

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

Molouk Sadat Hosseini Beheshti

PhD in General Linguistics; Assistant Professor; Iranian Research Institute for Information Science and Technology (IranDoc); Tehran, Iran;
Corresponding Author beheshti@irandoc.ac.ir

Hadi Abdi Ghavideh

Msc. Graduate in Computational Linguistics; Sharif University of Technology; Tehran, Iran habdi.cnlp@gmail.com

Received: 26, Jan. 2016 Accepted: 2, Aug. 2016

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

**Iranian Journal of
Information
Processing and
Management**

**Iranian Research Institute
for Science and Technology**

ISSN 2251-8223

eISSN 2251-8231

Indexed by SCOPUS, ISC, & LISTA

Vol. 32 | No. 4 | pp. 1143-1170

Summer 2017



طبقه‌بندی انواع دادگان مورد نیاز و روش‌های خطایابی و استانداردسازی متنی

ملوک‌السادات حسینی بهشتی

دکتری زبان‌شناسی همگانی؛ استادیار؛
پژوهشگاه علوم و فناوری اطلاعات ایران (ایرانداک)؛
پدیدآور رابط beheshti@irandoc.ac.ir

هادی عبدی قویدل

کارشناسی ارشد زبان‌شناسی رایانشی؛
دانشگاه صنعتی شریف habdi.cnlp@gmail.com



دریافت: ۱۳۹۴/۱۱/۰۶ | پذیرش: ۱۳۹۵/۰۵/۱۲ | مقاله برای اصلاح به مدت ۳۱ روز نزد پدیدآوران بوده است.

چکیده: یکی از پایه‌ای‌ترین مراحل پردازش خودکار متن، تشخیص خطاهای املایی و استانداردسازی نویسه‌هاست. بدون گذر از این مرحله، ذخیره‌سازی مستندات متنی با مشکلات متعددی مواجه شده و موجب اختلال در بازیابی ماشینی آن‌ها می‌گردد. بدین ترتیب، متخصصان حوزه‌های پردازش زبان طبیعی و زبان‌شناسی رایانشی همواره در تلاش هستند تا با ارائه روش‌ها و الگوریتم‌های مطلوب انواع داده‌ها را در بوتۀ پردازش قرار داده و به داده‌ای استاندارد دست یابند. در زبان انگلیسی و برخی زبان‌های دیگر، تحقیقات متعددی در این زمینه انجام شده و به دنبال آن زبان فارسی نیز در این زمینه مورد تحقیق قرار گرفته است. این تحقیقات متعدد گاهی در حد پژوهش به قوت خود باقی مانده و گاهی نیز در قالب محصول عرضه شده است. مقاله حاضر به طبقه‌بندی انواع روش‌ها و دادگان مورد نیاز در این تحقیقات پرداخته و فرایند هر کدام از آن‌ها را به‌طور خاص و نحوه سنجش میزان دقت پردازش آن‌ها را به‌طور عام شرح می‌دهد. در این مقاله همچنین، نحوه عملکرد سامانه‌های تک‌زبانۀ فارسی توصیف شده و به نحوه برخورد آن‌ها با چالش‌های زبان فارسی اشاره می‌گردد.

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

فصلنامه | علمی پژوهشی
پژوهشگاه علوم و فناوری اطلاعات ایران

شاپا (چاپی) ۲۲۵۱-۸۲۲۳

شاپا (الکترونیکی) ۲۲۵۱-۸۲۳۱

نمایه در SCOPUS و ISI، LISTA، و

jipm.irandoc.ac.ir

دوره ۳۲ | شماره ۴ | صص ۱۱۴۳-۱۱۷۰

تابستان ۱۳۹۶



۱. مقدمه

نگارش متن همواره دستخوش سلیقه‌های مختلفی در طول تاریخ بوده است. هیچ‌گاه بر سر این که کدام حالت نوشتاری درست است و کدام غلط، اتفاق نظر کاملی وجود نداشته و تنوع نگارش در انواع متون به‌وفور دیده می‌شود. گرچه تنوع نگارشی را شاید حاصل خلاقیت ذهنی بشر بدانیم، اما این خلاقیت پردازش ماشینی متن را با چالش‌های متعددی روبه‌رو می‌کند و دقت پردازش داده‌ها را به میزان چشمگیری کاهش می‌دهد. در کنار تنوع نگارشی، غلط‌های سهوی املائی نیز وجود دارد که فحوای متن را منحرف کرده و ماشین را از انجام پردازش‌های دقیق در حوزه‌هایی نظیر ترجمه ماشینی، پردازش و تشخیص گفتار، بازیابی اطلاعات و غیره باز می‌دارد. بدین ترتیب، متخصصان حوزه‌های پردازش زبان طبیعی و زبان‌شناسی رایانشی همواره سعی دارند که با ارائه روش‌ها و آزمایش الگوریتم‌های داده‌کاوی و پالایش داده به تشخیص غلط‌های املائی 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 مازول استانداردسازی مواردی مانند نوع نوشتن تاریخ، حروف فارسی و عربی، فاصله و نیم‌فاصله، خط تیره، علائم نگارشی و کاربردهای متنوع آن‌ها، واحدهای اندازه‌گیری مثل کیلوگرم و kg و یا ک.گ.، اعداد (به‌صورت حروفی و عددی)، فاصله حاشیه پاراگراف‌ها و در نهایت، ساختار بخش‌های مختلف مستندات نظیر مقالات و صفحات وب در سرتاسر دادگان متنی همگن و یکنواخت می‌شوند. از طریق مازول خطایابی املائی، انواع خطاهای املائی غیرعمدی شناسایی می‌شوند. این خطاها را «ژورافسکی و جیمز» چنین طبقه‌بندی کرده‌اند:

الف. خطاهای غیر کلمه‌ای^۱: در این نوع خطاها، کلمه صحیح به کلمه ناموجود در فرهنگ لغت یک زبان تبدیل می‌شود؛ مانند: مرسه (املائی غلط کلمه مدرسه)، الما (املائی غلط کلمه املا)

ب. خطاهای کلمه‌ای^۲: در این نوع خطاها، دو کلمه به لحاظ شباهت املائی و شناختی

1. non-word

2. real-word

(آوایی) به‌طور غلط جابه‌جا می‌شوند؛ مانند شعر و شر (املائی) و مانند ثواب و صواب (شناختی) (Jurafsky and James 2009).

وجود انبوه داده در زبان‌های مختلف، پژوهشگران متخصص در علوم زبان‌شناسی رایانشی و پردازش زبان طبیعی را بر آن داشته تا در مسیر خودکارسازی استانداردسازی و خطایابی داده‌های متنی گام بردارند. تنها دلیل این امر را می‌توان در صرف وقت و هزینه زیاد در عملیات پیش‌پردازش داده یافت. بدین ترتیب، در زبان انگلیسی و برخی زبان‌های دیگر نظیر عربی، چینی و عبری، تحقیقات متعددی در زمینه ساخت استانداردساز و خطایاب انجام شده است و به‌دنبال آن زبان فارسی نیز در این زمینه مورد تحقیق قرار گرفته است. 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. هیچ‌کدام از تحقیقات انجام‌شده به‌دقت ۱۰۰ درصد نرسیده است و بسیاری از پژوهشگران، اکنون نیز به ساخت استانداردساز و خطایاب مشغول هستند. از این رو، بررسی نحوه 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 هر کدام به‌طور خاص و نحوه سنجش میزان دقت پردازش آن‌ها به‌طور عام پردازیم. همچنین، قصد داریم چالش‌های زبان فارسی را نیز طبقه‌بندی کنیم و نحوه عملکرد سامانه‌های تک‌زبانۀ فارسی را در راستای چالش‌های زبان فارسی شرح دهیم.

۲. دادگان مورد نیاز و روش‌های خطایابی و استانداردسازی متن

۲-۱. دادگان مورد نیاز

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.

جدول ۱. انواع داده و ساختار آن

ساختار داده	داده
پیکره موازی	پیکره
پیکره غلط	
فرهنگ لغت نمایه‌گذاری نشده	فرهنگ لغت
فرهنگ لغت نمایه‌گذاری شده	

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

۲-۱-۱. پیکره

۲-۱-۱-۱. پیکره موازی: پیکره‌های موازی مناسب برای انجام استانداردسازی و خطایابی، از جملات درست و غلط هم‌تراز شده متشکل هستند که با به کارگیری الگوریتم‌های حوزه ترجمه ماشینی آموزش داده شده و قابل استفاده می‌باشند (Schlippe et al. 2010).

۲-۱-۱-۲. پیکره غلط: این نوع پیکره از مجموعه متونی جمع‌آوری می‌شود که به‌طور طبیعی دارای غلط املائی هستند. معروف‌ترین این نوع پیکره، پیکره انگلیسی «بیرکبک»^۳ دانشگاه آکسفورد و همچنین، مجموعه غلط‌های املائی «پتر نورویگ»^۴ است که از «ویکی‌پدیا» و «بیرکبک» استخراج شده است. پیکره «بیرکبک» دارای ۳۶۱۳۳ شکل غلط^۵ ۶۱۳۶ کلمه انگلیسی از گروه‌های سنی مختلف است.

۲-۱-۲. فرهنگ لغت

۲-۱-۲-۱. فرهنگ لغت نمایه‌گذاری نشده: فرهنگ لغت‌هایی که به‌طور بهینه نمایه‌گذاری نشده باشند، در زمره این گروه‌ها قرار می‌گیرند. این گونه فرهنگ لغت‌ها، در کارهای پردازشی حجیم به هیچ وجه قابل استفاده نیستند.

۲-۱-۲-۲. فرهنگ لغت نمایه‌گذاری شده: این نوع فرهنگ لغت‌ها با توجه به نوع تحقیق و یا نرم‌افزاری که در آن به کار می‌رود، نمایه‌گذاری می‌شوند. از انواع روش‌های نمایه‌گذاری می‌توان به روش ایجاد جدول درهم‌سازی یا هش^۶ (جدولی که لغات آن دارای کلیدهای خاصی نظیر نمایه، حرف‌های الفبا و یا بسامد باشند (Muth and Tharp 1977)

1. parallel corpus

2. misspelled corpus

3. Birkbeck: <http://www.ota.ox.ac.uk/headers/0643.xml>

4. Peter Norwig

5. lexiconlexicon

6. hash table

(Bhatti et al. 2014; و مدل درختی (لغات به صورت سلسله‌مراتبی مرتب شوند (Beliga, Pobar, and Martinčić-Ipšić 2015)) اشاره کرد.

۱-۲-۳. فرهنگ لغت آموزش داده شده: این نوع فرهنگ لغت‌ها با استفاده از الگوریتم‌های خوشه‌بندی^۱ و مدل‌های موضوع پنهان^۲ (مدلی که بر اساس آن توکن‌های مجموعه انبوهی از داده‌ها در موضوع‌هایی مانند موضوع ۰، موضوع ۱ و غیره قرار می‌گیرند) آموزش داده شده و سپس نمایه‌گذاری می‌شوند تا سرعت پردازش به میزان قابل توجهی افزایش یابد. انواع روش‌های خوشه‌بندی نظیر «کی-مینز»^۴، «کی-مدویدز»^۵ و غیره وجود دارد. «زمپیری» و همکاران بر این باور بوده‌اند که فرهنگ لغت هر چقدر 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، «مدویدز»ی که به کلمه غلط شبیه‌تر باشد انتخاب شده و تنها فاصله مدخل‌های آن خوشه با کلمه غلط مقایسه و بدین ترتیب، 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 به‌طور مختصر اشاره می‌کند.

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

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

1. clustering

2. Latent Topic Models

3. token

4. k-means

5. k-medoids

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

۲-۲-۱. روش فاصله حرف^۱: 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. به‌عنوان مثال، فاصله حرف در دو کلمه "nicer" و "nised" برابر با منفی ۵ است.

۲-۲-۲. بسامد کلمات: در این روش، الگوریتم به‌صورت دو مرحله‌ای عمل می‌کند. در مرحله نخست، الگوریتم کلمه را با کلمات پرسامد مقایسه می‌کند و در صورت عدم یافتن 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 به‌جای محاسبه بسامد محض، از بسامد وزن‌دار (حاصل ضرب بسامد پیکره‌ای و فاکتور وابستگی به طبقه خاص^۲ (Salton and Buckley 1988)) نیز استفاده نمود.

۲-۲-۳. فاصله ویرایشی کمینه^۳: روش‌های متفاوتی برای محاسبه این نوع فاصله وجود دارد که به تک‌تک آن‌ها اشاره می‌کنیم.

۲-۲-۳-۱. فاصله «همینگ»: اندازه فاصله (Hamming (1950) برای دو رشته با طول مساوی برابر با تعداد جایگاه‌هایی است که حرف‌های متناظر متفاوت باشند. برای مثال، فاصله «همینگ» دو کلمه "nice" و "bise" برابر با ۲ است. از آنجا که این عدد با اعداد متناظر در یک دامنه 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. در نتیجه، فاصله این دو کلمه، ۲ بر ۴ یا ۰/۵ در نظر گرفته می‌شود.

۲-۲-۳-۲. فاصله «لونشتاین»: بر اساس روش (Levenshtein (1965)، دو لغت در قالب تشکیل ماتریسی با هم مقایسه شده و کمترین تعداد تغییراتی که باید بر روی لغت اعمال کرد تا به لغت دیگر رسید، مشخص می‌شود. این تغییرات شامل درج حرف اضافه^۴، حذف

1. letter distance

2. the product of term frequency and inverse document frequency

3. minimum edit distance

4. insertion

حرف^۱ و جایگزینی حرف^۲ است. برای مثال، فاصله «لونشتاین» دو کلمه «زبان» و «زندان»^۲ است. در جدول ۳، آخرین عدد ماتریس که در گوشه پایین سمت راست قرار دارد، فاصله «لونشتاین» دو کلمه است.

جدول ۳. ماتریس لونشتاین «زبان» و «زندان»

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

در تازه‌ترین مطالعات، «لیو، ونگ، و جیانگ» روش فاصله افزایش یافته^۳ را پیشنهاد کردند. برای انجام این کار، آن‌ها پیکره‌ای را آماده کردند که دارای دو نوع فرایند آموزش بوده است.

الف. آموزش انتخاب جفتی آگاه از بافت^۴: در این مرحله، بین کلمه و توکن برداری تشکیل داده و میزان شباهت بافتی بین آن‌ها از طریق تعیین میزان فاصله کسینوسی بررسی می‌شود.

ب. برجسب‌زنی توالی کاراکتری^۵: با استفاده از مدل شرطی میدان تصادفی (سی آر اف^۶)، انواع یک کلمه را با برهم‌زدن نظام واجی، سیلابی و صرفی تولید می‌کنند. در این روش، هر کلمه موجود در فرهنگ لغت قبل از به هم خوردن شکافته شده و بر حسب لایه‌های واجی، سیلابی و صرفی برجسب‌های Begin-Inside-Last-Outside-Unit می‌خورد (Liu, Weng, and Jiang 2012). نمونه‌ای از این نوع برجسب‌زنی در شکل ۱، قابل مشاهده است.

1. deletion
2. substitution
3. enhanced level transformation
4. context-aware training pair selection
5. character-level sequence labeling
6. conditional random fields (CRF) در این مدل، احتمال رویداد یک متغیر تصادفی بر اساس رویداد متغیر تصادفی (CRF) همسایه تعریف می‌شود

Character	a d v e r t i s e m e n t s
Phoneme	A E D V E R E R T A Y Z _ M A H N T S
Phoneme boundary	O O O B I L I 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 برجسب‌زنی برای کلمه advertisement

«لیو، ونگ، و جیانگ» بر اساس این ایده، فرایند استانداردسازی را به سه زیراستانداردسازی^۱ تقسیم کردند و سپس خروجی‌های این سه زیراستانداردسازی را ترکیب کرده و عملیات نهایی استانداردسازی را بر اساس «لونشتاین» بر روی متن انجام دادند (همان).

۲-۳-۳. فاصله «دامرو- لونشتاین»: این روش تا حدودی همانند روش «لونشتاین» است، با این تفاوت که جابه‌جایی حرف‌ها^۲ نیز به تغییرات روش «لونشتاین» اضافه شده است. برای مثال، فاصله «لونشتاین» دو کلمه «زبان» و «بزان»،^۲ است، در صورتی که فاصله «دامرو- لونشتاین» این دو کلمه ۱ است، چرا که با یک عمل جابه‌جایی می‌توان این دو کلمه را به همدیگر تبدیل کرد. در جدول ۴، آخرین عدد ماتریس که در گوشه پایین سمت راست قرار دارد، فاصله «دامرو- لونشتاین» دو کلمه است.

جدول ۴. ماتریس دامرو- لونشتاین «زبان» و «زندان»

ن	۱	ب	ز		
ز	۳	۲	۱	۰	
ن	۲	۱	۱	۱	
د	۲	۱	۱	۲	
ا	۱	۲	۲	۳	
ن	۲	۳	۳	۴	

۲-۳-۴. فاصله «واگنر و فیشر»: روش Wagner and Fischer (1974) همانند روش «لونشتاین» دارای تغییرات درج حرف اضافه، حذف حرف و جایگزینی حرف است؛ با این تفاوت

1. subnormalizer

2. Dameru- Levenshtein

3. transposition

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 می‌گردد. به‌عنوان مثالی ساده، رشته‌های دو کلمه «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 زمره‌ی الگوریتم‌های برنامه‌سازی پویاست؛ بدین معنا که در فرایند انجام محاسبه‌ی این فاصله، رشته‌ها به بخش‌های مختلف تقسیم شده و تغییرات به همراه وزن منحصر به فرد در آن بخش محاسبه می‌شود.

۲-۳-۵. روش «اوکونن»: روش (Ukkonen (1992 تغییر یافته‌ی روش «واگنر-فیشر» است که با کاهش پیچیدگی زمانی و فضایی قابل توجهی همراه شده است.

۲-۳-۶. فاصله «نیدلمن و وانج»: روش (Needleman and Wunsch (1970 در تشخیص شباهت زن‌ها مورد استفاده قرار می‌گیرد. الگوریتم این روش، همانند روش «واگنر-فیشر» در زمره‌ی الگوریتم‌های برنامه‌سازی پویاست؛ با این تفاوت که در حوزه‌ی پردازش زبان طبیعی فاصله‌ی کمینه و در حوزه‌ی بیوانفورماتیک بیشینه‌ی شباهت ملاک سنجش است. در جدول ۵، آخرین عدد ماتریس که در گوشه‌ی پایین سمت راست قرار دارد، شباهت «نیدلمن-وانج» دو کلمه است.

جدول ۵. ماتریس شباهت «زبان» و «زندان»

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

۲-۳-۷. فاصله «هیرشبرگ»: روش (Hirschberg (1975 نوع تغییر یافته‌ی تقسیم و حل الگوریتم

«نیدلمن-وانچ» است. بدین گونه که «هیرشبرگ»، الگوریتم «نیدلمن-وانچ» را به چند مسئله دیگر تقسیم کرده و سعی می کند نخست آن‌ها را حل کند و سپس، به شباهت (یا فاصله کمینه) برسد. این نوع فاصله سبب شده است که میزان اشغال حافظه به میزان قابل توجهی کاهش یابد.

۲-۳-۸. فاصله «اسمیث و واترمن»: روش (Smith and Waterman (1981) همانند روش «نیدلمن-وانچ» است؛ با این تفاوت که کلیه نمایه‌های منفی در ماتریس «اسمیث-واترمن» برابر با ۰ است. پس از تشکیل این ماتریس، عملیات پس‌گردی^۱ از عدد آخر شروع و آن قدر ادامه پیدا می کند تا این که به صفر برسد.

۳-۳-۹. فاصله «جرو-وینکلر»: این روش که مدل تغییر یافته Jaro (1989) است، میزان تشابه دو کلمه (معمولاً اسامی خاص) را محاسبه می کند. در این روش میزان حرف‌های مشترک و تعداد جابه‌جایی حرف‌ها بسیار مهم است. فرمول محاسبه «جرو-وینکلر» بدین گونه است:

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

در فرمول ۱، m تعداد حروف مشترک بین دو کلمه، s کلمه، t تعداد حروف مشترک تا ۴ کاراکتر اول و p عدد ثابت «وینکلر» (معمولاً ۰/۱) است. به‌عنوان مثال، شباهت «جرو-وینکلر» دو کلمه «مهرناز» و «مهرانز» طبق فرمول ۱، ۰/۹۶ است.

۲-۲-۴. آماده‌سازی: پاسخ ذهن به محرک جدید (یادگیری مطلب جدید)، بر اساس تأثیرپذیری غیرصریح حافظه از محرک قبلی (خود مطلب و بافت مطلب قبلی). «لیو، ونگ، و جیانگ» معتقد هستند که آماده‌سازی حافظه نقش مهمی در درک توکن غلط دارد (Liu, Weng, and Jiang 2012). طبق ایده طراحان این سیستم، چند حرف اول کلمه نقش مهمی در تشخیص کلمه غلط دارند. آماده‌سازی طبق فرمول ۲ اندازه‌گیری می‌شود:

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

طبق این فرمول، نخست TF یا بسامد واژه در پیکره محاسبه می‌شود. سپس، اندازه

طول طولانی‌ترین کلمه‌ای که با حرف اول کلمه غلط آغاز شده باشد (LCS) len از جدول کلمات پیدا شده و بر اندازه 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 به کار رفته در مسائل شناختی از این لحاظ متفاوت است که در نوع شناختی بافت به کلمه ماقبل مربوط می‌شود و در این نوع آماده‌سازی، پیکره به‌طور کامل به‌عنوان بافت در نظر گرفته می‌شود.

۲-۵. مدل کانال نویزی: مدل کانال نویزی چارچوبی است که به‌صورت متداول در غلط‌یابی املائی استفاده می‌شود (Shannon 1948). در گام اول این مدل، یکی از روش‌های فاصله 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.

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

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

۲-۶. مدل‌های زبانی: مدل‌های زبانی، احتمال یک رشته از واحدهای زبانی را با استفاده از توزیع احتمال (چندنگاشتی)^۳ محاسبه می‌کند (Chen 2009 Liu et al. 2011; Wu et al. 2010;). در این مدل‌ها نیز، نخست یکی از روش‌های فاصله ویرایشی برای کلمه محاسبه شده و لیستی از گزینه‌های درست برای کلمه غلط ارائه می‌شود. سپس، احتمال‌های چندنگاشتی آن‌ها با توجه به کلمات قبل و بعد از آن در متن از روی پیکره محاسبه شده و از طریق آرگومان ماکزیمم، بهترین گزینه برای کلمه غلط انتخاب می‌شود.

۲-۷. مدل مبتنی بر دسته‌بندی: با استخراج انواع ویژگی‌های ذکرشده در ۲-۱-۳، می‌توان از انواع دسته‌بندی‌ها همچون «نایو بیس»^۵ و مدل بُردار پشتیبان در غلط‌یابی املائی و تصحیح آن استفاده کرد.

◇ مدل «نایو بیس» یا بیزین^۶ ساده، یک مدل احتمالی شرطی است؛ بدین صورت که احتمال وقوع رخداد وابسته به احتمال وقوع رخداد قبلی باشد. این روش دارای

1. Longest Common Subsequence

2. Noisy Channel Model

3. n-gram

4. classification

5. Naïve Bayes

6. Nayo-Biz Model

پیچیدگی به مراتب کمتری است و در بین دسته‌بندی‌ها از مقبولیت بالایی برخوردار است. «گولدینگ» از این مدل برای انتخاب گزینه درست از مجموع گزینه‌ها با توجه به بافت آن‌ها استفاده کرده است (Golding 1995). آرگومان ماکزیمم مدل «ناویو بیژ» طبق فرمول ۴ محاسبه می‌شود:

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

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

◇ مدل ماشین بردار پشتیبان: مدل بردار پشتیبان را (Cortes and Vapnik 1995) معرفی کرده‌اند. این مدل، از جمله مدل‌های غیراحتمالی است و با تشکیل صفحات هایپر^۲ و محاسبه 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).

۲-۸. مدل‌های آوایی^۳: مدل‌های آوایی کمک می‌کنند که بتوان به جای حروف الفبا، از تلفظ کلمات جهت تشخیص توکن‌های غلط استفاده نمود.

الف. سانداکس^۴: «سانداکس» روشی برای نمایه‌گذاری اسامی بر اساس تلفظ آن‌هاست (Hall and Dowling 1980). رمز آوایی «سانداکس» 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 هر کدام با عنوان شماره ۱ رمزگذاری می‌شوند.

ب. فونیکس^۵: «فونیکس» نوع بهبودیافته «سانداکس» است (Gadd 1990). هر یک از حرف‌ها به مجموعه‌ای از رمزهای «سانداکس» نگاشت می‌شوند. قبل از انجام فرایند نگاشت، حدود ۱۶۰ تغییر گروه-حرفی به کار برده می‌شود تا استانداردسازی رشته انجام شود. 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

1. Support Vector Machine

2. Hyperplane

3. Phonetic Models

4. Soundex

5. Phonix

رمز گذاری آوایی به وجود می‌آورند و امکان تشخیص رشته‌هایی نظیر "c" و "s" را مهیا می‌سازند.

پ. شیوه کیونگاشتی^۱: شیوه «کیونگاشتی» همان اندازه گیری فاصله رشته‌ای بر اساس تعداد «کیونگاشت» است (Ukkonen 1992). «کیونگاشت» رشته "s"، همان زیررشته "s" با طول ثابت "q" است. یک نمونه ساده از این اندازه گیری، انتخاب "q" و شمردن تعداد «کیونگاشت»های مشترک بین دو رشته است. با این حال، شمارش تنها «کیونگاشت»ها نمی‌تواند نمایانگر تفاوت طولی باشد. برای حل این مشکل، «اوکونن» نوع جدیدی از فاصله کیونگاشتی پیشنهاد کرده که می‌توان برای رشته‌های بدون تکرار کیونگاشتی به صورت فرمول ۵ تعریف کرد:

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

که در آن G_s مجموعه‌ای از «کیونگاشتی»ها در رشته "s" می‌باشد. برای مثال، طبق این فرمول فاصله بین "rhodes" و "rod"، برای q با مقدار ۲ یا ۳، ۵ است (همان).

ت. گرپ^۲: عبارت‌های باقاعده «گرپ» برای تطبیق الگو مورد استفاده قرار می‌گیرند. انواع آن «ای گرپ»^۳ (گرپ تقریبی)، «ای گرپ»^۴ و «اف گرپ»^۵ است (Wu and Manber 1992). از میان این روش‌ها، «ای گرپ» بیشتر برای خطایابی املائی به کار می‌رود. اساس کار این روش، الگوریتم «لونشتاین» است و برای بهبود کارایی از الگوریتم‌های مختلفی استفاده می‌کند. «ای گرپ» برای الگوهای ساده تشخیص خطا از الگوریتم «بویر مور»^۶ استفاده می‌کند که شامل الگوریتمی سریع برای شناسایی رشته‌های دارای زیررشته است. «ای گرپ» به جست‌وجوی حداکثر "k" درج، حذف یا جایگزینی می‌پردازد که در آن "k" ضریبی ثابت و از پیش تعریف شده است.

ث. ادیتکس^۷: «ادیتکس» سنجش فاصله آوایی است که ویژگی‌های فاصله ویرایشی را با روش گروه‌بندی حرف‌هایی که توسط «ساندکس» و «فونیکس» به کار گرفته شده، ترکیب می‌کند (Zobel and Dart 1996). بر اساس «ادیتکس»، مطابقت جفتی حاصل روش‌های آوایی است و به روش‌های رشته‌ای مربوط نمی‌شود. از طریق «ادیتکس» می‌توان مطابقت

1. q-gram

2. Global regular expression print (Grep)

3. A-grep

4. E-grep

5. F-grep

6. Boyer-Moore

7. editex

بین حرف‌ها و تلفظ مشابه را دقیق‌تر نشان داد.

ج. شیوه‌های آوامتری^۱: شیوه‌های آوامتری مهارت‌های تطبیقی هستند که بر اساس مطالعات آوایی به وجود آمده‌اند (Zobel and Dart 1996). الگوریتم‌های تطابق آوامتری شامل دو مرحله هستند:

- ◇ مرحله اول: رشته‌ای از حرف‌ها به وسیله الگوریتم تبدیل رشته به تلفظ، به رشته‌ای از واج‌ها تبدیل می‌شود. چندین الگوریتم مناسب برای این منظور وجود دارد، ولی الگوریتم‌های وابسته به بافت مؤثرترین 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.

ج. کاهش تدریجی^۲: این روش، بهبودی برای مهارت‌های فاصله ویرایشی با در نظر گرفتن ویژگی‌های انسانی است (Zobel and Dart 1996). تفاوت‌های آوای آغازین تلفظ نسبت به تفاوت‌های آوای پایانی تلفظ از اهمیت بیشتری برخوردار هستند. در فاصله ویرایش کاهش یافته تدریجی، حداکثر جریمه^۳ برای جایگزینی یا حذف در ابتدای رشته از حداقل دو برابر جریمه برای جایگزینی یا حذف در انتهای رشته بیشتر است.

ح. متافون^۴: فرایند رمزگذاری متافون اصلی، تعداد ۱۶ کاراکتر صامت را به کار می‌برد: "0BFHJKLMNPRSTWXY" که عدد 0 نشان‌دهنده "the" است (Philips 1990). حرف‌های مصوت‌های AEIOU نیز فقط در ابتدای رمز به کار برده می‌شوند. قوانین کلی «متافون» عبارت‌اند از: حذف دو حرف تکراری مجاور، تبدیل حروف به یکدیگر، تبدیل توالی دو حرف به یک حرف و حذف تمامی مصوت‌ها. به این خاطر که «متافون» اصلی خطاهای زیادی را در برداشت، «متافون دوپل» جایگزین آن شد. در گام بعد، «متافون سیل» جایگزین دوپل شد که بسیاری از رمزگذاری‌های اشتباه را که به وسیله دو نسخه قبلی وجود داشت، تصحیح کرده است.

1. Phonometric

2. tapering

3. penalty

4. metaphone

۲-۳. روش‌های ارزیابی سامانه‌های خطایاب و استانداردسازی

برای ارزیابی، در مرحله نخست ۲۵ درصد داده‌ها را که به صورت انسانی تأیید شده‌اند، کنار گذاشته و از آن‌ها جهت آزمایش سامانه 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 به درستی تشخیص دهد و یا به غلط. این چهار مفهوم در جدول ۶ آمده است.

جدول ۶. مفاهیم پایه ارزیابی سامانه‌های دسته‌بندی

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

با استفاده از چهار مفهوم جدول ۶، دقت، صحت و بازخوانی سامانه از طریق فرمول

۶، ۷ و ۸ محاسبه می‌شود:

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

تفاوت دقت و صحت را می‌شود چنین 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.

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

معیار دیگر ارزیابی رتبه واران‌های میانگین^۱ است که اغلب در سامانه‌های پرسش و پاسخ به کار می‌رود (Radev et al. 2002). در این رتبه‌بندی، فرایند تولید جواب‌های ممکن به یک سؤال (خطای املائی) ارزیابی می‌شود. درستی جواب‌ها بر اساس احتمالات آن‌ها مرتب می‌شود. این معیار از طریق فرمول ۱۰ محاسبه می‌گردد.

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

1. Mean Reciprocal Rank

در این فرمول، تعداد سؤال‌ها Q و رتبه جواب درست rank است.

۲-۴. 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 شناخته شده

نرم‌افزارهای بسیاری برای خطایابی املائی در متن وجود دارد. از برجسته‌ترین و شناخته شده‌ترین نرم‌افزارهای جهان «مایکروسافت ورد»^۱ است. این مجموعه بر اساس الگوریتم وابسته به بافت به خطایابی املائی و دستوری با صحت به مراتب بالا و بازخوانی کم می‌پردازد. خطایاب بعدی، خطایاب «گوگل» است که بر روی مرورگر نصب شده و کاربرد در تمامی نوشته‌ها می‌تواند بر حسب انتخاب به صورت وابسته به بافت به کار برد. این خطایاب از الگوریتم «دامرو لونشتاین» برای تصحیح خطا استفاده می‌کند. در نهایت، سومین نرم‌افزار معتبر جهان «جینجر»^۲ است که توسط «موآز شخت»^۳ طراحی گردیده و بر حسب الگوریتم وابسته به بافت به غلط‌یابی املائی و دستوری می‌پردازد. ایراد بارز این نرم‌افزار، تصحیح خودکار خطاهاست که در موارد بسیاری می‌تواند آسیب جدی به مفهوم متن وارد کند.

ب. افزونه‌های شناخته شده

۱. آی اسپل؛ «آی اسپل» یک غلط‌یاب املائی برای «یونیکس»^۵ است که توسط «گورین»^۶ طراحی شده است. «آی اسپل» با استفاده از الگوریتم «دامرو لونشتاین» و فرهنگ لغت «موبی وردز»^۷ به اصلاح خطاهای املائی می‌پردازد. این فرهنگ لغت دارای حدود ۱۷۷۲۶۷ مدخل به همراه تلفظ آن‌هاست. تنها ایراد این افزونه، نبود کلمات کافی و حالت‌های املائی عموماً پذیرفته شده و همچنین فرایند بافت‌محور است.

۲. آی اسپل؛ «آی اسپل» یکی از افزونه‌های تحت لیسانس «گنو»^۹ است که برای جایگزینی

1. Microsoft Word

2. Ginger

3. Mao Shacht

4. Ispell

5. Unix

6. Gorin

7. Moby Words

8. Aspell

9. GNU

«آی اسپل» توسط «کوین آتکینسون»^۱ نوشته شده است. تفاوت این افزونه با «آی اسپل» در این است که «ای اسپل» برای کلمه‌های غلط، گزینه‌های منتخب بیشتری را برای کاربر پیشنهاد می‌دهد. برای مثال، زمانی که به کلمه «trouble» می‌رسیم، «آی اسپل» فقط کلمه rubble را پیشنهاد می‌دهد، ولی «ای اسپل» گزینه‌های منتخب دیگری نظیر dribble، trouble، rubble و غیره را پیشنهاد می‌دهد. این افزونه قابلیت تحلیل مستندات «یوتی اف ۸»^۲ را نیز دارد. این ابزار مجهز به ماژول غلط‌یابی املائی بر اساس حرف و آواست و حافظه بیشتری را نسبت به «آی اسپل» اشغال می‌کند.

۳. **مای اسپل:** ماژولی است که قبلاً بر روی «او او- رایتر»^۳ نرم‌افزار OpenOffice.org کار می‌کرد. نویسنده این ماژول، «کوین هندریکس»^۴، روش‌های نسخه‌های لایه‌باز خطایاب‌های دیگر نظیر «آی اسپل» را با این افزونه ادغام کرده و «مای اسپل» را ساخته است. الگوریتم این افزونه ترکیبی از الگوریتم «آی اسپل» و «متافون» است. «مای اسپل» دارای فضا است که این فضا نیز دارای فایل‌هایی برای املا و فرهنگ لغت است. به این دلیل که از انباشته شدن حالت‌های صرفی مختلف برای واژه جلوگیری شود، فایل با فرمت dic. واژگان به همراه رابط^۵ آن به وندها در فایل با فرمت aff. ساخته می‌شود. تنها ایراد این ماژول، کندی عملکرد دقیق بخش پردازش صرفی آن است.

۴. **هانسپل:** «هانسپل»، نوع پیشرفته افزونه «مای اسپل» است که علاوه بر غلط‌یابی املائی، 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 «او کامورف»^۶ در کنفرانس چهل و هفتم «ای سی ال»^۷ پیشنهاد شده است (Tro et al. 2015). «هانسپل» دارای «او کامورف» (وندزدایی غیروابسته به زبان)، «مورف دی بی»^۸ (پایگاه داده لغوی و دستور صرفی که توسط «او کامورف» مورد استفاده قرار می‌گیرد) و «هانلکس» (کامپایلر^۹ مدیریت منبع است که کارایی الگوریتم را از نظر زمانی و فضایی بهبود می‌بخشد) است. این ماژول

1. Kevin Atkinson	2. utf-8	3. OOo Writer
4. Kevin Hendricks	5. locale	6. flag
7. Hunspell	8. Ocamorph	

۹. کنفرانس ACL بزرگ‌ترین و مهم‌ترین کنفرانس زبان‌شناسی رایانشی در جهان است.

ACL-IJCNLP (Association for Computational Linguistics, 2009, Singapore)

10. Morph D B	11. Compiler
---------------	--------------

با وجود قدرت خوب، در خطوط طولانی دارای دقت پایینی است.

۵. جزی: افزونه‌ای است که به زبان برنامه‌نویسی «جاوا» نوشته شده است. این افزونه می‌تواند برای غلط‌یابی املائی استفاده شود. طراح این ماژول، «میندوگس ایزدلیس»^۲ است. «جزی» از لحاظ عملکرد بسیار شبیه به «ای‌اسپل» است؛ با این تفاوت که در برخی موارد، به تک‌تک کاراکترها پیشنهاد املائی می‌دهد.

۱-۳. تشریح پژوهش‌های انجام‌گرفته بر روی زبان فارسی

۱-۳-۱. زبان فارسی

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

۱-۳-۲. چالش‌های پردازش املائی و استانداردسازی زبان فارسی

چالش‌های زبان فارسی را با اقتباس از (Megerdooian, 2004) «رسولی و مینایی» (۱۳۸۷)؛ (Shamsfard, Jafari, and Ilbeygi, 2010) «کاشفی، نصری، و کنعانی» (۱۳۸۹)؛ و (Feili, Montazery, and Pilehvar, 2014) چنین طبقه‌بندی می‌کنیم:

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

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

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 املائی درست، که هر چند سال یک‌بار از سوی فرهنگستان زبان و ادب فارسی اعلام می‌شود، دارای پایداری نیست. به‌عنوان مثال، بعضی کلمات دارای دو حالت املائی درست هستند، مانند «بلیت» و «بلیط».

ث. فاصله‌گذاری: فاصله‌گذاری (فاصله‌بندی) در زبان فارسی دارای قاعده‌ی قطعی نیست. زیرا گاهاً فاصله به نیم‌فاصله مانند «می‌خورد»، گاهی به تمام فاصله مانند «می‌خورد» و گاهی به فاصله 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 برجسته آن اشاره می‌کنیم.

- ◇ زبان فارسی زبانی اشتقاقی و زایشی است و در آن ممکن است با ترکیب واژه‌ها و وندها همواره 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 مکسر 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 واژه 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 اصلاً نوشته نمی‌شود. مانند «کتاب مینا» و «خانه‌ی». همچنین، بین نوشتن «ی» و «ء» و نوشتن 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^۱ است و ممکن است همین امر باعث شود که عملیات تقطیع با مشکلاتی مواجه شود.

د. کاربرد اعراب (هم‌نگاره‌بودن): عدم به کارگیری اعراب در متون فارسی می‌تواند فرایند خطایابی را دچار مشکل کند، مانند کلمه «مردم» که نبود اعراب منجر می‌شود به سه شکل مختلف خوانده شود: «مردم»، «مردم» و «مردم»

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 جعبه ابزار وجود دارد و یا ماژولی از عملیات بزرگ‌تر است. اما قبل از استانداردسازی نیز مرحله بسیار کلیدی تقطیع متن به توکن وجود دارد.

«مگردومیان» در دو مرحله تقطیع اولیه (غیروابسته با زبان) و پساتقطیع (وابسته به زبان فارسی) به تقطیع کلی متن پرداخته است. در مرحله پیش تقطیع، سیستم آن‌ها به صورت پایه‌ای کاراکترها را تقطیع می‌کند. معیار کاراکتر پایه، شکل پایانی حرف و فاصله 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 مرحله پساتقطیع (بعد از تقطیع اولیه)، الگوریتم هر دو کلمه تقطیع شده را در بین وندها جست‌وجو می‌کند (Megerdoomian 2004).

1. cursive script

«براری و قاسمی‌زاده» خطایاب «کلونایزر»^۱ غیروابسته به زبان را بر اساس ساختار داده‌های درختی سه‌تایی^۲ در حالت تطبیق‌پذیری معرفی کردند (Barari and QasemiZadeh, 2005). الگوریتم آن‌ها یادگیری الگوی خطا را از طریق رفتار کاربر یاد می‌گیرد. بخش‌های مختلف «کلونایزر» عبارت‌اند از: ماژول خطایاب (پردازش با الگوریتم دامرو لونشتاین)، واژگان (دارای ساختار درختی سه‌تایی)، هزینه انتقال (تغییر یا ویرایش)، ماژول یادگیری و تطبیق‌پذیری که به واسطه انتخاب کاربر الگوی خطا را استخراج می‌کند. ویژگی مهم این «کلونایزر» استفاده از حد آستانه‌ای محلی و جهانی است. حد آستانه‌ای محلی عمق جست‌وجو را بر اساس طول کلمه‌های پیشنهادشده محدود می‌کند و حد آستانه‌ای جهانی مانع تولید کلمه‌های نامناسب بر اساس طول رشته ورودی و تعداد کلمه‌های پیشنهادی می‌شود. «براری و قاسمی‌زاده» برای ارزیابی الگوریتم خود مجموعه دادگان ۵۵۹۵ کلمه‌ای از متن املاي دانش‌آموزان و مراکز تایپ با دو شکل صحیح و غلط را جمع‌آوری کردند.^۳ دقت آزمایش الگوریتم آن‌ها بر روی این مجموعه داده، حدود ۸۵ درصد نتیجه شده است (همان).

«رسولی و مینایی» (۱۳۸۷) عمل خطایابی را با استفاده از مقوله نحوی کلمه در متن کاربر انجام دادند که قابل پیاده‌سازی بر روی «مایکروسافت ورد» است. روش خطایابی آن‌ها بدین ترتیب است: تشخیص مقوله نحوی بر اساس ساختار کلمه، بررسی صحت مقوله نحوی از فرهنگ لغت و ارائه پیشنهادها، کاندیدهای املاي برای کلمه غلط در صورت تأیید عدم صحت از مرحله قبلی. در این الگوریتم به این دلیل که حجم عملیات پردازش پیشنهادها املاي 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 ۴/۵ برابر کم می‌شود. آن‌ها برای وزن‌دهی به پیشنهادها، این شاخص‌ها را در نظر گرفته‌اند: پیشنهادها موجود در متن و دارای بسامد بیشتر، پیشنهادهایی که مقوله نحوی آن‌ها با مقوله نحوی کلمه برابر باشد، پایان حروفی که فقط دارای شکل پایانی هستند، فشردن کلیدهای کناری، نزدن و یا زدن یک کلید، جابه‌جایی دو حرف مجاور، واژگان درست کاربر و مدل چندنگاشتی آن و در نهایت، نوع خطاهای کاربر. با توجه به شاخص‌های فوق، می‌توان وزن پیشنهادها را در

1. Clonizer

2. Ternary

3. <http://www.digitalclone.net/localization/spellchecker>

فرمول ۱۱ نشان داد:

$$(11) w_s = \sum_{i=1}^7 \alpha_i * \lambda_i$$

در این فرمول، λ شاخص و α ضریب شاخص است.

«شمس فرد، جعفری، و ایل‌بیگی» در استپ وان^۱ الگوریتم «مگردومیان» را توسعه داده و چالش‌های اصلی استانداردسازی زبان فارسی را تصحیح نموده‌اند (Shamsfard, Jafari, and Ilbeygi 2010). آن‌ها خطایاب جدیدی را پیشنهاد کردند. این خطایاب بر خلاف خطایاب‌های قبلی، به موقعیت کلمه در جمله و بافت کلمه توجه می‌کند و وب را به‌عنوان 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 مقایسه نتایج جست‌وجوی خروجی «ساندکس»، به‌واسطه فرمول ۱۲ از موتور جست‌وجوی «گوگل» محاسبه می‌شود.

$$(12) P = 1 + \frac{1}{9} \log\left(\frac{\#occ(T_p, Cs, T_n) + 1}{10^9}\right)$$

در این فرمول، Cs کاندیداهای پیشنهادی و T_p کلمه ماقبل و T_n کلمه مابعد است. هر دوی این سیستم‌ها به‌صورت قاعده‌بنیاد عمل می‌کنند. خطایاب «شمس فرد، جعفری، و ایل‌بیگی» روی سه متن (رمان، تاریخ و عمومی) که هر کدام دارای ۵۰۰ تا ۶۰۰ کلمه و در مجموع دارای ۱۰۰ غلط املائی بودند، آزمایش شد. نتایج این آزمایش در شکل ۱۵ قابل مشاهده است. در نهایت، ۹۹/۶ درصد خطاها تشخیص داده شد و ۹۲/۶ درصد از آن‌ها تصحیح گردید.

«دری 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 کمینه متناهی و قطعی بدون دور^۲ در «ماشین میلی»^۳ برای فشرده‌سازی واژگان استفاده کرده‌اند که حجم واژگان را ۳ تا ۶ برابر نسبت به فشرده‌سازی «آی‌اسپل» کاهش

1. STeP-1: <http://step1.nplab.sbu.ac.ir/>

2. Minimal Acyclic Deterministic Finite Automata

3. Mealy Machine

داده است. تمامی گذارهای این اتوماتا در آرایه‌ای ذخیره می‌شوند که هر عضو نماینده یک گذار و هر چند گذار نماینده یک حالت^۳ باشد. بدین ترتیب، کلمه اول به ستاک تبدیل می‌شود و از فهرست واژگان (واژگان «ای اسپل» فارسی که واژگان آن به صورت ستاک ذخیره شده‌اند) جست‌وجو می‌شود. واژگان از میان پیشنهادها بر اساس انتخاب کاربران دارای وزن متداول و مهجور شده و همچنین، بر اساس برجسب نحوی فعل و صفت و اسم مرتب می‌شوند. روش مورد استفاده در تصحیح خطا، روش معکوس حداقل فاصله ویرایشی^۴ است. در این روش، تغییرات ویرایشی زیر به صورت تک‌به‌تک بر واژه نادرست اعمال می‌شود و به محض این که به واژه درست منجر شد، عملیات اعمال تغییرات به اتمام می‌رسد.

«ویراستیار»^۵ افزونه‌ای برای «مایکروسافت ورد» است که برای استفاده کاربران فارسی‌زبان طراحی شده است (کاشفی، نصری، و کنعانی ۱۳۸۹). الگوریتم دقیق و جزئی این سامانه تشریح نشده و تنها گزارش قابلیت‌های اصلی نرم‌افزار موجود است که مهم‌ترین آن‌ها عبارت‌اند از: غلطیاب املائی، اصلاح نویسه‌های متن، اصلاح نشانه‌گذاری، تبدیل تقویم و تاریخ، تبدیل فینگلیش، پیش‌پردازش املائی متن، تبدیل اعداد و اصلاح نیم‌فاصله.

در ادامه این مسیر، Seraji (2012)، «پری‌پر»^۶ را برای زبان فارسی طراحی کرد. این سامانه که با ماژول «ویراستار» درهم آمیخته (Bargi 2011)، متن را برای پردازش‌های لازم در عملیات‌های زبان‌شناسی رایانشی آماده می‌کند. ویژگی‌های «پری‌پر» عبارت‌اند از: نرمال‌سازی متن، تبدیل سبک نوشتاری عربی به فارسی، اصلاح نیم‌فاصله و غیره.

«فیلی» و همکاران سامانه خطایاب «وفا»^۷ را برای زبان فارسی معرفی کردند (Feili et al. 2014). این سامانه هیبرید^۸ به صورت قاعده‌بنیاد و آماری به تشخیص و تصحیح خطاها می‌پردازد. واژگان پایه سامانه، ترکیبی از واژگان تصریف‌شده «فرهنگ دهخدا» (Dehkhoda 1998) به همراه میزان بسامد آن‌ها با استفاده از دو پیکره خبرنامه‌های «ایرنا» و «همشهری» و همچنین تعداد ۵۰۰۰ اسامی پربسامد از سامانه آموزشی است. در نهایت، واژگان سامانه تعداد ۱۲۰۰۰۰۰ واژه را در خود جای داده است. الگوریتم به کار رفته در این سامانه،

1. transition

2. array

3. state

4. Reverse minimum edit distance

5. <http://virastyar.ir/virastlive/index.html>

6. PrePer

7. <http://www.vafaspellchecker.ir/>

8. hybrid

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 صفحه کلید، حذف در حروف تایپ شده به وسیله انگشت کوچک، حروف تایپ شده به وسیله 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 چسباندن دو واژه، حذف و یا افزودن فاصله به کلمات مابعد و ماقبل می‌تواند تصحیح شود. سامانه «وفا» از فاصله ویرایشی کمینه استفاده می‌کند که سه ویژگی آوایی، شباهت ظاهری و تأثیرهای صفحه کلید را ملاحظه می‌کند. در نهایت، از حالت‌های اکتشافی ۱ استفاده کرده و لیست کاندیداهای درست را تولید می‌کند. پس از تولید این لیست، چسبیدن کلمه به کلمه ماقبل و مابعد نیز بر طبق ۷ حالت گفته شده بررسی می‌شود. داده‌های لازم جهت ارزیابی «وفا» از ۳۴۰ وبلاگ، جملاتی با طول ۱۴/۰۵ و با ۳۱۲ غلط املائی (۴۹ حذف، ۱۵ درج، ۶۱ جایگزینی و ۱۴ جابه‌جایی، ۱۴۲ سرهم‌نویسی و ۳۱ خطای نیم‌فاصله) استخراج شده و حالت درست آن به صورت انسانی تهیه شده است. علاوه بر این سامانه، دو سامانه «ویراستیار» و «مایکروسافت» نیز بر روی همین ارزیابی شده که نتایج رتبه و ارانه‌ای میانگین آن‌ها به ترتیب، ۰/۸۴۴، ۰/۸۲۸ و ۰/۸۱۵ به دست آمده است.

۴. جمع‌بندی و بحث

در این پژوهش انواع دادگان و روش‌های رایانشی و پردازشی برای فرایند خطایابی املائی 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 به عمل آمد. در پایان، چالش‌های زبان فارسی انجام گرفته در مورد خطایابی فارسی از دل تحقیقات مختلف استخراج شده و در ده طبقه مجزا معرفی گردید.

همچنین، هشت تحقیق بسیار مهم و برجسته تشخیص خطاهای املائی زبان فارسی و استانداردسازی آن از حیث مواد و روش‌ها و نتایج و ارزیابی آن‌ها به تفصیل تشریح شد. مطالعه حاضر نشان داد که بسیاری از چالش‌های موجود در زبان فارسی به‌طور کامل حل نشده و زمینه تحقیق در مورد خطایابی و استانداردسازی متنی همچنان ادامه خواهد داشت. بخشی از این چالش‌ها با از میان بردن اختلاف نظرهای متخصصان مطالعات نظری زبانی حل می‌شود و به تبع آن نوشتار فارسی هرچه بیشتر یکنواخت می‌گردد. این یکنواختی، مهندسان زبان را نیز از گرفتاری در سردرگمی‌های مداوم نجات می‌دهد؛ هرچند، هیچ زبانی در طول تاریخ همیشه یک شکل واحد نداشته و همواره دستخوش تغییرات مکرر بوده است. بنابراین، روش‌ها و الگوریتم‌های مهندسان زبان نیز باید به اندازه کافی قدرت یادگیری و عملکرد بالایی داشته باشد.

با افزایش آمار دانشجویان و همچنین گسترش انجام پژوهش و تحقیق، تولید مستندات علمی روزبه‌روز افزایش می‌یابد. مقاله 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 خلأ پژوهشی باید در پژوهش‌های بعدی برای زبان فارسی رفع گردد.

۵. یادداشت

مقاله حاضر برگرفته از طرح پژوهشی «طراحی و ساخت سامانه استانداردسازی و خطایابی متون فارسی» است که در پژوهشگاه علوم و فناوری اطلاعات ایران در حال انجام است.

فهرست منابع

- دری نوگرانی، صادق و محسن صبوریان. ۱۳۸۵. طراحی و پیاده‌سازی یک خطایاب فارسی. دومین کارگاه پژوهشی زبان فارسی و رایانه، دانشکده ادبیات دانشگاه تهران.
- رسولی، صادق و بهروز مینایی بیدگلی. ۱۳۸۷. روشی جدید در خطایابی املائی فارسی. دومین کنفرانس داده کاوی، دانشگاه امیرکبیر، تهران.
- کاشفی، امید، میترا نصری، و کامیار کنعانی. ۱۳۸۹. خطایابی خودکار در زبان فارسی. تهران: دبیرخانه شورای عالی اطلاع‌رسانی.

Bargi, A. A. 2011. Virastar. <https://github.com/aziz/virastar>. (accessed 2012)

Barari, L. and B. QasemiZadeh. 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., M. Pobar, and S. Martinčić-Ipšić. 2015. *Normalization of Non-Standard Words in Croatian Texts*. Opatija, Croatia: MIPRO CIS - Intelligent Systems.
- Bhatti, Z., I. A. Ismaili, W. J. Soomro, and D. N. Hakro. 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* 8 (1): Article 2. Doi: 10.1145/1482343.1482345
- Cortes, C. and V. Vapnik. 1995. Support-vector networks. *Machine Learning* 20: 273-297.
- Damerau, F. J. 1964. A technique for computer detection and correction of spelling errors. *ACM* 7: 171-176.
- Dehkhoda, A. 1998. *Loghatnameye Dehkhoda (Dehkhoda's dictionary)*. Vol. 12. Tehran: Tehran University Publications.
- Feili, H., N. Ehsan, M. Montazery, and M. T. Pilehvar. 2014. Vafa spell-checker for detection spelling, Feili, H., N. Ehsan, M. Montazery, and M. T. Pilehvar. 2014. Vafa spell-checker for detection spelling, grammatical, and real-word errors of Persian language, *Literary and Linguistic Computing Advance*, 31 (1): 95-117. Doi: 10.1093/lc/fqu043
- Gadd, T. 1990. PHONIX: The algorithm. *Program automated library and information systems* 24 (4): 363-366.
- Golding, A. R. 1995. A Bayesian hybrid method for context-sensitive spelling correction. In Proc. of 3rd Workshop on Very Large Corpora, Boston, MA.
- Hall, P.A.V. and G. R. Dowling. 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.* 29: 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 H. M. James. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. 2nd edition. Philadelphia: Prentice-Hall.
- Levenshtein, V. 1965. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10707-710 .:
- Liu, C., M. Lai, K. Tien, Y. Chuang, S. Wu, and C. Lee. 2011. Visually and Phonologically Similar Characters in Incorrect Chinese Words: Analyses, Identification, and Applications. *ACM Transactions on Asian Language Information Processing* 10 (2): 1-39.
- Liu, F., F. Weng, and X. Jiang. 2012. *A Broad-Coverage Normalization System for Social Media Languag*. Paper presented at: The 50th Annual Meeting of the Association for Computational Linguistics. 8-14 July 2012, Jeju Island, Korea.
- Megerdooimian, 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 A. L. Tharp. 1977. Correcting human error in alphanumeric terminal input. *Information Processing & Management*, 13 (6): 329-37. doi: 10.1016/0306-4573 (77) 90053-X.
- Needleman, S. B. and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* 48 (30): 443-53.
- Philips, L. 1990. Hanging on the Metaphone. *Computer Language* 7 (12): 39-43.
- Radev, D. R., H. Qi, H. Wu, and W. Fan. 2002. *Evaluating web-based question answering systems*.

- Proceedings of LREC., University of Las Palmas, Canary Islands, Spain.
- Salton, G. and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24 (5): 513–523.
- Schaback, J. and F. Li. 2007. Multi-Level Feature Extraction for Spelling Correction. Workshop on Analytics for Noisy Unstructured Text Data, IJCAI-07, Hyderabad, India.
- Schlippe, T., C. Zhu, J. Gebhardt, and T. Schultz. 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., H. S. Jafari, and M. Ilbeygi. 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 M. S. Waterman. 1981. Identification of Common Molecular Subsequences. *Journal of Molecular Biology* 147: 195–197.
- Tró, V., G. Gyepesi, P. Halácsy, A. Kornai, L. Németh, and D. Varga. 2015. *Hunmorph: Open Source Word Analysis*. Beijing, China: ACL.
- Ukkonen, E. 1992. Approximate string-matching with qgrams and maximal matches. *Theoretical Computer Science* 2: 191-211.
- Wagner, R. A. and M. J. Fischer. 1974. The String-to-String Correction Problem. *J. ACM* 21: 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) Washington D.C. USA 354–359.
- Wu, S., Y. Chen, P. Yang, T. Ku, and Chao-Lin Liu. 2010. *Reducing the False Alarm Rate of Chinese Character Error Detection and Correction*. In Proceedings of SIGHAN. Ningbo, China.
- Wu, S. and U. Manber. 1992. Fast text searching allowing errors. *Communications of the ACM* 35 (10): 83-91.
- Zampieri, M. and R. Cordeiro de Amorim. 2013. Effective Spell Checking Methods Using Clustering Algorithms. Recent Advances in Natural Language Processing. Hissar, Bulgaria.
- Zobel, J. and P. Dart. 1996. *Fnetik: An integrated system for phonetic matching*. Technical Report 96-6, Department of Computer Science, RMIT.

ملوک‌السادات حسینی بهشتی

دارای مدرک دکتری در رشته زبان‌شناسی همگانی از دانشگاه تهران است. ایشان هم‌اکنون استادیار پژوهشکده مدیریت دانش پژوهشگاه علوم و فناوری اطلاعات ایران است.

زبان‌شناسی، اصطلاح‌شناسی و مهندسی دانش از جمله علایق پژوهشی وی است.



هادی عبدی قویدل

دارای مدرک کارشناسی ارشد در رشته زبان‌شناسی از دانشگاه صنعتی شریف است. وی هم‌اکنون مدرس مدعو و پژوهشگر در دانشگاه شهید مدنی آذربایجان است. متن کاوی، پردازش خودکار مفاهیم استعاری، ترجمه ماشینی و مهندسی دانش از جمله علایق پژوهشی وی است.

