تمرین دوم درس پردازش زبان طبیعی

عنوان: دسته‌بندی متن

بخوانید:

توضیحات هر بخش در کامنت‌های هر سلول و کدهای markdown در فایل ipynb. قرار گرفته است و در این فایل توضیحات هر تابع و کلاس قرار دارد.

فایل‌های خواسته شده در فولدر Requested files قرار دارند.

معرفی توابع کلاس Corpus

```
__init__(self, data, replace_frequent=True, train = True, char = False, fr
equent_number = 10000)
```

با ساختن کلاسی از جنس corpus این تابع فراخوانی می‌شود. به عنوان ورودی دیتا را میگیرد. همینطور میتوان پامترهای پیش فرض آن را با توجه به کاربردی که این شی از کلاس قرار است برای ما داشته باشد تغییر داد. این پارامترها به ترتیب تعیین میکنند که کلمات کم تکرار جایگزین شوند، با دیتا به عنوان داده آموزش رفتار شود، اقدامات در سطح کاراکتر انجام شود و درنهایت تعداد کلماتی که به عنوان کلمات پرتکرار میخواهد را بدهد.
این تابع به کمک تابع clean و توابع دیگر دیتا را تمیز میکند و دیکشنری‌ها و سایر اشیا مورد خواسته شده را میسازد.

```
clean(self, data)
```

با دریافت کل دیتا مراحل تمیزکردن داده را به ازای هر خبر انجام می‌دهد و در نهایت لیستی از لیست کلمات هر خبر را به عنوان خروجی میدهد. این تابع از توابع کمکی replace_numbers و character_refinement که رشته تحویل میگیرند و اعداد را با N و کاراکترها را با کاراکتر مناسب جایگزین میکنند و تابع remove_stopwords که لیستی از اعداد میگیرد و کلمات توقف آنها را حذف میکند کمک میگیرد.

```
create_dictionary(self)
```

این تابع از کلمات و تکرارشان در دیتا دیکشنری میسازد.

```
number_of_all_tokens(self)
```

تعداد کل کلمات دیتا را به عنوان خروجی میدهد.

number_of_all_unique_tokens(self)
تعدا کلمات یکتا در دیتا را به عنوان خروجی میدهد.

most_frequent(self, number = 50)
به تعداد خواسته شده که به عنوان ورودی میگیرد مجموعه‌ای از کلمات پرتکرار بازمیگرداند.

replace_with_UNK(self)
کلماتی که پرتکرار نیستند را با UNK جایگزین میکند.

create_dictionaries(self)
به کمک self.dictionary و مجموعه کلمات پرتکرار self.most_frequent_words دو دیکشنری word2index و index2word را میسازد.

create_dictionary_char(self)
تمام کاراکترهای دیتای تمیزشده را به عنوان خروجی میدهد. (البته برای به دست آوردن کاراکترهای نامتعارف در مرحله تمیزکردن دیتا، فراخوانی تابع self.character_refinement(news) وجود نداشت و از دیتایی که از نظر کاراکتر تمیز نشده بود استفاده میکرد.)

load_requirements(self)
فایلهای کلمات توقف و جایگذاری کاراکترها در این تابع خوانده میشوند و به خروجی داده میشوند.

character_refinement(self, news)
با دریافت رشته جای تمام کاراکترها، کاراکتر مناسب قرار میدهد.

create_dictionaries_char(self)
همانند تابع create_dictionaries کار را در سطح کاراکتر انجام میدهد و دو دیکشنری char2index و index2char را به خروجی میدهد.

tokenize(self, level=0, data = 1)
دیتای داده شده را توکنایز میکند. اگر دیتا به عنوان ورودی نگیرد، از self.corpus استفاده میکند.

vectorize(self, data, vectorize_type='BoW')
با استفاده از تکنیک Bag of Words دیتا را وکتورایز میکند.

less_than_ave(self, y, add_pad = True)
در دیتای آموزش اگر طول خبر از میانگین بیشتر باشد، آن جمله را حذف میکند و اگر لازم باشد برای پر کردن خبر تا اندازه میانگین از PAD استفاده میکند. لازم است در فرایند حذف خبر، دسته خبر نیز حذف شود پس آن را نیز به عنوان ورودی میگیرد.

معرفی توابع کلاس NaiveBayes

```
__init__(self)
```
با ساخت شی از این کلاس تنها متغیرها و لیستها مقداردهی اولیه میشوند و اتفاق خاصی نمیافتد.

```
fit(self, X, y)
```
با فراخوانی این تابع، توابع لازم صدا زده میشوند و مدل آموزش میبینند.

```
predict(self, X_test)
```
برای دیتای داده شده به ازای هر خبر، به کمک تابع __pred دسته‌ای را پیشبینی میکند.

```
__pred(self, x)
```
برای لیست کلمات داده شده، احتمالات را محاسبه میکند.

```
evaluate(self, result, y)
```
مقدار صحیح دسته را همراه با آنچه به کمک مدل پیشبینی شده دریافت میکند و دقتهای خواسته شده و ماتریس آشفتگی را بدست می آورد. (به دلیل حالتهایی که تقسیم بر 0 ایجاد میشد، محاسبه recall، precision و f1 را کامنت کردم.)

```
__create_words_count_class(self)
```
این تابع کمکی، لیستی از دیکشنری میسازد که در هرکدام تعداد تکرار هر کلمه در آن کلاس قرار دارند که برای محاسبه احتمالات در تابع __pred از آن استفاده میشود.

```
__count_classes(self)
```
این تابع کمکی تعداد کلمات هر کلاس را میسازد که در تابع __pred از آن استفاده میشود.

```
__new_words_dict(self, x)
```
برای داده تست، برای هر خبر دیکشنری میسازد که در فرایند smoothing در محاسبه احتمالات استفاده میشود.