

## تمرين اول

## درس پردازش زبان های طبیعی

سيد محمدرضا حسيني

کد دانشجویی: ۴۰۲۲۱۱۶۸۳

مدرس: دکتر عسگری

برای این تمرین داده ها را از سایت crawl، www.theguardian.com کردیم. دادگان جمع آوری شده از بخش نقد فیلم ها میباشد. در ابتدا لینک های موجود در هر صفحه اصلی که لینک مرتبط به نقد هر فیلم میباشد را به دست می آوریم.

```
for i in range(1,300):

try:

Main_url = "https://www.theguardian.com/film+tone/reviews?page={0}".format(i)

page = requests.get(Main_url)

soup = BeautifulSoup(page.text,"html.parser")

script_tag = soup.findAll('script', {'type': 'application/ld+json'})

json_content = json.loads(script_tag[2].string)

for j in range(0,len(json_content)):
```

سپس از هر لینک اطلاعات را بدست می آوریم. اطلاعات ذخیره شده شامل هدلاین نقد ، امتیاز داده شده به فیلم و متن نقد و یک خط خلاصه نوشته شده میباشد.

```
for j in range(0,len(json_content)):
   url =json_content['itemListElement'][j ['url']
    try:
      news = requests.get(url)
      df = pd.DataFrame(reviews)
       with open('data.json', 'w') as f:
          json.dump(reviews, f, indent=2)
       df.to_csv("'reviews.csv', index=False")
        continue
    news_soup = BeautifulSoup(news.text, "html.parser")
    paragraphs= news_soup.findAll("p")
   temp =""
    stars = news_soup.find_all('div', class_="dcr-1tlk9ix")
   rating = sum(1 for star in stars if star.find('path').get('fill') = 'currentColor')
   for k in range(1,len(paragraphs)):
       temp+=(paragraphs[k].text)
    news_articles = [{'news_headline': news_soup.find("h1").text,
                      'news_firstline': paragraphs[0].text,
                      'rating': rating,
                      'news_article': temp
    reviews.extend(news_articles)
```

```
("nous_headline": "Godzilla x Kang: The New Empire review \u2013 breezy, forgettable monster sequel",
"nous_firstline": "There\u2019s a likable, light-hearted zip to the monster mash follow-up but energy dissipates when we\u2019re stuck with the humans",
"rating": 3,
"nous_naticle": "It was a strange old time when the creature feature mash-up Godzilla vs Kong was released, the first major blockbuster in cinemas since Covid shuttered them
"nous_firstline": "Many Poppins review \u2013 Disney\u2019s entertainment sugar rush possesses thermouslear brilliance",
"nous_firstline": "Many Poppins review \u2013 Disney\u2019s entertainment sugar rush possesses thermouslear brilliance",
"nous_firstline": "Many Poppins review \u2013 Disney\u2019s Julie Andrews is superb in the role of the flying nammy, in a film filled with amazing songs",
"rating": 3,
"nous_article": Brilliant, entrancing, exhausting, and with thermonuclear showtunes from Richard and Robert Sherman, Disney\u2019s hybrid live-action/animation classic from
"nous_firstline": "Many Poppins review \u2013 Jack Black and Aukwafina in hurricane of slapstick more miss than hit",
"nous_firstline": "The lead pair make a brilliant double act, but the franchise has run out of its signature sweetness and charm",
"rating": 3,
"nous_article": The cuddly kung fu master is back. Jack Black returns as dumpling-loving panda Po, the unlikeliest of lean, mean fightin\u2019 machines. It\u2019s been eigh
"nous_firstline": "In his last weeks of life, the Oscar-winning composer is filmed at the piano by his son. It is an almost wordless pacen to a remarkable career",
"nous_firstline": "Short of presenting nothing more than music and a blank screen, this documentary about the late Japanese composer-performer Ryuichi Sakamato\u2019s last app
"nous_firstline": "A perceptive and candid look at mother-daughter discord drives the boisterous energy of Naryam Keshavar\u2019s comic crowdpleaser",
"nous_firstline": "A perceptive and candid look at mother-daughter discord drives the boisterou
```

پس از اینکار به سراغ پردازش دادگان میرویم. در بخش اول تسک انتخاب شده به دست آوردن عبارات کلیدی و صفات استفاده شده در این نقد ها میباشد. مراحل مرتبط به کد ها در نوتبوک به شکل مناسب کامنت گذاری شده است.

برای اینکار باید پیش پردازش های گفته شده را انجام دهیم تا انجام اینکار راحت تر شود و اطلاعات با ارزش تر باقی بمانند. در ابتدا تمامی Punctuation ها را حذف میکنیم. سپس تمامی کلمات را Lowercase کرده و در نهایت s' های موجود در هر کلمه را حذف میکنیم. برای نشان دادن عملکرد از Porter stemmer استفاده کردیم. با استفاده از stemming اتفاقی که می افتد این است که پس از دادن دادگان به Spacy برای شناسایی صفات و عبارات کلیدی این مدل به علت stemming اتفاق افتاده دچار مشکل میشود و نمیتواند این کار را به شکل درست انجام دهد. به عنوان مثال کلمه strange در هنگام stemming به strang تبدیل میشود و آن را verb تشخیص میدهد که این اشتباه با توجه به اینکه تسک ما تشخیص بیشترین صفات به کار رفته در تحلیل فیلم ها میباشد غلط میباشد.

```
sentences_a = [sent for sent in doc_preproccessed[0].sents]
for x in sentences_a:
    print([(tok.pos_,tok.lemma_) for tok in x])

* [('SPACE', ' '), ('VERB', 'strang'), ('ADJ', 'old'), ('NOUN', 'time'), ('NOUN', 'creatur'), ('NOUN', 'featur'), ('NOUN', 'mash'), ('NOUN', 'godzilla'), ('ADP', 'vs'), ('PROPN', 'kong'), ('VERB', 'relea'), ('ADV', 'first'), ('ADJ', 'major'), ('PROPN', 'blockbust'), ('PROPN', 'cinema'), ('PROPN', 'sinc'), ('PROPN', 'covid'), ('P
```

```
sents = [sent for sent in doc[0].sents]
for x in sents:
    print([(tok.pos_,tok.lemma_) for tok in x])

[('PRON', 'it'), ('AUX', 'be'), ('DET', 'a'), ('ADJ', 'strange'), ('ADJ', 'old'),
    ('NOUN', 'time'), ('SCONJ', 'when'), ('DET', 'the'), ('NOUN', 'creature'), ('VERB',
    'feature'), ('NOUN', 'mash'), ('PUNCT', '-'), ('NOUN', 'up'), ('PROPN', 'Godzilla'),
    ('ADP', 'vs'), ('PROPN', 'Kong'), ('AUX', 'be'), ('VERB', 'release'), ('PUNCT', ','),
    ('DET', 'the'), ('ADJ', 'first'), ('ADJ', 'major'), ('NOUN', 'blockbuster'), ('ADP',
    'in'), ('NOUN', 'cinemas'), ('SCONJ', 'since'), ('PROPN', 'Covid'), ('VERB', 'shutter')
    , ('PRON', 'they'), ('DET', 'all'), ('DET', 'a'), ('NOUN', 'year'), ('ADV', 'prior'),
    ('PUNCT', '.')]
```

اطلاعات مرتبط با دیتای استفاده شده بعد از پیش پردازش گفته شده در بالا. این پیش پردازش شامل stemming نمیباشد.

```
import itertools
sentence_words = list(itertools.chain(*tokenized_reviews))
print ('%-16s' % 'Number of words', '%-16s' % len(sentence_words))
print ('%-16s' % 'Number of unique words', '%-16s' % len(set(sentence_words)))
avg=np.sum([len(word) for word in sentence_words])/len(sentence_words)
print ('%-16s' % 'Average word length', '%-16s' % avg)
avg_sentence_length_c=np.mean([len(' '.join(sentence)) for sentence in sentence_words])
print ('%-16s' % 'Average sentence length in characters', '%-16s' % avg_sentence_length_c)

Number of words 327502
Number of unique words 39178
Average word length 6.097312382825143
Average sentence length in characters 11.194624765650286
```

با مقایسه داده ای که این پیش پردازش ها بر روی آن انجام شده در مقابل داده ای که فقط جملات توکنایز شده اند مشخص میشود با توجه به اینکه کلمات حذف شده تاثیری در تسک گفته شده ندارن، با حذف آن ها میتواند اطلاعات را سریعتر و به صورت موثرتری بدست آورد. همانطور که در تصویر زیر مشخص است توانسته ایم تمامی Noun\_chuncks ها را بدست آوریم.

```
keywords = []
for sentence in doc:
    temp=[]
   for chunk in sentence.noun_chunks:
        if chunk.text.lower() not in nlp.Defaults.stop_words:
           temp.append(chunk.text)
    keywords.append(temp)
np.array(keywords)
 C:\Users\smrh1\AppData\Local\Temp\ipykernel_34188\2185074432.py:1: VisibleDeprecationWarni
 different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dty
   np.array(keywords)
        0
      3 [music, a blank screen, this documentar...
      4 [The only daughter, a family, nine Iran...
      5 [an almost entirely dialogue-free anima...
      6 [A malign presence, the corner, poison,...
      7 [all the hot takes, the cinema, the sug...
      8 [no leprechauns, this abvsmal romantic ...
```

پس از آن بیشترین صفات استفاده شده توسط نویسندگان در این نقد ها را بدست آوردیم.

```
adj_freqdist = FreqDist(itertools.chain(*adjectives))
                                             # show the top 20 (word, frequency) pairs
top40words=adj_freqdist.most_common(40)
print ('%-16s' % 'word', '%-16s' % 'Frequency','%-16s' % '% of the total')
for topword in top40words:
   percent=(topword[1]/len(tokenized_reviews))
   print ('%-16s' % topword[0], '%-16s' % topword[1],'%-16s' % percent)
                  Frequency
                                   % of the total
 word
 old
                  613
                                    0.5125418060200669
                  586
                                    0.4899665551839465
 new
                  561
                                    0.4690635451505017
 dood
 own
                  549
                                    0.4590301003344482
                  544
                                    0.45484949832775917
 young
                  538
                                    0.4498327759197324
 more
 other
                  513
                                    0.4289297658862876
                                    0.42558528428093645
 first
                  509
                  491
                                    0.4105351170568562
 little
                  441
                                    0.36872909698996653
```

در بخش دوم با بررسی هدلاین های هر نقد با توجه به پترن موجود در آن سعی میکنیم که اسم فیلم را استخراج کنیم.

با توجه به اینکه به دنبال اسم فیلم هستیم پیش پردازش خاصی را مد نظر نداریم چون در صورت حذف کردن حتی stopwords اسم فیلم ممکن است متفاوت شود. تنها کاری که میکنیم توکنایز کردن جملات است. سپس با استفاده از regex پترن مدنظر را وارد کرده و سعی میکنیم اسم فیلم را بدست آوریم.

همانطور که در تصویر زیر مشخص است با استفاده از ابزار پردازش متن توانسته ایم با دقت مناسبی اسامی فیلم ها را از هدلاین نقد ها استخراج کنیم. این اسامی میتواند برای بهتر و کامل کردن دادگان استفاده شده و سبب شود که به ازای اسم فیلم ما نقد نوشته شده را نگه داری کنیم.

```
import re
movie_names=[]
for i in movie_nouns:
   for j,item in enumerate(i):
        # Use regular expression to find the movie name before the word "review"
        match = re.search(r"^(.*?)\s*review", item, re.IGNORECASE)
        if match:
            if len(i[:j])>0:
                temp=((i[:j].pop().strip())+" " + match.group(1).strip())
            else:
                temp= match.group(1).strip()
            movie_names.append([temp])
np.array(movie_names)
          godzilla x kong the new empire
      0
                            mary poppins
      1
      2
                         kung fu panda 4
                   ryuichi sakamoto opus
      3
      4
                     the persian version
      5
                               the devil
```