



**گزارش فاز اول پروژه درس پردازش زبان های طبیعی**

**موضوع : تشخیص دسته بندی موضوع کانال های تلگرامی**

**استاد صالح اعتمادی**

**محمد یارمقدم**

96462104

## منبع داده :

برای جمع آوری داده از کتابخانه telethon در زبان python استفاده شد.

این کتابخانه قابلیت دریافت پیام های کانال های مختلف برانامه telegram را دارد.

در این بخش برای دو دسته ورزشی و خبری تعدادی کانال انتخاب و از هر کدام حدود 3000 پیام دریافت شد.

آدرس کانال ها در کد در فایل های NewsChannelMessages.py و SportChannelMessages.py قرار داده شده است . با

ساخت client و اجرای این دو کد به صورت اتوماتیک دریافت اطلاعات انجام خواهد شد.

آدرس کانال های ورزشی :

- <https://t.me/varzesh3>
- <https://t.me/footfun2020>
- <https://t.me/tarafdari>
- [https://t.me/khabare\\_varzeshi](https://t.me/khabare_varzeshi)
- [https://t.me/khabare\\_varzeshi](https://t.me/khabare_varzeshi)

آدرس کانال های خبری :

- [https://t.me/khabaar\\_ch](https://t.me/khabaar_ch)
- <https://t.me/ohnews>
- <https://t.me/akhbarefori>
- <https://t.me/yjcnewschannel>
- <https://t.me/parsinehnews>

## روش جمع آوری :

در کتابخانه telethon ابتدا یک client در سایت تلگرام بر روی اکانت موردنظر ساخته و سپس hash و id داده شده را در قسمت config جایگذاری می کنیم. سپس شماره و یوزرنیم را جایگذاری میکنیم.

با این کار کد به اکانت تلگرام متصل میشود.

```

1  [[Telegram]]
2  # no need for quotes
3
4  # you can get telegram development credentials in telegram API Development Tools
5  api_id = 3883014
6  api_hash = f832cfe5b84fa8be6dd86d92b9631847
7
8  # use full phone number including + and country code
9  phone = 00989306637036
10 username = MYM_78

```

سپس در مرحله بعد در فایل SportChannelMessages و NewsChannelMesaages آیدی کانال های داده شده را وارد کرده و سپس با ران کردن این دو قسمت توسط دستور python3 NewsChannelMessages.py و python3 SportChannelMessages.py داده های موردنظر در فولدر دیتا در قالب فایل json ذخیره میشوند.

## فرمت داده ها :

در این قسمت ساختار پوشه های فایل پروژه شرح داده میشود.

در فایل src کد های قسمت های مختلف پروژه قرار دارد.

فایل ها عبارت اند از :

- .config
- SportChannelMessages.py
- NewsChannelMessages.py
- Main.py

فایل کانفیگ که برای وارد کردن اکانت تگرام است.

دو فایل بعدی برای گرفتن پیام های کانال های ورزشی و خبری است.

فایل مین هم قسمت اصلی محاسبات است. که در این فایل جداسازی جملات و کلمات و آنالیز های خواسته شده و رسم نمودار هیستوگرام انجام میشود.

در فایل data فایل های txt که مربوط به داده های مختف خروجی است قرار دارد.

## فایل های این پوشه عبارت اند از :

- Clean\_<name>\_sentence
- Clean\_<name>\_data
- Initial\_<name>\_data
- Most\_repeated\_<name>\_data
- <name>\_words
- TF\_IDF\_<name>\_data
- <name>\_data.png

در اینجا به ازای <name> دو مقدار sport و news قرار می گیرد.

در فایل اول، جملات جداسازی شده و مرتب شده و تمیز شده قرار دارد.

در فایل دوم، کلمات جداسازی شده و تمیز شده قرار دارند. در این فایل ایموجی ها و حروف اضافه و نماد های نگارشی حذف شده است.

در فایل سوم، جملات خام و جداسازی بدون اعمال تغییر قرار دارند.

در فایل چهارم، 10 کلمه پر تکرار و غیر مشترک هر دسته به همراه تعداد تکرار آن قرار دارد.

در فایل پنجم هم کلمات خام و جداسازی شده قرار دارد.

در فایل ششم 10 کلمه برتر از نظر TDIDF قرار دارند که مقدار این رابطه هم در کنار هر کدام نوشته شده.

در نهایت عکس های موجود مربوط به نمودار هیستوگرام است.

## برچسب گذاری :

در این پروژه دو برچسب وجود دارد که بر روی پیام های کانال ها گرفته شده قرار داده شده.

چون پیام ها از کانال ها با موضوع های خبری و ورزشی به صورت جدا گرفته میشود، پس برچسب های پیام ها به صورت اتوماتیک بر روی آنها قرار دارد. خروجی فایل هم ابتدا در قالب json در پوشه src قرار دارد.

## پیش پردازش انجام شده:

از کتابخانه parsivar برای کارهای جداسازی کلمات و جملات و نرمال سازی جملات و کلمات استفاده شده است.

از این کتابخانه tokenizer و normalizer گرفته میشود که نرمالایز نیم فاصله های میان جملات و کلمات را حذف می کند و tokenizer هم توابع خاص برای جداسازی کلمات و جملات به صورت جداگانه دارد.

## روش جداسازی جملات و کلمات :

ابتدا متن پیام ها را از فایل json دریافت می کنیم و در یک لیست ذخیره میکنیم.

سپس از دستور زیر استفاده می کنیم :

```
for temp in list_of_sentences:
    item = my_tokenizer.tokenize_sentences(my_normalizer.normalize(temp))
    for line in item:
        clean_sentences.append(clean_sentence(line))
```

در اینجا جملات هر خط نرمال سازی شده و سپس تابع حذف موارد اضافه و جداسازی فراخوانی میشود.

پس از آن بر روی هر جمله مراحل زیر را پیش میبریم :

```
for item in clean_sentences:
    list_words = my_tokenizer.tokenize_words(my_normalizer.normalize(item))
    for temp in list_words:
        list_of_words.append(temp)
```

در ادامه تابع تمیز کردن کلمات صدا زده میشود که ایموجی و موارد اضافه را حذف میکند و سپس اعداد و کلمات انگلیسی حذف میشوند.

101306	تعداد کل جملات
810627	تعداد کل کلمات قبل از تمیزسازی
587022	تعداد کل کلمات بعد از تمیزسازی
7969	تعداد کل کلمات متمایز

آمار های محاسبه شده :

- دسته اول : کانال های خبری

34423	تعداد جملات
383487	تعداد کلمات قبل از تمیزسازی
285462	تعداد کلمات بعد از تمیزسازی
20189	تعداد کلمات متمایز
265273	تعداد کلمات تکراری

- دسته دوم : کانال های ورزشی

66883	تعداد جملات
427140	تعداد کلمات قبل از تمیزسازی
301560	تعداد کلمات بعد از تمیزسازی
18047	تعداد کلمات متمایز
283513	تعداد کلمات تکراری

## مراحل اجرای پروژه :

- در قسمت روت پروژه ابتدا کد python3 SportChannelMessages.py و python3 NewsChannelMessages.py برای دریافت داده از کانال های موردنظر برای این قسمت باید اطلاعات تلگرام را وارد کنید و فایل کانفیگ را پر کرده باشید.
- در قسمت بعد python3 main.py را اجرا میکنیم و تمامی فایل ها موردنظر و آنالیز ها تولید میشود.

قسمت <ز> و <ح> و <ط> نیز در فایل data در فایل txt تولید میشوند.

خروجی های مختلف در صفحه بعد نمایش داده میشود.

خروجی مربوط به روابط TDIFD برای داده های ورزشی و خبری :

TFIDF_news_data - Notepad	TFIDF_sport_data - Notepad
File Edit Format View Help	File Edit Format View Help
است	جمع
226.15	224.2
یک	بیونید
99.85	217.65
شد	لیگ
93.55	200.4
ایران	شد
93.45	154.45
کشور	تیم
73.65	154.2
کرونا	بازی
64.8	147.8
کرد	گل
64.35	112.65
هزار	مقابل
60.75	98.75
میشود	قهرمانان
59.15	90.75
سال	استقلال
56.35	89.85

**خروجی فایل RelativeNormalizedFrequency که بیشترین کلمات مرتبط به دسته خبری را با اعداد بزرگ تر و بیشترین اعداد مرتبط با دسته ورزشی را با اعداد کوچک تر نمایش می دهد.**

RelativeNormalizedFrequency\_data - Notepad

```
File Edit Format View Help
0.012428150527504022
آلومینیوم
0.01221263346633343
دروازهان
0.010256240726580988
نیمکت
0.009781414767017054
رفسنجان
0.009691677016860934
مغلوب
0.009603570862162198
درخشش
0.009603570862162198
رنال
0.008545138886453725
نساجی
0.008384069800300332
مادرید
0.008318053502660171
تاتهام
0.008220955601850908
گهر
0.0076550202524481295
بایرن
0.007223198597181824
گواردیولا
0.007042618632252278
آرسنال
0.006002231788851374
میلان
0.005804356015592538
لالیگا
0.005756908963693961
لژیونرها
0.005362399973796152
مونبخ
0.005362399973796152
دورتموند
0.0036054361598561154
بیونیدید
0.00024268155176610195
```

RelativeNormalizedFrequency\_data - Notepad

```
File Edit Format View Help
|ظریف
308.4666960926498
استانهای
229.23723647981166
سکه
218.67330853143324
وین
181.6995607121088
وزش
177.4739895327574
دانشگاه
175.36120394308173
تحریمها
174.3048111482439
فطر
159.5153120205141
بایدن
147.89499127729783
مسکن
136.2746705340816
انفجار
134.1618849444059
خارجہ
132.40123028634284
روحانی
130.99270655989238
دستگاه
130.99270655989238
امتحانات
128.8799209702167
وقوع
123.59795699602749
هسته‌ای
121.48517140635181
نظامی
113.03402904764908
قوه
110.92124345797339
کارگران
110.92124345797339
ترامپ
105.63927948378418
```