به نام خدا

گزارش تمرین اول درس پردازش زبان طبیعی

استاد درس:

جناب دکتر برادران

نام و نام خانوادگی دانشجو:

امیررضا صدیقین

شماره دانشجویی:

۹۹۳۶۱۴۰۲۴

بخش ۱:

ابتدا پکیج‌های مورد نیاز نصب شده است و فایل‌های مورد نظر خوانده و متون آن‌ها استخراج شده است.

بخش ۲:

در این بخش به ازای تمام متون مورد ارزیابی، توکن‌ها وتایپ‌ها و تعداد آن‌ها به وسیله‌ی ۴ روش tokenize کردن گفته شده در صورت سوال، به دست آمده است.

نکته : تایپ‌ها همان مجموعه‌ی غیر تکراری توکن‌ها است، برای همین از set استفاده شده است.

بخش ۳:

با استفاده از TreebankWordTokenizer توکن‌ها و تایپ‌های متون داده شده استخراج شده است. این بخش در بخش ۲ قبلا انجام شده است.

بخش ۴:

با استفاده از RegexpTokenizer کلمات مربوط به متن کوتاه انگلیسی و متن کوتاه فارسی و اعداد مربوط به متن نمونه انگلیسی را استخراج شده است.

نکته : برای پیدا کردن اعداد از عبارت منظم +[1-9] استفاده شده است.(رشته‌هایی که کاراکترهای آن‌ها اعداد باشد).

بخش ۵:

با استفاده از WhitespaceTokenizer کلمات متون استخراج شده است که کد مربوط به این قسمت در بخش ۲ آمده است.

نکته: WhitespaceTokenizer را میتوان با RegexpTokenizer نوشت که عبارت منظم آن +\s است.(در آن تطبیق فواصل مثل enter، space، tab و غیره بررسی می‌شود.)

بخش ۶:

با استفاده از WordPunctTokenizer کلمات متون استخراج شده است که کد مربوط به این قسمت در بخش ۲ آمده است.

نکته: WordPunctTokenizer را میتوان با RegexpTokenizer نوشت که عبارت منظم آن \w+|[^\w\s]+ است.(در آن هم کلمات و هم علائم نگارشی به عنوان توکن در نظر گرفته می‌شود و در آن هم بر اساس تطبیق فواصل هم تطبیق کلمات استفاده شده است.)

بخش ۷:

با استفاده از word_tokenize، regexp_tokenize و wordpunct_tokenize عملیات tokenize روی متن کوتاه انگلیسی انجام شد.

تابع word_tokenize بهبود یافته‌ی کلاس TreebankWordTokenizer است که در آن‌ها می‌توان نوع زبان را نیز انتخاب کرد و بر اساس قواعد آن زبان عملیات tokenize انجام شود. برای زبان‌های خاص با قواعد خاص بهتر است از این تابع به جای نوع کلاسی آن استفاده شود.

تابع regexp_tokenize مثل کلاس RegexpTokenizer عمل می‌کند که ورودی‌های آن در ورودی init کلاس استفاده می‌شود.

تابع wordpunct_tokenize نیز مثل کلاس WordPunctTokenizer عمل می‌کند و صرفا تابع tokenize آن روی یک شی از آن صدا زده می‌شود.

اگر می‌خواهیم از یک tokenizer خاص با تنظیمات اولیه ثابت استفاده کنیم بهتر است از نوع کلاسی آن‌ها استفاده شود ولی اگر برای متن خاص و با تنظیمات خاص می‌خواهیم استفاده کنیم از توابع. همچنین ممکن است توابع شرایط خاص خود را ارضا کند.

بخش ۸:

در این بخش عملیات stemming (ریشه یابی) روی کلمات صورت گرفت است. در این بخش از دو روش PorterStemmer و LancasterStemmer استفاده شده است که در آن PorterStemmer از قواعد ساده تری استفاده شده است و LancasterStemmer نتایج بهتری داشته است.

بخش ۹:

در این بخش عملیات Lemmatization روی دسته‌ای از کلمات صورت گرفته است. که در آن کلمات به حالت نگارشی اولیه‌ی خود برگردانده می‌شود.

این تابع با ورودی‌های پیش فرض خوب عمل نمی‌کند و برای بهتر عمل کردن آن نیاز به دانستن نقش کلمه (فعل، اسم، صفت و ...) دارد که در آن به عنوان ورودی pos نقش آن نیز مشخص می‌شود که همان طور که معلوم است دقت بالایی به دست می‌آورد.