت نگهداری چهار واژه‌ی «دانشمند»، «دانشگاه»، «درس» و «استاد» در شکل (۴-۸) نشان داده شده است. همان طور که مشاهده می‌شود در این ساختار داده، زیررشته‌های مشترک از واژه‌ها، مانند زیررشته‌ی «دانش» در دو واژه‌ی «دانشمند» و «دانشگاه» یک بار در حافظه نگهداری می‌شود که این امر موجب صرفه‌جویی قابل توجه در میزان حافظه‌ی مصرفی می‌شود. بارگذاری واژه‌نامه با استفاده از این ساختار داده بر روی حافظه، حجم کمتری از نگهداری آن بر روی دیسک خواهد داشت. درخت الفبایی با نگهداری حروف در گره‌ها، پیچیدگی زمانی مناسبی در درج و حذف نیز دارد اما با توجه به نرخ پایین درج و حذف در کاربرد خطایابی، پیچیدگی زمانیِ جستجو است که در ساختار داده‌ی مورد استفاده اهمیت ویژه دارد.

---

۱ معادل فارسی عبارت انگلیسی Data Structure

۲ معادل فارسی واژه‌ی انگلیسی Tree

۳ معادل فارسی واژه‌ی انگلیسی Node



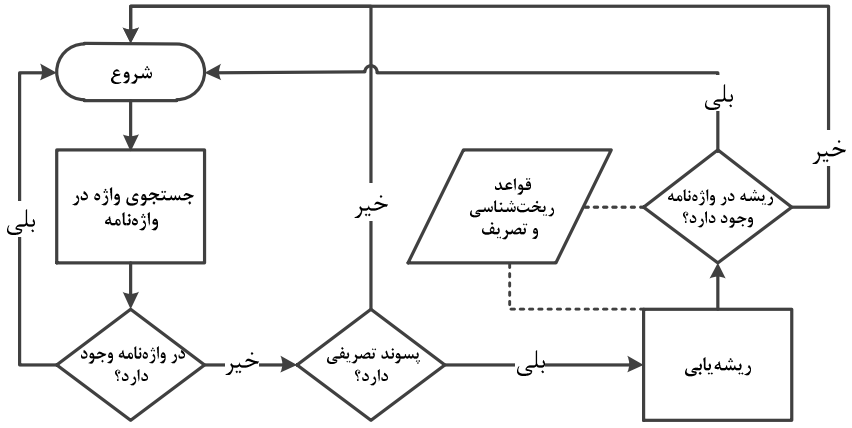
شکل (۴-۸) نمونه‌ی ساختار نگهداری واژه‌نامه در حافظه

۴-۴-۱-۳ *واژه‌های تصریفی*

جهت تشخیص وجود خطاهای املایی در متن، واژه‌های متن را در واژه‌نامه‌ای جامع از زبان جستجو می‌کنیم. واژه‌های تصریفی زبان فارسی در دو دسته‌ی تصریف فعل‌ها و تصریف واژه‌های غیر فعلی می‌گنجند که به تفصیل در فصل سوم مورد بررسی قرارگرفتند. در بخش قبل نیز پیشنهاد شد تا تصریف کامل فعل‌ها در زمان بارگذاری به واژه‌نامه افزوده شود و در این میان تنها واژه‌های تصریفی غیر فعلی باقی می‌مانند که در واژه‌نامه وجود ندارند اما از لحاظ املایی صحیح هستند.

با توجه به تعداد بسیار زیاد تصریف‌ها برای یک واژه، عملاً امکان افزودن تصریف‌های کامل صحیح واژه‌ها به واژه‌نامه امکان‌پذیر نیست. از این رو، ریشه‌یابیِ واژه‌های تصیفی غیر فعلی در زمان پردازش هر واژه پیشنهاد می‌شود.

با در نظر گرفتن وندهای تصریفی، ویژگی‌ها و قواعد ترکیب آن‌ها که در بخش ۳-۲-۲ از فصل سوم مورد بررسی قرار گرفتند، می‌توان فرایندی مشابه آن‌چه در شکل (۴-۹) آمده است را جهت تشخیص خطای املایی در واژه‌های تصریفی غیر فعلی در نظر گرفت.



شکل (۴-۹) فرایند تشخیص خطای املایی در واژه‌های تصریفی غیر فعلی

۴-۴-۲ تولید پیشنهاداتِ جایگزینی

پس از تشخیص واژه‌ی دارای خطای املایی، باید واژه‌هایی جهت جایگزینی با واژه‌ی مذکور ارائه شوند. جهت تولید پیشنهادات جایگزینی، با توجه به گونه‌های چهارگانه‌ی خطاهای املایی شامل درج، حذف، جابجایی و جایگزینی حروف، با فرض رخداد هر یک از این خطاهای املایی، مجموعه‌ای از واژه‌های جدید با درج و افزودن حروف به واژه‌ی دارای خطا، حذف حرف از واژه‌ی دارای خطا، جایگزین کرد حروف واژه‌ی دارای خطا با حروف دیگر و جابجایی دو حرف مجاور در واژه‌ی دارای خطا ایجاد می‌شود. چون ممکن است برخی از واژه‌های تولید شده واژه‌ی صحیحی نباشند، واژه‌های موجود در مجموعه‌ی پیشنهادات در واژه‌نامه مورد جستجو قرار می‌گیرند تا از میان آن‌ها واژه‌های صحیح زبان استخراج گردد.

نکته‌ی دیگری که باید مورد توجه قرار گیرد، فاصله‌ی ویرایشی مورد نظر در خطایابی است. فاصله‌ی ویرایشیِ میان دو واژه، کمینه‌ی تغییرات لازم شامل درج، حذف، جابجایی و جایگزینی حروف جهت تولید یک واژه از واژه‌ای دیگر است. به عنوان مثال فاصله‌ی ویرایشی میان «چدا» و «خدا» به علت نیاز به یک عمل جایگزینی حروف، معادل یک است و فاصله‌ی ویرایشی میان «واژک‌شناسی» و «داژک‌شنایی» به علت نیاز به یک عمل جایگزینی و یک عمل حذف حروف معال دو است.

فاصله‌ی ویرایشی میان پیشنهادات جایگزینی و واژه‌ی خطا از مسائلی است که باید با ظرافت مورد توجه قرار گیرد. از طرفی تولید پیشنهادات در فاصله‌های ویرایشی بیشتر، کیفیت تصحیح خطا را افزایش می‌دهد و می‌توان خطاهای پیچیده‌تر را نیز اصلاح نمود اما از طرف دیگر تولید پیشنهاد در فاصله ویرایشی بالاتر، بسیار زمان‌گیرتر و پرهزینه‌تر است[۱].

این روش تولید پیشنهادات جایگزینی منجر به تولید پیشنهادات جایگزینی‌ای خواهد شد که در واژه‌نامه موجود هستند. اما در روش پیشنهادی این کتاب، واژه‌های تصریفی غیر فعلی در واژه‌نامه نیستند. از این رو، هنگامی که یک واژه‌ی تصریفی مانند «امیدهایشان» به اشتباه «انیدهایشان» نوشته شود، پیشنهادات تولید شده که پسوند تصریفیِ ترکیبی «هایشان» را به دنبال دارند در واژه‌نامه موجود نبوده و از لیست پیشنهادات نهایی حذف می‌گردند. بنا بر این، واژه‌های غیر فعلی تصریفی باید به طور خاص مورد توجه قرار گیرند.

خطاهای املایی که به دلیل فاصله‌گذاری اشتباه، خصوصاً در مواردی که منجر به درج فاصله میان اجزای واژه می‌شود رخداده‌اند، مانند «زبا نفارسی»، نیز با روشِ معمولِ تولیدِ پیشنهاد به طور بهینه قابل تصحیح نیستند و باید به طور خاص مورد توجه قرار گیرند.

۴-۴-۲-۱ *واژه‌های غیر فعلی تصریفی*

با توجه به وجود نداشتن واژه‌های غیر فعلیِ تصریفی در واژه‌نامه، یک راه می‌تواند تولید پیشنهاداتِ جایگزینی به طور معمول و سپس ریشه‌یابی پیشنهادات جهت جستجو در واژه‌نامه باشد. با این که در این روش دقت مناسبی در تولید پیشنهادات فراهم می‌گردد، اما با توجه به تعداد بسیار زیاد تصریف‌های و پیشنهادات ممکن، حتی در فاصله‌های ویرایشی

---

۱ با توجه به تعداد حروف الفبای فارسی که با در نظر گرفتن نیم‌فاصله و شکل‌های مختلف همزه معادل ۳۸ عدد است، یک واژه در فاصله‌ی ویرایشی ۱ می‌تواند (تعداد حروف واژه‌ی خطا) × ۳۸ پیشنهادِ جایگزینی با جایگزینی یک حرف از آن با حروف دیگر، (۱ + تعداد حروف واژه‌ی خطا) × ۳۸ پیشنهادِ جایگزینی با درج و افزودن یک حرف به آن، تعداد (۱- تعداد حروف واژه‌ی خطا) پیشنهادِ جایگزینی با جابجایی دو حرف مجاور، و (تعداد حروف واژه‌ی خطا) پیشنهادِ جایگزینی با حذف یک واژه از آن داشته باشد که در مجموع یک واژه در فاصله‌ی ویرایشی ۱ در زبان فارسی می‌تواند معادل ۳۷ + (تعداد حروف واژه‌ی خطا)×۷۶ پیشنهاد جایگزینی داشته باشد. در حالی که در فاصله‌ی ویرایشی ۲، تعداد ۴۲۹۴ + (تعداد حروف واژه‌ی خطا)×۸۵۸۸ + ۲(تعداد حروف واژه‌ی خطا)×۶۰۰۴ پیشنهادِ جایگزینی برای یک واژه وجود خواهد داشت که بسیار بسیار بیشتر از تعداد پیشنهادها در فاصله‌ی ویرایشی ۱ است.

کم، این عمل بسیار زمان‌گیر خواهد شد و کاربران را از استفاده از خطایاب خودکار دل‌زده می‌کند.

روش دیگر می‌تواند ریشه‌یابیِ واژه‌ی دارای خطای املایی و سپس تولید پیشنهادات برای ریشه‌ی آن باشد و در نهایت پیشنهادات تولید شده، با توجه به قواعد واژک‌شناسی، همانند واژه‌ی دارای خطای املایی صَرف شوند. این فرایند بسیار سریع‌تر از روش قبلی است اما هنگامی که بخش تصریفی واژه (پسوند) دارای خطای املایی باشد، عملاً ریشه‌یابی نتایج نادرستی را در پی‌خواهد داشت و نتایج تولید پیشنهادات با این روش نامطمئن خواهد بود. با این حال، استفاده‌ی هوشمندانه از این روش کارایی مناسب‌تری را به دنبال خواهد داشت.

۴-۴-۲-۲ *فاصله‌گذاری*

در بخش ۳-۲-۴ از فصل سوم، هفت گونه از خطاهای املایی ناشی از فاصله‌گذاری اشتباه به تفصیل مورد بررسی قرار گرفتند. با توجه با این نکته که خطاهای ناشی از فاصله‌گذاری اشتباه، واژه‌ی پیشین و/یا واژه‌ی پَسین واژه‌ی جاری را شامل می‌شوند، تولید پیشنهادات جایگزینی به طور مجزا و با در نظر گرفتن هر یک از هفت خطای ممکن، برای یک ترکیب سه واژه‌ای (واژه‌ی دارای خطای املایی، واژه‌ی پسین و واژه‌ی پیشین) پیشنهاد می‌گردد. در این حالت می‌توان هر یک از هفت گونه خطاهای ذکر شده را در این ترکیب سه‌تایی مورد آزمون قرار داد تا به یک یا چند ترکیب سه‌تایی درست (هر سه واژه در واژه‌نامه موجود باشند) از واژه‌ها رسید.

فرایند صحت‌سنجیِ فاصله‌گذاری می‌تواند برای واژه‌هایی از متن که در واژه‌نامه موجود هستند نیز انجام پذیرد. با این کار، در صورتی که ترکیب‌های اشتقاقی با فاصله‌گذاری صحیح در واژه‌نامه موجود باشند، همچنین قواعد فاصله‌گذاری ترکیب‌های تصریفی نیز به درستی رعایت شوند، می‌توان چالش فاصله‌گذاری در ترکیب‌های زبان فارسی را تا حد قابل قبولی مرتفع گرداند.

۴-۴-۲-۳ *تکمیل زیررشته‌ی آغازین*

تکمیل زیررشته‌ی آغازین می‌تواند روشی دیگر در تولید پیشنهادات جایگزینی، در صورت استفاده از ساختارهای داده با پشتیبانی از تکمیل خودکارِ زیررشته‌ی آغازین، مانند درخت الفبایی با حروف به عنوان گره، باشد. این روش به تنهایی نمی‌تواند پیشنهادات

جامعی را جهت تصحیح خطاهای املایی ارائه دهد، ولی در ترکیب با روش‌های دیگر می‌تواند بر غنای پیشنهادات بیافزاید. به عنوان مثال با استفاده از این روش می‌توان برای واژه‌ی «معن» پیشنهادات «معنا»، «معناگرا» و «معناشناسی» را در زمان بسیار کوتاه، در حالی که برخی از آن‌ها در فاصله‌ی ویرایشی زیادی هستند، ارائه نمود. این روش در مواجهه با واژه‌های دارای خطای املایی، که خطا در ابتدای واژه، خصوصاً حرف اول واژه رخ داده باشد، نتایج نامطمئنی ارائه می‌نماید.

۴-۴-۳ رتبه‌بندی پیشنهادات

تعداد پیشنهاداتِ جایگزینی تولید شده برای یک واژه‌ی دارای خطای املایی می‌تواند چنان زیاد باشد که انتخاب واژه‌ی صحیح (واژه‌ای که نوشتار اشتباه آن واژه‌ی دارای خطای املایی شده است) از میان واژه‌های پیشنهادی را از سوی کاربر با مشکل مواجه سازد. از طرفی برخی کاربردهای خطایابی املایی خودکار، مانند خطایابی در نویسه‌خوانی نودری، در تعامل با کاربر اقدام به تصحیح خطا نمی‌کنند و عمل تصحیح نیز به صورت خودکار صورت می‌پذیرد. از این رو، رتبه‌بندی پیشنهادات ارائه شده و سپس رتبه‌بندی آن‌ها بر اساس رتبه فرایندی ضروری در خطایابی املایی خودکار است.

فرایند رتبه‌بندی پیشنهادات جایگزینی به محاسبه‌ی میزان شایستگی هر واژه جهت جایگزینی با واژه‌ی دارای خطای املایی می‌پردازد. به عنوان مثال ممکن است واژه‌ی «می‌شود» پیشنهاد بهتری نسبت به واژه‌ی «میوه» برای جایگزینی با واژه‌ی خطای «میوشد» باشد. رتبه‌بندی می‌تواند بر اساس فاصله ویرایشی و میزان شباهت رشته‌ای، بسامد تکرار و احتمالات خطانویسی، مشابهت رشته‌ای یا آوایی حروف و بخش‌ها، مبتنی بر قوانین یا ترکیبی از این روش‌ها باشد.

روش‌های رتبه‌بندیِ پیشنهادات جایگزینیِ معمول و مطرح، چالش‌های خاص زبان فارسی را مورد نظر قرار نداده‌اند و نتایج مناسبی برای زبان فارسی به دست نمی‌دهند. از این رو، در این کتاب به ارائه‌ی روشی جهت رتبه‌بندی پیشنهادات جایگزینی با پشتیبانی از چالش‌های خاص زبان فارسی ارائه کرده‌ایم. این روش چیدمانِ صفحه‌ی کلید فارسی و تاثیر آن بر خطاهای حروف‌چینی، حروف که به صورت ترکیبی درج می‌شوند، هم‌آواها، هم‌شکل‌ها، و الگوهای توزیع خطاهای املایی در زبان فارسی را مورد توجه قرار می‌دهد.

۴-۴-۳-۱ *الگوی توزیع خطا در زبان فارسی*

در بخش قبل به بررسی الگوها و احتمالات رخداد خطاهای املایی در زبان فارسی پرداختیم. با در نظر گرفتن نرخ رخداد خطاهای املاییِ تکی، احتمال رخداد هر یک از خطاها را بر اساس فرمول (۴-۱) تعریف می‌شود.

$$P_{se}(t) = \frac{Total\ Number\ \text{of}\ Misspellings\ Caused\ by\ Single\text{-}error\ t}{Total\ Number\ of\ Misspellings}$$ فرمول (۴-۱)

پیشنهادات جایگزینی‌ای که بر اساس رخداد یک خطای املاییِ تکیِ محتمل‌تر در واژه‌ی دارای خطا به وجود آمده‌اند، به احتمال بیشتری پیشنهاد صحیح جهت جایگزینی با آن هستند. بنابراین، پیشنهادات بهتر باید فاصله‌ی کمتری با واژه‌ی دارای خطا داشته باشند. از این رو، فاصله‌ی هر خطای تکی طبق فرمول (۴-۲) تعریف می‌شود.

$$Distance_{se}(t) = \sum P_{se} - P_{se}(t)$$ فرمول (۴-۲)

احتمال رخداد و فاصله‌ی هر یک از خطاهای املاییِ تکی بر اساس تحلیل پیکره‌های بزرگ زبان فارسی در بخش قبل، در جدول (۴-۴) آمده است. لازم به ذکر است که با توجه به اطلاعات ارائه شده در جدول (۴-۳)، خطاهای املایی تکیِ زبان فارسی ۷۸٪ از خطاهای املایی را شامل می‌شوند و از این رو احتمال رخداد خطاهای تکی و در نتیجه مجموع احتمال رخداد خطاهای جایگزینی، حذف، درج، و جابجایی ۰/۷۸ است.

جدول (۴-۴) احتمال رخداد و فاصله‌ی خطاهای املاییِ تکی در زبان فارسی

نوع خطا	درصد رخداد	احتمال رخداد	فاصله
جایگزینی	٪ ۵۳	۰/۴۱۳	۰/۳۶۷
حذف	٪ ۲۷	۰/۲۱۱	۰/۵۶۹
درج	٪ ۱۳	۰/۱۰۱	۰/۶۷۹
جابجایی	٪ ۷	۰/۰۵۵	۰/۷۲۵
مجموع	**٪ ۷۸**	**۰/۷۸۰**	**۲/۳۴۰**

۴-۴-۳-۲ چیدمان صفحه‌ی کلید

جایگزینی حروف با یکدیگر تحت تاثیر موقعیت قرارگیری آن‌ها بر روی صفحه کلید است، به طوری که احتمال جایگزینی دو حرف مجاور بیشتر از دو حرفی است که از یکدیگر فاصله‌ی بیشتری دارند [57]. از این فاصله به فاصله‌ی میان نویسه‌ها[1] تعبیر می‌شود. برای توضیح بیشتر، یک نمونه از چیدمان صفحه کلید انگلیسی در شکل (۴-۱۰) و یک نمونه از چیدمان صفحه کلید فارسی در شکل (۴-۱۱) نشان داده شده‌اند.

2	q	w	e	r	t	y	u	i	o	p	[	]
1	a	s	d	f	g	h	j	k	l	;	‘	
0	z	x	c	v	b	n	m	,	.	/		
y/x	0	1	2	3	4	5	6	7	8	9	10	11

شکل (۴-۱۰) نمونه‌ی چیدمان صفحه کلید انگلیسی

2	ض	ص	ث	ق	ف	غ	ع	ه	خ	ح	ج	چ	پ
1	ش	س	ی	ب	ل	ا / آ	ت	ن	م	ک	گ		
0	ظ	ط	ز / ژ	ر / ؤ	ذ / إ	د / أ	ئ / ء	و	.	/			
y/x	0	1	2	3	4	5	6	7	8	9	10	11	12

شکل (۴-۱۱) نمونه‌ی چیدمان صفحه کلید فارسی

فاصله‌ی میان دو نویسه‌ی c_1 که در محل (x_1, y_1) و c_2 که در محل (x_2, y_2) بر روی چیدمان صفحه کلید قرار گرفته‌اند، با استفاده از فاصله‌ی اقلیدسی[2] [58] همانند شکل (۴-۱۲) محاسبه می‌شود. به عنوان مثال فاصله‌ی میان نویسه‌ی "a" که در مکان (0, 1) و نویسه‌ی "p" که در مکان (9, 2) قرار گرفته‌اند برابر است با:

$$EuclideanDistance(p, a) = \sqrt{(9-0)^2 + (2-1)^2} = 9.05538$$

۱ معادل فارسی عبارت انگلیسی Character Distance

۲ معادل فارسی عبارت انگلیسی Euclidean Distance

```
1: EuclideanDistance(c1 , c2)
2: {
3:     (x1, y1) = Cartesian coordination of c1 in keyboard layout
4:     (x2, y2) = Cartesian coordination of c2 in keyboard layout
5:     de(c1 , c2) = sqrt((x2 − x1)^2 + (y2 − y1)^2)
6:     return de
7: }
```

شکل (۴-۱۲) الگوریتم محاسبه‌ی فاصله‌ی اقلیدسی میان دو نویسه

همچنین فاصله‌ی میان نویسه‌ی «ظ» که در مکان (0, 0) از چیدمان صفحه کلید فارسی و نویسه‌ی «چ» که در مکان (11, 2) قرار گرفته‌اند برابر است با:

$$EuclideanDistance\left(\text{ظ}, \text{چ}\right) = \sqrt{(11-0)^2 + (2-0)^2} = 11.18033$$

۴-۴-۳-۲-۱ کلید شیفت

در چیدمان صفحه کلید فارسی، درج برخی از نویسه‌ها نیازمند استفاده و نگهداشتن کلید شیفت[۱] پیش از درج آن‌ها است. این نویسه‌ها به *نویسه‌های ترکیبی*[۲] موسومند که در شکل (۴-۱۱) با "/" از نویسه‌ی پایه جدا شده‌اند، مانند نویسه‌ی «ژ» و «آ».

فاصله‌ی میان نویسه‌های ترکیبی و دیگر نویسه‌ها به آسانی توسط فاصله‌ی اقلیدسی قابل محاسبه نیست. توجه با این نکته نیز ضروری است که احتمال عدم نگه‌داشتن کلید شیفت به اشتباه، بیشتر از احتمال نگه‌داشتن اشتباه آن است. به عنوان مثال، احتمال درج حرف «ز» به جای «ژ» به دلیل عدم نگه‌داشتن کلید شیفت بیشتر از درج «ژ» به جای «ز» است. این حالت در مورد فاصله‌ی میان نویسه‌های ترکیبی و دیگر نویسه‌ها، خواه ترکیبی یا غیرترکیبی، نیز صادق است. شکل (۴-۱۳) روش پیشنهادی جهت محاسبه‌ی فاصله‌ی میان نویسه‌ها را با پشتیبانی از نویسه‌های ترکیبی نشان می‌دهد. در این شکل c_1 نویسه‌ی اشتباه و c_2 نویسه‌ی مورد نظر جهت جایگزینی است.

---

۱ نوشتار فارسی واژه‌ی انگلیسی Shift

۲ نزدیک‌ترین معادل فارسی عبارت انگلیسی Upper Character

```
1:  CharacterDistance(c1 , c2)
2:  {
3:      dmax = maximum Euclidean distance of characters on keyboard layout
4:      dmin = minimum Euclidean distance of characters on keyboard layout
5:      davg = (dmax + dmin) / 2
6:      if (c2 is an UpperCharacter)
7:          if (c1 is LowerCharacter of c2)
8:              costs = davg
9:          else if (c1 is a LowerCharacter)
10:             costs = EuclideanDistance(c1 , c2) + davg
11:         if (c1 is an UpperCharacter)
12:             costs = EuclideanDistance(c1 , c2)
13:     else
14:         if (c1 is UpperCharacter of c2)
15:             costs = davg
16:         else if (c1 is a LowerCharacter)
17:             costs = EuclideanDistance(c1 , c2)
18:         if (c1 is an UpperCharacter)
19:             costs = EuclideanDistance(c1 , c2) + davg
20:     return ds
21: }
```

شکل (۴-۱۳) الگوریتم محاسبه‌ی فاصله میان نویسه‌ها با پشتیبانی از نویسه‌های ترکیبی

۴-۴-۳-۳ هم‌آواها

در بخش ۳-۳ از فصل سوم هم‌آواهای زبان فارسی مورد بررسی قرار گرفتند و تعداد بسیار زیاد هم‌آواها به عنوان یکی از چالش‌های خطایابی املایی در زبان فارسی بر شمرده شد. در بخش ۴-۳ نیز نرخ بالای رخداد خطاهای املایی‌ای که بر اثر هم‌آوایی حروف و جایگزینی اشتباه آن‌ها رخ داده است مورد بررسی قرار گرفت.

خطاهای املایی هم‌آوا به دلیل جایگزینی حروف هم‌آوا رخ می‌دهند. این حروف عموماً مجاور یکدیگر در چیدمان صفحه کلید فارسی نیستند. بنابراین، تنها استفاده از فاصله‌ی میان نویسه‌ها که در شکل (۴-۸) ارائه شد، نمی‌تواند در رتبه‌بندی این گونه خطاها چندان مؤثر باشد. از این رو، استفاده از کمینه‌ی فاصله‌ی میان نویسه‌ها در چیدمان صفحه کلید به عنوان فاصله‌ی میان حروف هم‌آوا در هر خانواده‌ی آوایی پیشنهاد می‌شود. شکل (۴-۹) فاصله‌ی میان نویسه‌های بهبود یافته را با پشتیبانی از حروف هم‌آوا نشان می‌دهد.

۴-۴-۳-۴ هم‌شکل‌ها

حروف هم‌شکل بیش از ۷۰ ٪ از حروف زبان فارسی را تشکیل داده‌اند. حروف هم‌شکل، خصوصاً هنگامی که یک کاربر تازه‌کار هنگام درج حروف به دنبال آن‌ها بر روی صفح کلید می‌گردد یا حین بازشناسی نوری نویسه‌ها، ممکن است اشتباهاً به جای یکدیگر به کار روند. به علاوه، حروف‌چین‌های حرفه‌ای نیز متون دست‌نوشته را بدون توجه به معنی، همان‌گونه که به نظر می‌آیند حروف‌چینی می‌کنند. بنابراین، ممکن است حروف هم‌شکل به جای یکدیگر مورد استفاده قرار گیرند [42].

حروف هم‌شکلِ زبان فارسی عموماً مجاور یکدیگر در چیدمان صفحه کلید هستند و از همین رو، فاصله‌ی میان نویسه‌ها برای جایگزینی این حروف با یکدیگر کم خواهد بود. اما مواردی مانند حروف هم‌شکلِ خانواده‌ی همزه، مجاور یکدیگر نیستند. بنابراین، همانند هم‌آواها، در نظر گرفتن کمینه‌ی فاصله‌ی میان نویسه‌ها در چیدمان صفحه کلید به عنوان فاصله‌ی میان حروف هم‌شکل در هر خانواده پیشنهاد می‌شود.

فاصله‌ی میان نویسه‌های بهبود یافته با پشتیبانی از نحوه‌ی چیدمان صفحه کلید، نویسه‌های ترکیبی، هم‌آواها، و هم‌شکل‌ها در شکل (۴-۹) نشان داده شده است. این فاصله عددی نرمال در بازه‌ی [۰ و ۱] خواهد بود که ۰ نشان‌گر شباهت کامل (یکسان بودن) و ۱ نشان‌گر عدم شباهت کامل است.

```
1:  ExtendedCharacterDistance(c1 , c2)
2:  {
3:      dmin = minimum Euclidean distance of characters on keyboard layout
4:      if (c1 ∈ HomophoneFamily(c2) or c1 ∈ HomomorphFamily(c2))
5:          costs = dmin
6:      else
7:          costs = CharacterDistance(c1 , c2)
8:      ds(c1 , c2) = costs / dmax
9:      return ds
10: }
```

شکل (۴-۱۴) الگوریتم محاسبه‌ی فاصله‌ی میان نویسه‌های بهبود یافته

۴-۴-۳-۵ فاصله‌ی دو واژه

فاصله‌ی میان نویسه‌های پیشنهاد شده تنها فاصله‌ی جایگزینی یک حرف با حرف دیگر را با در نظر گرفتن چیدمان صفحه کلید، نویسه‌های ترکیبی، هم‌آواها، و هم‌شکل‌ها محاسبه می‌کند. محاسبه‌ی فاصله‌ی درج یک حرف، حذف یک حرف، و جابجایی دو حرف مجاور، به صورت آماری و بر اساس جدول (۴-۴) پیشنهاد می‌گردد.

شکل (۴-۱۰) نحوه‌ی محاسبه‌ی فاصله‌ی میان دو واژه را نشان می‌دهد. در این شکل نحوه‌ی محاسبه‌ی فاصله‌ی میان دو واژه‌ی q و l نشان داده شده است که q واژه‌ی دارای خطای املایی است که می‌تواند یک یا چند خطای املاییِ تکی داشته باشد و l یک پیشنهاد جایگزینی است. در این جا نیز یک عدد نرمال در بازه‌ی [۰ و ۱] به عنوان فاصله‌ی میان دو واژه محاسبه خواهد شده که ۰ نشان‌گر یکسان بودن دو واژه و ۱ نشان‌گر متفاوت بودن کامل آن دو است.

1: *StringDistance (q , l)*

2: *{*

3: $f_k(0 , 0) = 0$

4: $d(q_i , \varepsilon) = Distance_{se}(omission)$ */* From Equation 2 */*

5: $d(\varepsilon , l_j) = Distance_{se}(insertion)$

6: $t(q_i , l_j) = Distance_{se}(transposition)$

7: $d(q_i , l_j) = ExtendedCharacterDistance(c_1 , c_2) \times Distance_{se}(substitution)$

8: $f_k(i , j) = min((f_{wf}(i-1, j)+ d(q_i , \varepsilon)), (f_{wf}(i, j-1)+ d(\varepsilon , l_j)), (f_{wf}(i-1, j-1)+ d(q_i , l_j)), (f_{dl}(i-2, j-2)+ t(q_i , l_j)))$

9: ***return*** $f_k(|q| , |i|) / max(|q| , |i|)$

10: *}*

شکل (۴-۱۵) الگوریتم محاسبه‌ی فاصله‌ی میان دو واژه

۴-۴-۳-۶ بسامد واژه‌ها

بسامد تکرار واژه‌ها، اگر به نسبت صحیح استخراج شده باشند، می‌توانند میزان کاربرد آن‌ها را مشخص سازند. از طرفی، واژه‌ای که بسامد و در نتیجه کاربرد بیشتری در زبان داشته باشد، چون احتمال رخداد آن در زبان بیشتر است، احتمالاً واژه‌ای مناسب‌تر برای

جایگزینی با واژه‌ی دارای خطای املایی، نسبت به واژه‌ی کم‌کاربردتر خواهد بود. شکل (۴-۱۱) نحوه‌ی محاسبه‌ی امتیاز جایگزینی هر واژه را جهت جایگزین شدن با واژه‌ی دارای خطای املایی با در نظر گرفتن فاصله‌ی رشته‌ای میان دو واژه همچنین بسامد آن‌ها نشان می‌دهد.

1: *ReplacementScore (q , l)*

2: *{*

3: $f_k(0 , 0) = 0$

4: $d(q_i , \varepsilon) = Distance_{se}(omission)$ /* From Equation 2 */

5: $d(\varepsilon , l_j) = Distance_{se}(insertion)$

6: $t(q_i , l_j) = Distance_{se}(transposition)$

7: $d(q_i , l_j) = ExtendedCharacterDistance(c_1 , c_2) \times Distance_{se}(substitution)$

8: $f_k(i , j) = min((f_{wf}(i-1 , j) + d(q_i , \varepsilon)), (f_{wf}(i , j-1) + d(\varepsilon , l_j)), (f_{wf}(i-1 , j-1) + d(q_i , l_j)), (f_{at}(i-2 , j-2) + t(q_i , l_j)))$

9: ***return*** $f_k(|q| , |i|) / max(|q| , |i|)$

10: *}*

شکل (۴-۱۶) الگوریتم محاسبه‌ی امتیاز جایگزینی

۴-۵ ارزیابی

در این بخش روش پیشنهاد شده جهت محاسبه‌ی امتیاز جایگزینی واژه‌ها ارزیابی می‌شود.

۴-۵-۱ روش ارزیابی

برای ارزیابی روش پیشنهاد شده، واژه‌های دارای خطای املایی از پیکره‌های بزرگ زبان استخراج و بازبینی شدند. نرخ، میزان و الگوهای خطاهای املایی در زبان فارسی به تفصیل در بخش ۴-۳ مورد بررسی قرار گرفتند. در حوزه‌ی خطایابی املایی و ارائه دادن یک لیست رتبه‌بندی شده از پیشنهادات جایگزینی، نمی‌توان به راحتی مشخص نمود که چه پیشنهادی باید در لیست باشد و چه پیشنهادی نباید باشد. تنها می‌توان پیرامون پیشنهاد

صحیح (واژه‌ی اصلی که خطا نوشته شده) با قطعیت اظهار نظر کرد که این پیشنهاد باید در سر لیست باشد. از این رو، هیچ منفیِ حقیقی‌ای[۱] در لیست پیشنهادی وجود ندارد، بنابراین نمی‌توان از روش‌های سنتی ارزیابی مانند محاسبه‌ی فراخوانی[۲] و دقت[۳] در ارزیابی روش پیشنهادی استفاده نمود. برخی پژوهشگران [22, 30, 39, 42, 59] از شاخص ارائه شده در فرمول (۴-۳) جهت ارزیابی کاراییِ خطایابیِ املایی استفاده می‌کنند. این شاخص، دقت خطایاب را در رتبه‌بندیِ پیشنهاد صحیح در مکان nام یا نزدیکتر به سر لیست ارائه می‌دهد.

$$P_n = \frac{Number\ of\ Times\ Rank_{Correct\ Suggestion} \leq n}{|Misspellings|} \qquad \text{فرمول (۴-۳)}$$

با این حال، شاخص ارائه شده در فرمول (۴-۳) نمی‌تواند به عنوان یک شاخص واحد برای ارزیابی کارایی کلی خطایاب به کار رود. به عنوان مثال در صورتی که دقت قرارگیری پیشنهاد صحیح در رتبه‌ی دهم یا پایین‌تر محاسبه شود، قرارگیری پیشنهاد صحیح در سر لیست یا رتبه‌ی نهم تفاوتی در مقدار این شاخص ایجاد نخواهد کرد. از این رو، دو شاخص دیگر تحت عنوان دقت *میانگین*[۴] [60] که در در فرمول (۴-۴) و *رتبه‌ی وارانه‌ی میانگین*[۵] [61] که در در فرمول (۴-۵) نشان داده شده‌اند نیز مورد استفاده قرار می‌گیرند.

$$MAP = \frac{1}{10}\sum_{n=1}^{10} P_n \qquad \text{فرمول (۴-۴)}$$

$$MRR = \frac{1}{|Misspellings|}\sum \frac{1}{Rank_{Correct\ Suggestion}} \qquad \text{فرمول (۴-۵)}$$

با توجه با این که پیکره‌ی مورد استفاده برای ارزیابی، همان پیکره‌ای است که برای محاسبه‌ی الگوها و احتمالاتِ خطاهای املایی زبان فارسی استفاده شده است (شامل حدود

۱ معادل فارسی عبارت انگلیسی True Negative
۲ معادل فارسی واژه‌ی انگلیسی Recall
۳ معادل فارسی واژه‌ی انگلیسی Precision
۴ معادل فارسی عبارت انگلیسی Mean Average Precision (MAP)
۵ معادل فارسی عبارت انگلیسی Mean Reciprocal Rank (MRR)

۵۳۰ خطای املایی)، از روش احراز متقاطع[۱] با تازَنی[۲] پنج مرحله‌ای استفاده شده تا نتایج ارزیابی قابل اتکا، معنی‌دار[۳] و منصفانه باشند. برای این کار پیکره به صورت یک‌نواخت پنج قسمت تقسیم شد. الگوها، احتمالات و توزیع خطاهای املایی در هر قسمت جداگانه محاسبه و برای رتبه‌بندی در چهار قسمت دیگر پیکره مورد استفاده قرار گرفت. این فرایند برای هر پنج بخش از پیکره تکرار و در نهایت میانگینِ حسابیِ نتایج محاسبه شد.

۴-۵-۲ نتایج

در این بخش، روش پیشنهاد شده با پژوهش‌ها و کارهای دیگر پیرامون خطایابی املایی مقایسه خواهد شد. از میان پژوهش‌های بررسی شده در بخش ۴-۲، روش نسیم و همکارانش [42] بر اساس کلیدهای شباهت برای زبان اُردو بنا شده بود که تولید و استخراج کلیدهای مشابهت برای زبان فارسی مشکل و غیرقعی است. روش بَراری و همکارانش [44]، روش قاسمی‌زاده و همکارانش [45]، همچنین روش شالان و همکارانش [43] فاقد جزئیات کافی برای پیاده‌سازی بودند و نتایج گزارش شده نیز با توجه به تفاوت پیکره‌ها قابل مقایسه نبودند. از طرفی پژوهش‌گران دیگر[10, 16, 26] روش‌های مبتنی بر کلیدهای مشابهت و کلیدهای آوایی [27-30] را مورد ارزیابی قرار داده‌اند که این روش‌ها نتایجی ضعیف‌تر را در مقایسه با روش‌های آماری و مبتنی بر شباهت رشته‌ای به دست داده‌اند. خطایاب فَیلی و همکارانش [46, 47] و خطایاب ویرا [48]، به جهت در دسترس نبودن کتابخانه و رابط‌های برنامه‌نویسی، همچنین عدم امکان پیاده‌سازی کامل آن‌ها، امکان ارزیابیِ ماشینی را دارا نبودند. از میان روش‌های بررسی شده و مطرح، تنها روش‌های هَمینگ [49]، لِوِنشتاین [25]، دَمِرا-لِوِنشتاین [9]، وَگنِر-فیشِر [24] و جَرو-وینکلِر [53] قابل پیاده‌سازی یا در دسترس بودند. بنابراین، تنها این موارد را می‌توان مورد بررسی و مقایسه قرار داد. روش وَگنِر-فیشِر که امکان مقداردهی میزان تأثیر خطاهای حذف، درج، جابجایی، و جایگزینی را دارد، جهت مقایسه‌ی هر چه منصفانه‌تر، با الگوهای زبان فارسی که در بخش ۴-۳ ارائه شدند، مقداردهی شده است. روش پیشنهادی در این کتاب در پنج سطح مورد بررسی قرار گرفته است: (۱) رتبه‌بندی بر اساس الگوهای خطا در زبان فارسی،

---

۱ معادل فارسی عبارت انگلیسی Cross-validation

۲ معادل فارسی واژه‌ی انگلیسی Folding

۳ معادل فارسی واژه‌ی انگلیسی Significant

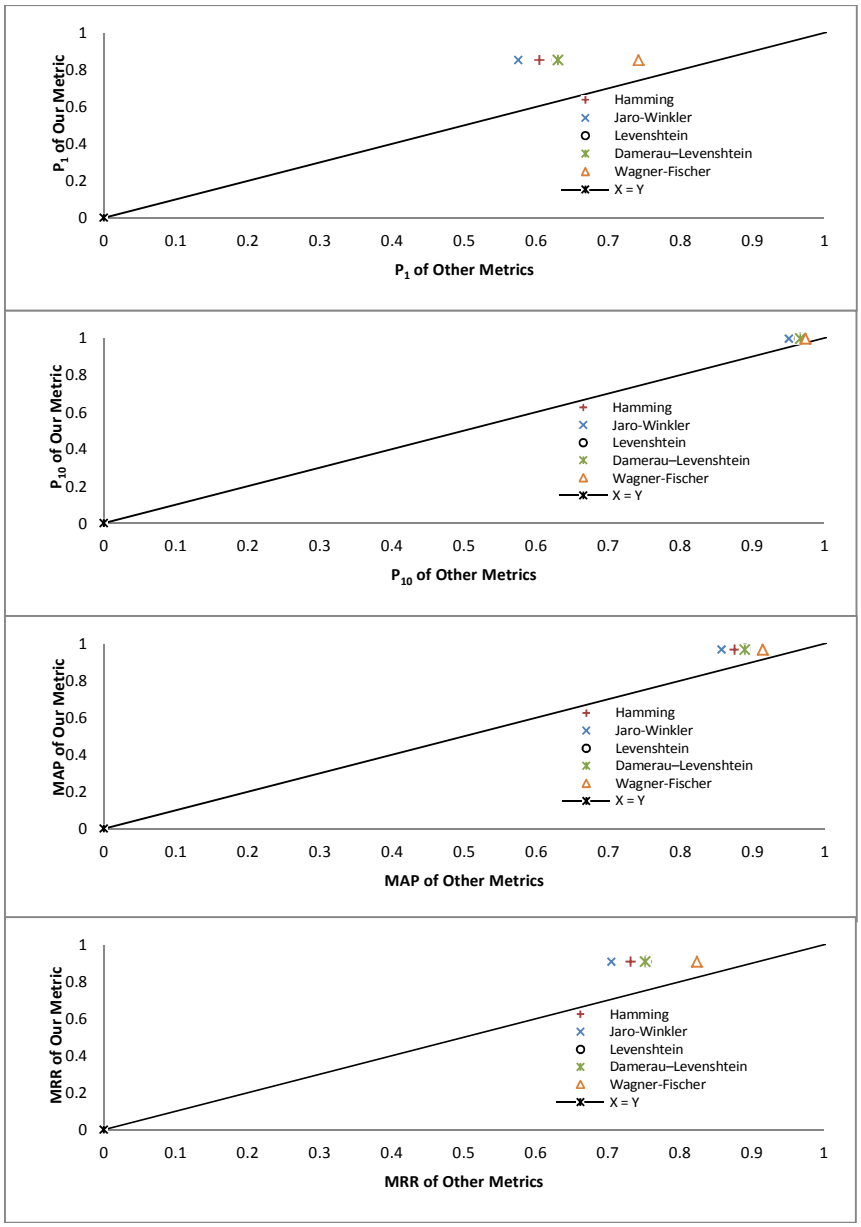
(۲) رتبه‌بندی بر اساس الگوهای خطا و چیدمان صفحه‌کلید، (۳) رتبه‌بندی بر اساس الگوهای خطا، چیدمان صفحه‌کلید و تأثیر هم‌آواها، (۴) رتبه‌بندی بر اساس الگوهای خطا، چیدمان صفحه‌کلید، تأثیر هم‌آواها و تأثیر هم‌شکل‌ها، و (۵) رتبه‌بندی بر اساس الگوهای خطا، چیدمان صفحه‌کلید، تأثیر هم‌آواها، تأثیر هم‌شکل‌ها و در نظر گرفتن بسامد واژه‌ها. در ادامه، از شماره‌های ذکر شده برای هر سطح، جهت ارجاع با آن‌ها استفاده خواهد شد.

جدول (۴-۵) نتایج ارزیابی و مقایسه‌ی روش پیشنهادی را با روش‌های دیگر نشان می‌دهد. *P1* دقت رتبه‌بندیِ پیشنهاد صحیح را در سر لیست نشان می‌دهد، *P10* دقت رتبه‌بندیِ پیشنهاد صحیح را در رتبه‌ی دهم یا پایین‌تر نشان می‌دهد. *MAP* دقت میانگین و *MRR* رتبه‌ی وارانه‌ی میانگین را نشان می‌دهد.

جدول (۴-۱) ارزیابی و مقایسه‌ی روش رتبه‌بندی پیشنهادی با روش‌های دیگر

روش‌های رتبه‌بندی	P_1	P_{10}	MAP	MRR
هَمینگ	۰/۶۰۴	۰/۹۶۴	۰/۸۷۵	۰/۷۳۲
جَرو-وینکلِر	۰/۵۷۵	۰/۹۵۱	۰/۸۵۷	۰/۷۰۵
لِونِشتاین	۰/۶۳۰	۰/۹۶۷	۰/۸۹۰	۰/۷۵۳
دَمِرا-لِونِشتاین	۰/۶۳۰	۰/۹۶۷	۰/۸۸۹	۰/۷۵۲
وَگنِر-فیشِر	۰/۷۴۲	۰/۹۷۴	۰/۹۱۴	۰/۸۲۴
روش پیشنهادی (۱)	۰/۷۴۲	۰/۹۷۴	۰/۹۱۴	۰/۸۲۴
روش پیشنهادی (۲)	۰/۸۲۶	۰/۹۸۹	۰/۹۴۲	۰/۸۷۱
روش پیشنهادی (۳)	۰/۸۴۸	۰/۹۹۷	۰/۹۶۱	۰/۹۰۳
روش پیشنهادی (۴)	۰/۸۵۳	۰/۹۹۷	۰/۹۶۸	۰/۹۰۸
روش پیشنهادی (۵)	۰/۸۶۱	۱	۰/۹۷۸	۰/۹۱۷

شکل (۴-۱۲) به مقایسه‌ی روش پیشنهاد شده با روش‌های دیگر می‌پردازد. روش پیشنهاد شده در این کتاب در عرض محور مختصات و روش‌های دیگر در ط.ل محور مختصات به نمایش در آمده‌اند و هر نقطه بر محور نمایانگر یک روش است. نقاطی که بالای خطِ منصف محور ($x=y$) قرار گرفته‌اند نتایجی ضعیف‌تر از روش پیشنهادی ارائه کرده‌اند.



شکل (۴-۱۷) مقایسه‌ی روش‌های مختلف رتبه‌بندی

۶-۴ پژوهش‌های آتی

در این کتاب روشی جهت خطایابی املایی خودکار در زبان فارسی ارائه شده است که تا حد امکان واژک‌شناسی و ویژگی‌های خاص زبان فارسی را در سطح صَرف از زبان پوشش داده بود. در ادامه می‌توان سطوح دیگر از زبان را نیز تحت پوشش قرار داد. در ابتدا لازم است که خطایابی، تصحیح خطا و ارائه‌ی پیشنهاد، وابسته به بافت واژه‌ی دارای خطا صورت پذیرد. این کار می‌تواند با استخراج یک مدل آماری از هم‌نشینی‌های رایج در زبان فارسی صورت پذیرد که جهت خطایابی، تصحیح خطا و ارائه‌ی پیشنهاد از این مدل استفاده شود. لازم است خطاهای حوزه‌ی نحو که به نحوه‌ی چیدمان واژه‌ها در شکل‌گیری جملات می‌پردازد نیز تشخیص داده شده و تصحیح شوند. برای امکان تشخیص و تصحیح خودکار خطا در لایه‌ی نحو به دادگان زبانی در این سطح از زبان نیاز است که متاسفانه این دادگان به ندرت در زبان فارسی موجود و در دسترس هستند. از نمونه‌های دادگان مورد نیاز در سطح نحو می‌توان به الگوهای رایج خطاهای نحوی و ویرایشی، پیکره‌های قطعه‌بندی[۱]، پیکره‌های ظرفیت[۲] واژگان، پیکره‌های تجزیه‌ی وابستگی[۳]، و پیکره‌های موجودیت‌های اسمی[۴] اشاره نمود. با توجه به عدم وجود چنین دادگانی در زبان فارسی، قدم اول می‌تواند اقدام به تولید و گردآوری چنین اطلاعاتی باشد.

خطایابی و تصحیح خطا می‌تواند در سطوح معنا، کاربردشناسی، مباحثه و گفتمان نیز ادامه یابد؛ اما با توجه به عدم وجود تجربه و پژوهش‌های کافی در سطح نحو، در حال حاضر نمی‌توان چشم‌انداز دقیقی از خطایابی خودکار در این سطوح از زبان ارائه داد. جهت‌گیری پژوهش‌های معتبر و به‌روز، در حوزه‌ی پردازش زبان در سطوح معنا، کاربردشناسی، مباحثه و گفتمان به سمت استفاده از مدل‌های آماری است؛ از این رو تنها می‌توان متصور شده که آینده و راه‌حل خطایابی و تصحیح خطا در سطوح معنا، کاربردشناسی، گفتمان و حتی نحو در زبان فارسی نیز روش‌های آماری باشند که البته نیازمند حجم بسیار زیادی از دادگان مربوطه در هر زمینه هستند.

---

[۱] معادل فارسی واژه‌ی انگلیسی Chunk

[۲] معادل فارسی واژه‌ی انگلیسی Valency

[۳] معادل فارسی عبارت انگلیسی Dependency Parse

[۴] معادل فارسی واژه‌ی انگلیسی Name Entity

مراجع

[1] J. Hutchins, "Retrospect and prospect in computer-based translation," presented at the Proceedings of MT Summit VII, 1999.

[2] D. Jurafsky and J. Martin, *Speech and language processing*, 2000.

[3] C. D. Manning and H. Schtze, *Foundations of statistical natural language processing*: MIT Press, 1999.

[۴] ف. آقاگل‌زاده, «تحلیل گفتمان انتقادی و ادبیات»، ادب پژوهی، شماره ۱، صفحه‌ی ۱۷-۲۷، ۱۳۸۶.

[5] J. W. Backus, *et al.*, "Revised report on the algorithm language ALGOL 60," *Commun. ACM*, vol. 6, pp. 1-17, 1963.

[6] O. Kashefi, *et al.*, "A Novel String Distance Metric for Ranking Persian Respelling Suggestions," *Lang Resources & Evaluation*, 2009.

[7] C. M. Eastman and D. S. McLean, "On the need for parsing ill-formed input," *Comput. Linguist.*, vol. 7, pp. 257-257, 1981.

[8] C. Young, *et al.*, "An analysis of ill-formed input in natural language queries to document retrieval systems," *Inf. Process. Manage.*, vol. 27, pp. 615-622, 1991.

[9] F. J. Damerau, "A technique for computer detection and correction of spelling errors," *Commun. ACM*, vol. 7, pp. 171-176, 1964.

[10] E. Galli and H. Yamada, "Experimental studies in computer-assisted correction of unorthographic text," *IEEE Transactions on Engineering Writing and Speech*, vol. 11, pp. 75-84, 1968.

[11] A. Hanson, *et al.*, "Context in word recognition," *Pattern Recognition*, vol. 8, pp. 35-45, 1976.

[12] J. Pollock and A. Zamora, "Collection and Characterization of Spelling Errors in Scientific and Scholarly Text," *Journal of the American Society for Information Science*, vol. 34, pp. 51-58, 1983.

[13] C. Sterling, "Spelling errors in context," *British journal of psychology(1953)*, vol. 74, pp. 353-364, 1983.

[14] R. Mitton, "Spelling checkers,spelling correctors and the misspellings of poor spellers," *Inf. Process. Manage.*, vol. 23, pp. 495-505, 1987.

[15] M. Kyongho, *et al.*, "Typographical and orthographical spelling error correction," presented at the Proceedings of 2nd International Conference on Language Resources and Evaluation, Athens, Greece, 2000.

[16] J. J. Pollock and A. Zamora, "Automatic spelling correction in scientific and scholarly text," *Commun. ACM,* vol. 27, pp. 358-368, 1984.

[17] J. L. Peterson, "A note on undetected typing errors," *Commun. ACM,* vol. 29, pp. 633-637, 1986.

[18] K. Kukich, "Techniques for automatically correcting words in text," *ACM Computing Surveys,* vol. 24, pp. 378- 439, 1992.

[19] E. Yannakoudakis and D. Fawthrop, "The Rules of Spelling Errors," *Information Processing and Management,* vol. 19, pp. 87-99, 1983.

[20] J. L. Peterson, "Computer programs for detecting and correcting spelling errors," *Commun. ACM,* vol. 23, pp. 676-687, 1980.

[21] J. L. Peterson, *Computer Programs for Spelling Correction*: Springer-Verlag New York, Inc. Secaucus, NJ, USA, 1980.

[22] R. Mitton, "Ordering the suggestions of a spellchecker without using context," *Natural Language Engineering,* vol. 15, pp. 173-192, 2009.

[23] R. Mitton, "Spellchecking by computer," *Journal of Simplified Spelling Society,* vol. 20, pp. 4-11, 1996.

[24] R. A. Wagner and M. J. Fischer, "The String-to-String Correction Problem," *J. ACM,* vol. 21, pp. 168-173, 1974.

[25] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady,* vol. 10, pp. 707-710, 1966.

[26] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," presented at the Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference, Boston, Massachusetts, 1959.

[27] M. K. Odell and R. C. Russell, "U.S. Patent Numbers, 1,261,167 (1918) and 1,435,663 (1922)," Washington, D.C. Patent, 1918.

[28] J. Zobel and P. Dart, "Phonetic string matching: lessons from information retrieval," presented at the Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval, Zurich, Switzerland, 1996.

[29] D. Holmes and M. C. McCabe, "Improving precision and recall for Soundex retrieval," in *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, 2002, pp. 22-26.

[30] V. J. Hodge and J. Austin, "A Comparison of Standard Spell Checking Algorithms and a Novel Binary Neural Approach," *IEEE Trans. on Knowl. and Data Eng.,* vol. 15, pp. 1073-1081, 2003.

[31] L. G. Means, "Cn yur cmputr raed ths?," presented at the Proceedings of the second conference on Applied natural language processing, Austin, Texas, 1988.

[32] K. Min, *et al.*, "Typographical and orthographical spelling error correction," presented at the Proceedings of 2nd International Conference on Language Resources and Evaluation, Athens, Greece, 2000.

[33] D. E. Rumelhart, *et al.*, "Learning internal representations by error propagation," in *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*, ed: MIT Press, 1986, pp. 318-362.

[34] A. R. Golding, "A Bayesian hybrid method for context-sensitive spelling correction," presented at the Proceedings of the Third Workshop on Very Large Corpora, Cambridge, MA., 1996.

[35] E. S. Ristad and P. N. Yianilos, "Learning String-Edit Distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, pp. 522-532, 1998.

[36] G. Hirst and A. Budanitsky, "Correcting real-word spelling errors by restoring lexical cohesion," *Nat. Lang. Eng.*, vol. 11, pp. 87-111, 2005.

[37] J. R. Ullman, "A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words," *Computer Journal*, vol. 20, pp. 141-147, 1977.

[38] A. R. Golding and D. Roth, "A Winnow-Based Approach to Context-Sensitive Spelling Correction," *Mach. Learn.*, vol. 34, pp. 107-130, 1999.

[39] E. Brill and R. C. Moore, "An improved error model for noisy channel spelling correction," presented at the Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Hong Kong, 2000.

[40] R. Morris and L. L. Cherry, "Computer detection of typographical errors," *IEEE Transactions on Professional Communication*, vol. PC-18, pp. 54-64, 1975.

[41] C. Comeau and W. J. Wilbur, "Non-Word Identification or Spell Checking Without a Dictionary," *Journal Of The American Society For Information Science and Technology*, vol. 55, pp. 169-177, 2004.

[42] T. Naseem and S. Hussain, "A novel approach for ranking spelling error corrections for Urdu," *Language Resources and Evaluation*, vol. 41, pp. 117-128, 2007.

[43] K. Shaalan, *et al.*, "Towards Automatic Spell Checking for Arabic," presented at the Conference on Language Engineering, Cairo, Egypt, 2003.

[44] L. Barari and B. QasemiZadeh, "CloniZER Spell Checker Adaptive, Language Independent Spell Checker," presented at the AIML, Cairo, Egypt, 2005.

[45] B. QasemiZadeh, *et al.*, "Adaptive Language Independent Spell Checking using Intelligent Traverse on a Tree," presented at the IEEE Conference on Cybernetics and Intelligent Systems, 2006.

[46] H. Faili, "Detection and Correction of Real-Word Spelling Errors in Persian Language," presented at the 6th IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'10), August 2010.

[47] N. Ehsan and H. Faili, "Towards Grammar Checker Development for Persian Language," presented at the 6th IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLP-KE'10), August 2010.

[48] SRRF. *Vira*. Available: http://www.spellchecker.ir/

[49] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Tech. J.*, vol. 29, pp. 147-160, 1950.

[50] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Communications of the ACM*, vol. 18, pp. 341-344, 1975.

[51] E. Ukkonen, "Algorithms for approximate string matching," *Inf. Control*, vol. 64, pp. 100-118, 1985.

[52] W. Masek, "A faster algorithm computing string edit distances," *JOURNAL OF COMPUTER ANDVSVBIEM SCIENCES*, 1980.

[53] M. A. Jaro, "Advances in record linking methodology as applied to the 1985 census of Tampa Florida," *Journal of the American Statistical Society*, vol. 84, pp. 414-420, 1989.

[54] W. Winkler and Y. Thibaudeau, "An Application of the Fellegi-Sunter Model of Record Linkage to the 1990 US Decennial Census," *Research Report RR91/09, US Bureau of the Census*, 1991.

[55] W. Winkler, "The state of record linkage and current research problems," *Statistics of Income Division, Internal Revenue Service Publication R*, vol. 4, 1999.

[56] A. AleAhmad, *et al.*, "Hamshahri: A standard Persian text collection," *Know.-Based Syst.*, vol. 22, pp. 382-387, 2009.

[57] K. Min and W. H. Wilson, "Syntactic recovery and spelling correction of ill-formed sentences," presented at the Proceedings of 3rd Conference of the Australasian Cognitive Science, 1995.

[58] T. Heath, "The Thirteen Books of Euclid's Elements, Vol. 1," *New York: Doner*, 1956.

[59] K. Toutanova and R. C. Moore, "Pronunciation modeling for improved spelling correction," presented at the Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, 2002.

[60] S. Gerald, *Automatic text processing*: Addison-Wesley Longman Publishing Co., Inc., 1988.

[61] P. B. Kantor and E. M. Voorhees, "The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text," *Inf. Retr.*, vol. 2, pp. 165-176, 2000.

ضمیمه اول

مبدل اعداد

سینا ایروانیان (sina@sinairv.com)

۱-۱ مقدمه

از بخش‌های مطرح در زمینه‌ی درک متون فارسی توسط رایانه، تشخیص خودکار اعداد در انواع گوناگون صورت‌های نوشتاری در متن فارسی است. این صورت‌های گوناگون، عبارت‌اند از *صورت نوشتاری رقمی* و *صورت نوشتاری حرفی*.

صورت نوشتاری رقمی به معنی بیان اعداد فارسی با استفاده از ارقام است. صورت‌های رقمی قابل تشخیص نظام پیشنهادی در جدول (۱-۱) خلاصه شده‌اند.

جدول (۱-۱) صورت نوشتاری رقمی قابل تشخیص نظام پیشنهادی

شرح	مثال
اعداد با ارقام فارسی	۱۶۴۸۱۹۵
اعداد با ارقام انگلیسی	1648195
اعداد با جدا کننده‌ی هزارگان با ارقام فارسی	۱٬۶۴۸٬۱۹۵
اعداد با جدا کننده‌ی هزارگان با ارقام انگلیسی	1,648,195
اعداد اعشاری با ارقام فارسی	۲۰٬۸۷۳٫۲۵
اعداد اعشاری با ارقام انگلیسی	20,873.25
اعداد با ارقام عربی	١٢٣٤٥٦٧٨٩

صورت نوشتاری حرفی به معنی بیان اعداد فارسی با استفاده از حروف است. صورت های حرفی قابل تشخیص راهکار پیشنهادی در جدول (۱-۲) خلاصه شده‌اند.

جدول (۱-۲) صورت نوشتاری حرفی قابل تشخیص نظام پیشنهادی

شرح	مثال
اعداد طبیعی	هزار و سیصد و شصت و دو
اعداد اعشاری با ذکر واژه‌ی «ممیز»	پنجاه ممیز هفت ده هزارم
اعداد اعشاری	صد و بیست و پنج صدم

۱-۲ تبدیل و یکسان‌سازی انواع نوشتارهای رقمی

در این بخش به بررسی صورت‌های مختلف نمایش ارقام خواهیم پرداخت، سپس راهکارهای تبدیل ارقام از صورت‌های نوشتاری مختلف به یکدیگر را بررسی خواهیم کرد.

۱-۲-۱ انواع صورت‌های نمایش ارقام

ارقام فارسی عبارت‌اند از ۰۱۲۳۴۵۶۷۸۹. در جدول یونی‌کد این ارقام از کد 6F0 هگزادسیمال تا کد 6F9 هگزادسیمال قرار گرفته‌اند. ارقام عربی عبارت‌اند از: ۰۱۲۳٤٥٦۷۸۹. این ارقام نیز در جدول یونی‌کد از کد 660 هگزادسیمال تا کد 669 هگزادسیمال قرار گرفته‌اند. ارقام انگلیسی عبارتند از: 0123456789. کد این ارقام در جدول یونی‌کد از کد 30 تا 39 هگزادسیمال قرار گرفته‌اند. بنابراین، برای تشخیص این‌که آیا نویسه‌ی دلخواهی، نمایانگر یک رقم است، کافی است کد آن نویسه را با سه بازه‌ی ذکر شده در بالا مقایسه کنیم. مقدار عددی‌ای که هر نویسه‌ی رقمی اختیار کرده است را می‌توان از تفریق کد آن نویسه با کد رقم صفر در همان بازه به دست آورد. مثلا کد رقم ۴ فارسی 6F4 است و کد رقم صفر فارسی نیز 6F0 است که تفریق این دو، عدد ۴ را به دست می‌دهد. عکس قضیه‌ی فوق نیز بدین معنی بر قرار است که برای به دست آوردن نویسه‌ی ۴ فارسی کافی است عدد ۴ را با کد نویسه‌ی صفر فارسی جمع کنیم. این دو قاعده مبنای تبدیل ارقام فارسی، عربی، و انگلیسی به یکدیگر را تشکیل می‌دهند. در شکل (۱-۱) الگوریتم تبدیل ارقام دلخواه به ارقام انگلیسی آمده است.

```
Function ConvertDigitToEnglish

input in_digit: character
output out_digit: character
locals value: integer

if (6F0 <= code_of(in_digit) <= 6F9) then
 value = code_of(in_digit) – 6F0
 out_digit = char_of(30 + value)
else if (660 <= code_of(in_digit) <= 669) then
 value = code_of(in_digit) – 660
 out_digit = char_of(30 + value)
else
 out_digit = in_digit
```

شکل (۱-۱) الگوریتم تبدیل ارقام به انگلیسی

۱-۲-۲ دیگر نویسه‌های مورد استفاده در نوشتار رقمی اعداد

در نوشتارهای رقمی اعداد، به غیر از نویسه‌های ارقام، از نویسه‌های دیگری نیز استفاده می‌شود. برای نمونه می‌توان از جداکننده‌های هزارگان، ممیز، و علائم مثبت و منفی (+ و -) نام برد. همچنین در نماد علمی (بالاخص در نوشتار انگلیسی اعداد) از نویسه‌ی e برای نمای ۱۰ استفاده می‌شود. مثلا 2e+3 به معنای عدد ۲۰۰۰ می‌باشد. نویسه‌های به‌کارگرفته شده برای جداکننده‌های هزارگان و ممیز در نوشتارهای فارسی و انگلیسی متفاوت‌اند. با استفاده از یک نگاشت ساده می‌توان این دو را به یکدیگر تبدیل کرد. در جدول (۱-۳) این نویسه‌ها با یکدیگر مقایسه شده‌اند.

جدول (۱-۳) مقایسه‌ی نویسه‌های به کار رفته در نمایش اعداد

شرح	نویسه‌ی فارسی/ عربی	نویسه‌ی انگلیسی
جداکننده‌ی هزارگان	، (0x66C)	, (0x2C)
ممیز اعشاری	٫ (0x66B)	. (0x2E)

گفتنی است که در زبان‌های فارسی و عربی نویسه‌های یاد شده در بالا منحصر به فرد نیستند. در بسیاری مواقع در میان ارقام فارسی از نویسه‌های انگلیسی استفاده می‌شود و

همین‌طور به جای ممیز اعشاری، استفاده از نویسه‌ی اِسلَش[1] (/) در فارسی بسیار رایج است که در روال‌های تبدیل نوشتارها این موضوع را نیز باید مد نظر داشت.

۱-۲-۳ یکسان‌سازی نوشتارهای رقمی اعداد

در تمامی بخش‌های نظام پیشنهادی برای تبدیل اعداد، با نوشتار رقمی اعداد و روال‌های مرتبط با آن سر و کار خواهیم داشت. چون همان گونه که ذکر شد، نوشتار رقمی می‌تواند به سه صورت فارسی، عربی، و انگلیسی باشد بنابراین بهتر است که یک صورت نوشتاری از سه صورت نوشتاری فوق را برای صورت نرمال نوشتار رقمی برگزینیم، و تمامی الگوریتم‌ها را بر پایه‌ی آن نوشتار پیاده‌سازی کنیم و در نهایت نتیجه را به هر نوشتار دلخواه تبدیل کنیم. از آن‌جا که در زبان‌های برنامه‌نویسی متداول، توابع کتابخانه‌ای فراوانی بر پایه‌ی زبان انگلیسی وجود دارد، بنابراین نوشتار انگلیسی را برای صورت نرمال نمایش اعداد برمی‌گزینیم.

۱-۳ تبدیل اعداد از نوشتار رقمی به نوشتار حرفی

در این بخش به شیوه‌ی تبدیل اعداد طبیعی از نوشتار رقمی به نوشتار حرفی خواهیم پرداخت، و الگوریتم‌های مرتبط ارائه خواهیم داد. برای مثال، الگوریتمی که تا پایان این بخش معرّفی می‌کنیم، می‌تواند ورودی «۲۱۳۴۰۰۱» را به «دو میلیون و صد و سی و چهار هزار و یک» تبدیل کند. در این بخش ابتدا به چگونگی تبدیل هر یک از اجزای بسیط سازنده‌ی یک عدد از نوشتار رقمی به نوشتار حرفی به کمک یک نگاشت ساده می‌پردازیم سپس با بررسی ساختار کلّی یک عدد طبیعی به معرّفی الگوریتم تبدیل اعداد طبیعی از نوشتار رقمی به نوشتار حرفی خواهیم پرداخت.

۱-۳-۱ تشکیل نگاشتی از مقادیر به واژگان

برای یکی از ابتدایی‌ترین مراحل تبدیل اعداد از نوشتار رقمی به نوشتار حرفی یک نگاشت از مقادیر عددی کلیدی به واژگانی که در زبان فارسی برای آن‌ها استفاده می‌شود، می‌سازیم. منظور از مقادیر عددی کلیدی مقادیری است که تمامی اعداد با ترکیبی از آن‌ها

---

۱ نوشتار فارسی واژه‌ی انگلیسی slash

ساخته می‌شود. این اعداد عبارت‌اند از:

- اعداد ۰ تا ۹
- اعداد ۱۰ تا ۱۹
- ضرایب ۱۰ از ۲۰ تا ۹۰
- ضرایب ۱۰۰ از ۱۰۰ تا ۹۰۰
- اعداد ۱۰۰۰ (هزار)، ۱۰،۰۰۰،۰۰۰ (میلیون)، ۱۰،۰۰۰،۰۰۰،۰۰۰ (میلیارد)، و ۱۰،۰۰۰،۰۰۰،۰۰۰،۰۰۰ (تریلیارد)

زین‌پس در شبه‌کدهای این بخش، نگاشت یاد شده را با تابعی به نام MapNum2Str خواهیم شناخت. مثلا MapNum2Str(200) رشته‌ی «دویست» را به دست خواهد داد.

۱-۳-۲ تبدیل اعداد حداکثر سه رقمی به نوشتار فارسی

اعداد طبیعی حداکثر سه رقمی یکی از اجزای اصلی تشکیل دهنده‌ی اعداد بزرگ‌ترند، به طوری که اعداد بزرگ‌تر به نحوی ترکیب این اعداد با یک مجموعه ضریب هستند. برای مثال، تبدیل عدد زیر را در نظر بگیرید.

۷۵۲،۰۰۳ ← هفتصد و پنجاه و دو هزار و سه

عدد طبیعی حداکثر ۳ رقمی — ضریب — عدد طبیعی حداکثر ۳ رقمی

الگوریتم تبدیل با استفاده از نگاشت یاد شده در شکل (۱-۲) آمده است.

۱-۳-۳ تبدیل اعداد طبیعی در حالت کلی

اکنون با در دست داشتن الگوریتم تبدیل اعداد طبیعی حداکثر ۳ رقمی، می‌توان مبدّل اعداد طبیعی در حالت کلی‌تر را طراحی کرد. این الگوریتم نیز از لحاظ ساختار، بسیار شبیه الگوریتم تبدیل اعداد طبیعی حداکثر ۳ رقمی است. این الگوریتم در شکل (۱-۳) ارائه شده است.

```
Function ConvertUpTo3Digits

inputs n: integer
outputs s: string
locals c: integer

if (n > 99) then
 c = n / 100;
 s = MapNum2Str(c * 100);
 n = n - (c*100);
 if(n < = 0) return;
 else s = s + " و ";
if (n > 20) then
 c = n / 10;
 s = s + MapNum2Str(c * 10);
 n = n - c*10;
 if (n <= 0) return;
 else s = s + " و ";
if (n > 0) then
 s = s + MapNum2Str(n);
```

شکل (۱-۲) الگوریتم تبدیل اعداد حداکثر سه رقمی از حالت رقمی به حالت حرفی

۱-۳-۴ تبدیل اعداد حقیقی از نوشتار رقمی به نوشتار حرفی

هدف این قسمت تعمیم الگوریتم تبدیل اعداد طبیعی به حالت حقیقی است. در واقع، به‌دنبال آن هستیم که چنان‌چه عدد ورودی، یک عدد با ممیز اعشاری بود، بتوانیم آن را با موفقیت به حالت نوشتار فارسی تبدیل کنیم. برای این منظور ابتدا باید بررسی کنیم که آیا عدد ورودی یک عدد طبیعی است یا اعشاری. چنان‌چه ورودی یک عدد طبیعی بود، از روال ConvertIntegers که در بخش قبل توضیح داده شد، می‌توان استفاده کرد، در غیر این صورت با انجام چند عملیات ساده می‌توان عدد اعشاری را به اجزای طبیعی خرد کرد و کار را با روال آشنای ConvertIntegers پیش برد. بدین منظور ابتدا عدد را به رشته‌ی انگلیسی (نرمال) تبدیل می‌کنیم و از مکان ممیز (نویسه‌ی نقطه در انگلیسی) رشته را به دو بخش تقسیم می‌کنیم. این الگوریتم در شکل (۱-۴) آمده است. در الگوریتم شکل (۱-۴) از تابعی به نام CreateOrdinalNumber استفاده شده است. این تابع، رشته‌ی نوشتار عدد را می‌گیرد و رشته‌ی نوشتار ترتیبی معادل را به دست می‌دهد. برای مثال، «سه» را به «سوم» و «میلیون» را به «میلیونیم» تبدیل می‌کند.

```
Function ConvertIntegers

inputs n: integer
outputs s: string
locals c: integer

if (n < 0) then
 s = "منهای ";
 n = -n;
if (n == 0) then
 s = MapNum2Str(n);
 return;
if (n > 999999999999)then
 c = n / 1000000000000;
 s = s + ConvertUpTo3Digits(c) + " " + MapNum2Str(1000000000000);
 n = n - (c * 1000000000000);
 if(n < = 0) return;
 else s = s + " و ";
if (n > 999999999) then
 c = n / 1000000000;
 s = s + ConvertUpTo3Digits(c) + " " + MapNum2Str(1000000000);
 n = n - (c * 1000000000);
 if(n < = 0) return;
 else s = s + " و ";
if (n > 999999)then
 c = n / 1000000;
 s = s + ConvertUpTo3Digits(c) + " " + MapNum2Str(1000000);
 n = n - (c * 1000000);
 if(n < = 0) return;
 else s = s + " و ";
if (n > 999)then
 c = n / 1000;
 s = s + ConvertUpTo3Digits(c) + " " + MapNum2Str(1000);
 n = n - (c * 1000);
 if(n < = 0) return;
 else s = s + " و ";
if (n > 0) then
  s = s + ConvertUpTo3Digits(n);
```

شکل (۱-۳) الگوریتم تبدیل اعداد طبیعی در حالت کلی

```
Function ConvertRealNumbers

inputs r: real
outputs s: string
locals r_str: string
 dot_index: integer
 n1: integer
 n2: integer
 order: integer

r_str = RealToString(r);
dot_index = index-of-point-in(r_str);
if (dot_index >= 0) then
 n1 = the-number-before-dot
 n2 = the-number-after-dot
 if (n1 != 0) then
 s = ConvertIntegers(n1) + " ممیز ";
 s = s + ConvertIntegers(n2);
 order = 10 ^ number-of-digits(n2);
 s = s + " " + CreateOrdinalNumber(order);
else
 s = ConvertIntegers(r as integer);
```

شکل (۱-۴) الگوریتم تبدیل اعداد حقیقی از نوشتار رقمی به نوشتار فارسی

۱-۴ تبدیل اعداد از نوشتار حرفی به نوشتار رقمی

در این بخش به بررسی روش تبدیل اعداد از نوشتار حرفی به نوشتار رقمی می‌پردازیم، و الگوریتم‌های آن را ارائه می‌دهیم. این قسمت، اصلی‌ترین و بااهمّیت‌ترین قسمت تشخیص اعداد است، زیرا آن‌چه در زمینه درک خودکار متون فارسی اهمّیت دارد، تبدیل عدد نوشته شده به صورت حرفی در متون، به قالب قابل فهم توسّط ماشین است. اعداد در نوشتار رقمی با یک تجزیه‌ی حرفی ساده به قالب قابل فهم توسّط ماشین تبدیل می‌شوند. به همین دلیل، بر اهمّیت درک اعداد نوشته شده به صورت حرفی تأکید می‌کنیم. در این قسمت پس از بررسی ساختار اعداد نوشته شده به صورت حرفی و معرّفی اجزای بسیط این ساختار، به بررسی الگوریتم‌های تبدیل اعداد برای عددهای طبیعی، عددهای اعشاری با ذکر واژه‌ی «ممیز» و اعداد اعشاری و کسری در حالت کلّی می‌پردازیم.

۱-۴-۱ ساختن نگاشت از رشته‌های بسیط اعداد به مقادیر متناظر

همانند آن‌چه در مورد تبدیل اعداد از نوشتار رقمی به حرفی بحث شد، در این قسمت نیز به عنوان ماده‌ی خام تمامی الگوریتم‌ها، نیاز به نگاشتی از رشته‌های بسیط اعدادی که در متن‌های فارسی به کار می‌روند به مقادیر عددی متناظرشان وجود دارد. بدین منظور رشته‌ی اعداد، به همراه فرم ترتیبی‌شان به نگاشت افزوده خواهند شد. رشته‌هایی که باید در نگاشت مورد بحث قرار بگیرند، از قرار زیرند:

- رشته‌های «صفر» و «صفرم».
- رشته‌های «یک» تا «نه» به همراه حالت ترتیبی «یکم»، «اول» تا «نهم».
- رشته‌های «ده» تا «نوزده» به همراه حالت ترتیبی.
- ضرایب ۱۰ از «بیست» تا «نود» به همراه حالت ترتیبی.
- ضرایب ۱۰۰ از «صد» تا «نهصد» به همراه حالت ترتیبی.
- رشته‌های «هزار»، «میلیون»، «میلیارد»، «تریلیارد».
- برخی پیشوندهای خاص، مانند «پان» که به عدد ۵ نگاشته شده است.

در صورتی که قرار است الگوریتم تبدیل علاوه بر شیوه‌های رسمی بیان اعداد، شیوه‌های غیر رسمی و محاوره‌ای را نیز شامل شود، می‌بایست عبارت‌های محاوره‌ای و غیر رسمی نیز به این نگاشت افزوده شوند. مثلا، در نگاشت فوق علاوه بر «شش» رشته‌ی «شیش» نیز باید افزوده شود. همچنین، علاوه بر «پانصد» رشته‌ی «پونصد» را نیز باید به نگاشت افزود. زین‌پس در شبه‌کدهای این مستند، این نگاشت را با تابعی به نام MapStr2Num خواهیم شناخت.

۱-۴-۲ یافتن قطعه‌های متن شامل رشته‌های اعداد

فرض کنید، متن نسبتاً بزرگی در اختیار دارید که در نقاطی از آن چند عدد به حالت نوشتاری حرفی وجود دارند، و هدف تشخیص و به دست آوردن مقدار آن‌ها باشد. رشته‌هایی که این اعداد را تشکیل می‌دهند همان رشته‌های موجود در دامنه‌ی نگاشت فوق است به انضمام رشته‌هایی که در ادامه آمده‌اند:

- حرف «و» که حرف ربط بین اجزای اعداد است. مانند «صد و ده».
- حرف «م» در انتهای واژه. این حرف عدد ترتیبی و همین‌طور کسری می‌سازد. مانند «پنجم»، یا، «سه هفتم».

- واژه‌ی «ممیز» که گاهی اوقات از آن برای بیان ممیز استفاده می‌شود. مانند «صد و بیست ممیز بیست و پنج صدم».
- واژه‌ها ی «منفی» و «منهای» که نشان دهنده‌ی اعداد منفی هستند.
- واژه‌ی «نیم» به معنای «پنج دهم».

توجه کنید که در میان رشته‌های فوق حرف «و» هم می‌تواند یک واژه‌ی جدا باشد، و هم این که در انتهای واژه ظاهر شده باشد (گرچه دومی از لحاظ املایی خطا به حساب می‌آید؛ امّا چون خطای بسیار رایجی است، قابلیت تشخیص این وضعیت نیز در الگوریتم اعمال شده است). همچنین حرف «م» نیز باید در انتهای واژه ظاهر شود، به شرطی که واژه‌ی پیش از آن در دامنه‌ی نگاشت موجود باشد.

با این تفاسیر در داخل متن محدوده‌ای که در آن احتمالاً عددی ظاهر شده باشد تشخیص داده می‌شود، و چون هنوز روی محتویات این محدوده هیچ تحلیلی صورت نگرفته، آن را «قطعه» می‌نامیم و به واژه‌هایی که یک قطعه را می‌سازند «عنصر قطعه[1]» می‌گوییم. هر عنصر قطعه می‌تواند در بر دارنده‌ی یک یا چند ثابت عددی باشد. مثلا واژه‌ی «نیم» به تنهایی شامل سه ثابت (۵، ۱۰، م)، به معنای «پنج دهم» است[2]. به دنباله‌ی ثابت‌های عددی که هر عنصر قطعه در بر دارد، فهرست مقادیر آن عنصر قطعه می‌گوییم. برای مثال قطعه‌ی، «هفت‌صد و هشتاد و شش و پنج صدم» از عناصر قطعه‌ی «هفت»، «صد»، «و»، «هشتاد و»، «شش»، «و»، «پنج»، «صدم» تشکیل شده، که با کنار هم گذاشتن فهرست مقادیر تمامی این عناصر قطعه خواهیم داشت (۷، ۱۰۰، و، ۸۰، و، ۶، و، ۵، ۱۰۰، م). حال صورت مسئله به یافتن عددی که از کنار هم قرار دادن اعضای این فهرست ساخته خواهد شد، کاهش می‌یابد.

در نظر گرفتن چند نکته به هنگام استخراج قطعه‌ها می‌تواند قطعه‌های بهینه‌تری را به دست دهد. یکی آن که هیچ‌گاه عددی با عنصر قطعه‌ی «و» شروع نمی‌شود و با آن خاتمه نمی‌یابد. دیگر آن که «م» نشان‌دهنده‌ی انتهای یک عدد است. بنابراین، می‌توان به راحتی آن را انتهای یک قطعه دانست و قطعه‌ی جدید را (در صورت وجود) از بعد از آن آغاز

---

۱ معادل فارسی عبارت انگلیسی Chunk-Element

۲ در نگاشت به جای واژه‌هایی چون «م» و «و» و «منفی» و ... می‌توان از اعداد ثابت منفی استفاده کرد. در آن صورت، فرض بر ثابت عددی بودن این واژه‌ها، فرض اشتباهی نیست.

کرد. همین‌طور، عنصر قطعه‌ی «منفی» یا «منها» همواره نشان دهنده‌ی ابتدای یک عددند، بنابراین می‌توان در صورت مشاهده‌ی آن در وسط یک قطعه، آن قطعه را از آن نقطه به دو قطعه تقسیم کرد، به طوری که «منفی» ابتدای قطعه‌ی دوم قرار بگیرد.

۱-۴-۳ استخراج اعداد صحیح

فرض کنید که قطعه‌ای از اعداد داریم که تنها دربرگیرنده‌ی اعداد صحیح است و فاقد اعداد اعشاری است. در این قسمت نشان می‌دهیم که چگونه می‌توان عدد مورد نظر را از این قطعه استخراج کرد. به این نکته توجه داشته باشید، که یک قطعه ممکن است دربرگیرنده‌ی بیش از یک عدد باشد. مثلا قطعه‌ی «هفتصد و شصت و پنج» در برگیرنده‌ی یک عدد است، در حالی که قطعه‌ی «هفت و شصت و پنج» در برگیرنده‌ی دو عدد، و قطعه‌ی «هفت و شش و پنج» در برگیرنده‌ی سه عدد است.

در ابتدا به الگوریتمی احتیاج داریم که فهرست مقادیر یک قطعه را دریافت کند و عددی را که در آن فهرست به آن اشاره شده را برگرداند، اما در صورتی که موفق به استخراج این عدد نشد، ما را راهنمایی کند که فهرست مزبور را از کجا باید به دو قسمت تقسیم کرد. مثلاً، این الگوریتم باید به ازای ورودی فهرست متناظر با «هفتصد و شصت و پنج» به ما عدد ۷۶۵ را برگرداند و به ازای ورودی «هفت و شصت و پنج» فهرست را به دو قمست «هفت» و «و شصت و پنج» تقسیم کند. فراخوانی این الگوریتم به طور بازگشتی به ازای فهرست‌های جدید ادامه خواهد یافت تا جایی که یا یک مجموعه عدد به دست بیاید، یا یک مجموعه زیرفهرست که قابل تبدیل به عدد نیستند.

اگر مثال قبل را ادامه بدهیم در انتها ۳ زیر فهرست ساخته خواهد شد که متناظرند با «هفت»، «و»، «شصت و پنج» که در آن زیرفهرست «و» نمایانگر هیچ عددی نیست، و زیرفهرست‌های «هفت» و «شصت و پنج» به ترتیب نمایانگر اعداد ۷ و ۶۵ می‌باشند. در این متن الگوریتمی را که چنین کاری برای ما انجام دهد، ExtractIntegerFromList می‌نامیم، که الگوی فراخوانی آن در شکل (۱-۵) آمده است.

```
Function ExtractIntegerFromList

inputs list_of_values: list of integers
 start_index: integer
 end_index: integer

outputs v: integer
 c: cut information
```

شکل (۱-۵) الگوی فراخوانی الگوریتم تشکیل عدد صحیح از فهرستی از اعداد

```
Function ExtractIntegers

inputs chk: Chunk

outputs s: list of integers
locals l: list of integers
 n: integer
 c: cut information
 chk1: Chunk
 chk2: Chunk

l = list of values for all chunk-elements in chk;
s = emplty list of integers

[n, c] = ExtractIntegerFromList(l);

if (c is empty) then
 s.append(n);
else if l.length == 1 and c is not empty then
 s = empty list of integers;
else
 chk1 = sub-chunk of chk before cut index in c;
 chk2 = sub-chunk of chk after cut index in c;
 s.append(ExtractIntegers(chk1));
 s.append(ExtractIntegers(chk2));
```

شکل (۱-۶) الگوریتم استخراج مجموعه‌ی اعداد صحیح داخل یک قطعه‌ی عددی

اکنون با در دست داشتن الگوریتم ExtractIntegerFromList می‌توانیم الگوریتم کلی ExtractIntegers را ارائه دهیم که در آن با استفاده از روش‌های بازگشتی توضیح داده شده استفاده می‌شود، تا یک قطعه‌ی عددی به مجموعه‌ای از اعداد صحیح که ممکن است داخل آن باشد تبدیل کند. این الگوریتم در شکل (۱-۶) ارائه شده است، که ماهیت بازگشتی آن قابل مشاهده است.

۱-۴-۴ تشخیص اعداد اعشاری با ذکر واژه‌ی «ممیز» آن

ساختار صورت نوشتاری حرفی یک عدد اعشاری که با ذکر واژه‌ی «ممیز» بیان شده باشد به شکل زیر است:

عدد صحیح - ممیز - عدد طبیعی - عدد طبیعی- م

برای مثال عدد زیر را در نظر بگیرید:

صد و هفده ممیز پنجاه ده هزارم

که در آن عدد صحیح ابتدایی «صد و هفده» و بعد از آن واژه‌ی «ممیز» ذکر شده است، و پس از آن عدد طبیعی «پنجاه» و بدون فاصله بعد از آن نیز عدد طبیعی «ده هزار» قرار گرفته است که کل عبارت با حرف «م» به پایان می‌رسد. بنابراین کافی است الگوریتمی ارائه دهیم، که الگوی فوق را در اعداد اعشاری «ممیز» دار بررسی کند و اجزای اصلی این الگو را استخراج کند. همانگونه که مشاهده می‌شود، اجزای اصلی این الگو، همگی اعداد طبیعی هستند که چگونگی استخراج آن‌ها در بخش‌های قبل توضیح داده شد. اکنون با ترکیب ساده‌ی همان الگوریتم‌ها الگوی جدید را استخراج می‌کنیم.

در بخش‌های قبل توضیح داده شد، که الگوریتم ExtractIntegers، تمامی نمونه‌های اعداد طبیعی واقع در یک قطعه‌ی واژه را استخراج می‌کند. این ویژگی الگوریتم، به ما کمک می‌کند که دو عدد طبیعی بعد از ممیز را با موفقیب استخراج کنیم. سپس باید مطمئن شویم که این اعداد طبیعی استخراج شده، شرایط زیر را دارند:

- تعداد اعداد طبیعی استخراج شده باید دقیقا دو عدد باشد.
- حرف «م» بدون فاصله بعد از عدد طبیعی دوم باشد.
- بین عدد طبیعی اول و عدد طبیعی دوم، هیچ حرف یا واژه‌ای قرار نگرفته باشد.

```
Function ExtractFloatingPart

inputs chk: Chunk
 dot_index: integer
outputs numerator: integer
 denominator: integer
locals l: list of integers
 sub_chk: Chunk

if chk does not end with "م" then
 denominator = 0;
 return;
sub_chk = sub-chunk of chk from dot_index to final "م" exclusive
l = ExtractIntegers(sub_chk);
if l.length == 2 and
 there are no chunk-elements in chk between sub_chunks for l[1]
and l[2] and
 there are no chunk-elements in chk between l[2] and final "م" then

 numerator = l[1];
 denominator = l[2];
else
 denominator = 0;
```

شکل (۱-۷) الگوریتم تشخیص و استخراج قسمت اعشاری عدد

با توجه به شرایط یاد شده در بخش اعشاری عدد، الگوریتم ExtractFloatingPart را به صورت شکل (۱-۷) ارائه می‌دهیم. توجه کنید که در این الگوریتم اندیس ممیز به صورت ورودی به الگوریتم ارسال می‌شود. همچنین، خروجی این الگوریتم یک عدد اعشاری نیست، بلکه دو عدد است که یکی صورت، و دیگر مخرج بخش اعشاری است. این خصوصیات به ما کمک خواهد کرد تا در وضعیت‌های عمومی‌تری که در آن‌ها محل ممیز به درستی مشخص نیست و تشخیص اعداد کسری از این الگوریتم استفاده کنیم.

اکنون با در دست داشتن الگوریتم استخراج قسمت اعشاری اعداد حقیقی، می‌توانیم الگوریتم تشخیص و استخراج اعداد حقیقی را که در آن‌ها واژه‌ی «ممیز» ذکر شده است تدوین کنیم. جزئیات این الگوریتم در شکل (۱-۸) مشاهده می‌شود.

```
Function ExtractRealNumberWithMomayez

inputs chk: Chunk
outputs mantissa: integer
 numerator: integer
 denominator: integer
locals chk1: Chunk
 dot_index: integer
 m: integer
 num: integer
 denom: integer

if that chk does not end with "م" then
 denominator = 0;
 return;
dot_index = find chunk-element in chk that equals "ممیز"
chk1 = sub-chunk of chk from beginning up to dot_index exclusive
mantissa = ExtractIntegers(chk1)[1];
[numerator, denominator] = ExtractFloatingPart(chk, dot_index);
if denominator != 0 then
 value = mantissa + numerator / denominator;
```

شکل (۱-۸) الگوریتم استخراج اعداد حقیقی با ذکر واژه‌ی «ممیز»

۱-۴-۵ تشخیص اعداد اعشاری در حالت کلی

در بخش قبل حالت ساده‌ای را بررسی کردیم که در آن هنگام بیان اعداد اعشاری، واژه‌ی «ممیز» صریحاً بیان شده باشد. اما در بیشتر حالات هنگام بیان اعداد اعشاری، واژه‌ی ممیز بیان نمی‌شود. این موضوع می‌تواند دو مشکل ایجاد کند. یکی آن که محل ممیز باید یافت شود و دوم آن که ممکن است محل منحصر به فردی برای ممیز وجود نداشته باشد، که در این صورت الگوریتم ما باید چند خروجی داشته باشد. مثلاً عدد «هجده و بیست و پنج صدم» تنها یک خروجی ماکسیمال دارد که آن عدد ۱۸٫۲۵ است. اما عدد «صد و بیست و پنج صدم» می‌تواند سه خروجی داشته باشد:

«۱۲۰٫۰۵» و «۱۰۰٫۲۵» و $\frac{۱۲۵}{۱۰۰}$.

آن‌چه روشن است این است که یک یا چند تا از «و» ها نقش ممیز را بازی می‌کنند که در صورت جایگزینی آن «و» با واژه‌ی «ممیز» صورت تشخیص اعداد اعشاری در حالت کلی به صورت تشخیص اعداد اعشاری با «ممیز» کاهش می‌یابد.

یک حالت اضافی ممکن است وجود داشته باشد، و آن این که عدد ورودی کسری باشد. بنابراین، برای تشخیص اعداد اعشاری در حالت کلی، فرض می‌کنیم «و» ها، ممیز هستند. با شروع از آخرین «و» و جانشینی آن با ممیز و حرکت به سمت ابتدای عدد، و اعمال الگوریتمی شبیه شکل (۱-۸) می‌توان اعداد اعشاری را در این حالت تشخیص داد. الگوریتم تشخیص اعداد اعشاری در حالت کلی، در شکل (۱-۹) ارائه شده است.

```
Function ExtractFloatingPointNumbers

inputs chk: Chunk
outputs values: list of real numbers
locals vaavs: list of integer numbers
  value: real
  mantissa: integer
  numerator: integer
  denominator: integer
  chk1: Chunk

values = empty list of real numbers;
if chk does not end with "م" then
  return;
vaavs = find all indices of "و" in chk
reverse for each dot_index in vaavs do
  chk1 = sub-chunk of chk from beginning up to dot_index exclusive
  [numerator, denominator] = ExtractFloatingPart(chk, dot_index);
  if denominator != 0 then
    mantissa = ExtractIntegers(chk1)[1];
    value = mantissa + numerator / denominator;
    values.append(value);
```

شکل (۱-۹) الگوریتم تشخیص اعداد اعشاری در حالت کلی

۱-۵ نتیجه‌گیری

در این نوشتار به ارائه‌ی الگوریتم‌های درک اعداد در متون فارسی نوشته شده به صورت حرفی یا عددی پرداخته شده است. این قابلیت یکی از بخش‌های لازم برای یک سامانه درک خودکار متون فارسی است. ابتدا به بررسی و درک عددهای نوشته شده به صورت رقمی پرداخته شد که الگوریتم‌های مربوط به آن بسیار ساده هستند. همچنین به منظور حفظ قابلیت‌های ویرایشگری سامانه پیشنهادی به بررسی چگونگی تبدیل اعداد نوشته شده به صورت رقمی به اعداد نوشته شده به صورت حرفی پرداختیم. همچنین در قسمتی به طور مفصّل به چگونگی درک و تبدیل اعداد نوشته شده به صورت حرفی پرداخته شد.

از کاربردهای چنین سامانه‌ای می‌توان، از قابلیت‌های آن در هوش مصنوعی (همچون سامانه‌های درک خودکار متون فارسی و ارتباط انسان با رایانه)، و کاربردهای آن در ویرایش متون (تبدیل اعداد یک متن از نوشتار رقمی به نوشتار حرفی و به عکس)، و کاربردهای دیگر از جمله بررسی صحّت مبالغ درج شده در فرم‌های بانکی و اداری و تطبیق اعداد وارد شده به صورت حرفی و عددی با یکدیگر نام برد.

ضمیمه دوم

مبدل تقویم

سینا ایروانیان (sina@sinairv.com)

۲-۱ مقدمه

یکی دیگر از نیازمندی‌های یک نظام درک متون فارسی، قابلیت تشخیص عبارت‌های تاریخ از تقویم‌های رایج است. یک سامانه ویراستاری متون فارسی علاوه بر تشخیص این عبارت‌ها باید بتواند عبارت‌های تاریخ از یک را تقویم به تقویم دیگر نیز تبدیل کند. در این بخش در مورد ساختار و الگوریتم‌های به کارگرفته شده در یک نظام پیشنهادی که قابلیت تشخیص و تبدیل تاریخ از تقویم‌های رایج را دارد بحث خواهیم کرد. انواع عبارات تاریخ که قابلیت تشخیص و تبدیل به یکدیگر را دارند عبارت‌اند از: تاریخ به صورت حرفی به زبان فارسی در تقویم‌های خورشیدی، میلادی، و هجری قمری؛ تاریخ به صورت حرفی به زبان انگلیسی در تقویم‌های خورشیدی، میلادی، و هجری قمری؛ و همچنین تاریخ به صورت عددی. در جدول (۲-۱) مثال‌هایی از این عبارت‌ها ارائه شده است.

جدول (۲-۱) مثال‌هایی از صورت‌های نوشتاری تاریخ‌های قابل تشخیص

شرح	مثال
تاریخ حرفی فارسی،تقویم خورشیدی	شنبه اول تیر یک هزار و سیصد و هشتاد و شش
تاریخ حرفی فارسی،تقویم میلادی	چهارشنبه پنجم اوت دو هزار و هشت
تاریخ حرفی فارسی،تقویم هجری قمری	جمعه اول رمضان هزار و چهارصد و بیست و پنج
تاریخ حرفی انگلیسی، تقویم خورشیدی	Saturday, Khordad 15th, 1380
تاریخ حرفی انگلیسی، تقویم میلادی	Sunday, July 7, 2010
تاریخ حرفی انگلیسی، تقویم هجری قمری	Friday, Ramadan 1st, 1430
تاریخ عددی	۱۰/۷/۱۳۸۸

۲-۲ تشخیص اعداد طبیعی به کمک عبارت‌های با قاعده

به منظور تشخیص عبارت‌های تاریخ که به صورت نوشتاری درج شده‌اند، باید روال مناسبی برای تشخیص و استخراج اعداد طبیعی وجود داشته باشد. اعدادی که در عبارت‌های تاریخ به کار می‌روند، صرفاً اعداد طبیعی مثبت (به صورت ترتیبی و عادی) هستند، که این امر کار را نسبت به آن‌چه در بخش تشخیص و تبدیل اعداد مشاهده شد، بسیار ساده‌تر می‌کند. البته سامانه پیش‌نهادی باید توانایی تشخیص ترکیب ارقام و حروف را داشته باشد؛ مثلاً، برخی افراد به جای «دو هزار» می‌نویسند، «۲ هزار». قواعد نگارش اعداد طبیعی به کمک عبارت‌های با قاعده قابل پیاده‌سازی است. در شکل (۲-۱) این قواعد را مشاهده می‌کنید. توجه کنید که هدف از قواعد زیر ساختن قاعده «عدد طبیعی» (آخرین قاعده) است، که چون در ساختن عبارت‌های تاریخ، نقش بازی می‌کند، زین پس با نام «عدد طبیعی» از این قاعده یاد خواهد شد.

در زمینهٔ قواعد ساختن «عدد طبیعی» در بالا، به چند نکته توجه کنید: اول، منظور از فاصله، لزوماً نویسه‌ی فاصله نیست، بلکه یک *یا چند* نویسه، که به عنوان نویسه‌های فاصله‌ی خالی از آن‌ها یاد می‌شود، مانند فاصله، tab و ... است. از آن‌جا که این قواعد در متون فارسی به کار گرفته می‌شوند، نویسه‌ی نیم‌فاصله نیز، برای فاصله در این قواعد شناخته می‌شود. نکته‌ی دیگر آن که، این قواعد به تنهایی برای بررسی صحت اعداد وارد شده کافی نیستند. برای نمونه، در قواعد فوق عبارت‌هایی چون «دو هزار و سه هزار» و «دو هزار و سه میلیون» برای عبارت‌های عددی معتبر شناخته می‌شوند. به همین دلیل، عبارت‌های ورودی که در قواعد فوق، معتبر شناخته می‌شوند، باید طی یک مرحله‌ی دیگر ارزیابی شوند. این مرحله صرفاً یک یا دو مقایسه‌ی ساده است، و از لحاظ زمان اجرا سرباری ایجاد نمی‌کند.

قواعد فوق و پیاده‌سازی آن، می‌تواند نوشتار حرفی تمامی اعداد طبیعی (از جمله صفر) به طور عادّی یا ترتیبی (مثل «هفده» و «هفدهم») و همین‌طور نوشتار رقمی آن‌ها نیز را تشخیص دهد و استخراج کند (قواعد نوشتار رقمی در قواعد مربوط به «بلوک سه رقمی» گنجانده شده است).

استخراج اعداد طبیعی متناظر با رشته‌های ورودی، به کمک درخت اشتقاق ساخته شده از روی رشته‌ی ورودی انجام می‌شود؛ به علاوه باید برگ‌های این درخت (عبارت‌های بسیط سازنده‌ی اعداد) با یک نگاشت، به مقادیر عددی متناظر آن‌ها نگاشته شود، که این کار به سادگی قابل انجام است.

سوم | اول | نه | ... | یک :: **یکان**
نود | ... | بیست :: **دهگان**
هیژده | هژده| هیجده| هیفده| شونزده | پونزده| نوزده | ... | ده :: **ده‌یکان**
نه | هشت |هفت | شیش | شش | پون | پان | چار | چهار | سی | دو | یک :: **ضریب‌صد**
دویست | صد?(**ضریب‌صد**) :: **صدگان**
?(<فاصله> و <فاصله> **دهگان**) | **ده‌یکان**)?(<فاصله> و <فاصله> **صدگان**) :: **سه‌رقمی کامل**
((ام | م) ?<فاصله>)(**یکان**
((ام | م) ?<فاصله>) (**دهگان** ?(<فاصله> و <فاصله> **صدگان**)) :: **سه‌رقمی‌تادهگان**
(?رقم ?رقم رقم) | **صدگان** | **سه‌رقمی‌تادهگان** | **سه‌رقمی کامل** :: **بلوک‌سه‌رقمی**
ترلیون | تریلیون | تریلیارد | ترلیارد | بیلیون | بلیون | میلیارد | میلیون | ملیون | هزار :: **ضرایب**
هزار | (**ضرایب** <فاصله> **بلوک‌سه‌رقمی**) :: **هزارگان**
<فاصله> و <فاصله>) *(**هزارگان** <فاصله> و <فاصله>) **هزارگان** :: **عددباهزارگان**
?(**بلوک‌سه‌رقمی**
صفرم | صفر | **بلوک‌سه‌رقمی** | **عددباهزارگان** :: **عددطبیعی**

شکل (۲-۱) قواعد تولید اعداد طبیعی

۲-۳ تشخیص عبارت‌های تاریخ به کمک عبارت‌های با قاعده

در شکل (۲-۲) قواعد عبارت‌های تاریخ در نوشتار فارسی درج شده است. باید توجه داشت که هدف از این قواعد ساختن قاعده «تاریخ» (آخرین قاعده) است. همچنین، قواعد ساختن «عدد طبیعی» نیز در قبل توضیح داده شده است. در قسمت «نام ماه» لازم است تمامی نام‌های رایج و نوشتارهای آن‌ها در نظر گرفته شوند. مثلاً در قسمت نام ماه‌های خورشیدی نام‌هایی چون «اسپند» و «امرداد» و در قسمت ماه‌های میلادی، نام‌هایی چون «آگوست»، «اگوست»، و «اوت» و در قسمت ماه‌های قمری، نام‌هایی چون «محرم الحرام»، «جمادی الاولی»، «ذو الحجه» و بسیاری دیگر، گنجانده شده است. همان گونه که در قواعد «تاریخ» مشاهده می‌شود، عبارت‌هایی که تنها از روز ماه و نام ماه تشکیل شده‌اند، برای یک عبارت تاریخ معتبر تشخیص داده می‌شوند، مثلاً «هفده فروردین»، اما چنین عبارتی، مادامی که عدد سال آن مشخص نباشد، قابل تبدیل به هیچ تقویم دیگری نیست.

۵ | ... | ۱ | پنج | ... | یک :: **شماره‌ی روز**
جمعه | شنبه ?(?[نیم‌فاصله] **شماره‌ی روز**) :: **روز هفته**
عدد طبیعی :: **روز ماه**
ذی‌الحجه | ... | محرم | دسامبر | ... | ژانویه | اسفند | ... | فروردین :: **نام ماه**
+ رقم | **عدد طبیعی** :: **عدد سال**
?(سال <فاصله>) ?(ماه <فاصله>) **نام ماه** <فاصله> **روز ماه** ?(<فاصله> **روز هفته**) :: **تاریخ**
?(**عدد سال** <فاصله>)

شکل (۲-۲) قواعد تولید عبارت تاریخ به زبان فارسی

برخی از عبارت‌های تاریخی قابل تشخیص با قاعده‌ی «تاریخ» عبارت‌اند از:

- هفده فروردین.
- پنج‌شنبه هجدهم اسفند هزار و سیصد و شصت.
- جمعه اول ربیع‌الاول سال هزار و چهارصد.
- ۴شنبه پنجم دی‌ماه سال شصت و دو.
- ۳شنبه ۷ مه ۱۹۹۰.

همچنین، در این قسمت نیز، باید عبارت‌هایی که با این قواعد معتبر شناخته شده‌اند، ارزیابی مجدد شوند. مثلاً، قواعد فوق عبارت‌هایی چون «صفرم اردیبهشت» و «چهلم فروردین» را معتبر تشخیص می‌دهند. چون این ارزیابی صرفاً از چند مقایسه‌ی عددی ساده تشکیل شده است، بنابراین بسیار سریع انجام می‌شود، و سرباری از لحاظ سرعت اجرا به برنامه تحمیل نمی‌کند.

استخراج مقادیر سازنده‌ی عبارت‌های تاریخ نیز از طریق درخت اشتقاق قابل انجام است. کافیست نگاشتی وجود داشته باشد، که برگ‌های درخت را (عبارت‌های بسیط سازنده‌ی عبارت تاریخ، از جمله نام ماه‌ها و روزهای هفته) به عدد مناسبی بنگارد. مثلاً، نام ماه به یک عدد بین ۱ و ۱۲، و روزهای هفته از «شنبه» تا «جمعه» به ترتیب به اعداد ۰ تا ۶ نگاشته شوند. روز ماه و عدد سال نیز به خودی خود عددند. همچنین، به کمک نام ماه، نوع تقویم (هجری خورشیدی، میلادی، یا هجری قمری) نیز قابل استخراج است.

۲-۴ تشخیص عبارت‌های تاریخ به انگلیسی با عبارت‌های با قاعده

در شکل (۲-۳) قواعد مربوط به عبارت‌های تاریخ به زبان انگلیسی آمده است. باید توجه داشت که هدف از قواعد زیر ساختن قاعده «تاریخ» (آخرین قاعده) است.

روز هفته :: saturday | ... | friday | sat | ... |fri | sat. | ... | fri.
روز ماه :: رقم رقم? (th | st | rd | nd)?
نام ماه :: january | ... | december | jan | ... | dec | jan. | ... | dec. |
farvardin | ... | esfand | muharram | ... | dhu al-hijjah
عدد سال :: رقم +
تاریخ :: (**روز ماه** <فاصله>? **نام ماه** <فاصله> (**روز هفته**<فاصله>? [,]? <فاصله>? **عدد سال**)? [,]?

شکل (۲-۳) قواعد تولید عبارت تاریخ به زبان انگلیسی

در قسمت «نام ماه» سعی شده است تمامی نام‌های رایج و نوشتارهای آن‌ها در نظر گرفته شود. مثلاً، در قسمت نام ماه‌های خورشیدی نام‌هایی چون "espand" و "amordād" و در قسمت ماه‌های قمری، نام‌هایی چون "muḥarram ul ḥaram"، "ramadan ul Mubarak" و بسیاری دیگر، گنجانده شده است. همچنین، در تاریخ میلادی، استفاده از اختصار بسیار رایج است. مثلاً، نوشتن تاریخ به صورت "Sat., Aug 14, 2004". به همین دلیل روزهای هفته، و نام ماه‌های میلادی به صورت مختصر نیز درج شده‌اند. همان گونه که در قواعد «تاریخ» مشاهده می‌شود، عبارت‌هایی که تنها از روز ماه و نام ماه تشکیل شده‌اند، برای یک عبارت تاریخ معتبر تشخیص داده می‌شوند، مثلاً "July 13th"، اما چنین عبارتی، مادامی که عدد سال آن مشخص نباشد، قابل تبدیل به هیچ تقویم دیگری نیست. چند مثال از عبارت‌های تاریخی قابل تشخیص با قاعده‌ی «تاریخ» عبارت‌اند از:

July 17th –
Saturday, July 17, 2004 –
Sat., Sha'abān ul Moazam 13th, 1403 –

استخراج مقادیر سازنده‌ی عبارت‌های تاریخ نیز از طریق درخت اشتقاق حاصله قابل انجام است. کافی است نگاشتی وجود داشته باشد، که برگ‌های درخت را (عبارت‌های بسیط سازنده‌ی عبارت تاریخ، از جمله نام ماه‌ها و روزهای هفته) به عدد مناسبی تبدیل کند. مثلاً

نام ماه به یک عدد بین ۱ و ۱۲، و روزهای هفته از «شنبه» تا «جمعه» به ترتیب بـه اعـداد ۰ تـا ۶ نگاشته شوند. روز ماه و عدد سال نیز به خودی خود عددند. همچنین، به کمـک نـام مـاه، نوع تقویم (هجری خورشیدی، میلادی، یا هجری قمری) نیز قابل استخراج است.

۲-۵ تشخیص عبارت‌های تاریخ به صورت عددی

نحوه‌ی تشخیص عبارت‌های تاریخ به صورت عددی بسیار ساده‌تر از دو حالت ذکر شـده‌ی دیگر است. عبارت تاریخ عددی صرفاً سه عدد است که با یـک نویسـه‌ی جداکننـده از هـم جدا شده‌اند. مثلاً، در فارسی تـاریخ عـددی بـه شـکل مقابـل مرسـوم اسـت: «۸۸/۴/۱» و در زبان‌های دیگر بـه کمـک نویسـه‌هـای دیگـری نیـز اعـداد را در عبـارت تـاریخ از هـم جـدا می‌کنند، مـثلاً «6-8-2002» یـا «6.8.2002». بـدین منظـور قواعـد تشـخیص تـاریخ عـددی بـه کمک عبارت‌های با قاعده به صورت شکل (۲-۴) طراحی شده است.

```
. | / | - :: جداکننده
( +رقم ) جداکننده ( +رقم ) جداکننده ( +رقم ) :: تاریخ
```

شکل (۲-۴) قواعد تولید عبارت تاریخ به صورت عددی

این قواعد عبارت تاریخ به صورت عددی را که در متن درج شده باشند تشخیص مـی‌دهـد، اما هر چه که تشخیص می‌دهد، لزوماً یک عبارت تاریخی معتبر نیست. به همین دلیـل، بایـد چند پس‌پردازش روی عبارت استخراج شده صورت پذیرد: اول، هـر عبـارت تـاریخی بایـد دو جداکننده‌ی یکسان داشته باشد. مثلاً «۸/۷-۲۰۰۳» با قواعد فوق معتبر شـناخته مـی‌شـود، اما یک عبارت تاریخی معتبر نیست. دوم ، الگـوی تـاریخ در مـتن بلافاصـله بعـد از عبـارت اسـتخراج شـده، و بلافاصـله قبـل از آن نبایـد ادامـه یابـد. مـثلاً، قواعـد فـوق از عبـارت «۷.۸.۹۵.۴۳» دو عبارت تاریخی استخراج می‌کنند، در صورتی که متنی که ایـن عبـارت در آن روی داده، یک آدرس آی.پی. است، نه یک عبارت تاریخی.

به کمک درخت اشتقاق قواعد فوق، هر سـه عـدد سـازنده‌ی تـاریخ، بـه سـادگی قابـل استخراج‌اند؛ اما تشخیص این‌که هر یک از این اعداد معرف روز است، یا ماه، یا سال، بسـته به مقدار عدد و موقعیت درج آن قابل حدس است. مـثلاً، عـدد مـاه نمـی‌توانـد بیشـتر از ۱۲ باشد، و عدد روز نیز نمی‌تواند بیشتر از ۳۱. نوع تقویم نیـز از طریـق مقایسـه‌ی عـدد سـال بـا تاریخ فعلی قابل حدس است. اما هیچ یک از حدس‌های فوق لزوماً صحیح نیستند.

۲-۶ تبدیل تاریخ‌ها از تقویمی به تقویمی دیگر

فرض کنید که عدد روز، ماه، و سال و نوع تقویم یک تاریخ را در اختیار دارید. این اطلاعات ممکن است که مثلاً از طریق استخراج تاریخ به کمک روش‌های استخراجی که در بالا توضیح داده شده، به دست آمده باشد. در این صورت، برای تبدیل تاریخ یاد شده از تقویمی به تقویم دیگر می‌توان از الگوریتم‌های تبدیل تقویم موجود که بدون خطا این کار را انجام می‌دهند استفاده کرد. این الگوریتم‌ها در اکثر زبان‌های برنامه‌نویسی موجودند. برای تبدیل از تاریخ خورشیدی به هجری قمری و به عکس می‌توان ابتدا تاریخ مبدأ را به تاریخ میلادی و سپس تاریخ میلادی حاصل را به تاریخ مقصد تبدیل کرد.

۲-۷ تشخیص نوع تقویم

هنگامی که با تاریخ عددی سروکار داریم، تنها می‌توانیم اجزای آن را حدس بزنیم (این که هریک از اعداد معرف روز است یا ماه، یا سال، و تاریخ درج شده متعلق به چه تقویمی است). اگر مقدار خود اعداد، نقش آن عدد را مشخص نکند، با توجه به متن و ترتیب چینش اعداد، نقش اعداد حدس زده می‌شود. مثلاً، در تقویم خورشیدی رایج است که، ابتدا روز، سپس ماه، و در آخر سال درج شود. در تقویم میلادی نیز بسته به کشور، ممکن است به ترتیب، روز، ماه و سال درج شود و یا ماه، روز، و سپس سال درج شود. بنابراین، ترتیب اعداد در متن فارسی مشخص است، و در متن انگلیسی ترتیب اعداد، ترتیب تنظیمات سامانه در نظر گرفته می‌شود.

بحث تشخیص تقویم تا حدودی متفاوت است. مثلاً، اگر در تاریخی سال عدد ۲۰۰۹ درج شده باشد، نمی‌توان گفت که به طور قطع این تاریخ متعلق به تقویم میلادی است (شاید منظور نگارنده سال ۲۰۰۹ شمسی باشد). حتی زمانی مشکل از این پیچیده‌تر می‌شود که برای درج سال تنها از دو رقم انتهایی استفاده شده باشد. مثلاً «۱/۴/۶۲» به جای «۱/۴/۱۳۶۲» یا «1-9-09» به جای «1-9-2009». از آن‌جا که حل کردن این دست مشکلات در نهایت منوط می‌شود به جویا شدن نظر کاربر؛ بنابراین روش‌هایی که برای حل این مشکلات به کارگرفته شده‌اند، در لایه‌ی رابط کاربر باید پیاده‌سازی شوند. مراحل کار بدین شرح است:

- اگر عدد سال چهار رقمی است، تفاضل عدد سال را از عدد سال جاری در تاریخ خورشیدی، میلادی و هجری قمری به دست می‌آورد. هریک از سه تفاضل که کمتر

بود تقویم مربوط به آن را برای تقویم پیشنهادی به کاربر پیشنهاد می‌دهد. اما در هر حال به کاربر این اجازه داده می‌شود که نوع تقویم پیشنهادی راتغییر دهد.

- اگر عدد سال دو رقمی است، از عدد سال دو رقمی فوق، سه عدد سال چهار رقمی ساخته می‌شود؛ یکی با شماره‌ی صده‌ی خورشیدی، دیگری با شماره‌ی صده‌ی میلادی، و سومی با شماره‌ی صده‌ی تاریخ هجری قمری زمان جاری. هر یک از سه تاریخ به دست آمده، با سال چهار رقمی زمان جاری برای تقویم متناظر آن مقایسه می‌شود هر یک که نزدیک‌تر بودند، آن تقویم به کاربر پیشنهاد می‌شود. اما در هر حال به کاربر این اجازه داده می‌شود که نوع تقویم پیشنهادی را تغییر دهد.

ضمیمه سوم

مبدل نوشتار فارسی با حروف انگلیسی به فارسی

مهرداد صنوبری وایقان (senobari@modares.ac.ir)

۳-۱ مقدمه

پیش از همه‌گیر شدن چیدمان استاندارد صفحه‌کلید فارسی، روشی به نام پینگلیش (یا فینگلیش) در بین فارسی‌زبانان به وجود آمده بود که در آن از حروف الفبای انگلیسی برای نوشتن واژه‌های فارسی استفاده می‌شد. استفاده از این روش در سال‌های اولیه‌ی ورود اینترنت به ایران بسیار رواج داشت؛ چرا که هنوز استاندارد مشخصی برای تبادل حروف و متون فارسی که اولاً مورد قبول اکثریت باشد و ثانیاً در همه‌ی سیستم‌های رایانه‌ای قابل دسترس و استفاده باشد، وجود نداشت. امروزه استاندارد یونیکد این مشکل را تا حدود بسیاری مرتفع کرده است. با وجود این، هنوز هم استفاده از روش پینگلیش میان کاربران رایج است. مهم‌ترین حوزه‌ای که امروزه استفاده از پینگلیش در آن رواج دارد، دستگاه‌های قابل حمل ارتباطی مانند تلفن همراه است. متأسفانه در اینترنت نیز برخی از کاربران به دلیل نداشتن آشنایی کامل با چیدمان فارسی صفحه کلید، متون خود را پینگلیش تایپ می‌کنند.

حجم قابل توجهی از متون برخی سایت‌ها (مانند فروم‌ها) نیز پینگلیش است. برخلاف نوشتنِ پینگلیش که به نظر ساده می‌آید، خواندن متون پینگلیش (به ویژه متنی که بیش از چند پاراگراف باشد) امری خسته‌کننده و وقت‌گیر است. با در دست داشتن ابزاری برای تبدیل متون پینگلیش به فارسی، می‌توان این نقیصه را برطرف نموده و از سوی دیگر با تشویق کاربران به استفاده از حروف فارسی در هنگام تایپ متون، به تدریج این رسم (ناپسند) را به فراموشی سپرد.

۳-۲ ساختار متون پینگلیش

بررسی متون پینگلیش نشان می‌دهد که قواعد آن ثابت و مشخص نیست. دلیل این امر نیز تا حدودی روشن است: پینگلیش را عامه‌ی کاربران فارسی‌زبان ایجاد کرده‌اند و مرکز یا مبدأ مشخصی برای تدوین قواعدِ پینگلیش‌نویسی وجود نداشته است، لذا تنوع زیادی در قواعد و الگوهای پینگلیش‌نویسی وجود دارد. برای مثال، واژه‌ی «اعتماد» در پینگلیش ممکن است به صورت‌های etemad، etemaad، e'temad، e'temaad یا eatemad نوشته شود. همان گونه که مشاهده می‌شود، گاهی برای یک واژه‌ی فارسی ممکن است بیش از چند نوع نحوه‌ی نگارش پینگلیش وجود داشته باشد. یکی از دلایل این چندگانگی، داشتن چندین حالتِ معادل انگلیسی برای برخی از حروف فارسی است. یکی دیگر از ویژگی‌های متون پینگلیش این است که شامل واژه و اصطلاحات غیر رسمی و محاوره‌ای است و واژه در این متون عموماً به صورت شکسته بیان می‌شود. دخیل کردنِ حالات و احساسات در بیان واژه (مانند بیان salam به صورت salaaaam) نیز امری متداولَ در پینگلیش به شمار می‌رود.

۳-۳ نگاشت حروف فارسی و انگلیسی

در جدول (۳-۲) حروف فارسی و معادل‌های رایج آن در پینگلیش به تفکیک آمده است. در برخی خانه‌های ستون دوم جدول (معادل‌های رایج پینگلیش) کاراکتر "$" مشاهده می‌شود؛ این کاراکتر برای نمایش کاراکترِ «خالی» مورد استفاده قرار گرفته است. برای مثال، به حرف «ع»، در واژه‌ی «اعتماد» به نگاشت ذکر شده در جدول (۳-۱) دقت کنید.

جدول (۳-۱) نگارش واژه اعتماد در پینگلیش

اعتماد	etemad
ا	E
ع	$
ت	T
ـِ	E
م	M
ا	A
د	D

جدول (۳-۲) حروف فارسی و معادل‌های آن در پینگلیش

حرف فارسی	معادل‌های رایج پینگلیش	متداول‌ترین معادل	مثال
	حروف بی‌صدا		
ب	b	-	bache
پ	p	-	pedar
ت	t	-	tatilat
ث	s, c	-	mosbat
ج	j, g	j	jahat
چ	ch	-	cheghadr
ح	h	-	mohit
خ	x, kh	kh	khanevadeh
د	d	-	dar
ذ	z	-	begzarim
ر	r	-	rah
ز	z	-	sabz
ژ	zh, j	j	mojdeh
س	s,c	s	sabz
ش	Sh	-	shabih
ص	s,c	s	saboor
ض	z	-	zaroori
ط	t	-	tathir
ظ	z	-	zaher
ع	a, a’, o, o’, e, ee, ’, $		moeen, mo’een
غ	gh, q	-	ghalat, qalat
ف	f,ph	f	farda
ق	gh, q	-	ghader
ک	k,c	k	kah
گ	g	-	gom
ل	l	-	lazem
م	m	-	mamnoon
ن	n	-	naan
ه	h, $	h	hamishe

حرف فارسی	معادل‌های رایج پینگلیش	متداول‌ترین معادل	مثال
ی	i, y, ei, ie, ee, e, iy, ey, ye, yi	y,i	yeki, ieki, yeky
	حروف صدادار		
آ	a, aa, $	a	aab
ا	a, aa, e, $	a	
ـُ(ضمه)	o, $	o	mohit
و	o, oo, ou, uo, v, u, w	o,oo, ou,v	zood, va'de, wared
ـِ(کسره)	e, $	e	mehman
ـَ(فتحه)	a, $	a	mashroot
	حروف و نشانه‌های غیر رسمی		
سی	C, 30	-	merC, mer30
سه	3	-	madre3
عت	@	-	sa@
ت	@	-	
تی	T	-	rafT
بی	B	-	charB
دی	D	-	boD hala
شما	U (در حالتی که به عنوان یک واژه به کار رود)	-	
من	I (در حالتی که به عنوان یک واژه به کار رود)	-	
اس	S	-	Smaeel
ان	N	-	Nhedam (انهدام)
ام	M	-	Mkanat (امکانات)

همان‌طور که مشاهده می‌شود، گوناگونی و تنوع بسیاری در نحوه‌ی تبدیل واژه‌های فارسی به معادل پینگلیش آن‌ها وجود دارد. در جدول فوق تنها پرکاربردترین این قواعد گردآوری شده است. با استفاده از جدول (۳-۲) و بررسی برخی روش‌های رایج در پینگلیش‌نویسی، جدول (۳-۳) را برای یافتن نگاشت از حروف انگلیسی به حروفِ فارسی ایجاد می‌کنیم.

جدول (۳-۳) نگاشت حروف انگلیسی به حروف فارسی

حرف انگلیسی	معادل محتمل	متداول‌ترین معادل
	حروف با یک معادل	
B	ب	-
D	د	-
R	ر	-
F	ف	-
L	ل	-
M	م	-
N	ن	-
V	و	-
W	و	-
Y	ی	
	حروف با چندین معادل	
‘	ع، کاراکتر نیم‌فاصله	
H	ه، ح	ه
X	خ، کس	خ
t	ت، ط	ت
T	تمامی حالات t + تی	ت
s	ث، س، ص، ش = sh	س
S	تمامی حالات s + اس	س

حرف انگلیسی	معادل محتمل	متداول‌ترین معادل
c	ث، س، ص، ک، چ = ch	س
C	تمامی حالات c + سی	س
p	ف= ph، پ = p	پ
P	تمامی حالات p + پی	پ
J	ژ، ج	ژ
G	ج، ق = gh، غ = gh، گ	ج
Z	ز، ذ، ض، ظ، ژ = zh	ز
A	ا، آ، أ، عا، اً = an، ـَ(فتحه) ع، ی، ه، وا، آ = aa	
e	ـِ(کسره)، ی، ا، ه = eh، ع = ee، ی = ei، ی = ee، ی = ey، اع	ـِ(کسره) ی
I	ی، ای، ی = ie، ی = iy	ی
K	ک، خ = kh	ک
O	و او ـُ(ضمه) و = oo و = ou	و
U	و او و شما	و
Q	ق غ	ی

در جدول فوق بوضوح مشاهده می‌شود که فقط تعداد بسیار کمی از حروف انگلیسی، تنها یک معادل در فارسی دارند و بسیاری از حروف انگلیسی دو یا چند معادل در فارسی دارند. همین امر مشکلات بسیاری را در توسعه‌ی یک مبدل پینگلیش ایجاد می‌کند.

۳-۳-۲ تکرار حروف

در متون پینگلیش، تکرار حروف معمولاً به دو دلیل رخ می‌دهد:

- به منظور بیان حروفی مانند «و»، برای مثال: mushroom
- به منظور بیان احساسات و ابراز هیجان: mercccccc، salaaaaaaam

علاوه بر دلایل ذکر شده، ممکن است برای بیان «تشدید» نیز از تکرار حروف استفاده شود، اما بررسی متون پینگلیش نشان می‌دهد که کاربران تقریباً در اکثر موارد، تشدید را در نگارشِ واژه دخالت نمی‌دهند. برای مثال، املای واژه‌ی مفرح به صورت "mofarah" رایج‌تر از "mofarrah" است.

۳-۳-۳ استفاده از واژه شکسته

کاربرد متون پینگلیش ایجاب می‌کند که این متون عمدتاً به صورت محاوره‌ای نوشته شده و واژه در آن به صورت شکسته به کار رود. برای مثال، واژه‌ی «خانه» معمولاً به صورت محاوره‌ایِ آن یعنی "khoone" نوشته می‌شود و نه "khane".

۳-۳-۴ استفاده از واژگان انگلیسی

الگوی دیگری که در متون پینگلیش مشاهده می‌شود، استفاده از برخی واژه انگلیسی است. این واژه عمدتاً در دسته‌ی واژه فنی قرار می‌گیرند و رایج‌ترین حوزه‌ی آن‌ها، حوزه‌ی واژه تخصصیِ رایانه و اینترنت است. برای نمونه، برخی از این واژه در جدول (۳-۴) آورده شده است. (ذکر این نکته ضروری است که این جدول حاوی تمامی این واژه‌ها نبوده و فقط تعداد محدودی از پرکاربردترین آن‌ها به عنوان نمونه آورده شده است).

جدول (۳-۴) برخی از واژه‌های انگلیسی رایج در متون پینگلیش

معادل فارسی	واژه‌ای انگلیسی
کپی	Copy
سی‌دی	Cd
سایت	Site
صفحه	Page
حساب کاربری	Account
آی‌پی	IP
تایپ	Type
چک	Check
آزاد، خالی	Free
دایال‌آپ	Dialup
رایانه	Computer
عنوان	Title
سی‌پی‌یو، پردازنده	CPU
حافظه	Memory
پروژه	Project
ایمیل	mail, email
الصاق کردن	Paste
ابزار	Tool
سلام	Hi
خداحافظ	Bye
خوب	Good
باشه، بسیارخب،	Ok
آدرس	Address
تلویزیون	Tv
صبر کن	Wait

۳-۳-۵ استفاده از حروف و کاراکترهای ویژه در واژه

از این الگو عمدتاً برای کمتر کردن طول واژه‌ها و خلاصه‌سازی استفاده می‌شود. جدول (۳-۵) پرکاربردترینِ آن‌ها را شامل می‌شود.

جدول (۳-۵) کاراکترهای ویژه در واژه‌های پینگلیشی

حرف – کاراکتر انگلیسی	معادل	مثال
0	o	z00d -> zood
1	ye, yek	1doone -> yedoone
2	too, doo, do	be2ni -> betooni, bedooni 2a -> doa
3	se	3tar -> setar
4	chahar, char, cha, for	4kerim -> chakerim
8	eit	w8-> weit (صبر کن)
@	at	sa@ -> saat
30	si, ci	mer30 -> merci
T	ti	rafT -> rafti
C	si, ci	merC -> merci

۳-۴ مبدل پینگلیش

متأسفانه گزارش‌های دقیقی در مورد ساختار و نحوه عملکرد مبدل‌های پینگلیش در دست نیست. اما در بررسی برنامه‌های متداول تبدیل خودکار متون پینگلیش به فارسی، نکته‌ای که جلب توجه می‌کند این است که روش‌های مورد استفاده در آن‌ها نمی‌توانند همه‌ی الگوهای جمع‌آوری شده در بخش پیشین را پوشش دهد. البته با توجه به گستردگی روش‌های پینگلیش‌نویسی و مواردی که در بخش پیشین آمد، چنین نتیجه‌ای قابل پیش‌بینی است.

برخی برنامه‌ها واژه‌ی پینگلیش را فقط در صورتی به درستی تبدیل می‌کنند که در نوشتن آن، قاعده خاصی رعایت شده باشد (برای مثال، حرف «ع» به صورت «'» نوشته شده باشد، یا حرف «ا» به صورت «aa» در واژه‌ی e'temaad). بررسی‌ها نشان می‌دهد ملزم نمودنِ کاربران به استفاده از یک یا چند الگوی نوشتاری در پینگلیش، با توجه به تعدد الگوهایَ

رایج میان کاربران، موفقیت چندانی در پی ندارد. با در نظر گرفتن موارد فوق، در ادامه‌ی این بخش به ارائه یک روش کلی برای تبدیل متون پینگلیش می‌پردازیم. ایده‌ی اصلی این روش، یادگیری و کشف الگوهای تبدیل با نرم‌افزار است. این روش دو مرحله دارد:

- **کشف و یادگیری الگوهای تبدیل، از طریق تعدادی واژه‌ی نمونه:** در این مرحله، تعدادی واژه‌ی پینگلیش و معادل آن برای ورودی به نرم‌افزار داده می‌شود. نرم‌افزار طبق مراحلی که در ادامه می‌آید، معادل‌های حروف انگلیسی را شناسایی و ذخیره می‌کند.
- **استفاده از الگوهای شناسایی شده در مرحله‌ی قبل:** در این مرحله (استفاده کاربر از نرم‌افزار)، نرم‌افزار واژه پینگلیش را برای ورودی دریافت و با استفاده از الگوهای کشف شده در مرحله‌ی قبل، معادل‌های محتمل را برای آن واژه‌ها را ارائه می‌کند.

یکی از مزیت‌های این روش اضافه کردن الگوهای جدید در هر لحظه به نرم‌افزار است و به بازنویسی نرم‌افزار به منظور ارائه‌ی معادل‌های درست برای آن الگو نیاز نیست. در ادامه، توضیح هر یک از مراحل فوق می‌آید.

۳-۴-۱ کشف و یادگیری الگوهای تبدیل

در این مرحله، تعدادی واژه‌ی پینگلیش و نحوه‌ی نگارش آن‌ها به فارسی به نرم‌افزار داده می‌شود. برای مثال ورودی‌هایی که در ادامه آمده‌اند را در نظر بگیرید:

مثال ۱. واژه "no"

ن <-- n
و <-- o

مثال ۲. واژه "amadegi"

آ <-- a
م <-- m
ا <-- a
د <-- d
ِ <-- e
گ <-- g
ی <-- i

مثال ۳. واژه "cheshme"

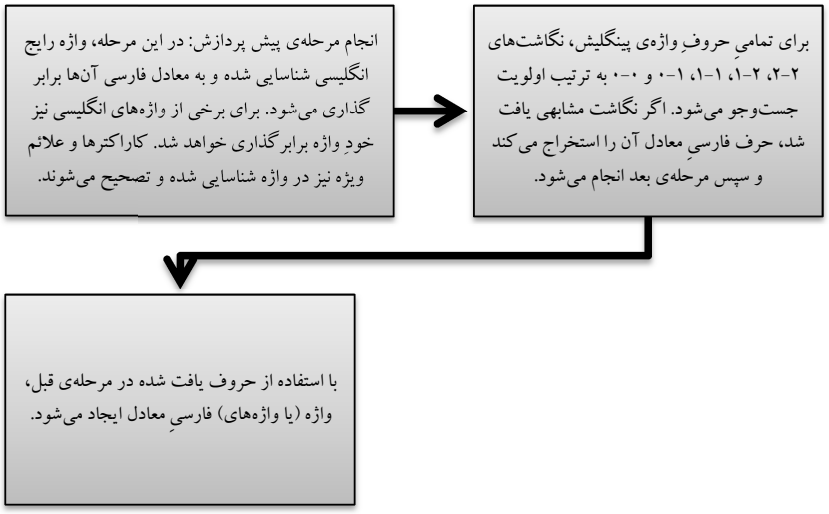
چ <-- c
--> h
ِ <-- e
ش <-- s
--> h
م <-- m
ه <-- e

با در دست داشتن نگاشت‌هایی مانند مثال‌های فوق، نرم افزار سعی می‌کند الگوهای نگاشت را کشف کند. بدین ترتیب که برای تک‌تک حرف در واژه‌ی پینگلیش، نگاشت‌های زیر را شناسایی و ذخیره می‌کند:

- **نگاشت ۲-۲:** ۲ حرف قبلی+حرف مورد نظر+۲ حرف بعدی حرف فارسیِ معادل
  مثال: نگاشت ۲-۲ برای حرف e در واژه‌ی ch**e**shme برابر است با {ch**e**sh ← کسره}
- **نگاشت ۲-۱:** ۱ حرف قبلی+حرف مورد نظر+۲ حرف بعدی←حرف فارسیِ معادل
- **نگاشت ۱-۱:** ۱ حرف قبلی+حرف مورد نظر+۱ حرف بعدی←حرف فارسیِ معادل
- **نگاشت ۱-۰:** حرف مورد نظر+۱ حرف بعدی ← حرف فارسیِ معادل
  مثال: نگاشت ۱-۰ برای حرف c در واژه‌ی **c**heshme برابر است با {**c** h ← چ}
- **نگاشت ۰-۰:** حرف مورد نظر ← حرف فارسیِ معادل

۳-۴-۲ استفاده از الگوهای شناسایی شده

در این مرحله، نرم‌افزار از نگاشت‌های ذخیره شده در مرحله‌ی قبل استفاده می‌کند تا معادل فارسیِ واژه جدید را تولید کند. برای این منظور، از الگوریتم شکل (۳-۱) و شکل (۳-۲) استفاده می‌شود.



شکل (۳-۱) الگوریتم تولید معادل‌های فارسی

شکل (۳-۲) الگوریتم کلی تبدیل واژه‌های پینگلیش

۳-۴-۳ الگوریتم کلی تبدیل واژه پینگلیش

پس از تولید معادل‌های فارسی، باید پیشنهادها را طوری مرتب کرد که معادل‌های محتمل‌تر، جایگاه بالاتری در فهرست داشته باشند. برای دستیابی به این هدف می‌توان از روش‌های رتبه‌بندی (بخش۴-۴-۳) مانند روش بسامد واژه‌ها استفاده کرد. با توجه به این که واژه‌های محاوره‌ای کاربرد فراوانی در پینگلیش‌نویسی دارند، برای استفاده از روش بسامد واژه‌ها باید یک پیکره حاوی هر دو نوع واژه‌ها محاوره‌ای و معیار تهیه کرد.

۳-۵ نتیجه‌گیری

دقت روش ارائه شده، ارتباط مستقیمی با مجموعه داده‌های اولیه‌ی آن دارد. هرچه تعداد واژه‌های پینگلیش و تنوعِ نگارشیِ آن‌ها در مجموعه داده‌های اولیه بیشتر باشد، دقت الگوریتم در ارائه‌ی درستِ معادل برای واژه‌های جدید، بیشتر خواهد بود.

به همین علت، برای تولید مجموعه داده‌های اولیه ابزار جداگانه‌ای مورد نیاز است. چنین ابزاری باید یک متن پینگلیش را برای ورودی دریافت و به ازای تک‌تک واژه‌های موجود در آن، همه‌ی معادل‌های فارسی آن واژه را تولید کند و به کاربر نمایش دهد (ایجاد تمامیِ معادل‌های فارسیِ یک واژه‌ی پینگلیش، با استفاده از جدول (۳-۳) صورت می‌گیرد.

بدیهی است که تعداد این معادل‌ها در برخی حالات ممکن است به چند هزار معادل برسد). کاربر از میان معادل‌های نمایش داده شده، واژه‌های صحیح را انتخاب کرده و نرم‌افزار به صورت خودکار نگاشت‌های آن واژه را یافته و ذخیره می‌کند. با کمک این ابزار به راحتی می‌توان یک مجموعه داده‌ی غنی از واژه‌ها و نگاشت‌های آن‌ها ایجاد کرد و دقت نرم‌افزار را افزایش داد.

البته دستیابی به دقت صددرصد نیز ممکن نیست، برای نمونه، می‌توان واژه‌هایی را مثال آورد که بیش از دو معادل فارسی دارند و تعیین معادل صحیحِ آن‌ها، جز با پردازش معناییِ متن امکانپذیر نیست. "madar" و "dar" از جمله‌ی این واژه هستند که اولی ممکن است «مادر» یا «مدار» و دومی «دار» یا «در» باشد و در هر دو حالت، نرم‌افزار نمی‌تواند معادل حقیقی را انتخاب کند و انتخاب نهایی به عهده کاربر خواهد بود.

ضمیمه چهارم

اصلاح علائم نشانه‌گذاری

کامیار کنعانی (kanani@ce.sharif.edu)

۴-۱ مقدمه

منظور از خطاهای نشانه‌گذاری، خطاهایی است که در اثر کاربرد نادرست یا نبود علائم نگارشی مانند نقطه، ویرگول و علامت سوال، ایجاد می گردد. چگونگی رعایت قواعد نشانه‌گذاری برای انسان و رایانه با یکدیگر تفاوت اساسی دارد. یک شخص قواعد نشانه‌گذاری را با فهم معنی جملات و تشخیص نحو واژه به راحتی رعایت می‌کند؛ اما تشخیص معنی و نحو واژه برای رایانه به سادگی امکان‌پذیر نیست. بدون تشخیص نحوی و معنایی، تشخیص و اصلاح خطاهای نشانه‌گذاری محدود می‌شود. در اینجا قواعد نگارشی را با توجه به مطلب فوق در دو دسته زیر جای می‌دهیم:

- قواعدی که رعایت آنها نیازمند دانستن معنا و نحو هستند.
- قواعدی که بدون دانستن معنا و نحو می‌توان آنها را رعایت کرد.

در دسته اول، اشکالات وابسته به ساختار نحوی یا معنای جمله هستند. در این دسته توجه ما معطوف است به لزوم یا نبود نماد به کار رفته؛ یعنی با توجه به معنای جمله و یا ساختار نحوی بررسی می‌شود که آیا کاربرد یک نماد لازم است یا خیر. برای تشخیص این دسته از اشکالات نیازمند تحلیل نحوی جمله و به دست آوردن نقش واژه‌ها در جمله و تا حدی نیازمند درک معنایی از جمله هستیم. در ادامه برخی از اشکالاتی که در این دسته طبقه‌بندی می‌شوند ذکر شده‌اند.

- تشخیص پایان جمله و پیشنهاد علامت (نقطه، علامت سوال، نقطه ویرگول، تعجب).
- تشخیص محل‌های مناسب در جملات طولانی و قرار دادن ویرگول در محل مناسب.
- تشخیص قرار دادن علامت نقل قول.

در قواعد غیر وابسته به معنا، چگونگی قرارگیری نمادها در کنار یکدیگر و در میان واژه‌ها توجه می‌شود. در اینجا ما کاری با لزوم یا عدم لزوم نماد به کار رفته نداریم، بلکه فرض ما بر این است که نماد در جایگاه درست خود به کار رفته است و ما فقط با چگونگی قرار گرفتن آن کار داریم. از جمله اشکالاتی که در دسته اول جای می‌گیرند می‌توان از موارد زیر نام برد:

- نبود هماهنگی بین نمادهای دوگانه: پرانتز، گیومه، براکت.
- فاصله‌های بی‌مورد بین نمادها و واژه‌ها.
- ترکیب نادرست نمادها: مثلا گذاشتن چهار نقطه به جای سه نقطه.

۴-۲ روش تشخیص خطاهای نگارشی

در اینجا روشی ارایه می‌شود که می‌تواند اشکالات دسته دوم را به طور کامل تشخیص و اصلاحات لازم را پیشنهاد دهد. همچنین، تا محدوده کوچکی از اشکالات دسته دوم را نیز پوشش دهد. این روش بر اساس تشخصیص الگوی زبان منظم کار می‌کند؛ به این معنا که ابتدا الگوهای خطا به الگوریتم داده می‌شوند و سپس الگوریتم بر اساس این الگوها، خطاها را شناسایی می‌کند و شکل صحیح را پیشنهاد می‌دهد. نقطه قوت این روش در طراحی نرم‌افزار نمود پیدا می‌کند؛ به طوری که به سادگی می‌توان الگوها را به صورت دینامیک به آن افزود یا از آن حذف کرد و نیازی به کامپایل کردن مجدد کد وجود ندارد. این قابلیت تولیدکنندگان را قادر می‌سازد تا با جایگزینی فایل، الگوی جدید نرم‌افزار را به‌روزرسانی کنند.

نکته اصلی در انتخاب الگوهای خطا برای پایگاه داده، تواتر خطا است. یعنی هرچه تعداد رخداد خطا بیشتر باشد، لزوم قرار گرفتن آن در پایگاه داده بیشتر است. برای اینکه نرم‌افزار بتواند هرچه بیشتر خطاها را شناسایی کند، باید تعداد الگوهای خطا حداکثر باشد. در عمل، به خاطر محدودیت‌های سیستمی و انسانی، امکان تعریف تمام الگوهای خطا وجود ندارد، لذا ناگذیر به اتخاذ ساز و کاری برای انتخاب زیرمجموعه‌ای از الگوها هستیم که بیشترین کارایی را داشته باشند. در واقع، معدودی از خطاهای نشانه‌گذاری در عمل بسیار اتفاق می‌افتند که و می‌توانند درصد بالایی از خطاها را پوشش دهند، در نتیجه، شناسایی و انتخاب آنها دقت نرم‌افزار را بیشتر می‌کند.

۴-۳ الگوریتم

در این بخش الگوریتم خطایاب نشانه‌گذاری و مشکلات آن را توضیح می‌دهیم. در شکل (۴-۱) الگوریتم خطایاب نشان داده شده است. در شکل (۴-۱) مجموعه تمام الگوها با R نمایش داده شده است. در این مجموعه الگوهای زبان منظم به صورت NFA ذخیره شده‌اند. هر کدام از این الگوها بیانگر یک اشتباه نشانه‌گذاری هستند که ممکن است توسط کاربر اتفاق بیفتند. در خط ۸ متن ورودی را به ازای تمام الگوها بررسی می‌کنیم. تابع *Match()* بررسی می‌کند که آیا NFAی *r* در متن ورودی *text* تطبیق می‌کند یا خیر. الگویی که کوچک‌ترین اندیس را داشته باشد با عنوان اولین خطای تشخیص داده شده باز می‌گردد. همچنین، الگوی اصلاحی آن نیز در خروجی بازگردانده می‌شود.

```
1:   R = {Patterns of punctuational error}
2:   FindPunctuationMistake (text)
3:   {
4:       MachedCase.FirstMatchIndex = 0
5:       MachedCase.FirstMatchLength = 0
6:       MachedCase.CorrectForm = ∅
7:       Foreach r ∈ R do
8:           (Match_Index, Match_Length) = Match (text, r)
9:           If ((Match Found) and Match_Index < FirstMatchIndex) then
10:              MachedCase.FirstMatchIndex = r.Match_Index
11:              MachedCase.FirstMatchLength = r.Match_Length
12:              MachedCase.CorrectForm = Corrected (r.Index)
13:      Return MachedCase
14:  }
```

شکل (۴-۱) الگوریتم خطایاب نشانه‌گذاری

اشکال‌زدایی یک متن، با فرستادن پاراگراف‌های آن به الگوریتم خطایاب نشانه‌گذاری انجام می‌شود. در واقع واحدهای ورودی به خطایاب، متن‌های ساده پاراگراف‌ها هستند که پس از پیرایش و حذف کاراکترهای کنترلی به صورت خام به خطایاب داده می‌شوند.

سپس، خطایاب متن ورودی را برای یافتن الگوها جستجو می‌کند. الگوها یکی یکی در متن جستجو می‌شوند و هر جا که تطبیقی رخ دهد مشخص می‌شود. یک استثنا در مورد علامت‌های دوتایی (مثل گیومه و پرانتز) وجود دارد. الگوهای مربوط به علامت‌های دوتایی به دلیل اینکه الگوی زبان منظم را ندارند با توسط الگوریتم گفته شده قابل شناسایی نیستند. در این مورد به جای استفاده از الگو، از الگوریتم‌های کلاسیک مربوط به ارزیابی عبارت‌های جبری استفاده می‌کنیم. در یافتن خطاها، ممکن است چندین تطابق (با چندین الگو) رخ بدهد. در این حالت خطایاب آن خطایی را که از نظر موقعیت مکانی از همه جلوتر است برمی‌گرداند. یعنی در هر بار روال خطایاب فقط یک خطا را باز می‌گرداند و به همراه هر خطا پیشنهادی برای اصلاح آن ارائه می‌دهد که بسته به نظر کاربر می‌توان آن را اعمال کرد یا نادیده گرفت. می‌توان با فراخوانی‌های متعدد روال خطایاب، به صورت خودکار، کل متن را اصلاح نمود.

۴-۴ تعریف الگو و ملاحظات مربوط به آن

همانطور که گفته شد تئوریبه لحاظ نظری می‌توان تعداد بسیاری الگو تعریف کرد، ولی الگوهایی از نظر ما اهمیت دارند که در دنیای واقعی بسیار رخ می‌دهند. تسلط برنامه نویس بر استفاده از عبارت‌های منظم و همچنین بر زبان و ادبیات فارسی، از فاکتورهای مهم در کیفیت الگوهای تعریف شده است. مثال‌هایی که در ادامه این بخش ارائه می‌شود بر اساس نمادگذاری عبارت‌های منظم در کتابخانه Net. است.

۴-۴-۱ ساختار الگوها

الگوها از طریق یک فایل ورودی به برنامه داده می‌شود. هر رکورد در این فایل شامل ۴ فیلد است که فیلدهای هر رکورد به صورت زیر تعریف می‌شوند:

- فیلد اول: الگوی خطا به صورت عبارت منظم.
- فیلد دوم: توضیحی در مورد اشکال پیدا شده.
- فیلد سوم: توضیحی برای رفع اشکال پیدا شده.
- فیلد چهارم: الگوی اصلاحی برای رفع اشکال (با استفاده از زیرالگوها).

برای مثال، شکل (۴-۲) به ترتیب فیلدهای یک رکورد را نشان می‌دهد.

```
([\»\]])[،]
ویرگول قبل از علامت بسته قرار نمی‌گیرد
ویرگول را به بیرون انتقال دهید
$1،
```

شکل (۴-۲) نمونه‌ای از فیلدهای یک رکورد

در سطر اول از شکل (۴-۲) تعریف الگو قرار گرفته است. بخشی از الگو که با پرانتز توپر مشخص شده است برای زیرالگو با نام $1 مشخص می‌شود که در سطر چهارم در الگوی اصلاحی دیده می‌شود.

۴-۴-۲ هم‌پوشانی

هم‌پوشانی دو الگو وقتی ایجاد می‌شود که بازه‌های آنها روی یکدیگر قرار گیرند. در این صورت، الگویی که اندیس شروع آن جلوتر است بازگردانده می‌شود. هم‌پوشانی موقعی مشکل‌ساز می‌شود که اندیس شروع بازه‌ها یکسان باشد. در این حالت باید یکی از آنها برای خطای تشخیص داده شده باز گردانده شود. در این حالت، حق تقدم با الگویی است که در فایل ورودی، جلوتر از دیگری تعریف شده است. در واقع ترتیب تعریف الگوها در فایل ورودی اهمیت دارد.

۴-۴-۳ الگوهای زیر مجموعه

تعریف الگوها نیازمند داشتن تسلط بر عبارت‌های منظم است. در تعریف الگوها باید حالت‌های زیرمجموعه‌ای از الگوها را در نظر گرفت. نباید پایگاه داده را با الگوهایی که زیرمجموعه الگوهای دیگر هستند انباشت. این کار باید در صورت لزوم و مراقبت از حالت هم‌پوشانی و نیز الگوهای اصلاحی آن‌ها انجام پذیرد.

۴-۴-۴ دور

فرض کنید دو الگو داریم: الف و ب. وقتی خطای الف تشخیص داده می‌شود و اصلاح می‌شود، منجر به ایجاد خطای ب می‌شود و وقتی خطای ب اصلاح می‌شود منجر به خطای الف می‌شود؛ به این حالت دور می‌گوییم. برای مثال به دو الگوی ارائه شده در شکل (۴-۳) و شکل (۴-۴) توجه کنید.

```
(»)([^\b])
بعد از علامت بسته فاصله یا نیم‌فاصله لازم است
فاصله بگذارید
$1 $2
---------------------------------------------
[ \b]+(،)
قبل از ویرگول فاصله نمی‌آید
فاصله را حذف کنید
$1
```

شکل (۴-۳) یک نمونه از الگوی دارای دور

در مثال فوق پس از اصلاح الگوی اول، الگوی دوم ایجاد می‌شود و پس از اصلاح الگوی دوم، الگوی اول؛ یعنی در یک دور قرار می‌گیرد. باید الگوی اول به صورت شکل زیر اصلاح شود.

```
(»)([^\b!؛،؟.\)\]\n\u0002])
بعد از علامت بسته فاصله یا نیم‌فاصله لازم است
فاصله بگذارید
$1 $2
```

شکل (۴-۴) الگوی دارای دورِ اصلاح شده

در الگوی بالا تمام علائمی که مجاز هستند بدون فاصله بعد از گیومه بسته قرار گیرند، در پرانتز دوم مشخص شده‌اند. دور یکی از مشکلات متداولی است که در هنگام تعریف الگوها رخ می‌دهد. روال مشخصی برای رفع حالات دور وجود ندارد. مهارت برنامه نویس در تهیه الگوها و تست کامل و جامع برنامه می‌تواند تا حد زیادی این مشکل را مرتفع کند.

۴-۵ عبارت منظم

پیاده‌سازی خطایاب نشانه‌گذاری با استفاده از عبارت منظم انجام شده است. عبارت منظم وسیله‌ای را برای برنامه نویسان فراهم می‌کند که به کمک آن‌ها بتوانند عبارت‌های، واژه‌ها و الگوهای مورد نظر را در یک رشته متنی پیدا کنند. این امکان در کتابخانه‌های اکثر زبان‌های برنامه‌نویسی وجود دارد. عبارت‌های منظم در بستر Net. نسبتا پیشرفته‌تر از سایر زبان‌های برنامه‌سازی هستند و امکانات جالبی را ارائه می‌دهند.

از عبارت‌های منظم می‌توان برای تشخیص اشکالات نشانه‌گذاری استفاده کرد. اکثر «خطاهای غیر وابسته به معنا» را به راحتی می‌توان عبارت‌های منظم شناسایی کرد، در مقابل «قواعد وابسته به معنا» نیازمند استفاده از روش‌های پیشرفته هستند اما برخی از آن‌ها را می‌توان با روش عبارت منظم شناسایی کرد. الگوهایی که با عبارت‌های منظم C# توصیف می‌شوند، برای کاربرد ما مناسب هستند و می‌توان به وسیله آن‌ها طیف وسیعی از اشکالات نگارشی را شناسایی کرد. در ادامه مطالبی را که برای نوشتن عبارت‌های منظم در C# لازم هستند به طور خلاصه بیان می‌کنیم.

۴-۵-۱ نمادها در عبارت‌های منظم

در ادامه به طور خلاصه علائم پراستفاده در عبارت منظم را (برای مراجعه سریع) نشان داده‌ایم. مطالعه آن‌ها برای فهم مثال‌های ارائه شده در این بخش مفید است.

جدول (۴-۱) علائم تکرار و تعداد تکرار در عبارت‌های منظم

نشانه	توضیح
*	Repeat any number of times
+	Repeat one or more times
?	Repeat zero or one time
{n}	Repeat n times
{n, m}	Repeat at least n, but no more than m times
{n,}	Repeat at least n times

جدول (۴-۲) علایم جانشینی حروف و کاراکترها در عبارت‌های منظم

عبارت	توضیح
.	Match any character except newline
\w	Match any alphanumeric character
\s	Match any whitespace character
\d	Match any digit
\b	Match the beginning or end of a word
^	Match the beginning of the string
$	Match the end of the string
[x]	Match any character specified by x

جدول (۴-۳) علائم جانشین معکوس حروف و کاراکترها در عبارت‌های منظم

عبارت	توضیح
\W	Match any character that is NOT alphanumeric
\S	Match any character that is NOT whitespace
\D	Match any character that is NOT a digit
\B	Match a position that is NOT the beginning or end of a word
[^x]	Match any character that is NOT x
[^aeiou]	Match any character that is NOT one of the characters aeiou

برای مثال، به عبارت‌های مشخص شده در جدول زیر توجه کنید.

جدول (۴-۴) نمونه‌ای از عبارت‌های منظم

عبارت	توضیح
\b\d{3}-\d{4}\b	سه رقم، خط فاصله و سپس چهار رقم
\b,	فاصله قبل از ویرگول
[a-zA-Z]+[0-9]+	رشته حرفی منتهی به رشته رقمی

۴-۵-۲ گروه‌بندی یا استخراج زیرالگو

از جمله امکاناتی که Net. ارائه می‌کند، امکان استخراج زیرالگو از یک الگوی منظم است. این امکان کمک می‌کند تا بتوانیم الگوی جانشین (اصلاحی) را برای یک الگوی خطا به راحتی بسازیم. هر بخشی از عبارت منظم که داخل پرانتز قرار گیرد، می‌تواند از شیء آن استخراج شود. اگر چندین پرانتز داشته باشیم هر پرانتز با یک اندیس متناظر می‌شود که می‌توان به آن ارجاع داد. به جای اندیس می‌توان آن‌ها را نام‌گذاری کرد (با قرار دادن نام در داخل علامتهای < و >). در جدول (۴-۵) عبارت‌های گروه‌بندی نشان داده شده‌اند.

جدول (۴-۵) عبارت‌های گروه‌بندی

عبارت	توضیح
(exp)	Match exp and capture it in an automatically numbered group
(?<name>exp)	Match exp and capture it in a group named name
(?:exp)	Match exp, but do not capture it

برای جانشینی زیرالگوهای استخراج شده در الگوی جانشین از شماره‌های آن‌ها به همراه علامت $ استفاده می‌کنیم. مثلا $1 برای پرانتز اول و $2 برای پرانتز دوم و غیره.

ضمیمه پنجم

واژه‌نامه‌ی انگلیسی به فارسی

معادل فارسی	واژه‌ی انگلیسی
تجزیه وابستگی	Dependency Parse
اشتقاق	Derivation
حرف تعریف	Determiner
گفتمان	Discourse
فاصله‌ی ویرایشی	Edit Distance
فاصله‌ی اقلیدسی	Euclidean Distance
نادرست	False
مثبت نادرست	False Positive
تازنی	Folding
هم‌آوا	Homophone
هم‌شکل	Homoshape
ترکیبی	Hybrid
میانوند	Infix
تصریف	Inflection
حرف صوت	Interjection
چیدمان	Layout
ریشه	Lemma

معادل فارسی	واژه‌ی انگلیسی
تطبیقی	Adaptive
صفت	Adjective
قید	Adverb
وند	Affix
شاخص	Cardinal
فاصله‌ی میان نویسه‌ها	Character Distance
قطعه	Chunk
ترکیب	Composition
زبان‌شناسی محاسباتی	Computational Linguistics
تصریف فعلی	Conjugation
حرف ربط	Conjunction
بافت	Context
مبتنی بر بافت	Contextual
هم‌نشینی، هم‌آیی	Co-occurrence
احراز متقاطع	Cross-validation
ساختار داده	Data Structure
تصریف اسمی	Declension

معادل فارسی	واژه‌ی انگلیسی
حرف اضافه	Preposition
احتمالی	Probabilistic
ضمیر	Pronoun
فراخوانی	Recall
اتصال رکوردها	Record Linkage
زمان اجرا	Run Time
معناشناسی	Semantics
معنی‌دار	Significant
کلید مشابهت	Similarity Key
خطاهای املایی تکی	Single-error
پیچیدگی فضایی	Spatial Complexity
آماری	Statistical
فاصله‌ی رشته‌ای	String Distance
پسوند	Suffix
نحو	Syntax
سه‌برگچه‌ای	Ternary
پیچیدگی زمانی	Time Complexity
درخت	Tree
منفی حقیقی	True Negative
حروف‌چینی	Typographical
نویسه‌ی ترکیبی	Upper Character

معادل فارسی	واژه‌ی انگلیسی
ریشه‌یابی	Lemmatization
فاصله‌ی حروف	Letter Distance
لغوی	Lexical
یادگیری ماشین	Machine Learning
دقت میانگین	Mean Average Precision (MAP)
رتبه‌ی وارانه‌ی میانگین	Mean Reciprocal Rank (MRR)
واژک‌شناسی	Morphology
واج‌شناسیِ تکواژها	Morphophonology
خطاهای املایی چندگانه	Multi-error
موجودیت اسمی	Name Entity
پردازش زبان طبیعی	Natural Language Processing (NLP)
چند–وزنی	N-Gram
گره	Node
بازشناسی نوریِ نویسه‌ها	Optical Character Recognition (OCR)
نگارشی	Orthographical
اداتِ سخن	Part of Speech
آواشناسی	Phonetics
واج‌شناسی	Phonology
کاربردشناسی	Pragmatics
دقت	Precision
پیشوند	Prefix

ضمیمه ششم

واژه‌نامه‌ی فارسی به انگلیسی

معادل فارسی	واژه‌ی انگلیسی
اتصال رکوردها	Record Linkage
احتمالی	Probabilistic
احراز متقاطع	Cross-validation
اداتِ سخن	Part of Speech
اشتقاق	Derivation
آماری	Statistical
آواشناسی	Phonetics
بازشناسی نوریِ نویسه‌ها	Optical Character Recognition (OCR)
بافت	Context
پردازش زبان طبیعی	Natural Language Processing (NLP)
پسوند	Suffix
پیچیدگی زمانی	Time Complexity
پیچیدگی فضایی	Spatial Complexity
تجزیه وابستگی	Dependency Parse
پیشوند	Prefix
تازَنی	Folding

معادل فارسی	واژه‌ی انگلیسی
ترکیبی	Hybrid
ترکیب	Composition
تصریف	Inflection
تصریف اسمی	Declension
تصریف فعلی	Conjugation
تطبیقی	Adaptive
چند-وزنی	N-Gram
چیدمان	Layout
حرف اضافه	Preposition
حرف تعریف	Determiner
حرف ربط	Conjunction
حرف صوت	Interjection
حروف‌چینی	Typographical
خدمات تحت وب	Web Service
خطاهای املایی تکی	Single-error
خطاهای املایی چندگانه	Multi-error

معادل فارسی	واژه‌ی انگلیسی
درخت	Tree
دقت	Precision
دقت میانگین	Mean Average Precision (MAP)
رتبه‌ی وارانه‌ی میانگین	Mean Reciprocal Rank (MRR)
واژک‌شناسی	Morphology
ریشه	Lemma
ریشه‌یابی	Lemmatization
زبان‌شناسی محاسباتی	Computational Linguistics
زمان اجرا	Run Time
ساختار داده	Data Structure
سه‌برگچه‌ای	Ternary
شاخص	Cardinal
صفت	Adjective
ضمیر	Pronoun
ظرفیت	Valency
فاصله‌ی اقلیدسی	Euclidean Distance
فاصله‌ی حروف	Letter Distance
فاصله‌ی رشته‌ای	String Distance
فاصله‌ی میان نویسه‌ها	Character Distance
فاصله‌ی ویرایشی	Edit Distance
فراخوانی	Recall
قطعه	Chunk

معادل فارسی	واژه‌ی انگلیسی
قید	Adverb
کاربردشناسی	Pragmatics
کلید مشابهت	Similarity Key
گره	Node
گفتمان	Discourse
لغوی	Lexical
مبتنی بر بافت	Contextual
مثبت نادرست	False Positive
معناشناسی	Semantics
معنی‌دار	Significant
منفی حقیقی	True Negative
موجودیت اسمی	Name Entity
میانوند	Infix
نادرست	False
نحو	Syntax
نگارشی	Orthographical
نویسه‌ی ترکیبی	Upper Character
واج‌شناسی	Phonology
واج‌شناسیِ تکواژها	Morphophonology
وند	Affix
هم‌شکل	Homoshape
هم‌نشینی، هم‌آیی	Co-occurrence

ضمیمه هفتم

نمایه

ا

آ

ب

پ

ت

چ

ح

خ

د

ر

ز

س

ش

ص

ض

ظ

ف

ق

ک

گ

ل

م

ن

و

ه

Towards Automatic Persian Spell Checking

Omid Kashefi, Mitra Nasri, Kamair Kanani

Appendixes:
Number and Date Converter, Sina Iravanian
Pinglish Converter, Mehrdad Senobari
Punctuation Checker, Kamiar Kanani

Supreme Council of Information and Communication Technology
(SCICT)
Tehran, Iran.
2010