تمرین یکم درس پردازش زبان طبیعی

معین سلیمی

۴۰۱۳۰۰۵۰۳

پیاده‌سازی سامانه‌ای برای استخراج و پردازش خبرهای بازارهای مالی

توضیح

تمامی کدهای این تمرین در نشانی گیت‌هاب آن موجود است:

https://github.com/mnsalimi/telegram-crawler

در این تمرین داده‌های بازارهای مالی از دو دسته منبع کلی گردآوری شده‌اند. ۱) کانال‌های تلگرامی. ۲) خبرگزاری‌ها. هدف آن است که بتوان با استفاده از گردآوری این خبرها، تحلیل‌های مختلفی نیز بر روی آن‌ها اعمال کرد. در این تمرین از فریم‌ورک جنگو و همچنین موتور جستجوی الستیک‌سرچ برای ذخیره‌ی دادگان متنی استفاده شده است.

این پروژه شامل سه فاز کلی است:

۱) گردآوری خبر: بیش از ده کانال تلگرام برای این کار انتخاب شدند. کانال‌هایی که تعداد اعضای نسبتا بالایی (چنده‌هزار نفر و بیش‌تر) داشته باشند.

۲) پیش‌پردازش متن‌ها و آماده‌سازی برای تحلیل‌های مرحله‌ی سوم. برای نمونه تشخیص آن‌که یک پست مشخص، درباره‌ی کدام نماد(ها) صحبت می‌کند در این بخش و تحت عنوان پیش‌پردازش انجام می‌شود.

۳) تحلیل متن‌ها: ساختن نمایش ابری و تصمیم‌گیری درباره‌ی مثبت یا منفی بودن خبر بر روند سهم (ارز) و همچنین بررسی تنوع و یا جهت‌گیری کانال‌های بورسی از لحاظ پوشش نمادها

۱) گام یکم؛ گردآوری خبر

برای گردآوری متن‌های کانال‌های تلگرامی از کتابخانه‌ی `telethon` در پایتون استفاده شده است. با استفاده از این ابزار بیش از ۵۰۰ هزار پست تلگرامی از ده کانال گردآوری شده‌آند. لیست برخی از این کانال‌ها:

Bourseonline

Ariyaz_1

BOChannel

ATBourse

برای پیاده‌سازی کراولر مربوطه در پرونده‌ی telegram_crawler_pipeline.py یک کلاسه به نام CrawlerPipeline نوشته شده است. در تابع init آن، متغیرهای کلی مانند api_id، api_hash وجود دارند. این دو متغیر توسط توسعه‌دهندگانی که از کراولر تلگرام می‌خواهند استفاده کنند پر می‌شود. متغیر دیگری به نام channel_id وجود دارد که کانال مربوطه‌ای که می‌خواهید گردآوری شود را در آن می‌نویسیم. تابع crawl_channel برای گردآوری پست‌های کانال مربوطه طراحی شده.

```python
def crawl_channel(self, limit_datetime=None):
    with TelegramClient(self.sesseion, self.api_id, self.api_hash) as client:
        channel_name = client.get_entity(self.channel_id).title
        messages = client.get_messages(self.channel_id, limit=3000)
        print("received messages...")
        grouped_images = []
        image_counter = self.get_start_number_of_image()
        for i in range(0, len(messages)):
            temp = {}
            if limit_datetime is not None and messages[i].date < limit_datetime:…

            if type(messages[i].media) == telethon.tl.types.MessageMediaDocument:…

            if messages[i].message is None:…

            if messages[i].grouped_id:…

            else:…
```

همچنین بیش از ۵ سایت پربازدید خبری بورسی نیز برای گردآوری خبرهای‌شان استفاده شد. برخی از آن‌ها:

Nabzebourse

Sena

boursepress

برای گردآوری خبرهای سایت‌های خبری از ماژول Scrappy استفاده شده است. پوشه‌ی news-sites شامل پرونده‌های مربوط به گردآوری ان‌هاست. نتیجه‌ی گردآوری خبرهای در یک فایل CSV به نام result ذخیره می‌شود. برای هر سایت یک کراولر جداگانه در پوشه‌ی SPiders نوشته شده است.

برای نمونه کد مربوط به گردآوری خبرهای سایت boursepress در زیر نشان داده شده است:

```python
import scrapy
from news_sites.normalize import clean_text
from news_sites.items import NewsSitesItem


class BoursePressSpider(scrapy.Spider):

    name = "boursepress"
    allowed_domains = ["boursepress.ir"]
    start_urls = [
        "https://boursepress.ir/page/archive?category=-1&newstype=-1&fromday=11&frommonth=10&fromyear=1278&today=3&tomonth=6&toyear=1401&count=20&p=1",
    ]

    def parse_page_content(self, response):
        item = NewsSitesItem()
        item['website'] = self.name
        item['title'] = clean_text(response.xpath('//title/text()').get())
        item['published_datetime'] = clean_text(
            response.xpath('//div[@class="news-map"]/div/text()').getall()[1]
            )
        item['body'] = \
            ' '.join(
                [   clean_text(item)
                    for item in
                    response.xpath('//div[@class = "news-text"]/p/text()|//div[@class = "news-text"]/p/a/text()').getall()
                ]
            )
        item['tags'] = ','.join(
            [
                clean_text(item) for item in
                response.xpath('//div[@class = "tags-content"]/a/text()').getall()
            ]
        )
        item['link'] = response.request.url
        yield item

    def parse_archive_page(self, response):
        for link in response.xpath(
            '//ul[@class = "news-list-t"]/li/a/@href'
            ).getall():
            yield scrapy.Request(link, callback=self.parse_page_content)
```

*: همچنین یادآوری می‌شود که پست‌های تلگرامی با استفاده از فیلدهای تعریف شده در فایل models.py در الستیک‌سرچ ذخیره می‌شوند.

گام دوم پیش‌پردازش:

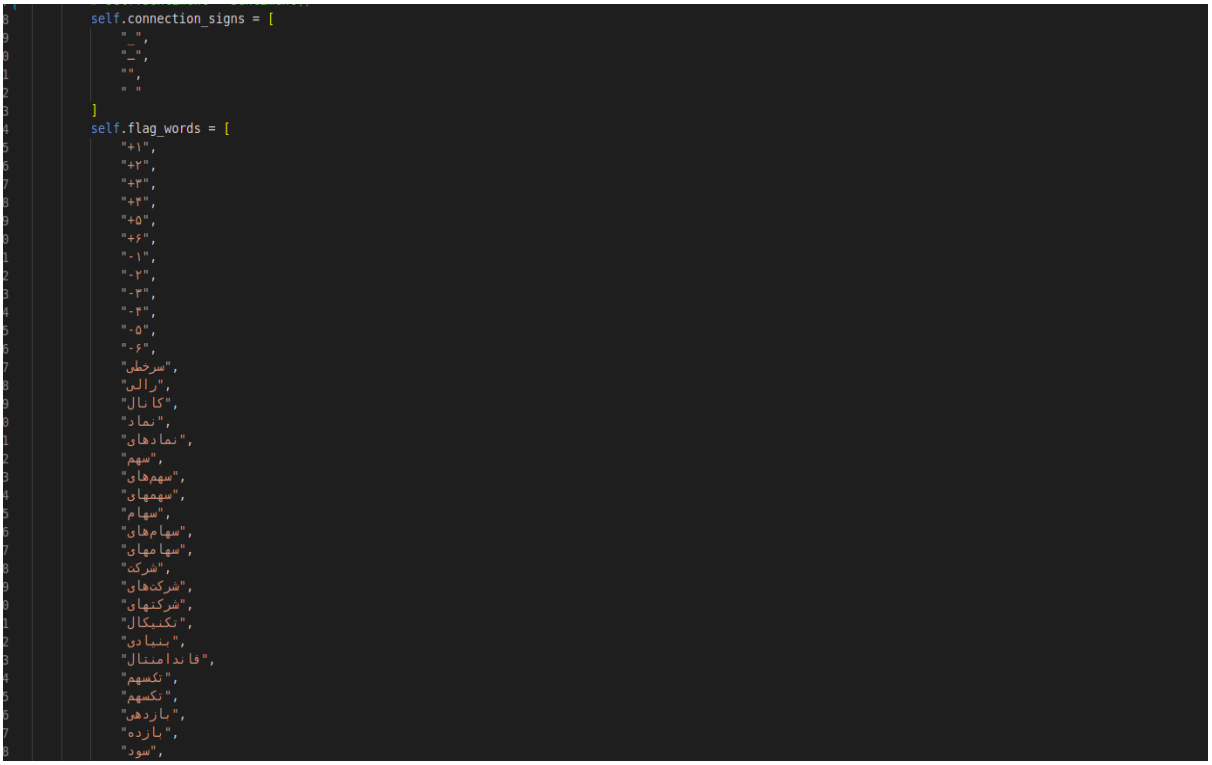
در این مرحله، از پیش‌پردازش‌های رایج مانند حذف نشانه‌های نگارشی، یکسان‌سازی نویسه‌ها و ... استفاده شده است. ولی برخی موارد به پروژه افزوده شده‌اند. برای نمونه حذف ایموجی‌هایی که در پست‌های کانال‌های تلگرامی وجود دارند به پروژه افزوده شده. همچنین کانال‌های تلگرامی دارای برخی الگوهای یکسان در هر پست هستند. برای نمونه برخی از کانال‌ها همیشه در آغاز پست، نام کانال را می‌نویسند. همواره در آخر پست شناسه‌ی کانال قرار دارد. این‌ها نیز حذف می‌شوند.

در گام آخر و پس از آن‌که متن آماده شد، یک پیکره شامل حدود ۱۱۴۸ نماد بورس ایران گردآوری کرده‌ایم.

	A	B	C	D	E	F	G	H	I	J
1	symbol_code	group	industry_group	tablo	english_symbol	glish_symbol_nai	persian_symbol	rsian_symbol_nai	is_certain	_certain_with_rul
2	IRB5IKCO8751	N2	درو و ساخت قطعا	فهرست اولیه	IKCQ1	Iran Khodro-D	۱۸۷۱۹۱۰۱	مشارکت ایران خو	0	0
3	IRO1NBAB0001	N2	ه برق، گاز، بخاروآ	فهرست اولیه	NBAB1	Abadan PG	آبادا	ولید نیروی برق آباد	1	1
4	IRO1APPE0001	N2	فعالیت‌های وابسته	فهرست اولیه	APPE1	san Pardakht Pe	آپ	سان پرداخت پرشیر	0	1
5	IRO1ASIA0001	N1	بازنشستگی به جز	تابلو اصلی	ASIA1	Asia Bime	آسیا	بیمه آسیا	0	1
6	IRO1CONT0001	N1	کی، اپتیکی و اندازه	تابلو فرعی	CONT1	Iran Counter	آکنتور	کنتورسازی‌ایران	1	
7	IRR1CONT0101	N1	کی، اپتیکی و اندازه	تابلو فرعی	CONX1	Iran Counter-R	آکنتورح	ح. کنتورسازی‌ایران	1	
8	IRO1ALMN0001	N2	فلزات اساسی	فهرست اولیه	ALMN1	Alomina Iran	آلومینا	آلومینای ایران	1	
9	IRO1OPAL0001	N2	تخراج کانه‌های فلز	فهرست اولیه	OPAL1	Opal Kani Pars	اپال	ی معدنی اپال کانی	1	
10	IRO1ETKA0001	N2	بازنشستگی به جز	فهرست اولیه	ETKA1	Amin Company	انکام	بیمه انکایی امین	1	
11	IRR1ETKA0101	N2	بازنشستگی به جز	فهرست اولیه	ETKX1	Amin Company-R	انکامح	ح. بیمه انکایی امین	1	
12	ب	N2	ت و خدمات وابسته	فهرست اولیه	ZBAL1	Ajdad Zarbal Co.	اجداد	مرغ اجداد زربال	0	1
13	IRO1MKBT0001	N1	مخابرات	تابلو اصلی	MKBT1	Iran Tele. Co.	اخابر	مخابرات ایران	1	
14	IRR1MKBT0101	N1	مخابرات	تابلو اصلی	MKBX1	Iran Tele Co.-R	اخابرح	ح. مخابرات ایران	1	

این گردآوری به این دلیل است که بسیاری از کانال‌های بورسی، خبرهای مهم سیاسی و ورزشی و حتی برخی وقت‌ها طنزگونه را نیز پوشش می‌دهند. ما با استفاده از این پیکره مشخص می‌کنیم که هر پست مشخص، در مورد کدام نمادها صحبت می‌کند؟ همچنین قابل توضیح است که این پیکره از سایت tsetmc گردآوری شده است. دو ستون به این فایل که توسط کد گردآوری شده اضافه شده و به صورت دستی برای هر نماد بررسی شده است. ستون is_certain نشان‌دهنده‌ی آن است که آیا وجود این کلمه می‌تواند به طور صددرصدی نشان‌دهنده‌ی صحبت کردن آن متن در مورد آن نماد باشد یا خیر؟ برای نمونه ما نمادی به نام برکت داریم. لزومی ندارد که هر پستی که در آن از واژه‌ی برکت استفاده شده است، درباره‌ی نماد برکت

صحبت کرده باشد. بسیاری از نمادها این‌چنینند. در ابتدا این نمادها مشخص شده‌اند. برای نمادهایی که این ستون برای‌شان با مقدار ۰ پر می‌شود، در ستون دیگری مشخص کرده‌ایم که آیا با قوانین می‌توان مشخص کرد که این کلمه نشان‌دهنده‌ی یک نماد بورسی است یا خیر؟ این rule ها در متن کد مشخص شده‌اند که عکس آن‌ها را در زیر می‌بینید:



این قوانین با استفاده از re و جستجوهای خطی در لیست تلاش می‌کنند مشخص کنند که آیا حوالی یک کلمه‌ی مشخص مانند «برکت»، این لیست کلمات وجود دارند؟ برای نمونه window_size پیش‌فرض برابر با ۳ در نظر گرفته شده است. یعنی در صورتی که در فاصله‌ی ۳ کلمه از نماد مورد نظر، کلمه‌ای که در لیست بالا موجود است وجود داشت، در آن صورت واژه‌ی «برکت» نشان‌دهنده‌ی نماد برکت است. در غیر این‌صورت این‌گونه نیست و برکت به عنوان نماد تشخیص داده نمی‌شود. بخشی از کد مربوط به بررسی و تشخیص وجود نماد در پست نیز در زیر نشان داده شده است. تابع get_symbols وظیفه‌ی این کار را دارد.

```
def get_symbols(self, text):
    symbols = []
    words = text.split()
    for symbol in self.symbols:

        if len(symbol.split()) == 1:

            if symbol in words and self.symbols_dict[symbol]["is_certain"]=="1":…

            elif "#" + symbol in words and self.symbols_dict[symbol]["is_certain"]=="1":…

            elif "#" + symbol in words and self.symbols_dict[symbol]["is_certain_with_rules"]=="1"\
            or self.symbols_dict[symbol]["is_certain_with_rules"]=="":…

            elif symbol in words and self.symbols_dict[symbol]["is_certain_with_rules"]=="1"\
            or self.symbols_dict[symbol]["is_certain_with_rules"]=="":…
```

گام سوم؛ تحلیل پست و متن‌ها

۱. نمایش ابری: در این گام چند کار مشخص انجام می‌شود. نخست آن که با استفاده از یک api که با جنگو نوشته شده است، می‌توانید یک تاریخ خاص را انتخاب کنید و نمایش ابری فراوانی تکرار و کاربرد نمادهای بورسی در کانال‌های مختلف در آن زمان خاص را ببینید.

localhost/crawler/wordcloud?date=2022-09-09

خروجی این api یک عکس است که نمایش ابری نمادها در آن تاریخ خاص را نشان می‌دهد.

۲. تحلیل احساسات برای پست مورد نظر:

از مدل T5[1] که در hugginigface وجود دارد برای این کار استفاده شده است. در واقع هنگامی که یک پست کراول می‌شود، مثبت یا منفی بودن آن پست نیز مشخص می‌شود. تابع `get_sentiment` در فایل `telegram_crawler_pipeline` احساس پست را بررسی می‌کند.

۳. بررسی آن‌که هر کانال بیش‌تر در رابطه با کدام نماد صحبت می‌کند! در یک بازه‌ی یک ماهه نتیجه‌ی بررسی کانال‌های تلگرامی نشان می‌دهد که برخی از کانال‌ها علاقه‌مند به صحبت در رابطه با برخی از نمادهای مشخص هستند. برای نمونه پس از استخراج فراوانی نمادها در کانال‌های مختلف مشخص شد که کانال ariyaz_1 بیش‌تر به نمادهای خودرویی مانند «خودرو»، «خمحرکه» و «وساپا» علاقه‌مند است. در حالی که کانال bourseonline تنوع بیش‌تری را حفظ کرده است و سه نماد پربازدید آن از سه گروه مختلفند. «برکت»، «فولاد» و «خساپا».

[1] https://huggingface.co/erfan226/persian-t5-paraphraser

پرتنوع‌ترین کانال نیز کانال tse_fundamental است که در بازه‌ی یک ماه درباره‌ی ۱۳۴ نماد صحبت کرده است!