IDMC2008
11-12 NOV
Amir Kabir University

به نام خداوند بخشنده مهربان

روشی جدید در خطایابی املایی در زبان فارسی

محمّدصادق رسولی[۱]، بهروز مینایی بیدگلی[۲]

چکیده

در این مقاله با بررسی روش‌های مختلف خطایابی واژگانی در زبان فارسی، به روشی برای یافتن خطاهای املایی واژگان پرداخته شده است. با اشاره به روش‌های مختلف برای خطایابی واژگانی، چالش‌ها و مشکلات پیش روی این روش‌ها نیز یادآوری شده‌اند. در این روش علاوه بر داشتن ویژگی‌های خاص، مشکلات موجود در رسم‌الخط رایانه‌ای زبان فارسی ــ در مورد حروفی که در رسم‌الخط رایانه‌ای دارای چند نوع حرف هستند ــ حل شده است. بدین ترتیب مشکلات رسم‌الخط فارسی حاصل از انتقال برنامه به رایانه‌های مختلف به طور کامل رفع شده است. خطایاب پس از خطایابی واژگان، پیشنهادهای صحیح را به کاربران ارائه می‌دهد. رهیافت‌های مختلف برای پیشنهاددهی به کاربران مورد بررسی و پیاده‌سازی قرار گرفته‌اند. برنامه برای پیدا کردن پیشنهادهای درست برای واژگان نادرست، از واژگان قبلی و بعدی واژهٔ مورد نظر استفاده می‌کند و خود واژهٔ نادرست را نیز مورد تجزیه قرار می‌دهد تا بتواند ترکیبی از دو واژهٔ درست را از واژهٔ نادرست مورد نظر استخراج کند. ریشه‌یابی اسم‌ها، صفت‌ها، قیود و فعل‌های زبان فارسی در خطایاب املایی مورد بررسی و پیاده‌سازی قرار گرفته است. مصادر افعال در زبان فارسی بر اساس زمان، جداسازی شده و ریشه‌یابی را با توجه به زمان آن‌ها بررسی کرده و وضعیت ضمایر متصل نیز در آن حل شده است. به همین دلیل، دو روش مجزا برای بازیابی افعال زبان فارسی استفاده شده است. همین‌طور در مورد اسم‌ها، مفرد یا جمع بودن، نکره یا معرفه بودن و داشتن وند، بررسی شده است. در ضمن این برنامه قابلیت پیاده‌سازی بر روی نرم‌افزار مایکروسافت آفیس را داراست.

کلمات کلیدی

خطایاب املایی، خطایابی واژگانی، پیشنهاددهی، ریشه‌یابی، زبان فارسی.

A new approach for Persian spellchecking

Mohammad Sadegh Rasooli, Behrouz Minaei-Bidgoli

ABSTRACT

In this paper a method for spellchecking is studied through surveying several methods of spellchecking in Persian language. Referring to several methods of spellchecking, challenges and problems facing these methods are reminded. Besides having special attributes, problems of Persian characters in computer editors -for characters which have more than one code in computer- are solved in this method. Therefore, the problem of portability of the program is removed completely. After spellchecking, the program presents the right suggestions to the user. Different approaches for giving suggestions to the users are studied and implemented. In order to find right suggestions of wrong words, the words before and next to the wrong word are used and the wrong word is also analyzed in

[۱] دانشجوی کارشناسی مهندسی نرم‌افزار دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت ایران
rasooli@comp.iust.ac.ir
[۲] عضو هیأت علمی دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت ایران
b_minaei@iust.ac.ir

order to derive three right words from those. Persian nouns, adjectives, adverbs and verbs stemming are studied and implemented in this spellchecker. Persian verb infinitives are divided into categories according to their tense. Stemming is also done based on the tense. Thus, two separated ways for recovering Persian verbs are used. For nouns, the states of being single or plural, definite or indefinite and having affixes are studied. This program can be implemented into Microsoft Office software.

KEYWORDS
Spellchecker, spellchecking, word suggesting, stemming, Persian Language.

۱. مقدّمه

پردازش زبان‌های طبیعی[۱] از جمله علومی است که پس از به وجود آمدن علوم رایانه‌ای مورد توجه دانشمندان قرار گرفت. در تعریفی که تورینگ از هوش مصنوعی ارائه کرده بود، شاخه‌ای که بیشتر مورد توجه تورینگ بود علمی بود که به پردازش زبان‌های طبیعی معروف شد[۱۲]. با گسترش متون تحریری رایانه‌ای، چالش‌های زیادی در مورد پردازش زبان‌های طبیعی صورت گرفت که نتیجهٔ این تلاش‌ها پیاده‌سازی انواع خطایاب‌های املایی[۲]، مترجمین هوشمند ماشینی، نرم‌افزارهای پردازش و تشخیص گفتار[۳] و نرم‌افزارهای تبدیل متن به صدا[۴] بوده است. با وجود همهٔ این تلاش‌ها، در زبان فارسی شاهد پیشرفت چشمگیری در این زمینهٔ کاربردی نبوده‌ایم. هم‌اکنون برای اجرای یک طرح و یا سامانهٔ نرم‌افزاری در مورد پردازش متون و گفتار فارسی نیاز به دادگان آماری بسیار بزرگی است که از عهدهٔ بسیاری از دانشجویان و محققان، خارج است. با وجود اینکه در بسیاری از افراد متخصص در هوش مصنوعی و نرم‌افزار، این توانایی وجود دارد که بتوانند در زمینهٔ پردازش متون فارسی به فعالیت بپردازند، به دلیل نبود اطلاعات آماری کارا و در دسترس، این فعالیت‌ها بسیار محدود و مقطعی انجام می‌شوند. یکی از گرایش‌های کاری در زمینهٔ پردازش زبان‌های طبیعی، پردازش لغوی و همچنین پردازش ساخت‌واژی می‌باشد که مانند دیگر شاخه‌های پردازش زبان‌های طبیعی در زبان فارسی زیاد مورد توجه قرار نگرفته است. در حالی که روز به روز استفادهٔ کاربران از تحریر[۵] رایانه‌ای زبان فارسی افزون‌تر می‌شود؛ کاربران با مشکلات بسیاری در خطایابی متون‌شان مواجه‌اند که در برخی از مواقع باعث صرف هزینه‌های هنگفت برایشان می‌شود[۶]. در این مقاله به روش‌های خطایابی واژگان فارسی بدون استفاده از دادگان و روش‌های مرسوم آماری پرداخته شده که از قابلیت یادگیرندگی هوشمند ماشینی[۶] بهره گرفته شده است. علاوه بر استقلال روش مورد نظر از دادگان آماری، کارایی اجرایی این روش نسبت به روش‌های مرسوم بسیار بالاتر است و به همین دلیل این برنامه می‌تواند در درازمدت در یک سامانهٔ رایانه‌ای به صورتی بسیار کارا به خطایابی املایی فارسی بپردازد. مشکل دیگری که وجود دارد، این است که امروزه بسیاری از کاربران از نسخه‌های مختلف نرم‌افزار مایکروسافت ورد[۷] برای کار با زبان فارسی استفاده می‌کنند ولی امکانات پردازش متون فارسی‌شان در این نرم‌افزار بسیار محدود و ناکارآمد است. برنامه‌ای که در این مقاله توسط نگارنده پیشنهاد می‌شود، قابل استفاده در نرم‌افزار ورد است.

۲. مشکلات فراروی پردازش زبان فارسی

در مورد پردازش زبان فارسی مشکلاتی وجود دارد که برخی از مهمترین این مشکلات عبارتند از[۱۰]:

مواجهه با چند معنایی و چند نقشی بودن کلمات: برخی لغات مانند کلمه «شیر» دارای چندین معنی هستند که با توجه به بافتاری که در آن واقع می‌شوند، معنی آنها مشخص می‌گردد. بعضی کلمات نیز ماند "در" و "چرا" علاوه بر چند معنی دارای چند مقولهٔ نحوی یا نقش دستوری هستند. این ویژگی منجر به بالا رفتن سطح ابهام در متن می شود.

حذف کلمات و عبارات به قرینهٔ لفظی یا معنوی: در بسیاری موارد کلمات یا عباراتی در یک جمله به قرینهٔ لفظی یا معنوی حذف می‌شود و شنونده باید با تکمیل بخش‌های حذف شده، معنای عبارت را در ذهن خود بازسازی کند. مانند حذف "هستم" در جمله «من خسته هستم و گرسنه» و حذف حروف اضافه "در" در جملهٔ "دکتر خانه نیست".

استفاده از افعال مرکب، اصطلاحات و ضرب المثل‌ها: در زبان‌های مختلف افعال مرکب، اصطلاحات، ضرب‌المثل‌ها و استعارات موارد مشکل‌آفرین در پردازش متون هستند. چرا که معمولا معنایی کاملاً متفاوت با معنای ظاهریشان (ترکیب معانی اجزائشان) دارند. به علاوه

اجزاء چنین ترکیبی لزوماً در جمله به دنبال هم ظاهر نمی‌شوند و ممکن است بین آنها کلمه یا حتی جملهٔ دیگری واقع شود و این یافتن و پردازش سازه مرکب در کل جمله را مشکل می‌کند.

تعیین طبقهٔ اسامی: گاهی اسامی برخلاف ویژگی‌های ظاهری، طبقهٔ خود را تغییر می‌دهند. مثلا در جمله «خاک‌ها را بریز توی باغچه» کلمه خاک‌ها اگرچه علامت جمع دارد ولی تعدد و شمارش را القاء نمی‌کند بلکه بیشتر به مفهوم نکره، جنس و کلیت اشاره دارد. همچنین تشخیص اسامی عام و خاص در کاربردهای مختلف آنها ممکن است ساده نباشد. مانند استفاده از اسامی خاص در نقش عام ("ایران سرزمین فردوسی‌هاست") و یا اسامی عام در نقش خاص (کلمهٔ "پروانه" به عنوان اسم دختران) و هم‌آوایی اسامی خاص در اطلاق به بیش از یک مصداق (اسم "حافظ" برای اطلاق به یک شاعر و یک خیابان).

بی ترتیب بودن زبان: اگرچه فارسی دارای ترتیب مرکزی فاعل – مفعول - فعل است ولی دارای استثنائات فراوان و مکرر در ترتیب کلمات وجود دارد. این مسئله باعث می‌شود ساخت دستور زبان مدون و محاسباتی برای زبان و در نتیجه تجزیه و تحلیل نحوی جملات مشکل شود. مثلا جملهٔ ساده "دیروز من کتاب را در مدرسه به مریم دادم" می تواند به انواع اشکال مختلف با ابجایی متمم‌ها و قید در طول جمله نوشته شود (مانند "من دیروز کتاب را در مدرسه به مریم دادم."، "من دیروز در مدرسه کتاب را به مریم دادم."، "من کتاب را در مدرسه دیروز به مریم دادم. "و "دیروز من در مدرسه کتاب را به مریم دادم.").

کسرهٔ اضافه و حذف آن: در زبان فارسی کسرهٔ اضافه که معمولاً حذف می‌شود، دارای چند نقش است. این علامت، صفت و موصوف و مضاف و مضاف‌الیه را بهم مرتبط می‌نماید. در بعضی موارد علامات"*s'*" و "*of*" در انگلیسی معادل این کسره هستند و در برخی موارد دیگر (مثل اتصال صفت و موصوف) معادلی در انگلیسی ندارد. تشخیص میان نقش‌های مختلف این علامت توسط ماشین کار ساده‌ای نیست، علی‌الخصوص زمانی که سیستم دانش معنایی و کاربردی اندکی داشته باشد. در ضمن حذف کسره اضافه در نوشتار منجر به ایجاد مشکل در تشخیص مرزهای عبارات اسمی می‌شود.

عدم تطابق اجزاء جمله : در زمینهٔ نظری لازم است میان اجزاء مختلف جمله تطابق‌هایی برقرار باشد. مانند مطابقهٔ فاعل و فعل از جهت تعداد، مطابقهٔ اجزاء جمله و اجزاء عبارات اسمی از نظر معنایی. در بعضی موارد برخی از این تطابقات بدون ایجاد خدشه به ساختار معنایی جمله نادیده گرفته می‌شوند. مثل استفاده از فعل جمع برای فاعل مفرد در حالت احترام ("آقای مدیر آمدند") و یا فعل مفرد برای فاعل جمع غیرجاندار (" برگ‌ها می‌ریزد").

وجود ساختار جملات یکسان با معانی و نقش های متفاوت: در زبان فارسی بسیاری نقش های دستوری با علامت و حرف اضافه مشابه مشخص می‌شوند. برای مثال نقش های همراه، حالت، ابزار، وسیله، مقابله یا معاوضه، داشتن و ... با حرف اضافه "با" ظاهر می‌شوند. لذا جملات زیر گرچه از لحاظ ظاهری مشابهند، دارای نقش‌های موضوعی متفاوتی هستند و گاهی برای تشخیص این نقش نیاز به دانش معنایی و کاربردی داریم. برای مثال "با" در جملات "علی با ناراحتی رفت."، "علی با لباس سیاه رفت."، " علی با گربه اش رفت." و " علی با اسبش رفت." به ترتیب نشان‌دهندهٔ حالت فعل، توصیف فاعل، همراه و ابزار است.

مشکلات ناشی از ابهام زبان طبیعی: از ویژگی‌های زبان طبیعی وجود ابهام در آن (حتی برای خوانندهٔ انسانی) است که برخی از انواع آن در زیر برشمرده شده‌اند.

مسائل پردازش محاسباتی: این مشکلات به دلیل این واقعیت پیش می‌آیند که سعی در ایجاد الگوی محاسباتی از زبان برای ماشین داریم. عمدهٔ این مشکلات عبارتند از (۱) فقدان دانش زبانی محاسباتی و مدون مانند عدم وجود واژگان، دستور زبان و الگوها و قواعد زبانی مدون و قابل درک برای ماشین در بسیاری زبان‌ها و به خصوص زبان فارسی، (۲) عدم وجود ابزارها و لوازم پردازش زبان طبیعی برای زبان فارسی مانند تجزیه‌گرهای ساخت‌واژی، نحوی و معنایی معتبر و کارآ (۳) فقدان دانش محاسباتی عرفی و تخصصی که برای رفع آن نیاز به در اختیار داشتن هستان‌شناسی‌های استاندارد عمومی و تخصصی است.

مشکلاتی که در بالا به آن‌ها اشاره شده است در مجموع فراروی پردازش زبان فارسی است. امّا در مورد پردازش لغوی و ساخت‌واژی در زبان فارسی می‌توان به چالش‌های زیر اشاره نمود:

مشکلات رایانه‌ای خط فارسی از قبیل وجود چند مقدار برای حروفی مانند «ک» و «ی»[۹].

مشکلات رسم‌الخطی در فارسی در مورد اعراب‌ها، از قبیل اختلاط رسم‌الخط رایانه‌ای فارسی و عربی. به عنوان مثال ممکن است کسی کلمهٔ «امداد» را به صورت «إمداد» بنویسد که از لحاظ املایی در مورد برخی از واژگان درست و در مورد برخی دیگر نادرست است.

عدم وجود یک معیار قطعی در رسم‌الخط فارسی، به عنوان نمونه می‌توان به طریقهٔ نوشتن اسامی مرکب و مشتق‌مرکب و مشتق در فارسی اشاره کرد که هنوز بر سر جدانویسی، سرهم‌نویسی و یا تلفیقی از این دو بین زبان‌شناسان اختلاف سلیقه وجود دارد. در مورد یای آخر واژگان و این‌که مثلاً واژهٔ «خانه‌ی» درست است یا «خانهٔ» نیز اختلاف نظرهایی هست.

کاربرد کم اعراب در فارسی که در صورت حضور علائم اعرابی در فارسی، پردازشگر زبانی با مشکل مواجه خواهد شد.

مشکل دیگر بحث فاصله‌ها و نیم‌فاصله‌هاست که متأسفانه بسیاری از تحریرگران متون فارسی به این امر توجه نمی‌کنند. به عنوان مثال آن‌ها ممکن است فعل «می‌نویسند» را به صورت «می نویسند» بنگارند. در این نمونه به دلیل این‌که هم واژهٔ «می» و هم واژهٔ «نویسند» خود به تنهایی درستند، پردازشگر به نادرستیِ املا پی نبرد.

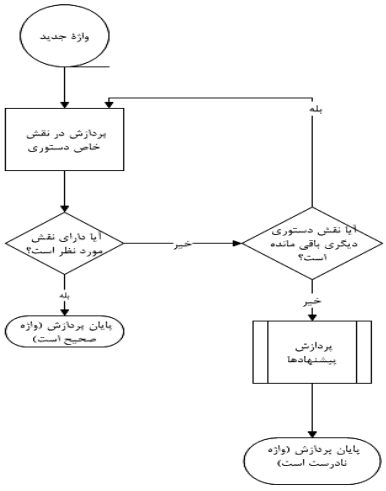
نبودن گنجینهٔ واژگانی مطمئنی در زبان فارسی که به صورت مدوّن تهیه شده باشد.

در مورد حروفی مانند «ر،ژ،ز،د،ا،و،أ» و غیره در فارسی بعضی از تحریرگران متون فارسی به اشتباه در پایان واژگانی که به این گونه از حروف برمی‌خورند، فاصله نمی‌اندازد. مثلاً به جای نوشتن «شیر را خوردم» بنویسد «شیرراخوردم» که باعث اتلاف زمانی زیادی در سامانهٔ پردازشگر می‌شود.

در صورتی که سامانهٔ پردازشگر دارای الگوریتمِ تعمیم‌یافته‌ای برای یافتن ریشهٔ یک واژه باشد، امکان بروز اشتباه بسیار بالا می‌رود. به عنوان مثال، ترکیبِ اسم + لاخ ← اسم که از این ترکیب اسمی می‌توان واژه‌ای مانند «سنگلاخ» را ریشه‌یابی کرد در حالی واژهٔ «خانه‌لاخ» نادرست است.

۳. روش خطایابی

در دیدگاه سنتی بیشتر به تحلیل واژه فارغ از روابط نحوی پرداخته می‌شود و دسته‌های واژگانی ارائه‌شده در این دیدگاه عبارتند از: اسم، فعل، صفت، قید، ضمیر، عدد، صوت و حرف. این دسته‌بندی دو مشکل اساسی دارد. مشکل اوّل این دیدگاه کارا نبودن آن در تحلیل‌های نحوی زبان به روش‌های نوین و مشکل دیگر آن عدم وجود دسته برای برخی از واژگان است[۷]. در مجموع خطاهای متنی در زبان‌ها ۴۰٪ از خطاهای خطاهای واقعی هستند و مابقی خطاهای املایی[۱۷]. خطاهای املایی در حوزهٔ پردازش لغوی و ساخت‌واژی و خطاهای واقعی در حوزهٔ نحوی و بیشتر در حوزهٔ معنایی مورد بررسی قرار می‌گیرند. البته به دلیل اینکه مشکلات این دیدگاه مربوط به پردازش نحوی و معنایی است، در این برنامه الگوریتمی که برای خطایابی هر واژه مورد استفاده قرار می‌گیرد، به این صورت است که در برنامه چند نوع فرهنگ لغت به تفکیک نقش نحوی - مورد استفاده در دیدگاه سنتی-، وجود دارد و همان‌طور که در شکل ۱ دیده می‌شود، هر واژه بسته به ساختار نحوی‌ای که دارد مورد شناسایی و یک فرهنگ لغت برای اسامی خاص نیز وجود دارد که آن نیز به عنوان یکی از فرهنگ‌ها مورد استفاده قرار می‌گیرد. در این روش، واژگان ریشه‌یابی می‌شوند. به همین دلیل برای فعل‌ها دو نوع ریشه‌یابی وجود دارد. یک نوع برای ریشه‌یابی افعال مضارع و مستقبل با استفاده از فرهنگ لغت بن‌های مضارع و دیگری برای ریشه‌یابی افعال ماضی با استفاده از فرهنگ لغت بن‌های ماضی است. اگر برای هر فعل تمام حالات صرفی در فرهنگ لغت اضافه می‌شد و هیچ‌گونه ریشه‌یابی صورت نمی‌گرفت، برای هر بن ۸۰ واژهٔ جدید تولید می‌شد[۴]. در واقع برای واژه‌ای که از لحاظ املایی نادرست است، همهٔ نقش‌ها، همان‌طور که در شکل ۱ دیده می‌شود، برای آن واژه بررسی می‌شود.



شکل ۱: نمودار جریانی پردازش خطایابی یک واژه

۴. روش پیشنهاددهی واژگان

در زمینهٔ پیشنهاددهی واژگان صحیح جایگزین برای واژگان نادرست، بر اساس اینکه با یک متن تحریری مواجهیم یا یک متنی که طی فرایند پردازش تصویر از عکس حاصل شده است، پیشنهادهایی ساخته خواهند شد. در مورد خطاها در تحریر متون به موارد زیر می‌توان اشاره کرد:

- اولین احتمالی که می‌توان داد این است که کاربر سهواً کلیدهای کناری یک حرف را به اشتباه زده باشد. مثلاً به جای واژهٔ «کلنجار» بنویسد «کلمجار»، که کلید «م» کلید کناری «ن» است. یا حتی ممکن است اشتباهاً کلید ***shift*** را زده باشد یا بالعکس؛ مثلاً به جای واژهٔ «ژاله» بنویسد «زاله»، که حرف «ژ» با فشردن همزمان دکمه‌های ***shift*** و «ز» نوشته می‌شود. این امر برای کاربرانی که در حین تحریر بیشتر به صفحه کلید نگاه می‌کنند، بسیار اتفاق می‌افتد. در همین زمینه می‌توان احتمال خطا را برای کاربرانی که به صورت دو دستی تحریر می‌کنند، در نظر گرفت. یعنی حتی کلیدهای غیرمجانب را، با ملاحظهٔ نوع معیار تحریر دودستی (ده‌انگشتی)، نیز در نظر گرفت.
- احتمال دیگری که می‌توان داد این است که کاربر جای دو حرف را در حین تحریر جابجا بنویسد. این امر برای کاربرانی که با سرعت بالا تحریر می‌کنند، خیلی زیاد به وقوع می‌پیوندد. به عنوان مثال کاربر به جای واژهٔ «التیام» بنویسد «الیتام»، که حروف «ی» و «ت» به صورت جابجا قرار گرفته‌اند.
- احتمال دیگر این است که خود کاربر فکر کند که واژهٔ درستی را نوشته است، امّا خود واژه از لحاظ املایی نادرست باشد. مثلاً به جای واژهٔ «باغبان» بنویسد «باقبان»، که حرف «ق» به اشتباه به جای حرف «غ» آمده است. این امر برای کاربرانی که با املای زبان فارسی به خوبی آشنا نیستند و فراگیران غیرایرانی زبان فارسی بسیار اتفاق می‌افتد.
- ممکن است کاربر در حین تحریر یک حرف را اضافه نوشته و یا بالعکس از نوشتن یک حرف خودداری کرده باشد. مثلاً به جای «کاروان» بنویسد «کاوان» و یا به جای «آسمان» بنویسد «آستمان»، که در اوّلین مثال حرفِ «چ» درج نشده و در مثال دوم حرفِ «ت» به اشتباه درج شده است.
- ممکن است کاربر به اشتباه بر روی کلید ***space*** فشرده باشد که باعث جداشدن حروف پیوستهٔ یک واژه به دو واژه می‌شود. به عنوان مثال به جای «کلمات» ترکیب «کل مات» درج شود که با برداشتن فاصله این مشکل رفع می‌شود.
- بعضی اوقات کاربر باید فاصله‌ای را درج نماید و این کار را انجام نمی‌دهد. مثلاً به جای «برای من» بنویسد «برایمن».

۵. بررسی روش معمول

در روش معمول برای خطایابی همهٔ واژگان جداگانه مورد پردازش خطایابی قرار می‌گیرند. برای هر واژه‌ای که نادرست شناخته شد، تعدادی پیشنهاد بر اساس روش‌های مذکور در سرفصل بالا، ساخته می‌شود[۱۸]و[۱۹]. برای هر کدام از پیشنهاد ساخته‌شده باید به صورت جداگانه پردازش صورت می‌گرفت که جزئیات این فرآیند از قرار زیر است:

- به ازای حروف صفحه کلید در یک نوع خاص از صفحه کلید فارسی که در رایانه‌های کیفی بسیار متداول است، ۴۰ نوع حرف تایپی وجود دارد که کلیدهای هم‌تلفظ و همسایهٔ آن بین ۳ تا ۱۱ عدد هستند که به طور متوسط ۷ کلید هم‌تلفظ و همسایه برای هر کلید در رایانه‌ها وجود دارد (البته در این محاسبه کلیدهای همسایه در تحریر دودستی لحاظ نشده‌اند).
- برای یک واژهٔ نادرست c حرفی که حروفشان به طور متوسط g کلید همسایه دارند، باید در نظر داشت به طور متوسط $c*g$ جابجایی باید برای این کار انجام گیرد (به طور دقیق $\sum g_i$ که g_i برابر است با تعداد کلیدهای همسایه به ازای هر یک از حروف در واژه $1<i<=c$) که اگر تمام کلیدهای صفحه کلید را با بسامد یکسان در نظر بگیریم به طور متوسط $c*7$ بار پردازش جابجایی صورت می‌گیرد. (در واقع $c\times7$ کلمهٔ جدید)
- برای یک واژهٔ نادرست c حرفی برای حذف هر کدام از حروف باید در مجموع c بار کلمهٔ جدید تولید کنیم.
- برای جابجایی حروف کنار هم یک واژهٔ نادرست ($c>=2$) باید $c-1$ بار جابجایی صورت بگیرد که در این صورت $c-1$ واژهٔ جدید دیگر خواهیم داشت.
- برای هر واژهٔ نادرست c حرفی باید در بین حروفشان درج صورت بگیرد، در نتیجه با وجود ۴۰ حرف قابل درج $(c+1)\times40$ واژهٔ جدید خواهیم داشت.
- برای هر واژهٔ c حرفی ($c>=4$) می‌توان واژه را به دو واژهٔ جدا تقسیم‌بندی کرد که در نتیجه به تعداد $c-4$ واژهٔ دیگر خواهیم داشت.
- برای واژه‌های کناری هم ۳ حالت پیوستن به قبلی، پیوستن به بعدی و پیوستن به قبلی و بعدی خواهیم داشت.
- در نتیجه جمع اینها می‌شود(K تعداد کلیدهاست):

$$S_{avg}=(c\times g)+c+(c-1)+(c+1)\times k+(c-4)+3=c\times(g+k+3)+(k-2) \quad (۱)$$

یا داریم:

$$S_{\max}=c\times(k+3)+(k-2)+\sum_{i=1}^{c}g_i \quad (۲)$$

در نتیجه اگر در یک متن n واژه‌ای w واژهٔ نادرست وجود داشته باشد، حداکثر تعداد پیشنهادهای ممکن برابر می‌شود (برای جملاتی که کلمات کناری ندارند بین ۱ تا ۳ پیشنهاد کمتر وجود دارد):

$$S_{\max}=\sum_{i=1}^{w}S_i=\sum_{i=1}^{w}\left(C_i\times(K_i+3)+(K_i-2)+\left(\sum_{j=1}^{C_i}g_j\right)\right) \quad (۳)$$

حداقل s ممکن هم برابر است با:

$$S_{\min}=S_{\max}-w\times3 \quad (۴)$$

در نتیجه حجم پردازش بسیار بالایی برای s پیشنهاد به وجود آمده، خواهیم داشت.

- اگر برای پردازش هر واژه p ^ ۸ واحد پردازش انجام دهیم، برای هر کدام از n واژه یک بار بررسی و خطایابی شود در نتیجه برای n کلمه یک بار این کار را انجام می‌دهیم، یعنی به میزان $n\times p$ پردازش برای n واژه خواهیم داشت. برای هر کدام از s پیشنهاد هم باید پردازش صورت بگیرد. یعنی $s\times p$ پردازش هم برای پیشنهادها صورت بگیرد. در نتیجه برای یک متن n واژه‌ای با w واژهٔ خطا که در مجموع s پیشنهاد صحیح وجود دارد، به مقدار زیر پردازش (W_p) نیاز داریم:

$$W_p = n \times p + S \times p = p \times (n + S) \quad (۵)$$

حال اگر فرض کنیم ما فقط ۳٪ خطا در یک متن داشته باشیم (یعنی ۳٪ خطا با توجه به فرهنگ لغت موجود در سامانه کشف شود) و همچنین فرض کنیم متوسط طول واژگان ۴.۵ و متوسط g هم ۷ باشد. برای یک متن ۱۰۰۰۰۰ کلمه‌ای حجم پردازش زیر را خواهیم داشت:

$$\begin{aligned} &N = 100000 \\ &k = 40 \\ &w = 3000 \\ &g = 7 \\ &C_{avg} = 4.5 \end{aligned} \quad (۶)$$

$$S_{avg} = c \times (g + k + 3) + (k - 2) = 4.5 \times (7 + 40 + 3) + (40 - 2) = 263$$

$$S_{total} = w \times s_{avg} = 3000 \times 263 = 789000 \quad (۷)$$

$$W_p = n \times p + S \times p = p \times (n + S) = p \times (100000 + 789000) = 889000 \times P \approx 900000P \quad (۸)$$

۶. الگوریتم بهبوددهندهٔ مورد پیشنهاد

- برای n واژه درون متن نخست با یک سیر یک درخت نسبتاً متوازن می‌سازیم تا بسامد تکرار واژگان در آن ثبت شود. در نتیجه برای کلماتی که بیش از یک بار در سطح درخت تکرار شده‌اند، مجموعاً یک بار مورد پردازش خطایابی قرار می‌گیرند. اگر فرض کنیم که به طور متوسط هر واژه ۱.۷ بار در یک متن به وقوع می‌پیوندد داریم:

$$W_p = (n \div 1.7) \times p + S \times p = p \times ((n \div 1.7) + S) = \frac{p}{1.7} \times (n + 1.7S) \quad (۹)$$

که برای مثال بالا مقدار حجم پردازش به ۸۵۰۰۰۰ واحد پردازش کاهش می‌یابد. (بیش از ۵۰۰۰۰ کاهش در واحد پردازش).

احتمالی که می‌توان داد، بزرگ شدن درخت واژگان است. برای این کار می‌توان از ساختار درخت B[۹] استفاده کرد؛ با این تفاوت که درخت شاخص‌ها[۱۰] در حافظهٔ اصلی ذخیره‌سازی می‌شوند که در نتیجه تعداد مراجعات به حافظهٔ سخت[۱۱] به $O(1)$ کاهش می‌یابد. البته زمانی که درخت واژگانی آن‌قدر بزرگ شود که نیاز به ذخیره‌کردن آن در حافظهٔ جانبی باشد، پروندهٔ متنی هم به همان نسبت بزرگ می‌شود.

- برای واژگان درون متن، یک درخت با حداکثر تعداد شاخه‌های b که $b<k$ است. (k تعداد کلیدهای ممکن است) یعنی با متوازن کردن درخت موجود فرایند جستجو را به حداقل ممکن کاهش می‌دهیم. یعنی به عنوان مقال حروف الف، ب و نون را که پربسامد هستند، یک شاخهٔ مستقل در نظر می‌گیریم ولی حروفی مانند ژ، ظ و ط را می‌توان به عنوان یک شاخه ادغام کرد. در نتیجه با این عما در درخت B ساخته شده برای واژگان، برای هر سطح تعداد شاخص‌ها به اندازه‌ای می‌شود که امید ریاضی دسترسی به حافظهٔ جانبی به $O(1)$ کاهش یابد. درست کردن این درخت به صورت بهینه کاری است که با فرایندهای سادهٔ آماری حل خواهد شد.
- اگر در مجموع w واژهٔ نادرست داشته باشیم و برای همهٔ این واژگان در کل، S پیشنهاد ساخته شود، به جای پردازش کل واژگان به صورت جداگانه تمام S پیشنهاد را در درختی دیگر درست مانند درختی که برای n واژهٔ موجود در متن بود می‌سازیم. این درخت کارایی بسیار بالایی در مورد متونی که خطاهای بسیاری دارند، خواهد داشت. هر کدام از پیشنهادهای S_i که در کل ساخته می‌شوند، نخست در درخت واژگان متن ساخته شده جستجو می‌شوند و اگر لغت پیشنهادشده در متن وجود داشت دیگر نیاز به پردازش نیست و از اطلاعات خطایابی شدهٔ موجود در درخت استفاده می‌شود. در غیر این‌صورت این واژه در درخت درج شده و در صورت وجود در درخت واژگان پیشنهادی، دیگر مورد پردازش قرار نخواهد گرفت[شکل ۲]. بدین‌وسیله حجم بالایی از پردازش کاسته می‌شود. در مجموع تعداد کل پیشنهادها بسیار بالا خواهد رفت ولی با استفاده از ساختار درخت B، مانند آنچه که در درخت واژگان متن بود، می‌توان فضای مورد استفاده از حافظهٔ اصلی را کاهش داد.

به عنوان مثال اگر فرض کنیم با وجود شرایط مندرج در بخش ۱ این مطلب ۱۰٪ از واژه‌های پیشنهادشده در متن وجود داشتند و از ۹۰٪ باقیمانده به طور متوسط هر واژه پیشنهادی ۵ بار پیشنهاد شده باشد، مقدار کل پردازش در رابطهٔ «۸» را داریم:

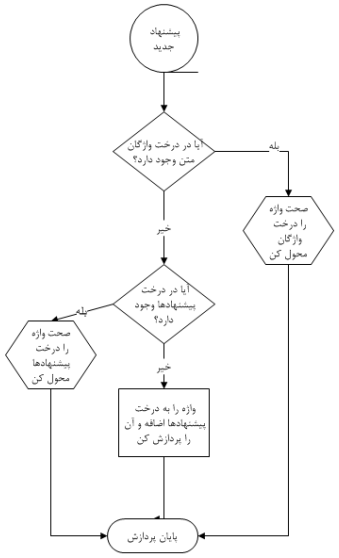
$$S_{total} = 789000$$

$$W_p = (n \div 1.7) \times p + (0.9 \times (S \div 5)) \times p = p \times ((n \div 1.7) + 0.18 \times S) \qquad (۱۰)$$

$$= \frac{p}{1.7} \times (n + 0.306S)$$

که جواب معادلهٔ بالا تقریباً برابر است با ۲۰۰۰۰۰ است که نسبت به مقدار اوّلیه در روش متداول ۴.۵ برابر بر سرعت پردازش برنامه افزوده شده است. البته مقادیر بالا خیلی بدبینانه مطرح شده‌اند و میزان تکرار کلمات پیشنهادی در یک متن با توجه به واحدیت موضوعی متون و استفادهٔ تکراری از واژگان خیلی بیشتر از این مثال است. روند کلی پیشنهاددهی واژگان در شکل ۲ آمده است.

- در مورد نقش‌های دستوری امکان وجود هر نقش در جمله متفاوت است. مثلاً تعداد اسم در یک عبارت بسیار بیشتر از تعداد فعل است. لذا در مورد خطایابی نخست به بررسی نقش‌هایی پرداخته می‌شود که امکان بودنشان بیشتر است (یعنی واحد پردازش p را بدین‌وسیله می‌توان کاهش داد). این کار را می‌توان به صورت تدریجی در یک سامانهٔ رایانه‌ای انجام داد. بدین‌صورت که با اضافه‌شدن هر تعداد واژهٔ صحیح به سامانه، بسامد حضور نقش‌های مختلف دستوری به‌روزرسانی می‌شود.
- در مورد اشکالی که دربارهٔ رسم‌الخط فارسی و وجود چند مقدار برای یک حرف در مورد برخی از حروف اشاره شد، می‌توان از نگاشت این حروف مشترک به یک حرف استفاده کرد. با این کار از حجم فرهنگ لغت و همین‌طور حجم پردازش کاسته می‌شود.



شکل۲: نمودار جریانی پردازش پیشنهادها

۶-۱. روش‌های وزن‌دهی به پیشنهادها

برای پیشنهادهایی که در روش متداول ارائه می‌شود، اگر سامانهٔ آماری مناسبی وجود نداشته باشد هیچ‌گونه روشی بر ارزش‌گذاری یا وزن‌دهی به واژگان نیست. در روش‌های نظیر n-نگاشت هم نیاز به اطلاعات آماری است. در این روش از معیارهای زیر برای وزن‌دهی به پیشنهادها استفاده می‌شود:

۶-۱-۱. پیشنهادهایی که در متن موجودند، وزن بیشتری می‌گیرند و از میان آنها پیشنهادهایی که با بسامد بیشتری در متن بوده‌اند، وزن بیشتری خواهند گرفت.

۶-۱-۲. هر پیشنهادی که به صورت یک پیشنهاد درست مطرح می‌شود، از یک نوع نقش خاص نحوی برخوردار است. به عنوان مثال واژهٔ «می‌بینم» اگر به عنوان یک پیشنهاد مطرح شود از نوع فعل است (در برنامهٔ متداول از ساختارهای نحوی و ریشه‌یابی استفاده می‌شود؛ با این تفاوت که هر کلمه بدون توجه به جایش در جمله بررسی می‌گردد). در بین پیشنهادها نقش‌های نحوی که بیشتر مطرح شده‌اند وزن بیشتری می‌گیرند. مثلاً اگر در مورد یک واژه ۷ پیشنهاد درست وجود داشت که ۴ پیشنهاد فعلی و ۲ پیشنهاد اسمی و ۱ پیشنهاد صفت بود، پیشنهاد فعلی با وزن بیشتری خواهند بود.

در بین حروفی که در مورد پیشنهادها جابجا می‌شوند، نیز یک نوع اولویت‌گذاری می‌توان کرد. به عنوان نمونه در مورد حرف «ن» در رایانه، امکان فشردن اشتباه «م» و «ت» بیشتر از حرفی مثل «ه» است. در حالی که هر سهٔ این حروف در کنار کلید «ن» نشسته‌اند. در ضمن می‌توان در این مورد به تشابه ظاهری بین پیشنهاد و واژهٔ نادرست نیز وزن دارد (برای حالتی که کاربر به زبان فارسی آشنایی ندارد).

۶-۱-۳. مورد حروفی مانند «ر،ژ،ز،د،ا،و،أ» در فارسی بعضی از تحریرگران رایانه‌ای متون فارسی به اشتباه در پایان واژگانی که به این گونه از حروف برمی‌خورند، فاصله نمی‌اندازد. مثلاً به جای نوشتن «شیر را خوردم» بنویسد «شیرراخوردم» که باعث اتلاف زمانی زیادی در سامانهٔ پردازشگر می‌شود. در نتیجه برای چنین حروفی به صورت خاص برای گذاشتن فاصله بین حروف و ساختن دو واژهٔ پیشنهادی از یک واژهٔ نادرست، می‌توان ارزش و وزن بیشتری را قائل شد.

۶-۱-۴. طبق آماری که از به دست آمده است، بسامد نوع خطاها متفاوت است[۱۷]. آمار زیر را داریم:

فشردن کلیدهای کناری: ٪۳۳.۴۵

نزدن یک کلید: ٪۳۳

زدن یک کلید اضافی: ٪۱۹.۵

جابجایی دو حرف کنار هم: ٪۷.۸۵

بقیهٔ حالات: ٪۶.۲

با توجه به آمار فوق می‌توان پیشنهاد را وزن‌دهی کرد. البته این آمار برای زبان فارسی تهیه نشده است، آمار زیر برای خطاهای تحریری در زبان فارسی است:

فشردن کلیدهای کناری: ٪۳۹

نزدن یک کلید: ٪۳۵

زدن یک کلید اضافی: ٪۱۷

جابجایی دو حرف کنار هم: ٪۹

البته در مقادیر فشردن کلیدهای کناری، فشردن کلیدهای دیگر از جمله کلیدهای مجانب در تحریر دودستی نیز لحاظ شده‌اند.

۶-۱-۵. در این الگوریتم دو نوع امکان یادگیری هوشمند وجود خواهد داشت:

۶-۱-۵-۱. در این برنامه واژگان پیشنهادی که مورد قبول واقع شده‌اند، در حافظهٔ برنامه به صورت بلندمدت نگاهداری می‌شوند. بدین‌ترتیب در درازمدت روی یک سامانهٔ رایانه‌ای خاص، دقت برنامهٔ پیشنهاددهندهٔ خطایاب بالاتر می‌رود.

۶-۱-۵-۲. حافظه‌ای برای کل واژگان درستی که از آغاز استفاده از برنامه مورد استفاده قرار گرفته‌اند، به صورت ساختاری با دسترسی تصادفی -برای داشتن سرعت بالا- داشت. در این صورت واژگانی که بیشتر توسط کاربر در درازمدت مورد استفاده قرار گرفته‌اند، دارای وزن بیشتری در پیشنهادها خواهند بود. این ساختار در درازمدت مانند یک پیکرهٔ متنی[۱۲] عمل می‌کند. در ضمن با ذخیره‌کردن واژه‌های قبل و بعد هر واژهٔ درست در یک حافظهٔ جانبی، می‌توان به مرور سامانهٔ مناسبی برای پیاده‌سازی روش n-نگاشت داشت.

۶-۱-۵-۳. در درازمدت می‌توان رفتار نوشتاری-تحریری کاربر را پیش‌بینی کرد. یعنی از روی پیشنهادهایی که کاربر مورد قبول قرار می‌دهد، سامانه به مرور زمان می‌تواند در مورد اینکه با چگونه کاربری کار می‌کند، اطلاعات کسب کند. به عنوان مثال کاربری که بیشتر اشتباهاتش در حوزهٔ واژگانی است که از لحاظ املایی نادرستند، سامانه درمی‌یابد که کاربر آشنایی خوبی با واژگان زبان فارسی ندارد. یا

اگر کاربر بیشتر دچار اشتباه در فشردن کلیدها می‌شود. در نتیجه با توجه به نوع کاربر اعم از اینکه از متون حاصل از پردازش تصویر۱۳ استفاده می‌کند، با زبان فارسی آشنایی ندارد و یا در تحریر (تایپ) فارسی ضعف دارد، به پیشنهادها وزن‌دهی می‌شود.
در نتیجه به ازای هر کدام از یک معیارهای مطروحه در بند ۶-۱-۵ می‌توان یک معادلهٔ خطی با ضرایب مورد نظر (به طوری که بین مولفه‌ها تعادلی برقرار شود) نوشت و در نهایت وزن پیشنهادها را با عددی به صورت W_s ($0<W_s<=1$) نشان دارد. یعنی اگر هر یک ۷ مولفه را λ_1 تا λ_7 بنامیم و این مولفه‌ها دارای ضرایب α_1 تا α_7 باشند، خواهیم داشت:

$$W_s = \sum_{i=1}^{7} \alpha_i \times \lambda_i \quad (۱۱)$$

۷. امکان پیاده‌سازی برنامه در مایکروسافت ورد

مشکلی که هم‌اکنون وجود دارد این است که کاربران بیشتر از نرم‌افزار مایکروسافت ورد برای نوشتن متون فارسی استفاده می‌کنند. در میان‌افزار دات‌نت امکانی به نام افزونه[۱۴] وجود دارد که با این امکان می‌توان اجزایی را به نرم‌افزار آفیس اضافه کرد. با استفاده از روش چندریسمانی[۱۵]، می‌توان یک ریسمان کاری به عنوان خطایاب به این نرم‌افزار اضافه کرد.

۸. نتیجه

همان‌گونه که در متون بالا اشاره شد؛ چند مشکل عدیده در تعامل با پردازش زبان فارسی وجود دارد که در این مقاله به راهکارهایی برای حلّ مشکلِ نداشتن دادگان آماری، با استفاده از الگوریتم‌های هوشمند و یادگیرنده اشاره شد. بنابراین روش ارائه‌شده برای افرادی که قصد استفاده از خطایاب‌های املایی را دارند ولی در عین حال، دادگان آماری مناسبی را در اختیار ندارند؛ بسیار مناسب است. امکان پیاده‌سازی این برنامه بر روی نسخه‌های مختلف نرم‌افزار آفیس نکتهٔ حائز اهمیتی است که می‌توان به آن اشاره کرد. اصلی‌ترین مزیت روش مورد استفاده در این مقاله، امکان تولید دادگان آماری در درازمدت توسط سامانه به صورتی هوشمند بود. در حالی که در نرم‌افزارهای خطایابی از قبیل اسپل[۱۶] و ورد در زبان فارسی، چنین امکاناتی وجود ندارد. از مباحثی که در این زمینه بیشتر جای تحقیق و پژوهش دارد؛ روش‌های تولید دادگان آماری از روی نرم‌افزارهای خطایاب یادگیرنده همزمان با ریشه‌یابی واژگان و دست‌یافتن به بسامد کاربرد واژگان در متون مورد استفاده توسط یک فرد، سازمان و یا گروه خاص است. در نتیجه این آمارها بسته به حرفه، نوع متون و تحصیلات فرد و یا افراد استفاده‌کننده از نرم‌افزار متغیر است و دقت بالاتری خواهد داشت.

۹. تقدیر و تشکر

از جناب آقای مهندس امید کاشفی به خاطر نکاتی که در زمینهٔ ویرایش و اصلاح محتوای این مقاله یادآوری کرده‌اند، بسیار تقدیر و تشکر می‌شود. جا دارد از جناب آقای دکتر فیلی در شرکت داده‌پردازان دوران و همچنین آقای بحرانی در شرکت گویش‌پرداز به خاطر راهنمایی‌های به‌جا و شایسته‌شان تقدیر نماییم. از آقایان مهندس مصطفی کیخا، مهندس ابراهیم شناسا دانشجوی ارشد دانشگاه آزاد تهران، مهندس مهدی نصیری و مهندس مجید ایرانپور دانشجویان کارشناسی ارشد هوش مصنوعی دانشگاه علم و صنعت ایران کمک‌هایشان، بسیار متشکریم. از خانم دکتر کارن مگردومیان و آقای جاناتان دهداری که با وجود مشغلهٔ کاری بسیارشان ما را از راه دور مورد یاری و راهنمایی خودشان قرار می‌دادند، بسیار ممنون و متشکر هستیم. همچنین جا دارد از آقای مهدی لطفی و کمک‌های ایشان در مورد نرم‌افزار مایکروسافت آفیس تشکر شود.

۱۰. مراجع

[۱] مشکوه‌الدینی، مهدی؛ **دستور زبان فارسی: واژگان و پیوندهای ساختی**، سمت، چاپ اول، پاییز ۱۳۸۴.

[۲] **"کاربرد نحو زبان فارسی در پیش‌بینی واژه: ارائهٔ یک مدل آماری از زبان فارسی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۳] **"کاربرد نرم‌افزار پیش‌بینی واژه برای کاربران دچار معلولیت جسمی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۴] **"ارائه یک سیستم تصحیح‌گر املایی زبان فارسی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۵] نعمت زاده، دکتر شهین؛ یزدی پور، دکتر احمد؛ **معرفی اولین هزار واژه پر بسامد نوشتاری فارسی و مقایسه آن با اولین هزار واژه پر بسامد نوشتاری انگلیسی**، دبیرخانه شورای عالی اطلاع رسانی کارگروه خط و زبان فارسی، بهمن ۱۳۸۴.

[۶] **"جهانی‌سازی زبان فارسی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۷] **"ارائهٔ یک واژگان برای کلمات فارسی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۸] **"بررسی روند هم‌نشینی وندهای غیرفعلی فارسی با مقوله‌های مختلف واژگانی و کاربرد آن در پردازش رایانه‌ای زبان فارسی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۹] **"طراحی و پیاده‌سازی یک خطایاب فارسی"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[۱۰] **"پردازش متون فارسی: دستاوردهای گذشته، چالشهای پیش رو"**، دومین همایش فارسی و رایانه، تهران، ۱۳۸۴.

[11] Megerdoomian, Karine, "*Unification-Based Persian Morphology*".

[12] Jurafsky, Daniel; Martin, James H.; *Speech and Language Processing An introduction to natural language processing, computational linguistics and speech recognition*, Prentice Hall, 1999.

[13] Megerdoomian, Karine; "*Finite-State Morphological Analysis of Persian*".

[14] Charniak, Eugene; "*Statistical Language Learning*", Massachusetts Institute of Technology, 1993.

[15] Allen, James; *Natural Language Understanding*, University of Rochester, 1995.

[16] Kukich, K; *Techniques for automatically correcting words in text*, *ACM Computing Survey*, Vol. 14, No. 4, December 1992.

[17] Hussain ,Dr. Sarmad; Naseem, Tahira; "*Spell Checking*", CRULP, NUCES, PAKISTAN, www.crulp.org.

[18] J. Peterson, "Computer programs for detecting and errors", Communications of the ACM, 1980, 23(12): 676

[19] R. Mitton, "Ordering the suggestion of spellcheckers: isolated correction", May, 2006.

---

[1] Natural Language Processing (NLP)
[2] Spell Checkers
[3] Speech Recognition
[4] Text To Speech
[5] Type
[6] Machine Learning
[7] Microsoft Word

[۸] می‌توان p را با $O(1)$ در نظر گرفت.

[۹] تفاوت درخت B با درخت‌های دودویی این است که هر گره از این درخت می‌تواند بیش از یک شاخص در خود داشته باشد و برای ذخیرهٔ اطلاعات انبوه در حافظهٔ جانبی با دسترسی $O(1)$ استفاده می‌شود.

[10] Index
[11] Hard Disk
[12] Corpus
[13] OCR
[14] Add-In
[15] Multithreading
[16] Aspell