به نام خدا

دانشگاه تهران

دانشکدگان فنی

دانشکده مهندسی برق و کامپیوتر

درس پردازش زبان طبیعی

تمرین شماره یک

بهمن ماه ۱۴۰۲

فهرست سوالات

سوال اول

یکی از روش‌های مرسوم پیاده‌سازی tokenizerها، استفاده از عبارت‌های منظم[1] است. در ادامه، تکه کد مربوط به یک tokenizer سفارشی که با استفاده از این روش پیاده‌سازی شده است را مشاهده می‌کنید. با توجه به این tokenizer، به سوالات زیر پاسخ دهید. (درباره ساختار عبارت‌های منظم در پایتون می‌توانید در این لینک بیشتر مطالعه کنید).

```
import re

def custom_tokenizer(text):
    pattern = r'\b\w+\b'

    tokens = re.findall(pattern, text)

    return tokens
```

بخش اول

Tokenizer فوق، چه نوع tokenizer می‌باشد؟ یعنی مبتنی بر حرف[2]، مبتنی بر زیرکلمه[3] یا مبتنی بر کلمه[4] است؟ با ذکر مثال، ایرادات این نوع tokenizer را شرح دهید.

بخش دوم

ابتدا جمله زیر را با tokenizer مذکور tokenize نمایید و سپس دو نمونه از ایرادات tokenization انجام شده را ذکر کنید.

- Just received my M.Sc. diploma today, on 2024/02/10! Excited to embark on this new journey of knowledge and discovery. #MScGraduate #EducationMatters.

بخش سوم

با توجه به ایرادات ذکر شده در بخش دوم، tokenizer سفارشی را به گونه‌ای تغییر دهید که حداقل یکی از ایرادات ذکر شده را رفع نماید.

---

1. Regular Expression
2. Character-based
3. Subword-based
4. Word-based

سوال دوم

درباره tokenizerهای مورد استفاده در مدل‌های زبانی بزرگ مانند BERT و GPT تحقیق کنید و سپس به سوالات زیر پاسخ دهید.

مجموعه داده

برای انجام بخش سوم این سوال، فایل متنی کتاب «در مدار ماه»[5] در مسیر زیر در اختیار شما قرار گرفته است.

- ./data/All_Around_the_Moon.txt

بخش اول

Tokenizer مورد استفاده در هر یک از مدل‌های زبانی BERT و GPT از کدام یک از انواع مبتنی بر حرف، مبتنی بر زیرکلمه و یا مبتنی بر کلمه است؟ علت انتخاب این نوع از tokenizerها در مدل‌های زبانی بزرگ را توجیه کنید.

بخش دوم

مدل‌های زبانی BERT و GPT از دو الگوریتم متفاوت برای پیاده‌سازی tokenizerهای سفارشی خود استفاده نموده‌اند. این دو الگوریتم را نام ببرید و به طور خلاصه تفاوت آن‌ها را شرح دهید.

بخش سوم

به منظور درک شهودی تفاوت دو الگوریتم مورد اشاره در بخش دوم، موارد زیر را به ترتیب انجام دهید.

- برای هر یک از دو الگوریتم، یک tokenizer مجزا پیاده‌سازی نموده و آن را به صورت جداگانه بر روی مجموعه متنی که در اختیار شما قرار گرفته است، آموزش دهید. (استفاده از پیاده‌سازی‌های آماده الگوریتم‌های مورد نظر بلامانع است؛ توجه کنید که هدف از این تمرین، پیاده‌سازی tokenizerهای سفارشی و پیشرفته مدل‌های زبانی BERT و GPT و یا استفاده از نسخه آموزش دیده آن‌ها مانند BertTokenizer و GPTTokenizer نمی‌باشد؛ بلکه هدف این است که الگوریتم مورد استفاده در tokenizer این مدل‌ها را شناسایی کرده و سپس ساده‌ترین نسخه آن را پیاده‌سازی نمایید.)
- اندازه واژگان[6] ساخته شده توسط هر یک از tokenizerها را گزارش دهید. آیا این دو عدد با هم متفاوت هستند؟ علت چیست؟
- دو جمله زیر را به صورت جداگانه با هر یک از tokenizerهای پیاده‌سازی شده در قسمت قبل tokenize کنید و tokenهای تولید شده توسط هر یک را گزارش دهید. در صورت وجود تفاوت میان tokenهای تولید شده، این تفاوت را با توجه به تفاوت عملکرد دو tokenizer توجیه نمایید.

- This darkness is absolutely killing! If we ever take this trip again, it must be about the time of the sNew Moon!
- This is a tokenization task. Tokenization is the first step in a NLP pipeline. We will be comparing the tokens generated by each tokenization model.

[5]. Verne, Jules. Around the Moon. 1870.
[6]. Vocabulary size

سوال سوم

همانطور که می‌دانید، مدل های زبانی سعی دارند که احتمال وقوع دنباله‌ای از کلمات را پیش‌بینی کنند. یکی از ساده‌ترین مدل‌های زبانی، n-gramها هستند. n-gramها از جمله مدل‌های احتمالاتی هستند که سعی دارند به کمک روابط موجود در احتمال، کلمه بعدی را با در نظر گرفتن کلمات قبلی موجود در متن پیش‌بینی کنند. در این بخش قصد داریم تا یک مدل زبانی n-gram را به منظور انجام ماموریت تکمیل متن[7] آموزش دهیم.

مجموعه داده

برای انجام این سوال، فایل متنی کتاب «تارزان، فرمانروای جنگل»[8] در مسیر زیر در اختیار شما قرار گرفته است.

- ./data/Tarzan.txt

بخش اول

فایل متنی دادگان را در برنامه خود بارگیری[9] و ذخیره کنید و سپس به جهت آموزش مدل زبانی n-gram، پیش‌پردازش‌های لازم (tokenization) را بر روی داده‌های متنی ورودی انجام دهید. (به منظور بهبود عملکرد مدل n-gram خود، می‌توانید از سایر روش‌های نرمال‌سازی داده‌های متنی در کنار tokenization استفاده نمایید اما الزامی به انجام این مورد وجود ندارد.)

بخش دوم

پس از آماده‌سازی ورودی، bigram (n-gram with n=2)های موجود در مجموعه متنی را یافته و با محاسبه احتمال مربوط به هر bigram مدل زبانی خود را آموزش دهید؛ توجه کنید که یکی از چالش‌های موجود در مدل زبانی n-gram مشکل data sparsity است؛ در این باره تحقیق کنید و این چالش را در زمان آموزش مدل زبانی خود رفع نمایید. (استفاده از توابع آماده به جهت یافتن bigramها و همچنین فرکانس حضور آنها در متن بلامانع است اما دقت کنید که محاسبه احتمالات و پیاده‌سازی مدل زبانی باید توسط خود شما انجام شده باشد.)

بخش سوم

با استفاده از مدل زبانی آموزش دیده شده، عبارات زیر را تا حداقل ۱۰ توکن دیگر تکمیل نمایید.

- Knowing well the windings of the trail he …
- For half a day he lolled on the huge back and …

بخش چهارم

---

7. Text Continuation Task
8. Burroughs, Edgar Rice. Tarzan, Lord of the Jungle. 1914.
9. Load

بخش‌های دوم و سوم را برای مدل زبانی n-gram با n=3 و n=5 تکرار نموده و جملات تکمیل شده توسط این دو مدل را با مدل bigram مقایسه کنید.

بخش پنجم

آیا می‌توان پارامتر n در مدل زبانی n-gram را بدون محدودیت افزایش داد؟ چرا؟

سوال چهارم

پیش از توسعه شبکه‌های عصبی و به کارگیری آن‌ها در انجام ماموریت تحلیل احساسات[10]، از مدل‌های زبانی n-gram برای انجام این تسک استفاده می‌شد. در این بخش قرار است که با نحوه انجام این تسک توسط n-gram آشنا شوید. بدین منظور، یک برنامه پایتون نیمه‌کامل در مسیر codes/Q4.ipynb/. در اختیارتان قرار گرفته است؛ ساختار این برنامه به شرح زیر است:

- ابتدا دادگان مورد بررسی در سوال، بارگیری و ذخیره شده است و سپس مجموعه داده با نسبت ۸۰ به ۲۰ به دو قسمت مجموعه داده آموزش و آزمایش تقسیم شده است.

```
# Step 1: Load the data
data = pd.read_csv('google_play_store_apps_reviews.csv')

# Step 2: Split the data
train_data, test_data = train_test_split(data, test_size = 0.2, random_state = 42)
```

- در مرحله بعدی، تابعی با عنوان train_ngram پیاده‌سازی شده است که در ورودی، داده‌های آموزش (متن به همراه برچسب) به همراه پارامترِ n مدل زبانی n-gram را دریافت نموده و یک مدل زبانی ساده به جهت انجام ماموریت تحلیل احساسات را آموزش می‌دهد.

```
# Step 3: Build the n-gram Language Model
def get_ngrams(text, n):
    tokens = nltk.word_tokenize(text)
    return list(ngrams(tokens, n))

def train_ngram(data, n):
    positive_ngrams = []
    negative_ngrams = []

    for index, row in data.iterrows():
        grams = get_ngrams(row['review'], n)
        if row['polarity'] == 1:
            positive_ngrams.extend(grams)
        elif row['polarity'] == 0:
            negative_ngrams.extend(grams)

    positive_freq = FreqDist(positive_ngrams)
    negative_freq = FreqDist(negative_ngrams)

    return positive_freq, negative_freq
```

حال با توجه به برنامه و توضیحاتی که درباره قسمت‌های مختلف آن ارائه شد، موارد خواسته شده زیر را انجام دهید.

[10]. Sentiment Analysis

مجموعه داده

برای انجام این سوال، مجموعه داده‌ای حاوی اطلاعات مربوط به نظرات کاربران GooglePlay درباره برنامه‌های کاربردی منتشر شده در آن، به همراه دو برچسب مثبت (۱) و منفی (۰) در مسیر زیر در اختیار شما قرار گرفته است.

- ./data/ google_play_store_apps_reviews.xlsx

بخش اول

مراحل آموزش مدل و خروجی آن را به دقت مطالعه کرده و تابعی با عنوان test_ngram بنویسید که بتوان به کمک آن داده‌های تست را با استفاده از مدل n-gram آموزش داده شده برچسب‌گذاری نمود. نمونه‌ای از امضای تابع[11] خواسته شده در ادامه قابل مشاهده است اما الزامی به استفاده از این امضا نیست و شما می‌توانید تابع آموزش خود را به دلخواه سفارشی‌سازی کنید.

```
# Step 5: test the n-gram

def test_ngram(data, positive_freq, negative_freq, n):
  pred_labels = []

  # Implement testing n-gram language model process here.

  return pred_labels
```

بخش دوم

برچسب‌های پیش‌بینی شده توسط مدل را با برچسب‌های واقعی داده‌ها مقایسه کرده و با استفاده از معیارهای ارزیابی (مانند دقت)، گزارشی از عملکرد مدل ارائه نمایید.

[11]. Function Signature

ملاحظات (حتما مطالعه شود)

تمامی نتایج شما باید در یک فایل فشرده با عنوان NLP_CA1_StudentID تحویل داده شود.

- خوانایی و دقت بررسی‌ها در گزارش نهایی از اهمیت ویژه‌ای برخوردار است. به تمرین‌هایی که به صورت کاغذی تحویل داده شوند یا به صورت عکس در سایت بارگذاری شوند، ترتیب اثری داده نخواهد شد.
- کدهای نوشته شده برای هر بخش را با نام مناسب مشخص کرده و به همراه گزارش تکلیف ارسال کنید. همه‌ی کدهای پیوست گزارش بایستی قابلیت اجرای مجدد داشته باشند. در صورتی که برای اجرا مجدد آنها نیاز به تنظیمات خاصی می‌باشد بایستی تنظیمات مورد نیاز را نیز در گزارش خود ذکر کنید.
- تمرین تا یک هفته بعد از مهلت تعیین شده با تاخیر تحویل گرفته می شود. دقت کنید که شما جمعا برای تمام تکالیف، ۱۴ روز زمان تحویل بدون جریمه دارید که تنها از ۷ روز آن برای هر تمرین می‌توانید استفاده کنید، در صورتی که این ۱۴ روز به اتمام رسیده باشد، به ازای هر روز تاخیر در ارسال تمرین، ده درصد جریمه میشوید.
- **توجه کنید این تمرین باید به صورت تک نفره انجام شود و پاسخ‌های ارئه شده باید نتیجه فعالیت فرد نویسنده باشد (همفکری و به اتفاق هم نوشتن تمرین نیز ممنوع است). در صورت مشاهده تشابه به همه افراد مشارکت کننده، نمره تمرین صفر و به استاد نیز گزارش می‌گردد.**
- در صورت بروز هرگونه مشکل با ایمیل زیر در ارتباط باشید:

marziehbagherinia@gmail.com

مهلت تحویل بدون جریمه: ۱۴ اسفندماه ۱۴۰۲

مهلت تحویل با تأخیر، با جریمه ۱۰ درصد: ۲۱ اسفندماه ۱۴۰۲