دانشگاه علم و صنعت ایران

Natural Language Processing
Project Proposal

Project Title: Effects of Seasonal Change on Mental Health

Instructor: Dr. Sauleh Eetemadi

Sara Kodeiri

Summer 2021

Setup and Environment

For this phase of the project, I decided to use Google Colab to use the needed libraries easier, specially Tweepy. This way I didn't need to have my VPN on my local machine running all the time. Also, keeping all the data on a cloud made more sense to me as I needed to download them after all the modifications were over, and push it to Github only once. All the required libraries are imported in the first section of the Jupyter notebook and the ones that were not already available have been installed.

Data Gathering

Since I chose twitter as the form of data, I decided to use the Tweepy library. Using my account's secret tokens, I was able to log into twitter API and get the tweets I could. The API has a limit of the most recent 3200 tweets per user, so by getting the tweets at the right time, I was able to have a semi-normalized dataset among winter and spring.

First, I logged in and checked the connection by getting my timeline printed. Afterwards I got all tweets of 24 people who I knew tweeted in Farsi, and were frequent enough to be able to detect Seasonal Adhesive Disorder from what they said. In addition to the tweets themselves, I got their IDs, time of creation, number of likes, and whether it was retweeted or an original tweet from the user themself. Some of this info might not be useful right now, but might be in the future if this project expands to a society instead of individual users. All of these tweets have been saved in csv files in a folder named "raw", as they have not been modified in any way.

Cleaning Data

This was the most challenging part of this phase. Cleaning Persian data is something I didn't even know how to think about. I removed any tweet that was before the timeframe I needed, removed links and mentions from tweet bodies, detected the language using the langdetect library, and normalized the text using the hazm library. The cleaning process was done for each user individually and all the cleaned data is stored in csv files in a folder named "cleaned". After this was done, for both the raw and cleaned tweets, I combined the csv files into one and sorted them all by date. To keep the anonymity of the users, only these two files (allraw, and allcleaned) are available in the repository. To see how much data was cleaned, I took the size of these two csv files:

```
Raw tweet count: 77196
Cleaned tweet count: 32513
```

It's not a surprise that more than half of the tweets were dropped, because even though I chose Iranian people who are fluent in Farsi, none of them are monolinguals and have a lot of English tweets and retweets.

Classification and Tokenization

There were three things to do for this part: Determine which class each tweet belonged to, breaking the tweet body to word, and also sentences. For this reason, I added three new columns to the dataframe: “class”, “sentences" and “words". The classification was quite simple because the dataframe was already sorted by time. All the tweets before March 21st were marked as “cold”, and the rest as “warm”, indicating a rough estimation of how the weather was when the tweet had been made, in the “class” column for each separate tweet. (So, naturally, each tweet belongs to a label and the labeling unit is a tweet.)

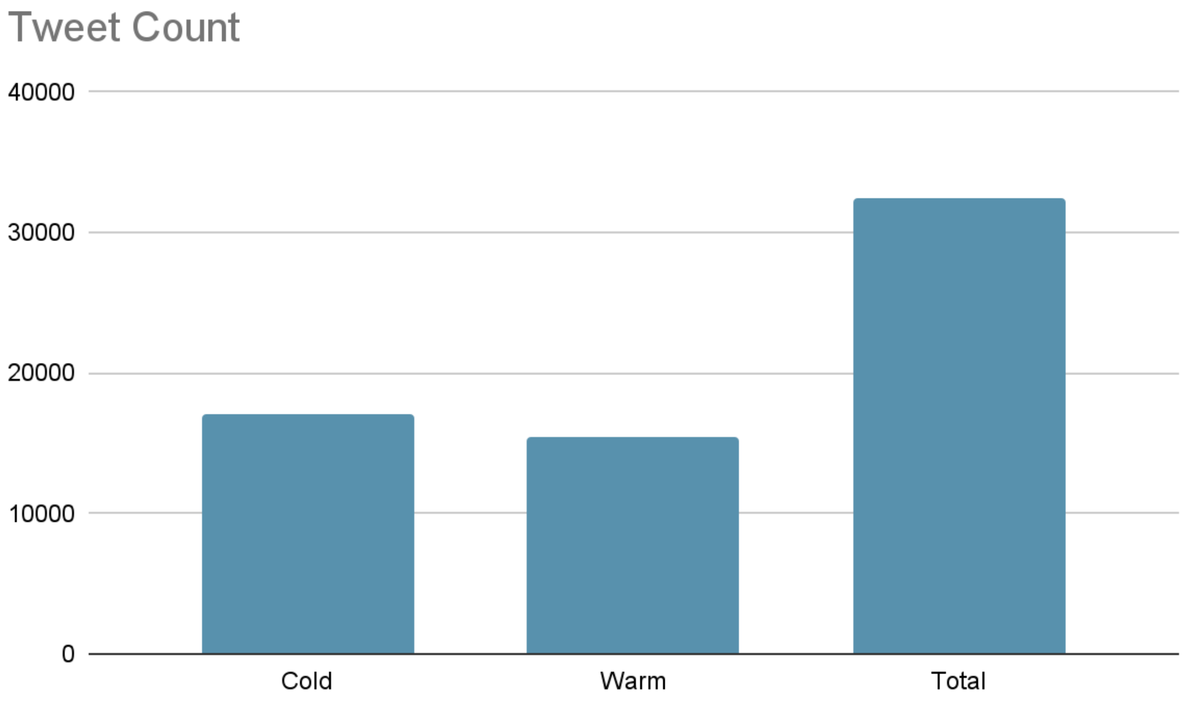
For breaking the tweets into sentences and words, I used the hazm library again and it worked pretty well. For each tweet, the separate words and sentences were added in front of them in their respective columns in the form of a list.

At the end of this part, I saved the new dataframe with the three additional columns in a new file called “allcleaned_complete.csv” which can be accessed through data\clean. This is the final version of the data that will be used for future analysis.

Statistics

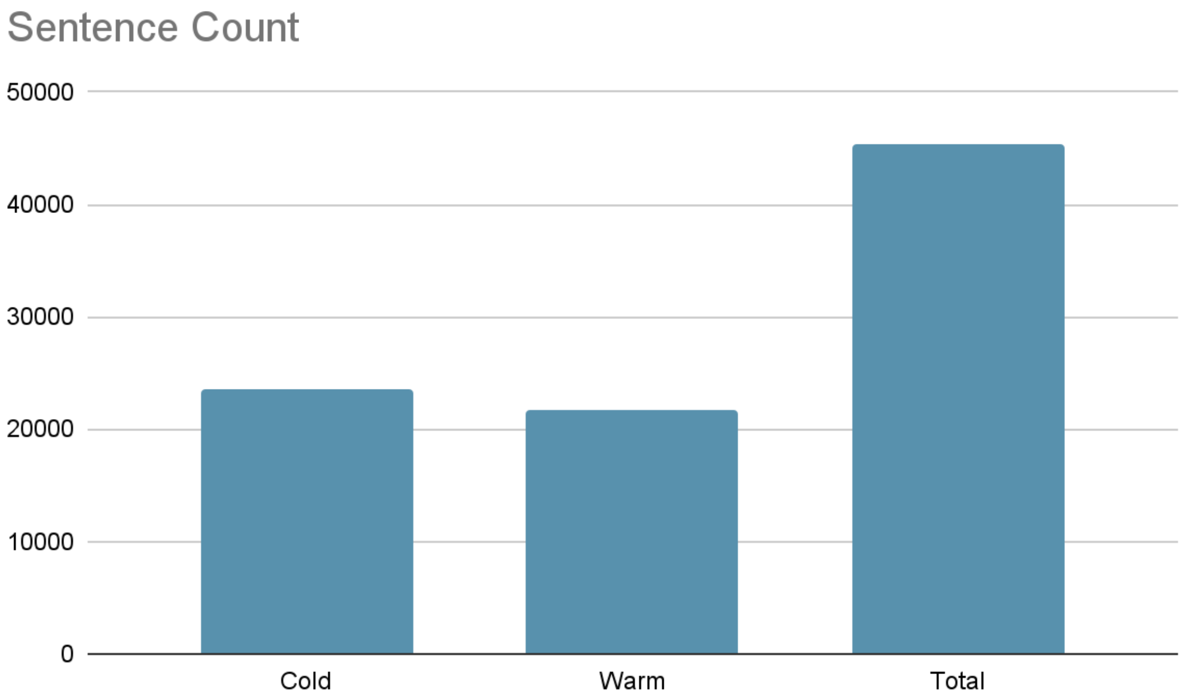
Tweet Statistics

Cold Tweet Count	17059
Warm Tweet Count	15454
Total Tweet Count	32513



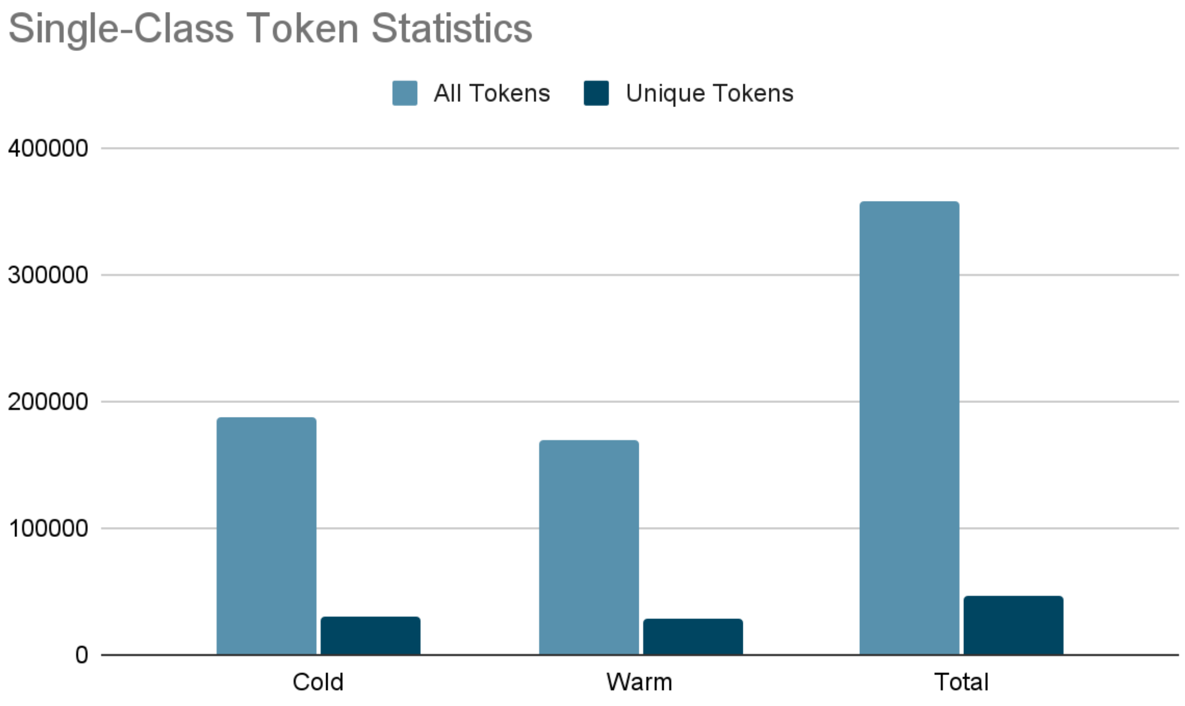
Sentence Statistics

Cold Sentence Count	23583
Warm Sentence Count	21692
Total Sentence Count	45275



Single-Class Word Statistics

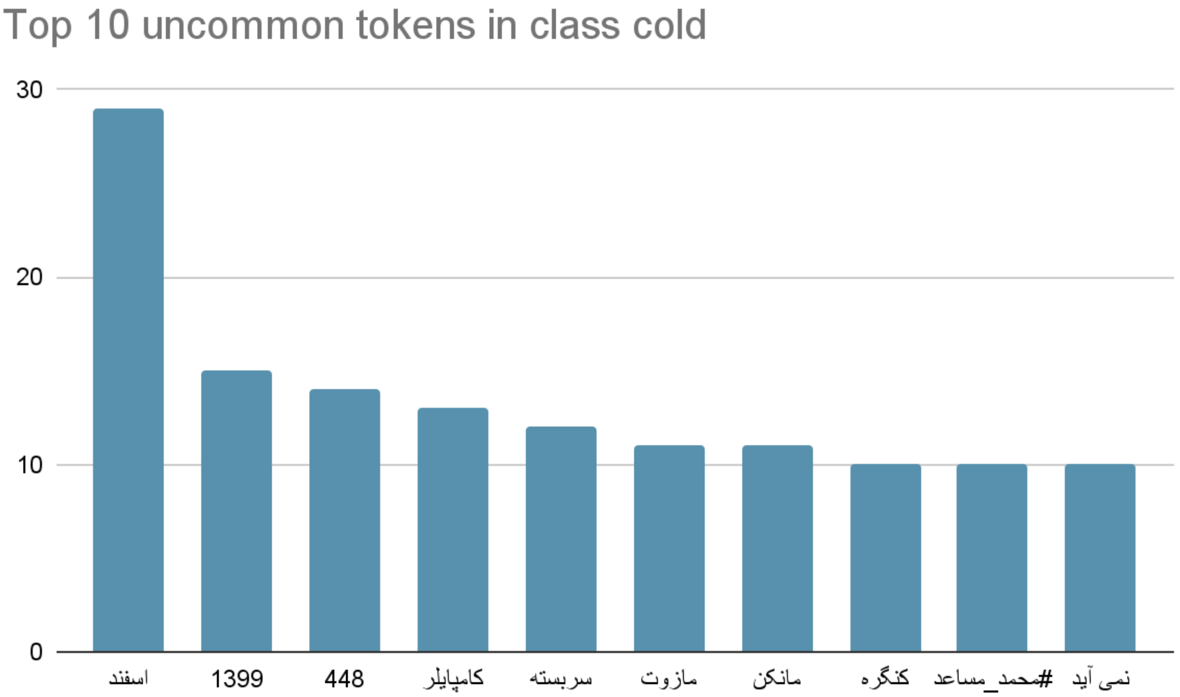
Class	All Words Count	Unique Words Count
Cold	187955	29799
Warm	169858	29323
Total	357813	46640



Inter-Class Word Statistics

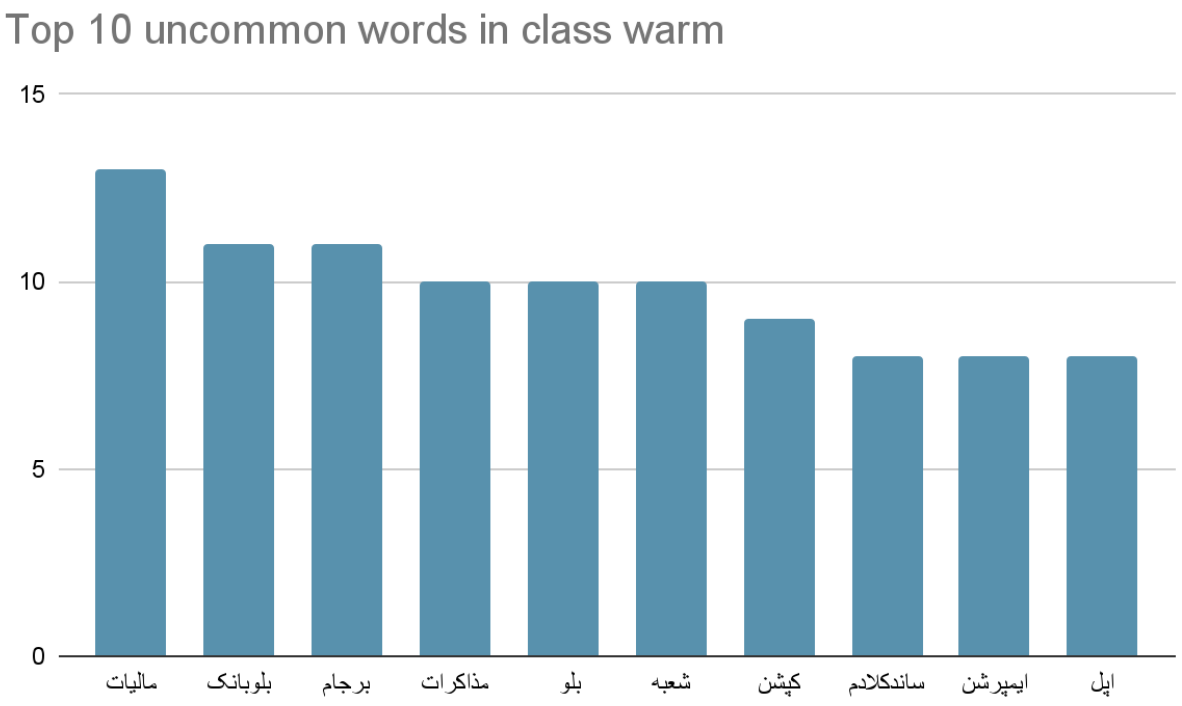
Top 10 uncommon words in class cold:

Ranking	Count	Word
1	29	اسفند
2	15	۱۳۹۹
3	14	۴۴۸
4	13	کامپایلر
5	12	سربسته
6	11	مازوت
7	11	مانکن
8	10	کنگره
9	10	#محمد_مساعد
10	10	نمی آید



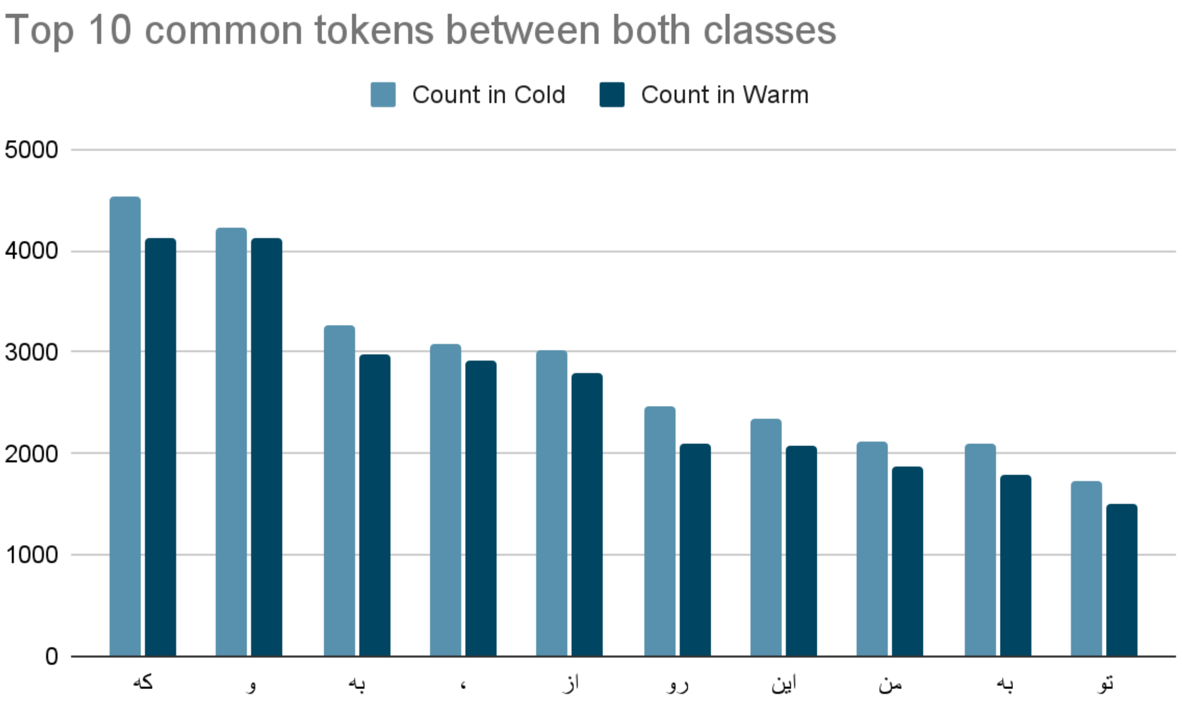
Top 10 uncommon words in class warm:

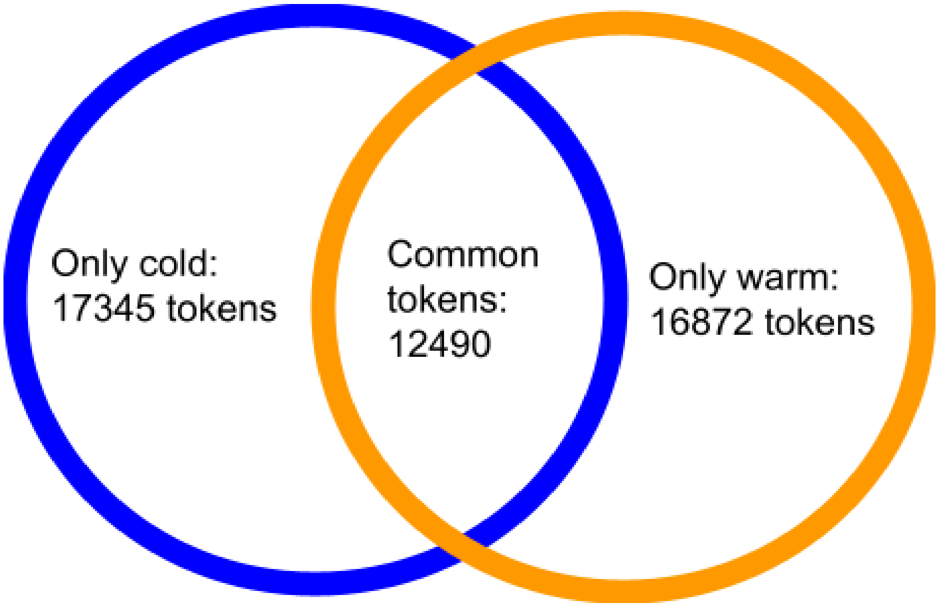
Ranking	**Count**	**Word**
1	13	مالیات
2	11	بلوبانک
3	11	برجام
4	11	مذاکرات
5	10	بلو
6	10	شعبه
7	9	کپشن
8	8	ساندکلادم
9	8	ایمپرشن
10	8	اپل



Top 10 common words (tokens) between both classes:
The tokens were the same, even in ranking. Only the count differed between the two classes.

Ranking	Count in Cold	Count in Warm	Token
1	4542	4135	که
2	4236	4134	و
3	3256	2972	به
4	3090	2916	،
5	3028	2787	از
6	2457	2092	رو
7	2346	2085	این
8	2125	1876	من
9	2097	1801	به
10	1735	1501	تو



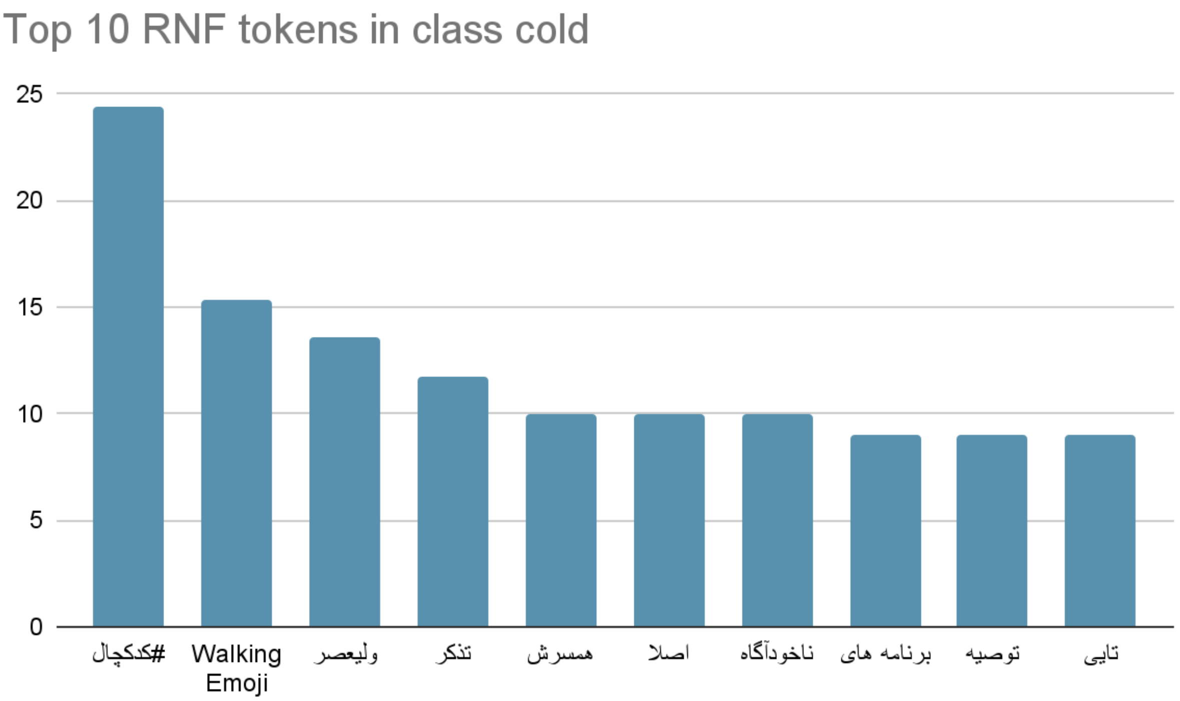


Total number of unique tokens: 46640

Relative Normalized Frequency

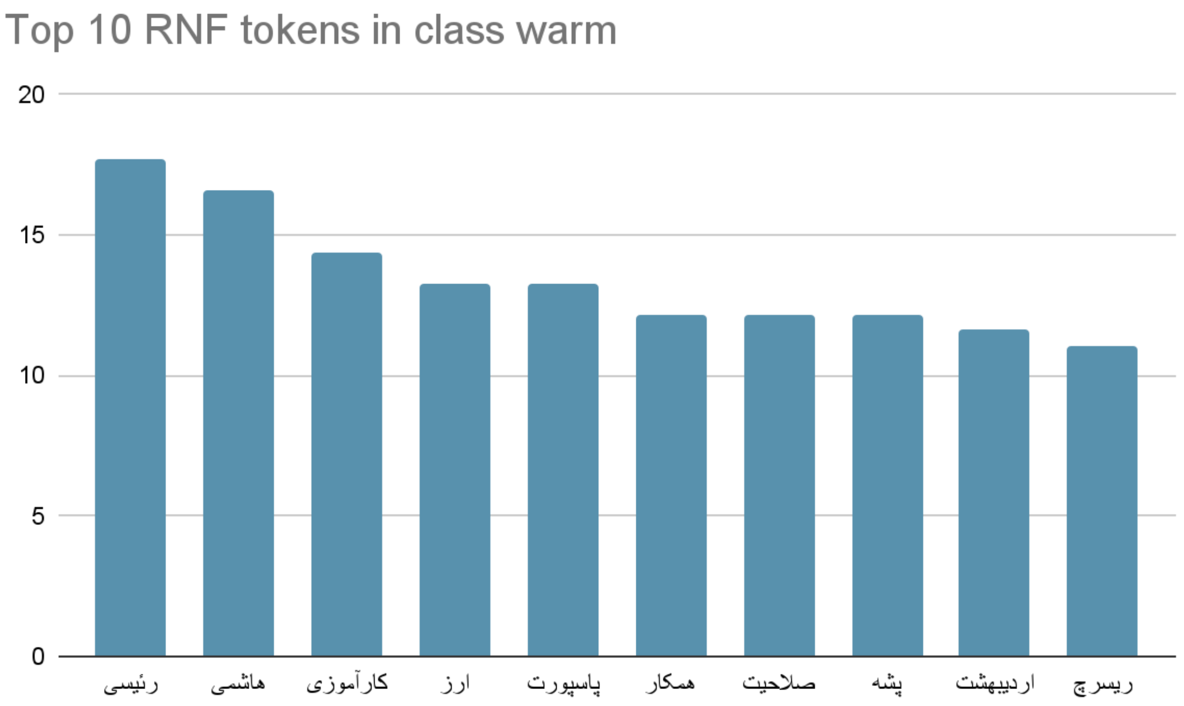
Top 10 common words (tokens) in class cold based on their RNF value:

Ranking	RNF Value	Token
1	24.40	#کدکچال
2	15.36	🚶
3	13.55	ولیعصر
4	11.74	تذکر
5	9.94	همسرش
6	9.94	اصلا
7	9.94	ناخودآگاه
8	9.03	برنامه های
9	9.03	توصیه
10	9.03	تایی



Top 10 common words (tokens) in class warm based on their RNF value:

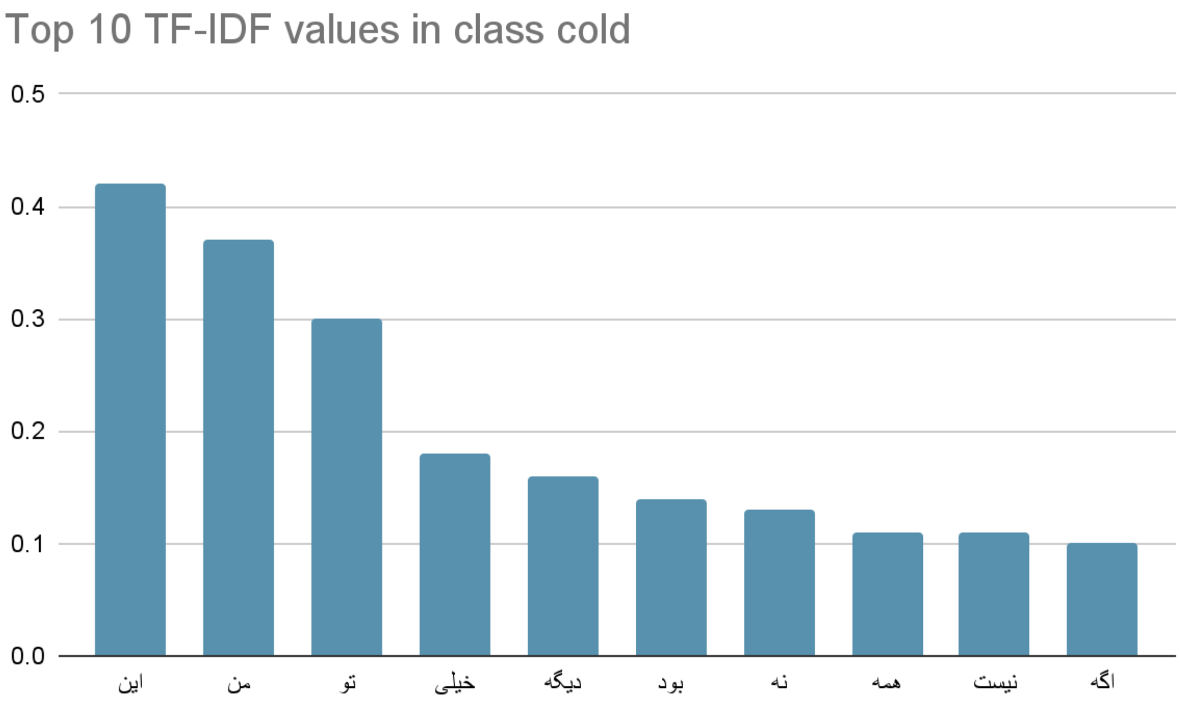
Ranking	RNF Value	Token
1	17.70	رئیسی
2	16.59	هاشمی
3	14.38	کارآموزی
4	13.27	ارز
5	13.27	پاسپورت
6	12.17	همکار
7	12.17	صلاحیت
8	12.17	پشه
9	11.61	اردیبهشت
10	11.06	ریسرچ



TF-IDF

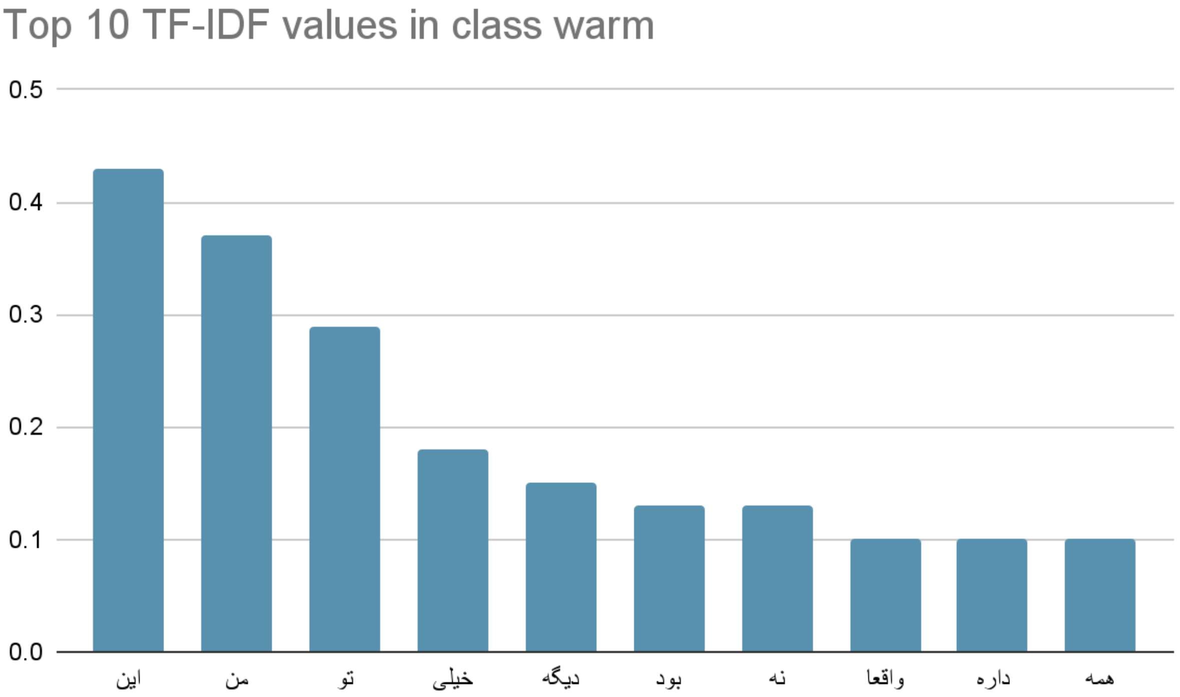
Top 10 common words (tokens) in class cold based on their TF-IDF value:

Ranking	**TF-IDF value**	**Token**
1	0.42	این
2	0.37	من
3	0.30	تو
4	0.18	خیلی
5	0.16	دیگه
6	0.14	بود
7	0.13	نه
8	0.11	همه
9	0.11	نیست
10	0.10	اگه



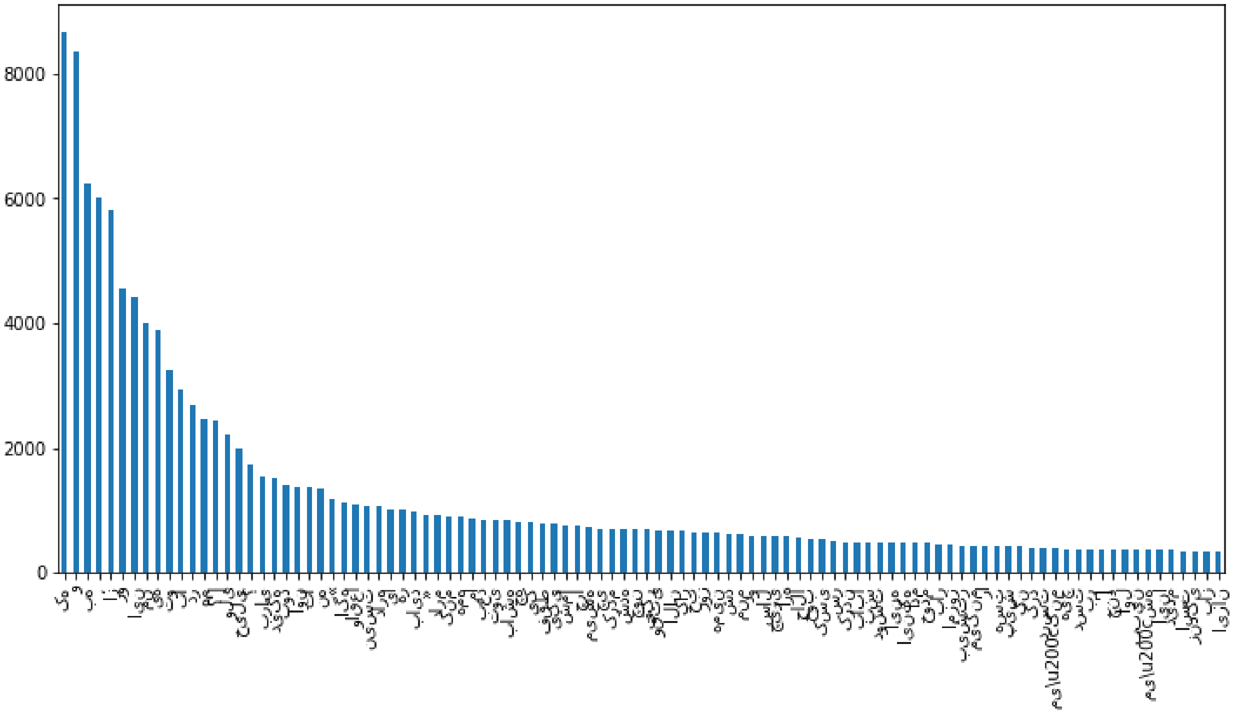
Top 10 common words (tokens) in class warm based on their TF-IDF value:

Ranking	**TF-IDF value**	**Token**
1	0.43	این
2	0.37	من
3	0.29	تو
4	0.18	خیلی
5	0.15	دیگه
6	0.13	بود
7	0.13	نه
8	0.10	واقعا
9	0.10	داره
10	0.10	همه



Histogram

Because there are a lot of words in the corpus, I decided to draw the histogram for the first 100 tokens only. The result is as follows:



	Word	Count
0	که	8677
1	و	8370
2	به	6228
3	.	6006
4	از	5815
5	رو	4549
6	این	4431
7	من	4001
8	یه	3898
9	تو	3236
10	با	2934
11	در	2688
12	هم	2471
13	؟"]	2449
14	ولی	2211
15	خیلی	1989
16	؟	1730
17	برای	1543
18	دیگه	1504
19	بود	1390
20	اون	1379
21	تا	1366
22	نه	1357
23	«	1192
24	اگه	1132
25	واقعا	1086
26	نیست	1069
27	داره	1067
28	یا	1005
29	هر	999
30	باید	974
31	»	925
32	دارم	923
33	کنم	902
34	همه	897
35	ما	869
36	بعد	851
37	توی	844
38	باشه	839
39	چه	818
40	یک	804
41	فقط	788
42	یکی	783
43	شما	765
44	چرا	765
45	میشه	739
46	چی	716
47	کردم	716
48	شده	707
49	چون	706
50	فکر	694
51	وقتی	686
52	الان	682
53	کار	665
54	خب	656
55	روز	649
56	همین	648
57	شد	629
58	منم	614
59	دو	592
60	سال	586
61	چیزی	584
62	آره	581
63	حالا	571
64	خوب	532
65	کسی	531
66	سر	497
67	کردن	489
68	بابا	482
69	پس	481
70	دوست	476
71	اینه	474
72	اینکه	474
73	آدم	468
74	خودم	466
75	بار	442
76	امروز	439
77	بیشتر	435
78	میکنم	435
79	را	428
80	هست	427
81	پیش	426
82	کن	423
83	کرد	407
84	درست	406
85	می\u200cکنم	401
86	چیه	380
87	دست	378
88	بر	378
89	...']	376
90	چند	374
91	اول	370
92	ببین	370
93	می\u200cشه	364
94	اینا	357
95	دیدم	356
96	است	354
97	زندگی	353
98	باز	351
99	ایران	351

References

1. LangDetect https://pypi.org/project/langdetect/
2. Hazm https://www.sobhe.ir/hazm/
3. Scikit Learn TF-IDF https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
4. Python Collections, Counter https://docs.python.org/3/library/collections.html