گزارش پایانی پروژه دستیار نگارش فارسی

درس پردازش زبان های طبیعی

مبینا پورنعمت ، پیمان ناصری ، نرگس جاوید ، زینب احیایی، زهرا خرم نژاد

- چکیده

با توجه به توسعه مدل های مبتنی بر توجه نظیر برت برای پردازش زبان های طبیعی،کاربرد این نوع مدل ها در مسایل مختلف پردازش زبان بسیار مورد توجه قرار گرفته است. تا کنون تلاش های بسیار برای توسعه ابزار های پردازشی مختلف برای زبان انگلیسی صورت گرفته است اما درحوزه ی زبان فارسی بسیاری از ابزار ها در ابتدای مسیر توسعه خود قرار دارند و مسیری طولانی برای رسیدن به دقت های مناسب را در پیش دارند. یکی از ابزار های بسیار پرکاربرد در زبان فارسی دستیار نگارش فارسی است. در این پروژه سعی کردیم با استفاده از مدل برت ابزاری برای نگارش زبان فارسی طراحی کنیم تا قابلیت هایی نظیر اصلاح غلط‌های املایی و دستوری مختلف و پیشنهاد کلمات معادل مناسب در جمله را به عنوان یک دستیار نگارش فارسی داشته باشد. در نهایت یک دموی تحت وب برای درک بهتر عملکرد دستیار نگارش پیاده سازی شد.

کلمات کلیدی : دستیار نگارش فارسی، برت، غلط املایی، غلط دستوری

1. مقدمه:

a. دستیار نگارش زبان فارسی

یک دستیار نگارش زبان فارسی ابزاری است که بتواند تمام ایرادات نگارشی متن و دستوری را مشخص کنند و امکانات ویراستاری مختلف و همچنین امکاناتی نظیر تبدیل فرم گفتاری لغات به نوشتاری و پیشنهاد کلمات معادل و سایر کار های پیش پردازشی متن فارسی را به صورت آنلاین در حین تایپ یا باز خوانی متن به نویسنده متن ارایه دهد. واضح است که توسعه تمام ویژگی هایی یک دستیار نگارش فارسی در قالب این پروژه درسی ممکن نیست و این پروژه در راستای تحقق برخی ویژگی های گفته شده، توسعه داده شده است.

b. کار های پیشین

در تلاش های پیشینی که اعضای گروه در راستای توسعه مدلی برای تشخیص غلط های املایی انجام گرفته بود، مدلی پیاده سازی شده بود که میتوانست غلط های املایی مشهود در جملات را تشخیص دهد و غلط هایی نظیر کاربرد کلمات درست در جای اشتباه را نمیتوانست تشخیص دهد.

همچنین ابزار های آنلاینی به تازگی برای ویراستاری متن فارسی در قالب افزونه ی ورد پیاده سازی شده اند نظیر ویراویراست که بیشتر برای ویراستاری متون کاربردی است و تمام قابلیت های مد نظر ما را نظیر تبدیل الگوری گفتاری به نوشتار یا پیشنهاد دهنده کلمات معادل ندارد.

c. روند اصلی پروژه

در این پروژه ابتدا بخش هایی را برای پیاده سازی دستیار نگارش فارسی انتخاب کردیم. در بخش اول که اصلاح ایرادات املایی است سعی شده است با بهبود عملکرد مدل قبلی ایرادات املایی جدیدی نیز توسط مدل تشخیص داده شود. در بخش بعد برخی ایرادات دستوری مصطلح در نگارش زبان فارسی مثل تطابق شناسه فعل با فاعل پیاده سازی شد و سپس ماژولی طراحی شد تا بتواند بر اساس مترادف های ممکن برای کلمات بهترین کلمه معادل هر کلمه را درجمله به کار برده شده تشخیص دهد و به کاربر پیشنهاد دهد. در نهایت هم دموی وب برای نمایش بهتر عملکرد های مختلف دستیار نگارشی پیاده

سازی شد که در بخش بعد نحوه پیاده سازی و مدل ها و دیتا ست هایی که برای هر بخش استفاده شدند با جزییات بیشتر بیان میشود.

2. روش ها
a. دیتاست

یکی از مجموعه داده های استفاده شده، "مجموعه افعال تصریف شده فارسی" از وب سایت پیکره گان است که برای اصلاح ایراد های دستوری به کار گرفته شده است.

b. بخش های اصلی پروژه
i. اصلاح ایراد های دستوری
1. مدل

در زبان فارسی، برای تشخیص مطابقت نهاد با فعل جمله، مدلی از پیش توسعه داده نشده است. بنابراین فرایند طی شده برای این منظور، شامل جمع آوری داده، پیش پردازش های متعدد، به دست آوردن الگوهای متنوع و در نهایت ساخت مدل مورد نظر است. برای تصحیح شخص و شمار فعل، به داده ای حجیم از افعال فارسی به همراه تصریف آن ها نیاز بود که از مجموعه داده ی "مجموعه افعال تصریف شده فارسی" بهره بردیم. همان طور که انتظار میرفت داده ها به همان شکل اولیه قابل استفاده نبودند و مراحل متعددی از پیش پردازش روی آن ها شکل گرفت تا در نهایت دو فایل داده ای مورد نیاز را از روی آن استخراج کردیم. در ادامه نیز به ماژولی نیاز داشتیم که به کمک آن بتوانیم نهاد و فعل را در جملات تشخیص دهیم تا تطابق و یا عدم تطابقشان را بررسی کنیم؛ برای این منظور از مدل Dependency Parser از کتابخانه هضم (hazm) استفاده کردیم. مدل نهایی با در نظر گرفتن الگوهای متعدد در زبان فارسی و با بهره گیری از مدل ها و داده های متنوع جمله ای را ورودی می گیرد و حالت تصحیح شده را خروجی می دهد.

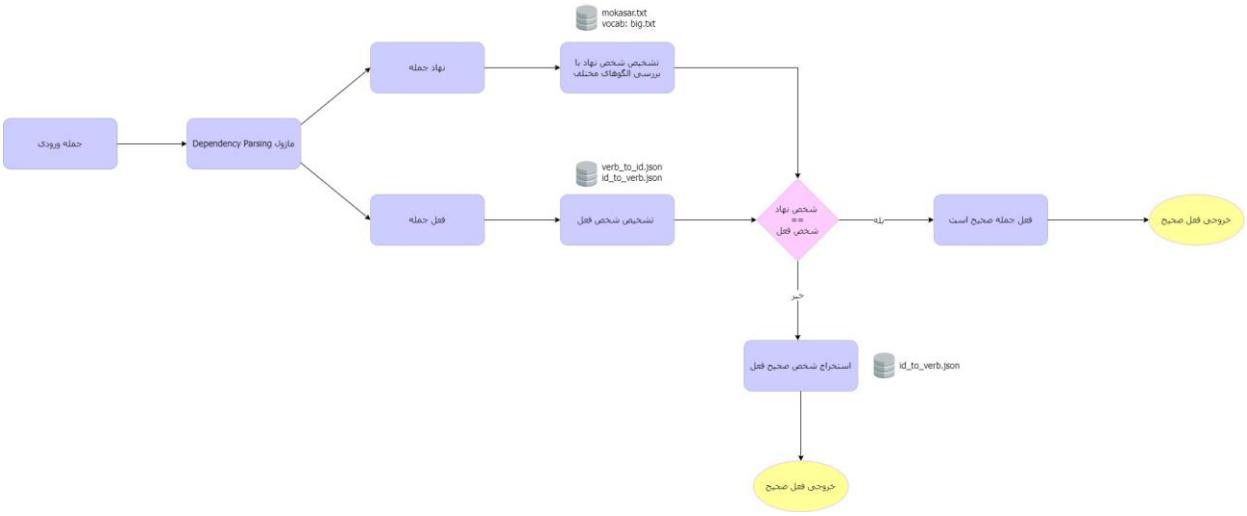
2. نحوه پیاده سازی

همان طور که پیش تر اشاره شد، اولین بخش جمع آوری داده های مورد نیاز است. برای این منظور از روی "مجموعه افعال تصریف شده فارسی" دو فایل مورد نظر را استخراج کردیم. در فایل اول هر فعل به یک اندیس نگاشت شده است و شخص های متفاوت هر فعل نیز به یک اندیس یکسان نگاشت شده اند. در فایل دوم به ازای هر اندیس، تصریف فعل متناظر با آن اندیس، نگهداری شده است. برای تبدیل فعل اولیه به فعل صحیح از این دو فایل استفاده می کنیم. داده ی دیگری که جمع آوری کردیم، مجموعه اسم های جمع مکسر فارسی است که با جستجو در منابع متفاوت، حدود چهارصد مورد جمع آوری شد.

در ادامه برای تشخیص تطابق و یا عدم تطابق نهاد با فعل، باید شخص و شمار نهاد تشخیص داده شود. برای این منظور الگوهای متعدد در ساخت اسم های فارسی مطالعه و استخراج شد که برای هر یک راهکاری ارائه شد. به عنوان مثال کلمات فارسی مختوم به "ات"، "ان"، "ون"، "ین"، "گان"، "یان"، ضمایر متصل، جمع های مکسر و ... . پس از تشخیص شخص نهاد، باید همان شخص از فعل موجود از طریق فایل های ذکر شده استخراج شود.

3. عملکرد

مدل توسعه داده شده روی بازه مناسبی از افعال فارسی و الگو های متعددی از اسامی جمع فارسی و جمع های مکسر عملکرد مناسبی دارد. البته در برخی بخش ها فضا برای توسعه هر چه بهتر وجود دارد که زمان و منابع قابل توجهی نیاز دارد. مانند توسعه ی مدل قوی تری برای تشخیص نهاد، جمع آوری داده های بیشتری از افعال فارسی، توسعه ی راهکار مناسب برای الگوهای پیچیده تر و ...



ii. اصلاح غلط های املایی

1. مدل

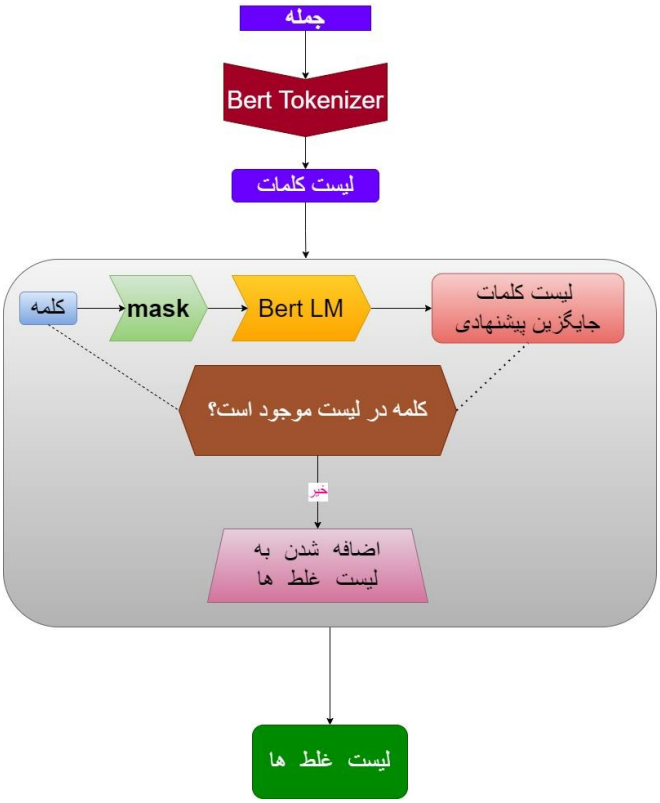
مدل استفاده شده در این بخش مدل فارسBert آزمایشگاه هوشواره است که با استفاده از آن امبدینگ کلمات در جمله استخراج شده است.

2. نحوه پیاده سازی

با کمک تابع find_possible_mistake ما ابتدا غلط بودن یک کلمه را پیدا میکنیم سپس آن را تصحیح میکنیم. این تابع به این شکل عمل می کند که جمله مورد نظر و مدل زبانی مورد استفاده را به عنوان ورودی میگرد سپس به ازای هر کلمه موجود در جمله ورودی آن را mask میکند سپس با کمک مدل زبانی ما به تعداد متغیر top_k محتمل ترین کلماتی را که میتواند جایگزین شود پیدا کند. حال بین این کلمات آن کلمه ای کمترین فاصله Levenshtein را با کلمه ای که mask کردیم پیشنهاد میدهد اگر این کلمه پیشنهادی همان کلمه باشد که خب غلط املایی نداریم در غیر این صورت این کلمه را به عنوان جایگزین پیشنهاد میدهد

3. عملکرد

ابتدا با top_k =1000 و fine tune کردن مدل روی داده های اخبار امتحان کردیم اما تابع دقت چندانی نداشت سپس top_k را به 10000 افزایش دادیم و تابع عملکرد خوبی از خود به جای گذاشت.



iii. پیشنهاد کلمات معادل

1. مدل

مدل استفاده شده در این بخش مدل فارس برت آزمایشگاه هوشواره است که با استفاده از آن امبدینگ کلمات در جمله استخراج شده است.

دیکشنری استفاده شده در این بخش، یک نسخه‌ی دیجیتال فرهنگ جامع واژگان مترادف و متضاد زبان فارسی می‌باشد.

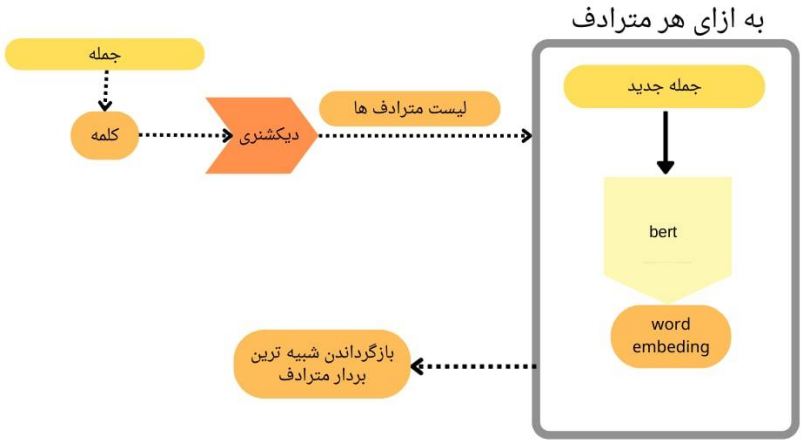
2. نحوه پیاده سازی

در این بخش یک ماژول که با توجه به کاربرد کلمات در جمله کلمه معادل پیشنهاد میدهد، پیاده سازی کردیم. ابتدا سعی داشتیم با استفاده از فارس نت به معادل های معنایی کلمات دسترسی داشته باشیم که api مربوطه قابل اتصال نبود. سپس با جستجو در دیتاست های موجود یک دیتاست مناسب از دیکشنری فارسی پیدا کردیم. سپس دیکشنری را با انجام پیش پردازش‌هایی به فرمتی که میخواستیم به ازای تمام کلمات فقط لیست مترادف‌ها موجود باشد بازنویسی کردیم.

در نهایت نحوه عملکرد این ماژول به این صورت است که ابتدا با دریافت جمله و کلمه مورد توجه کاربر، اگر کلمه در دیکشنری موجود بود، کلمات مترادف پیشنهادی تعیین شده و سپس با استفاده از مدل ترین شده امبدینگ‌های وابسته به سیاق کلمه‌ی اصلی و تمام مترادف‌های پیشنهادی را محاسبه کرده و با استفاده از شباهت کسینوسی تعداد مطلوبی از شبیه‌ترین بردارهای مترادف‌ها به بردار کلمه‌ی اصلی را پیدا کرده و کلمات مترادف متناظر را خروجی می‌دهد.

3. عملکرد

به علت اینکه مدل برت مبتنی بر توجه است و بردار کلمه را با توجه به سیاق و جمله ورودی کاربر استخراج میکند، کلمه ای که در متن ورودی شبیه ترین بردار به بردار اصلی را داشته باشد که بهترین جایگزین در آن سیاق است، اعلام می‌شود.



iv. پیشنهاد کلمه بعدی

1. مدل

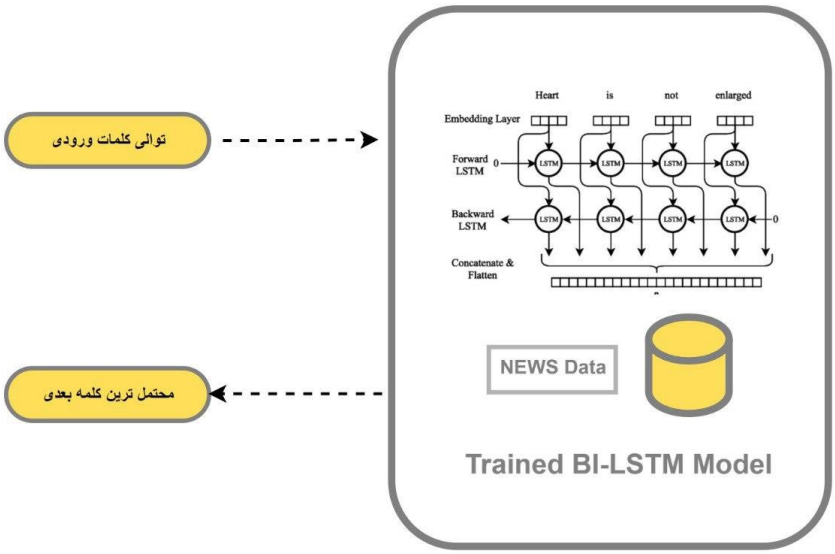
در این بخش ماژولی پیاده سازی شده است که کلمه بعدی را در متن پیشنهاد می کند. این ویژگی به عنوان یک ویژگی اضافه پیاده سازی شده است و جز بخش های اصلی پروژه نمی باشد.

2. نحوه پیاده سازی

در این بخش از یک مدل Bidirectional LSTM استفاده شده است. برای آموزش این مدل از عناوین موجود در داده های اخبار FarsNew، داده های Hidoctor و Donya-e-Eqtesad موجود در گیتهاب درس استفاده شده است.

3. عملکرد

ماژول پیاده سازی شده به گونه ای به دموی وب اضافه شده است که همزمان با ورودی گرفتن از کاربر بعد از هر بار وارد کردن فاصله یا همان space توسط کاربر، ورودی که تاکنون توسط کاربر نوشته شده است به عنوان ورودی به ماژول داده شده و کلمه بعدی پیشنهادی داخل editor برای کاربر نمایش داده می شود. عملکرد و کیفیت کلمات پیشنهادی این مدل وابسته به داده های آموزش و دامنه به کار بردن آن می باشد.



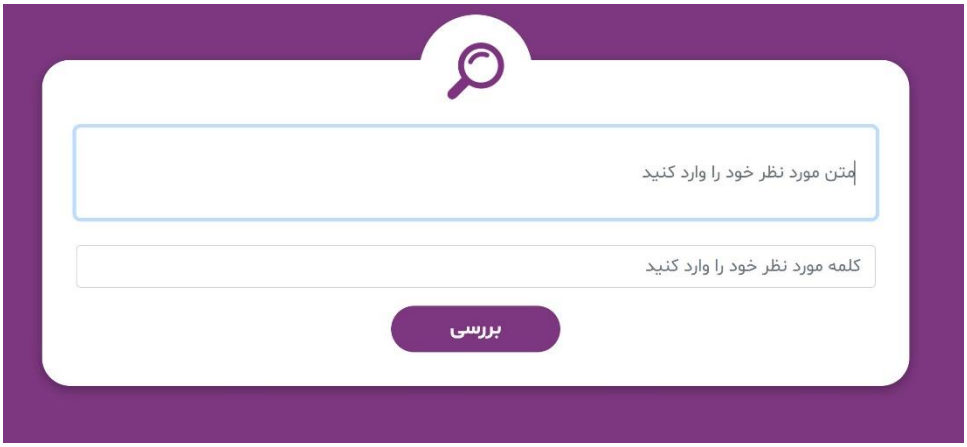
v. دموی وب

برای پیاده سازی دموی وب از چارچوب جنگو استفاده شده است. در صفحه اصلی برنامه می توان هر یک از خدمات پیشنهاد کلمات معادل، اصلاح ایراد های دستوری و اصلاح غلط های املایی را انتخاب کرده و در صفحه مربوطه از خدمات پیاده سازی شده استفاده شود.

1. نحوه راه اندازی

برای راه اندازی کافی است requirement های موجود در فایل requirements.txt را با استفاده از دستور pip install -r requirements.txt نصب کرده و پروژه را با دستور python manage.py runserver اجرا کنید

2. تصاویری از محیط برنامه



c. تقسیم کار

ایرادات املایی: آقای ناصری

ایرادات دستوری: خانم پور نعمت

پیشنهاد کلمات معادل: خانم خرم نژاد و خانم جاوید

پیشنهاد دهنده کلمه بعدی: خانم احیایی

دموی وب: خانم جاوید و خانم احیایی

3. عملکرد نهایی

در این بخش نمونه ای از عملکرد دستیار نگارش را بر روی بستر تحت وب پیاده سازی شده مشاهده میکنید.

- اصلاح غلط املایی

- اصلاح غلط دستوری

- پیشنهاد کلمات معادل

- پیشنهاد کلمه بعدی

4. جمع بندی و کار های آینده

در این پروژه سعی شد بخشی از ایرادات املایی و دستوری نظیر غلط های املایی در اثر تغییر حروف ، غلط های املایی در اثر به کاربردن کلمه درست درجای نامناسب، غلط های دستوری در تطبیق شناسه ها توسط این مدل کشف و تصحیح گردد. همچنین این مدل قابلیت پیشنهاد کلمات معادل با هر کلمه را با توجه به سیاق جمله دارد و نیز در حین تایپ میتواند کلمه بعدی را پیشنهاد دهد.

برای توسعه یک ابزار قدرتمند نگارش زبان فارسی دیتاست جامع و حجیم اهمیت به سزایی دارد که در این پروژه به علت محدودیت منابع امکان استفاده از آنها نبود.

همچنین دایره ی ایرادات نگارشی محدود به این حوزه نمی شود و باید به تدریج ایرادات نگارشی مختلف نظیر تبدیل الگوی محاوره به الگوی نوشتاری و سایر ایرادات املایی و دستوری را به این ابزار اضافه کرد تا بتوان ابزاری جامع و قدرتمند ایجاد کرد.

5. مراجع

[1] peykaregan) .2019 ,January 1 .(Peykaregan .Retrieved August 17 ,2022 ,from https://www.peykaregan.ir/dataset?query=%D8%AA%D8%B5%D8%B1%DB%8C%D9%81&sort_by=changed&sort_order=DESC

[2] Adin ,S) .2021 ,March 14 .نشـــانـه هـای جمع در فـارســی .(utype .Retrieved July 18 ,2022 ,from https://utype.ir/docs/%D9%86%DA%AF%D8%A7%D8%B1%D8%B4-%D9%81%D8%A7%D8%B1%D8%B3%DB%8C/%D8%AF%D8%B3%D8%AA%D9%88%D8%B1-%D8%B2%D8%A8%D8%A7%D9%86-%D9%81%D8%A7%D8%B1%D8%B3%DB%8C/%D9%86%D8%B4%D8%A7%D9%86%D9%87-

%D9%87%D8%A7%DB%8C-%D8%AC%D9%85%D8%B9-%D8%AF%D8%B1-%D9%81%D8%A7%D8%B1%D8%B3%DB%8C/

[3] sobhe) .2020 ,January 1 .(GitHub - sobhe/hazm :Python library for digesting Persian text .Hazm . Retrieved July 15 ,2022 ,from https://github.com/sobhe/hazm

[4] Shirzadeh ,F) .2010 ,January 1 .(Tebyan .Retrieved July 18 ,2022 ,from https://article.tebyan.net/258615/%D8%A7%D8%B3%D9%85-%D8%AC%D9%85%D8%B9

[5] peykaregan) .2018 ,January 1 .پیکره گان | پایگاه انتشـار و تولید داده‌های زبانی .(Retrieved July 18 ,2022 , from https://www.peykaregan.ir/dataset/%D9%81%D8%B1%D9%87%D9%86%DA%AF-%D8%AC%D8%A7%D9%85%D8%B9-%D9%88%D8%A7%DA%98%DA%AF%D8%A7%D9%86-%D9%85%D8%AA%D8%B1%D8%A7%D8%AF%D9%81-%D9%88-%D9%85%D8%AA%D8%B6%D8%A7%D8%AF-%D9%81%D8%A7%D8%B1%D8%B3%DB%8C

[6] Zhou ,W., Ge ,T., Xu ,K., Wei ,F .and Zhou ,M., 2019 ,July .BERT-based lexical substitution .In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp .3368-3373).