Journal of Applied And Basic Machine Intelligence Research (ABMIR)

پژوهشهای نظری و کاربردی هوش ماشینی

سال اول، شماره اول، بهار و تابستان ۱۴۰۲، صفحات ۳۳-۲۳

یک مدل نوین مبتنی بر شبکه‌های عصبی عمیق برای برچسب‌گذاری اجزای واژگانی کلام

فریبا تقی نژاد [۱]، محمد قاسم‌زاده [۲*]

[۱] دانشجوی دکترای دانشکده مهندسی کامپیوتر، پردیس فنی و مهندسی، دانشگاه یزد، یزد، ایران

[۲] دانشیار دانشکده مهندسی کامپیوتر، پردیس فنی و مهندسی، دانشگاه یزد، یزد، ایران

چکیده

مقاله پژوهشی

برچسب‌گذاری اجزای واژگانی کلام موضوع تحقیقاتی مهمی در حوزهٔ پردازش زبان طبیعی است و پایه بسیاری از دیگر مباحث مطرح در این حوزه است. در این مقاله یک روش نوین برای برچسب‌گذاری اجزای واژگانی کلام به کمک شبکه‌های عصبی عمیق معرفی می‌گردد. هدف اصلی مدل پیشنهادی، استخراج ویژگی‌های عمیق و سطح بالا از متون و سپس طبقه‌بندی این ویژگی‌های سطح بالا است. روش پیشنهادی متکی بر این ایده است که از شبکه عصبی عمیق کوچک می‌توان برای یافتن ویژگی‌های عمیق و تولید خروجی مطلوب بهره برد. روش موردنظر با استفاده از کتابخانه‌های تخصصی **Tensorflow** و **Kera API** در پایتون پیاده‌سازی و عملکرد آن بر روی مجموعه داده استاندارد **coNLL2000** ارزیابی گردید. نتایج آزمایش حاکی از آن است که روش پیشنهادی قابلیت استخراج ویژگی‌های سطح بالای واژگان زبان طبیعی را داشته و قادر است به ازای برچسب‌های پرتکرار و پرکاربرد به‌دقت قابل‌توجهی برسد. میانگین دقت مدل پیشنهادی به ازای برچسب‌های مختلف برابر ۸۰/۲۶٪ بوده است. بعلاوه، این روش قابلیت استفاده در محیط‌های متنوع و بر روی دستگاه‌های مختلف را نیز دارد.

تاریخ دریافت:
۱۴۰۰/۰۸/۰۳

تاریخ پذیرش:
۱۴۰۰/۱۲/۲۱

کلیدواژه‌ها:
برچسب‌گذاری اجزای واژگانی کلام، پردازش زبان طبیعی، شبکه‌های عصبی عمیق، شبکه‌های عصبی کانولوشن

نویسنده مسئول:
m.ghasemzadeh@yazd.ac.ir

doi: 10.22034/ABMIR.2022.2660

۱- مقدمه

پردازش زبان طبیعی یکی از زیرشاخه‌های بااهمیت در حوزه گسترده علوم رایانه، هوش مصنوعی و نیز زبان‌شناسی محاسباتی است که به تعامل بین کامپیوتر و زبان‌های (طبیعی) انسانی می‌پردازد؛ بنابراین پردازش زبان‌های طبیعی بر ارتباط انسان و رایانه، متمرکز است. به تعریف دقیق‌تر، پردازش زبان‌های طبیعی عبارت است از استفاده از رایانه برای پردازش زبان گفتاری و زبان نوشتاری. بدین معنی که رایانه‌ها را قادر سازیم که گفتار یا نوشتار تولیدشده در قالب و ساختار یک‌زبان طبیعی را تحلیل و درک نموده یا آن را تولید نمایند. در این صورت با استفاده از آن می‌توان به ترجمه زبان‌ها پرداخت، از صفحات وب و بانک‌های اطلاعاتیِ نوشتاری جهت پاسخ دادن به پرسش‌ها استفاده کرد، یا با دستگاه‌ها، مثلاً برای مشورت گرفتن، به گفت‌وگو پرداخت. چالش اصلی و عمده در این زمینه، درک زبان طبیعی و ایجاد زبان طبیعی است.

یکی از بخش‌های کلیدی در پردازش متن، تعیین نقش کلمه در جمله است. هدف از برچسب‌گذاری اجزای واژگانی کلام[1]، نسبت دادن برچسب‌های واژگانی به واژه‌ها و نشانه‌های بکار رفته در یک متن است [۱]. معیارهای ارزیابی الگوریتم‌های برچسب‌گذاری، دقت برچسب‌گذار[2] و نرخ خطا[3] می‌باشند. برچسب‌گذاری اجزای واژگانی کلام یکی از مهم‌ترین مراحل میانی در بسیاری از کاربردهای پردازش زبان طبیعی است؛ به‌عنوان نمونه می‌توان به ترجمه خودکار، تجزیه‌وتحلیل نحوی، بازیابی اطلاعات، خطایابی املایی و نحوی و کاربردهای متفاوت متن‌کاوی اشاره کرد.

اهمیت برچسب‌گذاری به‌عنوان یک مرحله میانی، در این کاربردها ازآن‌جهت است که در بسیاری از موارد میزان دقت برچسب‌های نسبت داده‌شده، در دقت نهایی تأثیرگذار است [۲]. تحلیل احساسات به کمک برچسب‌گذاری کلمات[۳] و تجزیه‌وتحلیل احساسات فیلم با استفاده از برچسب‌گذاری کلمات و فرکانس کلمات[٤]و استخراج احساسات و حالات موجود در متن وبلاگ‌ها با استفاده از ویژگی‌های استخراج‌شده بر اساس برچسب‌گذاری کلمات]۵[نیز از دیگر کاربردهای برچسب‌گذاری کلمات در زبان‌های طبیعی است. در]٦[به کمک POS های مؤثر در تحلیل احساسات، اقدام به تجزیه‌وتحلیل کارآمد احساسات شده است. در این مقاله، ابتدا به ازای POS Tag های مهم در عبارت یا جمله s، سه درجه قطبیت Obj و Pos و Neg محاسبه می‌شود که به ترتیب، بیانگر میزان بی‌طرفی، مثبت بودن و منفی بودن آن POS Tag است. سپس از طریق جمع درجه قطبیت تمام POS Tag های موجود در عبارت یا جمله، مقادیر Obj(s) و Pos(s) و Neg(s) محاسبه می‌شود. در انتها، بر اساس سه مقدار به‌دست‌آمده جهت احساسات یا SO (Sentiment Orientation) موجود در آن عبارت محاسبه می‌شود. اگر مقدار SO یک عبارت یا جمله از حد آستانه بیشتر باشد به‌عنوان یک عبارت Positive دسته‌بندی می‌شود و در غیر این صورت به‌عنوان یک عبارت Negative دسته‌بندی می‌شود.

در این مقاله مدلی مبتنی بر شبکه‌های عصبی عمیق برای مسأله برچسب‌گذاری اجزای واژگانی کلام ارائه می‌شود. مدل پیشنهادی با دریافت یک ماتریس شامل تصویر ویژگی هر واژه در جمله، به دنبال یافتن بهترین برچسب برای واژگان به‌کاررفته در جمله ورودی است. ادامه مقاله به‌صورت زیر بیان می‌شود. در بخش دوم کارهای انجام‌گرفته در حوزه برچسب‌گذاری واژگان کلام ارائه می‌شود و در بخش سوم مفهوم شبکه‌های عصبی عمیق و کاربرد آن‌ها در مسائل طبقه‌بندی بیان می‌شود و سپس برچسب‌گذاری اجزای واژگان کلام تشریح می‌شود، در بخش چهارم ساختار مدل پیشنهادی تشریح می‌شود، در بخش پنجم نتایج اجرای مدل پیشنهادی به ازای مجموعه داده‌ای coNLL2000 ارائه می‌شود و در بخش ششم نتیجه‌گیری نهایی بیان می‌شود

۲- مرور سوابق

در سال‌های اخیر الگوریتم‌ها و تکنیک‌های زیادی برای برچسب‌گذاری اجزای واژگانی کلام در زبان‌های مختلف ارائه شده‌اند. روش‌های برچسب‌گذاری کلمات را به دسته‌های کلی زیر می‌توان تقسیم کرد [۷]: ۱- روش‌های مبتنی بر قواعد[4]، ۲-

[1] Part of Speech Tagging (POS Tagging)
[2] Tagging Accuracy (TA)
[3] Error Rate (ER)
[4] Rule-Based Methods

روش‌های آماری[1]، ۳-روش‌های مبتنی بر گذار[2]، ٤-روش‌های مبتنی بر حافظه[3]، ۵-روش‌های ترکیبی[4].

در روش‌های مبتنی بر قواعد، با استفاده از یک واژه‌نامه و مجموعه‌ای از قواعد زبانی، اقدام به برچسب‌گذاری واژگان زبانی می‌شود. واژه‌نامه حاوی تمام واژه‌های یک‌زبان و نیز نقش‌ها یا برچسب‌هایی است که هر واژه می‌تواند داشته باشد. قواعد نیز شامل قواعد زبانی است و توسط زبان‌شناسان فراهم می‌شود. به ازای هر ورودی یا عبارت جدید، ابتدا به کمک واژه‌نامه تمام واژه‌ها و برچسب‌هایشان مشخص می‌شود و سپس به کمک قواعد زبانی از میان برچسب‌های هر واژه تنها یک برچسب به‌عنوان برچسب نهایی انتخاب می‌شود. این روش‌ها به‌صورت با ناظر و بدون ناظر قابل‌اجرا هستند.

در روش‌های آماری از یک پیکره از پیش برچسب خورده استفاده می‌شود و با استفاده از توابع آماری و قواعد احتمالی، برچسب یک واژه در متن پیش‌بینی می‌شود. اساس بسیاری از روش‌های آماری بر تخمین احتمال بیشنه است. Dalal و همکاران [۹]توانستند با استفاده از بیشنه آنتروپی مارکو[5] و مجموعه‌ای غنی از ویژگی‌های زبانی در زبان هندی به‌دقت ۹٤/۸۹٪ برسند. AlKhwiter و Al-Twairesh[۱۰]دو برچسب‌گذار با ناظر و مبتنی بر فیلدهای تصادفی مشروط[6](CRF) و مدل‌های حافظه کوتاه‌مدت بلندمدت دوطرفه[7](Bi-LSTM) برای برچسب‌گذاری کلمات عربی در توییتر ارائه کرده‌اند. نتیجه آزمایش‌ها نشان داده‌اند که مدل Bi-LSTM دارای دقت بالاتری بوده و به ازای سه مجموعه داده‌ای دارای دقت ٪ ۹٦/۵ ,٪ ۹۵/٦ ,٪ ۹۵ بوده است. الگوریتم CRF در بهترین حالت دارای دقت ۹۱/۰۳٪ بوده است.

روش‌های مبتنی بر گذار، ترکیبی از روش‌های مبتنی بر قواعد و روش‌های آماری هستند [۱۱]. در روش‌های مبتنی بر حافظه از یادگیری مبتنی بر حافظه استفاده می‌شود که به‌صورت با ناظر عمل نموده و از استدلال مبتنی بر تشابه استفاده می‌کند[۷].

در روش‌های ترکیبی از ترکیب روش‌های فوق و روش‌های جستجوی حریصانه یا الگوریتم‌های بهینه‌سازی به‌منظور یافتن بهترین برچسب برای واژگان کلام استفاده می‌شود. Forsati و Shamsfard [۱۲] ابتدا مسأله برچسب‌گذاری واژگان را به‌عنوان یک مسأله بهینه‌سازی با اهداف مشخص تعریف نموده و سپس با کمک الگوریتم‌های تکاملی مستقل از زبان اقدام به یافتن پاسخ بهینه می‌نمایند. دقت الگوریتم پیشنهادی در بهترین حالت و به ازای محاسبه تابع احتمالی بر اساس یک کلمه ماقبل و یک کلمه مابعد هر کلمه در متن و استفاده از الگوریتم بهینه‌سازی کلونی زنبورعسل، ۹٦/٦۵٪ بوده است. یکی از معایب این روش زمان لازم برای یافتن پاسخ بهینه است. Alhasan و Al-Taani [۱۳] یک روش برچسب‌زنی کارآمد برای واژگان زبان عربی و به کمک الگوریتم بهینه‌سازی کلونی زنبورعسل ارائه کرده‌اند. دقت روش ارائه‌شده برای پیکره‌ای با زبان عربی ۹۸/۲٪ بوده است.

امروزه با توجه به توانایی بالای شبکه‌های عصبی عمیق[8] و به‌ویژه شبکه‌های عصبی کانولوشن[9] در استخراج اتوماتیک ویژگی‌های سطح بالا برای داده ورودی و سپس طبقه‌بندی ورودی به ازای ویژگی‌های استخراج‌شده، تحقیقاتی درزمینه‌ٔ استفاده از شبکه‌های عصبی عمیق برای برچسب‌گذاری انجام‌گرفته است. Dhumal و Kiwelekar [۱٤]دو برچسب‌گذار مختلف را بر روی پیکره زبان مراتی، چهارمین زبان تحت تکلم توسط هندیان، ارائه داده‌اند که مبتنی بر مدل Bi-LSTM و شبکه‌های عصبی عمیق بوده است. دقت برچسب‌گذار مبتنی بر Bi-LSTM برابر ۹۷٪ بوده و دقت برچسب‌گذار مبتنی بر شبکه‌های عصبی عمیق ۸۵٪ گزارش‌شده است.

۳- طرح مسأله

یکی از چالش‌های اصلی در تحقیقات NLP، در مقایسه با حوزه‌های دیگر مانند بینایی کامپیوتری، پیچیدگی دستیابی به نمایش عمیق زبان با استفاده از مدل‌های آماری و نیز پیچیدگی

[1] Stochastic-Based Methods
[2] Transition-Based Methods
[3] Memory-Based Methods
[4] Hybrid Methods
[5] Maximum Entropy Markov (MEM)

[6] Conditional Random Fields (CRF)
[7] Bi-Long Short Time Memory (Bi-LSTM)
[8] Deep Neural Networks (DNN)
[9] Convolutional Neural Networks (CNN)

استخراج ویژگی‌های مناسب از داده‌ها است. وظیفه اصلی در برنامه‌های NLP، ارائه نمایشی از متون، مانند اسناد است که شامل یادگیری ویژگی، یعنی استخراج اطلاعات معنی‌دار برای فعال کردن پردازش و تحلیل بیشتر داده‌های خام است. روش‌های سنتی از طریق تجزیه‌وتحلیل دقیق انسانی، ویژگی‌های مناسب برای نمونه‌ها را شناسایی و استخراج می‌نمایند و سپس با استفاده از الگوریتم‌های مختلف اقدام به طبقه‌بندی نمونه‌ها می‌نمایند. از سوی دیگر، روش‌های یادگیری ویژگی نظارت‌شده عمیق بسیار داده محور هستند و می‌توانند در تلاش‌های عمومی باهدف ارائه یک نمایش داده قوی مورد روش‌های سنتی از طریق تجزیه‌وتحلیل دقیق انسانی، ویژگی‌های مناسب برای نمونه‌ها را شناسایی و استخراج می‌نمایند و سپس با استفاده از الگوریتم‌های مختلف اقدام به طبقه‌بندی نمونه‌ها می‌نمایند. از سوی دیگر، روش‌های یادگیری ویژگی نظارت‌شده عمیق بسیار داده محور هستند و می‌توانند در تلاش‌های عمومی باهدف ارائه یک نمایش داده قوی مورداستفاده قرار گیرند. با توجه به برتری بالقوه رویکردهای یادگیری عمیق در برنامه‌های NLP، پیاده‌سازی مسأله برچسب‌گذاری اجزای واژگانی کلام به کمک یادگیری عمیق ضروری به نظر می‌رسد.

هدف از برچسب‌گذاری اجزای واژگانی کلام، تعیین نقش واژگان در جمله یا جملاتی از زبان طبیعی یا تعیین نوع علایم نگارشی به‌کاررفته در متن است. برچسب‌ها نشان‌دهنده نقش زبانی هر واژه در متن هستند. البته به هر یک از علامت‌های نگارشی موجود در متن نیز رده‌ای از این برچسب‌ها نسبت داده می‌شود که نوع هر یک از آن‌ها را مشخص می‌کند. مجموعه برچسب‌ها در دادگان مختلف متفاوت است.

٤- راهکار پیشنهادی

هدف اصلی مقاله، طراحی یک مدل و تنظیم بهینه پارامترهای آن است به‌گونه‌ای که ورودی آن، جمله یا جملاتی در زبان طبیعی و در قالب متن نوشتاری است و خروجی آن نیز لیستی شامل نقش یا برچسب کلمات جمله یا جملات ورودی است. بخش مرکزی این سیستم یک مدل دسته‌بندی است که به ازای هر کلمه در جمله ورودی، برچسب آن را تخمین می‌زند. برچسب‌ها بیانگر کلاس‌های مسأله دسته‌بندی هستند لذا در این مسأله به دنبال یافتن یک مدل دسته‌بندی بهینه و با دقت بالا برای یک مسأله چندکلاسه هستیم.

جمله ورودی شامل لیستی از کلمات و به‌صورت S=[w1,w2,…,wn] داده‌شده است. هدف یافتن بهترین برچسب برای هر کلمه در جمله ورودی است: T=[t1,t2,…,tn]. برای این منظور مدلی پیشنهاد می‌گردد که ساختار کلی آن مطابق شکل (۱) است. مدل پیشنهادی شامل دو مرحله زیر است.

٤-١ مرحله اول: پیش‌پردازش داده‌ها

در این مرحله، ابتدا به ازای هر کلمه در جمله ورودی، یک تصویر ویژگی ساخته می‌شود که ماتریسی به ابعاد (۲۰*۱٤) است. برای ساخت تصویر ویژگی هر کلمه، ابتدا با توجه به لغات موجود در پیکره coNLL-2000 لغت‌نامه‌ای حاوی کلمات مجزا ساخته می‌شود و سپس به هر کلمه در پیکره و با توجه به جایگاه کلمه در لغت‌نامه کلمات، کدی ١٦ بیتی اختصاص می‌یابد که برابر نمایش باینری عدد مربوطه است. به‌عنوان مثال کلمه‌ای که در ردیف ۲۷۵ لغت‌نامه قرار دارد دارای کد '0000000100000001' خواهد بود. پیکره coNLL-2000 شامل ۱۹٤۹۰ کلمه مجزا است و از رشته‌های عددی صرف‌نظر کرده‌ایم. در شکل (۲) قسمتی از لغت‌نامه و کد هر کلمه در لغت‌نامه را مشاهده می‌کنید.

در قدم بعدی و به ازای تمام برچسب‌های موجود در پیکره coNLL-2000، لغت‌نامه برچسب‌ها ساخته می‌شود و به هر برچسب در پیکره و با توجه به جایگاه برچسب در لغت‌نامه برچسب‌ها، کدی ١٦ بیتی اختصاص می‌یابد.

لغت‌نامه برچسب‌ها شامل ٤٣ برچسب مختلف است. در شکل (۳)، نمایی از لغت‌نامه برچسب‌ها و کد باینری اختصاص‌یافته به تعدادی از برچسب‌ها را مشاهده می‌کنید.

سپس به ازای هر کلمه در جمله ورودی، ماتریسی با ابعاد (١٦*١٠) تشکیل می‌شود. این ماتریس شامل اطلاعات زیر است:

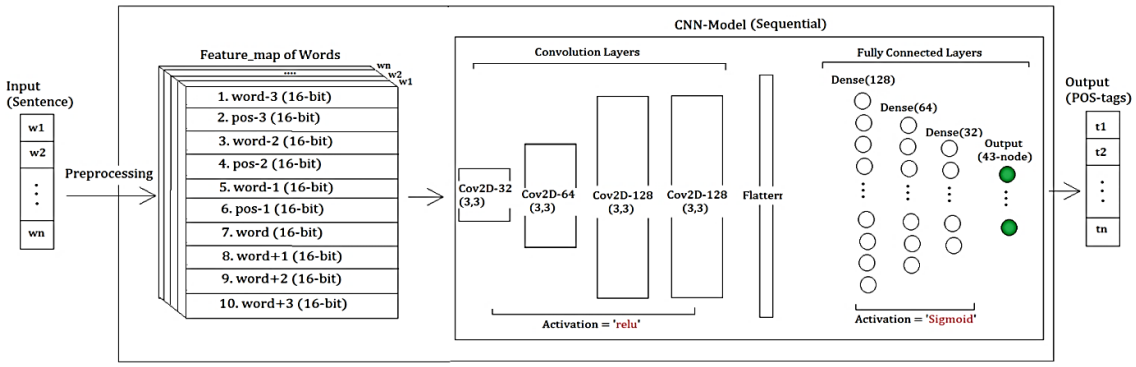
- سه کلمه ماقبل (ردیف اول) و برچسب سه کلمه ماقبل (ردیف دو)

- دو کلمه ماقبل و برچسب دو کلمه ماقبل (ردیف سه و چهار)
- کلمه ماقبل و برچسب کلمه ماقبل (ردیف پنج و شش)
- کلمه اصلی (ردیف هفت)
- کلمه مابعد، دو کلمه مابعد و سه کلمه مابعد (ردیف‌های هشت و نه و ده ماتریس)

هرکدام از این اطلاعات، در قالب یک عدد باینری و به طول ۱۶ کدگذاری شده‌اند. در شکل (٤) نمونه‌ای از تصویر ویژگی‌های ساخته‌شده برای تعدادی از کلمات مجموعه داده‌ای و با برچسب‌های مختلف را مشاهده می‌کنید. قابل‌ذکر است که تصویر ویژگی یک کلمه به جایگاه آن در جمله‌اش بستگی دارد لذا یک کلمه می‌تواند دارای تصویر ویژگی‌های مختلفی باشد.

برای کلمات ابتدای جمله که دارای برخی کلمات ماقبل و برچسب آن‌ها نمی‌باشند و یا کلمات انتهای جمله که دارای دو یا سه کلمه مابعد نیستند مقدار Null در نظر گرفته‌شده است. این مقدار یک عدد به طول ۱۶ و با مقادیر "-1" است. در شکل (۵) نمونه‌ای از تصویر ویژگی‌های کلمات ابتدا یا انتهای جملات را مشاهده می‌کنید. در مرحله نهایی دولایه با مقدار "0" حول هر تصویر تشکیل می‌شود تا تأثیر مقادیر حاشیه‌ای بر فرایند استخراج ویژگی حفظ شود. در شکل (٦) نمونه‌ای از تصویر ویژگی‌های نهایی برای تعدادی از کلمات با برچسب‌های مختلف را مشاهده می‌کنید.

با توجه به جایگاه هر برچسب در لغت‌نامه برچسب‌ها و به ازای هر برچسب، کد صحیح و منحصربه‌فردی اختصاص می‌یابد. برای مثال، برچسبی که در جایگاه ۱۵ لغت‌نامه برچسب‌ها قرار دارد دارای کد ۱۵ خواهد بود. تعداد کل برچسب‌ها برابر ٤۳ است و لذا به هر برچسب، کدی صحیح در بازه [0,42] اختصاص می‌دهیم.



شکل (۱): ساختار کلی مدل پیشنهادی

trade	0	1	0	0	0	1	1	1	1	1	1	0	1	0	0	0
figures	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1
September	0	0	0	1	0	1	0	0	1	1	0	0	1	1	0	1
،	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
due	0	0	1	0	0	1	1	1	0	1	0	1	0	1	1	1
for	0	0	1	0	1	0	1	1	1	0	0	0	0	1	0	0
tomorrow	0	1	0	0	0	1	1	1	1	0	0	1	1	1	1	1

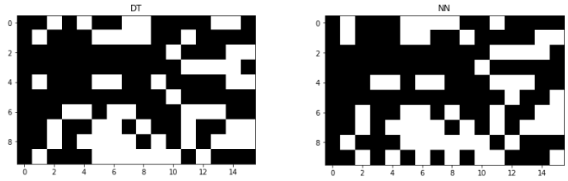
شکل (۲): نمایی از لغت‌نامه واژگان و کدگذاری آن‌ها

NN	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
NNS	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
NNP	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
,	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
JJ	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
IN	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

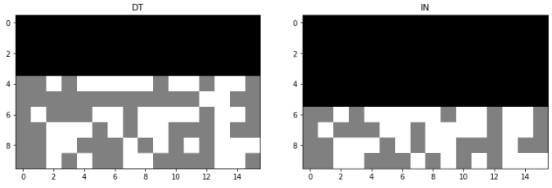
شکل (۳): نمایی از لغت‌نامه برچسب‌ها و کدگذاری آن‌ها

۲-٤ مرحله دوم: استخراج ویژگی‌ها و طبقه‌بندی به کمک شبکه‌های عصبی عمیق (CNN)

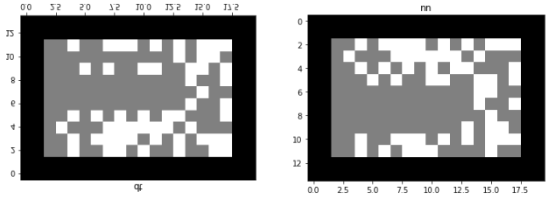
در این مرحله، ابتدا با استفاده از لایه‌های Convolution اقدام به استخراج ویژگی‌های سطح بالا نموده‌ایم و سپس به کمک لایه‌های مربوط به شبکه‌های عصبی عمیق، عمل طبقه‌بندی بردارهای ویژگی استخراج‌شده انجام می‌پذیرد. ورودی این مرحله، تصویر ویژگی‌ها به ازای هر کلمه است و خروجی نیز لایه‌ای شامل ٤٣ نود مختلف به ازای ٤٣ کلاس مختلف است. به ازای هر کلمه در ورودی و توسط شبکه CNN، یک عدد احتمالاتی برای هر نود خروجی محاسبه می‌شود و سپس نودی با بیشترین احتمال، دارای مقدار ۱ شده و بقیه نودها دارای مقدار 0 خواهند بود. خروجی نهایی به‌صورت یک عدد صحیح و دربازه [0,42] خواهد بود که بیانگر کد برچسب انتخاب‌شده است. شبه کد الگوریتم پیشنهادی در شکل (۷) قابل‌مشاهده است.



شکل (٤): تصویر ویژگی‌ها مربوط به کلماتی با برچسب‌های مختلف



شکل (۵): نمونه‌ای از تصویر ویژگی‌ها مربوط به کلماتِ ابتدای جمله



شکل (٦): نمونه‌ای از تصویر ویژگی‌های نهایی تولیدشده برای کلمه با برچسب NNو DT

۵- نتیجه آزمایشات

پیاده‌سازی مدل پیشنهادی با استفاده از زبان پایتون و در پلتفرم Tensorflow و به کمک کتابخانه‌ها و توابع Keras-API انجام‌گرفته است. نحوه تنظیم پارامترهای مدل پیشنهادی و نتایج اجرای مدل به ازای مجموعه داده‌ای coNLL2000 به‌صورت زیر بوده است.

۵-۱ تنظیم پارامترها

به دلیل ماهیت چند کلاسه بودن مسأله برچسب‌گذاری و خروجی‌های طبقه‌بندی‌شده و قیاسی و نیز به دلیل صحیح بودن خروجی، تابع خطای مدل از نوع Sparse-CategoricalEntropy در نظر گرفته‌شده است. در ابتدا و با توجه به ماهیت مسأله تصمیم بر این بود که از تابع Categorical-CrossEntropy به‌عنوان تابع خطا استفاده گردد که به‌دقت بالایی نیز دست‌یافتیم، ولی در مرحله آزمایش دستی و عملی (بر روی جملات دلخواه)، دقت مدل بسیار کاهش یافت و لذا در مرحله بعد، از تابع خطای Sparse-Categorical-CrossEntropy برای آموزش مدل استفاده شد و دقت آن در مقایسه با دقت Categorical-CrossEntropy بسیار کم شد. ولی به دلیل موفقت آمیز بودن آزمایش دستی و عملی

(برای جملات دلخواه) تصمیم بر آن شد تا از تابع خطای Sparse-Categorical-CrossEntropy در این مقاله استفاده شود.

نرخ یادگیری مدل نیز ثابت و برابر ۰/۱ و ۰/۰۱ فرض شده است. با توجه به اینکه مقادیر بالای نرخ یادگیری باعث جلوگیری از همگرایی الگوریتم به مقادیر بهینه می‌شود و مقادیر کوچک نرخ یادگیری نیز احتمال رسیدن به نقاط اکسترمم محلی را افزایش می‌دهد لذا تصمیم بر این شد تا مدل ارائه‌شده، هم به ازای یک مقدار کوچک و هم به ازای یک مقدار بزرگ آموزش و آزمایش شود و در پایان و با مقایسه دقت مدل‌ها، مقادیر بهینه انتخاب شوند.

پیشنهاد می‌شود برای کارهای آینده از ویژگی‌های Early Stopping و نرخ‌های یادگیری کاهشی در شبکه‌های عصبی عمیق استفاده شود تا مقادیر بهینه پارامترها به‌صورت اتوماتیک محاسبه شود.

شاخص استفاده شده برای ارزیابی دقت مدل پیشنهادی نیز برابر Sparse-Categorical-CrossEntropy بوده است، فرمول محاسبه تابع خطای فوق‌الذکر طبق رابطه (۱) انجام می‌گیرد:

```
Pseodocode POS-Tagging ( ) {

    #Read datasets
    read_dataset ( )

    #Preprocessing
    construct_words_dictionary ( )
    construct_pos_tag_dictionary ( )
    extract_index_for_each_word ( )
    extract_index_for_each_pos_tag ( )

    extract_feature_map_for_each_word_in_dataset ( )

    #Construct and Traning Proposed CNN_based Model
    import necessary libraries from tensorflow & keras
    split dataset to train/validation/test datasets

    for epoch in [10,50,100] do
        construct a sequential CNN Model based on figure-7
        train_the_model ( epoch ) #Based on train/validation datasets
        evaluate_model ( ) #Calculating accuracy of model based on test dataset

    select_best_parameters ( )
```

شکل (۷): شبه کد الگوریتم پیشنهادی

$$J(w) = -\sum_{c=0}^{M} y_c \log(p_c) \qquad (۱)$$

که در آن؛ w پارامترهای مدل است، y_c برچسب واقعی به ازای کلاس c است و p_c نیز مقدار خروجی مدل به ازای کلاس c است که به‌صورت یک عدد احتمالی در بازه [0,1] و به ازای هر نود خروجی تولید می‌شود و پارامتر M بیانگر تعداد کلاس‌ها یا تعداد برچسب‌هاست و داریم: M=42.

در این مقاله از مجموعه داده‌ای پیکره coNLL2000 استفاده کرده‌ایم که شامل ۲۴۸۸۷۱ کلمه/برچسب در زبان انگلیسی است. برای اجرای مدل بر روی مجموعه داده‌ای، ۶۰٪ و ۲۰٪ و ۲۰٪ کل

داده‌ها، به ترتیب، به داده‌های Train، Test و Validation اختصاص یافته است. اسامی و تعداد برچسب‌ها در جدول (۱) بیان شده است. این جدول شامل اسامی برچسب‌ها، تعداد وقوع برچسب در پیکره coNLL2000 است. فرایند یادگیری مدل به ازای epoch=[10,50,100] انجام‌گرفته و نتایج به ازای تمامی حالات ارائه‌شده است. مقدار پارامتر Batchsize=50 فرض شده است.

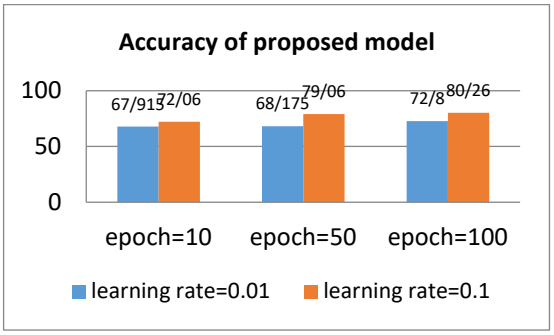
جدول (۱): اسامی برچسب‌ها و تعداد وقوع هر برچسب در کل پیکره coNLL2000

Tag Name	Tag Count	Tag Name	Tag Count	Tag Name	Tag Count
CC	6586	PRP	4634	SYM	**6**
DT	22355	PRP$	2302	TO	**6259**
EX	254	RB	7961	UH	**17**
FW	42	RBR	392	VB	**7286**
IN	27835	RBS	240	VBD	**8424**
JJ	16049	RP	95	VBG	**4000**
JJR	1055	S0	47	VBN	**5867**
JJS	451	S1	2134	VBP	**3407**
MD	2637	S2	1809	VBZ	**5561**
NN	36789	S3	351	WDT	**1157**
NNP	24690	S4	358	WP	**639**
NNPS	550	S5	13160	WP$	**39**
NNS	16653	S6	10802	WRB	**571**
PDT	65	S7	1285		
POS	2203	S8	1854		

مدل پیشنهادی برای شبکه CNN، دارای ساختاری ترتیبی و مطابق شکل (۸) است. همان‌گونه که در شکل مشخص است، تعداد کل پارامترهای مدل ۴۱۵۹۷۹ پارامتر است که در طی فرایند یادگیری و به‌صورت برگشت به عقب به‌روزرسانی و تنظیم می‌شوند.

الف) بررسی وقوع پدیده Overfitting در حین فرایند آموزش:

برای شناسایی و جلوگیری از وقوع پدیده overfitting از نمودار دقت به ازای epoch های مختلف و برای مجموعه‌های داده‌ای Train و Validation استفاده کرده‌ایم. شکل (۹) دقت مدل را در Batch های مختلف و به ازای epoch های مختلف نمایش می‌دهد که در آن، شکل اول مربوط به epoch=10 و شکل دوم مربوط به epoch=100 است. محور افقی بیانگر epoch و محور عمودی بیانگر دقت مدل آموزش‌یافته تا آن لحظه و به ازای مجموعه داده‌ای Train(قرمز)، Validation(سیاه) است. همان‌گونه که در تمامی شکل‌ها مشاهده می‌شود، در حین فرایند آموزش، دقت مدل به ازای مجموعه داده‌ای Train و Validation، روند صعودی دارد و پس از یک epoch مشخص، اختلاف دقت دو نمودار افزایش می‌یابد ولی به دلیل روند صعودی نمودار دقت‌ها، می‌توان نتیجه گرفت که سیستم دچار overfitting نشده است و دارای قابلیت افزایش تعداد epoch ها و افزایش دقت نهایی است.



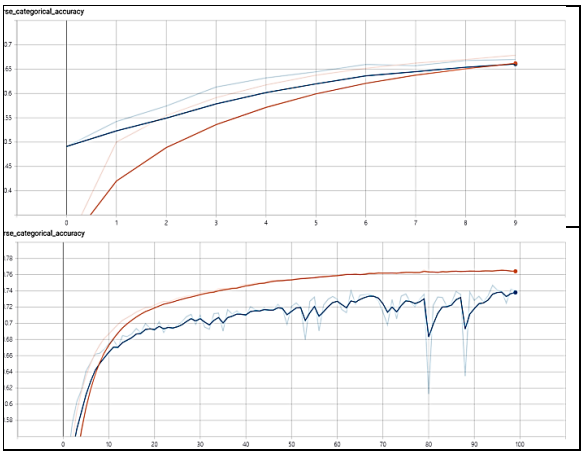
نمودار (۱): مقایسه دقت مدل پیشنهادی به ازای epoch ها و نرخ‌های یادگیری مختلف

Model: "Sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 12, 18, 32)	320
conv2d_1 (Covn2D)	(None, 10, 16, 64)	18496
conv2d_2 (Conv2D)	(None, 8, 14, 128)	73856
Max_pooling2d (MaxPooling2D)	(None, 2, 5, 128)	147584
conv2d_3 (Conv2D)	(None, 1280)	0
flatten (Flatten)	(None, 1280)	0
dense (Dense)	(None, 128)	163968
activation (Activation)	(None, 128)	0
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 64)	8256
activation_1 (Activation)	(None, 64)	0
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
activation_2 (Activation)	(None, 32)	0
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 43)	1419
activation_3 (Activation)	(None, 43)	0

Total params: 415,979
Trainable params: 415,979
Non-trainable params: 0

شکل (۸): ساختار ترتیبی مدل پیشنهادی برای شبکه عصبی عمیق



شکل (۹): نمودار overfitting به ازای مقادیر مختلف پارامتر epoch، در نمودار بالا epoch=10 و در نمودار پایین epoch=100 بوده است

ب) بررسی دقت مدل به ازای مجموعه داده‌ای Test: برای این منظور، دقت نهایی مدل به ازای مجموعه داده‌ای Test و به ازای epoch=[10,50,100] محاسبه‌شده است.

همان‌گونه که در نمودار (۱) مشاهده می‌شود، با افزایش تعداد دفعات آموزش مدل (epoch)، دقت نهایی مدل نیز افزایش می‌یابد و به ازای learning-rate=0.1 و برای epoch های مختلف، دقت مدل همواره بالاتر و بهتر بوده است. با مشاهده نمودار (۱) می‌توان نتیجه گرفت که مدل پیشنهادی به ازای epoch=100 و learning-rate=0.1 بهترین دقت را داشته است و دقت آن برابر است با: accuracy = 80.26%، لذا این مقادیر به‌عنوان بهترین مقادیر برای پارامترهای مدل در نظر گرفته می‌شوند و بعلاوه، مدل پیشنهادی دچار overfitting نشده و قابلیت افزایش تعداد دقعات آموزش و درنتیجه قابلیت بهبود دقت نهایی را دارد.

برای بررسی جزئی‌تر عملکرد مدل پیشنهادی، با در نظر گرفتن بهترین مقادیر برای پارامترها، دقت مدل پیشنهادی به ازای برچسب‌های مختلف را محاسبه کرده‌ایم که در جدول (۲) قابل‌مشاهده است. همان‌گونه که مشاهده می‌شود دقت مدل پیشنهادی به ازای ۲۷.۹٪ از برچسب‌ها، بالاتر از ۹۵٪، به ازای ۹.۳٪ از برچسب‌ها، بالاتر از ۹۰٪، و به ازای ۲۷.۹٪ از برچسب‌ها، برابر صفر بوده است.

با توجه به اینکه تعداد نمونه‌های موجود برای کلاس‌های مختلف، متفاوت است، لذا برای بررسی تأثیر تعداد هر برچسب بر دقت نهایی مربوط به آن برچسب، نمودار (۲) ارائه می‌شوند. در این نمودار تغییرات دقت و تغییرات تعداد نرمال شده برچسب‌ها نمایش داده‌شده است. برای نرمال‌سازی تعداد هر برچسب از رابطه (۲) استفاده کرده‌ایم

جدول (۲): دقت نهایی مدل پیشنهادی به ازای هر برچسب در کل پیکره coNLL2000

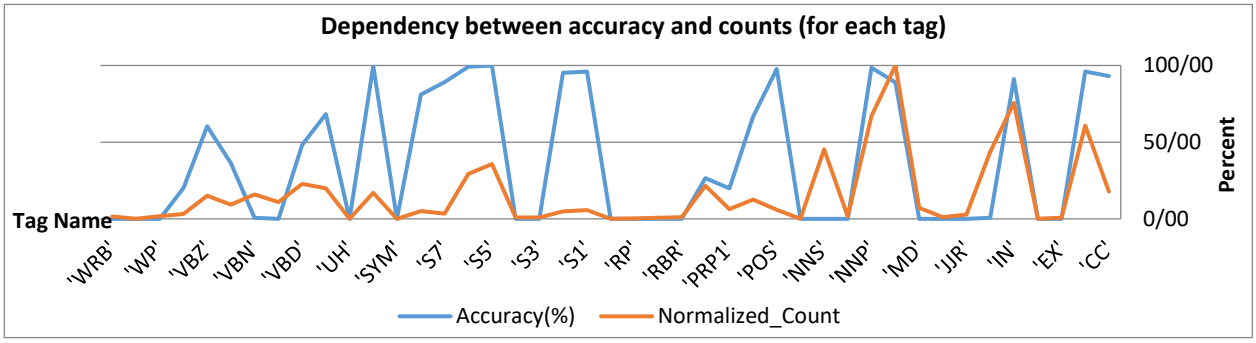
Tag Name	Acc.	Tag Name	Acc.	Tag Name	Acc.
CC	98.4	PRP	94.3	SYM	**0**
DT	98.06	PRP$	92.1	TO	**99.7**
EX	43.56	RB	49.4	UH	**0**
FW	0	RBR	0	VB	**77.6**
IN	97.44	RBS	0	VBD	**62.06**
JJ	46.6	RP	0	VBG	**0**
JJR	49.62	S0	0	VBN	**35.8**
JJS	0	S1	98.25	VBP	**61.34**
MD	89.8	S2	98.5	VBZ	**64.94**
NN	88.7	S3	95.56	WDT	**77.25**
NNP	98.2	S4	72.25	WP	**87.94**
NNPS	0	S5	100	WP$	**0**
NNS	34.03	S6	99.94	WRB	**90.75**
PDT	0	S7	93.44		
POS	96.9	S8	99		

$$Normalize - Count(tag_i) = \frac{Count_{tagi} - \text{Min}(Count)}{Max(Count) - Min(Count)} \quad (۲)$$

که در آن؛ Counttagi بیانگر تعداد کلمات با برچسب i، Min(Count)=17 بیانگر تعداد نمونه‌های کلاسی با کمترین تعداد نمونه است و Max(Count)=36789 که بیانگر تعداد نمونه‌های کلاسی با بیشترین تعداد نمونه است.

با توجه به نمودار (۲) می‌توان نتیجه گرفت که روند تغییرات دقت و روند تغییرات تعداد برچسب‌ها، به یکدیگر شبیه بوده و مدل پیشنهادی تحت تأثیر فراوانی برچسب‌ها قرارگرفته است و به ازای برچسب‌های پرتکرار، دقت مدل نیز افزایش می‌یابد و به ازای برخی از برچسب‌های با فراوانی کمتر نیز، دقت مدل بسیار کم و در حد صفر بوده است که خود عاملی تأثیرگذار در دقت کل مدل پیشنهادی است.

با توجه به اینکه مجموعه داده‌ای، مبتنی بر زبان طبیعی است لذا امکان کاهش تعداد برچسب‌های پرتکرار، جهت بهبود نتایج به ازای برچسب‌های کم تکرار و دستیابی به نتایج نرمال، امکان‌پذیر نبوده و از دقت نهایی مدل کم می‌کند. بنابراین به روش‌های دیگری جهت بهبود نتایج به ازای برچسب‌های کم تکرار نیاز داریم.



نمودار (۲): روند تغییرات دقت مدل پیشنهادی به ازای نمونه‌های هر کلاس و روند تغییرات تعداد نمونه‌ها در کلاس‌های مختلف

ج) مقایسه دقت مدل پیشنهادی با الگوریتم‌های موجود: برای مقایسه دقت روش پیشنهادی با الگوریتم‌های موجود، از الگوریتم‌های کتابخانه استاندارد NLTK [۱۵]و نیز الگوریتم مبتنی بر پرسپترون استفاده‌شده است و نتایج اجرای این الگوریتم‌ها به ازای مجموعه داده‌ای coNLL2000 در جدول (۳) قابل‌مشاهده است. الگوریتم ترکیبی مبتنی بر کلونی زنبورعسل نیز به ازای مجموعه داده‌ای زبان عربی دارای دقت ۹۸/۲ درصد بوده است. همان‌گونه که در جدول (۳) مشاهده می‌شود دقت کل مدل پیشنهادی کمتر از الگوریتم‌های دیگر است و نیاز به افزایش فرایند یادگیری و بهبود نتایج دارد.

جدول (۳): مقایسه دقت مدل پیشنهادی با الگوریتم‌های موجود و به ازای مجموعه‌های داده‌ای مختلف

Algorithm / Model	Dataset	Accuracy
NLTK Tagger	coNLL-2000 (Eng)	85.89%
Perceptron Tagger	coNLL-2000 (Eng)	94.54%
Proposed Model (CNN-based)	coNLL-2000 (Eng)	80.26%
Colony algorithm for POS Taggin[13]	Arabic-Text	98.2%

٦- نتیجه‌گیری نهایی

یکی از مشکلات کار با متون و اسناد، نحوه نگاشت آن‌ها به بردار عددی و استخراج ویژگی‌های عمیق است، به‌گونه‌ای که این بردار، دارای مفهوم بوده و قابل‌استفاده در الگوریتم‌ها و سیستم‌های طبقه‌بندی باشند. در این مقاله یک روش جدید برای پیش‌پردازش و نگاشت کلمات به ماتریس ویژگی‌ها و استخراج ویژگی‌های عمیق ارائه شد. در ادامه اقدام به طراحی و پیاده‌سازی یک مدل شبکه عصبی عمیق مناسب و بهینه نمودیم که قابلیت استخراج ویژگی‌های عمیق از واژگان را دارد.

یکی از نواقص مدل ارائه‌شده، استفاده از تنها یک مجموعه داده‌ای برای آموزش و اعتبارسنجی و آزمایش بوده است و بهتر است مدل پیشنهادی به ازای مجموعه‌های داده‌ای مختلف در زبان انگلیسی و نیز مجموعه‌های داده‌ای مختلف در زبان‌های غیر انگلیسی نیز آموزش و آزمایش شود. عدم به‌کارگیری ویژگی اتمام زودهنگام[1] در شبکه‌های عصبی عمیق نیز از دیگر کاستی‌های مدل ارائه‌شده است که امکان آموزش مدل برای رسیدن به حالت بهینه را امکان‌پذیر می‌سازد. تولید یک ساختار یک‌بعدی به‌عنوان بردار ویژگی برای هر کلمه و ارائه یک شبکه عصبی عمیق مبتنی بر ورودی‌های یک‌بعدی، از دیگر اقداماتی است که می‌توان برای این

[1] Early Stopping

مسأله استفاده کرد، به نظر می‌رسد استفاده از ساختار یک‌بعدی باعث کاهش حجم محاسبات و همگرایی سریع‌تر مدل پیشنهادی به حالت بهینه شود.

همان‌گونه که مشاهده شد مدل پیشنهادی به ازای مجموعه داده‌ای ذکرشده، دارای دقت کلی کمتری نسبت به دیگر الگوریتم‌های موجود بوده است ولی دقت این مدل نسبت به برخی از برچسب‌ها بسیار مناسب بوده است که قابلیت استفاده موردی از مدل را امکان‌پذیر می‌سازد. برای افزایش دقت مدل، پیشنهادی می‌شود تا فرایند آموزش مدل به ازای epoch های بزرگ‌تر ادامه یابد و به‌منظور کاهش تأثیر تعداد برچسب‌ها بر دقت نهایی، پیشنهادی می‌شود که چندین مدل مختلف طراحی و پیاده‌سازی شده و از ترکیب نتایج تولیدشده توسط مدل‌ها برای تولید خروجی نهایی استفاده شود.

بعلاوه بهتر است دقت مدل پیشنهادی به ازای مجموعه‌های داده‌ای مختلف و به ازای زبان‌های مختلف نیز آزمایش‌شده و نتایج آن‌ها مورد تجزیه‌وتحلیل قرار گیرد. به نظر می‌رسد که مدل پیشنهادی، به دلیل دقت بالا و پیچیدگی و حافظه مصرفی کم و با استفاده از یادگیری انتقالی[1]، قابل‌اجرا در محیط‌های واقعی و بر روی دستگاه‌های مختلف باشد.

References

[۱]ن. روح‌الامینی, م. شجاعی‌مهر, ط. فتحی, م. مکی آبادی, " مروری بر روش‌های برچسب‌گذاری اجزای واژگانی کلام برای متن فارسی،" سومین همایش ملی دانش و فناوری مهندسی برق، کامپیوتر و مکانیک ایران، ۱۳۹۸.

[2] D. Jurafsky and J. H. Martin, Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition: United state. Upper Saddle River, NJ: Pearson, 2000.

[3] A. M. Nerabie, M. AlKhatib, S. S. Mathew, M. E. Barachi, and F. Oroumchian, “The impact of Arabic part of speech tagging on sentiment analysis: A new corpus and deep learning approach,” Procedia Comput. Sci., vol. 184, pp. 148–155, 2021.

[4] O. Das and R. Chandra Balabantaray, “Sentiment Analysis of Movie Reviews using POS tags and Term Frequencies,” Int. J. Comput. Appl., vol. 96, no. 25, pp. 36–41, 2014.

[5] Y. Wang, K. Kim, B. Lee, and H. Y. Youn, “Word clustering based on POS feature for efficient twitter sentiment analysis,” Hum.-centric comput. inf. sci., vol. 8, no. 1, 2018.

[6] V. K. Singh, M. Mukherjee, and G. K. Mehta, “Sentiment and mood analysis of weblogs using POS tagging based approach,” in Communications in Computer and Information Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 313–324.

[7] J. Awwalu, S. E.-Y. Abdullahi, and A. E. Evwiekpaefe, “Parts of speech tagging: A review of techniques,” FUDMA JOURNAL OF SCIENCES, vol. 4, no. 2, pp. 712–721, 2020.

[8] B. Pham, “Parts of speech tagging: Rule-based,” p. 1, 2020.

[9] A. Dalal, N. Kumar,S. Uma,S. Sandeep,B. Pushpak, “Building Feature Rich POS Tagger for Morphologically Rich Languages: Experiences in Hindi,” 5th International Conference on Natural Language Processing, p.9, 2007.

[10] W. AlKhwiter and N. Al-Twairesh, “Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM,” Comput. Speech Lang., vol. 65, no. 101138, p. 101138, 2021.

[11] V. Delic, M. Secujski, A. Kupusinac, “Transformation-based part-of-speech tagging for Serbian language, ” Recent Advances Computing Intelligent Systems, 2011.

[12] R. Forsati and M. Shamsfard, “Hybrid PoS-tagging: A cooperation of evolutionary and statistical approaches,” Appl. Math. Model., vol. 38, no. 13, pp. 3193–3211, 2014.

[13] A. Alhasan and A. T. Al-Taani, “POS tagging for Arabic text using bee colony algorithm,” Procedia Comput. Sci., vol. 142, pp. 158–165, 2018.

[14] R. Dhumal Deshmukh and A. Kiwelekar, “Deep learning techniques for part of speech tagging by natural language processing,” in 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020.

[15] W. Wagner, “Steven bird, Ewan Klein and Edward Loper: Natural language processing with python, analyzing text with the natural language toolkit: O’Reilly media, Beijing, 2009, ISBN 978-0-596-51649-9,” Lang. Resour. Eval., vol. 44, no. 4, pp. 421–424, 2010.

[1] Transfer Learning

Journal of Applied And Basic Machine Intelligence Research (ABMIR)

Applied and Basic Machine Intelligence Research

Vol. 1, No. 1, Spring & Summer 2023, 23-33

A New Deep Neural Networks Based Model for Part of Speech Tagging

Fariba Taghinezhad[1], Mohammad Ghasemzadeh [2*]

[1] Ph.D Candidate, Faculty of Computer Engineering, Technical and Engineering Campus, University of Yazd, Yazd, Iran.

[2] Associate Professor, Faculty of Computer Engineering, Technical and Engineering Campus, University of Yazd, Yazd, Iran.

Article Information

Original Research Paper

Received:
25 October 2021

Accepted:
12 March 2022

Keywords:
part of speech tagging, natural language processing, deep neural networks, convolution neural networks.

Corresponding Author*:
m.ghasemzadeh@yazd.ac.ir

Abstract

Part of speech tagging is an important issue in natural language processing and is the base of many other major subjects in this field. In this article, a new method have been introduced for part of speech tagging using deep neural networks. The purpose of this method is solving problems that common other methods are facing with which are extract deep features from texts and classifying these features. The proposed method is based on that we can find deep features and product optimal output by using small deep neural networks. This method was implemented using Tensorflow's specialized libraries and Keras API in python and was evaluated on coNLL2000 standard dataset. The experimental results show that the proposed method is capable to extract high level features from natural language's words and is able to achieve considerable accuracy for repetitive tags. In addition, this method is able to be used in variant environments and on different devices.

: 10.22034/ABMIR.2022.2660