تحلیل و مقایسه روش های هوشمند شناسایی موجودیت‌های نام‌دار در زبان فارسی

زینب محمدنیا
مهندسی فناوری اطلاعات پزشکی، قم، ایران
golsamohamadi2020@gmail.com

چکیده

شناسایی موجودیت‌های نامدار (NER) به عنوان یک تکنیک استخراج اطلاعات عمل می‌کند که به شناسایی موجودیت‌های نامدار در متن می‌پردازد و شاخه ای از علم پردازش زبان طبیعی (NLP) است. به طور سنتی، چهار روش اصلی برای استخراج این موجودیت‌ها وجود دارد: روش‌های مبتنی بر قواعد، یادگیری ماشین، یادگیری عمیق و ترکیبی از این روش ها. در این مقاله، چند مدل شامل میدان تصادفی شرطی(CRF)، Google BERT، BiLSTM-CRF و فیلترهای محلی (Local Filters) برای پردازش موجودیت‌های زبان فارسی مورد بررسی قرار گرفته و نتایج آن‌ها مقایسه شده است. برای مدل CRF، نتایج نشان می‌دهد که این روش توانسته است دقت ۸۶٫۸۶٪، یادآوری ۸۰٫۲۹٪ و معیار F1 را به میزان ۸۳٫۴۴٪ به‌دست آورد]۲[. مدل Google BERTکه در مسابقه Task 7 از مسابقات NSURL-2019 مقام دوم را کسب کرده و با بهینه‌سازی، نتایج بهتری در رقابت‌های بین‌المللی به‌دست آمده است. نتایج مدل به ترتیب نمره F1 معادل ۸۳٫۵٪ و ۸۸٫۴٪ بر روی مجموعه داده Peyma در ارزیابی سطح عبارت و واژه بوده است]۳[. روشBiLSTM-CRF ، با معرفی مجموعه داده ArmanPersoNERCorpus که بر روی متون فارسی آموزش دیده، نتایج جالبی ارائه می‌دهد]۴[. ترکیب این معماری با جاسازی‌های کلمات پیش‌آموزش‌دیده، نمره F1 معادل ۷۷٫۴۵٪ را به ارمغان آورده است]۴[. در نهایت، رویکرد مبتنی بر مدل فیلترهای محلی (Local Filters) با استفاده از چند دیکشنری موجود در وب، جستجو را با ساختار داده درخت پیشوند (prefix-tree) به طور کارآمد انجام می‌دهد. نتایج این روش در زبان فارسی شامل دقت ۸۸٫۹۵٪، بازیابی ۷۹٫۶۵٪ و نمره F1 برابر با ۸۲٫۷۳٪ با استفاده از[۱]ASEM است]۵[.

کلمات کلیدی: NER، BERT، CRF، BiLSTM، Local Filters

[۱] ASEM (Automatic Semantic Entity Matching) یک روش برای شناسایی و استخراج موجودیت‌های نامدار در متن است. این رویکرد به‌ویژه در پردازش زبان طبیعی و استخراج اطلاعات کاربرد دارد و به شناسایی موجودیت‌هایی مانند اشخاص، سازمان‌ها و مکان‌ها کمک می‌کند.

مقدمه

شناسایی موجودیت‌های نامدار (NER) یا شناسایی موجودیت، یک زیرکار از پردازش زبان طبیعی (NLP) است NER در زبان‌های مختلف توسعه یافته است، اما کارهای محدودی روی متون فارسی به دلیل کمبود منابع و ابزارها انجام شده است. عملکرد NER می‌تواند در زبان‌های کم‌منبع بیشتر مشهود باشد. آماده‌سازی ابزارهای پایه در زبان‌های کم‌منبع با عملکرد بالا می‌تواند راه‌حلی خوب برای چنین زبان‌هایی باشد، در حالی که با مشکل کمبود داده برای آموزش این ابزارها مواجه هستیم. یشتر کارهای انجام شده در شناسایی موجودیت‌های نامدار(NER) در زبان فارسی عمدتاً از روش‌های مبتنی بر قواعد استفاده کرده‌اند که لزوماً در عملکرد خود کامل نیستند. این روش‌ها معمولاً شامل استفاده از دیکشنری‌ها و فهرست‌های موجودیت‌های نامدار هستند. ویژگی‌های روش‌های مبتنی بر قواعد: ۱- وابستگی به منابع: عملکرد این سیستم‌ها به شدت به کیفیت و جامعیت دیکشنری‌ها وابسته است. اگر دیکشنری‌ها شامل تمام موجودیت‌های نامدار نباشند، کارایی سیستم کاهش می‌یابد. ۲- عدم انعطاف‌پذیری: این روش‌ها معمولاً در برابر تغییرات زبانی و ناهماهنگی‌ها مقاوم نیستند.

علاوه بر روش‌های مبتنی بر قواعد، در سال‌های اخیر، استفاده از روش‌های یادگیری ماشین و یادگیری عمیق نیز در حال افزایش است. در یادگیری ماشین مدل‌هایی مانند میدان تصادفی شرطی (CRF) در برخی پروژه‌ها به کار رفته‌اند و به استخراج ویژگی‌ها و شناسایی موجودیت‌ها کمک کرده‌اند. در یادگیری عمیق روش‌های پیشرفته‌تری مانند BiLSTM-CRF و BERT نیز به تازگی در پروژه‌های NER فارسی مورد استفاده قرار گرفته‌اند. این مدل‌ها می‌توانند به‌طور خودکار ویژگی‌ها را از داده‌ها استخراج کرده و نتایج بهتری را ارائه دهند. به طور کلی، در حالی که بیشتر کارها به روش‌های مبتنی بر قواعد وابسته هستند، روند حرکت به سمت استفاده از روش‌های یادگیری ماشین و یادگیری عمیق در حال افزایش است و این تغییر می‌تواند به بهبود عملکرد سیستم‌های NER در زبان فارسی کمک کند. سیستم‌های یادگیری ماشین یاد می‌گیرند که زبان را از طریق استفاده از روش‌های آماری بدون نیاز به برنامه‌نویسی صریح یاد می‌گیرد. مشکل اصلی در استفاده از یادگیری ماشین در پردازش زبان طبیعی، کمبود داده‌های آموزشی حاوی نشانه‌گذاری شده است. با رفع این مشکل، این رویکرد به‌طور قابل توجهی توسعه سیستم‌های NLP را تسریع می‌کند. از مشکلات یادگیری ماشین: ۱- نیاز به داده‌های نشانه‌گذاری‌شده: این روش‌ها به داده‌های آموزشی با نشانه‌گذاری دقیق نیاز دارند که در زبان‌های کم‌منبع مانند فارسی به راحتی قابل دسترسی نیست. ۲- کمبود تنوع در داده‌ها: اگر داده‌های آموزشی شامل تنوع کافی نباشند، مدل ممکن است به خوبی تعمیم نیابد و در داده‌های جدید عملکرد خوبی نداشته باشد. ۳- پیچیدگی در استخراج ویژگی‌ها: استخراج ویژگی‌های مناسب برای آموزش مدل‌ها می‌تواند دشوار باشد و به دانش تخصصی نیاز دارد. روش‌های یادگیری عمیق هم دارای مشکلاتی است که شامل : ۱- نیاز به داده‌های بزرگ: مدل‌های یادگیری عمیق معمولاً به حجم زیادی از داده‌های نشانه‌گذاری‌شده نیاز دارند تا به خوبی آموزش ببینند. این موضوع در زبان‌های کم‌منبع یک چالش بزرگ است. ۲- پیچیدگی محاسباتی: این مدل‌ها به منابع محاسباتی زیادی برای آموزش و اجرا نیاز دارند که می‌تواند هزینه‌بر باشد. ۳- تفسیرپذیری پایی: مدل‌های یادگیری عمیق معمولاً به عنوان "جعبه سیاه" شناخته می‌شوند و درک اینکه چرا یک تصمیم خاص گرفته شده، دشوار است.

یکی از روش‌های معروف یادگیری ماشین که در بسیاری از سیستم‌های NER مانند سیستم Stanford NER که توسط گروه پردازش زبان طبیعی دانشگاه استنفورد توسعه یافته است و یکی از محبوب‌ترین و پرکاربردترین سیستم‌ها برای شناسایی موجودیت‌ها در متون مختلف است مورد استفاده قرار گرفته است، میدان تصادفی شرطی (CRF) است که به‌عنوان مدل‌سازی آماری عمل می‌کند. CRF یک روش یادگیری تحت نظارت است که احتمال توالی‌های برچسب‌گذاری شده ممکن را برای یک توالی مشاهده‌شده مشخص می‌کند. شناسایی موجودیت‌های نام‌دار یکی از وظایف مهم و اساسی در پردازش زبان طبیعی است که بخش‌های مختلف یک متن را به دسته‌های مناسب موجودیت‌های نام‌دار اختصاص می‌دهد. مجموعه‌های مختلفی از دسته‌های موجودیت نام‌دار (NER) معرفی و در مجموعه‌های مختلف برچسب‌گذاری شده استفاده می‌شوند. به عنوان مثال، مجموعه برچسب که شامل شخص، سازمان، مکان، تاریخ، پول، درصد و زمان است، در حالی که مجموعه برچسب Arman شامل شخص، سازمان، مکان، تأسیسات، محصول و رویداد می‌باشد.

در مدل دیگر یک میدان تصادفی شرطی را بر روی یک ترنسفورمر دوطرفه پیش‌آموزش دیده BERT آموزش داده اند. BERT را به عنوان یک مدل ترنسفورمر (یک نوع معماری شبکه عصبی است) دوطرفه پیش‌آموزش دیده برای وظایف درک زبان معرفی کردند.BERT در بسیاری از وظایف مانند پاسخگویی به سوالات، استنتاج زبانی و شناسایی موجودیت‌های نام‌دار به نتایج پیشرفته‌تری دست یافته است. از طرفی نیاز به داده‌های برچسب‌گذاری شده بزرگ، مشکل اصلی روش‌های نظارت شده اخیر مانند یادگیری عمیق است. یادگیری انتقال (Transfer Learning) یک استراتژی مؤثر در یادگیری ماشین و پردازش زبان طبیعی است که به‌ویژه می‌تواند در زبان‌های کم‌منبع به حل چالش‌ها کمک کند. یادگیری انتقال از به‌کارگیری یک مدل که قبلاً بر روی یک وظیفه خاص آموزش دیده است (معمولاً با داده‌های زیاد) و استفاده از آن برای یک وظیفه دیگر (معمولاً با داده‌های کمتر) استفاده می کند. این مدل‌ها معمولاً ویژگی‌های عمیق‌تری از داده‌ها استخراج می‌کنند و در وظایف خاص عملکرد بهتری دارند. یادگیری انتقال با از تعبیه کلمات (Word Embeddings) که یک آموزش غیرنظارتی است به فرآیند تبدیل کلمات به وکتورهای عددی اشاره دارد که معانی و روابط معنایی بین کلمات را در یک فضای چندبعدی نمایش می‌دهند. مدل‌هایی مانند Word2Vecو GloVe از این روش‌ها برای یادگیری روابط بین کلمات استفاده می‌کنند. پس از آموزش تعبیه کلمات، این وکتورها به عنوان ورودی برای مدل‌های یادگیری ماشین یا یادگیری عمیق (مانند LSTM یا BERT) در وظایف نظارت شده مانند شناسایی موجودیت‌های نامدار یا تحلیل احساسات به کار می‌روند. به این ترتیب، آن‌ها می‌توانند نیاز به داده‌های برچسب‌گذاری شده بزرگ را کاهش دهند. BERT بر روی ۱۰۴ زبان، از جمله فارسی، پیش‌آموزش دیده است و این یکی از مزایای بزرگ این مدل است. مدل CRF به عنوان یک مدل احتمالی، مانند مدل مارکوف پنهان(HMM)[۲]، امکان استخراج و در نظر گرفتن وابستگی‌های ساختاری بین برچسب‌ها در داده‌ها را فراهم می‌کند. در حالی که کدگذارهایی مانند BERT سعی در حداکثرسازی احتمال با انتخاب بهترین نمایش پنهان دارند، CRF با انتخاب بهترین برچسب‌های خروجی، احتمال را حداکثر می‌کند.

مدل‌های یادگیری عمیق به‌ویژه در پردازش زبان طبیعی (NLP) به سرعت در حال تحول هستند و یکی از پیشرفت‌های چشمگیر در این حوزه، استفاده از شبکه‌های عصبی بازگشتی (RNN) و به‌ویژه مدل BiLSTM است. شبکه های عصبی بازگشتی به دلیل توانایی‌شان در پردازش توالی‌های داده‌ای، به‌ویژه متن، بسیار مورد توجه قرار گرفته‌اند. با این حال، RNN های استاندارد گاهی با مشکلاتی مانند فراموشی اطلاعات در توالی‌های طولانی مواجه می‌شوند. مدل BiLSTM با گسترش معماریLSTM ، به این چالش پاسخ می‌دهد. این مدل به‌طور همزمان از دو جهت (به جلو و به عقب) به توالی‌ها نگاه می‌کند، که این امکان را فراهم می‌آورد تا اطلاعات بیشتری از بافت متن استخراج شود. به‌عنوان مثال، در پردازش جملات، دانستن کلمات قبل و بعد از یک کلمه خاص می‌تواند به درک بهتر معنا و نقش آن کلمه کمک کند. استفاده از BiLSTM در وظایف مختلفNLP، از جمله شناسایی موجودیت‌های نام‌دار، تحلیل احساسات و ترجمه ماشین، نتایج قابل توجهی به‌دنبال داشته است. این مقاله به بررسی ساختار و عملکرد مدلBiLSTM، مزایای آن نسبت به مدل‌های قبلی، و کاربردهای آن در پردازش زبان طبیعی می‌پردازد.

فیلترهای محلی یکی از اجزای کلیدی در شبکه‌های عصبی کانولوشنی (CNN) هستند که به‌طور مؤثری در پردازش داده‌های تصویری و متنی به کار می‌روند. این فیلترها به‌ویژه برای استخراج ویژگی‌های محلی از ورودی‌ها طراحی شده‌اند و به مدل کمک می‌کنند تا الگوهای پیچیده را شناسایی کند. در واقع، فیلترهای محلی با تمرکز بر روی نواحی کوچک از داده، قادر به شناسایی ویژگی‌های بنیادی مانند لبه‌ها، بافت‌ها و اشکال می‌باشند. استفاده از فیلترهای محلی به مدل این امکان را می‌دهد که از اطلاعات بافتی و ساختاری در داده‌ها بهره‌برداری کند. به‌عنوان مثال، در تصاویر، فیلترهای محلی می‌توانند لبه‌ها و نقاط عطف را شناسایی کنند، در حالی که در متن، این فیلترها می‌توانند ارتباطات معنایی بین کلمات را استخراج کنند. این ویژگی‌ها باعث می‌شود که فیلترهای محلی در بسیاری از کاربردها از جمله شناسایی اشیاء، تشخیص چهره، و پردازش زبان طبیعی بسیار مؤثر باشند.

---

[۲] مدل پنهان مارکوف به انگلیسی (Hidden Markov Model): یک مدل مارکوف آماری است که در آن سیستم مدل شده به صورت یک فرایند مارکوف با حالت‌های مشاهده نشده (پنهان) فرض می‌شود. یک مدل پنهان مارکوف می‌تواند به عنوان ساده‌ترین شبکه بیزی پویا در نظر گرفته شود.

چالش های زبان فارسی

پردازش زبان طبیعی (NLP) به خصوص در زمینه پردازش موجودیت های نامدار (NER) چالش هایی وجود دارد که به برخی از آنها اشاره می کنیم. ۱- عدم وجود حروف بزرگ و کوچک در زبان فارسی؛ ۲- رسم الخط زبان فارسی که توسط فرهنگستان ادب و زبان فارسی مورد تایید است [۱]؛ ۳- کمی صداها هنگام استفاده از بسیاری از الگوریتم ها در استخراج ویژگی ها؛ ۴- عدم وجود یک مجموعه قوی جهت آموزش سیستم که در مقایسه با زبان‌های دیگر، مجموعه‌های داده بزرگ و با برچسب برای زبان فارسی محدود هستند، که این موضوع بر آموزش مدل‌ها تأثیر می‌گذارد؛ ۵- پسوندها و پیشوندهای جدا[۱[که این ویژگی‌ها در تشخیص صحیح مابین اسم‌ها را دشوار می‌سازد. ۶- آزادی زیاد در ترتیب کلمات: در زبان فارسی، ترتیب کلمات می‌تواند به‌طور قابل توجهی متفاوت باشد و معانی یکسانی را منتقل کند. ۷-کمبود داده‌های آموزشی نشانه‌گذاری شده در فارسی: این موضوع محدودیت‌هایی را در توسعه مدل‌های یادگیری ماشین ایجاد می‌کند و ابزارها و منابع NLP برای زبان فارسی به اندازه زبان‌های دیگر توسعه نیافته‌اند، که این موضوع بر پژوهش‌ها و پروژه‌ها تأثیر می‌گذارد. ۸- تعدد گویش‌ها و لهجه‌ها: زبان فارسی به‌طور رسمی دارای چندین گویش و لهجه است که می‌تواند باعث تفاوت‌های معنایی و نحوی شود. ۹- پیچیدگی ساختاری: زبان فارسی دارای ساختارهای نحوی پیچیده‌ای است که شامل ترکیبات، جملات شرطی و جملات مرکب می‌شود. ۱۰- عدم استانداردسازی املایی: وجود املای مختلف برای کلمات مشابه و استفاده از حروف و نشانه‌های مختلف، باعث ابهام در پردازش متن می‌شود. ۱۱- معنی‌گذاری بافتی: کلمات در زبان فارسی به‌ویژه در جملات طولانی می‌توانند معانی متفاوتی داشته باشند، که نیاز به تحلیل بافتی دارد. ۱۲- حروف اضافه: استفاده از حروف اضافه در زبان فارسی می‌تواند ساختار جملات را پیچیده کند و تحلیل نحوی را دشوار نماید. ۱۳- چالش‌های مربوط به ترجمه ماشینیکه از زبان فارسی به زبان‌های دیگر و بالعکس به دلیل ساختار خاص و تفاوت‌های فرهنگی و معنایی می‌تواند مشکل‌ساز باشد. ۱۴- تحلیل احساسات و عواطف در متن‌های فارسی به دلیل زبان محاوره‌ای و استفاده از اصطلاحات غیررسمی چالش‌برانگیز است. ۱۵- موجودیت‌های چندمعنایی: یک نام ممکن است به چند موجودیت متفاوت اشاره داشته باشد مثلاً "کابل" می‌تواند هم به پایتخت افغانستان و هم به یک نوع کابل الکتریکی اشاره کند.

پارامترهای ارزیابی

روش های مدل از سه پارامتر ارزیابی استفاده می‌کند: دقت(Precision)، یادآوری (Recall) و نمره(F-measure) F .

دقت (Precision) : این پارامتر نشان می‌دهد که روش ما چقدر دقیق است، به این معنا که چند درصد از موجودیت‌های نامدار پیش‌بینی‌شده صحیح هستند. به عبارت دیگر، دقت نشان می‌دهد که از کل مواردی که مدل به‌عنوان مثبت شناسایی کرده، چند مورد واقعاً مثبت هستند. فرمول (۱) مربوط به Precision است.

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)+False Positive (FP)}}$$ **فرمول (۱)**

یادآوری (Recall): این پارامتر تعداد موجودیت‌های نامدار واقعی را که مدل ما در فرآیند برچسب‌گذاری شناسایی کرده است، محاسبه می‌کند. فراخوانی به نسبت نمونه‌های درست مثبت به کل نمونه‌های واقعی مثبت اشاره دارد. این معیار نشان می‌دهد که مدل چقدر از موارد مثبت واقعی را شناسایی کرده است. فرمول (۲) مربوط به Recall است.

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)+False Negatives (FN)}}$$ **فرمول (۲)**

نمره F (F-measure): این پارامتر تعادل و میانگین بین دقت و یادآوری را بررسی می‌کند. فرمول (۳) مربوط به Recall است.

$$\mathbf{F1} = 2 * \frac{\mathbf{Precision} * \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}}$$ **فرمول (۳)**

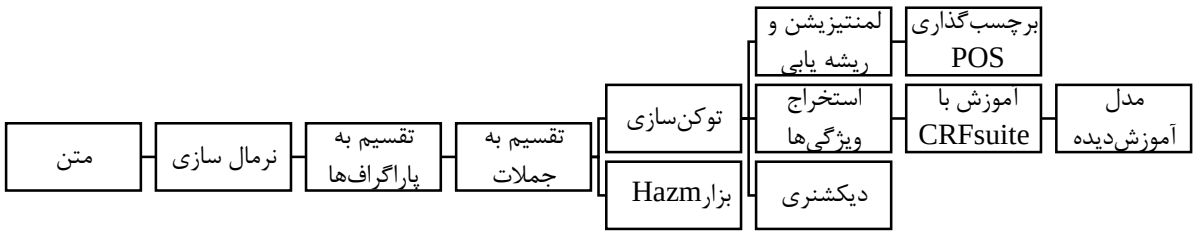
مدل میدان تصادفی شرطی (CRF)

مدل میدان تصادفی شرطی (CRF) دارای الگوریتم خاصی است که برای برچسب‌گذاری توالی‌ها و شناسایی الگوها در داده‌های دنباله‌ای مورد استفاده قرار می‌گیرد. CRF به عنوان یک مدل یادگیری نظارت‌شده، از اطلاعات متنی و ویژگی‌های استخراج‌شده برای پیش‌بینی برچسب‌های مربوط به هر عنصر در دنباله استفاده می‌کند. الگوریتم های مورد استفاده در میدان تصادفی شرطی(CRF): ۱- الگوریتم حداکثر احتمال :(Maximum Likelihood Estimation) این الگوریتم برای آموزش مدل استفاده می‌شود تا پارامترهای بهینه را برای بیشینه کردن احتمال مشاهده داده‌های آموزشی پیدا کند. ۲- استخراج ویژگی CRF: اجازه می‌دهد تا از ویژگی‌های متنوع (زبان‌شناختی و غیرزبان‌شناختی) برای مدل‌سازی استفاده شود، که در این مقاله شامل ویژگی‌های خاص زبان فارسی از جمله وابستگی گرامری است. ۳- برچسب‌گذاری دنباله‌ای CRF: به طور خاص برای مسائل برچسب‌گذاری دنباله‌ای طراحی شده است، که به شناسایی موجودیت‌های نامدار (NER) کمک می‌کند.

ما یک سیستم شناسایی موجودیت‌های نامدار (NER) مبتنی بر میدان تصادفی شرطی (CRF) پیشنهاد کردیم که موجودیت‌های نامدار را با استفاده از ویژگی‌های نحوی متعددی بر اساس گرامر وابستگی و همچنین برخی ویژگی‌های فارسی مانند ویژگی‌های مبتنی بر صرف مانند پیشوندها و پسوندها، ویژگی‌های مبتنی بر فهرست‌های طراحی شده، و ویژگی‌های نحوی بهره بردیم. چارچوب روش ما در شکل (۱) نشان داده شده است. این شکل مراحل مختلف سیستم ما را شامل پیش‌پردازش، استخراج ویژگی‌ها و یادگیری ماشین به تصویر می‌کشد. فرایند NER با نرمال‌سازی متن با استفاده از ابزار نرمال‌سازی Hazm [۳] آغاز می‌شود.

مراحل کلیدی شامل:

۱. پیش‌پردازش: شامل توکن‌سازی و نرمال‌سازی متن برای حذف نویز و اصلاح املایی.
۲. استخراج ویژگی‌ها: ویژگی‌های نحوی و صرفی از متن برای شناسایی موجودیت‌های نامدار استخراج می‌شوند.
۳. یادگیری ماشین: استفاده از الگوریتم CRF برای شناسایی موجودیت‌های نامدار بر اساس ویژگی‌های استخراج‌شده.



شکل(۱)

در شکل ۱. متن: ورودی اولیه که پردازش می‌شود. ۲. نرمال‌سازی: مرحله‌ای برای استانداردسازی متن. ۳. تقسیم به پاراگراف‌ها: متن به پاراگراف‌های جداگانه تقسیم می‌شود. ۴. تقسیم به جملات: پاراگراف‌ها به جملات تقسیم می‌شوند. ۵. ابزار Hazm : ابزاری برای پردازش زبان فارسی. ۶. توکن‌سازی: جملات به توکن‌ها (واحدهای معنایی) تقسیم می‌شوند. ۷. لمنتیزیش یا ریشه یابی: تبدیل کلمات به شکل اولیه خود. ۸. برچسب‌گذاری POS : تعیین بخش‌های سخن (مثل اسم، فعل، و غیره) برای هر توکن. ۹.استخراج ویژگی‌ها: استخراج ویژگی‌های لازم برای مدل. ۱۰. دیکشنری: استفاده از لیست‌های موجودیت‌های نامدار. ۱۱. آموزش باCRFsuite : مرحله‌ای برای آموزش مدل با استفاده از روش CRF. ۱۲. مدل آموزش‌دیده: خروجی نهایی که مدل آموزش‌دیده را نشان می‌دهد.

[۳] کتابخانه پایتونی برای پردازش زبان فارسی

در مرحله دوم، متن به‌طور جداگانه به پاراگراف‌ها و جملات تقسیم می‌شود. سپس جملات توکن‌سازی می‌شوند، به‌طوری‌که کلمات و نشانه‌گذاری‌ها (مانند نقطه ویرگول و نقطه کامل) از هم جدا می‌شوند]۲[. در مرحله بعد، برچسب بخش‌های گفتار (POS) برای هر کلمه مشخص می‌شود]۲[. پس از آن، ویژگی‌های طراحی‌شده برای هر کلمه با کمک لِماتایزر و فهرست‌های طراحی شده که در این رویکرد طراحی شده‌اند، استخراج می‌شوند. در نهایت، در مرحله یادگیری، این ویژگی‌ها برای آموزش CRFsuite استفاده می‌شوند که یک پیاده‌سازی از روش میدان تصادفی شرطی است]۲[. در مرحله آزمون، مدل آموزش‌دیده برای شناسایی موجودیت‌های نامدار استفاده می‌شود.

CRFsuite

اصلی‌ترین عیب CRF ناشی از محاسبات پیچیده آن در مرحله آموزش است]۲[. بنابراین، پس از افزودن نمونه‌های جدید داده، دشوار است که مدل دوباره آموزش داده شود. برای غلبه بر این مشکل، از پیاده‌سازی CRFsuite استفاده شده است. CRFsuite به‌عنوان یک پیاده‌سازی از CRF در میان پیاده‌سازی‌های مختلف، برای برچسب‌گذاری داده‌های توالی در رویکرد ما استفاده شده است. این ابزار نه تنها آموزش سریعی را فراهم می‌کند، بلکه یک فرمت ساده برای داده‌های آموزشی و برچسب‌گذاری مشابه سایر ابزارهای یادگیری ماشین ارائه می‌دهد. علاوه بر این، CRFsuite خروجی‌هایی مانند دقت (precision) ، یادآوری (recall) و نمره (F1) مدل ارزیابی شده را فراهم می‌کند. این ویژگی‌ها به ما کمک می‌کند تا عملکرد مدل را به‌طور دقیق ارزیابی کنیم و در صورت نیاز به بهبود، تغییرات لازم را اعمال کنیم. این ابزار به‌ویژه برای پروژه‌های شناسایی موجودیت‌های نامدار مناسب است و به ما این امکان را می‌دهد تا مدل‌های پیچیده‌تری را با کارایی بالا پیاده‌سازی کنیم. از مزیت های CRFsuite کارایی بالا و بهینه‌سازی شده است تا زمان آموزش و پیش‌بینی را کاهش دهد و به همین دلیل برای کار با مجموعه‌های داده بزرگ بسیار مناسب است. این ابزار دارای رابط کاربری ساده‌ای است که به کاربران اجازه می‌دهد به راحتی مدل‌های CRF را آموزش دهند و ارزیابی کنند. CRFsuit به کاربران این امکان را می‌دهد که از انواع مختلف ویژگی‌ها (زبان‌شناختی و غیرزبان‌شناختی) استفاده کنند و ویژگی‌های جدید را به سادگی اضافه کنند. این ابزار به طور خودکار دقت، فراخوان و اندازه‌گیری F را محاسبه می‌کند که برای ارزیابی مدل‌های یادگیری ماشین بسیار مفید است. از الگوریتم‌های مختلفی برای آموزش مدل‌ها پشتیبانی می‌کند، که به کاربران این امکان را می‌دهد تا بهترین گزینه را برای نیازهای خاص خود انتخاب کنند.

مجموعه داده

مجموعه داده مورد استفاده برای ارزیابی مدل مبتنی بر میدان تصادفی شرطی (CRF) بانک درخت وان وابستگی نحوی فارسی (Persian syntactic dependency Treebank) استفاده شده است که شامل حدود ۳۰،۰۰۰ جمله می‌باشد. و برچسب‌های موجودیت نامدار مانند شخص، سازمان و مکان است. این مجموعه داده به صورت دستی با برچسب‌های موجودیت نام‌دار (NER) و اطلاعات دیگر مانند وابستگی و بخش‌های گفتار (POS) علامت‌گذاری شده است. همچنین، در پروژه‌ای به نام بانک گزاره فارسی (Persian Proposition Bank) اطلاعات اضافی مانند روابط گزاره‌ای و اطلاعات هم‌ارجاعی نیز به این درختواره اضافه شده است.

استخراج ویژگی‌ها

در این رویکرد، علاوه بر ویژگی‌های مستقل از زبان، ویژگی‌های خاص زبان فارسی مانند ویژگی‌های نحوی استخراج‌شده از گرامر وابستگی شامل اطلاعات مربوط به ساختار جملات و نوع کلمات (فعل، اسم، صفت و ...) و ویژگی‌های مورفولوژیکی شامل ریشه کلمات، پسوندها و پیشوندها که به شناسایی بهتر موجودیت‌ها کمک می‌کند و ویژگی‌های متنی مانند موقعیت کلمه در جمله (ابتدا، وسط، انتها) که می‌تواند به شناسایی نوع موجودیت کمک کند و طول کلمات نیز به عنوان یک ویژگی در نظر گرفته شده است و ویژگی‌هایی که اطلاعاتی درباره کلمات همسایه (قبل و بعد از کلمه هدف) ارائه می‌دهند برای شناسایی موجودیت‌های

نامدار در متن استفاده شده است. به طور خلاصه، ما از ویژگی‌های مبتنی بر صرف مانند پیشوندها و پسوندها، ویژگی‌های مبتنی بر فهرست‌های طراحی شده، و ویژگی‌های نحوی بهره بردیم.
برخی از این ویژگی ها را می توانیم با استفاده الگوریتم های زیر استخراج کنیم: ۱- تحلیل واریانس(ANOVA)، ۲- تجزیه و تحلیل همبستگی، ۳- متدهای انتخاب ویژگی مبتنی بر درخت تصمیم: روش‌هایی مانند Random Forest یا CART به طور طبیعی اهمیت ویژگی‌ها را محاسبه می‌کنند و می‌توانند به عنوان روشی مکمل برای انتخاب ویژگی‌ها استفاده شوند. ۴- الگوریتم‌های یادگیری ماشین:استفاده از الگوریتم‌های یادگیری ماشین مانند SVM یا Gradient Boosting می‌تواند به شناسایی ویژگی‌های مؤثر کمک کند. ۵- روش‌های مبتنی بر یادگیری عمیق: در برخی از موارد، استفاده از شبکه‌های عصبی برای استخراج ویژگی‌ها و یادگیری الگوها نیز ممکن است به کار رود. ۶- Gain Information (IG)

انتخاب ویژگی‌ها

در میان بسیاری از ویژگی‌های اضافی یا نامربوط در پردازش زبان طبیعی(NLP) ، انتخاب ویژگی‌های مناسب یک فرآیند دشوار و زمان‌بر است، به‌ویژه زمانی که نمی‌توانیم رفتار داده‌ها را پیش‌بینی کنیم. بنابراین، استفاده از یک پارامتر برای انتخاب ویژگی‌ها این مسأله را ساده‌تر می‌کند. در اینجا، الگوریتم CART (Classification and Regression Trees) از چندین فرمول و معیار برای تقسیم‌سازی داده‌ها استفاده می‌کند. این معیارها به تعیین اینکه کدام ویژگی بهترین تقسیم را ایجاد می‌کند، کمک می‌کنند. ۱- Gini Impurity: سنجش میزان اختلاط یا عدم خلوص یک مجموعه داده است. فرمول (۴) مربوط به سنجش میزان اختلاط است. ۲- آنتروپی Entropy : به عنوان یک معیار برای تقسیم‌سازی استفاده می کند. فرمول (۵) مربوط به آنتروپی است. ۳- Mean Squared Error (MSE): برای رگرسیون، CART از Mean Squared Error استفاده می‌کند. فرمول MSE به فرمول (۶) آمده است. ۴- محاسبه کاهش معیار: CART برای هر تقسیم، کاهش معیار (مثلاً Gini ی Entropy) را محاسبه می‌کند. فرمول (۷) بیانگر این ویژگی است.

$$\mathbf{Gini(D)} = ۱ - \sum_{i=۱}^{c} Pi^{\wedge}۲$$

فرمول (۴): D مجموعه داده‌ها است. C تعداد کلاس‌ها است. Pi نسبت نمونه‌های کلاس i در مجموعه داده D است.

$$\text{Entropy(D)} = -\sum_{i=۱}^{c} \text{Pi}\log2(\text{Pi})$$

فرمول (۵): در اینجا N تعداد نمونه‌ها است yj مقدار واقعی برای نمونه j است $\bar{y}$میانگین مقادیر واقعی در مجموعه داده.

$$\text{MSD(D)} = \frac{۱}{\text{N}} \sum_{\text{i}=۱}^{\text{N}} (\text{Yj} - \overline{\text{Y}})^{\wedge}۲$$

فرمول (۶)

$$\text{Reduction(D)} = \text{Impurity(parent)} - (\frac{\mathbf{Nl}}{\mathbf{N}} \times \text{Impurity(left child)} + \frac{\text{N}r}{\text{N}} \times \text{Impurity(right child)})$$

فرمول(۷): در اینجا NL و NR تعداد نمونه‌ها در گره‌های فرزند چپ و راست هستندو N تعداد کل نمونه‌ها در گره والد است.

CART با استفاده از این معیارها و فرمول‌ها، بهترین ویژگی و نقطه تقسیم را برای ایجاد درخت تصمیم انتخاب می‌کند. این فرآیند به طور تکراری انجام می‌شود تا درخت کامل شود.

تست و ارزیابی با CART

این مرحله حیاتی در فرآیند توسعه مدل است، زیرا به ارزیابی واقعی عملکرد مدل و شناسایی نقاط بهبود کمک می‌کند. با استفاده از این معیارها و تحلیل‌ها، می‌توان بهینه‌سازی‌های لازم را انجام داده و دقت مدل را افزایش داد. در مرحله تست و ارزیابی مدلCART برای شناسایی موجودیت‌های نامدار، مجموعه داده را به دو بخش اصلی تقسیم می‌کند: مجموعه آموزشکه برای آموزش مدل استفاده می‌شود و مجموعه تست که برای ارزیابی عملکرد مدل استفاده می‌شود. نتایج به‌دست‌آمده از معیارهای ارزیابی باید تحلیل شوند. این تحلیل می‌تواند شامل موارد زیر باشد :شناسایی نقاط قوت و ضعف مدل، بررسی اینکه آیا مدل به

خوبی موجودیت‌های نامدار مختلف (مانند نام‌ها، مکان‌ها و سازمان‌ها) را شناسایی کرده است یا خیر و شناسایی کلاس‌هایی که مدل در تشخیص آن‌ها دچار مشکل است. تجزیه و تحلیل خطا می‌تواند به شناسایی الگوهای خاصی که مدل قادر به شناسایی آن‌ها نیست، کمک کند و در آخر بهینه‌سازی و تکرار می‌تواند ویژگی‌های جدیدی را اضافه کرد یا ویژگی‌های کم‌اهمیت را حذف کرد. همچنین ممکن است نیاز به تغییر در پارامترهای مدل یا استفاده از تکنیک‌های پیشرفته‌تر باشد.

مدل BERT

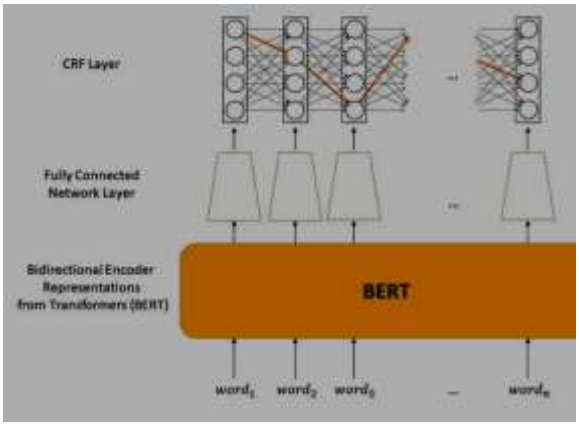
BERT(Bidirectional Encoder Representations from Transformers) یکی از مدل‌های پیشرفته یادگیری عمیق است که توسط گوگل توسعه یافته است. این مدل برای درک زبان طبیعی طراحی شده و در طور خاص برای وظایف پردازش زبان طبیعی (NLP) به کار می‌رود. از ویژگی‌های اصلی BERT می توان به دوطرفه بودن مدل اشاره کرد برخلاف مدل‌های قبلی که تنها از اطلاعات یک‌طرفه (چپ به راست یا راست به چپ) استفاده می‌کردند، BERT یک مدل زبان عمیق دو طرفه است بدین معنی که به صورت همزمان از دو سمت (چپ و راست) به متن نگاه می‌کند. این ویژگی به مدل کمک می‌کند تا زمینه بیشتری را درک کند. پیش‌آموزش و تنظیم دقیق ویژگی دیگر مدل BERT است که ابتدا بر روی مقدار زیادی از داده‌های متنی عمومی پیش‌آموزش می‌شود و سپس می‌تواند برای وظایف خاص NLP مانند طبقه‌بندی متن، پاسخ‌دهی به سوالات و...) تنظیم دقیق شود.

BERT از مکانیزم خاصی مبتنی بر پایه معماری Transformer ساخته شده است که از مکانیزم توجه (Attention) برای پردازش تسریع‌شده استفاده می‌کند. این معماری به مدل اجازه می‌دهد تا وابستگی‌های طولانی‌مدت را در متن بهتر درک کند. نسخه‌ای از BERT وجود دارد که می‌تواند به چند زبان مختلف کار کند، که این امر استفاده از این مدل را در کاربردهای جهانی افزایش می‌دهد. علاوه بر تشخیص موجودیت های نام دار توسط مدل BERT از کاربردهای آن می توان به پاسخ‌دهی به سوالات که می‌تواند به سوالات خاصی پاسخ دهد و اطلاعات مربوطه را از متن استخراج کند. تحلیل احساسات، ترجمه ماشین که به بهبود کیفیت ترجمه‌های ماشینی کمک می کند، خلاصه‌سازی متن، طبقه‌بندی جفت جملات (برچسب‌هایی که نشان می‌دهند آیا جملات با هم مرتبط هستند یا خیر) ، طبقه‌بندی جمله واحد (برچسب‌هایی که نشان می‌دهند جمله در کدام دسته قرار می‌گیرد (مثلاً مثبت یا منفی) و وظیفه برچسب‌گذاری جمله واحد اشاره کرد. همانطور که اشاره شد. BERT به صورت دوطرفه آموزش دیده است، این ویژگی به درک بهتر زمینه کلمات کمک می‌کند. به‌عنوان یک مدل دوطرفه، بافت هر کلمه را با توجه به کلمات قبل و بعد آن تحلیل می‌کند. این ویژگی به‌ویژه در زبان فارسی که ساختار جملات می‌تواند پیچیده باشد، کمک می‌کند تا مدل به درک بهتری از معنای کلمات و ارتباطات آن‌ها برسد. BERT از روش توکن‌سازی خاصی به نام Byte-Pair Encoding استفاده می‌کند که به آن امکان می‌دهد توکن‌ها را به صورت بهینه‌تری تقسیم کند و با کلمات ناآشنا یا ترکیبی بهتر عمل کند. BERT برای زبان فارسی، آموزش دیده است. با استفاده از یادگیری انتقالی، BERT می‌تواند با داده‌های کمتری آموزش ببیند و همچنان عملکرد خوبی داشته باشد. این موضوع به ویژه در زبان‌های کم‌منبع اهمیت دارد. یادگیری انتقالی (Transfer Learning) یک رویکرد در یادگیری ماشین است که در آن یک مدل که قبلاً بر روی یک وظیفه خاص آموزش دیده است، برای یک وظیفه جدید استفاده می‌شود. این روش به ویژه در شرایطی که داده‌های برچسب‌گذاری شده محدود هستند، مفید است. BERT انقلابی در پردازش زبان طبیعی ایجاد کرده و امکان ایجاد مدل‌هایی با دقت بالا و توانایی‌های گسترده را فراهم کرده است. این مدل به عنوان یکی از ابزارهای کلیدی در تحقیقات و کاربردهای NLP شناخته می‌شود.

روش کار در BERT

در این روش که از مدل پیش‌آموزش‌دیده BERT استفاده می‌کنیم که ما نیاز داریم که مناسب‌ترین برچسب را به هر توکن اختصاص دهیم و توکن‌سازی مناسب یک مرحله مهم است. در حالی که BERT توکن‌سازی خود را با استفاده از کدگذاری

Byte-Pair[۴] انجام می‌دهد و برچسب‌ها را به توکن‌های استخراج‌شده خود اختصاص می‌دهد، ما باید به این موضوع توجه کنیم. توکن‌های استخراج‌شده توسط BERT همیشه برابر یا کوچکتر از توکن‌های اصلی ما هستند، زیرا BERT توکن‌های مجموعه داده ما را یکی یکی می‌گیرد. در نتیجه، ما توکن‌های درون‌توکنی خواهیم داشت که برچسب i را دریافت می‌کنند. ما یک میدان تصادفی شرطی (CRF) و یک لایه کاملاً متصل را پس از نمای خروجی توکن‌های استخراج‌شده توسط BERT آموزش دادیم. این یک مرحله تنظیم دقیق است که مدل را برای وظیفه NER آماده می‌کند. شما می‌توانید یک طرح از مدل را در شکل (۲) مشاهده کنید.



شکل (۲) [۳]

BERT: این بخش نماینده‌ای از کلمات ورودی را تولید می‌کند. از تکنیک‌های پیشرفته برای درک متن و رابطه بین کلمات استفاده می‌کند Fully Connected Network Layer: پس از تولید نمایه‌های کلمات توسط BERT ، این لایه به پردازش و تحلیل این نمایه‌ها می‌پردازد و به هر توکن برچسب مناسب را اختصاص می‌دهد. لایه CRF: لایه میدان تصادفی شرطی (CRF) برای بهبود دقت برچسب‌گذاری توکن‌ها و استفاده از وابستگی‌های بین توکن‌ها در یک جمله به کار می‌رود. این لایه به تصمیم‌گیری نهایی درباره برچسب‌های توکن‌ها کمک می‌کند.

به طور کلی BERT شامل مراحل زیر است: ۱- آموزش پیشین مدل(Pre-training) : مدل BERT ابتدا بر روی مجموعه‌ای بزرگ از متن‌های فارسی آموزش می‌بیند. این مرحله شامل یادگیری ویژگی‌های بنیادی زبان و درک زمینه‌های مختلف است. ۲- تنظیم دقیق: پس از آموزش پیشین، مدل برای وظیفه خاص شناسایی موجودیت‌های نام‌دار تنظیم می‌شود. این مرحله شامل اضافه کردن لایه‌های خاص به مدل برای شناسایی موجودیت‌ها و آموزش آن با استفاده از داده‌های برچسب‌گذاری‌شده است که این لایه می تواند از CRF استفاده کند. ۳- استفاده از دیکشنری‌ها: در این روش، ممکن است از دیکشنری‌های موجود برای شناسایی و اعتبارسنجی موجودیت‌ها استفاده شود. دیکشنری‌ها به عنوان منبعی برای استخراج و شناسایی موجودیت‌ها عمل می‌کنند. ۴- فیلتر کردن مثبت‌های کاذب: پس از شناسایی موجودیت‌ها، مدل ممکن است نیاز به فیلتر کردن مثبت‌های کاذب داشته باشد. این مرحله به منظور افزایش دقت و کاهش خطاهای شناسایی انجام می‌شود. ۵- معیارهای ارزیابی: عملکرد مدل با استفاده از معیارهایی مانند دقت(Precision) ، بازیابی (Recall) و نمره F1 (F1 Score) ارزیابی می‌شود. این معیارها به تعیین کیفیت شناسایی موجودیت‌ها کمک می‌کنند.

مجموعه داده مورد استفاده در BERT

[۴] Byte-Pair Encoding (BPE) یک الگوریتم فشرده‌سازی است که با جایگزینی زوج‌های متوالی از بایت‌ها با یک بایت جدید، به کاهش حجم داده‌ها کمک می‌کند.

در روش مدل BERT از دو مجموعه داده متفاوت، Peyma و Arman استفاده کرده ایم و آنها را آموزش داده و آزمایش کردیم. مجموعه داده PEYMA را به ۵ زیرمجموعه مساوی تقسیم کردیم. Peyma شامل ۷۱۴۵ جمله است که هر زیرمجموعه شامل ۱۴۲۹ جمله و ۳۰۲،۵۳۰ توکن است و شامل انواع موجودیت‌ها مانند شخص، سازمان، مکان، تاریخ، پول، درصد و زمان. مجموعه دادهArman : این مجموعه شامل ۲۵۰،۰۱۵ توکن و ۷،۶۸۲ جمله است و شامل کلاس‌های مختلف موجودیت مانند شخص، سازمان، مکان، تسهیلات، محصول و رویداد و به عنوان یک مرجع دیگر برای ارزیابی عملکرد مدل به کار رفته است.
در مجموعه داده Peyma ، ما به نمره ۹۰،۵۹ F1 CONLL % در سطح عبارت و ۸۷،۶۲ F1 % در سطح واژه دست یافتیم. بهترین نتایج برای گروه شخص و بدترین نتایج برای گروه زمان مشاهده می‌شود. در مجموعه دادهArman ، ما به نمره ۷۹،۹۳ % F1 CONLL در سطح عبارت و ۸۴،۰۳ F1 % در سطح واژه رسیدیم. بهترین نتایج برای گروه شخص و بدترین برای گروه رویداد مشاهده می‌شود. یکی از دلایل دستیابی به نتایج متفاوت در هر کلاس، مقدار موجودیت‌های نام‌دار در مجموعه داده‌ها است]۳[.

روش BiLSTM-CRF

BiLSTM-CRF (Bidirectional Long Short-Term Memory with Conditional Random Fields) یک شبکه عصبی بازگشتی است که از ترکیب حافظه بلند-کوتاه‌مدت (LSTM) و میدان تصادفی شرطی (CRF) به‌دست می‌آید. ابتدا از LSTMبرای پردازش هر جمله توکن به توکن استفاده می‌شود تا یک نمایه میانی تولید کند. سپس، این نمایه میانی به‌عنوان ورودی برای CRF به‌کار می‌رود تا پیش‌بینی برچسب‌های همه توکن‌ها را فراهم کند. این دو مدل از ویژگی‌های مکمل هم هستند LSTM به‌عنوان یک مدل پیچیده و غیرخطی قادر است به‌طور مؤثر روابط توالی میان توکن‌های ورودی را شناسایی کند؛ در عوض، CRF اجازه می‌دهد تا پیش‌بینی مشترک و بهینه‌ای از همه برچسب‌ها در یک جمله انجام شود و روابط در سطح برچسب را ضبط کند. LSTM هر جمله را به‌صورت دو طرفه پردازش می‌کند تا وابستگی‌های توالی را در هر دو جهت درک کند. قابل ذکر است که BiLSTM-CRF یک روش یادگیری عمیق است.

مجموعه‌داده

این بخش به توصیف دو مجموعه‌داده‌ای می‌پردازد که ما برای NER در زبان فارسی ارائه می‌دهیم. توسعه این سیستم با حمایت از دو مجموعه‌داده انجام شد: الف) یک مجموعه‌داده بزرگ شامل جملات فارسی بدون برچسب، که برای آموزش جاسازی‌های کلمه (Word Embeddings) مورد استفاده قرار می‌گیرد. این داده‌ها به مدل کمک می‌کنند تا روابط معنایی و ساختاری کلمات را در زبان فارسی یاد بگیرد. یک ماتریس هم‌روی یا شبیه یاب که می‌تواند از یک مجموعه‌داده بزرگ و کافی محاسبه شود. که از سه منبع از متن‌های فارسی استفاده شده مجموعه‌داده‌های لایپزیگ که شامل تقریباً ۱ میلیون جمله از وب سایت های خبری و ۳۰۰ هزار جمله از ویکی‌پدیا است ، یک زیرمجموعه از اخبار VOA با ۲۲۷ هزار جمله و درخت‌بانک وابستگی فارسی که شامل تقریباً ۳۰ هزار جمله است]۴[. مجموعه‌داده نامبرده، با مجموع بیش از ۱،۶ میلیون جمله، از یک فرایند نرمال‌سازی متن و توکن‌سازی عبور کرده است. پس از نرمال‌سازی، ما سه ر روش مختلف شناسایی جاسازی کلمه و جداسازی آن را آموزش داده‌ایم. این روش‌ها شامل GloVe ، word2vec و Hellinger PCA (HPCA) هستند. ب) مجموعه‌داده برچسب‌گذاری‌شده موجودیت برای آموزش، که یک مجموعه‌داده که شامل جملات و برچسب‌های مربوط به موجودیت‌های نام‌دار است. این داده‌ها برای آموزش طبقه‌بندهای موجودیت‌های نام‌دار (NER) به کار می‌روند، که به شناسایی و طبقه‌بندی اشخاص، مکان‌ها، سازمان‌ها و دیگر موجودیت‌ها در متن کمک می‌کنند. برای ایجاد یک مجموعه‌داده موجودیت‌های نام‌دار فارسی، ما یک زیرمجموعه از ۷،۶۸۲ جمله خبری را از مجموعه‌داده BijanKhan انتخاب کرده‌ایم که معتبرترین مجموعه‌داده برچسب‌گذاری‌شده POS برای زبان فارسی است. این مجموعه داده پس از برچسب گذاری و ارزیابی یک مجموعه داده جدید را معرفی کرد که مجموعه‌داده برچسب‌گذاری‌شده، معرفی Arman PersoNER Corpus است که اولین مجموعه داده‌ی موجودیت برچسب‌گذاری شده برای زبان فارسی است]۴[. این دیتاست شامل ۷۶۸۲ جمله خبری با ۲۵۰۰۱۵ توکن است و به شناسایی موجودیت‌ها در دسته‌های

مختلف مانند شخص، سازمان، مکان و غیره کمک می‌کند. این دو نوع مجموعه‌داده به طور همزمان می‌توانند به بهبود عملکرد مدل‌های زبان کمک کنند، یکی از طریق یادگیری غیرنظارتی و دیگری از طریق یادگیری نظارتی.

روش‌های به کار گرفته شده برای جداسازی کلمات

شناسایی موجودیت‌های نام‌دار نظارت‌شده به دو مرحله تقسیم می‌شود: مرحله اولیه جداسازی کلمات و سپس مرحله طبقه‌بندی موجودیت‌های نام‌دار در سطح توکن. در مرحله جداسازی، کلمات متمایز را با روش های مختلف جدا کرده و موجویت هر کدام از آنها را تشخیصمی دهیم. GloVe، word2vec و Hellinger PCA (HPCA) نمونه‌های معروفی از جداسازی‌های کلمات غیرنظارت‌شده هستندکه به ما اجازه می‌دهند تا ویژگی‌های معنایی و روابط بین کلمات را در فضایی با ابعاد بالا مدل‌سازی کنیم، که به بهبود دقت سیستم‌های NER کمک می‌کند . این سه روش به عنوان ابزارهای پایه‌ای برای استخراج ویژگی‌ها و نمایش کلمات در مدل‌های یادگیری عمیق به کار می‌روند.

GloVe (Global Vectors for Word Representation) : یک مدل یادگیری غیرنظارت‌شده است که کلمات را به صورت وکتورهایی با ابعاد بالا نمایش می‌دهد]۴[. این مدل یک مدل یادگیری نمایه‌های واژه است که بر اساس ماتریس هم‌همسایگی واژه‌ها کار می‌کند. این مدل سعی می‌کند که نمایه‌های واژه‌ها را به گونه‌ای بیاموزد که روابط معنایی بین واژه‌ها حفظ شود. GloVe به خوبی می‌تواند روابط معنایی و نحوی را مدل کند و برای کاربردهایی مانند تحلیل احساسات و شناسایی موجودیت‌های نام‌دار بسیار مؤثر است. GloVe توسط محققان دانشگاه استنفورد توسعه یافته است. روش کار این مدل به این صورت که از اطلاعات آماری در مورد فراوانی هم‌همسایگی واژه‌ها در یک متن بزرگ استفاده می‌کند. سپس با استفاده از این اطلاعات، یک فضای برداری ایجاد می‌کند که در آن واژه‌های مشابه به یکدیگر نزدیک‌ترند. GloVe از ماتریس هم‌همسایگی واژه‌ها استفاده می‌کند که نشان‌دهنده تعداد بروز هم‌همسایگی واژه‌ها در یک متن بزرگ است. این ماتریس به دو نوع توزیع محاسباتی نیاز دارد :توزیع شرطی: احتمال اینکه واژه j در کنار واژه i ظاهر شود و توزیع جهانی: احتمال اینکه واژه i به طور کلی در متن ظاهر شود. این مدل نمایه‌های واژه‌ها را به گونه‌ای می آموزد که نسبت‌های هم‌همسایگی بین واژه‌ها در فضای برداری حفظ شود. و به خوبی روابط پیچیده معنایی را مدل می‌کند. برای مثال، می‌تواند روابطی مانند «پادشاه» و «ملکه» یا «مرد» و «زن» را در فضای نمایه‌ها حفظ کند و به دلیل استفاده از داده‌های بزرگ، GloVe می‌تواند نمایه‌های دقیقی تولید کند. Word2vec یک مدل یادگیری عمیق است که به طور خاص برای تولید نمایه‌های واژه ای با ابعاد کم طراحی شده است.

Word2vec توسط گوگل ارائه شده و به سرعت به یکی از محبوب‌ترین ابزارها برای تولید نمایه‌های واژه تبدیل شده است و به طور معمول با استفاده از شبکه‌های عصبی ساده پیاده‌سازی می‌شود. این مدل می‌تواند به دو روش مختلف Skip-Gram و Continuous (CBOW) Bag of Words اجرا شود. در واقع یک مدل تولیدی برای نمایش‌های پیوسته کلمات است که ساختارهای خطی میان کلمات را حفظ می‌کند]۴[. این مدل دو نوع مختلف دارد که شامل: مدلskip-gram : این روش به مدل اجازه می‌دهد تا واژه‌های هم‌همسایه را با استفاده از یک واژه مرکزی پیش‌بینی کند. برای مثال، اگر واژه «گربه» در متن ظاهر شود، مدل سعی می‌کند واژه‌های «سیاه»، «خواب»، و «موش» را پیش‌بینی کند. ولی مدل continuous bag of words (CBOW): سعی می‌کند واژه مرکزی را از یک مجموعه از واژه‌های هم‌همسایه پیش‌بینی کند. در این مدل، برعکس Skip-gram، هدف پیش‌بینی کلمه جاری با توجه به کلمات همسایه است. کلمات اطراف به عنوان ورودی داده می‌شوند و مدل باید کلمه مرکزی را پیش‌بینی کند. این مدل‌ها می‌توانند الگوهای معنایی و نحوی را به خوبی یاد بگیرند و برای تشخیص اسامی در متن‌های طبیعی مفید باشند. مدل Word2Vec به دلیل سرعت و دقت بالایش در پردازش زبان طبیعی بسیار محبوب است.

Hellinger PCA یک تکنیک برای کاهش ابعاد داده‌ها است که از معیار فاصله هیلینگر برای اندازه‌گیری شباهت بین توزیع‌های احتمال استفاده می‌کند. روش کار آن به این صورت است که ابتدا توزیع‌های احتمال را برای داده‌ها محاسبه می‌کند و سپس با استفاده از روش‌هایPCA ، ابعاد را کاهش می‌دهد و به دنبال یافتن محورهایی است که بیشترین واریانس را در داده‌ها نشان می‌دهند. این تکنیک به ویژه برای تحلیل داده‌های متنی و مقایسه‌های معنایی بین واژه‌ها مناسب است. HPCA می‌تواند به حفظ اطلاعات مهم در داده‌های متنی کمک کند و به مدل‌های یادگیری ماشین اجازه می‌دهد تا به‌طور مؤثرتری از داده‌ها

استفاده کنند. HPCA می‌تواند الگوهای معنایی را در داده‌ها شناسایی کند و به مدل‌های یادگیری ماشین کمک کند تا از داده‌های بزرگ و پیچیده به بهترین نحو بهره‌برداری کنند.
در نتیجه مدل‌های GloVe و word2vec به صورت گسترده‌ای در کاربردهای پردازش زبان طبیعی برای یادگیری نمایه‌های واژه‌ای استفاده می‌شوند. در حالی که HPCA بیشتر به کاهش ابعاد و تحلیل توزیع‌های احتمال مربوط می‌شود. این تکنیک‌ها به پژوهشگران و توسعه‌دهندگان این امکان را می‌دهند که روابط پیچیده معنایی در داده‌های متنی را به خوبی مدل‌سازی و تحلیل کنند.

روش فیلترهای محلی (Local Filters)

فیلتر محلی یک ابزار کارآمد برای بهبود کیفیت سیستم‌های تشخیص موجودیت نامدار است. با استفاده از این فیلترها، می‌توان به نتایج دقیق‌تری دست یافت و به طور کلی عملکرد سیستم را بهبود بخشید. در این روش از یک مدل فیلتر محلی (Local Filters)برای تشخیص موجودیت‌های نامدار (NER) در زبان فارسی استفاده شده است. این فیلترها به طور خاص در مرحله پس‌پردازش نتایج شناسایی شده به کار می‌روند و هدف آن‌ها حذف نتایج نادرست (False Positives) است.
این رویکرد شامل دو مرحله اصلی است:۱- شناسایی نامزدهای موجودیت نامدار: این مرحله با استفاده از دیکشنری‌های موجود انجام می‌شود که شامل مجموعه‌ای از عبارات توصیف‌کننده موجودیت‌ها هستند. ۲- فیلتر کردن نتایج نادرست: در این مرحله، با استفاده از فیلترهای محلی و فیلترهای مربوط به قسمت‌های کلام (POS) ، نتایج نادرست حذف می‌شوند. در این روش فیلتر محلی از یک مدل به نام ASEM (A Semantic Entity Model) که یک مدل برای شناسایی و طبقه‌بندی موجودیت‌های نام‌دار در متون است استفاده شده است این مدل به منظور بهبود دقت و کارایی در شناسایی موجودیت‌ها طراحی شده است و معمولاً از تکنیک‌های پردازش زبان طبیعی و یادگیری ماشین بهره می‌برد. مدل ASEM می‌تواند شامل استفاده از دیکشنری‌ها، الگوریتم‌های یادگیری عمیق یا روش‌های مبتنی بر قواعد باشد تا موجودیت‌های نام‌دار مانند افراد، مکان‌ها و سازمان‌ها را شناسایی کند. هدف اصلی این مدل‌ها، افزایش دقت در شناسایی موجودیت‌ها و کاهش نرخ مثبت‌های کاذب است.
مراحل عملکرد فیلتر محلی: ۱- شناسایی نامزدها: در ابتدا، سیستم نامزدهای موجودیت نامدار را با استفاده از دیکشنری‌ها شناسایی می‌کند. این نامزدها ممکن است شامل عبارات نادرست یا نامناسب باشند. ۲- اعمال فیلترهای محلی: فیلترهای محلی به بررسی ویژگی‌های نامزدها می‌پردازند. این ویژگی‌ها می‌توانند شامل نوع کلمه (مثل صفت، اسم و ...) و موقعیت آن‌ها در جمله باشند. این فیلترها به سیستم کمک می‌کنند تا نامزدهایی که احتمالاً وجودیت نامدار نیستند را حذف کند. به عنوان مثال، یک کلمه ممکن است به عنوان اسم شناسایی شود، اما در واقع در زمینه جمله به عنوان فعل یا صفت به کار رفته باشد. ۳- استفاده از فیلترهای بخش‌های کلام (POS): فیلترهای محلی معمولاً با فیلترهای مربوط به بخش‌های کلام ترکیب می‌شوند. این فیلترها می‌توانند به شناسایی نامزدهایی که به طور خاص به عنوان اسم (proper noun) یا اسم معمولی (common noun) شناسایی شده‌اند، کمک کنند.
استفاده از فیلتر های محلی باعث افزایش دقت:می شود زیرا با حذف نتایج نادرست، دقت کلی سیستم در شناسایی موجودیت‌ها افزایش می‌یابد. فیلترهای محلی می‌توانند به کاهش تعداد نتایج نادرست و در نتیجه کاهش بار پردازش کمک کنند. از مشکلات فیلترهای محلی می توان کمبود داده‌های آموزشی را نام برد زیرا برای آموزش مدل‌های یادگیری ماشین به داده‌های کافی نیاز است و در زبان‌های کم‌داده، این داده‌ها ممکن است به اندازه کافی موجود نباشند.اما با استفاده از روش‌های انتقال یادگیری (Transfer Learning) می‌توان به این مشکل کمک کرد. به این ترتیب می‌توان مدل‌های آموزش‌دیده روی زبان‌های دیگر را برای زبان فرارسی تنظیم کرد. در استفاده از مدل فیلترهای محلی وقتی به زبانی با ساختار خاص و متفاوت مثل فارسی که دارای تنوع زبانی است برخورد می کنیم شناسایی موجودیت‌ها را پیچیده می‌کند که طراحی فیلترهای محلی خاص برای زبان مورد نظر می‌تواند به بهبود دقت کمک کند. این فیلترها باید با توجه به ویژگی‌های زبانی و فرهنگی زبان مورد نظر طراحی شوند.فیلترهای محلی به دلیل اینکه از دیکشنری ها به عنوان مجموعه داده استفاده می کنند ممکن است به طور کامل نتوانند عملکرد خوبی داشته باشند زیرا دیکشنری‌های موجود برای زبان‌های کم‌داده ممکن است ناقص یا نادرست باشند. استفاده از

تکنیک‌های خودکار برای تولید دیکشنری‌ها و شناسایی موجودیت‌ها می‌تواند مفید باشد. به عنوان مثال، می‌توان از منابع وب یا متون موجود برای استخراج موجودیت‌ها بهره برد. روش‌های ترکیب فیلترهای محلی و یادگیری ماشین می‌توانند به طور مؤثری در زبان‌های کم‌داده به کار روند، به شرطی که چالش‌های موجود شناسایی و مدیریت شوند. با استفاده از تکنیک‌های مناسب و منابع موجود، می‌توان به بهبود دقت و کارایی سیستم‌های تشخیص موجودیت نامدار در چنین زبان‌هایی دست یافت. برای اینکه فیلترهای محلی به خوبی عمل کنند چند راهکار پیشنهاد می کنیم: ۱- روش‌های یادگیری نیمه‌نظارت (Semi-Supervised Learning): این روش‌ها می‌توانند به استفاده از داده‌های بدون برچسب برای بهبود دقت مدل کمک کنند. به عنوان مثال، می‌توان از داده‌های موجود برای تقویت داده‌های برچسب‌دار استفاده کرد. ۲- استفاده از دیکشنری‌های چندزبان: اگر زبان مورد نظر شباهت‌هایی با زبان‌های دیگر داشته باشد، می‌توان از دیکشنری‌های چندزبان برای شناسایی موجودیت‌ها استفاده کرد. ۳- توسعه ابزارهای پردازش زبان طبیعی (NLP) مخصوص زبان‌های کم‌داده: می‌توان ابزارهای خاصی برای پردازش زبان‌های کم‌داده طراحی کرد که به صورت خاص به ویژگی‌های زبانی توجه کنند. ۴- جلب مشارکت جامعه: جمع‌آوری داده‌ها و کمک از جامعه محلی برای ایجاد منابع و دیکشنری‌های بهتر می‌تواند بسیار مؤثر باشد.

نتیجه گیری و تحلیل این روش ها

برای شناسایی موجودیت‌های نامدار در زبان فارسی، هر یک از مدل‌های ذکر شده دارای ویژگی‌ها و مزایای خاص خود هستند. در زیر به بررسی هر یک می‌پردازم و عملکرد آن‌ها را مقایسه می‌کنم:

در بیشتر تحقیقات در زمینه شناسایی موجودیت‌های نامدار (NER)، نشان داده‌اند که CRF عملکرد بهتری در مقایسه با مدل های تصادفی دیگر از جمله مارکوف مخفی (HMM) در این حوزه دارد. دلایل زیر به این موضوع اشاره دارند: ۱- کیفیت برچسب‌گذاری CRF: با طراحی ویژگی‌های مناسب به‌ویژه برای وظیفه (NER) برچسب‌گذاری خوبی انجام می‌دهد. ۲- عدم نیاز به استقلال ویژگی‌ها: در هنگام استفاده ازCRF، نیازی به مستقل بودن ویژگی‌ها نیست. این امر انعطاف‌پذیری بیشتری در انتخاب ویژگی‌ها ایجاد می‌کند. ۳- استفاده از اطلاعات مختلف CRF: می‌تواند از اطلاعات زبانی (مانند کلمات و کاراکترها) و غیر زبانی (مانند نشانه‌های نگارشی و فضاها) استفاده کند.

اصلی‌ترین عیب CRF ناشی از محاسبات پیچیده و طراحی الگوریتم آن در مرحله آموزش و یادگیری ماشین است. بنابراین، پس از افزودن نمونه‌های جدید داده، دشوار است که مدل دوباره آموزش داده شود]۲].

مدلCRF یک مدل مبتنی بر گراف است که به طور خاص برای پیش‌بینی توالی‌ها و یادگیری وابستگی‌ها طراحی شده است. ساده و قابل فهم، اما ممکن است از وابستگی‌های طولانی‌مدت غافل باشد. کارایی خوب در داده‌های کوچک و مشخص دارد. عملکرد این مدل ممکن است در مقایسه با BiLSTM و BERT دقت کمتری داشته باشد، به ویژه در زمینه‌های پیچیده‌تر اما در ترکیب با این مدل ها می تواند دقت و کارایی بالایی برای پردازش موجودیت ها ارئه دهد. با این حال، این مدل دارای محدودیت‌هایی است که در ادامه به برخی از آن‌ها اشاره می‌شود: ۱- نیاز به ویژگی‌های دستی: CRF به ویژگی‌های ورودی وابسته است که باید به‌طور دستی استخراج شوند. این فرآیند می‌تواند زمان‌بر و نیازمند دانش تخصصی باشد. ۲- عدم توانایی در یادگیری عمیق: برخلاف مدل‌های مبتنی بر یادگیری عمیق مانندBERT، CRF نمی‌تواند به‌طور خودکار ویژگی‌ها را از داده‌ها یاد بگیرد و به همین دلیل ممکن است در مقایسه با آن‌ها دقت کمتری داشته باشد. ۳- کاهش عملکرد در داده‌های کم: CRF به داده‌های آموزشی کافی برای یادگیری مناسب نیاز دارد. در صورت کمبود داده، عملکرد آن ممکن است به شدت کاهش یابد. ۴- پیچیدگی محاسباتی: آموزش CRF به دلیل محاسبات پیچیده‌ای که برای به‌روزرسانی وزن‌ها و محاسبه احتمال‌ها نیاز دارد، ممکن است زمان‌بر باشد. ۵- عدم قابلیت در مدل‌سازی وابستگی‌های بلندمدت: CRF به‌طور عمده بر روی وابستگی‌های محلی تمرکز دارد و نمی‌تواند به‌خوبی وابستگی‌های بلندمدت در متن را مدل‌سازی کند. ۶- دشواری در تنظیم دقیق: تنظیم بهینه پارامترهای CRF می‌تواند چالش‌برانگیز باشد و نیازمند آزمایش‌های متعدد است.

یکی از مزایای بزرگBERT این است که بر روی ۱۰۴ زبان زنده دنیا آموزش دیده است و زبان فارسی یکی از آن‌هاست. مدل BERT یک مدل مبتنی بر ترنسفورمر است که به صورت دوطرفه به یادگیری نمایه‌های زبانی می‌پردازد. از مزایای این مدل می

توان به توانایی در درک زمینه و روابط بین کلمات به دلیل دوطرفه بودن و عملکرد عالی در بسیاری از وظایفNLP ، از جمله شناسایی موجودیت‌های نامدار و قابلیت[۵]fine-tuning بر روی داده‌های خاص زبان فارسی نام برد.

فیلترهای محلی (Local Filters) در سیستم‌های تشخیص موجودیت نامدار (NER) دارای محدودیت‌هایی هستند که می‌تواند بر دقت و کارایی سیستم تأثیر بگذارد. در زیر به برخی از این محدودیت‌ها اشاره می‌کنیم:

۱- وابستگی به داده‌های آموزشی: فیلترهای محلی معمولاً بر اساس الگوها و ویژگی‌های موجود در داده‌های آموزشی کار می‌کنند. اگر داده‌های آموزشی ناکافی یا نادرست باشند، عملکرد این فیلترها تحت تأثیر قرار می‌گیرد و ممکن است نتایج نادرستی تولید کنند. ۲- کاهش دقت با فیلترهای سختگیرانه: برخی فیلترها ممکن است به طور نامناسبی سختگیرانه باشند. این می‌تواند منجر به حذف نامزدهای درست شود. به عنوان مثال، اگر فیلتر نوع کلمه بسیار محدود باشد، ممکن است نامزدهای درست را که در موقعیت‌های نادرست ظاهر می‌شوند، حذف کند. ۳- عدم توانایی در شناسایی زمینه‌های پیچیده: فیلترهای محلی گاهی نمی‌توانند زمینه‌های پیچیده و چندبعدی را تشخیص دهند. به عنوان مثال، در جملاتی با ساختار پیچیده یا چندمعنایی، این فیلترها ممکن است قادر به شناسایی موجودیت‌های نامدار نباشند. ۴- عدم درک معانی ضمنی: فیلترهای محلی معمولاً به صورت سطحی به تحلیل متن می‌پردازند. این می‌تواند منجر به نادیده گرفتن معانی ضمنی یا شناخت روابط بین کلمات شود. به عنوان مثال، "مردم ایران" ممکن است به عنوان موجودیت نامدار شناسایی نشود زیرا "مردم" به نظر نمی‌رسد که یک موجودیت باشد. ۵- عدم انعطاف‌پذیری: فیلترهای محلی معمولاً بر اساس الگوهای ثابت کار می‌کنند. این می‌تواند منجر به عدم توانایی در شناسایی موجودیت‌های جدید یا غیرمعمول شود، به ویژه در زبان‌های پویا و در حال تغییر. ۶- افزایش بار پردازشی: اعمال چندین فیلتر محلی می‌تواند باعث مصرف منابع پردازشی بیشتر شود. این می‌تواند بر کارایی سیستم تأثیر بگذارد، به ویژه در زمان پردازش و اجرای آن. در نتیجه فیلترهای محلی ابزارهای قدرتمندی برای بهبود دقت تشخیص موجودیت نامدار هستند، اما محدودیت‌های آن‌ها می‌تواند تأثیرات منفی بر عملکرد کلی سیستم داشته باشد. برای بهبود کارایی و دقت، نیاز به ترکیب این فیلترها با روش‌های دیگر و استفاده از داده‌های آموزشی با کیفیت وجود دارد. مدل متغیرهای محلی به طور خاص بر روی ویژگی‌های محلی و فیلتر کردن نتایج نادرست تمرکز دارند. این مدل مانند مدل CRF می‌توانند به عنوان یک لایه اضافی در کنار مدل‌های دیگر عمل کنند و دقت را بهبود ببخشند. عملکرد این مدل به تنهایی ممکن است به اندازه مدل‌های پیشرفته‌تر مانند BERT و BiLSTM- CRF مؤثر نباشند، اما می‌توانند به عنوان یک فیلتر مفید عمل کنند.

مدلBiLSTM-CRF که از دو مدل به نام LSTM برای پردازش دنباله‌ای استفاده می‌کند و CRF که برای بهبود پیش‌بینی‌ها و حدس توالی به کار می‌رود. از مهمترین مزایای این مدل می توان به توانایی در پردازش اطلاعات توالی و یادگیری وابستگی‌های طولانی‌مدت نام برد. این مدل عملکرد خوبی در شناسایی موجودیت‌ها دارد، به ویژه در داده‌های با ساختار مشخص.

برای داده‌های کم‌حجم، انتخاب مدل مناسب می‌تواند تأثیر زیادی بر عملکرد سیستم شناسایی موجودیت‌های نامدار (NER) داشته باشد. در این شرایط، مدل‌هایی که نیاز به داده‌های آموزشی بیشتری ندارند و می‌توانند به خوبی با داده‌های محدود کار کنند، مناسب‌تر هستند. CRF به دلیل ساختار ساده و نیاز کمتر به داده‌های آموزشی، می‌تواند گزینه مناسبی برای داده‌های کم‌حجم باشد. این مدل قابلیت یادگیری وابستگی‌های محلی بین برچسب‌ها را دارد و می‌تواند در شناسایی موجودیت‌ها به خوبی عمل کند. و در صورت وجود ویژگی‌های خوب و طراحی مناسب، عملکرد قابل قبولی خواهد داشت. مدل BiLSTM-CRF که ترکیبی از LSTM و CRF است، می تواند تا وابستگی‌های طولانی‌مدت را یاد بگیرد در نتیجه می‌توان با داده‌های کم نیز آن را آموزش داد، اما ممکن است نیاز به تنظیمات بهینه داشته باشد.عملکرد این مدل در مقایسه با CRF ، ممکن است دقت بالاتری داشته باشد، اما ممکن است به زمان بیشتری برای آموزش نیاز داشته باشد. مدل متغیرهای محلی می‌توانند به عنوان یک روش ساده و کم‌هزینه برای بهبود شناسایی موجودیت‌ها در داده‌های کم‌حجم عمل کنند. می‌توانند به راحتی به مدل‌های دیگر افزوده شوند و به عنوان یک فیلتر عمل کنند و به تنهایی ممکن است به اندازه مدل‌های پیشرفته‌تر مؤثر نباشند، اما می‌توانند به بهبود

---

[۵] (Fine-Tuning) فرایندی است که در آن پارامترهای یک مدل باید به‌صورت خیلی دقیقی تنظیم شوند تا مدل با مشاهدات مشخصی تناسب پیدا کند. این فرایند در یادگیری ماشین یک فرایند بسیار متداول است که در آن با اضافه کردن لایه های جدید مدل از پیش آموزش داده شده‌ای، سعی می‌شود که پارامتر های آن مدل حاصل به گونه ای تنظیم شود که دقت مدل روی یک مجموعه داده ی جدید نیز بالا رود.

دقت کمک کنند. اگر امکان استفاده از مدل‌های پیش‌آموزش‌دیده(Pre-trained) مانند BERT برای زبان فارسی وجود داشته باشد، می‌توان از آن‌ها نیز استفاده کرد. این مدل‌ها می‌توانند با داده‌های کمتر نیز خوب عمل کنند. عملکرد این مدل با انتخاب fine-tuning مناسب، می‌توانند به دقت بالایی دست یابند. در نتیجه برای داده‌های کم‌حجم، مدل CRF یا مدل BiLSTM-CRFبه عنوان گزینه‌های مناسب پیشنهاد می‌شوند. اگر امکان استفاده از مدل‌های پیش‌آموزش‌دیده وجود داشته باشد، این گزینه‌ها می‌توانند عملکرد بهتری را ارائه دهند. همچنین، استفاده از مدل‌های متغیرهای محلی به عنوان فیلتر می‌تواند در بهبود دقت کمک کند. جدول ۱ پیچیدگی زمانی و مکانی این چهار مدل را بیان می کند.

جدول (۱): n طول دنباله ورودی است، m تعداد لایه های LSTM و تعداد ویژگی هایCRF است.

نام مدل	پیچیدگی زمانی	پیچیدگی حافظه‌ای
BERT	$O(n^2)$	بالاترین نیاز به حافظه
BiLSTM-CRF	O(n*m)	متوسط
CRF	$O(n^2*m)$	پایین تر
متغیرهای محلی	O(n)	کمترین نیاز به حافظه

با توجه به جدول (۱) برای شرایطی که منابع زمان و حافظه محدود هستند استفاده از CRF یا متغیرهای محلی می‌تواند مناسب‌تر باشد. اگر به دقت بالاتری نیاز است و منابع کافی وجود دارد، BiLSTM-CRF یا BERT گزینه‌های بهتری هستند. در نتیجه بهترین گزینه در مرحله اول مدل BERT به دلیل قابلیت‌های پیشرفته‌اش در فهم زمینه و روابط بین کلمات، همچنین توانایی fine-tuning بر روی داده‌های فارسی است. گزینه دوم مدل BiLSTM-CRF نیز عملکرد خوبی دارد و به ویژه در داده‌های با ساختار مشخص می‌تواند مؤثر باشد. گزینه سوم مدل CRF ممکن است برای داده‌های کوچک و ساده مناسب باشد، اما به طور کلی دقت کمتری نسبت به دو مدل بالا خواهد داشت. و در آخر مدل متغیرهای محلی می‌تواند به عنوان یک افزونه مفید برای بهبود دقت در کنار سایر مدل‌ها عمل کند.

منابع

[۱] اصفهانی، سید عبدالحمید، راحتی قوچانی، سعید، جهانگیری، نادر،۱۳۸۹، "سیستم شناسایی و طبقه بندی اسامی در متون فارسی"

[2] Jafar Tafreshi, L, Soltanzadeh, F, "A Novel Approach to Conditional Random Field-based Named Entity Recognition using Persian Specific Features", Journal of AI and Data Mining, 10.22044,2020

[3] Taher, Ehsan, Hoseini, Seyed Abbas, Shamsfard, Mehrnoush, " Beheshti-NER: Persian named entity recognition Using BERT ", Shahid Beheshti University, arXiv, arXiv:2003.08875, 2020

[4] Poostchi Hanieh, Zare Borzeshi, Ehsan, Piccardi, , Massimo, " BiLSTM-CRF for Persian Named-Entity Recognition: ArmanPersoNERCorpus: the First Entity-Annotated Persian Dataset ", arXiv, , 2020

[5] Kolali Khormuji, Morteza Bazrafkan, Mehrnoosh, " Persian Named Entity Recognition based with Local Filters ", International Journal of Computer Applications, 100, 2,2014

Analysis and Comparison of Intelligent Named Entity Recognition Methods in Persian Language

Zeinab Mohammadnia
Department of Medical Information Technology, Qom, Iran
golsamohamadi2020@gmail.com

Abstract

Named Entity Recognition (NER) acts as an information extraction technique that identifies named entities in text and is a branch of Natural Language Processing (NLP). Traditionally, there are four main methods for extracting these entities: rule-based methods, machine learning, deep learning, and a combination of these methods. This paper examines several models, including Conditional Random Fields (CRF), Google BERT, BiLSTM-CRF, and Local Filters, for processing entities in the Persian language and compares their results.

Keywords: NER, BERT, CRF, BiLSTM, Local Filters