# طبقه‌بندی انواع دادگان موردنیاز و روش‌های خطایابی و استانداردسازی متنی



**هادی عبدی قویدل** | کارشناسی ارشد زبان‌شناسی رایانشی
دانشگاه صنعتی شریف

**ملوک‌السادات‌حسینی بهشتی*** | دکتری زبان‌شناسی همگانی
استادیار؛ پژوهشگاه علوم و فناوری اطلاعات ایران(ایرانداک)

دریافت: ۱۳۹٤/۱۱/۰٦ پذیرش: ۹۵/۰۵/۱۲

**چکیده:** یکی از پایه‌ای‌ترین مراحل پردازش خودکار متن، تشخیص خطاهای املایی و استانداردسازی نویسه‌ها است. بدون گذر از این مرحله، ذخیره‌سازی مستندات متنی با مشکلات متعددی مواجه می‌شود که این امر موجب اختلال در بازیابی ماشینی آنها می‌گردد. بدین‌ترتیب، متخصصین حوزه‌های پردازش زبان طبیعی و زبان‌شناسی رایانشی همواره در تلاش هستند تا با ارائه‌ی روش‌ها و الگوریتم‌های مطلوب انواع داده‌ها را در بوته‌ی پردازش قرار داده تا به داده‌ی استاندارد دست یابند. در زبان انگلیسی و برخی زبانهای دیگر، تحقیقات متعددی در این زمینه انجام شده است که به دنبال آن زبان فارسی نیز در این زمینه مورد تحقیق قرار گرفته است. این تحقیقات متعدد گاها در حد پژوهش به قوت خود باقی مانده و گاها در قالب محصول عرضه شده است. مقاله‌ی حاضر به طبقه‌بندی انواع روش‌ها و دادگان موردنیاز در این تحقیقات متعدد پرداخته و فرایند هرکدام از آنها را به طور خاص و نحوه‌ی سنجش میزان دقت پردازش آنها را به طور عام شرح می‌دهد. این مقاله همچنین نحوه‌ی عملکرد سامانه‌های تک‌زبانه‌ی فارسی را توصیف نموده و به نحوه‌ی برخورد آنها با چالش‌های زبان فارسی اشاره می‌کند.

**کلیدواژه‌ها:** تشخیص خطاهای املایی، استانداردسازی نویسه‌ها، طبقه‌بندی روش‌ها، سامانه‌های تک‌زبانه‌ی فارسی، چالش‌های زبان فارسی

* **ملوک‌السادات‌حسینی بهشتی** beheshti@irandoc.ac.ir

فصلنامهٔ علمی پژوهشی
پژوهشگاه علوم و فنّاوری اطلاعات ایران
شاپا(چاپی) ۸۲۲۳-۲۲۵۱
شاپا(الکترونیکی) ۸۲۳۱-۲۲۵۱
**نمایه در LISA، SCOPUS وISC**
http://jist.irandoc.ac.ir
دوره xx | شمارهٔ x | ص ص xx-xx
۱۳xx x

**نوع مقاله: مروری**

**به این مقاله به شکل زیر استناد کنید:**
**دورن متن:**
(عبدی و حسی‌نی ، زودآیند)
**در فهرست منابع:**
عبدی قوی‌دل، هادی و حسینی بهشتی، ملوک‌السادات. زودآیند. طبقه‌بندی انواع دادگان موردنی‌از و روش‌های خطای‌ابی و استانداردسازی متنی. *پژوهشنامه پردازش و مدیریت اطلاعات*.
http://Jipm.irandoc.ac.ir (دسترسی در روز/ماه/سال)

**۱. مقدمه**

نگارش متن همواره دستخوش سلیقه‌های مختلفی در طول تاریخ شده است. هیچگاه بر سر اینکه کدام حالت نوشتاری درست است و کدام غلط، اتفاق نظر کاملی وجود نداشته و تنوع نگارش در انواع متون به وفور دیده می‌شود. گرچه تنوع نگارشی را شاید حاصل خلاقیت ذهنی بشر بدانیم، اما این خلاقیت پردازش ماشینی متن را با چالش‌های متعددی روبرو می‌کند و دقت پردازش داده‌ها را به میزان چشمگیری کاهش می‌دهد. در کنار تنوع نگارشی، غلط‌های سهوی املایی نیز وجود دارد که فحوای متن را منحرف کرده و ماشین را از انجام پردازش‌های دقیق در حوزه‌هایی نظیر ترجمه‌ماشینی، پردازش و تشخیص گفتار، بازیابی اطلاعات و غیره بازمی‌دارد. بدین‌ترتیب، متخصصین حوزه‌های پردازش زبان طبیعی و زبان‌شناسی رایانشی همواره سعی دارند تا با ارائه‌ی روش‌ها و آزمایش الگوریتم‌های داده‌کاوی و پالایش داده به تشخیص غلط‌های املایی بپردازند و نویسه‌های متنی را استاندارد کنند. در این صورت، استانداردترین متن ممکن بر طبق معیارهای زبان‌شناختی توسط ماشین دریافت شده و به تبع آن محصول‌های دقیق‌تری نیز از نظر پایایی و روایی توسط ماشین تولید می‌شود.

از طریق ماژول استانداردسازی مواردی مانند نوع نوشتن تاریخ، حروف فارسی و عربی، فاصله و نیم‌فاصله، خط تیره، علایم نگارشی و کاربردهای متنوع آن‌ها، واحدهای اندازه‌گیری مثل کیلوگرم و kg و یا ک.گ.، اعداد (به‌صورت حروفی و عددی)، فاصله‌ی حاشیه پاراگراف‌ها و درنهایت ساختار بخش‌های مختلف مستندات نظیر مقالات و صفحات وب در سرتاسر دادگان متنی همگن و یکنواخت می‌شوند. از طریق ماژول خطایابی املایی، انواع خطاهای املایی غیرعمدی شناسایی می‌شوند. این خطاها را «ژورافسکی» و «جیمز» چنین طبقه‌بندی کرده‌اند (James and Jurafsky 2009):

الف. خطاهای غیرکلمه‌ای[1] : در این نوع خطاها، کلمه‌ی صحیح به کلمه‌ی ناموجود در فرهنگ‌لغت یک زبان تبدیل می‌شود. مانند: مرسه (املای غلط کلمه‌ی مدرسه)، الما (املای غلط کلمه‌ی املا)

[1] non-word

ب. خطاهای کلمه‌ای[1] : در این نوع خطاها، دو کلمه به لحاظ شباهت املایی و شناختی (آوایی) به‌طور غلط جابجا می‌شوند. مانند شعر و شر (املایی) و مانند ثواب و صواب (شناختی)

وجود انبوه داده در زبان‌های مختلف، پژوهشگران متخصص در علوم زبان‌شناسی رایانشی و پردازش زبان طبیعی را برآن داشته تا در مسیر خودکارسازی استانداردسازی و خطایابی داده‌های متنی گام بردارند. تنها دلیل این امر را می‌توان در صرف وقت و هزینه‌ی زیاد در عملیات پیش‌پردازش داده یافت. بدین‌ترتیب، در زبان انگلیسی و برخی زبان‌های دیگر نظیر عربی، چینی و عبری، تحقیقات متعددی در زمینه ساخت استانداردساز و خطایاب انجام شده‌است که به دنبال آن زبان فارسی نیز در این زمینه مورد تحقیق قرارگرفته‌است. بسیاری از این تحقیقات گاه در قالب پژوهش آزمون و خطا انجام شده و گاه در قالب محصول تجاری در بازار بین‌الملل عرضه شده است. تحقیقات انجام‌شده هیچکدام به دقت ۱۰۰ درصد نرسیده‌اند و بسیاری از پژوهشگران اکنون نیز ساختن استانداردساز و خطایاب مشغول هستند. از این رو، بررسی نحوه‌ی ساخت داده و همچنین روش‌های انجام‌شده می‌تواند به‌عنوان مرجعی هدایتگر برای پژوهشگران حال حاضر عمل کرده و مسیر استانداردسازی و خطایابی متن را تبیین نماید.

در این مقاله سعی داریم این تحقیقات متعدد را در قالب دادگان مورد نیاز و روش‌ها طبقه‌بندی کرده و به شرح فرایند هرکدام به طور خاص و نحوه‌ی سنجش میزان دقت پردازش آنها به طور عام بپردازیم. همچنین، قصد داریم چالش‌های زبان فارسی را نیز طبقه‌بندی کنیم و نحوه‌ی عملکرد سامانه‌های تک‌زبانه‌ی فارسی را در راستای چالش‌های زبان فارسی شرح دهیم.

## ۲. دادگان مورد نیاز و روش‌های خطایابی و استانداردسازی متن

### ۲-۱. دادگان موردنیاز

در این بخش به طبقه‌بندی انواع داده و ساختار آن برای استانداردسازی و خطایابی متنی می‌پردازیم. جدول ۱ به این داده‌ها و انواع ساختار آن اشاره می‌کند.

[1] real-word

**جدول ۱ انوع داده و ساختار آن**

| داده | ساختار داده |
|---|---|
| پیکره | پیکره‌ی موازی |
| | پیکره‌ی غلط |
| فرهنگ‌لغت | فرهنگ‌لغت نمایه‌گذاری‌نشده |
| | فرهنگ‌لغت نمایه‌گذاری‌شده |

در ادامه، هرکدام از انواع داده‌های موجود در جدول ۱ تشریح می‌شود.

**۲-۱-۱. پیکره**

**۲-۱-۱-۱. پیکره‌ی موازی**[1] : پیکره‌های موازی مناسب برای انجام استانداردسازی و خطایابی متشکل از جملات درست و غلط همترازشده هستند که با به‌کارگیری الگوریتم‌های حوزه‌ی ترجمه‌ماشینی آموزش داده‌شده و قابل‌استفاده می‌باشند (Schlippe et al. 2010)

**۲-۱-۱-۲. پیکره‌ی غلط**[2] : این نوع پیکره از مجموعه‌ی متونی جمع‌آوری می‌شود که به طور طبیعی دارای غلط املایی هستند. معروف‌ترین این نوع پیکره، پیکره‌ی انگلیسی بیرکبک[3] دانشگاه آکسفورد و همچنین مجموعه‌ی غلط‌های املایی پیتر نورویگ[4] است که از ویکی‌پدیا و بیرکبک استخراج‌شده است. پیکره‌ی بیرکبک دارای ۳۶۱۳۳ شکل غلط ۶۱۳۶ کلمه انگلیسی از گروه‌های سنی مختلف است.

**۲-۱-۲. فرهنگ‌لغت**[5]

[1] parallel corpus
[2] misspelled corpus
[3] Birkbeck: http://www.ota.ox.ac.uk/headers/0643.xml
[4] Peter Norwig
[5] lexicon

**۲-۱-۲-۱. فرهنگ‌لغت نمایه‌گذاری‌نشده:** فرهنگ‌لغت‌هایی که به‌طور بهینه نمایه‌گذاری نشده باشند، در زمره این گروه‌ها قرار می‌گیرند. اینگونه فرهنگ‌لغت‌ها، در کارهای پردازشی حجیم به‌هیچ‌وجه قابل‌استفاده نیستند.

**۲-۱-۲-۲. فرهنگ‌لغت نمایه‌گذاری‌شده:** این نوع فرهنگ‌لغت‌ها با توجه به نوع تحقیق و یا نرم‌افزاری که در آن به کار می‌روند، نمایه‌گذاری می‌شوند. از انواع روش‌های نمایه‌گذاری، می‌توان به روش ایجاد جدول درهم‌سازی یا هش[1] (جدولی که لغات آن دارای کلیدهای خاصی نظیر نمایه، حرف‌های الفبا و یا بسامد باشند (Mouth et al. 1978؛ Bhatti et al. 2014)) و مدل درختی (لغات به‌صورت سلسله‌مراتبی مرتب شوند (Beliga et al. 2015)) اشاره کرد.

**۲-۱-۲-۳. فرهنگ‌لغت آموزش‌داده‌شده:** این نوع فرهنگ‌لغت‌ها با استفاده از الگوریتم‌های خوشه‌بندی[2] و مدل‌های موضوع پنهان[3] (مدلی که بر اساس آن توکن[4] های مجموعه‌ی انبوهی از داده‌ها در موضوع‌هایی مانند موضوع ۰، موضوع ۱ و غیره قرار می‌گیرند) آموزش‌داده‌شده و سپس نمایه‌گذاری می‌شوند تا سرعت پردازش به میزان قابل‌توجهی افزایش یابد. انواع روش‌های خوشه‌بندی نظیر کی-مینز[5]، کی-مدویدز[6] و غیره وجود دارد. «زمپیری» و همکاران بر این باور بوده‌اند که فرهنگ‌لغت هرچقدر هم بزرگ و حجیم باشد، می‌توان آن را از طریق خوشه‌بندی کی-مدویدز به خوشه‌های مختلفی تقسیم کرد (Zampieri et al. 2013). پس هنگام محاسبه فاصله لغات، مدویدی که به کلمه غلط شبیه‌تر باشد انتخاب شده و تنها فاصله‌ی مدخل‌های آن خوشه با کلمه غلط مقایسه شده و بدین‌ترتیب کلمه درست انتخاب می‌شود.

### ۲-۲. روش‌های خطایابی و استانداردسازی

در این بخش به طبقه‌بندی انواع روش‌های استانداردسازی و خطایابی متنی می‌پردازیم. جدول ۲ به این روش‌ها و معیارهای اصلی هر روش به طور مختصر اشاره می‌کند.

---

[1] hash table
[2] clustering
[3] Latent Topic Models
[4] token
[5] k-means
[6] k-medoids

**جدول ۲ روش‌های استانداردسازی و خطایابی و معیار کلیدی هر یک**

| روش | معیار |
|---|---|
| فاصله‌ی حرف | شباهت |
| بسامد کلمات | کلمات پرتکرار و کلیدی |
| فاصله‌ی ویرایشی کمینه | کمترین فاصله با کلمه‌ی غلط |
| آماده‌سازی | مدل شناختی |
| مدل نویزی | بسامد شرطی |
| مدل‌های زبانی | احتمال |
| مدل مبتنی بر دسته‌بندی | شباهت |
| مدل‌های آوایی | آوا به جای حرف |

در ادامه، هرکدام از انواع روش‌های مطرح‌شده در جدول ۲ تشریح می‌شود.

**۲-۲-۱. روش فاصله‌ی حرف**[1]**:** در این روش نخست حرف‌های دو کلمه، یک‌به‌یک و سپس دوبه‌دو مقایسه می‌شوند و در پایان حرف‌های اول مقایسه می‌شوند. بر اساس عدم وجود مشابهت در هر مرحله، امتیاز منفی در نظر گرفته می‌شود. به عنوان مثال، فاصله‌ی حرف در دو کلمه ''nicer'' و ''nised'' برابر با منفی ۵ است.

**۲-۲-۲. بسامد کلمات:** در این روش، الگوریتم به‌صورت دومرحله‌ای عمل می‌کند. در مرحله نخست، الگوریتم کلمه را با کلمات پربسامد مقایسه می‌کند و در صورت عدم یافت کلمه درست در بین این کلمات، به سراغ کلمات کم‌بسامدتر می‌رود. به‌عنوان مثال، می‌توان کلمات را بر اساس ویژگی بسامد در جدول هش مرتب نمود. لازم به ذکر است که می‌توان به جای محاسبه‌ی بسامد

[1] letter distance

محض، از بسامد وزن‌دار (حاصلضرب بسامد پیکره‌ای و فاکتور وابستگی به طبقه‌ی خاص[1] ( Salton and Buckley 1988)) نیز استفاده نمود.

**۲-۲-۳. فاصله‌ی ویرایشی کمینه[2]:** روش‌های متفاوتی برای محاسبه این نوع فاصله وجود دارد که به تک‌تک آنها اشاره می‌کنیم.

**۲-۲-۳-۱. فاصله‌ی همینگ[3]** (Hamming 1950): اندازه‌ی فاصله‌ی همینگ برای دو رشته با طول مساوی، برابر با تعداد جایگاه‌هایی است که حرف‌های متناظر متفاوت باشند. برای مثال، فاصله‌ی همینگ دو کلمه‌ی "nice" و "bise" برابر با ۲ است. از آنجایی که این عدد با اعداد متناظر در یک دامنه‌ی مشخص قرار ندارند، بهتر است این عدد تبدیل به عددی مابین ۰ و ۱ (نرمال) شود و به بزرگترین مقدار متناظرشان تقسیم گردد. این مقدار متناظر برابر است با طول واژه. درنتیجه فاصله‌ی این دو کلمه، ۲ بر ۴ یا ۰/۵ در نظر گرفته می‌شود.

**۲-۲-۳-۲. فاصله‌ی لونشتاین[4]** (Levenshtein 1965) : بر اساس این روش، دو لغت در قالب تشکیل ماتریسی باهم مقایسه شده و کمترین تعداد تغییراتی که باید بر روی لغت اعمال کرد تا به لغت دیگر رسید مشخص می‌شود. این تغییرات شامل درج حرف‌اضافه[5] ، حذف حرف[6] و جایگزینی حرف[7] است. برای مثال، فاصله لونشتاین دو کلمه‌ی "زبان" و "زندان" ۲ است. در جدول ۳، آخرین عدد ماتریس که در گوشه‌ی پایین سمت راست قرار دارد فاصله‌ی لونشتاین دو کلمه است.

[1] the product of term frequency and inverse document frequency
[2] minimum edit distance
[3] Hamming
[4] Levenshtein
[5] insertion
[6] deletion
[7] substitution

**جدول ۳ ماتریس لونشتاین "زبان" و "زندان"**

| | | ز | ب | ا | ن |
|---|---|---|---|---|---|
| | ۰ | ۱ | ۲ | ۳ | ۴ |
| ز | ۱ | ۰ | ۱ | ۲ | ۳ |
| ن | ۲ | ۱ | ۱ | ۲ | ۲ |
| د | ۳ | ۲ | ۲ | ۲ | ۳ |
| ا | ۴ | ۳ | ۳ | ۲ | ۳ |
| ن | ۵ | ۴ | ۴ | ۳ | ۲ |

در تازه‌ترین مطالعات، «لیو» و همکاران روش فاصله‌ی افزایش‌یافته[1] را پیشنهاد کردند (Liu et al. 2012). برای انجام این کار، آنها پیکره‌ای را آماده‌کردند که دارای دو نوع فرایند آموزش بوده است.

الف. آموزش انتخاب جفتی آگاه از بافت[2]: در این مرحله، بین کلمه و توکن برداری تشکیل داده و میزان شباهت بافتی بین آنها از طریق تعیین میزان فاصله‌ی کسینوسی بررسی می‌شود.

ب. برچسب‌زنی توالی کاراکتری[3]: با استفاده از مدل شرطی میدان تصادفی (سی آر اف[4])، انواع یک کلمه را با برهم زدن نظام واجی، سیلابی و صرفی تولید می‌کنند. در این روش، هر کلمه موجود در فرهنگ‌لغت قبل از بهم‌خوردن شکافته شده و برحسب لایه‌های واجی، سیلابی و صرفی برچسب‌های Begin-Inside-Last-Outside-Unit می‌خورد. نمونه‌ای از این نوع برچسب‌زنی در شکل ۱ قابل‌مشاهده می‌باشد.

[1] Enhanced Level Transformation
[2] Context-aware Training Pair Selection
[3] Character-level Sequence Labeling
[4] Conditional Random Fields: در این مدل، احتمال رویداد یک متغیر تصادفی بر اساس رویداد متغیر تصادفی همسایه تعریف می‌شود

| Character | a | d | v | e | r | t | i | s | e | m | e | n | t | s |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Phoneme | AE | D | V | ER | ER | T | AY | Z | _ | M | AH | N | T | S |
| Phoneme boundary | O | O | O | Bl | Ll | O | O | O | O | O | O | O | O | O |
| Syllable boundary | B | L | B | I | L | B | I | I | L | B | I | I | I | L |
| Morpheme boundary | B | I | I | I | I | I | I | I | L | B | I | I | L | U |
| Word boundary | B | I | I | I | I | I | I | I | I | I | I | I | I | L |

شکل ۱ نمونه برچسب‌زنی برای کلمه advertisement

«لیو» و همکاران بر اساس این ایده، فرایند استانداردسازی را به سه زیراستانداردساز[1] تقسیم کردند و سپس خروجی‌های این سه زیراستانداردساز را ترکیب کرده و عملیات نهایی استانداردسازی را بر اساس لونشتاین بر روی متن انجام دادند.

**۲-۲-۳-۳.** فاصله‌ی دامرو-لونشتاین[2] (Damerau 1964): این روش تا حدودی همانند روش لونشتاین است. با این تفاوت که جابجایی حرف‌ها[3] نیز به تغییرات روش لونشتاین اضافه شده است. برای مثال، فاصله‌ی لونشتاین دو کلمه‌ی "زبان" و "بزان" ۲ است. در صورتی که، فاصله‌ی دامرو-لونشتاین این دو کلمه‌ی ۱ است. چرا که با یک عمل جابجایی، می‌توان این دو کلمه را به همدیگر تبدیل کرد. در جدول ۴، آخرین عدد ماتریس که در گوشه‌ی پایین سمت راست قرار دارد فاصله‌ی دامرو-لونشتاین دو کلمه است.

**جدول ٤ ماتریس دامرو-لونشتاین "زبان" و "زندان"**

| | | ز | ب | ا | ن |
|---|---|---|---|---|---|
| | ۰ | ۱ | ۲ | ۳ | ۴ |
| ب | ۱ | ۱ | ۱ | ۲ | ۳ |
| ز | ۲ | ۱ | ۱ | ۲ | ۳ |
| ا | ۳ | ۲ | ۲ | ۱ | ۲ |
| ن | ۴ | ۳ | ۳ | ۲ | ۱ |

[1] Subnormalizer
[2] Dameru- Levenshtein
[3] transposition

۲-۲-۳-۴. فاصله‌ی ونگر-فیشر[1] (Wagner and Fischer 1974): این روش همانند روش لونشتاین دارای تغییرات درج حرف‌اضافه، حذف حرف و جایگزینی حرف است. با این تفاوت که از لحاظ زمانی بهینه شده و برای تشخیص هر تغییر در رشته، هزینه‌ای خاص در نظر گرفته می‌شود و سپس بهینه‌ترین مسیر برای دستیابی به جواب نهایی انتخاب می‌شود. به عنوان مثالی ساده، رشته‌های دو کلمه‌ی"زبان" و "بزان" به دو بخش زیر تقسیم می‌شود: ("زب" و "زن") و ("ان" و "ان"). بدین ترتیب مسیر بهینه‌ی اولی برای دستیابی به جواب نهایی ماتریس انتخاب شده و محاسبه‌ی فاصله‌ی لونشتاین بر روی آن انجام می‌شود. الگوریتم این روش، در زمره الگوریتم‌های برنامه‌سازی پویا است. بدین معنی که در فرایند انجام محاسبه این فاصله، رشته‌ها به بخش‌های مختلف تقسیم شده و تغییرات به همراه وزن منحصربفرد در آن بخش محاسبه می‌شود.

۲-۲-۳-۵. اوکونن[2] (Ukkonen 1992): روش تغییریافته ونگر فیشر است که با کاهش پیچیدگی زمانی و فضایی قابل توجهی همراه شده است.

۲-۲-۳-۶- فاصله‌ی نیدلمن وانچ[3] (Needleman and Wunsch 1970): این روش در تشخیص شباهت ژن‌ها مورد استفاده قرار می‌گیرد. الگوریتم این روش، همانند روش ونگر-فیشردر زمره الگوریتم‌های برنامه‌سازی پویا است. با این تفاوت که در حوزه پردازش‌زبان‌طبیعی فاصله کمینه و در حوزه بیوانفورماتیک بیشینه شباهت ملاک سنجش است. در جدول ۵، آخرین عدد ماتریس که در گوشه‌ی پایین سمت راست قرار دارد شباهت نیدلمن وانچ دو کلمه است.

[1] Wanger-Fischer
[2] Ukkonen
[3] Needleman–Wunsch

**جدول ۵ ماتریس شباهت "زبان" و "زندان"**

| | | ز | ب | ا | ن |
|---|---|---|---|---|---|
| | ۰ | -۱ | -۲ | -۳ | -۴ |
| ب | -۱ | -۱ | ۰ | -۱ | -۲ |
| ز | -۲ | ۰ | -۱ | -۱ | -۲ |
| ا | -۳ | -۱ | -۱ | ۰ | -۱ |
| ن | -۴ | -۲ | -۲ | -۱ | ۱ |

۲-۲-۳-۷. فاصله‌ی هیرشبرگ[1] (Hirschberg 1975): این روش، نوع تغییریافته‌ی تقسیم‌وحل[2] الگوریتم نیدلمن وانچ است. بدین‌گونه که هیرشبرگ، الگوریتم نیدلمن‌وانچ را به چند مسیله‌ی دیگر تقسیم کرده و سعی می‌کند نخست آنها را حل کند و سپس به شباهت (یا فاصله‌ی کمینه) برسد. این نوع فاصله سبب شده است که میزان اشغال حافظه به میزان قابل توجهی کاهش یابد.

۲-۲-۳-۸- فاصله‌ی اسمیث-واترمن[3] (Smith and Waterman 1981): این روش همانند روش نیدلمن وانچ است. با این تفاوت که کلیه‌ی نمایه‌های منفی در ماتریس اسمیث-واترمن برابر با ۰ می‌باشند. پس از تشکیل این ماتریس، عملیات پس‌گردی[4] از عدد آخر شروع و آنقدر ادامه پیدا می‌کند تا این که صفر برسد.

۳-۲-۳-۹. شباهت جرو-وینکلر[5]: این روش که مدل تغییریافته‌ی «جرو» (Jaro 1989) است، میزان تشابه دو کلمه (معمولا اسامی خاص) را محاسبه می‌کند. در این روش میزان حرف‌های مشترک و تعداد جابجایی حرف‌ها بسیار مهم است. فرمول محاسبه‌ی جرو-وینکلر بدین‌گونه است:

$$(1)\ d_w = \frac{1}{3}\left(\frac{m}{|s_1|}+\frac{m}{|s_2|}+\frac{m-t}{m}\right)+l\rho\left(1-\left(\frac{1}{3}\left(\frac{m}{|s_1|}+\frac{m}{|s_2|}+\frac{m-t}{m}\right)\right)\right)$$

در فرمول ۱، m تعداد حروف مشترک بین دو کلمه، s کلمه، l تعداد حروف مشترک تا ۴ کاراکتر اول و ρ عدد ثابت وینکلر (معمولا ۰/۱) است. به عنوان مثال، شباهت جرو-وینکلر دو کلمه‌ی "مهرناز" و "مهرانز" طبق فرمول ۱، ۰/۹۶ است.

[1] Hirschberg
[2] Divide and conquer
[3] Smith–Waterman
[4] backtracking
[5] Jaro-winkler

**۲-۲-٤. آماده‌سازی**[1] **:** پاسخ ذهن به محرک جدید (یادگیری مطلب جدید)، بر اساس تاثیرپذیری غیرصریح حافظه از محرک قبلی(خود مطلب و بافت مطلب قبلی). «لیو» و همکاران معتقد هستند که آماده‌سازی حافظه، نقش مهمی در درک توکن غلط دارد (Liu et al. 2012). طبق ایده‌ی طراحان این سیستم، چند حرف اول کلمه نقش مهمی در تشخیص کلمه غلط دارند. آماده‌سازی طبق فرمول ۲ اندازه‌گیری می‌شود:

$$(2)\ VisualPrime(s_i|t_i) = \frac{len(LCS(t_i,s_i))}{len(t_i)} \times \log(TF(s_i))$$

طبق این فرمول، نخست TF یا بسامد واژه در پیکره محاسبه می‌شود. سپس، اندازه‌ی طول طولانی‌ترین کلمه‌ای که با حرف اول کلمه غلط آغاز شده باشد (len(LCS[2]) از جدول کلمات پیدا شده و بر اندازه‌ی طول کلمه غلط (len(t تقسیم می‌شود. بر این اساس، بزرگ‌ترین عدد حاصل برای این معادله تعیین‌کننده کلمه درست خواهد بود. لازم به ذکر است که این نوع آماده‌سازی با نوع بکار رفته در مسائل شناختی از این لحاظ متفاوت است که در نوع شناختی بافت به کلمه‌ی ماقبل مربوط می‌شود و در این نوع آماده‌سازی، پیکره به‌طور کامل به عنوان بافت در نظر گرفته می‌شود.

**۲-۲-۵. مدل کانال نویزی**[3] **:** مدل کانال نویزی، چارچوبی است که به‌صورت متداول در غلط‌یابی املایی استفاده می‌شود (Shannon 1948). در گام اول این مدل، یکی از روش‌های فاصله‌ی ویرایشی برای کلمه محاسبه شده و لیستی از گزینه‌های درست برای کلمه غلط ارائه می‌شود. سپس، آرگومان ماکزیمم مدل کانال نویزی طبق فرمول ۳ برای کلمه محاسبه می‌شود.

$$(3)\ NCM = \underset{w \in V}{argmax} P(x|w)P(w)$$

در این فرمول، x کلمه غلط و w کلمه موجود در فرهنگ‌لغت است.

**۲-۲-٦. مدل‌های زبانی:** مدل‌های زبانی (Chen et al. 2009؛ Liu et al. 2011؛ Wu et al. 2010)، احتمال یک رشته از واحدهای زبانی را با استفاده از توزیع احتمال (چندنگاشتی[4]) محاسبه می‌کند. در این مدل‌ها نیز، نخست یکی از روش‌های فاصله‌ی ویرایشی برای کلمه محاسبه شده و لیستی از

[1] priming
[2] Longest Common Subsequence
[3] Noisy Channel Model
[4] n-gram

گزینه‌های درست برای کلمه غلط ارائه می‌شود. سپس، احتمال‌های چندنگاشتی آنها با توجه به کلمات قبل و بعد آن در متن از روی پیکره محاسبه شده و از طریق آرگومان ماکزیمم، بهترین گزینه برای کلمه غلط انتخاب می‌شود.

**۲-۲-۷. مدل مبتنی بر دسته‌بندی**[1]**:** با استخراج انواع ویژگی‌های ذکرشده در ۲-۱-۲-۳، می‌توان از انواع دسته‌بندها همچون نایو بیز و مدل بردار پشتیبان در غلط‌یابی املایی و تصحیح آن استفاده کرد.

- مدل نایو بیز [2] یا بیزین ساده، یک مدل احتمالی شرطی است. بدین‌صورت که احتمال وقوع رخداد وابسته به احتمال وقوع رخداد قبلی باشد. این روش دارای پیچیدگی به‌مراتب کمتری است و در بین دسته‌بندها، از مقبولیت بالایی برخوردار است. «گولدینگ» از این مدل برای انتخاب گزینه‌ی درست از مجموعه‌ی گزینه‌ها با توجه به بافت آنها استفاده کرده است (Golding 1995). آرگومان ماکزیمم مدل ناویو بیز طبق فرمول ۴ محاسبه می‌شود.

$$(4)\ p(c_{-k},\dots,c_{-1},c_1,\dots,c_k|w_i) = argmax \prod_{j\in -k,\dots,-1,1,\dots,k} p(c_j|w_i))$$

در این فرمول، w کلمه و c بافتی که کلمه در آن موجود است.

- مدل ماشین بردار پشتیبان[3] : مدل بردار پشتیبان را «کورتس» و «واپنیک» در سال **۱۹۹۵** معرفی کرده است (Cortes and Vapnik 1995). این مدل، از جمله مدل‌های غیراحتمالی است و با تشکیل صفحات هایپر[4] و محاسبه‌ی حد توابع آن عمل دسته‌بندی را انجام می‌دهد. «شبک» و «لی» با استخراج ویژگی‌های نویسه‌ای، آوایی، صرفی، نحوی و معنایی به دسته‌بندی آنها توسط ماشین بردار پشتیبان پرداختند (Schaback and Li 2007).

**۲-۲-۸. مدل‌های آوایی**[5]**:** مدل‌های آوایی کمک می‌کنند تا بتوان به جای حروف الفبا، از تلفظ کلمات جهت تشخیص توکن‌های غلط استفاده نمود.

---

[1] classification
[2] Naïve Bayes
[3] Support Vector Machine
[4] Hyperplane
[5] Phonetic Models

الف. ساندکس[1] (Hall and Dowling 1980): ساندکس، روشی برای نمایه‌گذاری اسامی بر اساس تلفظ آنها است. رمز آوایی ساندکس برای هر اسم، شامل یک حرف به همراه عددی سه‌رقمی است. این یک حرف، اولین حرف اسم است و ارقام حرف‌های صامت باقی‌مانده را رمزگذاری می‌کنند. برای حرف‌های صامتی که جایگاه تولید آنها یکسان است، رقم یکسان در نظر گرفته می‌شود. برای مثال، حرف‌های صامت لبی p، f، b و v هرکدام با عنوان شماره ۱ رمزگذاری می‌شوند.

ب. فونیکس[2] (Gadd 1990): فونیکس نوع بهبودیافته‌ی ساندکس است. هر یک از حرف‌ها به مجموعه‌ای از رمزهای ساندکس نگاشت می‌شوند. قبل از انجام فرایند نگاشت نیز، حدود ۱۶۰ تغییر گروه–حرفی به‌کاربرده‌می‌شود تا استانداردسازی رشته انجام‌شود. برای مثال رشته ''tjV'' (که ''V'' یک مصوت است) اگر در ابتدای رشته باشد، به ''chV'' نگاشت می‌شود و ''x'' تبدیل به ''ecs'' می‌شود (Zobel and Dart 1996). این تغییرات زمینه‌ای برای رمزگذاری آوایی به وجود می‌آورند و امکان تشخیص رشته‌هایی نظیر ''c'' و ''s'' را مهیا می‌سازند.

پ. شیوه‌ی کیونگاشتی[3] (Ukkonen 1992): شیوه‌ی کیونگاشتی همان اندازه‌گیری فاصله‌ی رشته‌ای بر اساس تعداد کیونگاشت است. کیونگاشت رشته‌ی ''s''، همان زیررشته ''s'' با طول ثابت ''q'' است. یک نمونه‌ی ساده از این اندازه‌گیری، انتخاب ''q'' و شمردن تعداد کیونگاشت‌های مشترک بین دو رشته است. بااین‌حال، شمارش تنها کیونگاشت‌ها نمی‌تواند نمایانگر تفاوت طولی باشد. برای حل این مشکل، اوکونن نوع جدیدی از فاصله کیونگاشتی پیشنهاد کرده که می‌توان برای رشته‌های بدون تکرار کیونگاشتی به‌صورت فرمول ۵ تعریف کرد:

$$(5)\quad |G_s| + |G_t| - 2|G_s \cap G_t|$$

که در آن $G_s$ مجموعه‌ای از کیونگاشتی‌ها در رشته ''s'' می‌باشد. برای مثال، طبق این فرمول فاصله بین ''rhodes''و ''rod''، برای q با مقدار ۲ یا ۳، ۵ است.

[1] Soundex
[2] Phonix
[3] q-gram

ت. گرپ[1] : عبارت‌های باقاعده‌ی گرپ برای تطبیق الگو مورد استفاده قرار می‌گیرند. انواع آن اِی‌گرپ (گرپ تقریبی)، ای‌گرپ و اف‌گرپ است (Wu and Manber 1992). از میان این روش‌ها، اِی‌گرپ بیشتر برای خطایابی املایی به کار می‌رود. اساس کار این روش، الگوریتم لونشتاین است و برای بهبود کارایی از الگوریتم‌های مختلفی استفاده می‌کند. اِی‌گرپ برای الگوهای ساده تشخیص خطا از الگوریتم بویر مور[2] استفاده می‌کند. شامل الگوریتمی سریع برای شناسایی رشته‌های دارای زیررشته می‌باشد. اِی‌گرپ به جستجوی حداکثر ''k'' درج، حذف یا جایگزینی می‌پردازد که در آن ''k'' ضریب ثابت از پیش‌تعریف‌شده‌ای می‌باشد.

ث. ادیتکس[3] (Zobel and Dart 1996): ادیتکس سنجش فاصله آوایی است که ویژگی‌های فاصله‌ی ویرایشی را با روش گروه‌بندی حرف‌هایی که توسط ساندکس و فونیکس به‌کارگرفته‌شده، ترکیب می‌کند. بر اساس ادیتکس، مطابقت جفتی حاصل روش‌های آوایی هست و به روش‌های رشته‌ای مربوط نمی‌شود. از طریق ادیتکس می‌توان مطابقت بین حرف‌ها و تلفظ مشابه را دقیق‌تر نشان داد.

ج. شیوه‌های آوامتری[4] (Zobel and Dart 1996): شیوه‌های آوامتری مهارت‌های تطبیقی هستند که بر اساس مطالعات آوایی به وجود آمده‌اند. الگوریتم‌های تطابق آوامتری شامل دو مرحله هستند:

– مرحله اول: رشته‌ای از حرف‌ها به‌وسیله الگوریتم تبدیل رشته به تلفظ، به رشته‌ای از واج‌ها تبدیل‌می‌شود. چندین الگوریتم مناسب برای این منظور وجود دارد، ولی الگوریتم‌های وابسته به بافت موثرترین آنها می‌باشد.

– مرحله دوم: در این مرحله شباهت رشته‌ای بین واج‌ها اندازه‌گیری می‌شود. فاصله بین تلفظ‌ها که توسط رشته‌هایی از واج‌ها به وجود می‌آید، می‌تواند بسیار دقیق‌تر از فاصله بین رشته‌های حرفی اندازه‌گیری شود. بنابراین، انتظار می‌رود روش آوامتری بهترین تطبیق آوایی را با توجه به الگوریتم رشته به تلفظ ارائه دهد.

---

[1] Global regular expression print
[2] Boyer-Moore
[3] editex
[4] Phonometric

چ. کاهش تدریجی[1] (Zobel and Dart 1996): این روش، بهبودی برای مهارت‌های فاصله‌ی ویرایشی با درنظرگرفتن ویژگی‌های انسانی است. تفاوت‌های آوای آغازین تلفظ نسبت به تفاوت‌های آوای پایانی تلفظ از اهمیت بیشتری برخوردار هستند. در فاصله ویرایش کاهش‌یافته‌ی تدریجی، حداکثر جریمه[2] برای جایگزینی یا حذف در ابتدای رشته از حداقل دو برابر جریمه برای جایگزینی یا حذف در انتهای رشته بیشتر است.

ح. متافون[3] (Philips 1990): فرایند رمزگذاری متافون اصلی، تعداد ۱۶ کاراکتر صامت را به کار می‌برد: "0BFHJKLMNPRSTWXY" که عدد 0 نشان‌دهنده "the" است. حرف‌های مصوت AEIOU نیز فقط در ابتدای رمز به‌کاربرده‌می‌شوند. قوانین کلی متافون عبارتند از: حذف دو حرف تکراری مجاور، تبدیل حروف به یکدیگر، تبدیل توالی دو حرف یه یک حرف و حذف تمامی مصوت‌ها. به این خاطر که متافون اصلی خطاهای زیادی را دربرداشت، متافون دوبل جایگزین آن شد. در گام بعد، متافون سبل جایگزین دوبل شد که بسیاری از رمزگذاری‌های اشتباه که به‌وسیله دو نسخه‌ی قبلی به وجود داشت را تصحیح کرده است.

### ۲-۳. روش‌های ارزیابی سامانه‌های خطایاب و استانداردساز

برای ارزیابی، در مرحله نخست ۲۵ درصد داده‌ها را که به‌صورت انسانی تایید شده‌اند کنار گذاشته و از آنها جهت آزمایش سامانه‌ی خود استفاده می‌کنیم. معیار ارزیابی سامانه‌های استانداردسازی و خطایابی و تصحیح خطاها، دو مفهوم درست و نادرست است. این درست و نادرست را ممکن است سامانه به‌درستی تشخیص دهد و یا به غلط. این چهار مفهوم در جدول ۶ آمده است.

**جدول ٦ مفاهیم پایه ارزیابی سامانه‌های دسته‌بندی**

| (کلمه نادرست) False | (کلمه درست) True |
|---|---|
| کلمه نادرست:fp | کلمه درست:tp |
| به اشتباه کلمه نادرست:fn | به اشتباه کلمه درست :tn |

[1] tapering
[2] penalty
[3] metaphone

با استفاده از چهار مفهوم جدول ۶، دقت، صحت و بازخوانی سامانه از طریق فرمول ۶، ۷ و ۸ محاسبه می‌شود:

$$(6)\ \boldsymbol{a} = \frac{tp + tn}{tp + tn + fp + fn} \qquad (7)\ \boldsymbol{p} = \frac{tp}{tp + fp} \qquad (8)\ \boldsymbol{r} = \frac{tp}{tp + fn}$$

تفاوت دقت و صحت را چنین می‌شود در نظر گرفت که در سنجش دقت، سامانه میزان نزدیکی به هدف را نشان می‌دهد و در سنجش صحت سامانه میزان نزدیکی جواب‌های درست را نشان می‌دهد. معیار اف سامانه از طریق فرمول ۹ محاسبه می‌شود. میانگین هارمونیک (از انواع میانگین‌های فیثاغورثی) صحت و فراخوانی از طریق این معیار محاسبه می‌شود.

$$(9)\ \ \boldsymbol{f - measure} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

معیار دیگر ارزیابی رتبه‌ی وارانه‌ای میانگین[1] است که اغلب در سامانه‌های پرسش و پاسخ به کار می‌رود (Radev et al. 2002). در این رتبه‌بندی، فرایند تولید جواب‌های ممکن به یک سوال (خطای املایی) ارزیابی می‌شود. درستی جوابها بر اساس احتمالات آنها مرتب می‌شود. این معیار از طریق فرمول ۱۰ محاسبه می‌شود.

$$(10)\quad \boldsymbol{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

در این فرمول، تعداد سوالها Q و رتبه‌ی جواب درست rank است.

### ۲-٤. معرفی سامانه‌های تجاری غیرفارسی

سامانه‌های غیرفارسی بسیاری وجود دارد که بعضی از آنها به صورت نرم‌افزار و برخی دیگر به صورت افزونه هستند.

#### الف. نرم‌افزارهای شناخته‌شده

[1] Mean Reciprocal Rank

نرم‌افزارهای بسیاری برای خطایابی املایی در متن وجود دارد. از برجسته‌ترین و شناخته‌شده‌ترین نرم‌افزارهای جهان مایکروسافت وورد[1] (وورد) است. این مجموعه بر اساس الگوریتم وابسته به بافت به خطایابی املایی و دستوری با صحت به‌مراتب بالا و بازخوانی کم می‌پردازد. خطایاب بعدی، خطایاب گوگل است که بر روی مرورگر نصب شده و در تمامی نوشته‌ها کاربر می‌تواند بر حسب انتخاب به صورت وابسته به بافت به کار رود. این خطایاب از الگوریتم دامرو لونشتاین برای تصحیح خطا استفاده می‌کند. در نهایت، سومین نرم‌افزار معتبر جهان جینجر است که توسط موآز شخت[2] طراحی گردیده و بر حسب الگوریتم وابسته به بافت به غلط‌یابی املایی و دستوری می‌پردازد. ایراد بارز این نرم‌افزار، تصحیح خودکار خطاهاست که در موارد بسیاری می‌تواند آسیب جدی به مفهوم متن وارد کند.

**ب. افزونه‌های شناخته‌شده**

**۱. آی‌اسپل[3]:** آی‌اسپل یک غلط‌یاب املایی برای یونیکس[4] است که توسط «گورین»[5] طراحی شده است. آی‌اسپل با استفاده از الگوریتم دامرو لونشتاین و فرهنگ‌لغت موبی ووردز[6] به اصلاح خطاهای املایی می‌پردازد. این فرهنگ‌لغت دارای حدود ۱۷۷۲۶۷ مدخل به همراه تلفظ آنها است. تنها ایراد این افزونه، نبود کلمات کافی و حالت‌های املایی عموما پذیرفته شده و همچنین فرایند بافت‌محور است.

**۲. اِی‌اسپل[7]:** اِی‌اسپل یکی از افزونه‌های تحت لیسانس گنو[8] است که برای جایگزینی آی‌اسپل توسط کوین آتکینسون[9] نوشته شده است. تفاوت این افزونه با آی‌اسپل در این است که اِی‌اسپل برای کلمه‌های غلط، گزینه‌های منتخب بیشتری را برای کاربر پیشنهاد می‌دهد. برای مثال، زمانی که به کلمه "trubble" می‌رسیم، آی‌اسپل فقط کلمه rubble را پیشنهاد می‌دهد، ولی اِی‌اسپل

[1] Microsoft Word
[2] Maoz Shacht
[3] Ispell
[4] Unix
[5] Gorin
[6] Moby Words
[7] Aspell
[8] GNU
[9] Kevin Atkinson

گزینه‌های منتخب دیگری نظیر trouble، dribble، rubble و غیره را پیشنهاد می‌دهد. این افزونه قابلیت تحلیل مستندات یوتی‌اف۸[1] را نیز دارد. این ابزار مجهز به ماژول غلط‌یابی املایی بر اساس حرف و آوا می‌باشد و حافظه بیشتری را نسبت به آی‌اسپل اشغال می‌کند.

**۳. مای‌اسپل:** ماژولی است که قبلا بر روی او او-رایتر[2] نرم‌افزار OpenOffice.org کار می‌کرد. نویسنده این ماژول، کوین هندریکس[3]، روش‌های نسخه‌های لایه‌باز خطایاب‌های دیگر نظیر آی‌اسپل را با این افزونه ادغام کرده و مای‌اسپل را ساخته است. الگوریتم این افزونه ترکیبی از الگوریتم آی‌اسپل و متافون است. مای‌اسپل دارای فضا[4] است که این فضا نیز دارای فایل‌هایی برای املا و فرهنگ‌لغت است. به این دلیل که از انباشته شدن حالت‌های صرفی مختلف برای واژه جلوگیری شود، فایل با فرمت dic. واژگان به همراه رابط[5] آن به وندها در فایل با فرمت aff. ساخته می‌شود. تنها ایراد این ماژول، کندی عملکرد دقیق بخش پردازش صرفی آن است.

**٤. هانسپل[6]:** هانسپل، نوع پیشرفته افزونه‌ی مای‌اسپل است که علاوه بر غلط‌یابی املایی، تحلیل صرفی واژگان را نیز بر روی زبان‌های مجاری انجام می‌دهد. این افزونه برای نخستین بار توسط «ترو» جهت پردازش سریع‌تر وندهای غیروابسته با زبان (اوکامورف[7]) در کنفرانس ۴۷م ای‌سی‌ال[8] ۲۰۱۵ پیشنهاد شده است (Tro et al. 2015). هانسپل دارای اوکامورف (وندزدایی غیروابسته به زبان)، مورف‌دی‌بی (پایگاه داده لغوی و دستور صرفی که توسط اوکامورف مورد استفاده قرار می‌گیرد) و هانلکس (کامپایلر[9] مدیریت منبع است که کارآیی الگوریتم را از نظر زمانی و فضایی بهبود می‌بخشد) است. این ماژول با وجود قدرت خوب، در خطوط طولانی دارای دقت پایینی است.

---

[1] utf-8
[2] OOo Writer
[3] Kevin Hendricks
[4] Locale
[5] flag
[6] Hunspell
[7] Ocamorph
[8] کنفرانس ACL بزرگ‌ترین و مهم‌ترین کنفرانس زبان‌شناسی رایانشی در جهان است.
[9] Compiler

**۵. جزی**[1]: افزونه‌ای است که به زبان برنامه‌نویسی جاوا نوشته شده است. این افزونه می‌تواند برای غلط‌یابی املایی استفاده شود. طراح این ماژول، «ایزدلیس میندوگس»[2] است. جزی از لحاظ عملکرد بسیار شبیه به اِی‌اسپل است با این تفاوت که در برخی موارد، به تک‌تک کاراکترها پیشنهاد املایی انجام می‌دهد.

### ۳-۱. تشریح پژوهش‌های انجام‌گرفته بر روی زبان فارسی

#### ۳-۱-۱. زبان فارسی

زبان فارسی یکی از زبان‌های هندواروپایی است که در کشورهای ایران، افغانستان، تاجیکستان و ازبکستان به عنوان زبان رسمی کشور شناخته شده است. زبان فارسی دارای ۳۲ کاراکتر الفبایی هست که در مقایسه با زبان عربی دارای چهار حرف (گ، چ، پ و ژ) بیشتر است. در نوشتار فارسی میان شکل نوشتاری برخی نویسه‌های آغازین، میانی و پایانی تفاوت وجود دارد مانند بـ ـبـ ـب.

#### ۳-۱-۲. چالش‌های پردازش املایی و استانداردسازی زبان فارسی

چالش‌های زبان فارسی را با اقتباس از (مگردومیان، ۲۰۰۴؛ رسولی و مینایی، ۲۰۰۸؛ شمس‌فرد و همکاران، ۲۰۱۰؛ کاشفی و همکاران، ۲۰۱۰؛ فیلی و همکاران، ۲۰۱۴) چنین طبقه‌بندی می‌کنیم:

الف. هم‌آوایی: ویژگی حروفی است که دارای آوای مشترک ولی املای متفاوت باشند. مانند: "ذ" و "ظ" و "ز" و "ض" که همگی دارای آوای [z] هستند.

ب. تاثیر حروف عربی بر متون زبان فارسی: دو آوای تنوین (ً) و همزه (ء) که از زبان عربی وارد زبان فارسی شده‌اند، همواره املای کلمات را به چندین شکل مختلف درآورده‌اند. مانند "پاییز" و "پائیز" و یا مانند "مساله" و "مسئله" و "مسأله". در مثال تنوین می‌توان به املای متفاوت "حتما" و "حتماً" اشاره کرد.

---

[1] Jazzy
[2] Mindaugas Idzelis

پ. ابهام یونی‌کد: برای برخی از حروف مانند دو حرف "ی" و "ک" ابهام یونی‌کد وجود دارد.به همین خاطر، نرم‌افزارهای مختلف حالت‌های مختلف (سلیقه‌ای) آن را در نظر می‌گیرند.

ت. چندامﻻیی بودن: شیوه‌های مختلف املایی درست که هر چند سال یکبار از سوی فرهنگستان زبان و ادب فارسی اعلام می‌شود، دارای پایداری نیست. به عنوان مثال، بعضی کلمات دارای دو حالت املایی درست هستند مانند "بلیت" و "بلیط".

ث. فاصله‌گذاری: فاصله‌گذاری (فاصله‌بندی) در زبان فارسی دارای قاعده‌ی قطعی نیست. زیرا گاها فاصله به نیم‌فاصله مانند "می‌خورد، گاهی به تمام فاصله مانند "می خورد" و گاهی به فاصله‌ی صفر (سرهم) مانند "میخورد" تبدیل می‌شود. در مثالی دیگر، بسیاری از فارسی‌زبانان بعد از حروفی که فقط دارای شکل پایانی هستند فاصله به کار نمی‌برند مانند "شیرراخوردم"

ج. شیوه‌ی نویسه‌گردانی: نویسه‌گردانی از زبان‌های دیگر به زبان فارسی دارای حالت‌های مختلف (سلیقه‌ای) است.

چ. دیدگاه صرفی: زبان فارسی دارای سلیقه‌های اعمال فرایندهای اشتقاق و تصریف است که به موارد برجسته‌ی آن اشاره می‌کنیم.

- زبان فارسی زبانی اشتقاقی و زایشی است و ممکن است همواره با ترکیب واژه‌ها و وندها کلمات نو تولید شوند.

- فعل‌سازی در زبان فارسی امری بدون قاعده است و همواره با ترکیب اسم‌ها و صفت‌ها تولید می‌شود.

- وجود انواع جمع برای اسم در زبان فارسی نیز مشکل‌ساز است. مانند اضافه کردن وندهای "ان" و "ون" و "ین" و "ها" به آخر کلمه و یا تبدیل اسم به جمع کسر عربی آن مانند تبدیل "حادثه" به "حوادث". همچنین درج برخی حرف (آوا) های اضافه هنگام جمع بستن کلمه‌ها مانند درج "ج" و حذف "ه" در تبدیل "کارخانه" و "کارخانجات" و یا درج "ی" در تبدیل "دانشجو" به "دانشجویان".

- وجود واژه‌بست (تکواژی که ویژگی نحوی یک واژه را دارد ولی از لحاظ آوایی به واژه‌ی دیگر وابسته است و به آن می‌چسبد) در زبان فارسی در بسیاری موارد موجب ناتوان کردن خطایاب می‌شود. مانند "کتابشان" که "ش" واژه‌بست و "ان" علامت جمع است که ممکن است با کلمه آبشان یکی در نظر گرفته شود و خطا محسوب شود.

ح. دیدگاه نحوی: پردازش رایانه‌ای ساخت اضافه در زبان فارسی دارای مشکلی بس بزرگ است که تا به حال حل نشده است. زیرا ساخت اضافه دارای یک کسره اضافه است که تنها در مواقعی نوشته می‌شود که به صداهای بلند ختم شود و به غیر از این مواقع اصلا نوشته نمی‌شود. مانند "کتاب مبینا" و "خانه‌ی". همچنین بین نوشتن "ی" و "ء" و ننوشتن آن بین زبان‌شناس‌ها اتفاق نظر وجود ندارد. همچنین می‌توان به عدم تطابق اجزای جمله (نهاد و گزاره از لحاظ شمار) اشاره کرد مانند "آقای مدیر آمدند" و "برگ‌ها می‌ریزد". این نوع عدم تطابق همیشه درست است ولی بسته به بافتار خاص استفاده می‌شود.

خ. پیوستگی حروفی: علاوه بر موارد فوق، زبان فارسی دارای خط پیوسته[1] است و ممکن است همین امر باعث شود که عملیات تقطیع با مشکلاتی مواجه شود.

د. کاربرد اعراب (هم‌نگاره بودن): عدم بکارگیری اعراب در متون فارسی می‌تواند فرایند خطایابی رو دچار مشکل کند مانند کلمه مردم که نبود اعراب منجر می‌شود به سه شکل مختلف خوانده شود: مَردَم، مَردُم و مُردَم

**۳-۱-۳. مروری بر برجسته‌ترین پروژه‌های خطایابی متون عمومی زبان فارسی**

در پژوهش‌های پردازشی زبان فارسی، پیش‌پردازش استانداردسازی متنی داده‌ها یا به صورت مستقل تولید شده و به صورت جعبه‌ابزار وجود دارد و یا ماژولی از عملیات بزرگتر است. اما قبل از استانداردسازی نیز، مرحله‌ی بسیار کلیدی تقطیع متن به توکن وجود دارد.

مگردومیان (۲۰۰۴) در دو مرحله‌ی تقطیع اولیه (غیر وابسته با زبان) و پساتقطیع (وابسته به زبان فارسی) به تقطیع کلی متن پرداخته است. در مرحله پیش‌تقطیع، سیستم آنها به صورت پایه‌ای

[1] cursive script

کاراکترها را تقطیع می‌کند. معیار کاراکتر پایه، شکل پایانی حرف و فاصله‌ی کامل است. بعضی از حرف‌ها شکل پایانی ندارند (مانند د) و فرایند تقطیع را دشوارتر می‌کنند. در این صورت، الگوریتم بعد از هر کدام از این کاراکترها فاصله‌ای اعمال کرده و سعی می‌کند چندین شکل مختلف ایجاد کند و سپس بهترین شکل را با اعمال تحلیل صرفی (ستاک‌بندی) از فرهنگ لغت برگزیند و بر این اساس تقطیع کند. در مرحله‌ی پساتقطیع (بعد از تقطیع اولیه)، الگوریتم هر دو کلمه‌ی تقطیع‌شده را در بین وندها جستجو می‌کند.

براری و قاسمی‌زاده (۲۰۰۵) خطایاب کلونایزر[1] غیروابسته به زبان را بر اساس ساختار داده‌های درختی سه‌تایی[2] در حالت تطبیق‌پذیری معرفی کردند. الگوریتم براری و قاسمی‌زاده یادگیری الگوی خطا را از طریق رفتار کاربر یاد می‌گیرد. بخش‌های مختلف کلونایزر عبارتند از: ماژول خطایاب (پردازش با الگوریتم دامرو لونشتاین)، واژگان (دارای ساختار درختی سه‌تایی)، هزینه‌ی انتقال (تغییر یا ویرایش)، ماژول یادگیری و تطبیق‌پذیری که به واسطه‌ی انتخاب کاربر الگوی خطارا استخراج می‌کند. ویژگی مهم این کلونایزر استفاده از حدآستانه‌ای محلی و جهانی است. حدآستانه‌ای محلی عمق جستجو را بر اساس طول کلمه‌های پیشنهادشده محدود می‌کند و حدآستانه‌ای جهانی مانع تولید کلمه‌های نامناسب بر اساس طول رشته‌ی ورودی و تعداد کلمه‌های پیشنهادی می‌شود. براری و قاسمی‌زاده برای ارزیابی الگوریتم خود مجموعه دادگان ۵۵۹۵ کلمه‌ای از متن املای دانش‌آموزان و مراکز تایپ با دو شکل صحیح و غلط را جمع‌آوری کردند[3]. دقت آزمایش الگوریتم آنها بر روی این مجموعه داده، حدود ۸۵ درصد نتیجه‌شده‌است.

رسولی و مینایی (۲۰۰۸) عمل خطایابی را با استفاده از مقوله‌ی نحوی کلمه در متن کاربر انجام دادند که قابل پیاده‌سازی بر روی مایکروسافت وورد است. روش خطایابی آنها بدین‌ترتیب است: تشخیص مقوله‌ی نحوی بر اساس ساختار کلمه، بررسی صحت مقوله‌ی نحوی از فرهنگ لغت و ارائه پیشنهادهای کاندیدهای املایی برای کلمه‌ی غلط در صورت تایید عدم صحت از مرحله‌ی قبلی. در این الگوریتم به این دلیل که حجم عملیات پردازش پیشنهادهای املایی برای تک‌تک واژه‌های متن بسیار بالا است، کلماتی که دارای بسامد وقوع بیش از یک‌بار باشند فقط یکبار پردازش می‌شوند. آزمایش‌های رسولی و مینایی نشان داده که به این طریق حجم پردازش حدود

[1] Clonizer
[2] Ternary
[3] http://www.digitalclone.net/localization/spellchecker

۴/۵ برابر کم می‌شود. رسولی و مینایی برای وزن‌دهی به پیشنهادها، این شاخص‌ها را درنظرگرفته‌اند: پیشنهادهای موجود در متن و دارای بسامد بیشتر، پیشنهادهایی که مقوله‌ی نحوی آنها با مقوله‌ی نحوی کلمه برابر باشد، پایان حروفی که فقط دارای شکل پایانی هستند، فشردن کلیدهای کناری، نزدن و یا زدن یک کلید، جابجایی دو حرف مجاور، واژگان درست کاربر و مدل چندنگاشتی آن و در نهایت نوع خطاهای کاربر. با توجه به شاخص‌های فوق، می‌توان وزن پیشنهادهارا در فرمول ۹ نشان داد:

$$(9)\ w_s = \sum_{i=1}^{7} \alpha_i * \lambda_i$$

در این فرمول، $\lambda$ شاخص و $\alpha$ ضریب شاخص است.

شمس‌فرد و همکاران در استپ وان[1] (۲۰۱۰) الگوریتم مگردومیان را توسعه دادند و چالش‌های اصلی استانداردسازی زبان فارسی را تصحیح نموده‌اند. شمس‌فرد و همکاران خطایاب جدیدی را پیشنهاد کردند. این خطایاب بر خلاف خطایاب‌های قبلی، به موقعیت کلمه در جمله و بافت کلمه توجه می‌کند و وب را به عنوان پیکره در نظر می‌گیرد. در گام نخست، خطایاب کاندیداهای احتمالی را از بانک داده ۹۰۰۰۰۰ کلمه‌ای همشهری به روش ساندکس استخراج می‌کند و سپس کلماتی که کد ساندکس آن با کلمات بانک داده شبیه بوده و دارای بسامد بالای ۱۰۰ باشد، انتخاب می‌شوند. معیار مقایسه‌ی نتایج جستجوی خروجی ساندکس، به واسطه‌ی فرمول ۱۰ از موتور جستجوی گوگل محاسبه می‌شود.

$$(10)\ P = 1 + \frac{1}{9}\log\left(\frac{\#occ\,(T_p.Cs.T_n)+1}{10^9}\right)$$

در این فرمول، Cs کاندیداهای پیشنهادی و $T_p$ کلمه‌ی ماقبل و $T_n$ کلمه‌ی مابعد است. هر دوی این سیستم‌ها به صورت قاعده‌بنیاد عمل می‌کنند. خطایاب شمس‌فرد و همکاران، روی سه متن (رمان، تاریخ و عمومی) مختلف که هر کدام داری ۵۰۰ تا ۶۰۰ کلمه و در مجموع دارای ۱۰۰ غلط املایی بودند آزمایش شد. نتایج این آزمایش در شکل ۱۵ قابل مشاهده می‌باشد. در نهایت، ۹۹/۶ درصد خطاها تشخیص داده شده‌اند و ۹۲/۶ درصد از آنها تصحیح شده‌اند.

[1] STeP-1: http://step1.nlplab.sbu.ac.ir/

نوگورانی و صبوریان (۱۳۸۵) برای کاهش چشمگیر حجم واژگان، خطایابی را مبتنی بر تحلیل صرفی واژه‌ها و ستاک‌یابی آن انجام داده‌اند. همچنین، آنها از روش اتوماتای کمینه‌ی متناهی و قطعی بدون دور[1] در ماشین میلی[2] برای فشرده‌سازی واژگان استفاده کرده‌اند که حجم واژگان را ۳ تا ۶ برابر نسبت به فشرده‌سازی آی‌اسپل کاهش داده است. تمامی گذار[3]های این اتوماتا در آرایه[4]ای ذخیره می‌شوند که که هر عضو نماینده‌ی یک گذار و هر چند گذار نماینده‌ی یک حالت[5] باشد. بدین‌ترتیب، کلمه اول به ستاک تبدیل می‌شود و از فهرست واژگان (واژگان ای‌اسپل فارسی: واژگان آن به صورت ستاک ذخیره شده‌اند.) جستجو می‌شود. واژگان دارای وزن متداول و مهجور بر اساس انتخاب کاربران از میان پیشنهادها شده و همچنین بر اساس برچسب نحوی فعل و صفت و اسم مرتب می‌شوند. روش مورد استفاده در تصحیح خطا، روش معکوس حداقل فاصله‌ی ویرایشی[6] است. در این روش، تغییرات ویرایشی زیر به صورت تک‌به‌تک بر واژه‌ی نادرست اعمال می‌شود و به محض اینکه به واژه‌ی درست منجر شد، عملیات اعمال تغییرات به اتمام می‌رسد.

ویراستیار[7] (کاشفی و همکاران، ۲۰۱۰) افزونه‌ای برای مایکروسافت وُرد است که برای استفادهٔ کاربران فارسی‌زبان طراحی شده است. الگوریتم دقیق و جزئی این سامانه تشریح نشده و تنها گزارش قابلیت‌های اصلی نرم افزار موجود است که مهم‌ترین آنها عبارتند از: غلط‌یاب املایی، اصلاح نویسه‌های متن، اصلاح نشانه‌گذاری، تبدیل تقویم و تاریخ، تبدیل پینگلیش، پیش‌پردازش املایی متن، تبدیل اعداد و اصلاح نیم‌فاصله

در ادامه این مسیر سراجی (۲۰۱۲) پری‌پر[8] را برای زبان فارسی طراحی کرد. این سامانه که با ماژول ویراستار (برقی، ۲۰۱۱) در هم‌آمیخته، متن را برای پردازش‌های لازم در عملیات‌های زبان‌شناسی رایانشی آماده می‌کند. ویژگی‌ها پری‌پر عبارتند از: نرمال‌سازی متن، تبدیل سبک نوشتاری عربی به فارسی، اصلاح نیم‌فاصله و غیره.

---

[1] Minimal Acyclic Deterministic Finite Automata
[2] Mealy Machine
[3] transition
[4] array
[5] state
[6] Reverse minimum edit distance
[7] http://virastyar.ir/virastlive/index.html
[8] PrePer

فیلی و همکاران (۲۰۱۴) سامانه‌ی خطایاب وفا[1] را برای زبان فارسی معرفی کردند. این سامانه‌ی هیبرید[2] به صورت قاعده‌بنیاد و آماری به تشخیص و تصحیح خطاها می‌پردازد. واژگان پایه‌ی سامانه، ترکیبی از واژگان تصریف‌شده‌ی فرهنگ دهخدا (دهخدا، ۱۹۹۸) به همراه میزان بسامد آنها با استفاده از دو پیکره‌ی خبرنامه‌های ایرنا و همشهری و همچنین تعداد ۵۰۰۰ اسامی پربسامد از سامانه‌ی آموزشی است. در نهایت واژگان سامانه تعداد ۱۲۰۰۰۰۰ واژه را در خود جای داده است. الگوریتم به کار رفته در این سامانه، الگوریتم دامرو لونشتاین است. همچنین، به صورت اکتشافی چند ویژگی برای خطاهای تایپی با احتمال وقوع بیشتر استخراج شده‌اند: جایگزینی و درج در حروف همجوار بر روی صفحه‌کلید، حذف در حروف تایپ شده به وسیله‌ی انگشت کوچک، حروف تایپ شده به‌وسیله‌ی دو دست مختلف (چپ و راست)، حروف هم‌نویسه، تکرار حروف و جایگزینی واج‌های /س، ص و ث/

پس از تشخیص خطا توسط الگوریتم، فاصله‌بندی کلمه بررسی می‌شود. در فاصله‌بندی، توکن نخست به دو بخش چپ و راست تقسیم می‌شود. این فرضیه در سه حالت(بین دو بخش، بخش چپ و توکن مابعد چپ، بخش راست و توکن ماقبل راست)، قرار گرفته و خطا در ۷ حالت ممکن نظیر چسبادندن دو واژه، حذف و یا افزودن فاصله به کلمات مابعد و ماقبل می‌توانند تصحیح شوند. سامانه‌ی وفا از فاصله‌ی ویرایشی کمینه استفاده می‌کند که سه ویژگی آوایی، شباهت ظاهری و تاثیرهای صفحه‌کلید را ملاحظه می‌کند. در نهایت، از حالت‌های اکتشافی[3] استفاده کرده و لیست کاندیداهای درست را تولید می‌کند. پس از تولید این لیست، چسبیدن کلمه به کلمه‌ی ماقبل و مابعد نیز بر طبق ۷ حالت گفته‌شده بررسی می‌شود. داده‌های لازم جهت ارزیابی وفا از ۳۴۰ وبلاگ، جملاتی با طول ۱۴/۰۵ و با ۳۱۲ غلط املایی(۴۹ حذف، ۱۵ درج، ۶۱ جایگزینی و ۱۴ جابجایی، ۱۴۲ سرهم‌نویسی و ۳۱ خطای نیم‌فاصله) استخراج شده و حالت درست آن به صورت انسانی تهیه شده است. علاوه بر این سامانه، دو سامانه‌ی ویراستیار و مایکروسافت نیز بر روی همین ارزیابی شده که نتایج رتبه‌ی وارانه‌ای میانگین آنها به ترتیب ۰/۸۴۴، ۰/۸۲۸ و ۰/۸۱۵ بدست آمده است.

[1] http://www.vafaspellchecker.ir/
[2] Hybrid
[3] heuristic

**٤- جمع‌بندی و بحث**

در این پژوهش انواع دادگان و روشهای رایانشی و پردازشی برای فرایند خطایابی املایی و استانداردسازی متنی در قالب طبقه‌های مختلف گنجانده شده و به تفصیل مورد بحث قرار گرفت. پس از طبقه‌بندی مواد و روش‌ها، مروری بر نرم‌افزارها و افزونه‌های قدرتمند جهان انجام گرفت. در پایان، چالش‌های زبان فارسی از دل تحقیقات مختلف انجام‌گرفته در مورد خطایابی فارسی استخراج شده و در ده طبقه‌ی مجزا معرفی گردیدند. همچنین، هشت تحقیق بسیار مهم و برجسته‌ی تشخیص خطاهای املایی زبان فارسی و استانداردسازی آن از حیث مواد و روش‌ها و نتایج و ارزیابی آنها به تفضیل تشریح شد.

مطالعه‌ی حاضر نشان داد که بسیاری از چالش‌های موجود در زبان فارسی به طور کامل حل نشده و زمینه‌ی تحقیق در مورد خطایابی و استانداردسازی متنی همچنان ادامه خواهد داشت. بخشی از این چالش‌ها با از میان‌بردن اختلاف‌نظرهای متخصصین مطالعات نظری زبانی حل‌می‌شود و به تبع آن نوشتار فارسی هرچه بیشتر یکنواخت می‌گردد. این یکنواختی مهندسین زبان را نیز از اسیری در سردرگمی‌های مدام نجات می‌دهد. هرچند، هیچ زبانی در طول تاریخ همیشه یک شکل واحد نداشته و همواره دستخوش تغییرات مکرر است. بنابراین، روش‌ها و الگوریتم‌های مهندسین زبان نیز باید به اندازه‌ی کافی قدرت یادگیری و عملکرد بالایی داشته‌باشد.

با افزایش آمار دانشجویان و همچنین گسترش انجام پژوهش و تحقیق، تولید مستندات علمی روزبه‌روز افزایش می‌یابد. مقاله‌ی حاضر نشان داد که اکثر افزونه‌ها و نرم‌افزارهای فارسی برای متون عمومی طراحی شده و در مورد متون تخصصی و علمی چندان آزمایش نشده‌اند. پس این خلا پژوهشی باید در مجموعه پژوهش‌های انجام‌شده برای زبان فارسی رفع گردد.

**٥- یادداشت**

مقاله‌ی حاضر برگرفته از طرح پژوهشی « طراحی و ساخت سامانه‌ی استانداردساز و خطایاب متون فارسی » است که در پژوهشگاه علوم و فناوری اطلاعات ایران در حال انجام است.

## فهرست منابع

دری نوگرانی، صادق و صبوریان، محسن. ۱۳۸۵. طراحی و پیاده‌سازی یک خطایاب فارسی. *دومین کارگاه پژوهشی زبان فارسی و رایانه*، دانشکده ادبیات دانشگاه تهران

رسولی، صادق ومینایی بیدگلی، بهروز. ۱۳۸۷. روشی جدید در خطایابی املایی فارسی. *دومین کنفرانس داده‌کاوی*، دانشگاه امیرکبیر، تهران

کاشفی، امید و نصری، میترا و کنعانی، کامیار. ۱۳۸۹. *خطایابی املایی خودکار در زبان فارسی*. دبیرخانه شورای عالی اطلاع‌رسانی

- Bargi, A. A., 2011. Virastar. https://github.com/aziz/virastar.

- Barari, L. and QasemiZadeh, B. 2005. Clonizer spell checker adaptive, language independent spell checker. In Proceedings of the first ICGST International Conference on Artificial Intelligence and Machine Learning AIML, Cairo, Egypt, pp. 19–21.

- Beliga, S., Pobar, M. and Martinčić-Ipšić, S. 2015. Normalization of Non-Standard Words in Croatian Texts. MIPRO CIS - Intelligent Systems, Opatija, Croatia

- Bhatti, Z., Ismaili, I. A., Soomro, W. J. and Hakro, D. N. 2014. Word Segmentation Model for Sindhi Text. American Journal of Computing Research Repositor, 2(1), 1-7

- Chen, B. 2009. Word Topic Models for Spoken Document Retrieval and Transcription. ACM Transactions on Asian Language Information Processing, Vol. 8, No. 1, pp. 2:1-2:27.

- Cortes, C. and Vapnik, V. 1995. Support-vector networks. Machine Learning, 20:273-297.

- Damerau, F. J., 1964. A technique for computer detection and correction of spelling errors. ACM, Vol. 7, pp. 171-176.

- Dehkhoda, A. 1998. Loghatnameye Dehkhoda (Dehkhoda's dictionary), Vol. 12. Tehran: Tehran University Publications.

- Feili, H., Ehsan, N., Montazery, M., Pilehvar, M. T. 2014. Vafa spell-checker for detection spelling, grammatical, and real-word errors of Persian language, Literary and Linguistic Computing Advance Access published September.

- Gadd, T. 1990. PHONIX: The algorithm. Program automatedlibrary and information systems, 24(4):363-366.

- Golding, A. R. 1995. A Bayesian hybrid method for context-sensitive spelling correction. In Proc. 3rd Workshop on Very Large Corpora, Boston, MA.

- Hall, P.A.V. and Dowling, G.R. 1980. Approximate string matching. ACM Comput. Surv, 12(4) 381–402.

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

- Hirschberg, D. S. 1975. A linear space algorithm for computing maximal common subsequences. Communications of the ACM, 18 (6): 341–343

- Jaro, M. A. 1989. Advances in record linkage methodology as applied to the 1985 census of Tampa Florida. Journal of the American Statistical Association, 84 (406): 414–20

- Jurafsky, D., and James H. M. 2009. Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall.

- Levenshtein, V. 1965. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady, Vol. 10, pp. 707-710.

- Liu, C., Lai, M.,Tien, K., Chuang, Y., Wu, S. and Lee, C. 2011. Visually and Phonologically Similar Characters in Incorrect Chinese Words: Analyses, Identification, and Applications. ACM Transactions on Asian Language Information Processing, Vol. 10, No. 2, pp. 1-39.

- Liu, F., Weng, F. and Jiang, X. 2012. A Broad-Coverage Normalization System for Social Media Languag. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012).

- Megerdoomian, K. 2004. Finite-state morphological analysis of Persian. In Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. University of Geneva: Association for Computational Linguistics, pp. 35–41.

- Muth, F., and Tharp, A. L. 1977. Correcting human error in alphanumeric terminal input. Inform. Processing and Mgmt.

- Needleman, S. B. and Wunsch, C D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 48 (3): 443–53

- Philips, L.1990. Hanging on the Metaphone. Computer Language, Vol. 7, No. 12.

- Radev, D. R., Qi, H., Wu, H., Fan, W. 2002. Evaluating web-based question answering systems. Proceedings of LREC.

- Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. Information Processing and Management, 24(5):513–523.

- Schaback, J. and Li, F. 2007. Multi-Level Feature Extraction for Spelling Correction. Workshop on Analytics for Noisy Unstructured Text Data, IJCAI-07, Hyderabad, India.

- Schlippe, T., Zhu, C., Gebhardt, J. and Schultz, T. 2010. Text Normalization based on Statistical Machine Translation and Internet User Support. The 11th Annual Conference of the International Speech Communication Association (Interspeech 2010), Makuhari, Japan.

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

- Shannon, C. 1948. A mathematical theory of communication. Bell System Technical Journal, 27(3): 379-423.

- Smith, T. F. and Waterman, M. S. 1981. Identification of Common Molecular Subsequences. Journal of Molecular Biology, 147: 195–197.

- Tró, V., Gyepesi, G., Halácsy, P., Kornai, A., Németh, L. and Varga, D. 2015. Hunmorph: Open Source Word Analysis. ACL, Beijing, China.

- Ukkonen, E. 1992. Approximate string-matching with qgrams and maximal matches. Theoretical Computer Science, 2:191-211.

- Wagner, R. A. and Fischer, M. J. 1974. The String-to-String Correction Problem. J. ACM, Vol. 21, pp. 168-173.

- Winkler, W. E. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods (American Statistical Association), 354–359.

- Wu, S., Chen, Y., Yang, P., Ku, T., and Liu, Chao-Lin. 2010. Reducing the False Alarm Rate of Chinese Character Error Detection and Correction. In Proceedings of SIGHAN.

- Wu, S. and Manber, U. 1992. Fast text searching allowing errors. Communications of the ACM, 35(10):83-91.

- Zampieri, M. and Cordeiro de Amorim, R., 2013. Effective Spell Checking Methods Using Clustering Algorithms. Recent Advances in Natural Language Processing. Hissar, Bulgaria.

- Zobel, J. and Dart, P. 1996. Fnetik: An integrated system for phonetic matching. Technical Report 96-6, Department of Computer Science, RMIT

# Categorization of Various Essential Datasets and Methods for Textual Spelling Detection and Normalization

Hadi Abdi Ghavidel

Msc. Graduate in Computational Linguistics, Sharif University of Technology, Tehran, Iran

e-mail: habdi.cnlp@gmail.com

Molouk Sadat Hosseini Beheshti

PhD in General Linguistics; Assistant Professor; Iranian Research Institute for Information Science and Technology(IranDoc); Tehran, Iran

e-mail: beheshti@irandoc.ac.ir

**Abstract**

One of the most primary phases of automatic text processing is spelling error detection and grapheme normalization. Storing textual documents faces several problems without passing this phase, which causes a disturbance in retrieving the documents automatically. Therefore, specialists in the fields of natural language processing and computational linguistics usually make an attempt to sample various data through presenting ideal methods and

algorithms in order to reach the normalized data. Several researches have been conducted on English and some other languages, which have been followed by a certain amount of researches on Farsi too. Sometimes, these several researches have remained to be a pure study and sometimes they have been released as a product. This paper carries out the categorization of the different methods and essential datasets in these researches and depicts each category individually and the evaluation measurements methods generally. Moreover, it describes the performance of the monolingual Farsi systems and the way they meet the Farsi challenges.

**Keywords**: spelling error detection, grapheme normalization, categorization of the methods, monolingual Farsi systems, Farsi language challenges