به نام خدا

گزارش تمرین دوم درس پردازش زبان طبیعی

استاد درس:

جناب دکتر برادران

نام و نام خانوادگی دانشجو:

امیررضا صدیقین

شماره دانشجویی:

۹۹۳۶۱۴۰۲۴

بخش ۱:

در این بخش فایل‌ها خوانده شدند. برای خواندن فایل csv از کتابخانه‌ی pandas استفاده شده است.

بخش ۲:

بخش a :

همانطور که در فایل مربوط به کدها استفاده شده است. برای استفاده از PunktSentenceTokenizer باید از آن شی ساخته شود و بعد تابع tokenize برای متن خواسته شده صدا زده شود. برای استفاده از sent_tokenize باید آن را برای متن خواسته شده صدا زد.

بخش b :

متون گفته شده با هر یک از روش‌های گفته شده به جملات آن تقسیم شده است و تعداد جملات آن بدست آمده است.

بخش ۳ :

همانظور که در فایل مربوط به کدها نشان داده شده است نمونه‌های آموزشی گفته شده load و با این روش سه متن گفته شده به جملاتشان تقسیم شدند و تعداد جملات آن نمایش داده شده است.

بخش ۴ :

برای متن SampleEnglish این روش‌ها، جمله بندی متفاوتی دارند. برای این متن در روش sent_tokenize ۳ جمله، در روش PunktSentenceTokenize ۴ جمله و در روش یادگرفته شده ۳ جمله تشخیص داده شده است.

دلیل تفاوت آن نیز آن است که روش PunktSentenceTokenize حساس به نقطه است (البته در شرایطی اگر نقطه ببیند جمله حساب می‌کند مثلا کلمه‌ی آخر دارای معنای مستقل باشد) و چون در این متن از نقاط برای جداسازی القاب و اسامی از هم استفاده شده است آن‌ها را جملات جدا فرض کرده است. ولی در روش‌های دیگر به درستی تشخیص داده شده است.

بخش ۵ :

خیر به درستی تشخیص داده نشده است و راه حل پیشنهادی من آن است که نقاط جمله را با \n (به منظور آن که هر جمله در یک خط باشد) و نقطه ی مربوط به آن جایگزین می‌کنیم. بعد از جایگزینی جملات به درستی جایگزین شده اند.

بخش ۶ :

متن ترکی به وسیله‌ی روش آموزش داده شده، به جملاتش تجزیه شد و تعداد آن پیدا شد.

بخش ۷:

ابتدا webtext در NLTK دانلود شد و سپس فایل overheard.txt خوانده شد.

برای آموزش PunktSentenceTokenize کافی است متن مورد نظر را به عنوان ورودی تابع سازنده‌ی آن دهیم. سپس با استفاده از این روش آموزش داده شده و روش sent_tokenize متن باز شده را به جملاتش تقسیم کردیم.

تعداد جملات در روش آموزش دیده بیشتر از روش sent_tokenize است به دلیل آن که روش آموزش دیده به درستی متوجه میشود که در یک دیالوگ آیا چند تا جمله موجود است یا یکی. ولی در روش sent_tokenize در بیشتر موارد متون در یک دیالوگ را با هم یک جمله در نظر می‌گیرد.

بخش ۸ :

بخش a:

فایل csv داده شده خوانده شده است و ستون text آن جدا شده است

بخش b :

هر متن به جملاتش با استفاده از روش sent_tokenize تقسیم شده است.

بخش c:

هر جمله به کلماتش تقسیم شده است. برای حذف علائم نگارشی از RegexpTokenizer با عبارت منظم \w+ استفاده شده است.

بخش d :

کلمات stopwords دانلود شد و در لیستی به همین عنوان قرار گرفت سپس کلمات به دست آمده در بخش قبل مقایسه شد و آن‌هایی که جز کلمات stopwords بوده اند را حذف شد.(برای تطبیق درست باید ابتدا کلمات به شکل lowercase در بیایند و بعد با کلمات stopwords مقایسه شوند.)

بخش e :

متن ۱۰ ام مجموعه قبل از پیش پردازش و بعد از پیش پردازش به نمایش درامده است و تعداد آن‌ها نیز نمایش داده شده است.

همانطور که معلوم است تعداد زیادی کلمه حذف شده است و این امر پردازش ما را راحتتر می‌کند و کلمات مانده داراری اهمیت پردازشی زیاد هستند و ما را از اهداف اصلی دور نمی‌کنند.