تمرین دوم درس پردازش زبان‌های طبیعی

استاد درس: دکتر حمیدرضا برادران کاشانی

دستیاران آموزشی: آیین کوپایی - هاجر مظاهری

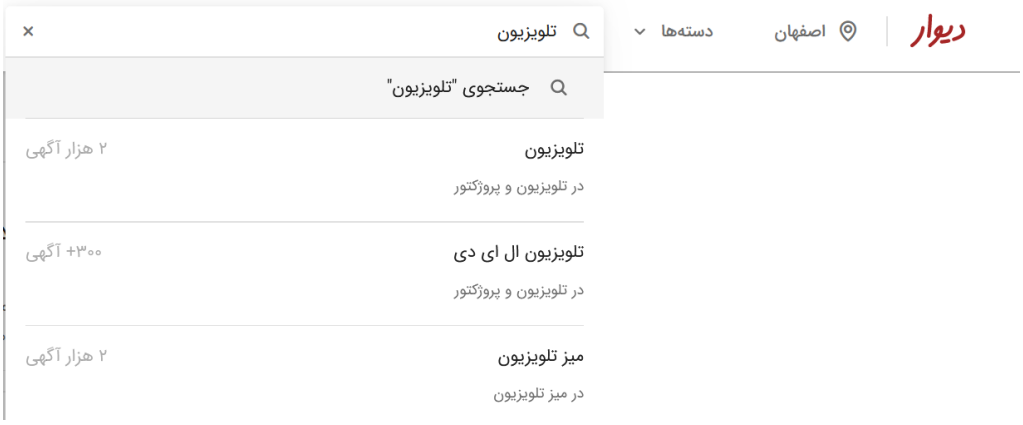
تاریخ بارگذاری تمرین: ۱۴۰۳/۰۱/۲۲

تاریخ تحویل تمرین: ۱۴۰۳/۰۲/۰۵

دانشگاه اصفهان

تکمیل خودکار

هدف از تمرین حاضر ساخت یک سیستم تکمیل خودکار است. سیستم‌های تکمیل خودکار به طور معمول در موتورهای جستجو، ویرایشگرهای متن، برنامه‌های پیام‌رسان و ... برای بهبود تجربه کاربری استفاده می‌شوند. این سیستم‌ها معمولا با تجزیه و تحلیل متن ورودی، لیستی از کلمات احتمالی بعدی را پیشنهاد می‌دهند. وقتی عبارتی را در گوگل جستجو می‌کنید، اغلب پیشنهادهایی برای کمک به شما برای تکمیل جستجو نمایش داده خواهد شد. هنگامی که در حال نوشتن ایمیل هستید، پیشنهاداتی دریافت می‌کنید که پایان‌های احتمالی جمله را به شما می‌گوید. در پایان این تمرین، شما یک نمونه اولیه از چنین سیستمی را توسعه خواهید داد. تصویر زیر گویای یک سیستم تکمیل خودکار است. در این تمرین بخشی از مجموعه داده سایت دیوار برای مدل سازی زبانی و ساخت سیستم تکمیل خودکار استفاده می‌شود.



۱- پیش پردازش

مجموعه داده مورد نظر در فایل divar_posts_dataset.csv موجود است. دیتاست سایت دیوار حاوی اطلاعاتی در خصوص آگهی های ثبت شده در این سرویس می باشد. با توجه به اینکه مجموعه داده مورد استفاده به زبان فارسی است، می‌توانید از کتابخانه hazm استفاده کنید.

- هر آگهی را به جملات آن تجزیه کنید.
- برای هر جمله علائم نگارشی، فضاهای خالی اضافه و ... را حذف کنید به طوری که در انتها فقط اعداد و کلمات را داشته باشید.

- هر گونه پیش پردازش دیگری که بتواند نتایج را بهبود دهد و مطابق با مسائل مدلسازی زبانی باشد انجام داده و توضیح دهید که چرا این مراحل پیش پردازش انتخاب شده است.

۲- ساخت مدل زبانی

برای مجموعه داده پیش پردازش شده مراحل زیر را انجام دهید:

۲-۱- در این قسمت یک مدل زبانی n-gram را پیاده سازی کنید که به n اجازه می‌دهد از یک تا سه تغییر کند. در واقع این تابع باید لیستی از unigram ، bigram و trigram ها را از درون مجموعه داده استخراج کند و تعداد تکرار هرn-gram را محاسبه کند. این لیست ها را نمایش دهید و سپس ۸ تا از پر تکرار ترین unigram وbigram و trigram ها را گزارش دهید.

۲-۲- توضیح دهید دلیل هموارسازی در محاسبه احتمالات n-gram ها چیست و سپس Laplace smoothing و Good-Turing smoothing را توضیح دهید.

۲-۳- تابعی برای محاسبه احتمال n-gram ها بنویسید. درمحاسبه احتمالات از Laplace smoothing برای unigram ها و از Good-Turing smoothing برای دیگر n-gram ها استفاده کنید.

۲-۴- تابعی برای محاسبه perplexity هریک مدل‌های ایجاد شده (unigram, bigram, trigram) بنویسید. سپس perplexity مدل‌ها را برای جملات زیر محاسبه کرده و گزارش دهید.

#	جمله	Unigram Perplexity	Bigram Perplexity	Trigram Perplexity
۱	گوشی بسیار بسیار تمیز و فقط سه هفته کارکرده و در حد آک			
۲	دوعدد پیراهن دخترونه مارک درحد نو مناسب تا یکسال ونیم			
۳	کفش طبی چرم مصنوعی مارک لمون طب که نو میباشد			
۴	دوربین عکاسی خانگی کنان سالم درحد نو			
۵	گل مصنوعی کاملا سالم بدون ایراد همراه با گلدان			

۲-۴- پیش‌بینی کلمات

در این مرحله می‌خواهیم با استفاده از مدل‌های ایجاد شده، کلمات جدید را با استفاده از دنباله‌ای از کلمات ورودی پیش‌بینی کنیم. برای این کار تابعی طراحی کنید که مدل و دنباله ای از کلمات را به عنوان ورودی دریافت کند و کلمات بعدی را به عنوان خروجی برگرداند و جمله را تا رسیدن به طول ۱۰ تکمیل کند. در واقع جمله خروجی این تابع باید دارای طولی به اندازه ۱۰ باشد که شامل کلمات ورودی و کلمات پیش بینی شده است. با استفاده از این تابع، عبارات زیر را تکمیل کنید. (در نهایت شما باید ۱۲ جمله داشته باشید، به ازای هر مدل ۴ جمله.)

#	عبارت	Predicted sentence by Unigram Model	Predicted sentence by Bigram Model	Predicted sentence by Trigram Model
۱	مبل هفت نفره خود رنگ			
۲	دستگاه تردمیل نو			
۳	کفش مردانه			
۴	تعدادی وسایل اداری و پزشکی			

۲-۵- perplexity مدل ها را برای جملات ساخته شده در مرحله قبل بدست آورده و گزارش دهید.

۲-۶- توضیح دهید در استفاده از مدل زبانی n-gram، چه عواملی در انتخاب مقدار n موثر است.

۳- برچسب گذاری کلمات

قسمت اول: برنامه نویسی

۳-۱- با استفاده از hazm عمل POS Tagging را روی مجموعه داده پیش پردازش شده اعمال کنید و تگ هر توکن را در خروجی نمایش دهید.

۳-۲- تعداد رخدادهای هر تگ POS را در کل مجموعه داده به دست آورید و گزارش دهید.

۳-۳- اسم‌ها را از جملات دارای تگ POS استخراج کنید و ۱۵ اسم پر تکرار اول را همراه با تعداد تکرار آن‌ها گزارش دهید.

قسمت دوم: رویکرد عملی بدون نیاز به برنامه نویسی

۳-۴- در این بخش می‌خواهیم با استفاده از روش HMM و بر اساس جملات زیر، عمل POS Tagging را بر روی جمله "Can Tom mark watch?" انجام دهیم. (راهنمایی) فرض کنید که سه برچسب verb، noun و modal داشته باشیم. مراحل زیر را به ترتیب انجام دهید.

Mark can watch

Will can mark watch

Can Tom watch?

Tom will mark watch

آ. عمل POS Tagging را بر روی جملات داده شده انجام دهید و جدولی بسازید که نشان دهنده احتمال هر کلمه بر اساس سه برچسب verb، noun و modal باشد. برای مثال در جدل زیر احتمال کلمه Tom نشان داده شده است.

کلمات	noun	modal	verb
Tom	2/6	0	0

ب. دو تگ شروع و پایان جمله را به جملات داده شده اضافه کنید. برای شروع جمله از تگ <S> و برای پایان جمله از تگ <E> استفاده کنید. برای مثال:

<S> Mark can watch <E>

سپس جدولی رسم کنید که احتمال وقوع همزمان هر دو برچسب را نشان دهد. جدول شما باید مطابق جدول زیر باشد.

	Noun	Modal	Verb	<E>
<S>				
Noun				
Modal				
Verb				

پ. بر اساس احتمالات محاسبه شده عمل POS Tagging را برای جمله زیر انجام دهید. (گرافی از جمله رسم کنید، سپس یال هایی که احتمال صفر دارند را حذف کنید تا به جواب برسید.)

Can Tom mark watch?

نکات تحویل

۱- پاسخ خود را در پوشه ای به اسم NLP_NAME_FAMILY_HW2 و در قالب zip بارگذاری نمایید.

۲- این پوشه باید حاوی موارد زیر باشد:

- کد نوشته شده در قالب یک فایل jupyter notebook
- فایل گزارش فنی در قالب یک فایل PDF

۳- لازم به ذکر است که رعایت قوانین نگارشی حائز اهمیت است.