Information Retrieval &

Natural Language Processing

Class 2

JINWOO JEON

HYUNJAE LEE

SEUNGYEOP SEON

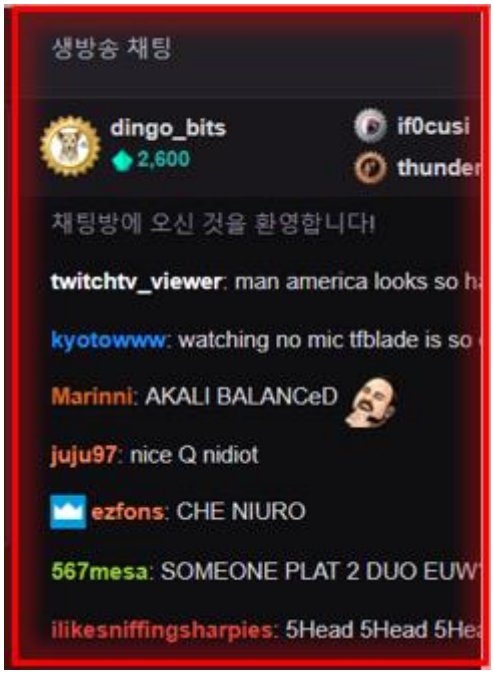
JUNGMIN KIM

Contents

- **Introduction**

1. **Streamer Recommendation.**

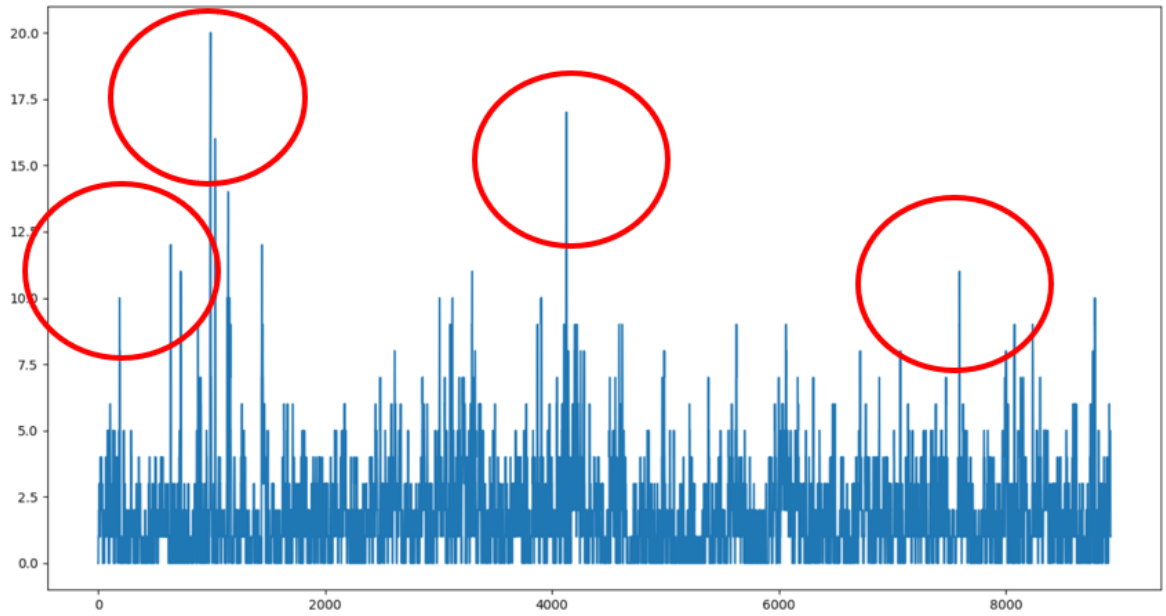
Providing Streamer recommendation on user's input. Use chatlogs to calculate similarity between streamers.



After that, Evaluate the recommendations utilizing followers each streamer has. If the streamer that our program recommended have high same followers rate, it means our recommendation is reasonable.

2. Highlight Extraction

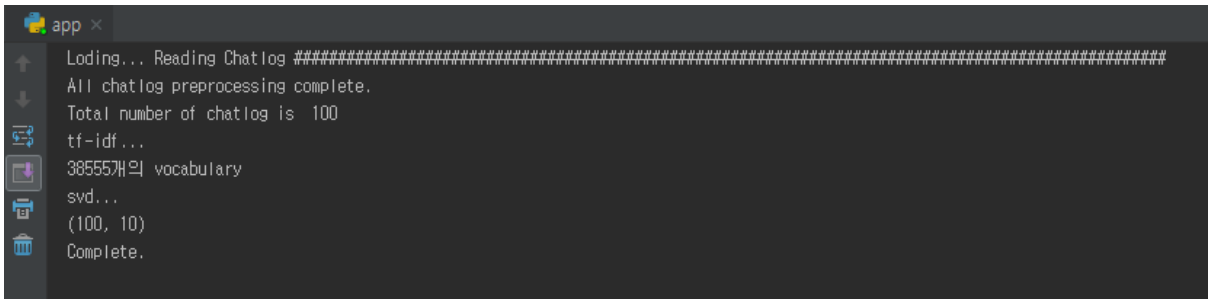
Choose common expression words which frequently exist in highlight videos. And score each timestamp with the number of label words appear cumulatively.



- **About Project**

<Streamer Recommendation>

1. Get Chatlogs and make streamer's vector



2. Remove stopwords

```
# 모든 게임 방송에 자주 등장하는 단어들 [reference : <add your reference>]
stopword = ['pog', 'poggers', 'pogchamp', 'holy', 'shit', 'wow', 'ez', 'clip', 'nice',
            'omg', 'wut', 'gee', 'god', 'dirty', 'way', 'moly', 'wtf', 'fuck', 'crazy', 'omfg']

# 영어 기본 stop words
my_stop_words = text.ENGLISH_STOP_WORDS.union(stopword)

print("tf-idf...")
tf_idf_vectorizer = TfidfVectorizer(
    min_df=3,
    stop_words=my_stop_words,
).fit(corpus)
print(str(len(tf_idf_vectorizer.vocabulary_)) + "개의 vocabulary")
```

After that, we remove stopwords. The stopwords contain our own stopwords list which can distinguish well each of streamers.

3. Pick one streamer

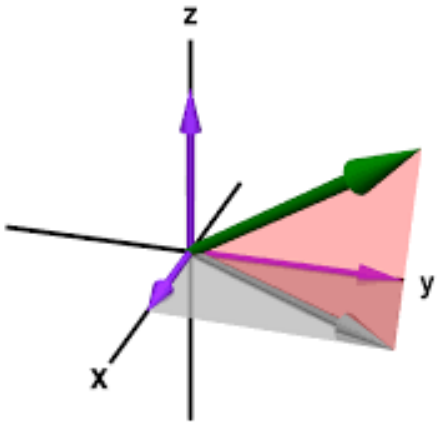
```
==============  We provide various streamers below  ==============

boarcontrolhs
c9sneaky
g2perkz
jordyx3
kingrichard
kolento
lord_kebun
moonmoon_ow
mrfreshasian
ninja
polecat324
purple_hs
rush
summit1g
tfue
thijs
vader
vaporadark
voyboy
zetalot

================================================================
Please input one streamer and we will give you the most similar streamer : voyboy
```

And we will compute similarity based on above selected streamer vector.

4. Measure similarity



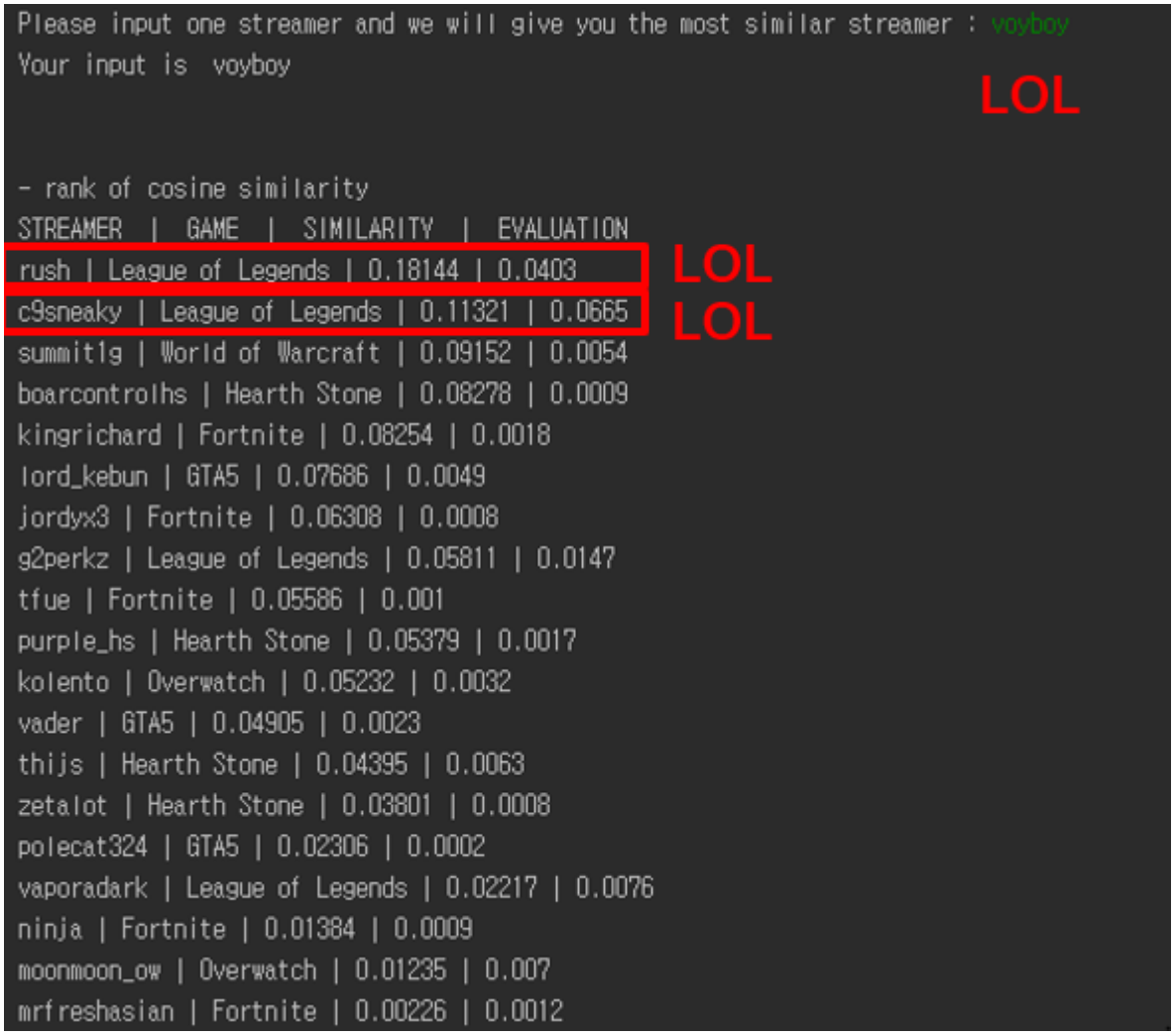
Use cosine similarity and distance similarity, compute the ranks of streamers.

```
Please input one streamer and we will give you the most similar streamer : polecat324
Your input is  polecat324

- rank of cosine similarity
STREAMER  |  GAME  |  SIMILARITY  |  EVALUATION
vader | GTA5 | 0.14594 | 0.0052
boarcontrolhs | Hearth Stone | 0.05489 | 0.0002
kingrichard | Fortnite | 0.0393 | 0.0007
jordyx3 | Fortnite | 0.03515 | 0.0008
g2perkz | League of Legends | 0.02613 | 0.0
tfue | Fortnite | 0.02354 | 0.0004
voyboy | League of Legends | 0.02306 | 0.0002
ninja | Fortnite | 0.02296 | 0.0
purple_hs | Hearth Stone | 0.02266 | 0.0001
c9sneaky | League of Legends | 0.02065 | 0.0002
rush | League of Legends | 0.01852 | 0.0001
kolento | Overwatch | 0.01787 | 0.0
mrfreshasian | Fortnite | 0.01642 | 0.001
vaporadark | League of Legends | 0.0153 | 0.0002
zetalot | Hearth Stone | 0.01474 | 0.0
lord_kebun | GTA5 | 5e-05 | 0.0057
thijs | Hearth Stone | -0.00603 | 0.0002
moonmoon_ow | Overwatch | -0.00922 | 0.0012
summitlg | World of Warcraft | -0.01305 | 0.0008
```

GTA5

GTA5

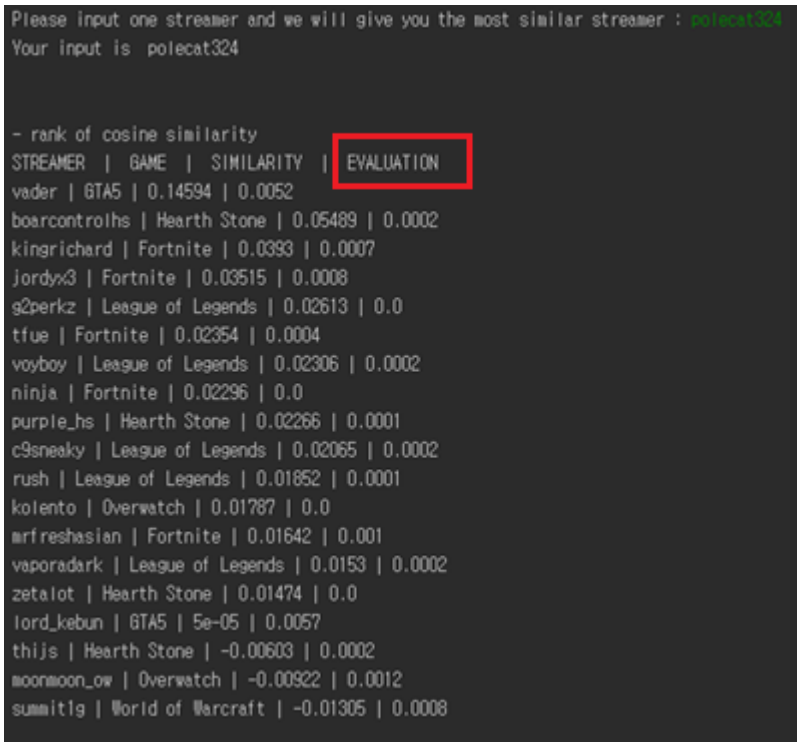


There are streamers playing a same game on high ranks.

5. Evaluate

$$Eval(q,r) = \frac{num(followers(q) \cap followers(r))}{num(followers(r))}$$

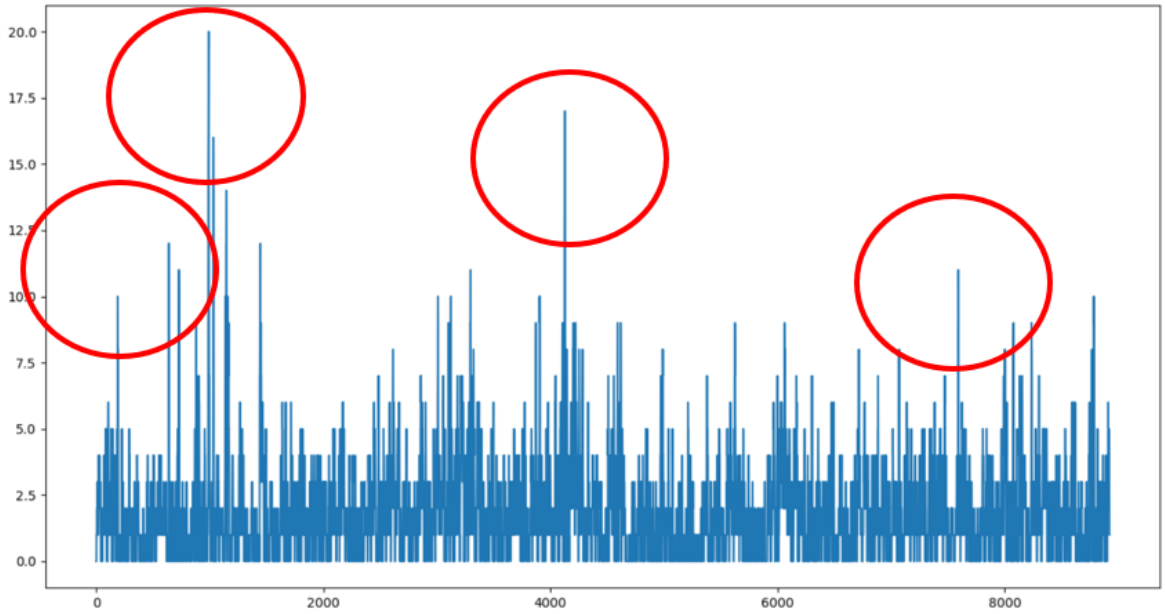
Evaluate the above result using followers. Alphabet q is a state which is consisted of the followers of streamer from user's input. And r is consisted of given followers scraped from the twitch website.



<Highlight Extraction>

1. Make label words from highlight video

```
[Label words]
['pog', 'poggers', 'pogchamp', 'holy', 'shit', 'wow', 'ez', 'clip', 'nice', 'omg', 'wut', 'gee', 'god', 'c
```



Select the words from chatlogs which frequently appeared in highlight videos.

2. Get parameters from user

```
 ======================
 Chat log Analyze START
 ======================
[numOfHighlights] The number of expected highlights for each chatlog
Please input your numOfHighlights : 5
[cummulative_sec] How many next seconds you want to consider for chat analyzing
Please input your cummulative_sec : 5
[delay] How long each highlight section is suppposed to be
Please input your delay : 5
```

First parameter is the number of highlights. Second parameter indicates the unit time of the chat log to be used for the highlight calculation. Third parameter means highlight time of each videos.

3. Score each timestamp

```
[(1) : Chat analyze result]
{'[2:44:03]': 0.75, '[2:44:09]': 1.0, '[2:44:11]': 1.0, '[2:44:17]': 1.0, '[2:44:24]': 0.75}
Merge List :  {9843: 9843, 9849: 9851, 9857: 9857, 9864: 9864}
Will be deleted :  [2]
```

Count cumulatively the number of label word appears.

4. Merge each of highlights

```
 << Highlight result for the chatlog C:₩Users₩Faust₩PycharmProjects₩TWIT₩data₩voyboy₩426328564.txt belonge
[['02:43:53', '02:44:03'], ['02:43:59', '02:44:11'], ['02:44:07', '02:44:17'], ['02:44:14', '02:44:24']]
```

Merge if there are more than two conflicting sections.

5. Example

[Example]

00:05:03 : **10**,
00:05:05 : **10 + a**,
00:05:06 : **9 + a + b**,
00:05:08 : **7 + a + b + c**,
00:05:13 : **4 + a + b + c + d**,
...

Each second has a score value
that counted at the specific moment

But, we decided to consider correlations

- **Additional information**

Programming Language: Python 3.7

Open Source: NLTK, TCD

Open API: Twitch API v5

https://github.com/twit-cau