بسم اللّه الرحمن الرحیم

تمرین سری اول

تشخیص و اصلاح فاصله‌گذاری در متون فارسی

پردازش زبان‌های طبیعی

بهار ۱۴۰۱

فهرست

تشخیص و اصلاح فاصله‌گذاری در متون فارسی

تکلیف سری اول

شناسنامه

تمرین سری اول	
۱۴۰۱–۱۴۰۰	دانشگاه صنعتی شریف – دانشکده کامپیوتر
مشخصات درس	
عنوان درس	پردازش زبان‌های طبیعی
استاد مربوطه	آقای دکتر احسان‌الدین عسگری
استاد حل تمرین	تیم اساتید حل تمرین
مشخصات گروه	
نام و نام خانوادگی	امیر پورمند پویا خانی مهدی آخی
مشخصات نویسنده و ارسال کننده گزارش	مهدی آخی – ۹۹۲۰۱۴۹۸
نشانی الکترونیکی	mahdiakhi@ce.sharif.edu
مشخصات سند	
عنوان	گزارش تمرین سری اول
تاریخ تحویل	۱۴۰۱/۱/۲۵

تعریف مسئله

از اصلی‌ترین ارکان هر زبان طبیعی‌ای اصول و قواعد نگارشی و ویراستاری آن هستند. زبان زیبای فارسی نیز از این قضیه مستثنی نیست. در زبان فارسی با توجه به شکل حروف الفبای آن که کلمات از به هم چسبیدن حروف به یک دیگر به وجود می‌آیند، فاصله‌گذاری نقش مهمی در فهم مطالب فارسی و درست نویسی آن دارد. در این بخش از تکالیف سعی شده تا یک ابزار برای بررسی و تصحیح اصول فاصله‌گذاری زبان فارسی توسعه داده شود. در ادامه گزارشی از مراحل انجام شده و نیز توضیحی از آنچه که در بخش فنی انجام شده تقدیم می‌گردد. بخش فنی کار به کمک زبان پایتون و بدون استفاده از هیچ گونه ابزار خارجی توسعه داده شده. از آنجا که قصد داشتیم این ابزار را به صورت متن‌باز برروی GitHub و به عنوان یک کتابخانه پایتون در اختیار عموم قرار دهیم، امکان توسعه آن در قالب یک نوت‌بوک وجود نداشت و معماری آن به صورت یک اپلیکیشن پایتون می‌باشد.

بررسی دستور زبان

برای دقت و سرعت کار کلمات را در قالب سه دسته اصلی «فعل»، «اسم» و «صفت» مورد بررسی قرار دادیم. سعی کردیم تا حد ممکن به کامل‌ترین شکل ممکن تمام این سه دسته را بررسی کنیم تا کمترین حالتی جا نماند.

افعال

برای پوشش نیم‌فاصله در افعال بعد از مطالعه منابع گوناگون دستور زبان فارسی به این نتیجه رسیدیم که به دلیل تنوع و گستردگی حالت‌ها، بهترین حالت ممکن با توجه به امکانات کنونی استفاده از یک لیست از افعال می‌باشد. به همین خاطر لیست با حدود بیش از ۶۸۰هزار فعل فارسی در اشکال(از نظر زمان فعل، شمار، مصدر و...) متفاوت تهیه کردیم.

از جمله پیچیدگی‌هایی که در بخش افعال با آنها مواجه شدیم می‌توان به این مورد اشاره کرد که اگر چندین فعل پشت سر هم بیاید، باید بین تمام آنها نیم‌فاصله قرار بگیرد که این مورد در ابزار ما مورد توجه قرار گرفته. به عنوان مثال:

داشتم می‌رفتم دیدمت ← داشتم‌می‌رفتم‌دیدمت

همچنین تمامی پیشوندهایی که با افعال می‌آیند نیز در این ابزار در نظر گرفته شده است. پیشوندهایی مانند: می ،بر، خواه، داشت و... . به علاوه این که این ابزار قادر است تا افعال به هم چسبیده را نیز تشخیص دهد و آنها را تصحیح کند. در تصاویر زیر نمونه‌ای از ورودی و خروجی این بخش را مشاهده می‌کنید:

اسامی و صفت‌ها

اسامی و صفت‌ها نیز حروف اضافه مختص خود را دارند با این تفاوت که مانند افعال حالات محدود(هرچند زیاد!) ندارند. به همین خاطر چالش این بخش بیشتر به مطالعه و استخراج قوانین و استثناهای حروف اضافه این بخش اختصاص داشت. بعد از بررسی که در این بخش انجام دادیم، قوانین مربوط به بیش از ۸۵ پسوند و بیش از ۱۵ پیشوند که مخصوص اسامی و صفت‌ها بودند استخراج شد. متاسفانه بسیاری از این حروف اضافه دارای استثناهایی بودند که بررسی آنها زمان زیادی را می‌طلبید. به عنوان مثال پسوند قیدساز «وار» همیشه به صورت چسبیده می‌باشد(مثل: سوگوار) مگر درمواردی که کلمه به «ه» یا «ی» ختم شود، در این صورت باید با نیم‌فاصله باشد مثل: طوطی‌وار، دیوانه‌وار. لیست کلمات پیشوندی و پسوندی در جدول زیر آمده است. به دلیلی تعداد زیاد گستردگی قوانین مرتبط، از ذکر آنها اجتناب می‌کنیم اما همه قوانین در فایلی در سورس کد برنامه وجود دارند.

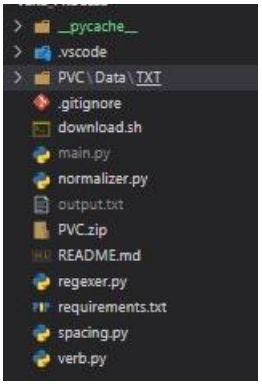
وند	لیست کلمات
پیشوند	با، بی، بیش، پاد، پسا، پیش، فرا، فرو، نا، هم، پر، کم، بد، پیش، این، آن
پسوند	طلبان، طلب، گرایی، گرایان، شناس، شناسی، گذاری، گذار، گذاران، شناسان، گیری، پذیری، بندی، آوری، سازی، بندی، کننده، کنندگان، گیری، پرداز، پردازی،ی، پردازان، آمیز، سنجی، ریزی، داری، دهنده، آمیز، پذیری، پذیر، پذیران، گر، ریز، آسا، آگین، زار، ریزی، فام، رسانی، یاب، یابی، گانه، گانه‌ای، انگاری، گا، بند، رسانی، دهندگان، هایی، ها، های، ایی، هایم، هایت، هایش، هایمان، هایتان، هایشان، گین، مند، ین، ینه، بان، دان، سار، ستان، سرا، دار، کده، گار، گان، گاه، گر، ناک، لاخ، مان، انه، گری، گانی، گون، وار، واره، واری، وانه، ور، وش، وند، دوز، دوزی

دیگر کلمات

غیر از سه دسته اصلی که گفته شد کلمات دیگری نیز هستند که نیاز به بررسی و اصلاح دارند. مثلا کلماتی مانند «همین»، «همان»، «طور»، «هیچ» و... برای این دسته از کلمات نیز به صورت موردی و مشخص قوانین استخراج شده‌اند(برای تعدادی که پیدا شده‌اند).

گزارش فنی

معماری توسعه‌ی این ابزار به صورت ماژولار می‌باشد تا درصورتی که نیاز به توسعه وجود داشت محدودیتی برای آن وجود نداشته باشد. در تصویر ۱ ساختار دایرکتوری پروژه آورده شده.



تصویر ۱- ساختار فایل‌های پروژه

کلاس‌ها

هر کدام از بخش‌های پروژه در قالب یک کلاس توسعه داده شده که در ادامه به شرح آنها می‌پردازیم. به عنوان اولین کلاس باید به سراغ کلاس Normalizer برویم. وظیفه این کلاس تصحیح مواردی کلی و به نوعی انجام پیش‌پردازش برروی متن ورودی است. این کلاس شامل دو متد به نام‌های «characterRefine» و «punctuationRefine» است که وظایف زیر را برعهده دارند:

- تمیز کردن فاصله‌های اضافی در متن(تبدیل چند فاصله، خط بعد و نیم‌فاصله به یکی)
- رعایت قوانین فاصله‌گذاری در خصوص علائم نگارشی(punctuation)
- تغییر اعداد و علائم انگلیسی به نوشتار فارسی
- تغییر کوتیشن به گیومه
- حذف اعراب‌ها(ً ٌ ٍ َ ُ ِ ّ ْ)
- تمیز کردن حروف کشیده و موارد زیبایی(تبدیل «ســـــلام» به «سلام»)
- تغییر یا و کاف عربی به «ی» و «ک» فارسی

کلاس بعدی کلاس Regexer است. وظیفه این کلاس واکشی قوانین از فایل‌های قوانین، تبدیل آنها به عبارات منظم و کامپایل آنها است. هر تابع داخل این کلاس برای یک دسته از قوانین توسعه داده شده است. مثلا تابع suffixGenerator برای انجام عملیات‌های ذکر شده برروی کلمات پسوندی می‌باشد. این تابع بعد از خواندن قوانین از فایل آنها را در قالب یک آرایه پردازش کرده و براساس نوع قوانین، عبارات منظم را تولید می‌کند. سپس قوانین را برای تابع compilePatterns ارسال می‌کند تا آنها را کامپایل کند و خروجی آن عبارات منظم کامپایل شده است.

کلاس verb برای پردازش افعال نوشته شده. این کلاس تنها یک تابع دارد. وظیفه این تابع خواندن تمام افعال جمع‌آوری شده و جست و جوی آنها در متن داده شده است. بدنه این تابع شامل تمام عبارات منظم مورد نیاز برای استخراج حالات مختلف افعال است.

در نهایت کلاس Spacing که وظیفه‌ی بررسی تمام حروف اضافه‌ی غیر از افعال را برعهده دارد. بعد از این که افعال متن تصحیح شدند، متن به این کلاس ارسال می‌شود تا دیگر حالات را مورد بررسی قرار دهد. توابع داخل این کلاس براساس این که کلمات پیشوندی، پسوندی و یا هیچ کدام هستند دسته‌بندی شده‌اند. وظیفه این توابع دریافت الگوها از کلاس Regexer و جست و جوی آنها داخل متن است. در آخر نیز الگوهای پیدا شده را با الگوهای درست جایگزین می‌کنند.

فایل‌های الگو

تمام قوانین استخراج شده در فایل‌های الگو قرار دارند. قالب محتوایی فایل‌های suffix.txt و prefix.txt به شکل زیر است:

Token, affix, exceptions, syllabus

نشانه	توضیح
Token	کلمه مورد نظر. مثل: گذر، بان، دار و...
affix	نحوه اتصال در حالت کلی. که سه مقدار دارد. h: برای نیم‌فاصله، a: برای حالت چسبیده، s: برای حالت با فاصله
exceptions	برای حالت‌های خاص کلمات. مثلا «وار» همیشه به کلمه قبلی چسبیده مگر اینکه کلمه به ه یا ی ختم شود. در این صورت مقدار این قسمت در فایل برابر است با: ی|ه

به عنوان مثال این سه‌تایی مرتب برای کلمه «آمیز» به شکل زیر است:

آمیز,h,,0

فایل ALL_VERBS.TXT نیز شامل بیش از ۶۸۰ هزار فعل به همراه نوع آنها می‌باشد. متاسفانه به دلیل در دسترس نبود ساده‌ترین ابزار کار با زبان فارسی مانند POS TAGGER باعث شد که در مواردی کیفیت کار کاهش یابد و نتوان تمام حالت‌ها را پوشش داد. مواردی مانند کلمات: ساز، بر، بد و... که برای تعیین نقش آنها در جمله نیاز به داشتن چنین ابزارهایی بود منتها ابزارهای موجود با خطاهای بسیار بالا این کار انجام می‌دادند که ما را از استفاده از آنها منصرف کردند.

از آنجا که این سیتم به صورت کاملا منعطف نوشته شده و براساس قوانین کار می‌کند، اگر حتی مواردی نیز در این ابزار در نظر گرفته نشده باشند، صرفا کافی است که قانون آن را به شکل سه‌تایی بالا که توضیح داده شده در فایل مربوطه وارد کنید تا آن کلمه نیز در پردازش‌ها لحاظ شوند. تمرکز اصلی ما نیز همین بود، زیرا جمع‌آوری تمام عبارات ممکن کار تقریبا غیرممکنی بود در یک مدت محدود، در نتیجه تصمیم گرفتیم تا سیستم را با این معماری توسعه دهیم تا به مرور زمان و توسط هر شخصی قابل تکمیل کردن باشد.

قصد داشتیم که این ابزار را به صورت یک وبسایت بالا بیاوریم که در دسترس عموم قرار داشته باشد که متاسفانه به دلیل کمبود وقت این امکان فراهم نشد. اما برای تحویل نهایی از وبسایت استفاده می‌کنیم.

برنامه آینده

هدف ما این است که این ابزار را به صورت یک کتابخانه متن‌باز در دسترس عموم قرار دهیم. در حال تکمیل کردن برخی قابلیت‌های مورد انتظار و البته برطرف کردن برخی کاستی‌ها هستیم تا به یک نسخه پایدار رسیده و آن را منتشر کنیم.

همچنین قصد داریم که یک نسخه تحت وب از آن را برای کلاس آماده کنیم تا امکان دسترسی و استفاده از آن در شرایط مختلف مهیا باشد. امیدواریم که این ابزار گام کوچکی در بهبود ابزارهای کار با زبان فارسی در دنیای پردازش زبان طبیعی باشد.

تشخیص و اصلاح فاصله‌گذاری در متون فارسی

تکلیف سری اول