به‌نام خداوند بخشنده و مهربان

پروژه: پروژه پایان‌ترم
استاد: رضا رمضانی
درس: داده کاوی پیشرفته
دستیار: امیر سرتیپی (@haj_amir_13)

۱- نام و نام‌خانوادگی نفر اول: فاطمه مومنی
۲- نام و نام‌خانوادگی نفر دوم: نگین شمس

تحلیل احساسات[1] یکی از وظایف[2] پردازش زبان طبیعی به شمار می‌رود. در چنین وظیفه‌ای شما مجموعه‌ای از جملات در اختیار دارید که باید مثبت یا منفی بودن آن را شناسایی کنید. برای مثال از این وظیفه می‌توان برای تحلیل نظرات کاربران برای اینکه آیا از محصولی رضایت دارند یا خیر، استفاده کرد. در این وظیفه هدف این است که برچسب رضایت و نارضایتی کاربران به درستی شناسایی شوند. سوال‌های مربوط به پروژه و نیازمندی‌های آن در ادامه بیان شده است.

مجموعه‌ی داده‌ای که در اختیار شما قرار گرفته است، شامل ۷۰.۰۰۰ نظر کاربران اپلیکیشن غذای اسنپ فود است که در این مجموعه داده، ۳۵.۰۰۰ نظر دارای احساس رضایت و ۳۵.۰۰۰ نظر نیز دارای احساس عدم رضایت از غذایی که سفارش داده‌اند وجود دارد. مجموعه‌ی داده Training، Validation و Test به ترتیب ۵۶.۷۰۰، ۶.۳۰۰ و ۷.۰۰۰ سطر را دارند که در مجموع ۷۰.۰۰۰ داده‌ی ما را تشکیل می‌دهند. در ادامه مثالی از نظر رضایت و عدم رضایت کاربران آورده شده است.

نمونه‌ای از رضایت مشتری:

قرار بود ۱ ساعته برسه ولی نیم ساعت زودتر از موقع رسید، شما ببین چقدررررررررررر پلاک خفنهههههه، من سالهاست مشتریشونم و سالهاست مزه بهشت میده غذاشون

نمونه‌ای از عدم رضایت مشتری:

با عرض سلام غذا به حدی سرد بود متوجه نشدم چی بود متاسفم

[1] Sentiment Analysis
[2] Task

۱. برای ساخت شبکه پیش‌بینی کننده خود از یکی از کتاب‌خانه‌های Torch یا TensorFlow استفاده کنید. از شبکه‌ی LSTM و CNN استفاده کرده و داده‌ها را به عنوان ورودی به آن‌ها بدهید و سپس موارد خواسته شده را همراه با توضیح مختصر در سند خود ذکر کنید.

۲. پیش‌پردازشی بر روی داده انجام دهید (در صورت لزوم) و دلیل هر پیش‌پردازش را ذکر کنید.

- **حذف کلمات توقف**

به‌منظور حذف کلمات توقف[3] و علائم نگارشی[4]، از فایل کلمات توقف فارسی[5] استفاده شده است اما لازم است در ابتدا پردازش مختصری صورت پذیرد و برخی کلمات مانند کلمات «دیر» یا «زود» از این لیست حذف شوند. زیرا در تعیین میزان رضایت کاربر نقش دارند. این کاراکترها و کلمات، به‌دلیل تمیز کردن مجموعه داده و آماده‌سازی آن برای ساخت مدل، حذف شده‌اند.

```
path = "/content/drive/MyDrive/persian_stopwords.txt"
stop_words = []
f = open(path, "r", encoding='utf-8-sig')
for x in f:
    stop_words.append(x.rstrip("\n"))

train_data['comment'] = train_data['comment'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop_words)
validation_data['comment'] = validation_data['comment'].apply(lambda x: ' '.join([word for word in x.split() if word not in (s
```

- **تبدیل متن انگلیسی به فارسی**

برخی از نظرات ثبت‌شده در این مجموعه داده یا بخشی از آن‌ها به زبان انگلیسی نوشته شده است. به‌منظور تبدیل این جملات به متن فارسی، از یک تابع استفاده شده است که در ابتدا بخشی از نظرات که با الفبای غیرفارسی نوشته شده را جدا نموده و سپس با استفاده از ماژول تشخیص زبان کتابخانه‌ی googletrans، اقدام به تشخیص زبان متن می‌کند. در صورتی‌که زبان متن مذکور انگلیسی باشد، با استفاده از تابع translate متن ترجمه شده و بازگردانده می‌شود. این کار برای پردازش تمامی نظرات درون مجموعه داده و استفاده کردن از آن‌ها در ساخت مدل انجام شده است.

[3] Stop Words
[4] Punctuation
[5] https://github.com/kharazi/persian-stopwords

```python
import re
from googletrans import Translator

def english_handle(text):
  english_part = ""
  farsi_part = ""

  translator=Translator()

  for word in text.split():
    z = re.match("[A-Za-z]+", word)
    if z:
      english_part += word + ' '
    else:
      farsi_part += word + ' '
```

```python
if english_part != "":
  a = translator.detect(english_part)
  if a.lang == 'en':
    text=(english_part)
    destination_language = {
        "Persian": "FA",

    }

    for key, value in destination_language.items():
        english_part = translator.translate(text, dest=value).text

new_text = farsi_part + english_part

return new_text
```

- **نرمال‌سازی متن**

به‌منظور نرمال‌سازی متن از کتابخانه‌ی پارسی‌وار[6] استفاده شده است. در صورتی‌که مقدار پارامتر pinglish_conversion_needed نرمال‌ساز برابر true قرار گیرد، متن‌های فارسی که با الفبای انگلیسی نوشته شده باشند به‌شکل املای فارسی بازگردانده می‌شوند. هم‌چنین از stemmer برای ریشه‌یابی کلمات استفاده شده است. در پایان اگر کلمه‌ای حاوی کاراکتر غیرفارسی باشد، حذف می‌شود. در برخی از کلمات نیز بعضی از حروف چندین بار تکرار شده‌اند. به‌منظور یکسان‌سازی این کلمات، تکرارهای اضافی حروف حذف شده‌اند. به‌دلیل ایجاد برخی نیم‌فاصله‌های نادرست بین کلمات، نیم‌فاصله‌های ایجادشده توسط نرمال‌ساز با فاصله جایگزین شده‌اند.

```python
from parsivar import Normalizer
from parsivar import Tokenizer
from parsivar import FindStems

my_normalizer = Normalizer(pinglish_conversion_needed=True)
my_tokenizer = Tokenizer()
my_stemmer = FindStems()

# second_normalizer = Normalizer(statistical_space_correction=True)

def preprocess(text):
  tokens = my_tokenizer.tokenize_words(my_normalizer.normalize(text))
  stemmed_tokens = []
  for token in tokens:
      stemmed_tokens.append(my_stemmer.convert_to_stem(token))

  clean_text = "".join([word + " " for word in stemmed_tokens])
  clean_text = re.sub('[a-zA-Z]+', '', clean_text)
  clean_text = clean_text.replace('\u200c', ' ')

  return clean_text
```

[6] Parsivar

```
def remove_consec_duplicates(s):
    new_s = ""
    prev = ""
    for c in s:
        if len(new_s) == 0:
            new_s += c
            prev = c
        if c == prev:
            continue
        else:
            new_s += c
            prev = c
    return new_s
```

```
train_data['comment']=train_data['comment'].apply(preprocess)
validation_data['comment']=validation_data['comment'].apply(preprocess)

train_data['comment']=train_data['comment'].apply(remove_consec_duplicates)
validation_data['comment']=validation_data['comment'].apply(remove_consec_duplicates)
```

از تعبیه کلمه‌های[7] word2vec و fastText[8] برای وزن‌های اولیه‌ی شبکه‌ی خود استفاده کنید. (شما برای هر کلمه‌ی نیاز دارید تا برداری از اعداد را جایگزین آن کنید، تعبیه کلمه برای همین عملیات استفاده می‌شود)

با استفاده از فایل 61.zip می‌توان ماتریس تعبیه کلمات از پیش آموزش یافته را به‌دست آورد.

```
import os
import zipfile
with zipfile.ZipFile('/content/drive/MyDrive/61.zip', 'r') as zip_ref:
    zip_ref.extractall('/tmp/pre_embedding')

embeddings_index = {}
f = open('/tmp/pre_embedding/model.txt', encoding = "ISO-8859-1")
for line in f:
    word, coefs = line.split(maxsplit=1)
    coefs = np.fromstring(coefs, "f", sep=" ")
    embeddings_index[word] = coefs

f.close()

print('Found %s word vectors.' % len(embeddings_index))

max_length = 100
word_index = tokenizer.word_index

embedding_matrix = np.zeros((len(word_index) + 1, max_length))
for word, i in word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        # words not found in embedding index will be all-zeros.
        embedding_matrix[i] = embedding_vector
```

سپس لایه embedding ایجاد می‌شود:

[7] Word embedding
[8] http://vectors.nlpl.eu/repository/

```
embedding_layer = Embedding(input_dim=len(word_index) + 1,
                            output_dim=max_length,
                            weights=[embedding_matrix],
                            input_length=max_length,
                            trainable=False)
```

۳. مدل خود را آموزش داده و معیار F1[۹] را گزارش دهید. معیارهایی از جمله صحت[۱۰]، دقت[۱۱] و فراخوانی[۱۲] را برای مدل خود ذکر کنید.

شما باید مدل خود را برای چند دوره[۱۳] آموزش دهید. در هر دوره مدل شما تمامی مجموعه‌ی داده را مشاهده می‌کند و در جهت بهتر شدن مدل (بهینه‌سازی) تلاش می‌کند. (برای مثال ۱۰ دوره)

مقدار معیارهای نام‌برده برای ارزیابی مدل CNN و LSTM با استفاده از مجموعه "dev" در جدول زیر آورده شده است.

مدل / معیار	F1	صحت	دقت	بازخوانی	هزینه
CNN	۰.۸۲	۰.۸۱۵	۰.۸۲	۰.۸۲	۰.۵۵۷
LSTM	۰.۷۸	۰.۸۱۵	۰.۷۸	۰.۷۸	۰.۵۵۷

با توجه به مقادیر جدول فوق، مدل CNN نسبت به LSTM اندکی بهتر است.

```
197/197 [==============================] - 1s 4ms/step - loss: 0.5572 - accuracy: 0.8156
Test accuracy : 0.8156265020370483
```

	precision	recall	f1-score	support
0	0.81	0.82	0.82	3148
1	0.82	0.81	0.82	3149
accuracy			0.82	6297
macro avg	0.82	0.82	0.82	6297
weighted avg	0.82	0.82	0.82	6297

[9] Macro-Averaged F-Score
[10] Accuracy
[11] Precision
[12] Recall
[13] Epoch

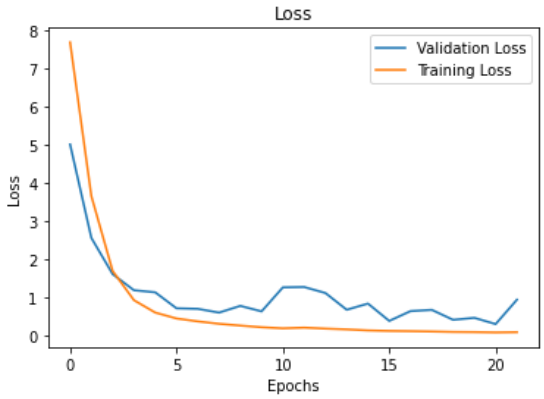
```
197/197 [==============================] - 1s 3ms/step - loss: 0.9134 - Accuracy: 0.7818
Test accuracy : 0.7818008661270142
```

	precision	recall	f1-score	support
0	0.78	0.78	0.78	3148
1	0.78	0.78	0.78	3149
accuracy			0.78	6297
macro avg	0.78	0.78	0.78	6297
weighted avg	0.78	0.78	0.78	6297

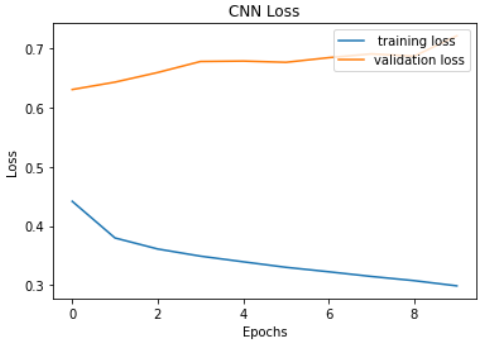
۴. نمودارهای traning loss و validation loss را ترسیم کنید و آن را تحلیل کنید.

پس از پایان هر دوره، مقدار loss شبکه‌ی خود در زمان train و در زمان validation را ذخیره کنید. پس از آموزش مدل برای مثلا ۱۰ دوره، مقادیری دوره‌های مختلف را بصورت نمودار ترسیم نمایید. مثالی از این نمودار آورده شده است که مدل پس از تقریبا ۳ دوره به اصطلاح overfit شده است. (محل تقاطع traning loss و validation loss).

با توجه به موارد گفته شده، بهترین مدل خود را ذخیره کرده و برای تست مدل از آن استفاده کنید.



با توجه به نمودارهای traning loss و validation loss و عدم تقاطع آن‌ها با یکدیگر، مدل از ابتدا overfit شده است. این حالت برای هر دو مدل رخ داده است.

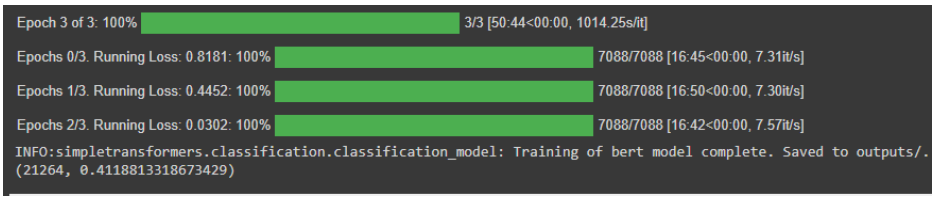


۵. از مدل پیش آموزش دیده شده، مانند mBERT و XLM-Roberta به عنوان مدل خود استفاده و آن را آموزش دهید. همانند بخش قبل معیار F1 را گزارش و نمودارهای validation loss و training loss را ترسیم کنید. مقایسه‌ای بین مدل‌های قبل و مدل جدید انجام دهید.

به‌منظور استفاده از این دو مدل، از کتابخانه‌ی simpletransformers استفاده شده است.

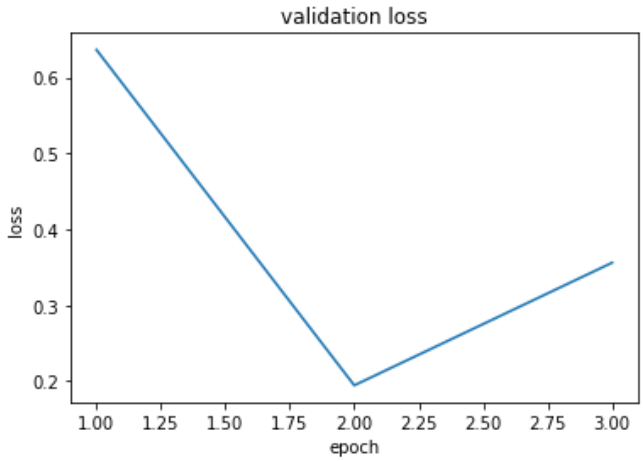
نتایج مدل mbert به‌صورت زیر است:

F1	Val_loss	accuracy
0.86	0.372	0.86



همچنین نتایج مدل xlm_roberta به‌صورت زیر می‌باشد:

F1	Val_loss	accuracy
0.863	0.35	0.863



با توجه به اعداد به‌دست‌آمده، مدل‌های از پیش آموزش یافته، نتیجه بهتری دارند.

نکات تکمیلی

- برای درگیر نشدن با کانفیگ‌ها پیشنهاد می‌شود از google colab استفاده کنید.
- برای سریع شدن محاسبات پردازش را بر روی GPU و بر روی colab انجام دهید.
- لطفا سند پروژه را حتما در سامانه‌ی کوئرا ارسال کنید.
- لطفا پاسخ‌های خود را در سند پروژه نوشته و در قالب یک فایل PDF ارسال کنید.
- نام سند ارسالی Final-{Name Family}-{Name Family} (یک نفر از اعضای گروه ارسال کند)
- برای تحویل پروژه لازم است کدی نوشته شود که یک دیتاست را دریافت و دقت و F1 را بر روی بهترین مدل ذخیره شده توسط شما گزارش می‌کند. مجموعه داده تست در روز تحویل در اختیار شما قرار خواهد گرفت.
- تمامی فایل‌های مورد نیاز این تمرین در این لینک قابل دسترس است.