عنوان زیرپروژه:

استخراج نیازمندی‌های ابزار خطایاب املایی در لایه نحو زبان فارسی به پیکره‌های فارسی مورد نیاز

فهرست مطالب

عنوان **شماره صفحه**

عنوان **شماره صفحه**

1. مقدمه

با به وجود آمدن نظریات مختلف در مورد هوش مصنوعی[۱]، یکی از شاخه‌هایی که با گسترش رایانه، بسیار مورد توجه قرار گرفت، پردازش زبان طبیعی[۲] است. با گسترش داده‌های رایانه‌ای حاوی متون زبان انسانی، نیازهای متفاوتی به پردازش زبان طبیعی به وجود آمده و همین امر باعث به وجود آمدن گرایش‌های مختلف در این علم شده است. سطوح مختلف علم پردازش زبان طبیعی را می‌توان به صورت ذیل تقسیم‌بندی نمود[1]:

1) آواشناسی و صداشناسی[۳] که به تشخیص آواها و صداها و بازشناسی گفتار می‌پردازد؛
2) ریخت‌شناسی[۴] که به ساختارهای کلمات و ریشه‌یابی واژگان می‌پردازد؛
3) نحو[۵] که به ارتباط کلمات به همدیگر و مباحث دستوری آن‌ها در گروه‌ها و جملات می‌پردازد؛
4) معناشناسی[۶] که به ارتباطات معنایی کلمات می‌پردازد؛
5) کاربردگرایی[۷] که کاربردهای زبان برای رساندن یک مطلب به مخاطب یا مخاطبان، در حالت عملی و یا در نوشتار و گفتار طبیعی می‌پردازد؛
6) مباحثه[۸] که به ارتباطات کلی یک زبان فرای یک یا چند جمله خاص می‌پردازد.

---

[۱] Artificial Intelligence (AI)
[۲] Natural Language Processing (NLP)
[۳] Phonetics and Phonology
[۴] Morphology
[۵] Syntax
[۶] Semantics
[۷] Pragmatics
[۸] Discourse

از کاربردهای اصلی پردازش زبان طبیعی می‌توان مواردی چون خلاصه‌سازی خودکار[۱]، کمک به خواندن زبان‌های طبیعی دیگر[۲]، کمک به نوشتن به زبان‌های طبیعی دیگر[۳]، استخراج اطلاعات[٤]، بازیابی اطلاعات[٥]، ترجمه‌ی ماشینی[٦]، تشخیص واحدهای اسمی[۷]، تولید زبان طبیعی[۸]، فهم زبان طبیعی[۹]، نویسه‌خوان نوری[۱۰]، تحلیل مرجع‌دارها[۱۱]، سامانه‌ی پرسش و پاسخ[۱۲]، تشخیص گفتار[۱۳]، مبدل متن به گفتار[۱٤]، نظام‌های مکالمه گفتاری[۱٥]، ساده‌سازی متن[۱٦] و تایید متن[۱۷] اشاره نمود.

یکی از نیازهایی که با همگانی شدن کاربری رایانه‌ها در جوامع بشری احساس شد؛ نیاز به ویرایش‌گرهای املایی، دستوری و معنایی برای متون مورد استفاده‌ی کاربران بوده است. از آن‌جایی که روزانه بر تعداد سندهای متنی رایانه‌ای افزوده و زمان بسیار زیادی صرف ویرایش نحوی و معنایی این مستندات می‌شود، نیاز است که برای هر زبانی ابزارهایی به وجود بیاید که با استفاده از آن ابزارها بتوان به ویرایش هوش‌مند متون پرداخت. یکی از ابزارهایی که برای بسیاری از زبان‌ها امروزه بسیار معمول و متداول شده، ابزارهای خطایاب نحوی[۱۸] است. ابزارهای خطایاب نحوی ضمن یافتن خطاهای نحوی به تصحیح خودکار این

---

[۱] Automatic Summarization

[۲] Foreign language reading aid

[۳] Foreign language writing aid

[٤] Information Extraction

[٥] Information Retrieval

[٦] Machine Translation

[۷] Name Entity Recognition

[۸] Natural language generation

[۹] Natural language Understanding

[۱۰] Optical Character Recognition - OCR

[۱۱] Anaphora Reservation

[۱۲] Question Answering System

[۱۳] Speech Recognition

[۱٤] Text-to-Speech

[۱٥] Spoken Dialogue System

[۱٦] Text Simplification

[۱۷] Text Proofing

[۱۸] Grammar Checking and Correction

خطاها نیز می‌پردازند. از لحاظ رده‌بندی شش‌گانه‌ی پردازش زبان طبیعی این ابزارها در رسته‌ی نحو قرار می‌گیرند.

متأسفانه در زبان فارسی هنوز گامی جدی برای توسعه‌ی چنین نرم‌افزارهایی برداشته نشده است. به دلیل این که پیش‌زمینه‌ی اصلی برای خطایابی نحوی ایجاد تجزیه‌گرهای نحوی است؛ در این نوشته ضمن پرداختن به چالش‌های پیش رو، علاوه بر پرداختن به روش‌های مرسوم برای خطایابی نحوی، به روش‌های مرسوم تجزیه‌ی نحوی هم پرداخته شده است.

برای ایجاد یک خطایاب نحوی مناسب برای زبان فارسی علاوه بر پرداختن به مسائل محاسباتی در زمینه‌ی پردازش زبان‌های طبیعی، نیاز است که در مورد ویژگی‌های خاص زبان فارسی هم اطلاع داشته باشیم. به دلیل این‌که عمده‌ی خصوصیات خط فارسی از خط عربی گرفته شده است و مشکلات بسیاری در زمینه‌ی رمزگذاری و خط رایانه‌ای در خط عربی وجود دارد؛ در متن حاضر علاوه بر بررسی روش‌های ممکن برای خطایابی نحوی، نخست به چالش‌های پیرامون پردازش متون فارسی پرداخته‌ایم. در ادامه نیز به جزئیاتی در مورد ابزارهای و روش‌های تجزیه‌ی نحوی زبان و رویکردهای مختلف در مورد مسائل پردازشی انسانی پرداخته شده است.

2. مسائل و چالش های پردازش متن فارسی

در پردازش متون زبان طبیعی با زبان نوشتاری سر وکار داریم. این مسئله باعث می‌شود اگر چه به جهت از دست دادن اطلاعات گویشی مانند لحن گوینده، آهنگ صدا، تاکید و مکث، با مشکلات و ابهاماتی مواجه شویم، ولی در مقابل با شکل محدودتری از زبان کار می‌کنیم. بسیاری از بی‌ترتیبی‌های زبان متعلق به زبان گفتاری است و در زبان نوشتاری بیشتر قالب‌های دستوری رعایت می‌شوند و لذا تهیه دستور زبان پوشاننده‌ی تمام متن ساده‌تر است.

در تلاش برای ساخت یک سامانه‌ی پردازش و درک متون فارسی با مسائل و مشکلاتی مواجه می‌شویم که بعضی در بیشتر زبان‌ها بروز کرده و برخی خاص زبان فارسی می‌باشند. هم‌چنین برخی از این پیچیدگی‌ها به طبیعت زبان و نارسایی‌های قواعد زبان‌شناسی مربوط و برخی دیگر برخاسته از مشکلات ایجاد سامانه‌های هوش مصنوعی است. برخی از مشکلات به شرح ذیل هستند[2]:

2-1. مواجهه با چندمعنایی و چندنقشی‌بودن کلمات

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

2-2. حذف کلمات و عبارات به قرینه‌ی لفظی یا معنوی

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

2-3. استفاده از افعال مرکب، اصطلاحات و ضرب المثل‌ها

در زبان‌های مختلف افعال مرکب، اصطلاحات، ضرب‌المثل‌ها و استعارات موارد مشکل‌آفرین در پردازش متون هستند چرا که معمولا معنایی کاملاً متفاوت با معنای ظاهری‌شان دارند. به علاوه اجزاء چنین ترکیبی لزوماً در جمله به دنبال هم ظاهر نمی‌شوند و ممکن است بین آنها کلمه یا حتی جمله‌ی دیگری واقع شود و این امر یافتن و پردازش سازه‌ی مرکب در کل جمله را مشکل می‌کند.

2-4. تعیین طبقه‌ی اسامی

گاهی اسامی برخلاف ویژگی‌های ظاهری، طبقه‌ی خود را تغییر می‌دهند. مثلاً در جمله‌ی « خاک‌ها را بِریز توی باغچه» کلمه خاک‌ها گر چه علامت جمع دارد ولی تعدّد و شمارش را القاء نمی‌کند بلکه بیشتر به مفهوم نکره، جنس و کلیت اشاره دارد. همچنین تشخیص اسامی عام و خاص در کاربردهای مختلف آنها ممکن است ساده نباشد. مانند استفاده از اسامی خاص در نقش عام (« ایران سرزمین مولوی‌هاست») و یا اسامی عام در نقش خاص (کلمه پروانه به عنوان اسم دختران) و هم‌آوایی اسامی خاص در اطلاق به بیش از یک مصداق (اسم «حافظ» برای اطلاق به یک شاعر و یک خیابان).

2-5. بی‌ترتیب‌بودن زبان

اگرچه فارسی دارای ترتیب مرکزی فاعل-مفعول-فعل است ولی دارای استثنائات فراوان و مکرر در ترتیب کلمات می‌باشد. این مسئله باعث می شود ساخت دستور زبان مدون و محاسباتی برای زبان و در نتیجه تجزیه و تحلیل نحوی جملات مشکل شود. مثلاً جمله‌ی ساده‌ی «دیروز من کتاب را در مدرسه به مریم دادم» می‌تواند به انواع اشکال مختلف با جابجایی متمم‌ها و قید در طول جمله نوشته شود (مانند "من دیروز کتاب را در مدرسه به مریم دادم."، "من دیروز در مدرسه کتاب را به مریم دادم."، "من کتاب را در مدرسه دیروز به مریم دادم. "و "دیروز من در مدرسه کتاب را به مریم دادم.").

2-6. کسره‌ی اضافه و حذف آن

در زبان فارسی کسره‌ی اضافه که معمولاً حذف می‌شود دارای چند نقش است. این علامت، صفت و موصوف و مضاف و مضاف‌الیه را به هم مرتبط می‌نماید. در بعضی موارد علامات «’s» و «of» در انگلیسی معادل این کسره هستند و در برخی موارد دیگر (مثل اتصال صفت و موصوف) معادلی در انگلیسی ندارد. تشخیص میان نقش‌های مختلف این علامت توسط ماشین کار ساده‌ای نیست، به خصوص زمانی که سامانه و برنامه‌مان دانش معنایی و کاربردی اندکی داشته باشد. در ضمن حذف کسره‌ی اضافه در نوشتار منجر به ایجاد مشکل در تشخیص مرزهای عبارات اسمی می‌شود.

2-7. عدم تطابق اجزاء جمله

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

2-8. وجود ساختار جملات یکسان با معانی و نقش‌های متفاوت

در زبان فارسی بسیاری نقش‌های دستوری با علامت و حرف اضافه مشابه مشخص می‌شوند. برای مثال نقش‌های همراه، حالت، ابزار، وسیله، مقابله یا معاوضه، داشتن و ... با حرف اضافه «با» ظاهر می‌شوند. لذا جملات ذیل اگر چه از لحاظ ظاهری مشابه‌اند، ولی دارای نقش‌های موضوعی متفاوتی هستند و گاهی برای تشخیص این نقش نیاز به دانش معنایی و کاربردی داریم. برای مثال "با" در جملات "علی با

ناراحتی رفت."، "علی با لباس سیاه رفت."، " علی با گربه‌اش رفت." و " علی با اسبش رفت." به ترتیب نشان‌دهنده‌ی حالت فعل، توصیف فاعل، همراه و ابزار است.

2-9. مشکلات ناشی از ابهام زبان طبیعی

از ویژگی‌های زبان طبیعی وجود ابهام در آن (حتی برای خواننده‌ی انسانی) است که برخی از انواع آن در زیر بر شمرده شده‌اند:

1) ابهام ساختاری معمولا ناشی از ابهام در دسته‌بندی کلمات در عبارات ایجاد می‌شود مانند ابهام در « زن و مردِ پیر»، «بازجویی او به درازا کشید»، «ما همه کار می کنیم»، «بطری شیرِ خشک» و «درِ باغ و خانه» .
2) ابهام واژگانی ناشی از معانی مختلف یک کلمه مثل ابهام در عبارت «زنِ خیاط».
3) ابهام ارجاعی یا ابهام در مرجع ضمیر مثل ابهام در تشخیص مرجع "او" در جملات «علی حسن را دید. کتاب او روی میز بود.» که معلوم نیست ارجاع «او» به علی است یا حسن.
4) ابهامات ناشی از حذف به قرینه‌ی لفظی یا معنایی. مثلاً در جمله‌ی «در آنجا سعدی شاعر بزرگ ایران را دید» فاعل حذف شده و تشخیص اینکه سعدی فاعل است یا مفعول بدون داشتن دانش پیش‌زمینه در مورد سعدی و موضوع بحث میسر نیست.
5) ابهام ناشی از صفت جانشین اسم مانند ابهام در عبارت «عصای سخن‌ران» که معلوم نیست که این عصا است که سخن‌رانی می‌کند و یا عصا متعلق به سخن‌ران است.
6) ابهام ناشی از محدودیت‌های نوشتاری (حذف اِعراب) در کلماتی بروز می‌کند که رسم‌الخط یکسان ولی تلفظ و معنای متفاوت دارند. این ویژگی در زبان فارسی که حرکات و اعراب حروف در نوشتار حذف می‌شوند، مشهودتر و فراوان‌تر است (مانند کلمات حُکم، حَکَم و حِکَم که دارای نوشتار یکسان و تلفظ و معنای متفاوتند.).

3. مسائل پردازش محاسباتی در پردازش متون فارسی

این مشکلات به دلیل این واقعیت پیش می‌آیند که سعی در ایجاد الگوی محاسباتی از زبان برای ماشین داریم. عمده این مشکلات عبارتند از:

1) فقدان دانش زبانی محاسباتی و مدون مانند عدم وجود واژگان، دستور زبان و الگوها و قواعد زبانی مدون و قابل درک برای ماشین در بسیاری زبان‌ها و به خصوص فارسی؛
2) عدم وجود ابزارها و لوازم پردازش زبان طبیعی برای زبان فارسی مانند تجزیه‌گرهای ساخت‌واژی، نحوی و معنایی معتبر و کارآ؛
3) فقدان دانش محاسباتی عرفی و تخصصی که برای رفع آن نیاز به در اختیار داشتن هستان‌شناسی‌های معیار عمومی و تخصصی است.

4. جمع بندی پیچیدگی‌های پردازش متون فارسی

با توجه به بحث اخیر می توان در کل اهم مشکلات فعلی پردازش متون فارسی را در چند دسته زیر خلاصه نمود:

1) عدم وجود منابع زبانی مناسب و کافی برای زبان فارسی مانند واژگان‌های تک‌زبانه و چندزبانه محاسباتی، واژگان‌های معنایی و متصل به هستان‌شناسی (هستان‌شناسی‌های لغوی)، هستان‌شناسی جامع عمومی و تخصصی، پیکره‌های عمومی و تخصصی ساده یا برچسب‌خورده (با برچسب‌های اجزاء کلام، کسره‌ی اضافه، نقش‌های موضوعی، مفاهیم و روابط مفهومی و غیره)، مجموعه مدون قوانین ساخت‌واژی و دستوری پوشا، عدم وجود معیار یکسان برای شیوه‌ی نگارش، فاصله‌گذاری و رمزگذاری حروف و علائم؛
2) مشکل تشخیص مرز کلمات (مسئله شیوه‌های نگارش متفاوت)؛
3) مشکل تشخیص مرز گروه‌های اسمی (مسئله کسره‌ی اضافه نامرئی)؛
4) از دست‌دادن اطلاعات گویشی؛
5) مسئله‌ی ابهام؛
6) افعال مرکب و اصطلاحات
7) مسئله‌ی هم‌نگاره‌ها[1] (و تحت آن مسئله حذف مصوت‌های کوتاه (اعراب) از نوشتار)؛
8) معناشناسی و مشکلات تحلیل معنایی.

[1] Homographs

5. تجزیه‌ی نحوی

برای دریافت و تفسیر جملات زبان طبیعی می‌بایست آن‌ها را از لحاظ نحوی مورد تجزیه قرار داد. حتی برای تحلیل معنایی و ترجمه‌ی ماشینی نیز نیاز به تحلیل نحوی است و در ترجمه‌ی ماشینی، بخش تحلیل و تجزیه‌ی نحوی[۱] به عنوان اصلی‌ترین واحد پیش‌پردازشی عملیات به حساب می‌آید. در مورد خطایاب‌های نحوی هم، رویکردهای متفاوتی وجود دارد. ولی آن‌چیزی که واضح و مبرهن است این است که به دلیل وجود ابهام و امکان وجود چندین تجزیه برای یک عبارت یا جمله در زبان طبیعی، نمی‌توان از روش‌های مرسوم و معمولی تجزیه که در زبان‌های برنامه‌سازی رایانه‌ای استفاده می‌شود، استفاده کرد.

5-1. برچسب‌گذاری ادات سخن

اوّلین قدم مهم در پردازش نحوی یک متن، اختصاص دادن ادات سخن یا نقش‌های دستوری به تک‌تک اجزا یا واژگان است. به این کار برچسب‌گذاری ادات سخن گفته می‌شود. به دلیل خاصیت ترکیبی برخی از واژگان در زبان‌های طبیعی و مسائلی از قبیل پیوسته‌نویسی به جای برچسب‌گذاری واژگان، به واحدها[۲] برچسب نسبت داد. علاوه بر این‌ها، می‌توان از برچسب‌گذاری در سامانه‌های بازیابی اطلاعات[3] استفاده کرد[3]. به سامانه‌های برچسب‌گذاری خودکار ادات سخن CLAWS[٤] گفته می‌شود[4].

از بین مسائل مطرح‌شده در برچسب‌دهی مسائلی که از ساخت‌واژه زبان فارسی ناشی می‌شوند مهم‌تر است، زیرا ساخت‌واژه علاوه بر شکل کلمه، برچسب کلمه را نیز تحت تاثیر قرار می‌دهد. ساخت‌واژه زبان فارسی باعث می‌شود کلمات با اشکال متفاوت از یک بن‌واژه یکسان ایجاد شوند که برچسب آن‌ها نیز در پیکره متفاوت خواهد بود و این امر باعث می‌شود تعداد برچسب‌های متمایز در پیکره بسیار زیاد شود.

---

[۱] Grammatical Parsing

[۲] Tokens

[۳] Information Retrieval (IR)

[٤] Constituent-Likelihood Automatic Word Tagging System

هنگامی که یک متن برچسب‌نخورده به سامانه‌ی برچسب‌گذاری داده می‌شود، اولین گام تشخیص کران جملات آن است. زیرا ورودی برچسب‌گذارها جمله است و اکثر برچسب‌گذارها برچسب‌گذاری را در واحد جمله انجام می‌دهند. تشخیص کران کلمات نیز بسیار حائز اهمیت است.

برچسب‌گذاری پیکره‌های زبانی به طور کلی می‌تواند در چهار سطح زبانی انجام شود که عبارتند از[5]:

1) تعیین برچسب مقوله‌ی کلمه[1]؛
2) برچسب‌گذاری نحوی: که شامل پردازش جملات و به دست آوردن درخت نحوی آن‌ها است.
3) برچسب‌گذاری معنایی: که عبارت است از استخراج صورت منطقی جملات و به دست آوردن تعبیر معنایی آن‌ها. پیش‌نیاز این نوع برچسب‌دهی، انجام برچسب‌دهی مقوله‌ی کلمات است، به این تعبیر که پیش از آن‌که تعبیر معنایی کلمات و جملات یک متن به دست داده شود، تعیین مقوله کلمات آن‌ها ضروری است؛ و
4) برچسب‌گذاری کاربردشناختی: که عبارت است از تعیین روابطی که میان دو کلمه در یک متن وجود دارد. به عنوان مثال مشخص کردن ضمایر و مرجع آن‌ها در متن در حوزه برچسب‌دهی کاربردشناختی قرار می‌گیرد.

برچسب‌هایی که برای یک پیکره در نظر گرفته می‌شود، به طور کلی به سه دسته برچسب تقسیم می‌شوند که عبارتند از[6]:

1) برچسب‌های نحوی-ساخت‌واژی[2]: که اصلی‌ترین برچسب‌ها هستند. این برچسب‌ها شامل مقوله‌های نحوی اصلی از جمله فعل، اسم، صفت، قید و غیره هستند. اغلب کلماتی که در متون وجود دارند به یکی از این مقوله‌های اصلی تعلق دارند. اصلی‌ترین مقوله‌های نحوی که در اغلب پیکره‌های زبانی در نظر گرفته می‌شوند شامل مقوله اسم، صفت، حرف اضافه، حرف ربط، حرف تعریف، قید و عدد هستند.

[1] Word-class tagging

[2] Morphosyntactic

2) برچسب‌های خاص[1]: که شامل واژگانی هستند که در طبقه‌ی مقوله‌های اصلی قرار نمی‌گیرند. امّا تعیین برچسب آن‌ها در استخراج اطلاعات زبانی از پیکره حائز اهمیت است. ادات شرط، حرف ندا، تکواژ صفت‌ساز از این دسته‌اند.

3) برچسب‌های متفرقه[2]: شامل کلماتی است که در طبقه‌ی مقوله‌های اصلی قرار نمی‌گیرند. امّا در متن وجود دارند و به عنوان واژگان مجزا استخراج شده‌اند. واژگان خارجی، نشانه‌ها و علائم ریاضی و نیز علائم اختصاری در این طبقه قرار می‌گیرند.

برچسب‌های پیکره متنی زبان فارسی نیز در سه دسته تقسیم‌بندی می‌شوند[5]:

- برچسب‌های نحوی-ساخت‌واژی: مانند اسم، فعل، صفت، قید، حرف ربط، حرف اضافه، حرف تعریف، ضمیر؛
- برچسب‌های خاص: مانند ادات شرط، کیفیت نما، کلمه پرسشی، جمله‌واره، حرف ندا، منادی، تکواژ صفت‌ساز و عربی؛ و
- برچسب‌های متفرقه: جداکننده، علامت ریاضی.

ساختار در نظر گرفته شده برای برچسب کلمات در پیکره‌ی متنی ساختار سلسله‌مراتبی است. در ساختار سلسله‌مراتبی، تمایز میان طبقات اصلی و زیربخش‌های آن‌ها نشان داده می‌شود. به این ترتیب که اگر برچسب از سمت راست به چپ خوانده شود، اصلی‌ترین برچسب در سمت راست قرار دارد و از راست به چپ جزئیات و اطلاعات دقیق‌تر به صورت زیربخش‌هایی به برچسب اصلی افزوده می‌شود. هر یک از طبقات اصلی و زیربخش‌های آن‌ها نیز به وسیله یک ویرگول یا نقطه از یک‌دیگر مجزا می‌شوند[5]. در پیکره‌ی متنی زبان فارسی هر مقوله یا برچسب به وسیله ویرگول از زیربخش‌های خود مجزا می‌شود. به عنوان مثال برچسب «اسم,عام,مفرد,مکان,کسره اضافه» نشان می‌دهد که این برچسب به کلماتی منتسب می‌شود که مقوله‌ی آن‌ها اسم است، نوع آن‌ها عام و به لحاظ شمار مفرد هستند و به لحاظ معنایی در دسته اسم‌های مکان قرار می‌گیرند و نیز یک نشانگر نحوی کسره اضافه دارند؛ چه این کسره‌ی اضافه به صورت آشکار در خط ظاهر شود و چه کسره‌ی اضافه در خط ظاهر نشود.

---

[1] Unique tags

[2] Residual /Miscellaneous tags

یک مشکل اساسی که در سامانه‌های آماری به وجود می‌آید، مشکل پراکندگی داده[1] است. این مشکل تأثیر زیادی بر کارآیی سامانه‌ها دارد. برای حل این مشکل روش‌هایی به نام روش‌های هموارسازی[2] استفاده می‌شود.

5-1-1. انواع برچسب‌گذاری ادات سخن

1) **برچسب‌گذاری قانون‌محور**[3]: در این روش از برچسب‌گذاری دو مرحله‌ی زیر را خواهیم داشت:
   - به هر واژه با توجه به واژه‌نامه‌ی موجود، فهرستی از برچسب‌های ممکن تعلق می‌گیرد؛
   - با استفاده از فهرست بزرگی از قوانین ابهام‌زدایی که به صورت دستی جمع‌آوری شده است، تعداد برچسب‌ها برای هر واژه به یک برچسب تقلیل می‌یابد.
2) برچسب‌گذاری تصادفی[4]: در این روش، از روش‌های احتمالی و آماری استفاده می‌شود. معروف‌ترین روش برای برچسب‌گذاری تصادفی استفاده از الگوی پنهان مارکوف[5] است. به برچسب‌گذاری که با الگوی پنهان مارکوف کار می‌کند، برچسب‌گذاری مارکوفی گفته می‌شود.
3) برچسب‌گذاری انتقال‌محور[6]: در این برچسب‌گذاری ، از روش یادگیری انتقال‌محور[7] استفاده می‌شود. در یادگیری انتقال‌محور سه مرحله‌ی زیر انجام می‌شود:
   - به هر واژه برچسبی که دارای بیش‌ترین احتمال است، تعلق می‌گیرد؛
   - هر انتقال ممکن در بین نشانه‌ها مورد آزمون قرار می‌گیرد و انتقالی که باعث بهبود در برچسب‌گذاری می‌شود، گزینش می‌شود؛ و

---

[1] Data Sparsity
[2] Smoothing methods
[3] Rule-based Tagging
[4] Stochastic Tagging
[5] Hidden Markov Model (HMM)
[6] Transformation-based Tagging
[7] Transformation-Based Learning (TBL)

- با استفاده انتقال‌های صورت‌گرفته و قوانین موجود برچسب‌گذاری واژگان بازبرچسب‌گذاری[1] می‌شوند.

5-1-2. انواع برچسب و ادات سخن و مسائل مربوط به آن‌ها

ادات سخن در دو رده‌ی اصلی زیر قرار می‌گیرند:

1) رده‌ی بسته[2]: این رده تعدادشان محدود است و عمده‌ی واژه‌های کارکردی[3] مانند حروف اضافه و ضمایر را شامل می‌شوند؛ و
2) رده‌ی باز[4]: تعداد این رده نامحدود بوده و هر روز از زبان‌های دیگر وارد زبان می‌شوند یا در همان زبان با اشتقاق و یا واژه‌سازی به زبان اضافه می‌شوند. مهم‌ترین نمونه‌ها از رده‌های باز اسم‌ها، افعال، صفات و قیدها هستند. اسم‌ها هم به دو گونه‌ی اسم خاص و اسم عام و از لحاظی دیگر به اسم‌های قابل شمارش و اسم‌های غیرقابل شمارش تقسیم می‌شوند. انواع مختلفی هم از قیدها وجود دارد که مهم‌ترین آن‌ها قیدهای مکان، درجه، رفتار و زمان هستند.

با توجه به انواع ادات‌سخن، مجموعه‌ای از برچسب‌ها به وجود می‌آید. به عنوان نمونه پیکره‌ی متنی براون[5] دارای 87 برچسب است. مسئله‌ی مهمی که در برچسب‌گذاری ادات سخن حائز اهمیت است، حل مشکل ابهام در زبان و پیدا کردن نشانه‌ی مناسب برای هر واژه است. در نتیجه، برچسب‌گذاری علاوه بر یک ابزار تحلیل نحوی، به عنوان یک ابزار مناسب برای ابهام‌زدائی به شمار می‌آید. به عنوان مثال در پیکره‌ی متنی براون حدود 11.5% از واژگان دارای ابهام هستند. طبق تحقیقاتی که در زمینه‌ی روش‌های رفع ابهام به عمل آمده است، 40% از ابهام‌ها به سادگی قابل رفع هستند[3].

یکی از مشکلات در روش‌های برچسب‌گذاری با استفاده از روش‌های احتمالی وجود واژگانی است که تنها یک بار در پیکره متنی رخ داده‌اند. به این واژه‌ها hapax legomenon گفته می‌شود. با استفاده از

---

[1] Re-Tag

[2] Closed Class Types

[3] Function Words

[4] Open Class Types

[5] Brawn Corpus

روش n-نگاشت رده‌محور[1] می‌توان روش‌های احتمالی برچسب‌گذاری را بهبود داد. یکی از ساختمان داده‌هایی که در این روش بسیار به کار می‌آید، استفاده از ماتریسی به نام ماتریس اغتشاش[2] است. این ماتریس، یک ماتریس دوبعدی است که در هر بعد نشانه‌ها مقادیر برچسب‌ها وجود دارد. هر درایه از این ماتریس نشانه‌ی احتمال حضور هر نشانه پس از برچسب دیگر است[7].

5-1-3. استفاده از الگوی پنهان مارکوف در برچسب‌گذاری ادات سخن

هدف غایی از یک سامانه‌ی برچسب‌گذار، استفاده از آن در راستای هدفی مشخص است که باید پس از آموزش صورت گیرد. الگوریتم ویتربی[3] [8] یک الگوریتم رمزگشایی[4] است که برای استفاده در سامانه‌های مبتنی بر الگوهای پنهان مارکوف[5] [9] ارائه شده است. الگوهای پنهان مارکوف بیشتر به دو روش استفاده می‌شوند. اوّل این‌که احتمال یک دنباله از خروجی‌ها برای چندین الگو محاسبه می‌شود تا مشخص شود کدام الگو احتمال بیشتری برای تولید این دنباله را از خروجی‌ها دارد. یک مثال از این مورد تشخیص گفتار است که در آن یک سیگنال صوتی با یک الگو مقایسه می‌شود تا مشخص شود که چه کلمه‌ای گفته شده است. در روش دوم، از الگو برای تعیین این‌که کدام مسیر برای تولید یک دنباله‌ی خروجی خاص طی شده است، استفاده می‌شود. این همان روشی است که از الگوی پنهان مارکوف برای برچسب‌گذاری ادات سخن استفاده می‌شود. در برچسب‌گذاری با الگوی پنهان مارکوف[10, 11] دنباله‌ی برچسب‌ها در یک متن به عنوان یک زنجیر مارکوف[6] در نظر گرفته می‌شود. یک زنجیر مارکوف دارای دو خاصیت افق محدود و مستقل از زمان بودن است. تفسیر این دو خاصیت در برچسب‌گذاری با الگوی پنهان مارکوف به این صورت است که ما فرض می‌کنیم

---

[1] Class-Based N-Grams

[2] Confusion Matrix

[3] Viterbi Algorithm

[4] Decoding

[5] Hidden Markov Models (HMMs)

[6] Markov Chain

برچسب یک واژه تنها وابسته به برچسب واژه‌ی قبلی است (افق محدود) و این وابستگی در طول زمان تغییر نمی‌کند (مستقل از زمان بودن).

اگرچه پر واضح است که این دو خاصیت مارکوف چندان منطبق بر واقعیت نیست. زیرا به عنوان مثال خاصیت اوّل وابستگی‌های با فاصله زیاد را بین برچسب واژگان نادیده می‌گیرد.

. فرض می‌کنیم که $\{w^1, w^2, \mathbf{K}, w^w\}$ یک مجموعه از واژگان و $\{t^1, t^2, \mathbf{K}, t^t\}$ یک مجموعه از برچسب‌های ممکن برای آن‌ها است. با فرض یک دنباله از کلمات از مجموعه کلمات، $w_{1,n}$، هدف یافتن محتمل‌ترین دنباله از برچسب‌ها از مجموعه برچسب‌ها، $t_{1,n}$ است. با به کار بردن قانون بیز[1] می‌توان نوشت:

$$\arg\max_{t_{1,n}} P(t_{1,n} \mid w_{1,n}) = \arg\max_{t_{1,n}} \frac{P(w_{1,n} \mid t_{1,n})P(t_{1,n})}{P(w_{1,n})} = \arg\max_{t_{1,n}} P(w_{1,n} \mid t_{1,n})P(t_{1,n})$$

خاصیت افق محدود به صورت زیر بیان می‌شود:

$$P(t_{i+1} \mid t_{1,i}) = P(t_{i+1} \mid t_i)$$

رابطه‌ی فوق همان احتمالات انتقال می‌باشد. علاوه بر فرض افق محدود دو فرض دیگر راجع به کلمات در نظر می‌گیریم:

1) کلمات از یکدیگر مستقل‌اند؛ و
2) یک کلمه مانند w تنها وابسته به برچسب خودش می‌باشد.

بنابراین، بر طبق مفروضات فوق می‌توان نوشت:

$$P(w_{1,n} \mid t_{1,n})P(t_{1,n}) = \prod_{i=1}^{n} [P(w_i \mid t_i) \times P(t_i \mid t_{i-1})]$$

[1] Bayes Rule

در عمده‌ی موارد الگوی مرتبه‌ی اوّل مارکوف[1] برای انتخاب درست بهترین پی‌رفت از برچسب‌ها از یک متن مناسب است[12]. نتایجی که به در پژوهشکده‌ی رایانه‌ای دانشگاه لنکستر انگلستان[2] با بیش از 90% صحّت نتیجه به دست آمده بود، گواه این مدّعا بوده است. برای حالت‌های خاص می‌توان از الگوی مرتبه‌ی دوم مارکوف (3 برچسب پشت هم) یا بالاتر که فهرست کوچک‌تری را تشکیل می‌دهند؛ استفاده کرد[12].

5-1-4. برخی از کارهای انجام‌شده در مورد برچسب‌گذاری ادات سخن در زبان فارسی

مگردومیان[13] با اشاره به دو رویکرد در برچسب‌گذاری زبان فارسی، توضیحاتی در مورد روند توسعه‌ی سامانه‌ی برچسب‌گذاری داده است. این دو رویکرد عبارتند از: آماری و نمادین[3]. در رویکرد نمادین از دانش از پیش آماده، استفاده می‌شود. در برچسب‌گذاری احتمالی از روی پیکره‌ی یادگیری[4]، ماتریس احتمالات برای نقش‌های مختلف ساخته می‌شود. در برچسب‌گذاری احتمالی از روش‌های تحلیل آماری-احتمالی استفاده می‌کنند ولی نیاز به این است که با پیکره‌هایی که از پیش برچسب‌گذاری شده‌اند، یادگیری شوند. یکی از مشکلاتی که او در مقاله‌اش به آن‌ها اشاره کرده بود، فاصله‌گذاری‌های نادرست در واحدهای زبان فارسی است. مثلاً کاربر به اشتباه بنویسد"رفتندمردم" به جای "رفتند مردم". واحدهای پیچیده‌ای مانند "بشیوه" نیز وجود دارند که کار برچسب‌گذاری با وجود آن‌ها مشکل می‌شود[13]. هم‌چنین مگردومیان[14] با استفاده از ابزار حالت متناهی زیراکس[5] برنامه‌ای دو مرحله‌ای برای تحلیل ریخت‌شناسی زبان فارسی ارائه کرده است. در این برنامه، واژگان با یک ماشین انتقال حالت متناهی[6] تفسیر و تحلیل می‌شوند.

---

[1] First Order Markov Model

[2] Unit for Computer research on the English Language (UCREL)

[3] Symbolic

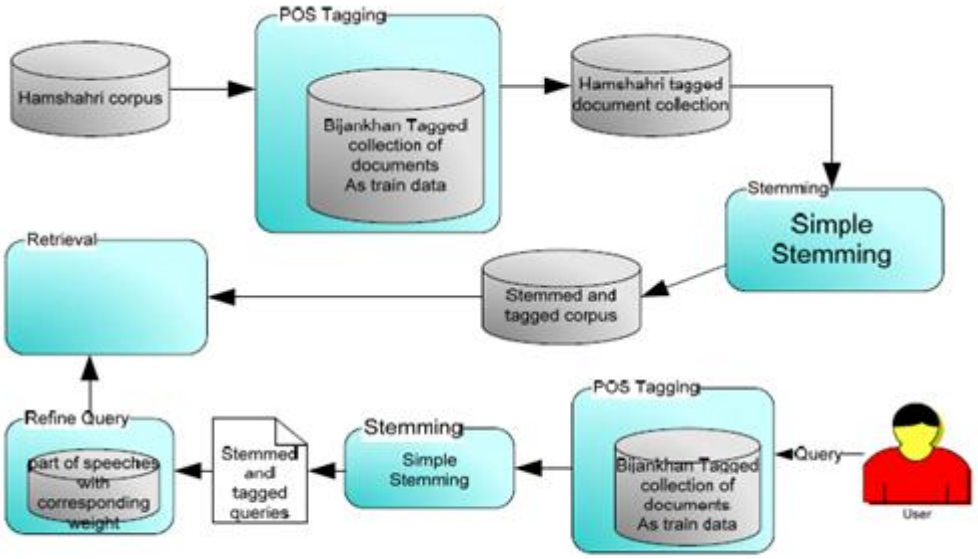
[4] Training corpus

[5] Xerox Finite State Tools

[6] Finite State Transducer (FST)

عاصی و حاجی‌عبدالحسینی[15] بر اساس پایگاه داده‌ی زبان‌شناسی فارسی[1] تهیه‌شده در فرهنگستان علوم انسانی به بررسی برچسب‌گذاری بر روی این پایگاه دانش پرداخته‌اند. شایان ذکر است که این پایگاه دانش، اوّلین پروژه‌ی رسمی برای تهیه‌ی پیکره‌ی متنی زبان فارسی بوده است[16]. عظیمی‌زاده و همکارانش[17] نیز بر اساس پیکره‌ی متنی بیجن‌خان[18] که در دانشگاه تهران تهیه شده است، اقدام به توسعه‌ی برچسب‌گذاری با استفاده از الگوی پنهان مارکوف کرده‌اند. البته پیکره‌ی متنی‌ای که آن‌ها استفاده کرده بودند، نسخه‌ی اصلی پیکره‌ی متنی بیجن‌خان نبوده است و تغییراتی که اورمچیان و همکارانش[19, 20] بر آن پیکره وارد کرده بودند، در پیکره‌ی مورد استفاده‌شان وجود داشته است. هم‌چنین عظیمی‌زاده و همکارانش[21] یک تحلیل‌گر ریخت‌شناسی را با استفاده از برچسب‌گذار ادات سخن برای زبان فارسی ساختند.

کریمی‌پور و همکارانش[22] با استفاده از پیکره‌های متنی بیجن‌خان[18] و همشهری[23]، به بررسی تأثیر روش‌های برچسب‌گذاری بر بازیابی اطلاعات پرداخته‌اند. شکل زیر نمایی از روند بازیابی اطلاعات را نشان می‌دهد:



شکل 1- معماری سامانه‌ی برچسب‌گذار زبان فارسی

[1] The Farsi Linguistic Database (FLDB)

رجا و همکارانش[24] به ارزیابی یک برچسب‌گذار بر مبنای پیکره‌ی بیجن‌خان [18] پرداخته‌اند. این پیکره دارای 550 نوع برچسب متمایز است که به خاطر ناکارآمد بودن این تعداد برچسب بر برچسب‌گذاری خودکار، برخی از این برچسب‌ها حذف شده‌اند[25]. آمتروپ و همکارانش[26] در سال 1997 نسخه‌ی اوّلیه سامانه‌ی مترجم شیراز را در دانشگاه ایالتی نیومکزیکو امریکا ارائه کردند که در آن از برچسب‌گذاری ادات سخن استفاده شده است.

6. مشکلات موجود در تحلیل نحوی با استفاده از دستور زبان‌های مستقل از متن

از مسائلی که در تحلیل نحوی با استفاده از دستور زبان مستقل از متن[1] ممکن است به وجود بیاید، می‌توان به موارد ذیل اشاره کرد:

1) رابطه‌ها بین واحدهای زبانی[2]؛
2) زیررسته‌های زبانی[3]؛ و
3) وابستگی‌ها[4].

با استفاده از دستور زبان‌های مستقل از متن می‌توان به تجزیه‌ی نحوی متون زبانی پرداخت. در نتیجه می‌توان درخت تجزیه‌ای برای هر عبارت یا جمله ساخت. به دستور زبان‌هایی که با استفاده از زبان‌های صوری[5] به مدل کردن زبان‌های طبیعی می‌پردازند، دستور زبان‌های مولد[6] گویند.

[1] Context-Free Grammars (CFGs)
[2] Constituency Grammatical Relations
[3] Subcategorization
[4] Dependencies
[5] Formal Languages
[6] Generative Grammars

7. تجزیه‌ی نحوی با استفاده از دستور زبان‌های مستقل از متن

با استفاده از برچسب‌گذاری و ایجاد درخت تجزیه‌ی نحوی می‌توان به تحلیل نحوی متون پرداخت. جمله‌ای که قابل تجزیه نباشد، دارای خطای نحوی خواهد بود و وظیفه‌ی خطایاب‌های نحوی پیدا کردن این خطاها و ارائه‌ی تصحیح‌های نحوی برای جملات نادرست است. علاوه بر این‌ها ایجاد درخت تجزیه، یکی از مراحل مهم میانی برای تحلیل معنایی جملات است. مشکل بزرگی که در تجزیه‌ی نحوی بسیار شایع است، وجود ابهام در تجزیه‌ی جملات است. ابهام را می‌توان این‌گونه معنا کرد که اگر برای جمله‌ای بتوان بیش از یک درخت تجزیه ایجاد نمود، آن جمله دارای ابهام نحوی است.

در مجموع دو رویکرد عمده در تجزیه وجود دارد:

1) تجزیه‌ی بالا به پایین (تجزیه‌ی هدف‌محور)[1]
2) تجزیه‌ی پایین به بالا (تجزیه‌ی داده‌محور)[2].

در تجزیه‌ی بالا به پایین، هیچ‌گاه زمان صرف ایجاد درخت‌های تجزیه‌ای که به هیچ وجه امکان تجزیه‌ی جمله‌ی مورد نظر را ندارند، نمی‌شود. در تجزیه‌ی پایین به بالا، درخت‌هایی که هیچ احتمالی برای تجزیه ندارند، ساخته نمی‌شوند. از جمله مشکلاتی که در روش تجزیه‌ی بالا به پایین وجود دارد، می‌توان به موارد ذیل اشاره کرد:

1) بازگشتی چپ[3]؛
2) ابهام[4]؛ و

---

[1] Top-Down Parsing (Goal-Directed Parsing)

[2] Bottom-UP Parsing (Data-Directed Parsing)

[3] Left Recursion

[4] Ambiguity

3) نمایش غیربهینه‌ی درخت تجزیه.

برای حل مشکل بازگشتی چپ دو راه عمده وجود دارد. راه اوّل این است که دستور زبان را بازنویسی کنیم و قوانینی را که در آن می‌توانند موجب ایجاد بازگشتی چپ شوند، حذف نماییم. راه دوم این است که عمق درخت تجزیه را در هر مرحله از تجزیه محدود کرده و نگذاریم عمق درخت تجزیه از حدی بیشتر شود.

عمده‌ترین نوع ابهام در تجزیه‌ی نحوی، ابهام ساختاری است. ابهام ساختاری به این معناست که برای یک جمله بیش از یک تجزیه ممکن وجود داشته باشد. سه رده از ابهام وجود دارد:

1) ابهام پیوستگی[1]: اگر واحد زبانی خاصی را بتوان بیش از یک بار در یک مکان از جمله به درخت تجزیه درج کرد، جمله دارای ابهام پیوستگی خواهد بود؛
2) ابهام تطبیقی[2]: وقتی مجموعه‌ی مختلفی از عباراتی که قابل اتصال با عطف به هم هستند؛ وجود داشته باشد، ابهام تطبیقی به وجود خواهد آمد؛ و
3) ابهام در عبارات اسمی[3]: اگر درخت تجزیه را با پرانتزگذاری بتوان نشان داد، برای عبارات اسمی با n واژه به تعداد عدد کاتالان حالت تجزیه وجود خواهد داشت.

نوع دیگری از ابهام نیز وجود دارد که به آن ابهام محلی[4] گویند. در این نوع از ابهام، قسمت‌هایی از یک جمله دارای ابهام است.

معروف‌ترین روش برای تجزیه، الگوریتم اِرلی[5] است که در این روش با استفاده از برنامه‌نویسی پویا برای جمله‌ای با N واژه با پیچیدگی زمانی $O(N^3)$ می‌توان جمله‌ای را پویش کرد.

---

[1] Attachment Ambiguity

[2] Coordination Ambiguity

[3] Noun-Phrase Ambiguity

[4] Local Ambiguity

[5] Early Algorithm

7-1. ساختارهای خصیصه و یکسان‌سازی

می‌توان با قوانین نحوی به صورت اشیاء روبرو شد که این اشیاء دارای مجموعه‌های پیچیده‌ای از ویژگی‌هایی هستند که این قوانین دارای چنین ویژگی‌هایی می‌باشند. اطلاعاتی که در این ویژگی‌ها نهفته است با محدودیت‌ها[1] نشان داده می‌شوند. به چنین الگوهایی، الگوی صوری‌گرایی محدودیت‌محور[2] گفته می‌شود. در این زمینه ساختارهایی تعریف می‌شوند که به این ساختارها، ساختارهای خصیصه[3] گفته می‌شود. این ساختارهای مجموعه‌هایی هستند از مقادیر چندتایی خصیصه‌ها که این خصیصه‌ها به صورت یک‌پارچه‌ی غیرقابل تجزیه هستند، تشکیل می‌شود. ساختار خصیصه به صورت ماتریسی به نام ماتریس مقادیر ویژگی[4] نمایش داده می‌شود. مسیر خصیصه[5] فهرستی است از خصیصه‌هایی در یک ساختار خصیصه است که با این خصیصه‌ها می‌توان به مقادیر خاصی رسید. به روش یک‌پارچه‌سازی دانش از محدودیت‌های متفاوت، یکسان‌سازی[6] گویند. هدف از استفاده از ساختار خصیصه در دستور زبان‌ها عبارتند از:

1) استفاده از ساختارهای خصیصه‌ی پیچیده با موارد واژگانی و نمونه‌های نحوی؛ و
2) به عنوان راهنما برای ترکیب ساختارهای خصیصه برای تولید محدودیت‌های دستوری، برای پیدا کردن سازگاری محدودیت‌ها بین اجزای مشخصه از ساختارهای دستوری.

یکی از مشکلاتی که در یکسان‌سازی به وجود می‌آید، مشکل وابستگی در فواصل زیاد[7] است. یکی از راه‌های حل این مشکل، نگهداری فهرست‌هایی با نام فهرست فاصله[8] است که به صورت فاصله‌ی بین خصیصه‌ها می‌باشد که از هر عبارت به عبارت دیگر در یک درخت تجزیه انتقال می‌یابد. ورودی مورد نیاز

---

۱ Constraints
۲ Constraint-Based Formalism
۳ Feature Structure
٤ Attribute-Value Matrix (AVM)
٥ Feature Path
٦ Unification
۷ Long-Distance Dependency
۸ Gap List

برای یکسان‌سازی یک گراف جهت‌دار بدون دور[1] از ساختارهای خصیصه است. خصیصه‌ها به صورت برچسب بر روی یال‌های جهت‌دار گراف هستند و ارزش این خصیصه‌ها می‌تواند نمادهای تجزیه‌ناپذیر یا یک گراف جهت‌دار بدون دور باشد. الگوریتم یکسان‌سازی به صورت یک حلقه‌ی تکرار در بین خصیصه‌ها در ورودی می‌گردد و به دنبال پیدا کردن هماهنگی بین خصیصه‌ها می‌رود. هر ساختار خصیصه دارای دو مقدار است. یکی مقدار محتوا[2] است و دیگری مقدار اشاره‌گر[3] است. مشکلی که امکان دارد در این الگوریتم به وجود بیاید، امکان گیرافتادن در دور است. برای پرهیز از ایجاد دور در الگوریتم استفاده از بررسی رخداد[4] است. این بررسی ورودی گراف جهت‌دار بدون دور را تحلیل می‌کند و اگر یکی از شناسه‌ها دارای زیرقسمتی از دیگری بود، خطا بر می‌گرداند. در عمل، به دلیل سربار محاسباتی رایانه‌ای این بررسی معمولاً پیاده‌سازی نمی‌شود. ساختارهای خصیصه‌ی پایه دارای دو مشکل اساسی هستند: اوّلین مشکل این است که هیچ راهی برای قرار دادن محدودیت‌ها بر روی ارزش خصیصه‌ها وجود ندارد. راه حل‌هایی برای این مشکل وجود دارد. به عنوان مثال می‌توان از دستورهای زبان واژگان کارکردی[5] و دستورهای زبان یکسان‌سازی کارکردی[6] استفاده کرد. دومین مشکل هم این است که هیچ راهی برای ضبط عمومیت‌ها[7] وجود ندارد. راه حل کلی برای این دو مشکل استفاده از گونه‌های زبانی[8] است. گونه‌های مورد نیاز برای دستور زبان‌های یکسان‌سازی دارای شاخصه‌های زیر است:

1) هر خصیصه با یک گونه برچسب می‌خورد؛
2) هر گونه دارای وضعیت‌های مناسبی برای خصیصه‌هایش است؛
3) گونه‌ها به صورت سلسله مراتب گونه‌ها هستند که گونه‌های موجود در سلسله مراتب پایین‌تر از ویژگی‌های مراتب بالاتر از خود ارث‌بری می‌کنند؛ و

---

[1] Directed Acyclic Graph (DAG)
[2] Content Field
[3] Pointer Field
[4] Occur Check
[5] Lexical Functional Grammar (LGF)
[6] Functional Unification Grammar (FUG)
[7] Generalization
[8] Types

4) عملیات یکسان‌سازی برای هماهنگ‌کردن گونه‌های موجود در ساختارهای خصیصه به علاوه‌ی یکسان‌سازی ویژگی‌ها و مقادیر، تغییر یافته است.

در کل دو نوع گونه وجود دارد. نوع اوّل گونه‌های ساده و نوع دوم گونه‌های پیچیده است. گونه‌های پیچیده شامل مجموعه‌ی خصیصه‌هایی که برای هر گونه‌ی مناسب، محدودیت‌ها بر روی مقادیر خصیصه‌ها و محدودیت‌های برابری بین مقادیر هستند.

7-2. تجزیه‌ی احتمالی و واژگانی

با استفاده از الگوریتم اِرلی می‌توان ابهام در تجزیه‌ی نحوی را کشف کرد ولی با این الگوریتم نمی‌توان مشکل ابهام را حل نمود. به همین دلیل دستور زبان‌هایی با نام دستور زبان‌های احتمالی به وجود آمده‌اند که به هر قانون در دستور زبان یک مقدار احتمالی تعلق می‌گیرد. در صورتی که مجموع مقادیر احتمالی قوانین برابر یک شود، این گونه از دستور زبان را پایدار گویند. ساده‌ترین روش برای یادگیری مقادیر احتمالی استفاده از پیکره‌های متنی تجزیه‌شده است. به این پیکره‌های متنی بانک درخت[1] گویند. یکی از مشکلاتی که در این روش وجود دارد این است که در مورد بسیاری از زبان‌ها از جمله زبان‌های انگلیسی و اسپانیولی ثابت شده است که احتمال حضور قوانین دستوری بسیار وابسته به مکان آن قانون در درخت تجزیه است. برای رفع این مشکل می‌توان از داده‌های آماری وابستگی‌های واژگانی استفاده کرد. روش معیاری برای ارزیابی تجزیه‌های نحوی وجود دارد که به این روش‌های سنجش PARSEVAL گویند.

7-3. پردازش نحوی انسانی

یکی از روش‌هایی که اخیراً مورد توجه قرار گرفته است؛ بررسی تجزیه‌ی نحوی انسانی یا پردازش جملات است. یکی از مهم‌ترین مواردی که در پردازش نحوی در ذهن انسان وجود دارد، رفع ابهام است. دو دیدگاه در مورد نحوه‌ی تجزیه‌ی نحوی انسان‌ها وجود دارد. دیدگاه اوّل دیدگاه پیمانه‌گرا است. طبق

[1] Tree Bank

این دیدگاه در ذهن انسان پیمانه‌های دانش نحوی وجود دارد. دیدگاه دیگر دیدگاه تعامل‌گرا است. طبق این دیدگاه تجزیه‌ی نحوی انسان‌ها یک روند متعامل است و هر انسانی دارای منابع دانش متفاوتی است که باعث می‌شود در برخی از موارد تفاسیر متفاوتی وجود خواهد داشت. برای الگوبرداری از تجزیه‌ی نحوی انسانی از شبکه‌ی باور بیزی[1] استفاده می‌شود. طبق این الگو در آن واحد انسان‌ها نمی‌توانند بیش از چند تفسیر محدود را در ذهن‌شان نگهداری کنند.

7-4. پیچیدگی زبانی

دو گونه پیچیدگی زبانی وجود دارد. یکی از پیچیدگی‌ها، پیچیدگی‌های موجود در خود زبان است ولی پیچیدگی دیگری نیز وجود دارد که این پیچیدگی منحصر به جملات خاصی است که برای بسیاری از انسان‌ها هم پیچیده است. مثالی از این پیچیدگی‌ها، پیچیدگی موجود در جملات ادبی است.

[1] Bayesian Belief Network (BFN)

8. خطایابی نحوی

تحلیل نحوی از حوزه‌هایی است که در همه‌ی زبان‌های امروز دنیا مورد توجه قرار گرفته است. توجه صِرف و تنها به واژگان بدون در نظر گرفتن جایگاه دستوری آن‌ها به هیچ وجه برای تحلیل و خطایابی یک متن کافی نیست. به عنوان مثال جمله‌ی "امروز دارد باران می‌بارم" از لحاظ واژگانی کاملاً درست است، ولی از لحاظ نحوی نادرست است. در واقع در جمله‌ی اخیر واژه‌ی "می‌بارم" یک واژه‌ی سردرگُم[1] است. واژه‌ی سردرگم در واقع کلماتی هستند که از لحاظ املایی درست ولی از لحاظ نحوی در جای نادرستی قرار گرفته‌اند[27]. اگر جمله‌ی "سیب مرا خورد" حتی از لحاظ نحوی درست است ولی از لحاظ معنایی نادرست است. با ساز و کارهای موجود در بسیاری از خطایاب‌های نحوی می‌توان به صورت محدود به خطایابی معنایی نیز پرداخت. جمله‌ی "رئیس جمهور امریکا یک کودک شش‌ساله است" از لحاظ معنایی نیز درست است ولی در حوزه‌ی عمل چنین چیزی با توجه به قوانین و محدودیت سنی ریاست جمهوری نادرست می‌شود. با وجود این‌که برای خطایابی کامل باید وارد حوزه‌ی معنا و کاربرد شد؛ نیاز است که با استفاده از روش‌های مرسوم از لحاظ نحوی جملات را مورد بررسی قرار داد. این روش‌ها وابسته به دستور زبان خاصی و زبان خاصی نیستند[28].

با اوج‌گیری استفاده‌ی کاربران فارسی‌زبان از زبان فارسی و کم‌اطلاعی بسیاری از آن‌ها از برخی از قواعد نحوی، این نیاز ضروری به نظر می‌رسد که یک خطایاب نحوی برای زبان فارسی ساخته شود. کما این‌که از ساخته‌شدن اوّلین خطایاب‌های نحوی برای زبان‌های انگلیسی، فرانسوی و آلمانی نزدیک به 30 سال می‌گذرد. حتی در صورتی که نیازی به خطایابی نحوی برای زبان فارسی وجود نداشته باشد، نیاز روزافزون به ابزارهای ترجمه و خلاصه‌ساز بسیار مشهود است. کمبود ترجمه‌ی به‌روز از کتب مرجع علمی در بازار شاهدی بر این مدّعاست. اوّلین گام برای ترجمه خودکار این است که متون به طور کامل تجزیه‌ی نحوی بشوند و در صورت وجود خطا از لحاظ نحوی اصلاح در آن‌ها صورت بگیرد. در ضمن از برنامه‌های خطایاب نحوی می‌توان به عنوان مددکار برای یادگیری زبان در افرادی که از زبان فارسی به عنوان یک زبان خارجه استفاده می‌کنند، استفاده کرد. با توجه به این مطلب این سوال به وجود می‌آید که چه کارکردهایی باید به برنامه‌های خطایاب نحوی به عنوان یاددهنده‌ی زبان اضافه کرد[29]. البته به

[1] Confused Word

جز خطایاب‌های نحوی، خطایاب‌هایی موسوم به خطایاب‌های سبکی[1] نیز وجود دارند. خطایابی نحوی مربوط به جملاتی است که از لحاظ نحوی نادرست هستند. در حالی که خطایابی سبکی مربوط به جملاتی است که از لحاظ نحوی درست هستند ولی همان جملات را می‌توان به صورتی مناسب‌تر ارائه کرد. در نتیجه خطایاب‌های سبکی دارای آزادی عمل بیشتری هستند و بیشتر با عمل اشتقاق سر و کار دارند[30].

8-1. نیازهای موجود برای خطایابی نحوی

برای این‌که بتوان یک سامانه‌ی خطایاب نحوی مناسب داشت، نخست نیاز داریم که مشکلات رمزگذاری خط فارسی در رایانه‌ها را حل کنیم. وجود چند نویسه‌ی مختلف برای یک حرف در زبان فارسی و هم‌چنین مشترک‌بودن برخی از نویسه‌ها بین زبان فارسی و عربی، باعث بروز مشکلاتی در زبان فارسی می‌شود که باید به عنوان کارهای پیش‌پردازشی در زبان فارسی آن را حل کرد. پس از رفع این مشکل نیاز است که مطمئن شویم تمام واژگان موجود در متن درست هستند. یعنی باید یک مرحله خطایابی واژگانی روی متن موجود صورت بگیرد تا بدین‌وسیله بتوان از واژگان نادرست مطلع شد و با استفاده از راهنمایی‌های خطایاب املایی گزینه‌های تصحیح مناسب را جایگزین واژه‌ی نادرست کرد. خطایاب‌های املایی در مرحله‌ی اوّل بدون در نظر گرفتن نقش دستوری واژگان پردازش روی متن انجام می‌دهند. خطاهای املایی که مربوط به کلماتی می‌شوند که در واژه‌نامه موجودند ولی نادرست هستند، در مرحله‌ی خطایابی نحوی پیدا خواهند شد[31].

اگر همه‌ی این مسائل حل شود، باید بتوان ابزارهای تجزیه و تحلیل نحوی مناسب ایجاد کرد که با آن بتوان از نادرستی احتمالی نحوی مطلع شد و با استفاده از ابزارهای نحوی مناسب گزینه‌های تصحیح مناسب را ارائه داد. در صورتی که این امر فراهم شود، راه برای ایجاد یک سامانه‌ی ترجمه‌ی هوش‌مند هم بسیار ساده می‌شود و تحلیل معنایی هم قابل انجام خواهد بود. البته بسیاری از خطایاب‌های نحوی موجود برای زبان‌های دیگر عمل تجزیه را به صورت کم‌عمق و سطحی انجام می‌دهند و برای رسیدن به سطحی که بتوان در آن بهره‌برداری ترجمه‌ای نیاز به تجدید ساختار این خطایاب‌ها است[32].

[1] Style Checkers

8-2. معیارهای کارآیی و بهبود یک خطایاب نحوی

سه مرحله را باید برای خطایابی نحوی گذراند:

1) تشخیص کران واژگان؛
2) تشخیص کران جمله؛ و
3) تشخیص کران عبارت.

پس از طی این مراحل آن‌گاه می‌توان به تجزیه‌ی نحوی زبان روی آورد. زیرا پس از طی مراحل فوق، اجزای واژگانی، جمله‌ها و عبارات در دسترس هستند و با استفاده از آن‌ها می‌توان به تجزیه‌ی نحوی پرداخت. برای تجزیه‌ی نحوی نیاز به استفاده از برچسب‌گذاری ادات سخن[۱] و در صورت لزوم استفاده از ابزارهای یکسان‌سازی[۲] برای بررسی مطابقت دستوری بین اجزای جمله است.

اصلی‌ترین چیزی که در یک خطایاب نحوی برای یک کاربر حائز اهمیت خواهد بود، پیدا کردن خطاهای نحوی و ارائه‌ی پیشنهادهای درست است[33].

منابعی که برای یک خطایاب نحوی مفید خواهند بود:

1) دانش زبان‌شناختی و زبانی؛
2) کتب و مراجع در مورد انواع خطاهای رایج در هر زبان از لحاظ نحوی؛
3) تحقیق از مشتری‌ها و کاربران؛
4) بررسی نرم‌افزارها و سامانه‌های خطایاب نحوی موجود در بازار؛ و
5) پیکره‌های متنی آماده.

معمولاً دو ملاک عمده و مرسوم برای آزمون خطایاب‌های نحوی وجود دارد. این دو ملاک دقت[۳] و فراخوانی[۴] نام دارند. دقت، برابر است با تعداد خطاهایی که اعلام شده‌اند و واقعاً خطا هستند؛ تقسیم بر

---

[۱] Part Of Speech (POS) Tagging

[۲] Unification

[۳] Precision

[۴] Recall

تعداد خطاهایی که سامانه پیدا کرده است. فراخوانی هم برابر است با تعداد خطاهایی که اعلام شده‌اند و واقعاً خطا هستند؛ تقسیم بر تعداد خطاهای واقعی در متن. از نظر کاربران بالا بودن فراخوانی از اهمیت بیشتری برخوردار است[33].

عملیات‌هایی که ممکن است در تحلیل نحوی صورت بگیرد[34]:

1) یافتن[۱]: پیدا شدن قسمت‌هایی که ممکن است، نادرست باشند؛
2) شناسایی[۲]: شناسایی قوانینی که احتمالاً نادیده گرفته شده‌اند یا اشتباه استفاده شده‌اند؛
3) تشخیص[۳]: تشخیص منابع احتمالی خطا؛ و
4) تصحیح[۴]: فرایند تصحیح شامل مراحل ذیل است:
   1. یافتن یا ساختن ساختارهای جایگزین؛
   2. رده‌بندی‌کردن جایگزین‌ها؛ و
   3. جایگزین‌کردن پراحتمال‌ترین ساختار جایگزین با ساختار نادرست.

یکی از روش‌های مرسوم این است که به صورت دستی یک پیکره‌ی متنی حاوی خطاهای نحوی را خطایابی کرده و با دسته‌بندی انواع خطاهای نحوی، بسامد هر الگو از خطا را بررسی نموده و بدین‌وسیله بر بهینگی خطایاب نحوی بیافزاییم[31]. برای واژه‌شناسی یا ریخت‌شناسی واژگان برای زبان‌هایی مانند زبان فارسی که واژه‌های ترکیبی در آن زیاد هستند، به دو مورد نیاز اساسی است[35]:

1) داشتن یک پایگاه دانش مناسب از واژگان؛ و
2) دانش مجموعه‌ی کاملی از قوانین اشتقاق واژگانی.

برای این‌که با خطاها بسیار بهینه‌تر برخورد کرد، می‌توان آمار جامعی از بسامد الگوهای انواع خطا را در زبان تهیه کرد[32, 35].

---

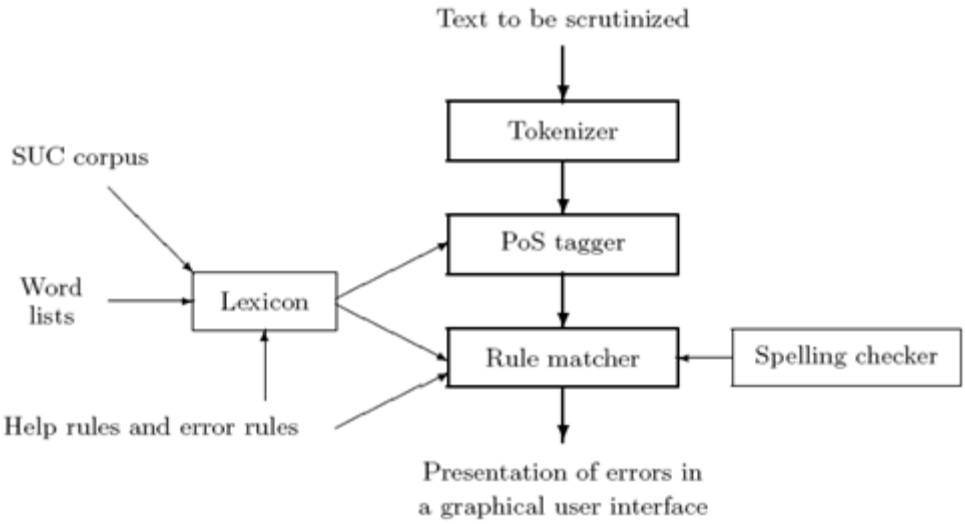
[۱] Detection
[۲] Recognition
[۳] Diagnosis
[۴] Correction

9. مروری بر خطایاب‌های نحوی موجود در زبان‌های دیگر

برخلاف زبان فارسی، در زبان‌های دیگر کارهای بسیاری بر روی تجزیه و تحلیل نحوی و معنایی صورت گرفته است. از اوایل دهه‌ی 1980 میلادی، در زبان‌های انگلیسی، فرانسوی و آلمانی سامانه‌های نرم‌افزاری خطایاب املایی، نحوی و معنایی توسعه پیدا کرده‌اند و به صورت روزافزونی در حال پیشرفت هستند. در این بخش به صورت مختصر به بررسی روش‌ها، الگوریتم‌ها و ابتکارات استفاده شده در زبان‌های دیگر می‌پردازیم و در برخی موارد کاربردپذیری آن‌ها را برای زبان فارسی مورد بررسی قرار می‌دهیم. زیرا همان‌طوری که گفته شد، بسیاری از روش‌های تحلیل و خطایابی نحوی مستقل از زبان هستند و می‌توان این روش‌ها را روی هر زبانی پیاده کرد.

ناتسون و همکارانش[29] در مقاله‌شان به بررسی خطایاب نحوی زبان سوئدی با نام گرانسکا[1] پرداخته‌اند. این برنامه توانسته است در مجموع 35% از خطاهای ویرایشی و نحوی را تشخیص دهد و 19% از خطاها (54% از خطاهای تشخیص‌داده شده) را به درستی تصحیح نماید. کارلبرگر و همکارانش[31] نیز به روند توسعه‌ی این سامانه اشاره‌ای داشته‌اند. طبق گفته‌ی آن‌ها نیز در خطایاب نحوی نیاز نیست که حتماً درخت تجزیه ساخته شود تا اگر درخت نادرست ایجاد شد؛ خطای نحوی پیدا شود. بلکه با تحلیل نحوی کم‌عمق هم می‌توان پی به خطاها برد[29]. در این خطایاب تعدادی قانون وجود دارد. برای هر قانون دستوری دو قسمت وجود دارد که به صورت بصری با یک پیکان از هم جدا شده‌اند. در صورتی که تجزیه‌ای با قسمت اوّل قانون مطابقت داشت؛ عملیاتی که در قسمت دوم قانون وجود دارد انجام خواهد شد. به دلیل حجم بالای محاسباتی و ابهام موجود در زبان‌های طبیعی استفاده از الگوریتم تجزیه‌ی معمول مانند LALR(1) یا LL(1) [36] مقرون به صرفه نیست.

[1] Granska



شکل 2- نمایی از معماری سامانه‌ی گرانسکا

اولیوا [37] در مقاله‌اش به پیاده‌سازی خطایاب نحوی بر روی دو زبان بلغاری و چکی پرداخته است. نقطه‌ی مشترک این دو زبان با زبان فارسی، بی‌ترتیب‌بودن این زبان‌ها است. لذا اغلب خطاهای نحوی را در زبان‌های بی‌ترتیب، می‌توان با استفاده از پیدا کردن عدم تطابق نقشی واژگان درون جمله دریافت. طبق ادعایی که در این کار شده بود، با آتاماتون می‌توان به این خطاها با پیچیدگی زمانی پایین‌تری نسبت به تجزیه‌گرهای نحوی رسید. لذا می‌توان نخست با آتاماتون این گونه از خطاها را بررسی کرد و در صورت عدم پیدا شدن خطا، به تجزیه‌گرها متوسل شد.

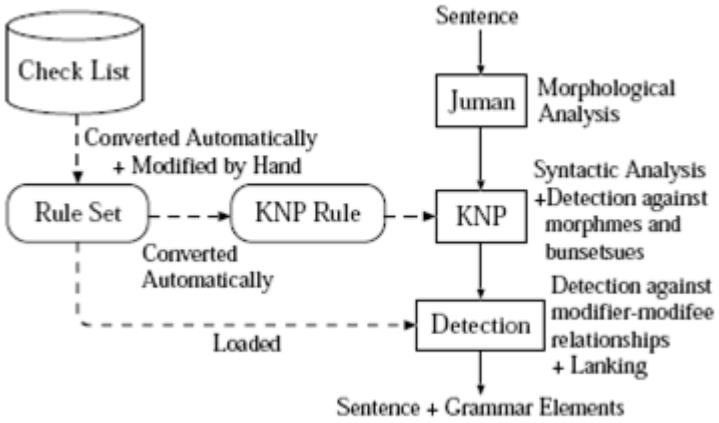
کوبون و پلاتک[38] در مقاله‌شان روی زبان‌های چکی و بلغاری کار کرده‌اند. روشی را که در این مقاله گفته شده است این است که می‌توان برای خطایابی از روش حذفی یا کاهشی استفاده کرد. در این روش اجزای جمله آن‌قدر حذف می‌شوند تا به یک تناقض آشکار در مطابقت بین اجزای جمله برسیم. در این صورت جمله خطای دستوری خواهد داشت و باید خطاگیری شود. در صورت تمام اجزای جمله حذف شوند، می‌توان نتیجه گرفت که جمله بی‌خطا بوده است. نحوه‌ی پیاده‌سازی این روش استفاده از ماشین حالت و آتوماتون است.

سوچیا و ساتو[32] بر مبنای روشی که کوروهاشی و ناگااو[39] پیشنهاد کرده بودند، به پیاده‌سازی خطایابی برای زبان ژاپنی پرداختند. در این کار از دو عنصر استفاده شده است:

1) فهرستی از دستورهای نحوی موجود در زبان؛ و

2) یک عنصر تشخیص‌دهنده‌ی عناصر دستوری برای زبان.

ساختار سامانه‌ی به کار رفته در شکل زیر نشان داده شده است.



شکل 3- معماری سامانه‌ی پیاده‌سازی شده در زبان ژاپنی

پاگیو[40] به پیاده‌سازی خطایاب نحوی اسکاری[1] پرداخته است. این برنامه برای زبان دانمارکی پیاده‌سازی شده است و خطایابی نحوی و املایی را تواماً انجام می‌دهد. طبق ادعای این مقاله، با استفاده از تجزیه‌گرهای با عمق پایین می‌توان خطایابی نحوی را انجام داد و با استفاده از بررسی پیکره‌های نحوی که در آن‌ها خطای نحوی وجود دارد؛ می‌توان به بسامد انواع خطاهای نحوی پی برد. در این سامانه نخست خطایابی املایی-واژگانی صورت می‌گیرد و پس از از بین بردن این خطاها به خطایابی نحوی می‌پردازد. آمار خطاهای به دست آمده در این مقاله در زبان دانمارکی در جدول 1 آمده است:

جدول 1 – اطلاعات به‌دست آمده از آزمون خطایاب اسکاری

نوع خطا	تعداد	%
خطای مستقل از متن	386	38

[1] SCARRIE

خطای وابسته به متن	308	30
خطا در حروف و علائم	212	21
خطای سبکی	89	9
خطای گرافیکی	24	2
مجموع	1019	100

طبق اطلاعات بالا خطاهای نحوی 30 درصد از کل خطاها را در یک متن تشکیل می‌دهند. باقی خطاها که 70% از خطاها را تشکیل می‌دهند در زبان دانمارکی به شرح ذیل می‌باشند[41]:

1) تعداد بسیار بالای اشکال مختلف فعلی محدود؛
2) خطا در عبارت‌های اسمی:
   1. خطا در تطابق[۱]؛
   2. اشتباه در گذاشتن حرف تعریف[۲]؛
   3. خطا در ترکیبات اضافی[۳]؛ و
   4. خطا در مورد ضمایر
3) جدا شدن[۴] و پیوسته‌نویسی[۵].

دستور زبان این برنامه یک دستور زبان الحاقی مستقل از متن[۱] با قوانین بازنویسی‌شده است که در آن نمادها با مشخصه‌ها[۲] در ارتباط هستند. وزن خطاها و پیام‌های اخطار خطا را می‌توان یا به قوانین و یا به

---

[۱] agreement errors
[۲] Determination
[۳] genitive
[۴] Split-up
[۵] Run-on

مشخصه‌های یک‌تایی اضافه کرد. قوانین با استفاده از یکسان‌سازی [۳] قابل استفاده هستند. امّا در مواردی که یک یا بیش از یک مشخصه غیرقابل یکسان‌سازی باشند، مشخصه‌های مورد تخلف باطل می‌شود.در این برنامه از دستور زبان‌های تکه‌ای [٤] و گذاشتن پیام خطا بر روی هر تکه استفاده شده است، دلیل این کار رسیدن به حداکثر بهینگی است. رامبل[42] بر اساس همین سامانه، به رده‌بندی انواع خطاهای نحوی و املایی برای زبان سوئدی پرداخته است.

کوبوفی و پلاتک[43] برنامه‌ای را برای خطایابی نحوی ارائه داده‌اند. این برنامه برای پروژه‌ای روی فناوری لات‌اسلاو [۵] در جمهوری چک انجام شده است. لات‌اسلاو یک فناوری زبانی برای زبان‌های اسلاویایی است. الگوی ارائه‌شده در این مقاله بر اساس نحو وابستگی [٦][44] است. این نحو برای زبان‌هایی مانند زبان‌های اسلاویک (مانند زبان چکی) که دارای بی‌ترتیبی [۷] هستند، بسیار مناسب است. با توجه به توضیحی که در این مقاله داده شده است، می‌توان نتیجه گرفت که نحو وابستگی برای زبان فارسی، که مانند زبان‌های اسلاویایی دارای بی‌ترتیبی است، بسیار مفید خواهد بود. در این برنامه از آتاماتای فهرست غیرقطعی [۸] برای پیدا کردن حذف [۹] استفاده شده است. در این برنامه از الگوی تعمیم‌یافته‌ی آتاماتای فهرست غیرقطعی استفاده شده است و این الگو ECA نام دارد. در ECA، یک آتوماتا به صورت دو صورته و دو حافظه‌ای وجود دارد که یکی برای ورودی و دیگری برای خروجی است. روی ورودی اعمال درج، حذف، جابه‌جایی و راه‌اندازی مجدد [۱۰] انجام می‌شود. اگر از لحاظ دستوری اجزای آتاماتای خروجی صحیح بود؛ یک به یک اجزای صحیح حذف می‌شوند تا آتاماتا تنها دارای عنصر نگه‌بان [۱۱] شود و در این صورت عبارت از لحاظ دستوری صحیح خواهد بود.

[۱] Augmented Context Free Grammar (ACFG)

[۲] Feature

[۳] Unification

[٤] fragment

[۵] LATESLAV

[٦] dependency syntax

[۷] free order

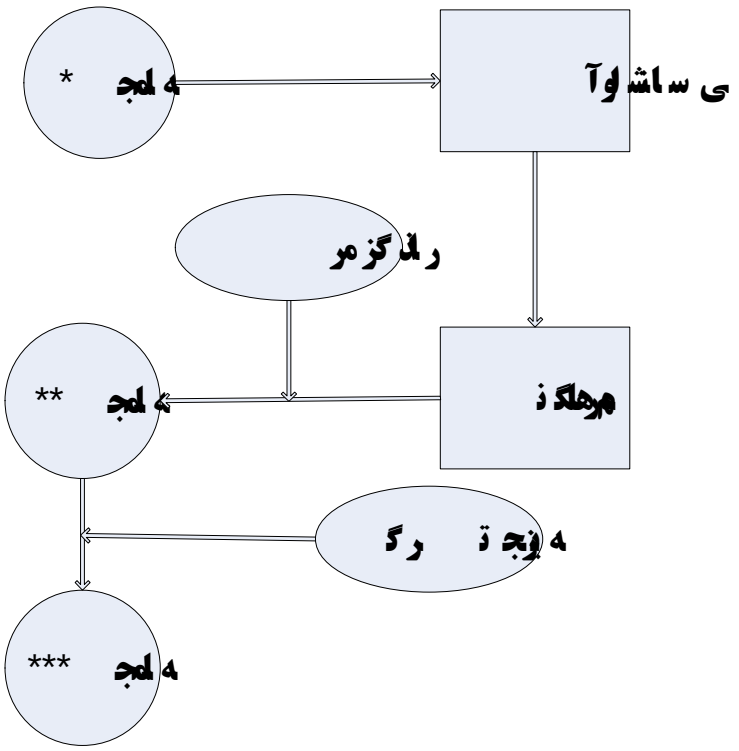
[۸] Non-deterministic List Automata (NLA)

[۹] deletion

[۱۰] restart

[۱۱] sentinel

چانود و همکارانش[45] یک سامانه‌ی املای خودکار[1] را به نام تانگورا[2] ارائه دادند. این سامانه در PLNLP[3] نوشته شده است[46] و دارای واژه‌نامه‌ی ریخت‌شناسی با بیش از 50هزار ریشه، فرهنگ نحوی، مجموعه‌ای از قواعد PLNLP با بیش از 300 قانون، مولدهای دستوری برای تفسیر و خطایابی نحوی و مولد ساختی که ساخت صحیح را ارائه می‌دهد، است. شکل 4 ساختار کلی این سامانه را نشان می‌دهد.



شکل 4- ساختار سامانه‌ی تانگورا

پارک و همکارانش[47] به کاربردهای مفید خطایاب‌های نحوی برای یادگیری زبان انگلیسی به عنوان زبان دوم[4] برای دانش‌پذیران این زبان به صورت مختصر پرداخته‌اند. هلفریچ و میوزیک[33] به بررسی یک خطایاب نحوی که تواماً برای سه زبان آلمانی، فرانسوی و اسپانیایی خطایابی می‌کند، پرداخته‌اند. در

[1] Automatic Dictation System (ADS)

[2] Tangora

[3] Programming Language for NLP

[4] English as a Second Language (ESL)

ضمن، آن‌ها در مقاله‌شان به نیازهای اصلی برای یک خطایاب چندزبانه اشاره کرده‌اند. این نیازها عبارتند از[33]:

1) دانش زبان‌شناختی و زبانی؛
2) کتب و مراجع در مورد انواع خطاهای رایج در هر زبان از لحاظ نحوی؛
3) تحقیق از مشتری‌ها و کاربران؛
4) بررسی نرم‌افزارها و سامانه‌های خطایاب نحوی موجود در بازار؛ و
5) پیکره‌های متنی آماده.

طبق ادعایی که این دو نفر داشته‌اند، بالا بودن نتایج فراخوانی بسیار با اهمیت‌تر از بالا بودن صحت خطایاب است.

وویسیک و همکارانش[48] به روش استفاده از تجزیه با دستورهای براکتی برای ارزیابی سامانه‌های خطایاب نحوی اشاره کرده‌اند. سه معیار در این برنامه در نظر گرفته شده است. دو معیار فراخوانی و صحّت هستند و معیار سوم امتیاز پرانتز[1] است. امتیاز پرانتز در واقع تعداد تجزیه‌هایی است برای یک عبارت یا جمله که با تجزیه‌هایی که به صورت دستی در پایگاه دانش ذخیره شده است، مطابقت دارند.

بوستامانته و لئون[49] در مقاله‌شان به یک سامانه‌ی خطایابی نحوی به نام GramCheck برای زبان‌های یونانی و و اسپانیایی پرداخته‌اند. در این سامانه برای مطابقت از نظر مفرد-جمع و جنس و از این جور موارد، از ارزش‌ها و مقادیر دودویی متناظر استفاده شده است.

جدول 2- آمار خطاها در سامانه‌ی GramCheck

نوع خطا	%
خطای غیرساختاری	18.5

[1] Parenthesis score

خطای ساختاری	9.7
علائم	32.2
حذف - اضافه	4.8
خطا در سطح واژگانی	6.3
خطا در سطح نویسه‌ها	8.0
ضعف در سبک	3.5
خطای ساختاری واژگانی	12.0
دیگر خطاها	5.0

بر اساس سامانه‌ی GramCheck، شعلان[50] با همین سامانه یک خطایاب نحوی را برای زبان عربی توسعه داده است.

کینوشیتا و همکارانش[51] به بررسی روند توسعه‌ی سامانه‌ی خطایاب املایی CoGrOO[1] برای محیط OpenOffice در سیستم عامل‌های لینوکسی برای زبان پرتقالی-برزیلی پرداخته‌اند. OpenNLP یکی از ابزارهای متن‌باز برای پردازش زبان انگلیسی است که در این پروژه با اندکی تغییر قابلیت استفاده برای زبان پرتقالی اضافه شده است. سامانه‌ی CoGrOO از پیمانه‌های ذیل تشکیل می‌شود:

1) کران‌یاب جملات[2]: این رویه کل متن را می‌گیرد و جملات جدا شده را به خروجی می‌دهد؛
2) واحدیاب[3]: جملات را از رویه‌ی تعیین‌کننده‌ی کران‌یاب می‌گیرد و آن را به واژه‌ها و علائم تقسیم می‌کند و به خروجی می‌دهد؛
3) نام‌یاب[1]: از واحدیاب، واحدها را می‌گیرد و اسم‌های خاص را تشخیص می‌دهد؛

[1] *(Grammar Checker for openoffice) CoGrOO – Corretor Gramatical para o OpenOffice*

[2] Sentence Boundary Detector

[3] Tokenizer

4) برچسب‌گذار ادات سخن: جمله‌ای را دریافت می‌کند و برچسب‌های ریخت‌شناسی بالقوه را به واژگان نسبت می‌دهد؛
5) قطعه‌یاب[۲]: خروجی برچسب‌گذار را می‌گیرد و عبارات اسمی و فعلی کوتاه را پیدا می‌کند؛
6) فاعل-فعل‌یاب[۳]: خروجی قطعه‌یاب را می‌گیرد و از بین عبارت‌های کوتاه فاعل و فعل جمله را پیدا می‌کند؛ و
7) خطایاب دستوری[٤]: پس از این‌که تمام مراحل تحلیل جملات برای هر جمله به پایان رسید، به دنبال خطاهای نحوی می‌گردد.

کران‌یاب جمله بر اساس علائم سجاوندی موجود، کران جملات را می‌یابد. مراحلی که برای کران‌یابی در نسخه‌ی یک این سامانه بود، به شرح ذیل است:

1) علائم و نشانه‌ها را در متن می‌یابد؛
2) به دنبال تطبیق سازه‌های دستوری مانند نشانی اینترنتی می‌رود که با وجود داشتن علائم و نقطه‌گذاری سازنده‌ی جمله نیستند؛
3) در قبال علائم جداکننده تشخیص می‌دهد که آیا نشان‌دهنده‌ی علائم اختصاری یا جداکننده‌ی جملات است؛ و
4) فقط در حالتی که یک فعل اسنادی در وجود داشته باشد؛ براکت‌ها، خطّ تیره‌ها و علائم نقل قول جداکننده‌ی جملات محسوب می‌شوند.

واحدیاب موجود در این سامانه از ماشین حالت[٥] استفاده می‌نماید. این رویه در دو گام عملیاتش را انجام می‌دهد:

1) بر اساس فاصله‌ها جمله را تقطیع می‌نماید؛

[۱] Name Finder
[۲] Chunker
[۳] Subject-Verb Finder
[٤] Grammar Error Detector
[٥] State Machine (SM)

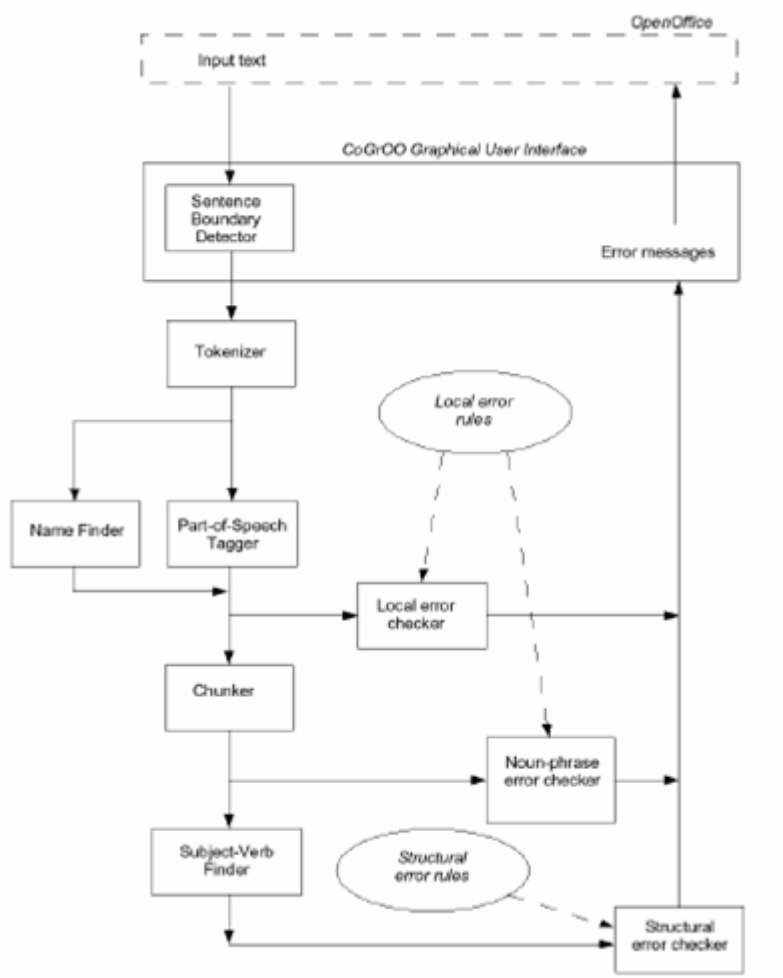
2) برای هر کدام از واحدها، مشخصه‌هایش را با توجه به پایگاه دانش و اطلاعات موجود، بررسی می‌نماید. برخی از این مشخصه‌ها به شرح ذیل هستند:

1. آیا در انتهای هر رشته یک جدا کننده‌ی بالقوه وجود دارد؟
2. آیا در انتهای هر رشته علائم جداکننده یا علائم نشان‌دهنده‌ی واژه‌های اختصاری وجود دارد؟
3. آیا رشته‌ی موجود سازه‌ای شبیه به سازه‌های معروف مانند نشانی پست الکترونیکی دارد؟
4. آیا نویسه‌ی قبل از علائم حروف بزرگ لاتین است؟

در این سامانه برچسب‌گذار مراحل ذیل را انجام می‌دهد:

1) همه‌ی برچسب‌های ممکن را به واژه‌های جمله نسبت می‌دهد؛ و
2) با توجه به بافت[1] پراحتمال‌ترین برچسب را به هر واژه نسبت می‌دهد.

[1] Context



شکل 5- معماری سامانه‌ی CoGrOO

لی و سنف[52] در مقاله‌شان بیشتر تمرکز را روی ساختن تصحیح‌های نحوی گذاشته‌اند. . طبق گفته‌ی آن‌ها، برای تصحیح نحوی باید تجزیه‌ی مناسب همراه واژه‌های مناسب را حدس زد. با استفاده از الگوهای زایشی زبان طبیعی[1]، در یک پیکره‌ی متن بسیار بزرگ در یک زبان می‌توان واژه‌ی مناسب را در یک بافت زبانی پیش‌بینی کرد. این رهیافت را می‌توان حتی برای حروف اضافه و ربط به کار برد [53, 54]. چودورو و همکارانش[54] نیز تنها به بررسی خطایابی نحوی بر روی نحوه‌ی استفاده از حروف اضافه و ربط پرداخته‌اند. در همین راستا نیز اولوفسون و ناتسون[55] در سامانه‌ی خطایابی گرانسکا که برای زبان سوئدی است، در زمینه‌ی خاص حروف اضافه و ربط به پژوهش پرداختند. طبق ادعای آن‌ها در

[1] Natural Language Generation (NLG)

خطایاب نحوی نیاز نیست که حتماً درخت تجزیه ساخته شود تا اگر درخت نادرست ایجاد شد؛ خطای نحوی پیدا شود. بلکه با تحلیل نحوی کم‌عمق هم می‌توان پی به خطاها برد[55].

هین[56] در مقاله‌اش به خطایاب نحوی اسکارچک[۱] پرداخته است. این سامانه در یک چارچوب نمودارمحور[۲] نوشته شده است. اسکارچک دارای دو پیمانه‌ی اصلی است. یک پویش‌گر نمودار[۳] و تجزیه‌گر نمودار[٤] دارد. تجزیه‌گر تا آن‌جایی که می‌تواند تجزیه‌های مختلف دستوری را می‌سازد و پویش‌گر در نمودار می‌گردد تا خطاها را بیابد. در این برنامه با رویکرد پایین به بالا عمل تجزیه‌ی نحوی انجام شده است. روشی که در این برنامه برای وزن‌دهی و امتیازدهی به انواع خطاها بوده؛ استفاده از بسامدهای به دست آمده از انواع خطاهای نحوی است. مین و ویلسون[57] با استفاده از دستور زبان مستقل از متن الحاقی، از روش تجزیه‌ی نمودارمحور استفاده کرده‌اند. سامانه‌ای که آن‌ها درست کرده‌اند چاپتر[۵] نام دارد. این سامانه یک راهبرد خطایابی دومرحله‌ای دارد و برای مرحله‌ی نحوی یک الگوریتم تجزیه‌ی نمودارمحورِ بالا به پایین انجام می‌دهد. این خطایاب انواع خطایابی را انجام می‌دهد. لذا در مرحله‌ی معنایی از توصیفات بولی استفاده می‌نماید. بازیابی خطاهای نحوی طی فرایندی چهار مرحله‌ای انجام می‌شود:

1) پیش‌بینی بالا به پایین[٦]: هدفی را با استفاده از قوانین نحوی موجود در دستور زبان گسترش می‌دهد؛
2) ارضای پایین به بالا[۷]: با استفاده از هدف و کمان‌های غیرفعال در درخت‌های تجزیه جملات صحیح از قبل تجزیه‌شده، به دنبال خطا می‌گردد و یک شبکه‌ی نمودارِ نیازمندی[۸] می‌سازد؛
3) موتور بازسازی سازه‌ها (واحدهای زبانی)[۱]: خطاها را درست می‌نماید و با استفاده از جستجوی مجدد در شبکه‌ی نمودار نیازمندی، درخت‌های محلی درست را می‌سازد؛ و

---

[۱] ScarCheck
[۲] Chart-based Framework
[۳] Chart Scanner
[٤] Chart Parser
[۵] CHArt Parser for Two-stage Error Recovery (CHAPTER)
[٦] Top-Down Expectation
[۷] Bottom-Up Satisfaction
[۸] Need Chart

4) خطایابی املایی[۲] [58]

دو مرحله‌ی اوّل صرفاً برای پیدا کردن یک خطای نحوی هستند و دو مرحله‌ی بعدی علاوه بر آن، تصحیح خطا را نیز انجام می‌دهد. در هر نمودار دو عنصر داریم: هدف[۳] و کمانِ نیازمندی[٤]. هدف، یک درخت جزئی است که ممکن است دارای یک یا چند رده‌ی نحوی در خود باشد. معمولاً هدف، یک زیردرخت از یک تجزیه‌ی نحوی، که نشان‌دهنده‌ی یکی از قوانین موجود در دستور زبان مستقل از متن موجود در پایگاه دانش می‌باشد، است. کمان نیازمندی هم مانند کمان فعال است و دارای اطلاعات ذیل است:

1) چه سازه‌های زبانی‌ای پیدا شده است؛ و
2) در یک نقطه از درخت چه سازه‌ای مورد نیاز است تا آن درخت بازیابی شود.

در مرحله‌ی بازیابی نحوی، رده‌بندی تصحیح‌ها بر دو مبنا انجام می‌شود:

1) نوع خطای مستقل از دستور زبان از جنبه‌ی اهمیت خطایی؛ و
2) اهمیت خطای وابسته به دستور زبان بر مبنای وزن‌های سازه‌های اصلاح‌شده در درخت‌های محلی‌شان.

در این برنامه خطاهای معنایی به دو رده تقسیم شده‌اند:

1) به دلیل این‌که درخت تجزیه‌ای برای یک عبارت پیدا نشده است، از لحاظ معنایی نادرست است؛ و
2) عبارت از لحاظ نحوی درست ولی از لحاظ معنایی نادرست است.

ملیش[28] در مقاله‌اش به مزیت‌های روش تجزیه‌محور اشاراتی داشته است. طبق گفته‌ی او مزیت روش‌های نمودارمحور این است که در صورتی که تجزیه‌گر خطایی را یافت، نیازی نیست که کار تجزیه برای تصحیح از نو آغاز شود. یکی از روش‌های مورد استفاده در این مقاله، استفاده از الگوریتم تپه‌نوردی[۵]

[۱] Constituent Reconstruction Engine
[۲] Spelling Correction
[۳] goal
[٤] need arc
[۵] hill climbing

برای پیدا کردن بهترین تصحیح است. الگوریتم‌هایی مانند بازیابی خطای طولانی‌ترین مسیر[۱] یا فاصله‌ی کمینه[۲] نیز مورد استفاده قرار گرفته‌اند. در تجزیه‌ی بالا به پایین می‌توان این ضمانت را کرد که همه‌ی کمان‌های فعال ممکن برای یک عبارت یا جمله بررسی شده و در صورت مطابقت تولید می‌شوند. از طرفی دیگر در تجزیه‌ی پایین به بالا این ضمانت وجود دارد که همه‌ی سازه و زیرسازه‌های ممکن در یک عبارت یا جمله پیدا شوند. راهبردی که در این مقاله به عنوان بهترین راهبرد در نظر گرفته شده است، این است که نخست از تجزیه‌ی پایین به بالا استفاده شود و در صورت پیدا شدن خطا از روش بالا به پایین تعمیم‌یافته برای تولید تجزیه‌های مناسب استفاده شود. روش بالا به پایین تعمیم‌یافته از حداقل خطاهای ممکن بهره می‌برد. دراین‌جا در هر یال فعال اطلاعاتی اضافی وجود خواهد داشت.

کاتو[59] با تعمیم روشی که ملیش[28] استفاده کرده بود، روشی را ارائه داده بود که در آن روش می‌توان با واژگان ناشناخته و حتی حذف شده از جمله هم برخورد مناسب داشت. در این‌جا به جای استفاده از یک روند برای تجزیه‌ی خطاها[28]، در رویه‌ای دومرحله‌ای انجام می‌شود و بدین‌وسیله فضای جست‌وجو هرس می‌گردد. در ضمن به جای این‌که از تجزیه‌ی بالا به پایین نمودارمحور[۳] استفاده شود از بالا به پایین معمولی استفاده می‌گردد و با الگوریتم A^* بهترین پاسخ پیدا می‌شود. طبق گفته‌ی کاتو[59] یکی از مشکلاتی که در کار ملیش[28] وجود داشت این بود که کارآیی برنامه بیش از اندازه وابسته به الگوریتم ابتکاری جستجو بود. در نتیجه طبق اذعان خود ملیش[28] برنامه‌اش با جملات با بیش از یک خطا با کارایی بسیار پایینی کار می‌کرد. در این برنامه به جای این که مستقیماً خطایابی انجام شود، دو مرحله تشخیص خطا در نظر گرفته شده است و به همین دلیل به خطایابی دوسویه[٤] معروف است. ساختمان داده‌ای که ملیش[28] معرفی کرده بود یک یال تعمیم‌یافته[٥] بود که هر یال فعال حاوی اطلاعاتی درون خود بود. تجزیه‌گر بالا به پایین تعمیم‌یافته همراه با یال تعمیم‌یافته دارای 6 قانون؛ 3 تا برای خطایابی و 3 تا برای تصحیح، بود. طبق الگوریتم A^*، دو معیار h(هزینه تا حال) و g(هزینه‌ی پیش‌بینی شده تا رسیدن به مقصد) داریم.

---

[۱] longest-path error recovery

[۲] minimum distance

[۳] chart-based top-down parsing

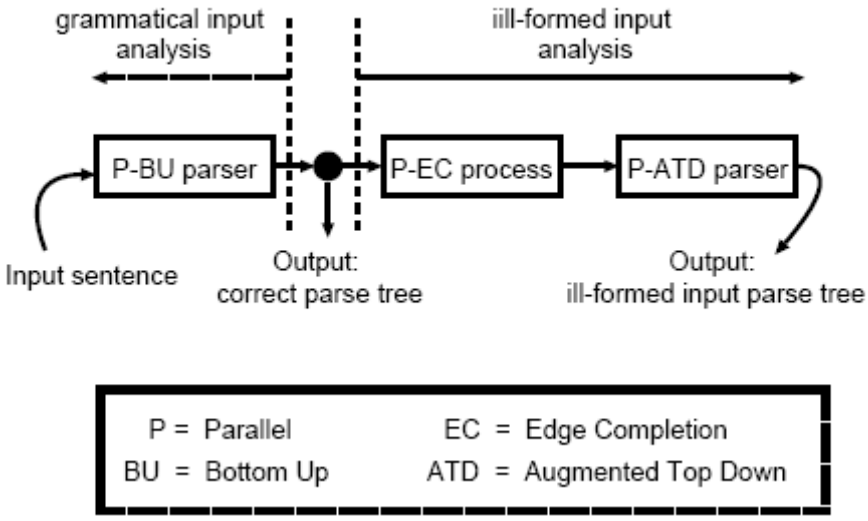
[٤] bidirectional error detection

[٥] Generalized edge

تیرامونگونک و تاناکا[60] با استفاده از زبان برنامه‌نویسی KL1 به پیاده‌سازی سامانه‌ی خطایابی مستقل از نحو پرداختند. طبق گفته‌ی آن‌ها دو رهیافت اصلی در برخورد با نحو و خطاهای نحوی در میان متخصصین وجود دارد: گروه چارچوب‌محور[۱] و گروه نحومحور[۲]. البته برخی اوقات، از روش‌های تطبیق الگو[۳] نیز استفاده شده است. تجزیه‌گر نمودار محور دو تا اطلاعات را در خود دارد. رکورد همه‌ی تجزیه‌های صحیح و رکورد همه‌ی زیررشته‌های صحیح. دو گونه عمل تولید یال[۴] و بسط یال[۵] در این روش انجام می‌شود. تولید یال، کاری است که با آن می‌توان یال غیرفعالی را با توجه به دستور زبان موجود به یک یال فعال تبدیل کرد. بسط یال، عمل ساخت یال جدید از یک یال فعال با استفاده از یک یال غیرفعال است. در این برنامه هم می‌توان یک یا تلفیقی از چهار نوع خطای مرسوم که ملیش[28] و کاتو[59] گفته‌اند داشته باشیم:

1) واژه‌ی اضافی: اگر در جمله یا عبارت واژه‌ای اضافی درج شده باشد که با وجود آن واژه نتوان تفسیری یا تجزیه‌ای درست از عبارت یا جمله داشت: "من با دوستان به به* بازار رفتم"
2) واژه‌ی حذف‌شده: واژه‌ای باید در جمله می‌بود ولی درج نشده است: "من * دوستان به بازار رفتم"
3) واژه‌ی ناشناس: واژه‌ای که از لحاظ املایی نادرست وارد متن شده باشد: "من با دوستپان* به بازار رفتم"
4) واژه‌ی جایگزین‌شده: واژه‌ای که از لحاظ نحوی به صورت نادرست در جمله آمده است: "من با دوستان به بازار رفتند*"

[۱] frame-based group
[۲] syntactically-oriented group
[۳] pattern matching
[۴] creating edge
[۵] extending edge



شکل 6- سه ساختار برای تجزیه [60]

کانگ و همکارانش[35] یک خطایاب نحوی برای زبان کره‌ای معرفی کردند. نام این خطایاب میرین[۱] بوده است. طبق ادعای آن‌ها برای واژه‌شناسی یا ریخت‌شناسی واژگان برای زبان‌هایی مانند کره‌ای که واژه‌های ترکیبی بسیار زیاد هستند، به دو مورد نیاز اساسی است. یکی داشتن یک پایگاه دانش مناسب از واژگان و دیگری دانش مجموعه‌ی کاملی از قوانین اشتقاق واژگانی. و همین‌طور برای این‌که با خطاها بسیار بهینه‌تر برخورد کرد، باید آمار جامعی از الگوهای خطا در زبان تهیه کنیم. بررسی‌هایی که روی پیکره‌های متنی کره‌ای انجام شده است، نشان می‌دهد که عمده‌ی خطاها شامل: نحوی، عدم تطابق نحو-واژه[۲] و عدم تطابق در عمل[۳] است. خطای فاصله‌های نادرست یکی از خطاهایی است که باعث ابهام در تجزیه‌ی نحوی می‌شود. طبق گفته‌ی آن‌ها عمده‌ی خطاهای املایی به چینش صفحه کلیدها برمی‌گردند[35]. بنابراین حالت آرمانی برای یک خطایاب نحوی این است که دارای پایگاه دانش زبان طبیعی گسترده و تجزیه‌ی کامل جملات باشد. امّا چنین حالتی با توجه به امکانات موجود و تغییرات سریع در زبان و منابع زبانی ناممکن به نظر می‌رسد. تجزیه‌ی کامل جملات دارای اطلاعات اضافه‌ای

[۱] Mirine

[۲] lexio-semantic incompatibility

[۳] Pragmatic incompatibility

خواهد بود که به درد خطایابی نحوی نخواهد خورد؛ به همین خاطر در این مقاله به بررسی تجزیه‌ی بخشی[۱] با استفاده از دستور زبان وابستگی[۲] پرداخته است.

تومیتا[61] یک تجزیه‌گر نحوی را برای زبان‌های طبیعی معرفی کرده است. این تجزیه‌گر تعمیم‌یافته‌ی تجزیه‌گر LR است که برای زبان‌های مستقل از متنی که ابهام دارند مناسب خواهد بود و با معرفی پشته با ساختار گراف[۳] بهینه‌گی را حفظ می‌نماید. این تجزیه‌گر به صورت برخط[۴] با ورود هر واژه‌ی جدید به متن به پردازش و تجزیه می‌پردازد و نیازی نیست که حتماً جمله تمام بشود. ایده‌ای که در این تجزیه‌گر پیشنهاد شده، این است که ورودی‌های غیرقطعی[۵] را هم‌زمان پذیرش کند. در این صورت فهرستی از پشته‌ها وجود خواهند داشت. در نتیجه تعدادی پردازه وجود خواهند داشت و با وجود این پردازه‌ها، برای هر پردازه یک پشته و در هر پشته همان عمل تجزیه‌ای صورت می‌گیرد که در LR انجام می‌گیرد. اگر دو پردازه تا جایی دارای پشته‌ی یکسان باشند، می‌توان از ساختار درخت پشته‌ها استفاده کرد که ریشه‌ی درخت برای دو پردازه یک پشته‌ی مشترک است و ادامه‌ی پشته‌ها بعد از ریشه متفاوت خواهند بود. طبق ادعاهایی که در مقالات گذشته شده است، احتمالاً روشی که تومیتا ارائه داده برای خطایابی نحوی دارای سربار محاسباتی و اطلاعاتی خواهد بود ولی این روش دارای صحت بسیار بالای است.

ووسه[62] روشی را برای خطایابی ساخت‌واژی و نحوی زبان آلمانی ارائه داده است. . در این برنامه از یک خطایاب املایی و یک تجزیه‌گر انتقال-کاهش برای دستور زبان مستقل از متن الحاقی[۶] استفاده شده است. الگوریتم تجزیه در این برنامه، تعمیم‌یافته‌ی الگوریتم تومیتا[61] است. سه مرحله برای کار در نظر گرفته شده است: 1)خطایابی املایی 2) تجزیه‌ی نحوی 3)خطایابی نحوی و تصحیح خطاها. در این مقاله اشاره شده است که سه گونه خطا داریم: خطای تحریری[۷]، خطای ویرایشی[۸] و خطای عدم تطابق ساخت‌واژی[۹]. عمدتاً خطاهای نحوی شامل عدم تطابق ساخت‌واژی جمله می‌شود[62]. سامانه‌ی

---

[۱] Partial

[۲] dependency grammar

[۳] Graph-Structured Stack

[۴] online

[۵] non-deterministic

[۶] shift reduce parser for augmented context-free grammar

[۷] typographical

[۸] Orthographical

[۹] morpho-syntatic

پیاده‌سازی شده در این مقاله، دارای دو سطح است: سطح واژگانی و سطح جمله‌ای. دستور زبان‌های مستقل از متن الحاقی، مبنای مناسبی برای خطایابی نحوی هستند. در این برنامه از تجزیه‌ی انتقال-کاهش مانند LALR(1) که تومیتا آن‌ها را بهبود داده، استفاده شده است. نکته‌ی مهمی وجود دارد و آن است است که خیلی کم پیش می‌آید که واژگان (گنج‌واژه‌ی) ما همه‌ی واژه‌های متن را داشته باشد. بنابراین لازم است که باید با این فرض که همه‌ی واژه‌ها درست‌اند و برخی در رسته‌ی واژه‌های ناشناخته[۱] قرار می‌گیرند، تجزیه و خطایابی نحوی صورت بگیرد.

یکی از روش‌های خطایابی مرسوم در بسیاری از زبان‌ها خطایابی در ضمنِ محدودکردن دستور زبان است. به عنوان نمونه می‌توان به کارهای آدریانز و اسکارئوس[63] و آدریانز[64] اشاره کرد. سامانه‌ی NOMAD[65] نیز یک سامانه‌ی تفسیر وخطایابی برای زبان انگلیسی محدودشده است. این برنامه صرفاً برای تفسیر پیام‌های کوتاه در بین کشتی‌رانان و ملوانان و افراد در حال ارتباط در ساحل به صورت تلگرافی است. در زبان اردو، کبیر[66] با استفاده از دستور زبان با ساختار عبارت به پیاده‌سازی یک خطایاب نحوی پرداخته است. این خطایاب، طی دو گذر[۲] خطایابی نحوی صورت می‌گیرد.

اِتول[67] به روشی برای خطایابی نحوی بدون تجزیه اشاره کرده است. او در این کار از سامانه‌ی CLAWS بهره برده است و بر روی پیکره‌ی متنی LOB که یک پیکره‌ی متنی انگلیسی است، این سامانه را پیاده‌سازی نموده است[68, 69, 70]. این سامانه صرفاً با استفاده از پیدا کردن انتقال رده‌های واژگانی نامأنوس خطاهای نحوی را می‌یابد و هیچ‌گونه تجزیه‌ی نحوی را انجام نمی‌دهد. با وجودی که چنین روشی برای تحلیل نحوی اصلاً مناسب نیست امّا اغلب خطاهای نحوی را می‌توان با این روش یافت[67].

هوانگ و پاورز[27] تحقیقات گسترده‌ای را بر روی دلایل به وجود آمدن واژه‌های سردرگم انجام دادند. شش دلیل مختلف ممکن است وجود داشته باشد که واژه‌ای نادرست است و باید با واژه‌ی مناسبی جایگزین شود:

1) خطای تحریری ("اتسان" به جای "انسان")؛
2) خطای هم‌نگاره‌ای ("سورت" به جای "صورت")؛

---

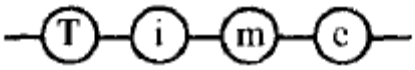
[۱] unknown words

[۲] pass

3) خطای نحوی؛
4) خطای ناهم‌خوانی اجزای جمله؛
5) خطای یادگیران زبان؛ و
6) خطای وابسته به طرز فکر[1]

جنتیال و همکارانش[71] به بررسی سامانه‌های نوشتاری با کمک رایانه[2] در زبان فرانسوی پرداخته‌اند. اصلی‌ترین مشخصه‌ی سامانه‌ی پیشنهادی استفاده از نمایش داخلی متن به شکل شبکه‌ی چندبعدی[3] است که این شبکه نقش تخته سیاه برنامه را ایفا می‌کند. ایده‌ی شبکه‌ی چندبعدی از بویت[72] گرفته شده است.

هر گره از این شبکه قسمتی از اطلاعات داخل متن را در بر دارد. هر گره یک زیردرخت دارد که یک ساختار مشخصه را در خود نگه‌داری می‌کند.



شکل 7- نمایی از گره‌های شبکه برای هر واژه

در این مقاله گفته شده که بسیار بهینه است که واحدهای واژگانی را به عنوان گره‌های شبکه در نظر بگیریم. در این شبکه ما می‌توانیم دو بعد داشته باشیم: یک بعد ترتیبی[4] برای پی‌رفتی از واژگان در جمله و بُعد دیگر، بعد ابهام (چون ممکن است از واژه‌ای درون جمله چند برداشت نقشی یا معنایی داشت). وقتی که تجزیه‌ی نحوی آغاز شد، بعد سوم نمایان می‌شود. در این مرحله گره‌های جدیدی به شبکه اضافه می‌شود که نشان‌دهنده‌ی درخت‌های وابستگی هستند.

جنتیال و همکارانش[73] در کاری دیگر در زبان فرانسوی، به بررسی دلایل عمده‌ی خطاهای نحوی در راستای ایجاد یک سامانه‌ی خطایابی کاربرپسندتر پرداختند. در آن سامانه، هر خطایی که باعث شود سامانه نتواند عبارتی را تفسیر نماید، خطای نحوی محسوب می‌شود. بنابراین دو خطا وجود دارد:

---

[1] idiosyncratic

[2] CAW (Computer Aided Writing)

[3] Multidimensional Lattice

[4] Sequential

1) به دلیل ناکافی بودن اطلاعات سامانه در مورد زبان؛ و
2) به دلیل اشتباه کاربر.

وقتی خطای عدم تطابق به وجود می‌آید حداقل دو جز از جمله وجود دارند که با تغییر یکی از آن‌ها تطابق به وجود می‌آید و این به کاربر بستگی دارد. ولی این برنامه روندی ابتکاری را برای وزن‌دهی به پیشنهادهای صحیح در نظر گرفته است. عوامل تعیین‌کننده در وزن‌دهی پیشنهادها عبارتند از:

1) بسامد خطاها در هر گروه؛
2) بهتر است تغییر طوری باشد که ترکیب آوایی جمله زیاد تغییر نکند؛
3) معمولا نویسنده کمتر می‌نویسد تا بیشتر پس خطای حذف s از آخر یک واژه بسیار بیشتر از خطای درج خواهد بود[1]؛ و
4) اولویت به سر جمله داده شود (یعنی سر جمله ترجیحاً تغییر نکند و اجزای دیگر تغییر کنند).

نوع وزن‌دهی به این چهار عامل در این برنامه به صورت پویا خواهد بود و هر عامل وزن ثابتی نخواهد گرفت. البته یک پیش‌فرض در تصحیح وجود دارد و آن پیش‌فرض این است که حداقل واحدها تغییر یابند. در ادامه این مقاله به بررسی چگونگی برخورد با خطای تطابق اجزای جمله پرداخته است و پیشنهاد به ساخت سه زیردرخت متصل به هم برای بررسی تطابق اجزای جمله داده شده است.

سامانه‌ی تجزیه‌گر نحوی فیدیچ[74] با الهام از کاری که مارکوس[75] کرده بود؛ ساخته شده است. . این تجزیه‌گر ورودی را می‌گیرد و به عنوان خروجی یک ساختار براکتی را نشان می‌دهد که این ساختار در واقع نمایشی از یک درخت تجزیه است. در دو مرحله کار تجزیه انجام می‌گیرد: نخست با توجه به پیکره‌ی موجود هر واژه‌ی درون متن یک برچسب می‌گیرد. سپس با استفاده از دو ساختمان داده یک ساختار عبارت برای درخت تجزیه درست می‌شود. این دو ساختمان داده عبارتند از: 1) یک پشته از گره‌های ناکامل و 2) حافظه‌ای میانگیر (بافر) از سازه‌های زبانی کامل‌شده.

ویشدل و رامشاو[76] به بررسی کلی در مورد تصحیح جملات با شکل نادرست پرداختند. ویشدل و سوندهیمر[77] با ادعای این‌که هر جا زبان طبیعی باشد، خطا هم هست؛ یک برنامه‌ی خطایاب نحوی ارائه دادند. رویه‌ی زیر برای خطایابی در این برنامه پیشنهاد شده است:

---

[1] user laziness

1) ورودی را پردازش کن؛
2) اگر تفسیری از ورودی وجود نداشت با استفاده از فراقانون‌ها[۱] بر اساس رده‌بندی اصلاحات جایگزین به اصلاح آن بپرداز:
   1. ایراد موجود را تشخیص بده. این که در استفاده از کدام قانون اشتباه شده و راه حل آن چیست؛
   2. ایراد را بر طرف کن؛
   3. یادداشتی مبنی بر وقوع خطا به تفسیر اضافه کن؛ و
   4. ورودی را به ورودی اصلاح‌شده تغییر بده و در صورت امکان به پردازش ادامه بده.
3) در صورت نیاز روند مرحله‌ی 2 را تکرار کن.

گریشمن و پنگ[78] در مقاله‌شان از طبقه‌بندی خطا در شکل ناقص جملات[۲] از ویشدل[77] استفاده کردند. طبق این طبقه‌بندی دو نوع خطای نقصان شکل وجود دارد: مطلق[۳] و نسبی[۴]. در این مقاله به بررسی یک سامانه‌ی پرسش و پاسخ زبان طبیعی کوچک پرداخته شده است که در آن از دستور مستقل از متن الحاقی استفاده شده است. دو مرحله در این خطایاب وجود دارد:

1) تجزیه با استفاده از دستور زبان؛ و
2) منظم‌سازی نحوی[۵] که اجزای جمله را به نقش‌های متعارف زبانی تبدیل می‌نماید.

اشمیت‌ویگر[79] در مقاله‌اش به بررسی تجزیه و خطایابی نحوی و سبکی در زبان آلمانی پرداخته است. این برنامه بر روی پروژه‌ی MULTILINT[80] بر روی زبان آلمانی انجام شده است. در این مقاله اشاره شده است که هر چه قواعد دستوری پیچیده‌تر شوند، تعداد تجزیه‌های نحوی برای یک جمله افزایش

---

[۱] meta-rule

[۲] ILL-FORMDEDNESS

[۳] absolute

[۴] relative

[۵] syntactic regularization

خواهد یافت و تجزیه‌گر باید بهترین تجزیه را در نظر بگیرد. مسأله‌ای که در نظر خطایاب‌های سبکی وجود دارد این است که چه معیاری باید برای سبک در نظر گرفت. مثلاً یک معیار می‌تواند این باشد که از جملات در حالت مجهول کم‌تر استفاده شود.

در این مقاله آمده که یک تجزیه‌گر نحوی برای خطایابی نحوی باید دارای موارد ذیل باشد:

1) باید تصحیح‌گر برای هر یک خطاها داشته باشد. دراین جا می‌توان از نمودار استفاده کرد؛
2) باید یک راهبرد اوّل‌بهترین استفاده شود که به‌ترین تجزیه‌ی ممکن به عنوان خروجی بیاید. هم‌چنین برای تصحیح بهترین حالت را داشته باشیم؛ و
3) باید امکان اتخاذ روندهای محلی مثل درج یا حذف برای تصحیح وجود داشته باشد.

جدول 3- آمار خطاهای به دست آمده در سامانه‌ی MULTILINT

نوع خطا	بسامد خطا
علائم	238
حروف بزرگ و کوچک (در زبان‌های با الفبای لاتین)	17
پیوسته و جدانویسی نادرست	46
تطابق اجزا	44
خطاهای خاص و نادر نحوی (مانند واژه‌ی تکراری)	18

جدول 4- مقایسه‌ی سامانه‌های دیگر با MUTILINT

	فراخوانی	دقت

خطایاب نحوی MULTILINT	%57	%81
خطایاب سبکی MULTILINT	%65	%92
SECC[64]	%87	%93
BSEC[81]	%89	%79

در این جا سامانه‌ی دستور زبانMETAL مورد استفاده قرار گرفته است. این سامانه وابسته به زبان نیست و می‌توان پایگاه دانش موجود در هر زبانی را به آن داد. دارای نمودار فعال برای تصحیح خطا است. دارای پایگاه دانش دستوری برای برخی از زبان‌ها است.

خطاهای معمول سبکی شامل موارد زیر است[82]:

1) جملات بسیار بلند؛
2) جملات بسیار پیچیده؛
3) اسامی با پیشوندهای بسیار زیاد؛
4) واژگان و اصطلاحات متناقض؛ و
5) روابط مبهم میان عبارات اضافی (بسیار وابسته به زبان است و برای هر زبانی نوع خاصی است).
6) پیچیدگی جملات نیز می‌تواند به دلایل ذیل باشد:

1. تعداد زیاد قوانینی که در طی تجزیه‌ی یک جمله به وجود آمده است (یعنی زیردرخت‌های یک درخت تجزیه که خود یک تجزیه‌ی مستقل محسوب می‌شوند؛ بسیار زیاد هستند)؛
2. تعداد زیاد گره‌ها در درخت تجزیه؛ و
3. تعداد زیاد گره‌ها در مورد یک ویژگی یا زیرعبارت در درخت تجزیه.

10.نتیجه‌گیری

با توجه به کارهای انجام‌شده در زبان‌های دیگر واقعاً، تولید یک خطایاب نحوی و سبکی برای زبان فارسی ضروری است. زبانی که به زعم بسیاری از کارشناسان، در صورت عدم توجه کافی خطر انحطاط را در پیش رو دارد. در ضمن، در بین کاربران زبان فارسی دقت کافی در رعایت اصول سبکی وجود ندارد و خطایاب نحوی‌ای که دارای خطایابی سبکی هم هست، می‌تواند بر رفع این مشکل به کار آید.

نکته‌ی جالب توجه در کارهای انجام‌شده استقلال روش‌ها از زبان بود. همه‌ی روش‌هایی که در بخش‌های پیش بدان‌ها اشاره شده است می‌توانند روش‌های مناسبی برای این کار باشند. روش‌هایی که مختص به زبان‌های با خاصیت بی‌ترتیبی هستند، قاعدتاً به زبان فارسی نزدیکی بیشتری دارند. در مجموع اگر می‌خواهیم که در زمینه‌ی پردازش زبان فارسی، اعم از تشخیص گفتار و ترجمه‌ی ماشینی، پیشرفت داشته باشیم، نیاز به یک ابزار قدرت‌مند خطایابی خواهیم داشت.

مراجع

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

[2] مهرنوش شمس‌فرد؛ "پردازش متون فارسی: دستاوردهای گذشته، چالش‌های پیش رو"؛ دومین کارگاه پژوهشی زبان فارسی و رایانه، ص 172-189، تهران، 1385.

[3] D. Jurafsky and M. James; "Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics"; Prentice-Hall, 2000.

[4] Roger Garside, Geoffrey Leech and Geoffrey Sampson; "*The Computational Analysis of English: A Corpus-Based* Approach", Longman, London and New York, 1987.

[5] محمود بیجن‌خان؛ "*طرح مدلسازی زبان فارسی، مرحله دوم*"؛ آزمایشگاه گروه زبان‌شناسی، دانشکده‌ی ادبیات و علوم انسانی، دانشگاه تهران، 1381.

[6] J. Carlberger, R. Domeij, V. Kann and O. Knutsson; "The Development and Performance of a Grammar Checker for Swedish: A Language Engineering Perspective"; Natural Language Engineering, Cambridge University Press, 2004.

[7] K. Kukich; "Techniques for automatically correcting words in text"; ACM Computing Surveys 24, 378– 439, 1992.

[8] A. Viterbi; "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm"; IEEE Transactions on Information Theory, Vol. 13, Issue 2, pp. 260-269, 1967.

[9] Lawrence R. Rabiner; "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition"; Proceeding of the IEEE, vol. 77, No. 2, 257-286, February 1989.

[10] E. Charniak, C. Hendrickson, N. Jacobson and M. Perkowitz; "Equation for part-of-speech tagging"; Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 784-789, 1993.

[11] J. Kupiec; "Robust part-of-speech tagging using a hidden Markov model"; Computer Speech and Language, 6(3): 225-242, 1992.

[12] Andrew David Beale; "Lexicon and grammar in probabilistic tagging of written English"; Proceedings of the fifth conference on European chapter of the Association for Computational Linguistics, pp. 155-161, 1991.

[13] *Karine Megerdoomian;* "Developing a Persian Part of Speech Tagger"; *Proceedings of First Workshop on Persian Language and Computers*. Invited talk. Tehran University, Iran. May 25-26, 2004.

[14] Karine Megerdoomian; "Finite-State Morphological Analysis of Persian"; *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*. Coling 2004, University of Geneva. August 28, 2004.

[15] M. Assi and M. Haji Abdolhossini; "Grammatical Tagging of Persian Corpus"; *International Journal of Corpus Linguistics*. 5(1), pp. 69-82, 2000.

[16] S. M. Assi; "Farsi Linguistic Database (FLDB)"; *International Journal of Lexicography*, Vol. 10, No. 3, EURALEX Newsletter p. 5, 1997.

[17] Ali Azimizadeh, Mohammad Mehdi Arab and Saeid Rahati Quchani; "Persian part of speech tagger based on Hidden Markov Model"; JADT: 9es Journées internationales d'Analyse statistique des Données Textuelles, 2008.

[18] M. BijanKhan; "The Role of the Corpus in Writing a Grammar: An Introduction to a Software"; Iranian Journal of Linguistics, Vol. 19, no. 2, 2004.

[19] F. Oroumchian, S. Tasharofi, F. Raja and M. Rahgozar; "Evaluation Of Statistical Part Of Speech Tagging Of Persian Text"; *International Symposium on Signal processing and its application*. Sharjah, United Arab Emirates, 2007.

[20] F. Oroumchian, S. Tasharofi, H. Amiri, H. Hojjat and F. Raja; "Creating a Feasible Corpus for Persian POS Tagging"; *UOWD Technical Report Series, 2006*.

[21] Azimizadeh and M. M. Arab; "The Persian Morphological parser by Using POS Tagger"; In *CAASL-2 Proceedings*. Stanford University, pp.22-29, 2007.

[22] Reza Karimpour, Amineh Ghorbani, Azadeh Pishdad, Mitra Mohtarami, Abolfazl AleAhmad, Hadi Amiri and Farhad Oroumchian;"Using Part Of Speech Tagging In Persian Information Retrieval"; CLEF 2008 Workshop, Aarhus, Denmark, 17-19 September 2008.

[23] Abolfazl AleAhmad, Hadi Amiri, Ehsan Darrudi, Masoud Rahgozar and Farhad Oroumchian;" Hamshahri: A standard Persian text collection"; Knowledge-Based

Systems, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, Vol. 22, Issue 5, pp. 382-387, July 2009.

[24] Raja, Amiri, Tasharofi, Sarmadi, Hojjat and Oroumchian; "Evaluation of Part of Speech Tagging on Persian Text"; *The Second Workshop on Computational Approaches to Arabic Script-Based Languages, LSA 2007 Linguistic Institute, Stanford University, USA, 2007.*

[25] Farhad Oroumchian, Samira Tasharofi, Hadi Amiri, Hossein Hojjat and Fahime Raja; "*Creating a Feasible Corpus for Persian POS Tagging*"; Technical Report, no. TR3/06, University of Wollongong (Dubai Campus), 2006.

[26] Jan W. Amtrup, Karine Megerdoomian, Hamid Mansouri Rad, and Rémi Zajac; "Persian-English Machine Translation: An Overview of the Shiraz Project"; NMSU, CRL, Memoranda in Computer and Cognitive Science (MCCS-00-319), 2000.

[27] Jin Hu Huang and David Powers; "Large Scale Experiments on Correction of Confused Words"; Proceeding of the 24th Australasian conference on Computer Science,ACSC 2001, 2001.

[28] Chris S. Mellish; "Some Chart-Based Techniques for Parsing Ill-Formed Input"; Proceedings of 27th ACL, 102-109, 1989.

[29] O. Knutsson, T. Cerratto Pargman, and K. Severinson-Eklundh; "Transforming grammar checking technology into a learning environment for second language writing"; In Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing, pp. 38–45, 2003.

[30] Gregor Thurmair ; "Parsing for Grammar and Style Checking for German"; Proceedings of the 13th conference on Computational linguistics - Vol 2, Finland, pp.365-370, 1990.

[31] J. Carlberger, R. Domeij, V. Kann and O. Knutsson; "The Development and Performance of a Grammar Checker for Swedish: A Language Engineering Perspective"; Natural Language Engineering, Cambridge University Press, 2004.

[32] Masatoshi Tsuchiya and Satoshi Sato; "Automatic Detection of Grammar Elements that Decrease Readability"; Proceedings of 41st Annual Meeting of the Association for Computational Linguistics, pp. 189-192, July 2003.

[33] Antje Helfrich and Bradley Music; "Design and evaluation of grammar checkers in multiple languages"; Proceedings of the 18th conference on Computational linguistics-Vol 2, Germany, pp. 1036 – 1040, 2000.

[34] Uszkoreit and Hans; “Grammar Checking. Theory, Practice, and Lessons learned in LATESLAV”; Concluding oral presentation at the final review meeting of the Lateslav Project (PECO 2824), Prague, August 1996.

[35] MI-YOUNG KANG, AESUN YOON, and HYUK-CHUL KWON; “Improving Partial Parsing Based on Error-Pattern Analysis for a Korean Grammar-Checker”; ACM Transactions on Asian Language Information Processing, Vol. 2, No. 4, pp. 301-323, December 2003.

[36] Aho, R. Sethi, and J. Ullman; “Compilers: Principles, Techniques and Tools”; Addison-Wesley, Reading, Mass., 1986.

[37] Karel Oliva; “Techniques for Accelerating a Grammar-Checker”; Proceedings of the fifth conference on Applied natural language processing; pp. 155–158; Association for Computational Linguistics, Washington, DC, 1997.

[38] V. Kubofi and M. Plitek ; “A Grammar Based Approach To A Grammar Checking Of Free Word Order Languages”; Proceedings of the 15th International Conference on computational linguistics (COLLIJING 94), 1994.

[39] Sadao Kurohashi and Makoto Nagao; “A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures”; *Computational Linguistics*, 20(4), 1994.

[40] Patrizia Paggio; “Spelling and Grammar Correction for Danish in SCARRIE”; Proceedings of the sixth conference on Applied natural language processing, ACL, Seattle, Washington, pp. 255 -261, 2000

[41] Claus Povlsen; “Three types of grammatical errors in Danish”; Technical report, Copenhagen: Center for Sprogteknologi, 1998.

[42] O.W. Rambell; “Error Typology for Automatic Proof-reading Purposes”, Master’s thesism, Uppsala University,Department of Linguistics Language Engineering, Automn 2000.

[43] Vladislav Kubofi and Martin Platek; “A Grammar Based Approach To A Grammar Checking Of Free Word Order Languages”; Proceedings of the 15th conference on Computational linguistics–Vol. 2, ACL, Kyoto, Japan, pp. 906-910, 1994.

[44] Pasi Tapanainen and Timo Järvinen; “A Non-Projective Dependency Parser”; Proceedings of the fifth conference on Applied natural language processing, ACL, Washington, DC, pp. 64-71, 1997.

[45] Jean-Pierre Chanod, Marc El-Beze And Sylvie Guillemin-Lanne; "Coupling An Automatic Dictation System With A Grammar Checker"; PROC OF COLING-92, NANTES. AUG. 23-28, 1992.

[46] G.E. Heidorn; "*Natural Language Inputs to a Simulation Programming System";* Ph.D. dissertation, Yale University, 1972.

[47] Jong C. Park, Martha Palmer and Clay Washburn; "An English Grammar Checker as a Writing Aid for Students of English as a Second Language"; Computer & Information Science – Citeseer.

[48] Richard H. Wojcik, Philip Harrison and John Bremer; "Using Bracketed Parses to evaluate a Grammar Checking application"; Proceedings of the 31st annual meeting on Association for Computational Linguistics, Columbus, Ohio, pp. 38-45, 1998.

[49] Flora Ramirez Bustamante and Fernando Sanchez Leon; "GramCheck: A Grammar and Style Checker"; Proceedings of the 16th conference on Computational linguistics – Vol. 1, ACL, Copenhagen, Denmark, 175-181, 1996.

[50] Khaled F. Shaalan; "Arabic GramCheck: a grammar checker for Arabic"; Software: Practice and Experience, Vol. 35, No. 7, John Wiley & Sons, Ltd., pp. 643-665, 2005.

[51] Jorge Kinoshita, Laís N. Salvador, Carlos E. D. Menezes and William D. C. M. Silva; "CoGrOO – an *OpenOffice* Grammar Checker"; IEEE Seventh International Conference on Intelligent Systems Design and Applications, 2007.

[52] *John Lee and Stephanie Seneff*; "AN ANALYSIS OF GRAMMATICAL ERRORS IN NON-NATIVE SPEECH IN ENGLISH"; Spoken Language Technology Workshop, IEEE, pp. 89-92, 2008.

[53] K. Knight and I. Chander; "Automated postediting of documents"; in *Proc. AAAI*, 1994.

[54] M. Chodorow, J. R. Tetreault, and N.-R. Han; "Detection of grammatical errors involving prepositions"; in *Proc. ACL-SIGSEM Workshop on Prepositions*, 2007.

[55] Jens Eeg-Olofsson and Ola Knutsson; "Automatic Grammar Checking for Second Language Learners – the Use of Prepositions"; Proceedings of the 21st annual ACM symposium on User interface software and technology, Monterey, CA, USA , pp. 121-130, 2008.

[56] Anna Sågvall Hein; "A Chart-Based Framework for Grammar Checking: Initial Studies"; Proc. of 11th Nordic Conference in Computational Linguistic, 1998.

[57] Kyongho Min and William H. Wilson; "Integrated Correction of Ill-Formed Sentences"; Proceedings of the 21st annual ACM symposium on User interface software and technology, Monterey, CA, USA, pp. 369-378, 1997.

[58] K. Min, and W. H. Wilson; "Are Efficient Natural Language Parsers Robust?"; *Eighth Australian Joint Conference on Artificial Intelligence*, 283-290, 1995.

[59] Tsuneaki Kato; "Yet Another Chart-Based Technique for Parsing Ill-Formed Input"; In Natural Language Processing, Information Processing Society Of Japan, 83-100, in Japanese, 1991.

[60] T. Theeramunkong and H. Tanaka; "Analyzing ill-formed input with Parallel chart-based techniques"; Department of Computer Science, Tokyo University.

[61] Masaru Tomita; "*Efficient parsing for natural language: a fast algorithm for practical systems*"; Dordrecht, Kluwer, 1986.

[62] T. Vosse; "Detecting and Correcting Morpho-syntactic Errors in Real Texts"; The Third Conference on Applied Natural Language Processing, ACL, 1992.

[63] Geert Adriaens and Dirk Schreors; "FROM COGRAM TO ALCOGRAM: TOWARD A CONTROLLED English GRAMMAR CHECKER"; Proceedings of the 14th conference on Computational linguistics – Vol. 2, ACL, Nantes, France, pp. 595-601, 1992.

[64] G. Adriaens, "Simplified English Grammar and Style Correction in an MT Framework: The LRE SECC Project", In *Proceedings of the 16th Conference on Translating and the Computer (London)*, pp. 78-88, 1994. Also in *Aslib Proceedings*, 47 (3), pp. 73-82, March 1995.

[65] R. Granger; "The NOMAD system: expectation-based detection and correction of errors during understanding of syntactically and semantically ill-formed text", American Journal of Computational Linguistics, 9(3-4), pp. 188-196, 1983.

[66] Hammad Kabir; "Two-Pass Parsing Implementation for an Urdu Grammar Checker"; INMIC 2002.

[67] Eric Steven Atwell; "HOW TO DETECT GRAMMATICAL ERRORS IN A TEXT WITHOUT PARSING IT"; Proceedings of the third conference on European chapter of the Association for Computational Linguistics, Copenhagen, Denmark, pp. 38-45, 1987.

[68] Eric Steven Atwell; "*LOB Corpus Tagging Project: Manual Pre-edit Handbook*"; Departments of Computer Studies and Linguistics, University of Lancaster, 1981.

[69] Eric Steven Atwell; "*LOB Corpus Tagging Project: Manual Postedit Handbook (A mini-grammar of LOB Corpus English, examining the types of error commonly*

made during automatic (computational) analysis of ordinary written English"; Departments of Computer Studies and Linguistics, University of Lancaster, 1982.

[70] Leech, Geoffrey, Roger Garside and Eric Steven Atwell; "The Automatic Grammatical Tagging of the LOB Corpus"; in *Newsletter of the International Computer Archive of Modern English (ICAME NEWS)* 7: 13-33, Norwegian Computing Centre for the Humanities, Bergen University, 1983.

[71] D. Gential, J. Courtin et al; "From Detection/Correction to Computer Aided Writing"; *Proceedings of 15th international conference on computational Linguistics (COL1NG-92):10]* 3-1018, 1992.

[72] Boitet; "Representation and computation of units of translation for Machine Interpretation of spoken texts", GETA & ATR Tech, Report-I-0035, August 1988.

[73] Gential D., J. Courtin and J. Menezo; "Towards a more user-friendly correction"; in Proceedings of the 16th International conference on computational Linguistics (COLING- 94), Kyoto, Japan, 1994.

[74] Donald Hindle ; "Deterministic Parsing of Syntactic Non-fluencies"; Proceedings of the 21st annual meeting on Association for Computational Linguistics, Cambridge, Massachusetts, pp. 123-128, 1983.

[75] Mitchell P. Marcus; "*A Theory of Syntactic Recognition for Natural Language";* MIT Press: Cambridge, MA, 1980.

[76] Ralph M. Weischedel and Lance A. Ramshaw; "Reflections on the Knowledge Needed to Process Ill-Formed Language"; Proceedings of the *Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Colgate University, Hamilton, New York, August 14-16, 1985.

[77] R. Weischedel and N. Sondheimer; "Meta-rules as Basis for Processing Ill-formed Input"; American Journal of Computational Linguistics, 9(3-4): 161-177, 1983.

[78] Ralph Grishman and Ping Peng; "RESPONDING TO SEMANTICALLY ILL-FORMED INPUT"; Proceedings of the second conference on Applied natural language processing, ACL, Austin, Texas, pp. 66-70, 1988.

[79] Antje Schmidt-Wigger; "Grammar and Style Checking for German"; Proceedings of the Second International Workshop on, 1998

[80] J. Haller; "MULTILINT, A Technical Documentation System with Multilingual Intelligence"; Translation and Computer 18, London, Aslib, The Association for Information Management, Information House, 1996.

[81] R. H. Wojcik, J.E. Hoard and K.C. Holzhauser; "The Boeing Simplified English Checker"; Proceedings of the International Conference on Human Machine Translation and Artificial Intelligence in Aeronautics and Space, Toulouse, Centre d'Etudes et de Recherché de Toulouse, pp. 43-57, 1990.

[82] M. Carl and A. Schmidt-Wigger; "Shallow Post Morphological Processing with KURD"; in Proceedings of NeMLaP, Sydney, 1998.