تأثیر ساخت‌واژه‌ها در تجزیهٔ وابستگی زبان فارسی

مجتبی خلاش و بهروز مینایی بیدگلی
دانشکدهٔ مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران

چکیده

سامانه‌های مبتنی بر داده به‌راحتی می‌توانند به سایر زبان‌ها یا دامنه‌ها منتقل شوند. استفاده از این رویکرد در تجزیهٔ وابستگی منجر به ارائهٔ روش‌های مبتنی‌بر داده شد که تنها نیازمند پیکره‌ای حاوی جملات و درخت وابستگی متناظر با آن به‌عنوان دادهٔ آموزشی است. الگوریتم‌های طراحی‌شده برای تجزیهٔ وابستگی با وجود صحت بالا در زبان انگلیسی، بر روی دسته‌ای از زبان‌ها با افت صحت مواجه می‌شوند که دلیل این امر را می‌توان در پر رنگ‌تر بودن عامل بی‌ترتیبی و غنای ساخت‌واژی آنها دانست. این بدان معناست که سامانه‌های مبتنی‌بر داده نیازمند انتخاب خصوصیات و تنظیم دقیق پارامترها به‌منظور رسیدن به کارایی بهینه هستند.

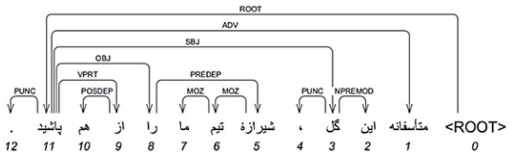
زبان فارسی که به‌تازگی پیکرهٔ وابستگی برای آن طراحی شده است، جزو زبان‌هایی است که دو عامل بی‌ترتیبی و غنای ساخت‌واژی را دارد. در این مقاله سعی شده است عوامل تأثیرگذار بر کاهش صحت تجزیهٔ وابستگی در زبان فارسی شناسایی و راهکارهایی برای بهبود صحت آن ارائه شود.

واژگان کلیدی: تجزیهٔ وابستگی، زبان‌های از نظر ساخت‌واژی غنی، خصوصیات ساخت‌واژی.

۱- مقدّمه

دستور وابستگی یکی از نظریه‌های زبان‌شناختی است که در بررسی نحو و دستور زبان به کار می‌رود. این دستور، تجزیهٔ جمله به دو قسمت نهاد و گزاره را شیوهٔ مناسبی برای بررسی ساخت اطلاعاتی جمله می‌داند؛ اما معتقد است برای تجزیه، تحلیل را باید از فعل (به‌عنوان مرکز ثقل جمله) آغاز کرد. بر اساس این تعریف تحلیل نحوی جمله از فعل آغاز می‌شود که در آن مفاهیم نهاد و گزاره، یکسره کنار گذاشته می‌شود و فعل در ساختی بالاتر از فاعل قرار می‌گیرد (طبیب‌زاده، ۱۳۸۵).

هر ساخت نحوی به‌صورت رابطهٔ وابستگی بین عناصر هسته و وابسته توصیف می‌شود که در مجموع یک درخت وابستگی[1] به دست می‌آید. شکل (۱-۱) نمونه‌ای از درخت وابستگی را نشان می‌دهد که مطابق با آن فعل جمله (*پاشید*) در ریشهٔ درخت قرار دارد، وابسته‌ها و وابستهٔ وابسته‌های فعل سایر اجزای درخت را شکل می‌دهند. به‌منظور سادگی محاسباتی، یک واژهٔ مصنوعی به نام ROOT به ابتدای جمله اضافه می‌شود که ریشهٔ درخت خواهد بود و با برچسب ROOT به فعل جمله متصل می‌شود.



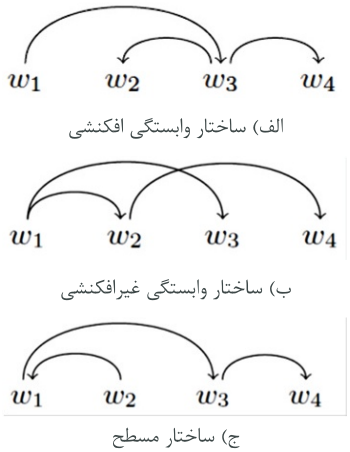
(شکل(۱- ۱)): نمونه‌ای از درخت وابستگی

درخت‌های وابستگی براساس وضعیت تقاطع یال‌هایشان به دو نوع افکنشی[2] و غیرافکنشی تقسیم می‌شوند. در درخت افکنشی، یال‌ها هیچگاه یکدیگر را قطع

[1] Dependency tree

[2] Projective

نمی‌کنند؛ در مقابل درخت غیرافکنشی حداقل دو یال وجود دارد که یکدیگر را قطع می‌کنند (Kübler et al., 2009). فرض افکنشی‌بودن درخت در یک الگوریتم منجر به کاهش فضای جستجو و افزایش سرعت اجرای آن می‌شود؛ اما درصدی از جملات را که ماهیت غیرافکنشی دارند به‌اشتباه تجزیه خواهند شد. خوشبختانه درصد درخت‌های غیرافکنشی ناچیز است و این فرض چندان نامعقول نخواهد بود. تلاش‌هایی برای ارائهٔ ساختاری عمومی‌تر از افکنشی که در عین حال سربار محاسباتی را نیز افزایش ندهد، انجام شده است. یکی از این تلاش‌ها پیشنهاد ساختارهای مسطح[1] هستند که تنها تفاوت آن با ساختار افکنشی، نبود واژهٔ مصنوعی ROOT است که ساخت نحوی تولیدشده، گراف خواهد شد. تفاوت این سه ساختار در شکل (۱-۲) نشان داده شده است. نکتهٔ اصلی در این ساختار قابلیت تعمیم آن به ساختار m-مسطح است. ساختار m-مسطح ساختاری است که از حداقل m ساختار مسطح تشکیل شده که هر ساختار به‌تنهایی هیچ یال متقاطعی ندارد؛ اما تقاطع بین یال‌های دو ساختار مسطح مجاز است. بنابراین تعریف هر ساختار وابستگی غیرافکنشی را می‌توان با ساختار m-مسطح بازنمایی کرد. مسأله ۲-مسطح به‌صورت خطی قابل حل است؛ اما راه‌حل آن برای مقادیر m بزرگ‌تر از ۲ غیرقطعی کامل[2] خواهد بود.



(شکل (۱-۲)): انواع درخت‌های وابستگی
(Wróblewska and Woliński, 2011)

راهکار ارائه‌شده در حوزهٔ زبان‌شناسی رایانه‌ای برای تحلیل نحوی، تجزیهٔ وابستگی نام دارد که از دستور وابستگی الهام گرفته شده است. روش‌های ارائه‌شده برای استنتاج دستور در این راهکار به دو دستهٔ «مبتنی‌بر داده» و «مبتنی بر دستور» تقسیم می‌شود. از این میان روش‌های مبتنی بر داده به‌دلیل ماهیت مستقل از زبان مورد توجه بیشتری قرار گرفتند. الگوریتم‌های این دسته نیاز به پیکرهٔ نمادگذاری‌شده[3] به‌عنوان دادهٔ آموزشی برای ارائه به روش‌های یادگیری ماشینی هستند تا جملات دریافتی را تجزیهٔ نماید. الگوریتم‌های مبتنی‌بر داده براساس نحوهٔ برخورد با اجزای جمله طی فرایند تجزیه به دو دسته تقسیم می‌شوند:

- روش‌های مبتنی بر گذار: این راهکارها از یک سامانهٔ گذار یا ماشین حالت برای نگاشت جمله به درخت وابستگی استفاده می‌کنند. واژه‌های جمله یکی پس از دیگری از داخل پشتهٔ ورودی دریافت شده و با استفاده از مجموعهٔ اعمال‌گذار از پیش تعریف‌شده، کار تجزیه را انجام می‌دهند. مسألهٔ یادگیری معادل استنتاج الگو برای پیش‌بینی گذار بعدی براساس تاریخچهٔ گذارهای مشاهده‌شده در داده‌های آموزشی است. مسألهٔ تجزیه معادل ساخت رشته‌ای از گذارهای بهینه برای جملهٔ ورودی توسط الگوی به دست آمده است. در این روش‌ها با یکبار گذر از روی واژه‌های جمله، درخت وابستگی تولید شده و به همین جهت اکثر این الگوریتم‌ها دارای پیچیدگی زمانی و حافظهٔ خطی هستند. عیب اصلی این الگوریتم‌ها استفاده از اطلاعات محلی و ماهیت حریصانهٔ آنهاست که منجر به انتشار خطا به سایر بخش‌های جمله می‌شود.
- روش‌های مبتنی بر گراف: این راهکارها فضایی از گراف‌های وابستگی نامزد برای جمله را در نظر می‌گیرد. مسألهٔ یادگیری معادل ارائهٔ الگو برای انتساب امتیاز به گراف وابستگی نامزد جمله و مسألهٔ تجزیه معادل یافت گراف وابستگی با بیشترین امتیاز برای جملهٔ ورودی توسط الگوست که برای این منظور از الگوریتم درخت پوشای بیشینه[4] استفاده می‌شود. این الگوریتم‌ها از اطلاعات سراسری جمله استفاده می‌کنند که مانع از انتشار خطا می‌شود؛ اما پیچیدگی زمانی و حافظهٔ بیشتری برای اجرا می‌طلبد.

[1] Planar
[2] NP-Complete
[3] Annotated corpus
[4] Maximum spanning tree (MST)

به‌منظور رفع نقائص و بهره‌برداری از مزایای این دو دسته الگوریتم، روش‌های ترکیبی پیشنهاد شده که به سه دسته زیر تقسیم می‌شوند:

- ترکیب در زمان آموزش (پشته‌سازی[1]): در این روش دو سطح تجزیه تعریف می‌شود که تجزیه‌گر حاضر در سطح دوم علاوه‌بر اطلاعات نمادگذاری‌شدهٔ موجود، از درخت وابستگی پیش‌بینی‌شده توسط تجزیه‌گر سطح اول نیز استفاده می‌کند (Nivre and McDonald, 2008)، (Martins et al., 2008).
- ترکیب در زمان تجزیه (یادگیری گروهی[2]): در این روش درخت‌های وابستگی پیش‌بینی‌شده توسط مجموعه‌ای از تجزیه‌گرهای پایه‌ای دریافت شده و سعی می‌شود، بهترین درخت تولید شود. ساده‌ترین رویکرد، استفاده از رأی اکثریت است (Zeman and Žabokrtský, 2005) که با وجود دقت مناسب، درخت تولیدشده در اکثر مواقع خوش‌ساخت نیست. به همین جهت از روش‌های تجزیهٔ مجدد[3] استفاده می‌شود (Sagae and Lavie, 2006) که طی آن کلیهٔ درخت‌های پیش‌بینی‌شده تجمیع و گراف وزن‌دار تولید شده که توسط الگوریتم درخت پوشای بیشینه قابل حل است (Hall et al., 2007)، (Surdeanu and Manning, 2010).
- روش‌های ابتکاری: در این روش‌ها سعی می‌شود به نوعی مدل یادگیری و یا تجزیه اصلاح شود که به‌طور همزمان عملکردی مشابه هر دو دسته الگوریتم داشته باشد (Zhang and Clark, 2008)، (Bohnet and Kuhn, 2012).

برای ارزیابی کارایی یک سامانهٔ تجزیهٔ وابستگی، معیارهای مختلفی وجود دارد که مهم‌ترین آنها عبارتند از:

۱. امتیاز اتصال بدون برچسب[4] (UAS): مطابق فرمول (۱) درصدی از یال‌های درخت وابستگی است که واژهٔ هسته به‌درستی پیش‌بینی شده است.

$$LAS=\frac{\#\text{Arcs with correct head}}{\#\text{Total arcs}} \qquad (۱)$$

۲. امتیاز اتصال با برچسب[5] (LAS): مطابق فرمول (۲) درصدی از یال‌های درخت وابستگی است که به‌صورت همزمان واژهٔ هسته و برچسب وابستگی آنها به‌درستی پیش‌بینی شده است.

$$LAS=\frac{\#\text{Arcs with correct head and deprel}}{\#\text{Total arcs}} \qquad (۲)$$

کلیهٔ نتایج این مقاله براساس معیار LAS ارائه شده‌اند.

۲- پیکرهٔ وابستگی

دو پیکرهٔ وابستگی برای زبان فارسی در دسترس موجود است:

- دادگان (نسخهٔ ۰,۱): توسط گروه پژوهشی دادگان تحت حمایت دبیرخانهٔ شورای عالی اطلاع رسانی تهیه شده که شامل ۱۲۴۵۵ جمله (۱۸۹۵۷۲ واژه) است (Rasooli et al., 2011).
- UPEDT[6] (نسخه آزمایشی): در دانشگاه اوپسالای سوئد با استفاده از نسخهٔ اصلاح شدهٔ پیکرهٔ بی‌جن‌خان طراحی شده که شامل ۱۲۸۲ جمله (۲۶۰۶۵ واژه) است (Seraji et al., 2012).

کلیهٔ آزمایش‌های انجام‌شده در این مقاله با استفاده از دادگان صورت گرفته است که به‌دلیل کوچک‌بودن دادهٔ آموزشی از اعتبارسنجی متقابل ۵ باره[7] استفاده شده است.

۳- الگوریتم‌های پایه‌ای

در بخش قبل روش‌های مبتنی‌بر داده را به دو دستهٔ مبتنی بر گذار و مبتنی بر گراف تقسیم کردیم. برای انجام آزمایش‌های این مقاله، یک الگوریتم از هر کدام از این دو دسته استفاده شده است.

۱. تجزیه‌گر مبتنی بر گذار MaltParser: این تجزیه‌گر شامل ۹ الگوریتم مختلف در حالت‌های افکنشی، غیرافکنشی و مسطح است. الگوریتم‌های مختلف در این تجزیه‌گر مورد بررسی قرار گرفته که نتایج حاصل از آن در جدول(۳-۱) ارائه شده است. براساس نتایج، الگوریتم غیرافکنشی کاوینگتون بهترین صحت را ارائه کرده که در سایر آزمایش‌ها از این الگوریتم استفاده خواهد شد.

۲. تجزیه‌گر مبتنی بر گراف MSTParser: این تجزیه‌گر دو الگوریتم برای حالت‌های افکنشی و غیرافکنشی دارد که هر کدام می‌تواند از خصوصیات مرتبهٔ اول و یا دوم استفاده نماید. الگوریتم‌های مختلف در این تجزیه‌گر

[1] Stacking
[2] Ensemble
[3] Re-parsing
[4] Unlabeled Attachment Score
[5] Labeled Attachment Score
[6] Uppsala Persian Dependency Treebank
[7] 5-fold cross-validation

مورد بررسی قرار گرفته که نتایج حاصل از آن در جدول (۳-۲) ارائه شده است. براساس نتایج، الگوریتم غیرافکنشی مرتبهٔ دوم بهترین صحت را ارائه کرده که در سایر آزمایش‌ها از این الگوریتم استفاده خواهد شد.

(جدول (۳-۱)): صحت الگوریتم‌های مختلف موجود در MaltParser

نوع	الگوریتم	صحت پایه‌ای
افکنشی	Arc Eager	۸۱/۶۸
	Arc Standard	۷۹/۸۲
	Covington	۸۱/۶۱
	Stack	۷۹/۹۴
غیرافکنشی	Covington	**۸۳/۱۶**
	Stack Eager	۷۷/۲۱
	Stack Lazy	۸۱/۳۱
مسطح	Planar	۸۰/۵۰
	2-Planar	۸۰/۶۸

(جدول (۳-۲)): صحت الگوریتم‌های مختلف موجود در MSTParser

مرتبه	نوع	صحت پایه‌ای
اول	افکنشی	۸۲/۹۴
	غیرافکنشی	۸۳/۵۹
دوم	اکنشی	۸۳/۷۴
	غیرافکنشی	**۸۴/۵۹**

(جدول (۳-۳)): نتایج سه مرحلهٔ بهینه‌سازی MaltOptimizer

مرحلهٔ اول	مرحلهٔ دوم	مرحلهٔ سوم	دادهٔ آزمون
۸۲/۲۵	۸۳/۶۴	۸۵/۰۶	**۸۴/۷۶**

۳-۱- بهینه‌سازی

تجزیه‌گر MaltParser شامل الگوریتم‌های مختلفی است که هر کدام پارامترهای خود را دارند. علاوه‌براین نحوه استفاده از خصوصیات نقش مهمی در کارایی تجزیه‌گر خواهد داشت. به‌منظور بهینه‌سازی و انتخاب پارامترهای مناسب، ابزار MaltOptimizer ارائه شده است (Ballesteros and Nivre, 2012). این ابزار کل دادهٔ آموزشی را دریافت کرده و طی سه مرحله وظیفهٔ بهینه‌سازی را انجام می‌دهد.

- مرحلهٔ اول (ارزیابی و تحلیل داده): در این مرحله تنها اطلاعات آماری جمع‌آوری می‌شود تا در مراحل بعدی مورد استفاده قرار گیرد.
- مرحلهٔ دوم: هدف این فاز انتخاب الگوریتم مناسب و بهینه‌سازی پارمترهای آن است.
- مرحلهٔ سوم: بهینه‌سازی نحوه استفاده از خصوصیات و پارامترهای الگوریتم یادگیری صورت می‌گیرد.

نتایج حاصل از سه مرحلهٔ بهینه‌سازی در جدول (۳-۳) ارائه شده است. در پایان مرحلهٔ دوم الگوریتم کاوینگتون غیرافکنشی به‌عنوان بهترین الگوریتم انتخاب و پارامترهای آن بهینه‌سازی شد. نتیجهٔ نهایی سه مرحلهٔ بهینه‌سازی، ۱/۶درصد بهبود را نسبت به صحت پایه‌ای ارائه‌شده در جدول (۳-۱) نشان می‌دهد.

۴- ساخت‌واژه

ساخت‌واژه به مطالعه ساختار درونی واژه‌ها می‌پردازد که در آن واژه‌سازی از واحدهای با معنای واضح توسط تکواژ[1]ها و شیوه‌های هم‌نشینی آنها با یکدیگر در قالب‌های نحوی بررسی می‌شود. تکواژ یا واژک کوچک‌ترین واحد زبانی است که دارای معنای دستوری یا واژگانی بوده و قابل تجزیه به واحدهای معنی‌دار دیگر نیست. براساس Jurafsky and Martin (2000) تکواژ به دو نوع «ریشه» یا «وند[2]» تقسیم می‌شود؛ اما این تقسیم‌بندی برای زبان فارسی جامع نیست (به‌عنوان مثال تکواژ «م» در «کتابم» نه ریشه است و نه وند بلکه واژه‌بست است). در زبان فارسی تکواژها دو نوع دسته‌بندی «آزاد-وابسته» و «واژگانی-دستوری» تقسیم‌بندی می‌شوند. بر این اساس چهار دسته تکواژ داریم: تکواژ آزاد واژگانی (شامل اسم، فعل، صفت، قید)، تکواژ آزاد دستوری (شامل ضمایر، حروف اضافه، حرف ربط و ندا، نقش نمای اضافه)، تکواژ وابستهٔ واژگانی، تکواژ وابستهٔ دستوری (شامل واژه‌بست و وند). زبان‌هایی که اطلاعات دستوری قابل توجهی در ساخت‌واژهٔ آن‌ها وجود دارد را زبان‌های از نظر ساخت‌واژی غنی[3] نامند. از خصوصیات بارز این زبان‌ها بی‌ترتیبی[4] است که این خصوصیت منجر به تولید ساختارهای غیرافکنشی خواهد شد. ویژگی دیگر این زبان‌ها میزان تصریف بالای آن‌هاست که منجر به کاهش اعتماد به داده‌های لغوی می‌شود.

در زبان فارسی بیش از ۱۲۰ تصریف مختلف فعل وجود دارد که با درنظر گرفتن ضمایر پیوسته، برای افعال

[1] Morpheme
[2] Affix
[3] Morphologically Rich Languages (MRLs)
[4] Free word-order

گذرا این تعداد به بیش از ۷۰۰ مورد تصریف می‌رسد (رسولی، ۱۳۹۰) که این نشان دهنده تصریف بالا در زبان فارسی است. با توجه به وجود خاصیت بی‌ترتیبی (Shamsfard, 2011)، می‌توان زبان فارسی را جزو زبان‌های از نظر ساخت‌واژی غنی دانست.

۵- چارچوب بررسی

شواهد فراوانی وجود دارد که کاربرد مدل‌های تجزیهٔ احتمالی در زبان‌های از نظر ساخت‌واژی غنی، مستعد کاهش کارایی است. براساس مرجع (Tsarfaty et al., 2010) به‌منظور ترکیب اطلاعات ساخت‌واژی در مدل‌های تجزیه با سه نوع چالش روبه‌رو خواهیم بود:

۱. معماری و تنظیمات اولیه
۲. بازنمایی و مدل‌کردن
۳. تخمین و هموارسازی

در این مقاله به‌منظور بررسی چالش‌های موجود در تجزیهٔ وابستگی زبان فارسی، به بررسی هر کدام از این چالش‌ها خواهیم پرداخت.

۵-۱- معماری و تنظیمات اولیه

در اکثر مدل‌های تجزیه، فرض بر این است که پس از دریافت جمله، قسمت‌بندی روی آن انجام شده و پیش از ارائه به تجزیه‌گر خصوصیات ساخت‌واژی و صرفی به آن اضافه خواهد شد. این در حالی است که در مسائل دنیای واقعی چنین فرضی وجود ندارد. قسمت‌بندی جمله به واژه‌ها به‌دلیل وجود ساخت‌واژه‌های غنی، امری ساده و بدیهی نخواهد بود. همچنین تولید خصوصیات ساخت‌واژی مانند برچسب اجزای سخن همراه با مقادیر نوفه‌ای و خطا خواهد بود که صحت تجزیه را تحت تأثیر قرار خواهد داد.

به‌منظور بررسی این چالش در زبان فارسی، تأثیر استفاده از برچسب اجزای سخن در حالت دستی و خودکار مورد بررسی قرار گرفته است. همچنین تأثیر اطلاعات رمزگذاری شده در برچسب اجزای سخن تحت عنوان مجموعهٔ برچسب مورد بررسی قرار گرفته است.

پیکرهٔ دادگان شامل دو مجموعهٔ برچسب ریز و درشت اجزای سخن است که به‌ترتیب شامل ۱۷ و ۳۰ برچسب مختلف است. به‌منظور تولید برچسب خودکار از ابزار MXPOST استفاده شده است که کار برچسب‌زنی را توسط مدل بیشینهٔ آنتروپی انجام می‌دهد. دقت برچسب‌زنی برای برچسب‌های ریز و درشت به‌ترتیب ۸۸/۸۴ و ۹۲/۹۰ درصد بوده است. نتایج حاصل از این آزمایش در جدول (۵-۱) آمده است.

(جدول (۵-۱)): تأثیر دو مجموعه برچسب اجزای سخن در حالت دستی و خودکار

مجموعه برچسب	حالت	MaltParser	MSTParser
POS (۳۰ برچسب)	دستی	۸۴/۷۳	۸۴/۵۶
	خودکار	۸۴/۰۶	۷۴/۷۹
CPOS (۱۷ برچسب)	دستی	۸۴/۴۸	۸۴/۵۷
	خودکار	۷۴/۳۰	۷۴/۴۶

نکته قابل توجه در هر دو تجزیه‌گر، افت حدود ده درصدی هنگام استفاده از برچسب‌های خودکار نسبت به حالت دستی است. در مورد دو مجموعهٔ برچسب نیز عمکرد دو تجزیه‌گر عکس یکدیگر بوده است.

- تجزیه‌گر MaltParser با کاهش مجموعهٔ برچسب‌ها در حالت دستی با افت صحت و در حالت خودکار با افزایش صحت مواجه شده است که نشان می‌دهد نسبت به اطلاعات رمزنگاری‌شده در برچسب اجزای سخن حساس است. اگر این اطلاعات به‌طور کامل مطمئن باشد با افزایش اطلاعات صحت بهبود می‌یابد، اما در صورت عدم اعتماد به این اطلاعات، کاهش مجموعهٔ برچسب می‌تواند تا حدی این افت صحت را جبران کند.
- تجزیه‌گر MSTParser برخلاف تجزیه‌گر مبتنی بر گذار در حالت دستی با افزایش اطلاعات تغییر چندانی نمی‌کند و حتی مقدار اندکی با کاهش صحت مواجه خواهد شد. در مقابل در حالت خودکار با دریافت اطلاعات بیشتر صحت بهبود می‌یابد. این بدان معناست که تجزیه‌گر مبتنی بر گراف نسبت به اطلاعات نوفه‌ای مقاوم‌تر است.

راهکار مورد بررسی قرار گرفته با نام رویکرد پیایی شناخته می‌شود که به‌صورت سنتی مورد استفاده قرار می‌گرفته است. در این رویکرد مرحلهٔ برچسب‌زنی و تجزیهٔ وابستگی به‌صورت مجزا انجام می‌شود که وجود خطا در مرحلهٔ اول منجر به انتشار خطا در مرحلهٔ بعدی خواهد شد. یکی از راهکارهای کاهش این خطا استفاده از رویکرد همزمان است. در این رویکرد سعی می‌شود دو وظیفهٔ

برچسب‌زنی و تجزیه به‌صورت همزمان انجام شود و طی آن برچسب‌زن و تجزیه‌گر از اطلاعات فراهم‌شده توسط یکدیگر برای بهبود کارایی استفاده نمایند (Naradowsky, and Smith, 2011)، (Bohnet and Nivre, 2012).

۵-۲- بازنمایی و مدل‌کردن

ورودی سامانهٔ تجزیه باید بازتاب کنندهٔ اطلاعات ساخت‌واژی باشد. اطلاعات ساخت‌واژی زیادی وجود دارد که می‌تواند به تجزیه‌گر ارائه شود. هدف از این آزمایش‌ها یافتن اطلاعات ساخت‌واژی بهینه برای ارائه به تجزیه‌گر است. خصوصیات ساخت‌واژی زیر به‌صورت دستی در پیکره نمادگذاری شدند:

- اتصال (A): این خصوصیت، وابستگی یا استقلال یک واژه را نشان می‌دهد. به‌عنوان مثال جملهٔ «رهایم نمی‌کند» را در نظر بگیرید:
  - واژهٔ مستقل (نمی‌کند)
  - وابسته به واژهٔ پیشین (یم)
  - وابسته به واژهٔ بعدی (رها)
- شمار (N): مفرد یا جمع‌بودن واژه را نشان می‌دهد.
- شخص (P): اول، دوم یا سوم شخص‌بودن واژه را نشان می‌دهد.
- زمان/وجه/نمود (TMA): این خصوصیت برای فعل تعریف می‌شود که بیان‌گر زمان دستوری (محل وقوع رخداد فعل روی خط زمان)، وجه (میزان اجبار، ضرورت، توانایی و غیره) و نمود (تمام‌شدن یا جریان‌داشتن رخداد فعل) است.

برای هر فعل، سه مقوله زمان، وجه و نمود، به‌نحوی با هم آمیخته شدند که بهتر است به جای گزینش مجزای هر یک از آنها به‌عنوان ارزش‌های جداگانه، ترکیب آنها به‌عنوان یک ارزش در نظر گرفته شود. مشاهده شده که در برخی از صیغه‌های افعال میان نمودها تداخل وجود دارد. به‌عنوان مثال صیغهٔ «گذشتهٔ نقلی استمراری اخباری» از لحاظ نمود هم نقلی و هم استمراری است. به‌منظور بررسی این مورد دو خصوصیت زمان و وجه را نیز به‌صورت جداگانه مورد بررسی قرار دادیم.

علاوه‌بر این خصوصیات، چهار خصوصیت مفهومی نیز مورد بررسی قرار دادیم:

- شناسهٔ خوشه معنایی فعل (VC): خوشه‌بندی معنایی افعال برای زبان فارسی انجام شده است (امینیان، ۱۳۹۱) که طی آن ۱۰۸۲ فعل پیکرهٔ بی‌جن‌خان به ۴۳ خوشه تقسیم شدند. این خصوصیت به کلیهٔ افعال موجود در پیکره اضافه شده است.
- شناسهٔ مترادف ادراکی (SID) و فایل مفهومی (SF): این دو خصوصیت توسط شبکهٔ واژگانی فارس‌نت (Shamsfard et al., 2010) به اسامی، صفات و افعال پیکره اضافه شده است. خصوصیت اول شناسهٔ اولین مترادف ادراکی یافت‌شده توسط فارس‌نت و خصوصیت دوم فایل مفهومی متناظر با این مترادف ادراکی از وردنت انگلیسی است.
- خوشه‌بندی واژه (WC): روشی برای ساخت خوشه‌هایی از واژه‌های مشابه است که می‌توان به کلیهٔ واژه‌های موجود در هر خوشه، شناسهٔ خوشه را به‌عنوان خصوصیت نسبت داد. برای این منظور واژه‌ها و ریشه‌ها با طول بیت‌های مختلف امتحان شد که نتایج آن در جدول (۵-۱) ارائه شده است. براساس این نتایج استفاده از ریشه با طول بیت پنج بهترین صحت را داشته که در ادامهٔ مقاله از آن به‌عنوان خصوصیت خوشه‌بندی واژه استفاده شده است.

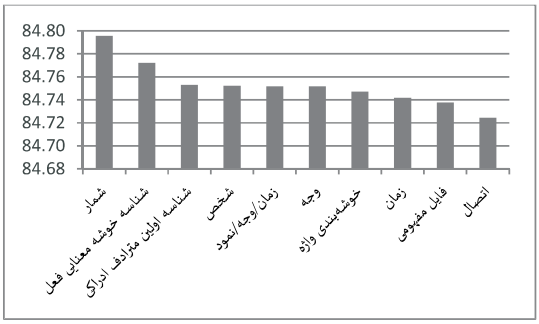
(جدول (۴-۱)): انتخاب روش خوشه‌بندی مناسب

	تعداد بیت	MaltParser	MSTParser
واژه	۳	۸۴/۷۸	۸۴/۴۰
	۵	۸۴/۸۱	۸۴/۴۷
	۷	۸۴/۸۲	۸۴/۵۳
	۹	۸۴/۸۲	۸۴/۵۵
ریشه	۳	۸۴/۸۰	۸۴/۴۲
	۵	**۸۴/۸۵**	۸۴/۵۳
	۷	۸۴/۸۰	۸۴/۵۴
	۹	۸۴/۸۲	**۸۴/۶۳**

به‌منظور کشف تأثیر هر کدام از خصوصیات و پیداکردن بهترین ترکیب آنها، دو آزمایش زیر را انجام دادیم:

- گزینش رو به جلو: ابتدا تمام خصوصیات را کنار گذاشته و تنها یکی از خصوصیات را به تجزیه‌گر ارائه می‌کنیم. نتایج حاصل از این آزمایش در شکل (۵-۱) نشان داده شده است. بر این اساس خصوصیت «شمار» به تنهایی بیشترین تأثیر را بر صحت تجزیه داشته است. در ادامه طی یک رویه تکرارشونده خصوصیتی که بیشترین سهم در بهبود تجزیه داشته باشد به مجموعهٔ جاری اضافه خواهد شد. این رویه زمانی پایان خواهد

یافت که افزودن هیچ خصوصیتی منجر به بهبود صحت تجزیه نشود. با انجام این آزمایش‌ها به مجموعه‌ای شامل چهار خصوصیت «*شمار، فایل مفهومی، خوشه معنایی فعل و شناسه اولین مترادف معنایی*» به صحت ۸۴/۸۷ رسیدیم.



(شکل (۵-۱)): تأثیر هر یک از ۱۰ خصوصیت ساخت‌واژی و مفهومی

- گزینش رو به عقب: ابتدا تمام خصوصیات یک‌جا به تجزیه‌گر ارائه می‌کنیم که صحت ۸۴/۸۹ درصدی به دست می‌آید. در ادامه طی یک رویه تکرار شونده خصوصیتی که حذف آن بیشترین سهم در بهبود صحت تجزیه داشته باشند از مجموعهٔ جاری حذف خواهد شود. این رویه زمانی پایان خواهد یافت که حذف هیچ خصوصیتی منجر به بهبود صحت تجزیه نشود. با انجام این آزمایش‌ها تنها با حذف دو خصوصیت «*وجه، شناسه اولین مترادف معنایی*» به صحت ۸۴/۹۱ رسیدیم.

۵-۳- تخمین و هموارسازی

یکی از ویژگی‌های زبان‌های از نظر ساخت‌واژی غنی، تمایل به تولید لغات بیشتر است که منجر به نرخ بالای لغات خارج از واژگان و تشدید مسأله تنکی داده‌های لغوی می‌شود. برای حل این مشکل چهار راهکار زیر مورد ارزیابی قرار گرفت:

- استفاده از ریشه به جای خود واژه (در این مقاله از ریشه‌های نمادگذاری‌شده در پیکره استفاده شده است، اما برای به‌دست‌آوردن ریشه در متون خام، می‌توان از ابزار پارسی‌پرداز (Sarabi et al., 2013) استفاده کرد که ریشه‌هایی به فرمت دادگان و با توجه به مسائل افعال مرکب فارسی تولید می‌کند).
- جایگزین کردن اعداد با نماد <num> نشان‌دهندهٔ عدد (نرمال‌کردن)
- جایگزین‌کردن اعداد با نمادهایی نشان‌دهندهٔ اعداد یک رقمی، دو رقمی و غیره (بلوکه‌بندی)
- استفاده از فایل‌های مفهومی به جای خود واژه (هدف تقسیم واژه‌ها به گروه‌های سطح بالاتر است. استفاده از فایل‌های مفهومی مثل اسامی حیوانات، اسامی نشان‌دهنده زمان، افعال مربوط به آب و هوا، افعال مربوط به پوشاندن و غیره (Agirre et al., 2011)).

نتایج جدول (۵-۲) نشان می‌دهد که رویکرد واحدی برای دو تجزیه‌گر منجر به بهبود نخواهد شد. برای MaltParser بلوکه‌بندی و برای MSTParser استفاده از ریشه بیشترین تأثیر در کاهش این مشکل خواهد داشت.

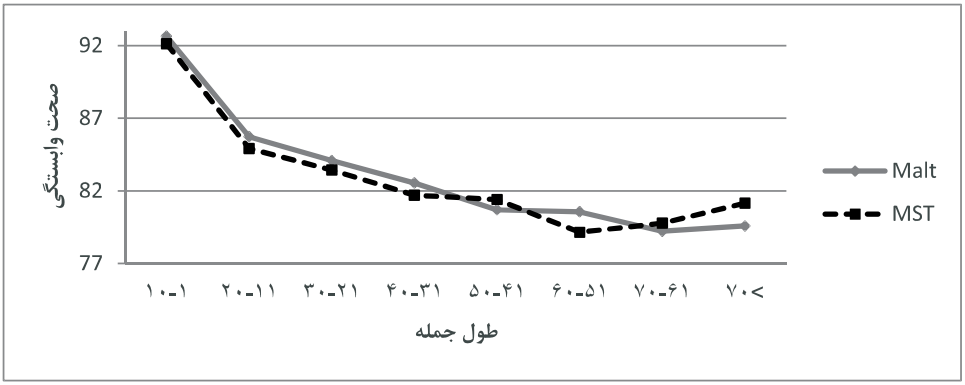
(جدول (۵-۲)): راهکارهای مختلف کاهش تنکی داده‌های لغوی

نوع		MaltParser	MSTParser
استفاده از ریشه		۸۴/۵۹	۸۴/۷۵
نرمال کردن اعداد	انگلیسی	۸۴/۷۹	۸۴/۵۹
	فارسی و انگلیسی	۸۴/۷۹	۸۴/۵۰
بلوکه‌بندی اعداد	انگلیسی	۸۴/۸۰	۸۴/۵۵
	فارسی و انگلیسی	۸۴/۸۰	۸۴/۵۲
فایل مفهومی		۸۴/۵۸	۸۴/۳۱

۶- تحلیل خطا

به‌منظور تحلیل خطا، الگوریتم‌های بهینه‌شده را با مجموعهٔ خصوصیات به دست آمده در گزینش رو به عقب و راهکار کاهش تنکی دادهٔ لغوی ارائه شده در جدول (۵-۲)، بر روی دادهٔ آزمون اعمال کردیم که برای MaltParser و MSTParser به‌ترتیب ۸۵/۳۸ و ۸۴/۸۰ درصد به‌دست آمد.

به‌طور کلی سامانه‌های تجزیه تمایل دارند در جملات طولانی صحت کمتری داشته باشند که دلیل اصلی آن افزایش حضور ساخت‌های نحوی پیچیده در جملات است. تأثیر این عامل بر صحت وابستگی هر کدام از دو تجزیه‌گر در شکل (۶-۱) نشان داده شده است. در جملات با طول کوتاه‌تر MaltParser اندکی بهتر عمل می‌کند؛ اما با افزایش طول جملات به‌دلیل ماهیت حریصانه خطا منتشر شده و منجر به کاهش صحت می‌شود. در جملات با طول بیشتر MSTParser بهتر عمل می‌کند به‌طوری که برای جملات با طول بیش از ۷۰، اختلاف صحت به ۱/۵درصد می‌رسد.



(شکل (۶-۱)): تأثیر طول جمله در صحت وابستگی

با استفاده نتایج به‌دست آمده می‌توان انتظار داشت که استفاده از حد آستانهٔ مناسب بر روی طول جملات به‌منظور ترکیب نظرات دو تجزیه‌گر می‌تواند منجر به بهبود صحت شود. با استفاده از حد آستانهٔ ۶۲ که در آن جملات با طول کمتر از ۶۲ توسط MaltParser و جملات با طول بیشتر از آن توسط MSTParser تجزیه می‌شوند به صحت ۸۵/۴۲ درصدی رسیدیم که اندکی بهتر از صحت هر کدام از دو تجزیه‌گر است. با استفاده از روش تجزیهٔ مجدد به‌منظور ترکیب نتایج دو تجزیه‌گر در زمان تجزیه، صحت ۸۵/۴۶ درصدی به‌دست خواهد آمد[1].

۷- نتیجه

آزمایش‌های ارائه‌شده در این مقاله نشان داد که مهم‌ترین چالش در استفادهٔ عملی از تجزیهٔ وابستگی زبان فارسی، برچسب اجزای سخن در حالت خودکار است. برچسب خودکار استفاده شده در این مقاله فاصلهٔ ده درصدی با برچسب دستی نشان داد که برای کاهش این فاصله باید تدبیری اندیشیده شود. دو راهکار پیشنهادی بهبود برچسب‌زن استفاده شده در رویکرد متوالی و یا استفاده از رویکرد همزمان است. در بخش دیگری از مقاله چهار خصوصیت مفهومی معرفی شده که به‌سادگی به هر متن برچسب نخورده‌ای می‌تواند اعمال شود. در این بین خصوصیت خوشه‌بندی فعل در هر دو رویکرد گزینش رو به جلو و رو به عقب مؤثر واقع شد. برای تولید دو خصوصیت «شناسهٔ مترادف ارداکی» و «فایل مفهومی» از اولین نتیجهٔ به دست آمده توسط فارس‌نت استفاده شده که با استفاده از الگوریتم‌های ابهام‌زدایی معنایی می‌تواند بهبود یابد. مجموعهٔ راه‌کارهای ارائه شده در این مقاله، منجر به بهبود ۲/۲۲ و ۰/۲۱ درصدی صحت تجزیهٔ MaltParser و MSTParser شده است.

۸- مراجع

ا. طبیب‌زاده, ۱۳۸۵، ظرفیت فعل و ساخت‌های بنیادین جمله در فارسی امروز، نشر مرکز.

م. ص. رسولی, ۱۳۹۰, "استنتاج بی‌ناظر ظرفیت فعل در زبان فارسی بر مبنای دستور وابستگی", پایان‌نامهٔ کارشناسی ارشد، دانشگاه علم و صنعت ایران, تهران.

م. امینیان, ۱۳۹۱, "خوشه‌بندی معنایی افعال زبان فارسی", پایان‌نامهٔ کارشناسی ارشد، دانشگاه صنعتی شریف, تهران.

A. F. T. Martins, D. Das, N. A. Smith, and E. P. Xing, 2008, "Stacking dependency parsers", in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 157–166.

A. Wróblewska and M. Woliński, 2011, "Preliminary experiments in polish dependency parsing", in Proceedings of the 2011 international conference on Security and Intelligent Information Systems (SIIS 2011), pp. 279–292.

B. Bohnet and J. Kuhn, 2012, "The Best of Both-Worlds – A Graph-based Completion Model for Transition-based Parsers", in Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 77–87.

[1] برای دریافت کدها و اطلاعات بیشتر از نتایج این آزمایش‌ها می‌توانید به سایت http://nlp.iust.ac.ir مراجعه نمائید.

M. Seraji, B. Megyesi, and J. Nivre, 2012, "Bootstrapping a Persian Dependency Treebank", Linguistic Issues in Language Technology, vol. 7, no. 18, pp. 1–10.

M. Shamsfard, 2011, "Challenges and Open Problems in Persian Text processing", in LTC 2011, pp. 65–69.

M. Shamsfard, A. Hesabi, H. Fadaei, N. Mansoory, A. Famian, S. Bagherbeigi, E. Fekri, M. Monshizadeh, and S. M. Assi, 2010, "Semi Automatic Development Of FarsNet: The Persian Wordnet", in Proceedings of 5th Global WordNet Conference (GWA2010), vol. 358.

M. Surdeanu and C. D. Manning, 2010, "Ensemble models for dependency parsing: cheap and good?", in Proceedings of the North American Chapter of the Association for Computational Linguistics Conference (NAACL-2010), pp. 649–652.

R. Tsarfaty, D. Seddah, Y. Goldberg, S. Kübler, M. Candito, J. Foster, Y. Versley, I. Rehbein, and L. Tounsi, 2010, "Statistical parsing of morphologically rich languages (SPMRL): what, how and whither", in Proceedings of NAACL HLT 2010 First workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010), pp. 1–12.

S. Kübler, R. McDonald, and J. Nivre, Dependency parsing, vol. 1, no. 1. A Publication in the Morgan & Claypool Publishers series, pp. 1–127, 2009.

Y. Zhang and S. Clark, 2008, "A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search", in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008), pp. 562–571.

Z. Sarabi, M. H. Mahyar, and M. Farhoodi. 2013, "ParsiPardaz: Persian Language Processing Toolkit", in 3rd International eConference on Computer and Knowledge Engineering (ICCKE 2013).

مجتبی خلاش: تحصیلات خود را در مقطع کارشناسی مهندسی نرم‌افزار در دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت ایران در سال ۱۳۸۹ به پایان رساند و سپس در همان دانشکده، کارشناسی ارشد هوش مصنوعی و رباتیک را در سال ۱۳۹۱ اخذ کرد. زمینهٔ تحقیقاتی اصلی ایشان پردازش زبان طبیعی و به‌طور خاص تجزیه وابستگی است.

نشانی رایانامهٔ ایشان عبارت است از

mkhallash@gmail.com

B. Bohnet and J. Nivre, 2012, "A Transition-Based System for Joint Part-of-Speech Tagging and Labeled Non-Projective Dependency Parsing", in Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012), pp. 1455–1465.

D. Jurafsky and J. H. Martin, Speech & Language Processing. Pearson Education India, 2000.

D. Zeman and Z. Žabokrtský, 2005, "Improving parsing accuracy by combining diverse dependency parsers", in Proceedings of the 9th International Workshop on Parsing Technologies, pp. 171–178.

E. Agirre, K. Bengoetxea, K. Gojenola, and J. Nivre, 2011, "Improving Dependency Parsing with Semantic Classes", in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11): shortpapers, Vol. 2, pp. 699–703.

J. Hall, J. Nilsson, and J. Nivre, 2007, "Single malt or blended? A study in multilingual parser optimization", in Proceedings of the Conference on Empirical Methods in Natural Language Processing and Conference on Computational Natural Language Learning (EMNLP-CoNLL), pp. 933–939.

J. Lee, J. Naradowsky, and D. A. Smith, 2011, "A discriminative model for joint morphological disambiguation and dependency parsing", in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11), vol. 1, pp. 885–894.

J. Nivre and R. McDonald, 2008, "Integrating graph-based and transition-based dependency parsers", Proceedings of ACL-08: HLT, pp. 950–958.

K. Sagae and A. Lavie, 2006, "Parser combination by reparsing", in Proceedings of the 2006 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2006), pp. 129–132.

M. Ballesteros and J. Nivre, 2012, "MaltOptimizer: An Optimization Tool for MaltParser", in Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 58–62.

M. S. Rasooli, A. Moloodi, M. Kouhestani, and B. Minaei-bidgoli, 2011, "A Syntactic Valency Lexicon for Persian Verbs: The First Steps towards Persian Dependency Treebank", 5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics, pp. 227–231.

بهروز مینایی بیدگلی: دکترای خود را در رشتهٔ علـوم و مهندسـی کـامپیوتر از دانشگاه ایالتی میشیگان آمریکا در سـال ۱۳۸۴ گرفت. تخصص او هوش مصنوعی و داده‌کاوی است. او هم اکنون به‌عنـوان عضو هیئت علمی دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت به تدریس دروس هوش مصنوعی و نرم‌افـزار مشـغول است. ایشان سرپرستی گروه متن‌کاوی بـرای متـون عربـی و فارسی را در پژوهشکدهٔ متن‌کاوی نور نیـز بـر عهـده دارد. از سال ۱۳۸۶ ریاست بنیاد ملی بازی‌هـای رایانـه‌ای بـر عهـدهٔ ایشان است.

نشانی رایانامهٔ ایشان عبارت است از

b_minaei@iust.ac.ir