In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Image
```

In [2]:
```
users=pd.read_excel('overview_stages_and_users.xlsx')
```

In [3]:
```
df=pd.read_excel('DATASET_1.xlsx')
del df['Unnamed: 0']
```

Introduction

The examinated dataset was created in 2020, for the exact dates see the dataframe below.

In [4]:
```
df.groupby('stage').savingTime.min().to_frame().join(df.groupby('stage').savingTime.max().to_frame(), lsuffix='stage').rename(col
```

Out[4]:

stage	start	stop
1	2020-02-13T10:56:34.059Z	2020-03-27T11:06:58.310Z
2	2020-02-13T11:07:54.149Z	2020-03-27T10:24:26.975Z
3	2020-02-13T12:59:21.586Z	2020-03-27T11:41:27.604Z

Stages and user variables in experiment

In [5]:
```
users.columns
```

Out[5]:
```
Index(['stage', 'user_name', 'browser', 'real_location', 'IP_location',
       'full_video_watch', 'history'],
      dtype='object')
```

In [6]:
```
users
```

Out[6]:

	stage	user_name	browser	real_location	IP_location	full_video_watch	history
0	first	User 1	Brave	Russia	USA	full video watch	zero history
1	first	User 2	Brave	Belgum	USA	full video watch	zero history
2	first	User 3	Brave	Italy	USA	full video watch	zero history
3	first	User 4	Brave	Germany	USA	full video watch	zero history
4	first	User 5	Chrome	Germany	Switzerland	full video watch	zero history
5	first	User 6	Brave	Germany	USA	full video watch	zero history
6	first	User 7	Chrome	Germany	France	full video watch	zero history
7	first	User 8	Brave	Germany	USA	full video watch	zero history
8	first	User 9	Chrome	Germany	Germany	full video watch	zero history
9	first	User 10	Brave	Germany	USA	full video watch	zero history
10	first	User 11	Chrome	Germany	France	full video watch	zero history
11	second	User 12	Chrome	Germany	no VPN	no full video watch	zero history
12	second	User 13	Brave	Germany	USA	no full video watch	zero history
13	second	User 6	Brave	Germany	USA	no full video watch	zero history
14	second	User 14	Modzilla	Germany	USA	no full video watch	zero history
15	second	User 15	MS Edge	Germany	USA	no full video watch	zero history
16	second	User 7	Chrome	Germany	no VPN	no full video watch	zero history
17	third	User 12	Chrome	Germany	no VPN	no full video watch	zero history
18	third	User 7	Chrome	Germany	no VPN	no full video watch	zero history
19	third	User 13	Brave	Germany	USA	no full video watch	zero history
20	third	User 6	Brave	Germany	USA	no full video watch	zero history
21	third	User 14	Modzilla	Germany	USA	no full video watch	zero history
22	third	User 15	MS Edge	Germany	USA	no full video watch	zero history

1. number of users at different stages

In [7]:
```
users.groupby('stage').user_name.nunique().to_frame()
```

Out[7]:

stage	user_name
first	11
second	6
third	6

2. used browser

In [8]:
```
users.groupby('stage').browser.value_counts().to_frame()
```

Out[8]:

stage	browser	browser
first	Brave	7
	Chrome	4
second	Brave	2
	Chrome	2
	MS Edge	1
	Modzilla	1
third	Brave	2
	Chrome	2
	MS Edge	1
	Modzilla	1

3. Use of VPN and changing of IP

In [9]:
```
users.groupby(['stage', 'real_location']).IP_location.value_counts().to_frame()
```

Out[9]:

			IP_location
stage	real_location	IP_location	
first	Belgum	USA	1
	Germany	USA	4
		France	2
		Germany	1
		Switzerland	1
	Italy	USA	1
	Russia	USA	1
second	Germany	USA	4
		no VPN	2
third	Germany	USA	4
		no VPN	2

4. watched video in full lenght or not

In [10]:
```
users.groupby('stage').full_video_watch.value_counts().to_frame()
```

Out[10]:

		full_video_watch
stage	full_video_watch	
first	full video watch	11
second	no full video watch	6
third	no full video watch	6

Characteristics of the first dataset

In [11]:
```
df.shape
```

Out[11]:
```
(2971, 26)
```

In [12]:
```
df [:2]
```

Out[12]:

	take	recommendedVideoId	savingTime	id	watcher	recommendedVerified	recommendationOrder	watchedId	watchedAu
0	1	0sZRR3cEGbs	2020-02-14T13:21:55.588Z	7c090fc9afa462983f19	User 1	True	16	iJRHMJZpLNM	Вселе Ист
1	1	8fVoQlYyVug	2020-02-14T13:21:55.588Z	7c090fc9afa462983f19	User 1	True	8	iJRHMJZpLNM	Вселе Ист

2 rows × 26 columns

Inspect data gaps

Due to technical problem with browser extension Tracking Exposed for YT data of some users are not complete for every take.

In [13]:
```
df.groupby(['stage', 'take']).watcher.nunique().to_frame()
```

Out[13]:

		watcher
stage	**take**	
1	**1**	10
2	**11**	11
	12	11
	13	11
	14	11
	15	7
	16	11
	17	11
3	**21**	5
	22	11
	23	11
	24	11
	25	11
	26	10
	27	7

Data from users are missing in 1, 15, 21, 27. How many recommendations are not collected?

In [14]:
```
df.groupby(['stage', 'take']).size().to_frame().rename(columns={0: 'num_obs'})
```

Out[14]:

stage	take	num_obs
1	1	200
2	11	220
	12	220
	13	218
	14	217
	15	139
	16	220
	17	219
3	21	100
	22	219
	23	219
	24	220
	25	220
	26	200
	27	140

Takes 15, 21 and 27 miss 30%-50% of recommended. To make comparisons between different stages, I can use relation to the number of observations.

Sort dataset after stages and takes (watched videos) and inspect number of unique recoomendations received after each take.

Below, you see the list of videos watched at each take in order of the experiment. Do recommendations users received after watching these videos contain clusters? Presence of a cluster might be indicated by more repetitive recommendations, when unique video is recommended more that one time.

In [15]:
```
unique=df.groupby('stage').recommendedVideoId.nunique().to_frame().join(df['stage'].value_counts()).rename(columns={'recommendedV
unique['in percent']=unique['unique videos']/unique['num_obs']*100
unique
```

Out[15]:

stage	unique videos	num_obs	in percent
1	79	200	39.500000
2	477	1453	32.828630
3	510	1318	38.694992

On the second stage, after watching pro-annexation videos, users became more repetitive recommendations (the percentage of unique to the number of observations is lower by 7%). To get a more differentiated view, I look at the number of unique videos across takes and also later in relation to the order of recommended (within the top three and first recommended).

In [16]:
```
takes=df.groupby(['stage', 'take', 'watchedId',  'watchedTitle', 'watchedAuthor']).recommendedVideoId.nunique().to_frame().join(
takes['unique in percent to observations']=takes['unique']/takes['num_obs']*100
```

In [17]:
```
takes
```

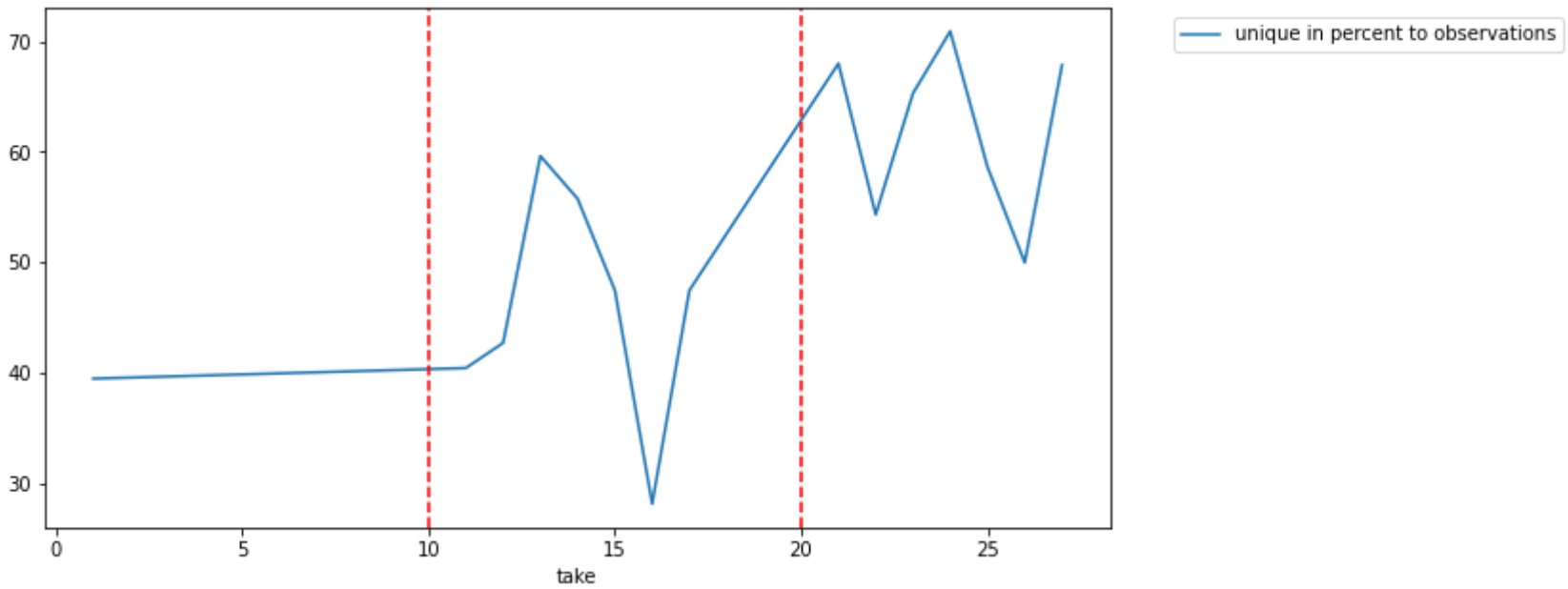
Out[17]:

					unique	num_obs	unique in percent to observations
stage	**take**	**watchedId**	**watchedTitle**	**watchedAuthor**			
1	**1**	**iJRHMJZpLNM**	**ЧЕЙ КРЫМ?**	**Вселенная Истории**	79	200	39.500000
2	**11**	**V7VXMU6QStc**	**Крым наш? Чей же Крым на самом деле? Объективно.**	**Политика Сегодня**	89	220	40.454545
	12	**hR_fIlxeijY**	**Вежливые люди в Крыму!**	**eleven79**	94	220	42.727273
	13	**oFegVdVIxRo**	**Крым - не наш, 〃 русских надо убивать 〃 : о формуле либералов, за которую нас посадят. А их - нет**	**Delyagin TV**	130	218	59.633028
	14	**QMAVnAvWv2k**	**Путин: После тяжёлого плавания Крым и Севастополь возвращаются в родную гавань**	**RT на русском**	121	217	55.760369
	15	**89VrhvBEzUY**	**Путин в Севастополе 40 тысяч на площади - Крым наш!**	**Dimitro Sevastopol**	66	139	47.482014
	16	**TJurAAfOXsc**	**Крым.Наш.Россия.История событий.Присоединение.**	**Denis.G.Film**	62	220	28.181818
	17	**i2pN8F4xOGk**	**Чей Крым? Крым наш!!**	**Anna News**	104	219	47.488584
3	**21**	**d2II8CxDKWI**	**〃 РУССКИЙ 〃 КРЫМ - МИФ для быдла! (о чем молчат российские СМИ)**	**hrendyabliki**	68	100	68.000000
	22	**DcOt7pn6vxo**	**Крым пожалел о приходе русского мира**	**Новости СВЕРХДЕРЖАВЫ**	119	219	54.337900
	23	**TLZXrZxyZms**	**КРЫМ НАШ? Россия НЕ ПРИЗНАЕТ Крым частью своей территории..**	**Реальная журналистика**	143	219	65.296804
	24	**eXSnvNxAy4A**	**Реальная журналистика. КРЫМ нах..? Больше не НАШ?**	**Real Russian News // Реальная журналистика**	156	220	70.909091
	25	**_XRS5HKsAgM**	**Почему Крым не Россия?**	**Ложь путинского режима**	129	220	58.636364
	26	**AmGL8Cwprx8**	**Как Россия оккупировала Крым. Хронология**	**Крым. Реалии**	100	200	50.000000
	27	**SEcQK1_Fdyc**	**РЖД: КРЫМ НЕ НАШ! МЕНЬШЕ ПАРКОВ, БОЛЬШЕ «СКВЕРНЫХ» ХРАМОВ!**	**КЛИРИК**	95	140	67.857143

In [18]:
```
takes=takes.reset_index()
```

In [19]:
```
ax=takes[['take', 'unique in percent to observations']].set_index('take').plot(figsize=[10,5])
ax.axvline(10, color="red", linestyle="--")
ax.axvline(20, color="red", linestyle="--")
plt.legend(bbox_to_anchor=(1.05, 1))
```

Out[19]:
```
<matplotlib.legend.Legend at 0x18cf21c5cd0>
```



Drops in the middle of the plot indicate the presence of clusters (more repetitions among recommended, less unique videos).

Inspect the number of unique recommendations in stages and takes.

I will examine these numbers across the whole dataset, within the top three recommended and within the first recommended.

In [20]:
```
unique_across_takes=df.groupby('take').size().to_frame().rename(columns={0: 'obs_total'}).join(
    df.loc[df['recommendationOrder']<=3].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommendedVideoI
    df.loc[df['recommendationOrder']<=1].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommendedVideoI
    df.groupby('take').recommendedVideoId.nunique().to_frame().rename(columns={'recommendedVideoId': 'total_unique'})).join(
    df.loc[df['recommendationOrder']<=3].groupby('take').recommendedVideoId.nunique().to_frame().rename(columns={'recommendedVid
    df.loc[df['recommendationOrder']<=1].groupby('take').recommendedVideoId.nunique().to_frame().rename(columns={'recommendedVid
unique_across_takes['total_unique_in_percent']=unique_across_takes['total_unique']/unique_across_takes['obs_total']*100
```

```
unique_across_takes['top_three_unique_in_percent']=unique_across_takes['top_three_unique']/unique_across_takes['obs_top_three']*
unique_across_takes['first_unique_in_percent']=unique_across_takes['first_unique']/unique_across_takes['obs_first']*100
```

In [21]:
```
unique_across_takes
```

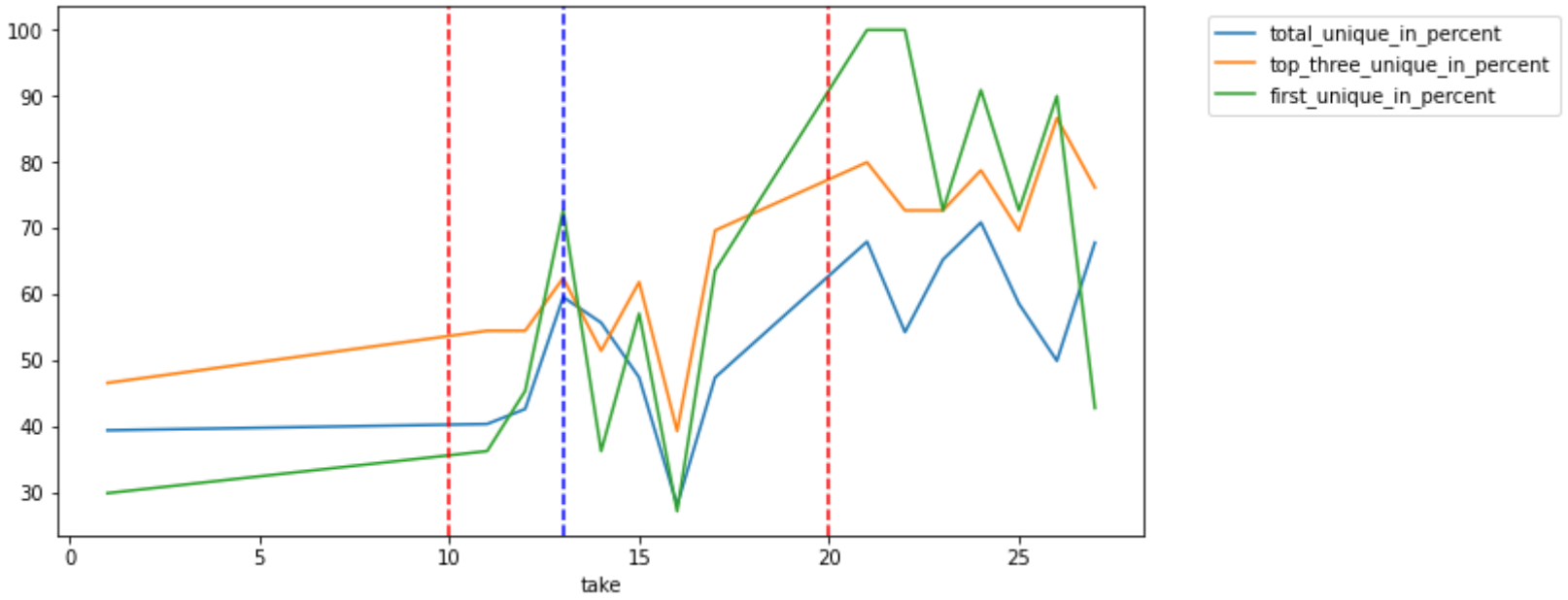
Out[21]:

take	obs_total	obs_top_three	obs_first	total_unique	top_three_unique	first_unique	total_unique_in_percent	top_three_unique_in_percent	first_unique_
1	200	30	10	79	14	3	39.500000	46.666667	
11	220	33	11	89	18	4	40.454545	54.545455	
12	220	33	11	94	18	5	42.727273	54.545455	
13	218	32	11	130	20	8	59.633028	62.500000	
14	217	33	11	121	17	4	55.760369	51.515152	
15	139	21	7	66	13	4	47.482014	61.904762	
16	220	33	11	62	13	3	28.181818	39.393939	
17	219	33	11	104	23	7	47.488584	69.696970	
21	100	15	5	68	12	5	68.000000	80.000000	
22	219	33	11	119	24	11	54.337900	72.727273	
23	219	33	11	143	24	8	65.296804	72.727273	
24	220	33	11	156	26	10	70.909091	78.787879	
25	220	33	11	129	23	8	58.636364	69.696970	
26	200	30	10	100	26	9	50.000000	86.666667	
27	140	21	7	95	16	3	67.857143	76.190476	

In [22]:
```
ax=unique_across_takes[['total_unique_in_percent',
       'top_three_unique_in_percent', 'first_unique_in_percent']].plot(figsize=[10,5])
ax.axvline(10, color="red", linestyle="--")
ax.axvline(20, color="red", linestyle="--")
```

```
ax.axvline(13, color="blue", linestyle="--")
plt.legend(bbox_to_anchor=(1.05, 1))
```

Out[22]:
```
<matplotlib.legend.Legend at 0x18cf20f2f10>
```



As the plot demostrates, variability of recommendations in the top is bigger on the third stage, when users watched contra-annexation videos. Overall, at the second stage the variability in recommendations is lower that on the third with recommendations being more repetitive in the top except of the video 13.

Does this cluster appear because the authors of watched are recommended again on the list of recommended?

Same Author as Watched

In [23]:
```
selection=df.loc[df['watchedAuthor']==df['recommendedAuthor']]
```

In [24]:
```
len(selection)
```

Out[24]:
```
360
```

In [25]:
```
unique_across_takes=unique_across_takes.join(
```

```
    selection.groupby('take').size().to_frame().rename(columns={0: 'au_obs_total'}).join(
    selection.loc[selection['recommendationOrder']<=3].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'reco
    selection.loc[selection['recommendationOrder']<=1].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'reco
    selection.groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommendedVideoId': 'au_total_size'})).join(
    selection.loc[df['recommendationOrder']<=3].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommended
    selection.loc[df['recommendationOrder']<=1].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommended
```

In [26]:
```
unique_across_takes=unique_across_takes.fillna(0)
```

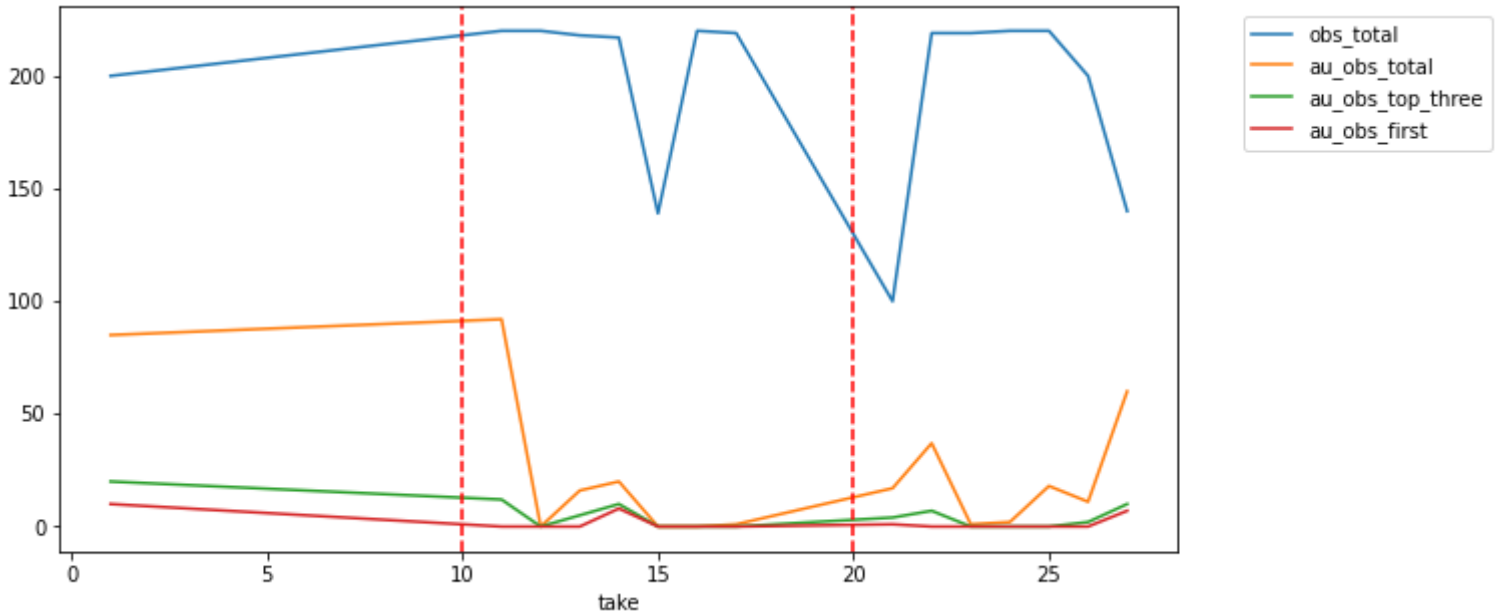
In [27]:
```
unique_across_takes
```

Out[27]:

take	obs_total	obs_top_three	obs_first	total_unique	top_three_unique	first_unique	total_unique_in_percent	top_three_unique_in_percent	first_unique_
1	200	30	10	79	14	3	39.500000	46.666667	
11	220	33	11	89	18	4	40.454545	54.545455	
12	220	33	11	94	18	5	42.727273	54.545455	
13	218	32	11	130	20	8	59.633028	62.500000	
14	217	33	11	121	17	4	55.760369	51.515152	
15	139	21	7	66	13	4	47.482014	61.904762	
16	220	33	11	62	13	3	28.181818	39.393939	
17	219	33	11	104	23	7	47.488584	69.696970	
21	100	15	5	68	12	5	68.000000	80.000000	
22	219	33	11	119	24	11	54.337900	72.727273	
23	219	33	11	143	24	8	65.296804	72.727273	
24	220	33	11	156	26	10	70.909091	78.787879	
25	220	33	11	129	23	8	58.636364	69.696970	
26	200	30	10	100	26	9	50.000000	86.666667	
27	140	21	7	95	16	3	67.857143	76.190476	

In [28]:
```
ax=unique_across_takes[['obs_total',
       'au_obs_total', 'au_obs_top_three', 'au_obs_first']].plot(figsize=[10,5])
ax.axvline(10, color="red", linestyle="--")
ax.axvline(20, color="red", linestyle="--")
plt.legend(bbox_to_anchor=(1.05, 1))
```

Out[28]:
```
<matplotlib.legend.Legend at 0x18cf2132490>
```



A very small amount of video by same author as watched is in recommended. Most likely these videos will appear below top three of recomended. As the formation of recommendations happens simultaneously to the start of video watch, the factor of channel watched may impact not the current, but the next list of recommended. Therefore the idea would be to show if there is such a dependence in the list of recommendations on precedent video viewing.

Same author as precedent

In [29]:
```
watched=pd.DataFrame(df.watchedAuthor.unique())[1:].reset_index().join(pd.DataFrame(df.watchedAuthor.unique()), lsuffix='index')
del watched['index']
```

In [30]:
```
df=df.set_index('watchedAuthor').join(watched, lsuffix='watchedAuthor')
```

In [31]:
```
df=df.reset_index()
```

In [32]:
```
df
```

Out[32]:

	watchedAuthor	take	recommendedVideoId	savingTime	id	watcher	recommendedVerified	recommendationOrder	w
0	Anna News	17	-8S3yM-qyf8	2020-02-14T15:26:33.200Z	de735cbbe3ff8679fb41	User 1	True	11	i2pN
1	Anna News	17	2YMkQW3hAn0	2020-02-14T15:26:33.200Z	de735cbbe3ff8679fb41	User 1	True	15	i2pN
2	Anna News	17	4x_1Ek4nSYc	2020-02-14T15:26:33.200Z	de735cbbe3ff8679fb41	User 1	True	5	i2pN
3	Anna News	17	8dNyTaJRfWQ	2020-02-14T15:26:33.200Z	de735cbbe3ff8679fb41	User 1	True	20	i2pN
4	Anna News	17	AdYqFZGqmHw	2020-02-14T15:26:33.200Z	de735cbbe3ff8679fb41	User 1	True	17	i2pN
...	...	...	...	...	...	...	...	...	
2966	Реальная журналистика	23	h0yY7cGvHP8	2020-02-16T11:09:33.142Z	3c63a55782d1da1cf902	User 2	True	18	TLZ
2967	Реальная журналистика	23	jzWCAgryXFQ	2020-02-16T11:09:33.142Z	3c63a55782d1da1cf902	User 2	True	13	TLZ
2968	Реальная журналистика	23	n6rgozEWL1U	2020-02-16T11:09:33.142Z	3c63a55782d1da1cf902	User 2	True	5	TLZ
2969	Реальная журналистика	23	oZ5vPV95Q4I	2020-02-16T11:09:33.142Z	3c63a55782d1da1cf902	User 2	True	2	TLZ
2970	Реальная журналистика	23	vgOTFcmqosc	2020-02-16T11:09:33.142Z	3c63a55782d1da1cf902	User 2	True	6	TLZ

2971 rows × 27 columns

In [33]:
```
selection=df.loc[df['recommendedAuthor']==df['precendentAuthor']]
```

In [34]:
```
len(selection)
```

Out[34]:
```
103
```

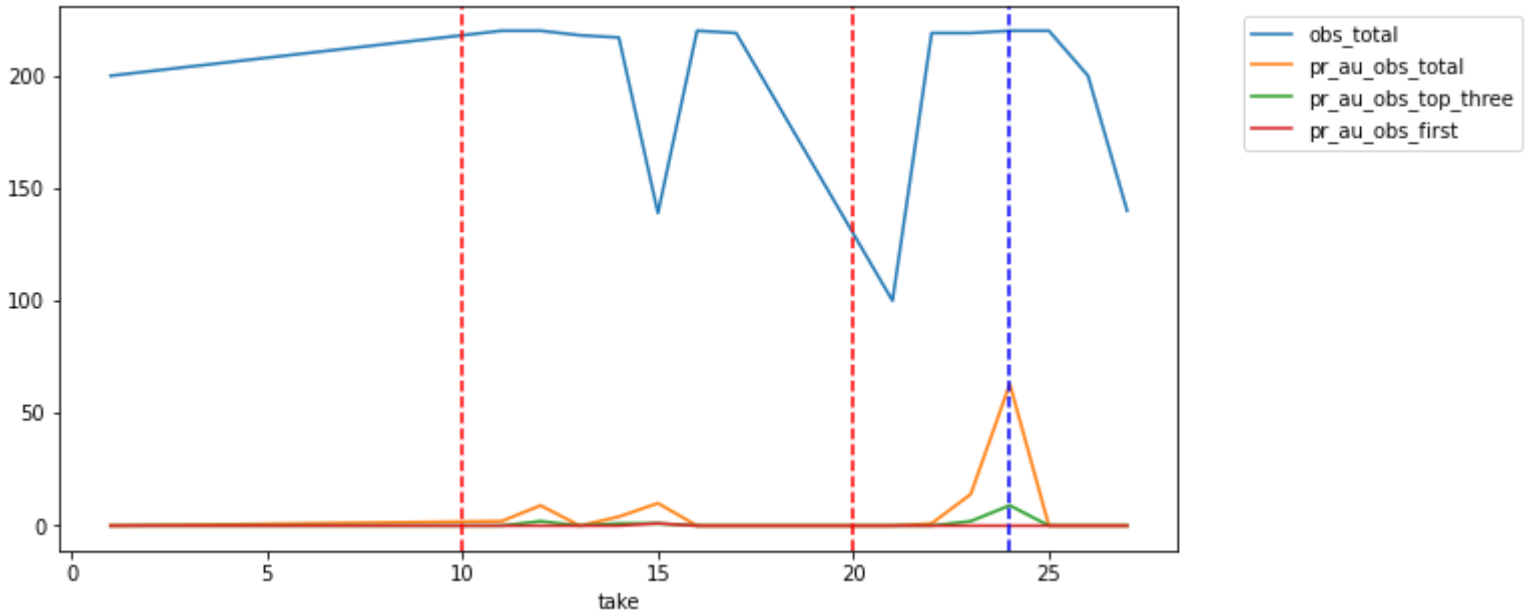
There even less matches between the author of recommended and the author of precedent video than between the author of recommended and the author of watched. However, there still might be a cooccurence?

In [35]:
```
unique_across_takes=unique_across_takes.join(
    selection.groupby('take').size().to_frame().rename(columns={0: 'pr_au_obs_total'}).join(
    selection.loc[selection['recommendationOrder']<=3].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'rec
    selection.loc[selection['recommendationOrder']<=1].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'rec
    selection.groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommendedVideoId': 'pr_au_total_size'})).j
    selection.loc[df['recommendationOrder']<=3].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommende
    selection.loc[df['recommendationOrder']<=1].groupby('take').recommendedVideoId.size().to_frame().rename(columns={'recommende
```

In [36]:
```
unique_across_takes=unique_across_takes.fillna(0)
```

In [37]:
```
ax=unique_across_takes[['obs_total',
       'pr_au_obs_total', 'pr_au_obs_top_three', 'pr_au_obs_first']].plot(figsize=[10,5])
ax.axvline(10, color="red", linestyle="--")
ax.axvline(24, color="blue", linestyle="--")
ax.axvline(20, color="red", linestyle="--")
plt.legend(bbox_to_anchor=(1.05, 1))
```

Out[37]:
```
<matplotlib.legend.Legend at 0x18cf53dbd90>
```



According to this, it is most unlikely that the author of precedent video will appear on the top positions in the next list. Such a tendency appears at 24, however it is the single example in our data. As the users in our collections did not view the video of the same channel more that once, this can not be verified but remains possible in case of different research design.

The next hypothesis is that the repetitive recommendations have to do with the caracteristics of the content. Therefore, I examine annotated categories such as relevance and evaluation.

Content analysis of recommended

Overview over annotation categories in dataset (for details on annotation see Codebook_annotation_dataset_1.pdf)

In [38]:
```
df['relevance'].value_counts()
```

Out[38]:
```
no     1674
yes    1297
Name: relevance, dtype: int64
```

In [39]:
```
df['evaluation'].value_counts()
```

Out[39]:
```
no            1676
positive       713
negative       459
ambivalent     123
Name: evaluation, dtype: int64
```

In [40]:
```
df['engagement'].value_counts()
```

Out[40]:
```
prorussian       786
antiputin        741
no               616
antirussian      518
proputin         175
proukrainian      85
antiukrainian     50
Name: engagement, dtype: int64
```

In [41]:
```
df['cathegory'].value_counts()
```

Out[41]:
```
russian tv                 948
professional blogging      821
amateur blogging           618
ukrainian tv               323
video from radio            94
other                       89
amateur smartphone video    37
Name: cathegory, dtype: int64
```

In [42]:
```
df['topic'].value_counts()
```

Out[42]:
```
politics         1828
economics         340
history           288
war               122
army              102
entertainment      87
medicine           39
news               32
sports             17
music               7
cinema              4
tourism             1
technics            1
Name: topic, dtype: int64
```

In [43]:
```
df['actual_topic'].value_counts()
```

Out[43]:
```
no                                2532
годовщина аннексии Крыма           258
коронавирус                        113
поправки к конституции              53
годовщина смерти Бориса Немцова      7
приговор по делу сети                3
120-летие крейсера "Аврора"          2
переговоры ОПЕГ                      2
взрыв в Магнитогорске                1
Name: actual_topic, dtype: int64
```

In [44]:
```
df['language'].value_counts()
```

Out[44]:
```
russian                 2918
ukrainian                 31
russian and ukrainian     13
english                    8
spanish                    1
Name: language, dtype: int64
```

Examination of relevance and evaluation in dataset

Hypothesis: relevance in recommendations would be equal in parts of datasets, because all users

watched videos to the topic

In [45]:
```
relevance=df.groupby('stage').relevance.value_counts().to_frame().rename(columns={'relevance': 'rel_count'}).unstack(level=0).T.
del relevance['level_0']
```

In [46]:
```
relevance['p_no']=relevance['no']/(relevance['no']+relevance['yes'])*100
relevance['p_yes']=relevance['yes']/(relevance['no']+relevance['yes'])*100
```

In [47]:
```
relevance
```

Out[47]:

relevance	stage	no	yes	p_no	p_yes
0	1	123	77	61.500000	38.500000
1	2	764	689	52.580867	47.419133
2	3	787	531	59.711684	40.288316

There are 7% more relevant videos on the second stage than in third. Does the situation differ for the top three and first of recommended?

In [48]:
```
relevance=df.loc[df['recommendationOrder']<=3].groupby('stage').relevance.value_counts().to_frame().rename(columns={'relevance':
del relevance['level_0']
```

In [49]:
```
relevance['p_no']=relevance['no']/(relevance['no']+relevance['yes'])*100
relevance['p_yes']=relevance['yes']/(relevance['no']+relevance['yes'])*100
```

In [50]:
```
relevance
```

Out[50]:

relevance	stage	no	yes	p_no	p_yes
0	1	12	18	40.000000	60.000000
1	2	110	108	50.458716	49.541284
2	3	102	96	51.515152	48.484848

There are more relevant recommendations in the top three across experiment stages. Will the tendency remain the same for the first recommended?

In [51]:
```
relevance=df.loc[df['recommendationOrder']==3].groupby('stage').relevance.value_counts().to_frame().rename(columns={'relevance':
del relevance['level_0']
relevance['p_no']=relevance['no']/(relevance['no']+relevance['yes'])*100
relevance['p_yes']=relevance['yes']/(relevance['no']+relevance['yes'])*100
```

In [52]:
```
relevance
```

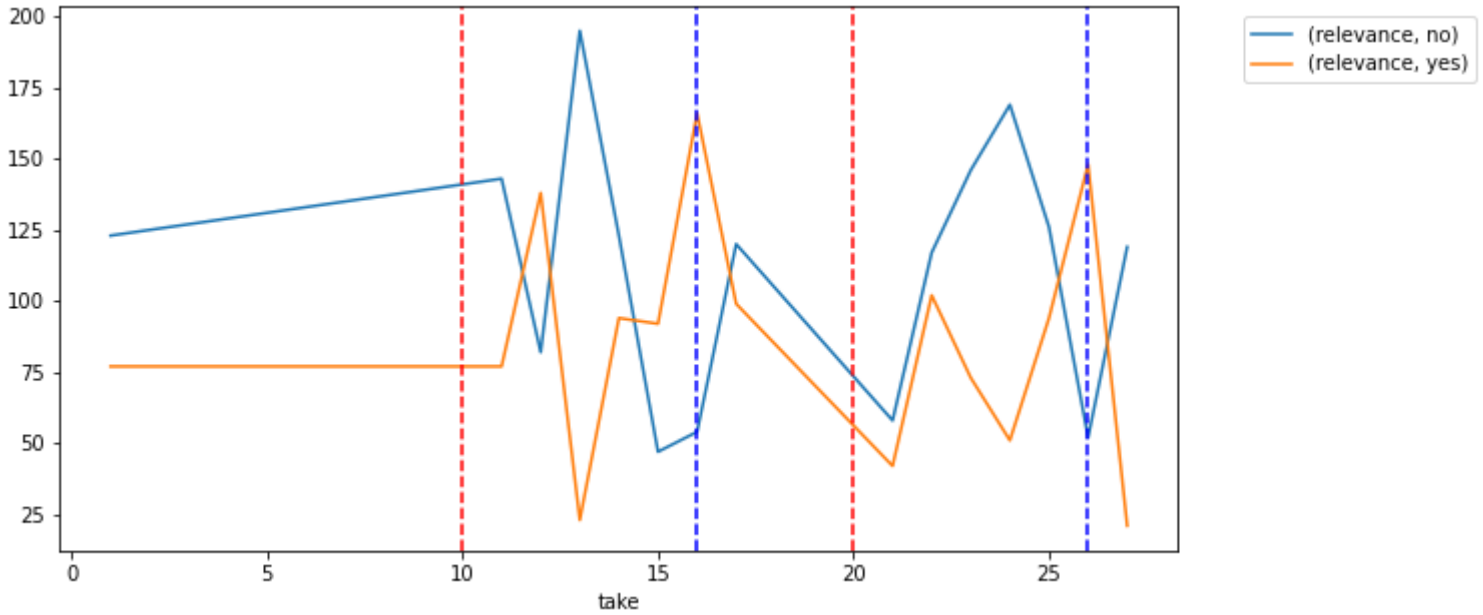
Out[52]:

relevance	stage	no	yes	p_no	p_yes
0	1	6	4	60.000000	40.000000
1	2	45	27	62.500000	37.500000
2	3	39	27	59.090909	40.909091

Comparing the first recommended, relevance is on the same level. Across the different takes, it behaves chaotically.

In [53]:
```
ax=df.groupby('take').relevance.value_counts().to_frame().unstack().plot(figsize=[10,5])
ax.axvline(10, color="red", linestyle="--")
ax.axvline(26, color="blue", linestyle="--")
ax.axvline(16, color="blue", linestyle="--")
ax.axvline(20, color="red", linestyle="--")
plt.legend(bbox_to_anchor=(1.05, 1))
```

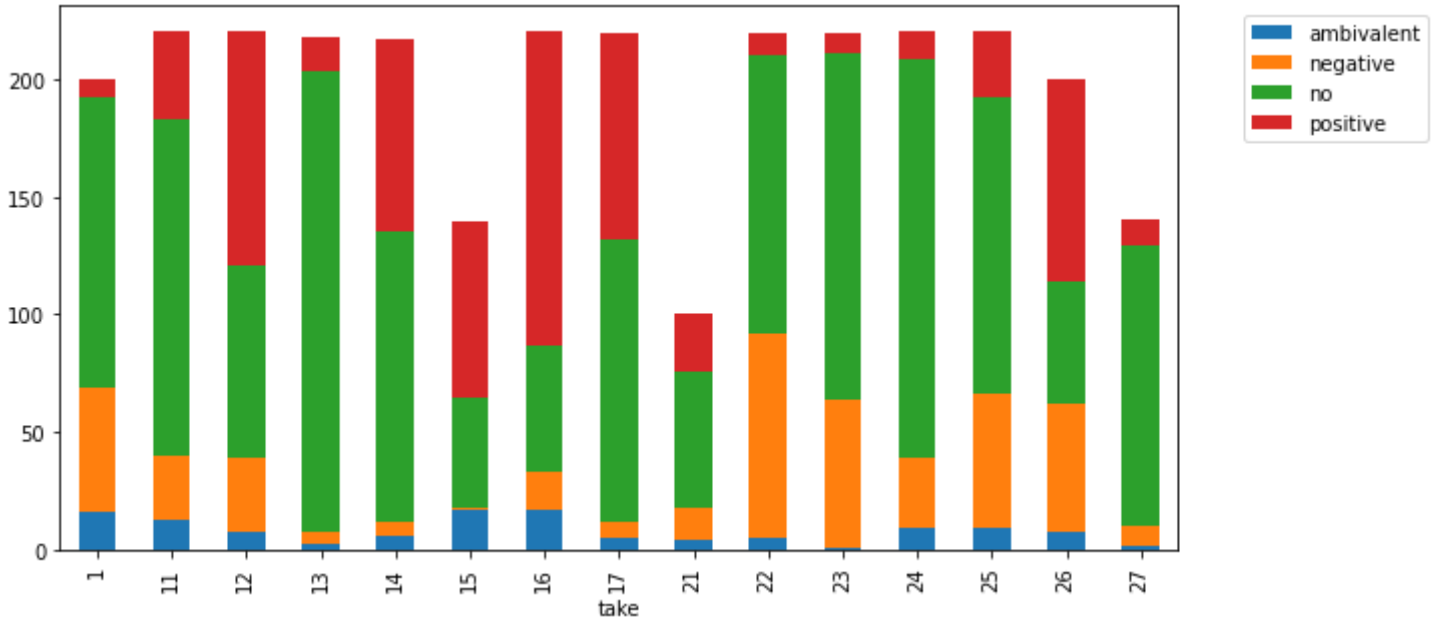
Out[53]:
```
<matplotlib.legend.Legend at 0x18cf54ac310>
```



Hypothesis: positive evaluation dominates after watching pro-annexation videos, while negative evaluation dominates after watching contra-annexation videos.

In [54]:
```
evaluation=df.groupby(['take', 'evaluation']).size().unstack().plot(kind='bar', stacked=True, figsize=[10,5])
plt.legend(bbox_to_anchor=(1.05, 1))
```

Out[54]:
```
<matplotlib.legend.Legend at 0x18cf54dc8b0>
```

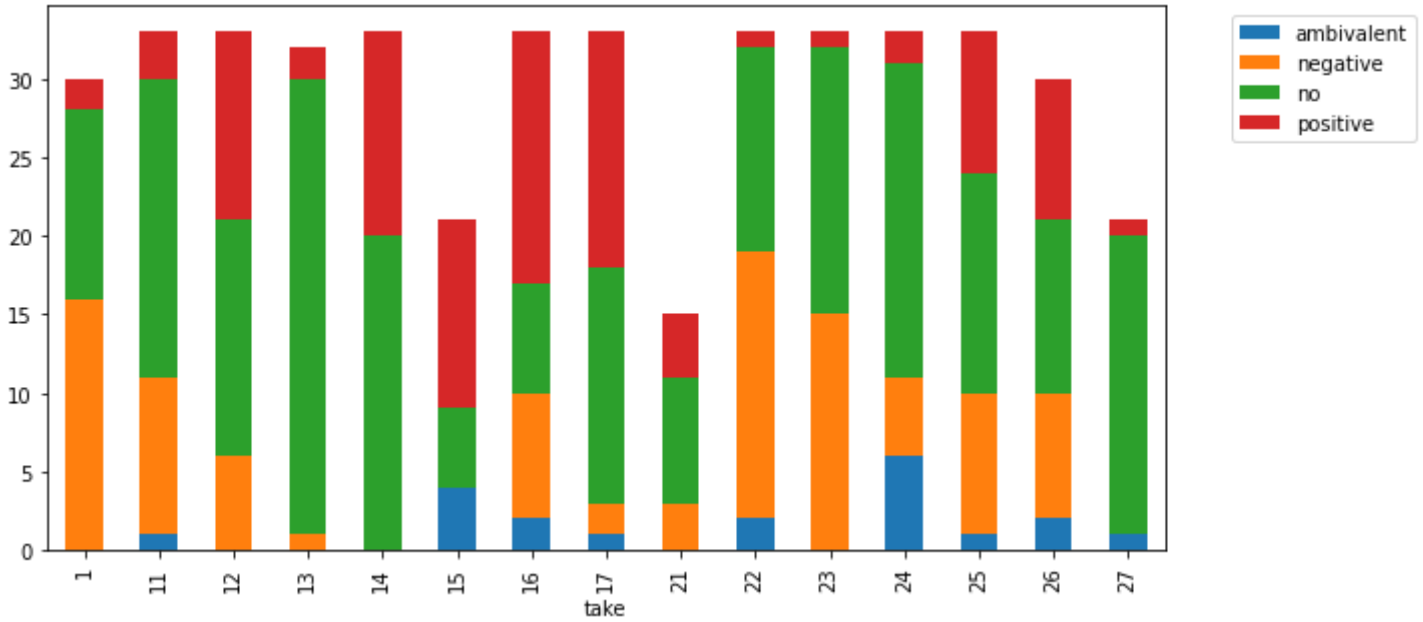


In the second stage (11-17) the positive evaluation clearly dominates in relevant recommendations, the ambivalent view is also more frequent that in third stage (21-27). Negative view on annexation dominates on the third stage, but positive attitude towards the annexation is more frequent than negative attitude after watching pro-annexation videos.

These tendencies grow bigger if first three and the first recommended are evaluated, see below.

In [55]:
```
evaluation=df.loc[df['recommendationOrder']<=3].groupby(['take', 'evaluation']).size().unstack().plot(kind='bar', stacked=True, 
plt.legend(bbox_to_anchor=(1.05, 1))
```

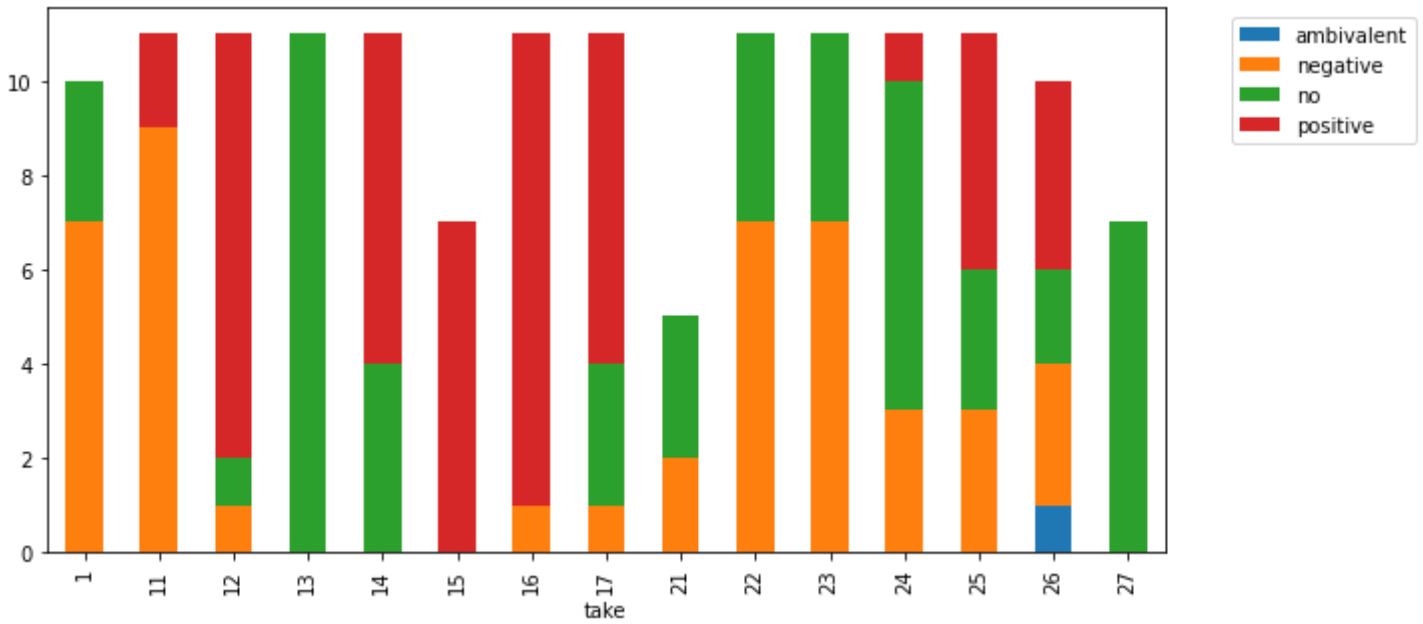
Out[55]:
```
<matplotlib.legend.Legend at 0x18cf5448100>
```



In 25-26, after watching videos with negative attitude towards the annexation, in 4 of 10 cases a video with positive evaluation of the annexation is recommended in the first place (see below). After watching video with positive attitude, a video with opposite view is only recommended once of 10 (see 12, 16 and 17). The ambivalent view is only presented in one recommended video (26).

In [56]:
```
evaluation=df.loc[df['recommendationOrder']==1].groupby(['take', 'evaluation']).size().unstack().plot(kind='bar', stacked=True,
plt.legend(bbox_to_anchor=(1.05, 1))
```

Out[56]:
```
<matplotlib.legend.Legend at 0x18cf56d2e20>
```



Hypothesis: While the relevance remains stable on all stages and evaluations behaves predictable after watching pro- and contra-annexation videos, among the most intrusive recommendations (recommended more that 5 times) will be the same relation, d.h. more repetitive positive after watching pro-annexation videos and the same for contra-annexation videos.

Are highly repetitive recommendations in parts of dataset relevant and how do they evaluate the annexation?

In [57]:
```
repetitive=df.groupby(['stage']).recommendedVideoId.value_counts().to_frame().rename(columns={'recommendedVideoId': 'count'})
```

In [58]:
```
repetitive=repetitive[repetitive['count']>5].reset_index()
```

In [59]:
```
repetitive=repetitive.set_index('recommendedVideoId')
```

In [60]:
```
df.columns
```

Out[60]:
```
Index(['watchedAuthor', 'take', 'recommendedVideoId', 'savingTime', 'id',
       'watcher', 'recommendedVerified', 'recommendationOrder', 'watchedId',
       'watchedPubtime', 'watchedTitle', 'watchedViews', 'watchedChannel',
       'recommendedViews', 'recommendedDuration', 'recommendedPubtime',
       'recommendedAuthor', 'recommendedTitle', 'evaluation', 'relevance',
       'actual_topic', 'engagement', 'topic', 'cathegory', 'language', 'stage',
       'precendentAuthor'],
      dtype='object')
```

In [61]:
```
repetitive=repetitive.join(df[['recommendedVideoId', 'evaluation', 'relevance']].set_index('recommendedVideoId'), lsuffix='recomm
```

In [62]:
```
repetitive.groupby('stage').relevance.value_counts().to_frame()
```

Out[62]:

		relevance
stage	**relevance**	
1	**yes**	53
	no	44
2	**yes**	709
	no	321
3	**yes**	625
	no	259

Although relevance among most frequent videos (recommended more that 5 times across the stage) shows only small difference between second and third take, videos with positive attitude prevail even on the third stage, after the viewings of the contra-annexation videos.

In [63]:
```
repetitive.groupby('stage').evaluation.value_counts().to_frame()
```

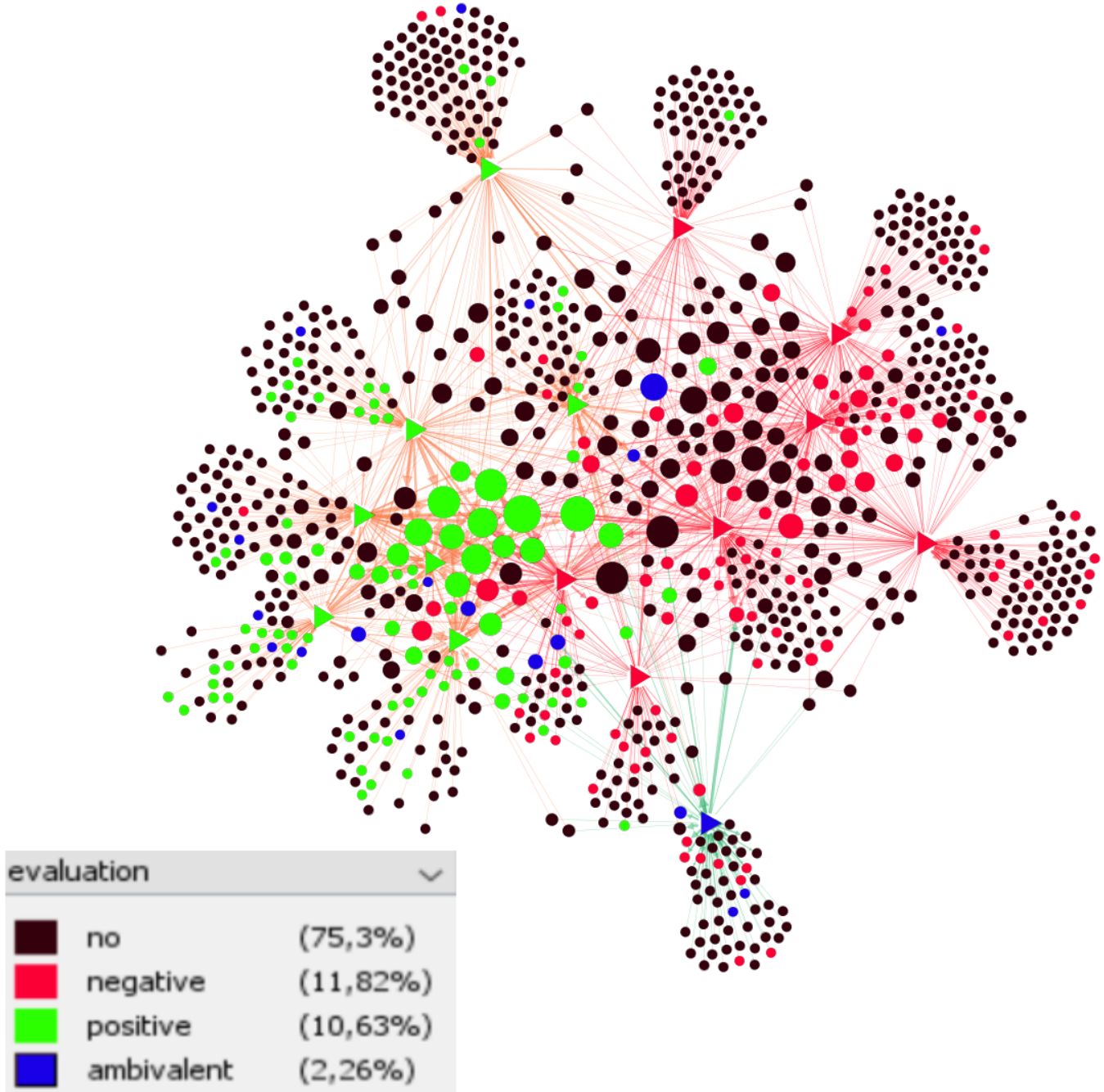
Out[63]:

		evaluation
stage	**evaluation**	
1	**negative**	45
	no	44
	ambivalent	8
2	**positive**	561
	no	321
	negative	76
	ambivalent	72
3	**positive**	393
	no	259
	negative	194
	ambivalent	38

There are more repetitions among pro-annexations recommended on the second and third stage. Videos with negative attitude don't appear as frequent as videos with positive attitude. To give one more visualization of the dataset I show below the picture of the graph representing the relations between the watched (Source) to recommended (Target), created in Gephi (layout ForceAtlas 2, noverlap, nodes ranking after in-degree between 10 and 50, watched videos as triangles with size 30, color after evaluation (see legend)).

In [64]:
```
Image(filename='evaluation.png', width=700, height=700)
```

Out[64]:


evaluation
no (75,3%)
negative (11,82%)
positive (10,63%)
ambivalent (2,26%)

On the graph, videos with positive evaluatin of the annexation build a cluster on the left side (green), but recommended with negative attitude don't build a clear cluster of frequently recommended videos with the same attitude.

To get more insight into the situation, I will concentrate on the political engagement in videos and the category of video as annotated in dataset.

The factor of political engagement and media type

In [65]:
```
df.groupby(['stage', 'engagement']).size().to_frame().unstack()
```

Out[65]:

	0						
engagement	**antiputin**	**antirussian**	**antiukrainian**	**no**	**proputin**	**prorussian**	**proukrainian**
stage							
1	26.0	65.0	NaN	82.0	3.0	20.0	4.0
2	211.0	98.0	38.0	327.0	136.0	613.0	30.0
3	504.0	355.0	12.0	207.0	36.0	153.0	51.0

Interestingly, on the second stage the pro-russian engagement is dominant, while on the third the critique on putinist regime prevails. Pro-russian recommended on the second stage are mostly among revant. That means that the annexation is mostly seen as a great event for the Russian people that an achievement of Putin (407 to 74 in the table below). On the contrary, the anti-putin videos on the third stage are mostly not relevant for the topic.

In [66]:
```
df.groupby(['stage', 'relevance', 'engagement']).size().to_frame().unstack().fillna(0).astype(int)
```

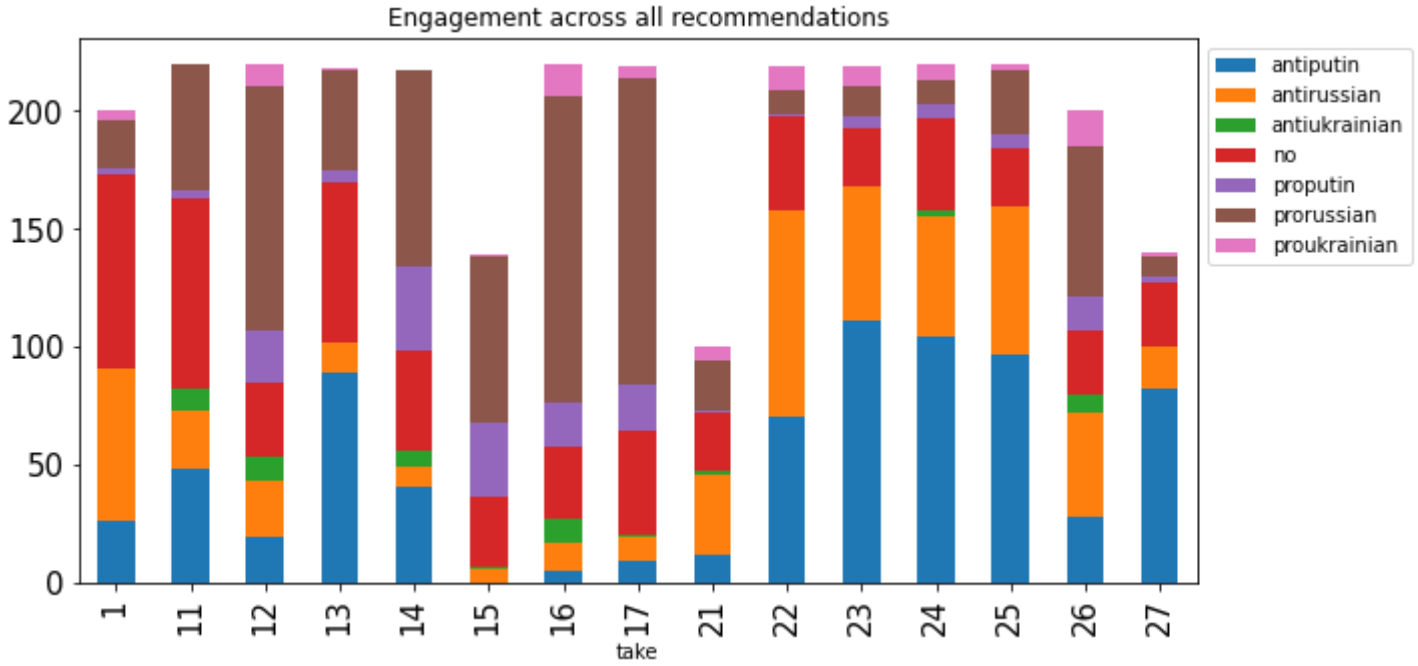
Out[66]:

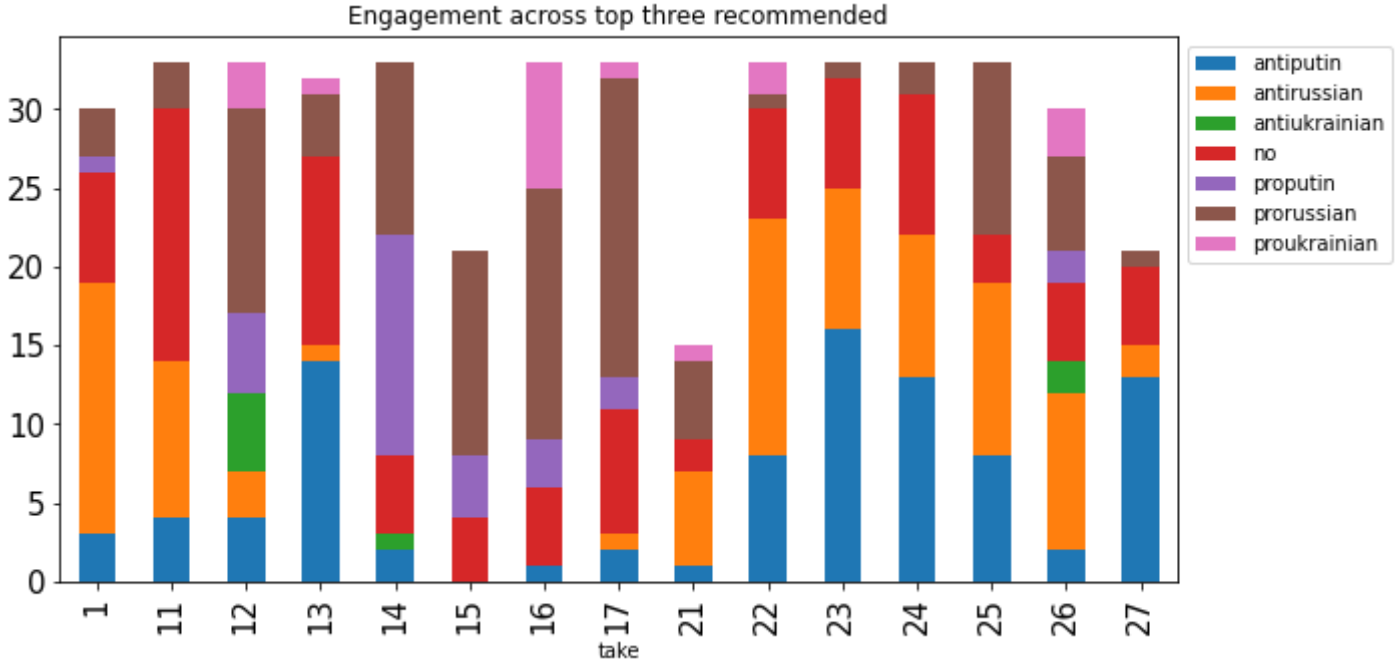
		0						
	engagement	antiputin	antirussian	antiukrainian	no	proputin	prorussian	proukrainian
stage	relevance							
1	no	23	6	0	76	1	13	4
	yes	3	59	0	6	2	7	0
2	no	202	29	8	255	62	206	2
	yes	9	69	30	72	74	407	28
3	no	415	139	2	163	9	34	25
	yes	89	216	10	44	27	119	26

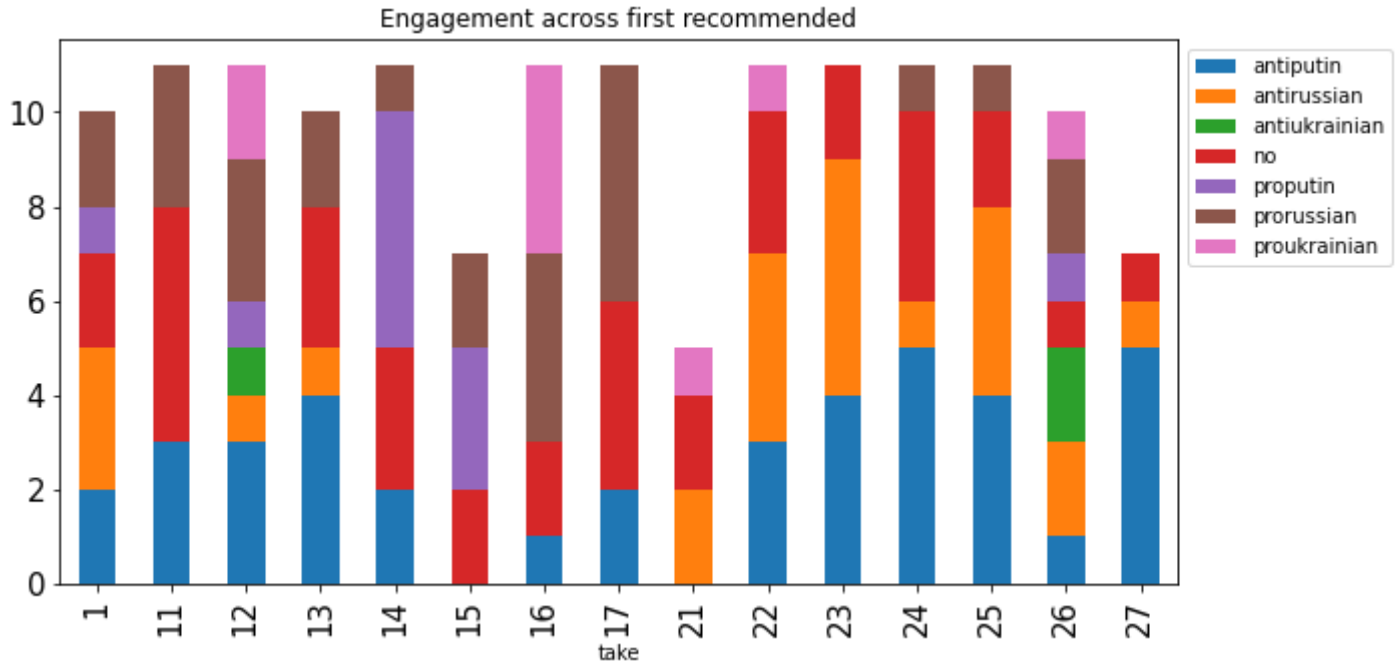
The dominance of the opposition prorussian/antiputin among recommended in the whole dataset, whithin first three and first recommended is shown on the plots below (brown vs. blue segments).

In [67]:
```
ax1=df.groupby(['take', 'engagement']).size().unstack().plot(kind='bar', stacked=True, figsize=(10,5),  fontsize=15)
plt.legend(bbox_to_anchor=(1.0, 1.0))
ax1.set_title('Engagement across all recommendations')
ax2=df.loc[df['recommendationOrder']<=3].groupby(['take', 'engagement']).size().unstack().plot(kind='bar', stacked=True, figsize=
plt.legend(bbox_to_anchor=(1.0, 1.0))
ax2.set_title('Engagement across top three recommended')
ax3=df.loc[df['recommendationOrder']==3].groupby(['take', 'engagement']).size().unstack().plot(kind='bar', stacked=True, figsize=
plt.legend(bbox_to_anchor=(1.0, 1.0))
ax3.set_title('Engagement across first recommended')
```

Out[67]:
```
Text(0.5, 1.0, 'Engagement across first recommended')
```


Engagement across all recommendations
antiputin
antirussian
antiukrainian
no
proputin
prorussian
proukrainian
0
50
100
150
200
1
11
12
13
14
15
16
17
21
22
23
24
25
26
27
take

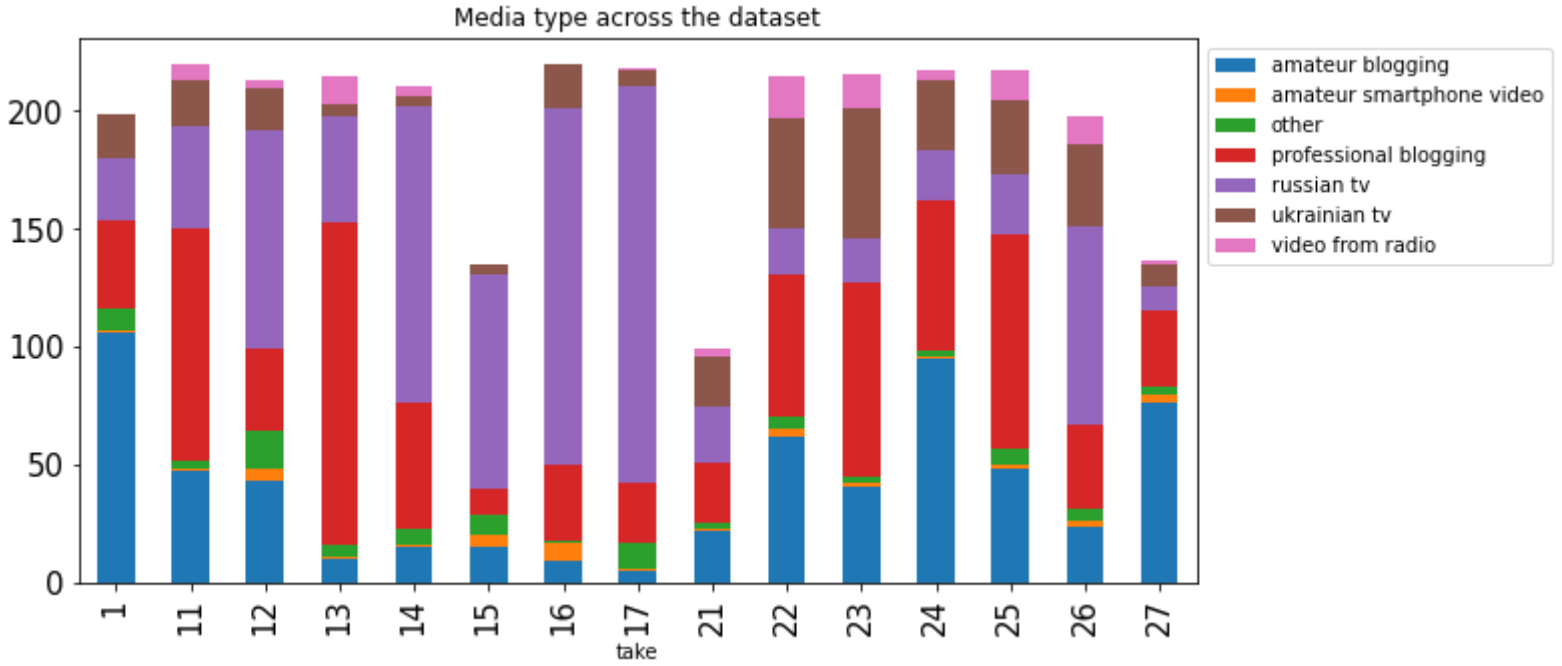

Engagement across top three recommended
antiputin
antirussian
antiukrainian
no
proputin
prorussian
proukrainian
0
5
10
15
20
25
30
1
11
12
13
14
15
16
17
21
22
23
24
25
26
27
take

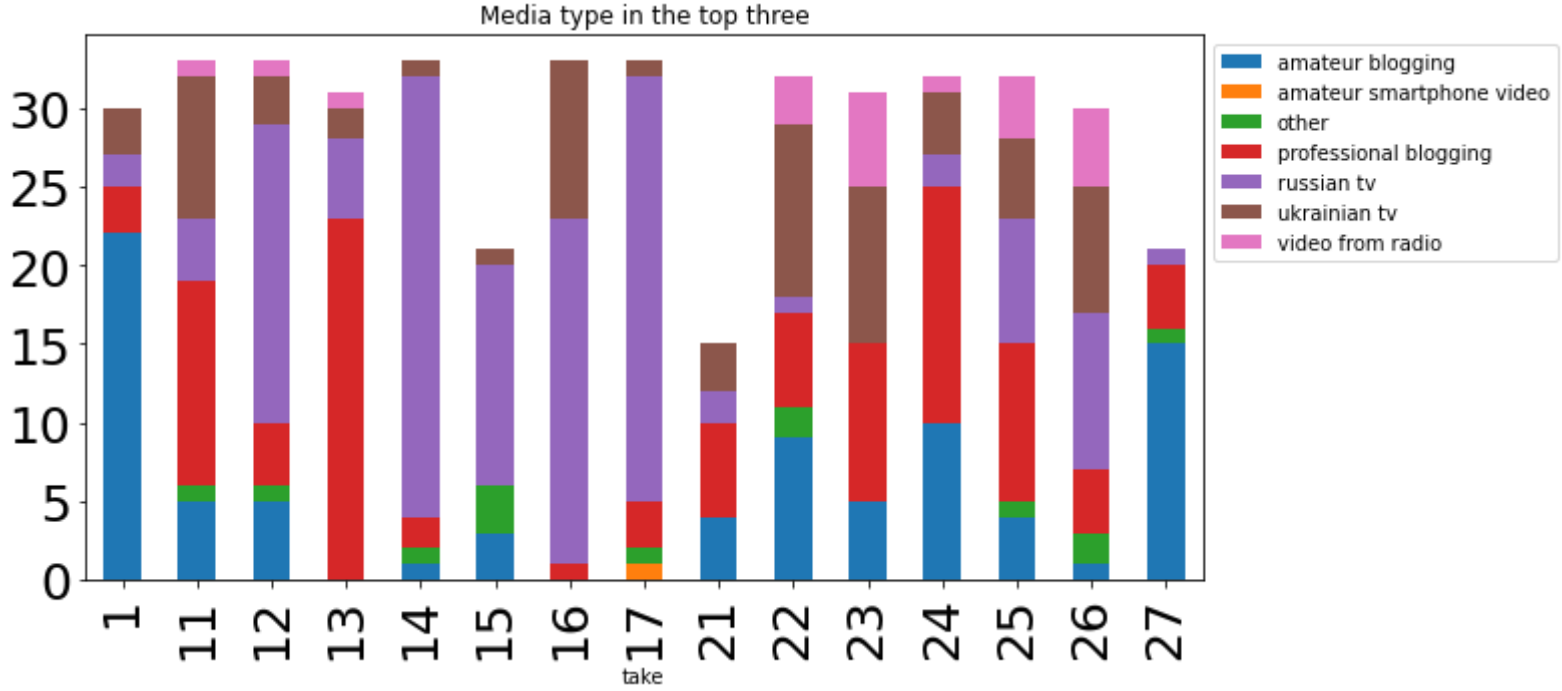


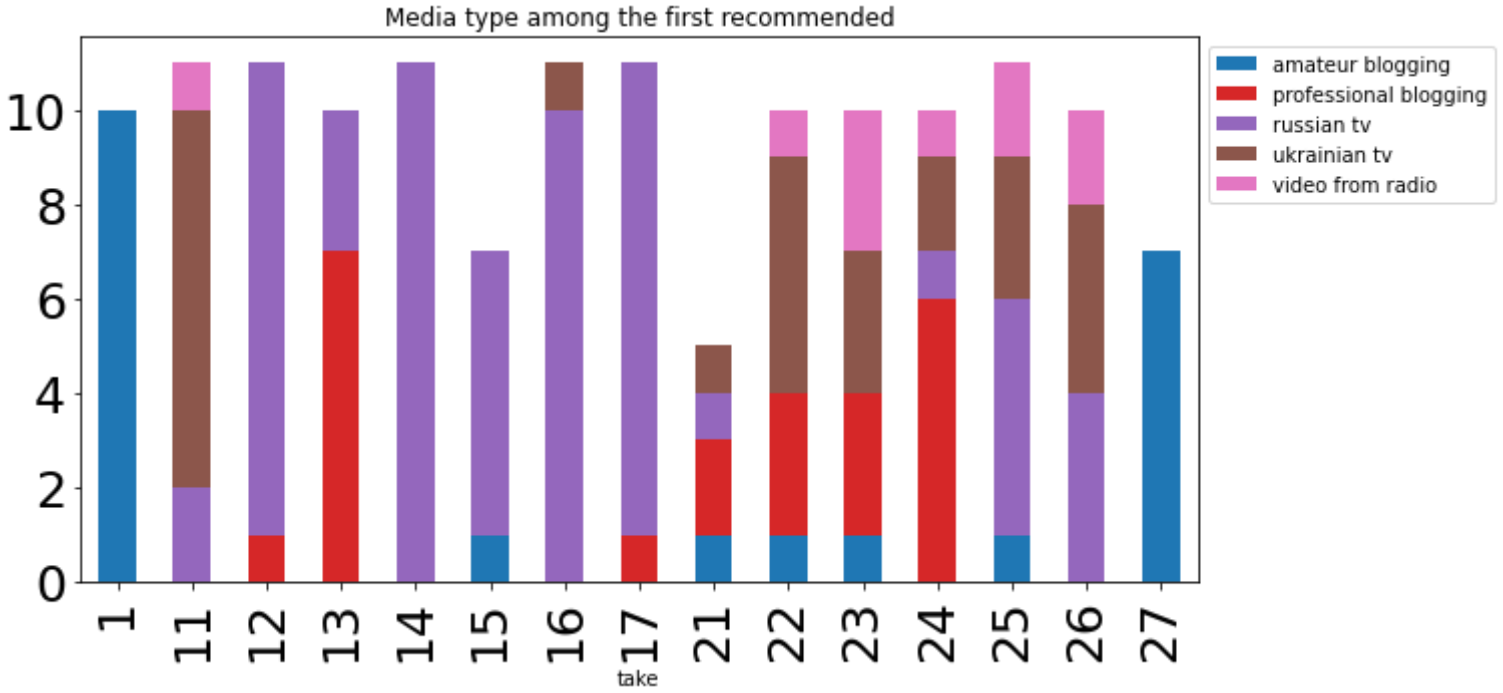
Another interesting feature to be observed is the distribution of media types across the dataset. As shown on the graphs below, the recommendations of the first dataset are dominated by Russian TV productions while on the third stage professional and amateur blogs are most frequent type of media. The last focus in general on critique towards putinism in different aspects, and this is why the videos recommended on the third stage are frequently irrelevant towards the estimation of the Crimea's annexation.

In [68]:
```
ax1=df.groupby(['take', 'cathegory']).size().unstack().plot(kind='bar', stacked=True, figsize=(10,5), fontsize=15, title='Media t
plt.legend(bbox_to_anchor=(1.0, 1.0))

ax2=df.loc[df['recommendationOrder']<=3].groupby(['take', 'cathegory']).size().unstack().plot(kind='bar', stacked=True, figsize=(
plt.legend(bbox_to_anchor=(1.0, 1.0))

colors=['tab:blue', 'tab:red', 'tab:purple', 'tab:brown', 'tab:pink']
ax3=df.loc[df['recommendationOrder']<=1].groupby(['take', 'cathegory']).size().unstack().plot(kind='bar', color=colors, stacked=T
plt.legend(bbox_to_anchor=(1.0, 1.0))
```

Out[68]:
```
<matplotlib.legend.Legend at 0x18cf5e2cf10>
```


Media type across the dataset
amateur blogging
amateur smartphone video
other
professional blogging
russian tv
ukrainian tv
video from radio
0
50
100
150
200
1
11
12
13
14
15
16
17
21
22
23
24
25
26
27
take

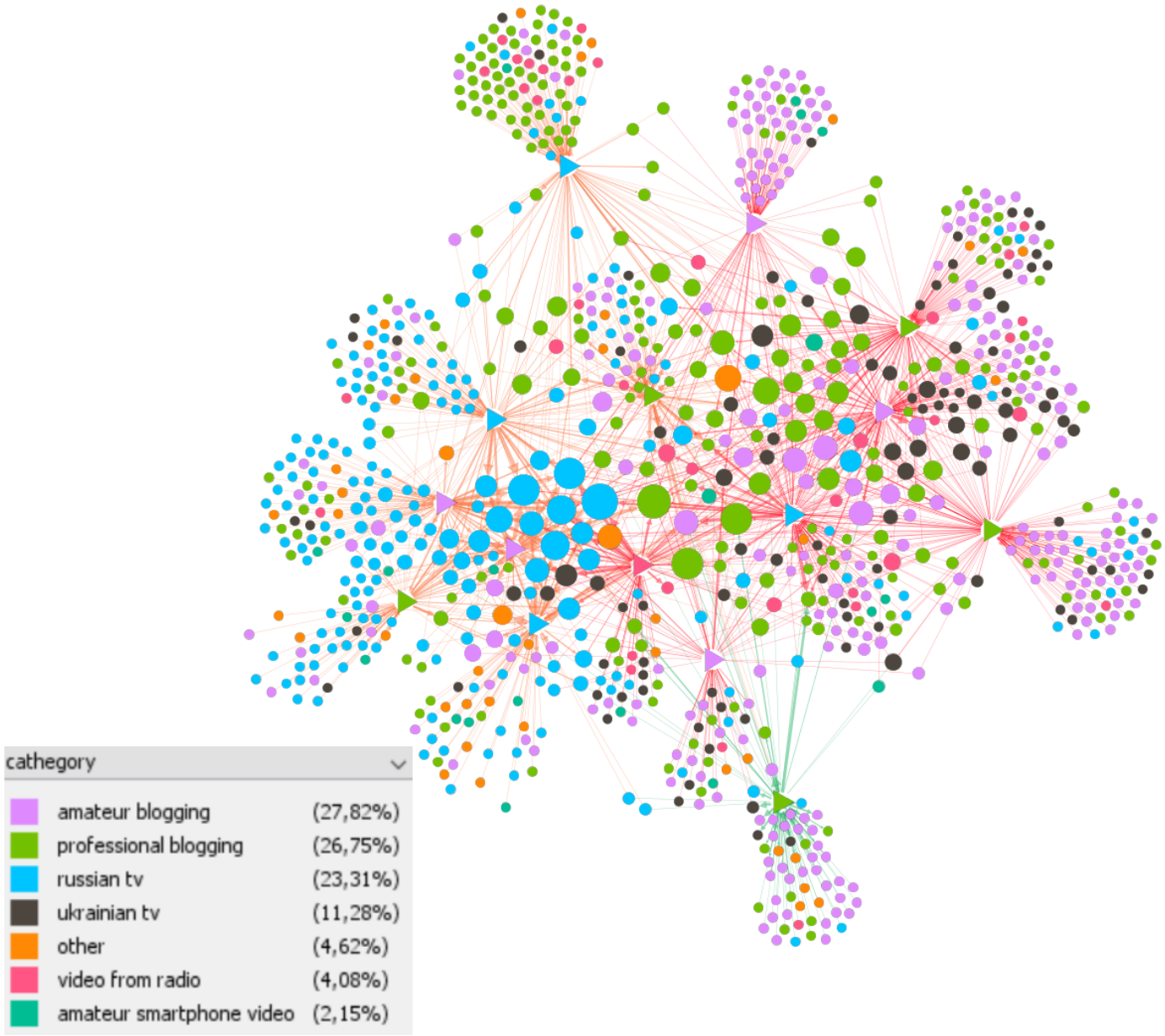

Media type in the top three
amateur blogging
amateur smartphone video
other
professional blogging
russian tv
ukrainian tv
video from radio
0
5
10
15
20
25
30
1
11
12
13
14
15
16
17
21
22
23
24
25
26
27
take



Among the first recommended, there are even more Russian TV productions than at lower positions in the ranking. To give one more visualization of the dataset I show below the picture of the graph representing the relations between the watched (Source) to recommended (Target), created in Gephi (layout ForceAtlas 2, noverlap, nodes ranking after in-degree between 10 and 50, watched videos as triangles with size 30, color after category (see legend)).

In [69]:
```
Image(filename='category.png', width=700, height=700)
```

Out[69]:



The dominance of the blue on the left indicates the influence of Russian TV channels among the recommended. The frequent recommended videos of Russia TV channels (blue cluster on the left) are also relevant and accompanied by the recommended in the same category. On the right, another cluster can be obseved, which was not visible before: green and rose recommended come from professional and amateur blogs.

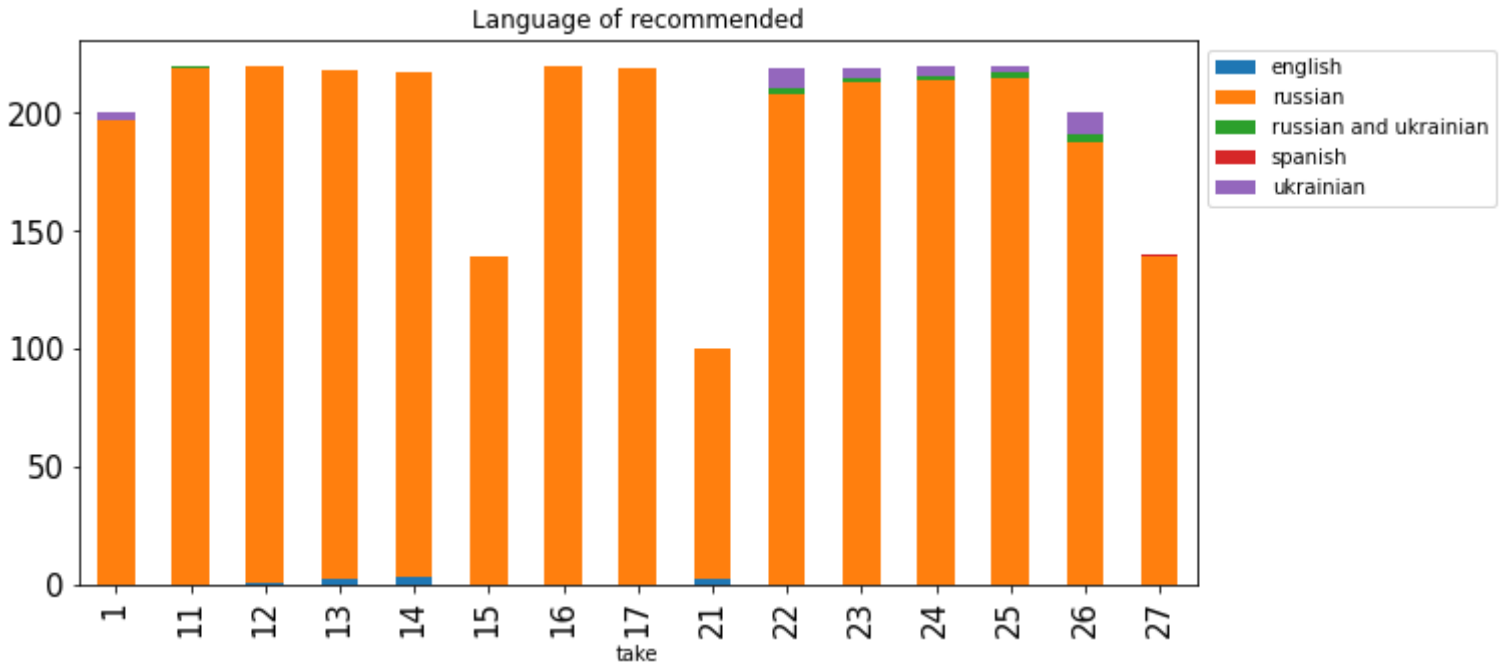
Such media controversy between Russian TV and YouTube blogs is comprehensible, since the political opposition was repressed from official TV channels and finds its place on YouTube. For the case of Crimea's annexation, the most interesting detail is that the right cluster is segmented more, that means that the videos in this cluster is recommended less frequently, with a bigger proportion of unique as demonstated above. The recommended in this cluster are frequently not directly relevant for the topic of the annexation, but generally negative towards the Putinist regime.

Factor of language

Let's take a quick look at another annotation parameter: the language of annotated videos. The plot below demonstrates the dominance of Russian across the dataset. Watching of first video in Russian was enough for YouTube recommended to render the recommendations almost exlusively in the same language.

In [70]:
```
language=df.groupby(['take', 'language']).size().unstack().plot(kind='bar', stacked=True, figsize=(10,5), fontsize=15, title='Lar
plt.legend(bbox_to_anchor=(1.0, 1.0))
```

Out[70]:
```
<matplotlib.legend.Legend at 0x18cf5427b50>
```



Comparison of first, second and third dataset

The second and the third datasets were collected starting from the most frequent relevant recommendations in the first and second dataset respectively. The goal was to collect more relevant videos and inspect the limitations of the bubble shown at the left side of the graph. 17 videos for the second and 44 videos for the third collection were selected from the top relevant videos from the previous collection. There was no separation in different opinions towards the annexation.

Nevertheless, the constitution of the dataset allows a comparison between dataset regarding their formal features: the frequency of the unique recommended and the impact of duration of collection on collected data. To test this factors on the more unified data parametes, I build two test samples: one test sample by selecting six users from the first dataset and the second by selecting the first 18 starting videos of the third dataset.

In [71]:
```
df_2=pd.read_excel('DATASET_2.xlsx')
```

In [72]:
```
df_2.columns
```

Out[72]:
```
Index(['Unnamed: 0', 'publicKey', 'login', 'id', 'savingTime', 'clientTime',
       'uxLang', 'parameter', 'recommendedId', 'recommendedVideoId',
       'recommendedAuthor', 'recommendedTitle', 'recommendedLength',
       'recommendedLengthText', 'recommendedPubTime', 'recommendedRelativeS',
       'recommendedViews', 'recommendedForYou', 'recommendedVerified',
       'recommendationOrder', 'recommendedKind', 'watchedVideoId',
       'watchedTitle', 'watchedAuthor', 'watchedChannel', 'watchedPubTime',
       'watchedViews', 'watchedLike', 'watchedDislike', 'take', 'watcher'],
      dtype='object')
```

In [73]:
```
df_2.savingTime.min()
```

Out[73]:
```
'2020-12-03T10:23:56.071Z'
```

In [74]:
```
df_2.savingTime.max()
```

Out[74]:
```
'2020-12-08T14:08:07.569Z'
```

In [75]:
```
df_2.groupby(['take', 'watchedVideoId',  'watchedTitle', 'watchedAuthor']).size().to_frame().rename(columns={0: 'observations'})
```

Out[75]:

				observations
take	watchedVideoId	watchedTitle	watchedAuthor	
1	kJ_mWtDyBUo	Крым глазами американца	RT на русском	119
2	IX62FV9TCSU	«Крымская весна». Документальный фильм Сергея Холошевского	НТВ	101
3	mhrPXMQUNBA	Обращение Владимира Путина по итогам референдума в Крыму	RT на русском	120
4	4x_1Ek4nSYc	Крым. Возвращение домой. Специальный репортаж	TV Center	120
5	oZ5vPV95Q4I	Как изменился Крым за 5 лет. Василий Волга	TOPINFORM	120
6	t9AWGnFCi9U	18.03.2014 Тысячи севастопольцев празднуют на площади Нахимова возвращение в состав России	НТС Севастополь	120
7	ymzxXczhc00	〃 Крим. Оточені зрадою 〃	Ukrainian military TV	80
			Військове телебачення України	40
8	VXt0-e0_aYM	Крымский 〃 Беркут 〃 . Документальный фильм - Россия 24	Россия 24	120
9	yGkVDeecVmU	Таблетки от истерии и отпечатки рогов: Путин поразил участников ПМЭФ откровенностью	Россия 24	19
10	g-oWNfImYQo	Крымская правда. Специальный репортаж	TV Center	121
11	wIg6WkwMd9k	Керчь: «Крымская весна». Как это было	kerchnettv	120
12	vgOTFcmqosc	Крым. Линия защиты	TV Center	120
13	aGNbdYyBBXI	ВЕЖЛИВЫЕ ЛЮДИ Редкие кадры	НОВОСТИ РОССИИ И УКРАИНЫ	120
14	YywmuqTC-l4	Спец-операция 〃 Вежливые Люди 〃 - Хроника в Крыму	Танковые Курсы ☆ World Of Tanks ☆ StubofMetal	120
15	8bQ48X7D7wo	За что Крым не любил Украину?	Bird In Flight magazine	120
16	otFAf5_Aiwo	Олександр Лукашенко про Крим, Зеленського і Україну – ЕКСКЛЮЗИВ ДЛЯ УКРАЇНСЬКИХ ЗМІ	Сегодня	120
17	6-gEYGG2ZG0	Крымчане поняли, куда они попали	Новости СВЕРХДЕРЖАВЫ	120
18	JztzWP3mZw4	Россияне устали от Крыма	Новости СВЕРХДЕРЖАВЫ	120

In [76]:
```
df_3=pd.read_excel('DATASET_3.xlsx')
```

In [77]:
```
df_3.savingTime.min()
```

Out[77]:
```
'2020-12-12T10:43:21.889Z'
```

In [78]:
```
df_3.savingTime.max()
```

Out[78]:
```
'2020-12-14T10:45:11.746Z'
```

In [79]:
```
df_3.groupby(['watchedVideoId',  'watchedTitle', 'watchedAuthor']).size().to_frame().rename(columns={0: 'observations'})
```

Out[79]:

watchedVideoId	watchedTitle	watchedAuthor	observations
0h1-iJLxi10	〃 Крым наш! Навсегда! 〃 4 серия	НТВ	120
1jkuNM4G5bE	Документальный фильм 〃 ГЕРОИ СЕВАСТОПОЛЬСКОЙ ВЕСНЫ 〃	РИА «Севастополь»	120
2rzfINElcSE	Оккупация Крыма: факты, о которых не расскажут в новостях - Секретный фронт	Телеканал ICTV	120
35X8XzKvCWo	Спецрепортаж. Операція Крим	ТСН	120
4x_1Ek4nSYc	Крым. Возвращение домой. Специальный репортаж	TV Center	120
6ukYA2kdl1g	Зачем Путину Крым и Донбасс – Больше чем правда	Телеканал ICTV	120
8ArP2vmkduY	Владимир Путин: Реакцию Запада на присоединение Крыма в Москве считают абсолютно неадекватной	RT на русском	120
BJgzVCIqVAg	Западные журналисты убедились, что Россия не вторгалась в Крым	RT на русском	120
Csogq_-pg2c	Мураев: Оккупанты не строят атомные станции, заводы, мосты и дороги. НАШ 09.10	НАШ	120
DYTzs6iCRso	Как обращение Путина смотрели в Севастополе	Dimitro Sevastopol	80
		Sevastopol Life	40
HCRKQ5YmRfU	Крым. Новая эра. Документальный фильм Аркадия Мамонтова - Россия 24	Россия 24	120
IX62FV9TCSU	«Крымская весна». Документальный фильм Сергея Холошевского	НТВ	120
K6WJu5hbL04	КРЫМ / Фильм. Мелодрама	EPIC MEDIA	80
		Epic Media	38
KzQHLXzyuOc	Крым берет под защиту бойцов «Беркута»	RT на русском	120
L7XcwUGw17s	Керченський міст. Спецпроект – РС | Крим.Реалії	Радіо Свобода	120
PNuPLtM7jwU	Беркут. Последний бой. Специальный репортаж	TV Center	117
RikohxTV6y4	Разговор Крымского Татарина и военного из РФ Крым 13 03 14	Ukrainian military TV	80
		Військове телебачення України	40
Tj7zEHRX0I8	Поклонская. Измена Родине, Янукович, Путин, Аксенов, 〃 няш-мяш 〃 , Николай II, Зеленский. 〃 ГОРДОН 〃 2020	Дмитрий Гордон	120

			observations
watchedVideoId	watchedTitle	watchedAuthor	
VTnBfzRcC_4	Крым. Испытание Украиной. Специальный репортаж	TV Center	120
VXt0-e0_aYM	Крымский 〃Беркут〃. Документальный фильм - Россия 24	Россия 24	120
YywmuqTC-l4	Спец-операция 〃Вежливые Люди〃 - Хроника в Крыму	Танковые Курсы ☆ World Of Tanks ☆ StubofMetal	120
ZlUhntshLpw	〃Крым наш! Навсегда!〃 1 серия	НТВ	120
bCznoaVNCk0	📺Вежливые люди Крым! ⚪ Одно лишь🔵слово Верь! 🔴 (2015)🏳	Евгений Argunov Live	120
b_P6DOdWWAI	Як захопили останній корабель українського флоту? | Крим.Реалії	Радіо Свобода	120
glkQB456CC8	«Острый Крым». Фильм Сергея Холошевского	НТВ	120
iDR7gxeJXGI	Українська частина ППО спробувала повернути контроль над аеропортом 〃Бельбек〃	Espreso.TV	120
jdVHKBcdzz4	Мировые СМИ активно комментируют открытие движения по Крымскому мосту.	Новости на Первом Канале	120
jeQr08Oqw_k	〃Вежливые люди〃 глазами очевидцев.	LORD13VV	120
k27ANNqog04	Наш Крым. Документальный фильм	Россия 24	120
kJ_mWtDyBUo	Крым глазами американца	RT на русском	118
mhrPXMQUNBA	Обращение Владимира Путина по итогам референдума в Крыму	RT на русском	120
oHiH9IjaM1w	〃Крым наш! Навсегда!〃 3 серия	НТВ	120
otFAf5_Aiwo	Олександр Лукашенко про Крим, Зеленського і Україну – ЕКСКЛЮЗИВ ДЛЯ УКРАЇНСЬКИХ ЗМІ	Сегодня	120
ov_9tAwBem8	Как поменялся Крым и крымчане с 2014-го. Медведева отвечает | Чей Крым? 5 лет без Украины | Часть 4	Страна.ua	120
rOakGH-nvYs	«Форсаж. Возвращение» - «Крым. Небо родины», реж. Н. Гугуева, 2016	Natalia Gugueva	120
sTpkZVUh6Sw	⚪ВЕЖЛИВЫЕ ЛЮДИ!🔵 КАК ЭТО БЫЛО!🔴 2014	Евгений Argunov Live	120
t9AWGnFCi9U	18.03.2014 Тысячи севастопольцев празднуют на площади Нахимова возвращение в состав России	НТС Севастополь	120
tXGET1-fzmI	Крим, як це було / Crimea. As it was / Крым, как это было	#BABYLON'13	120

	watchedVideoId	watchedTitle	watchedAuthor	observations
	tXHuKIeHUZU	Чем недовольны жители Крыма на пятый год российской оккупации - Гражданская оборона	Телеканал ICTV	120
	uAKuFx9VZsc	09.05.2014 БЕЗ КОММЕНТАРИЕВ: ВЛАДИМИР ПУТИН ПРИБЫЛ В СЕВАСТОПОЛЬ	НТС Севастополь	120
	vgOTFcmqosc	Крым. Линия защиты	TV Center	120
	wIg6WkwMd9k	Керчь: «Крымская весна». Как это было	kerchnettv	120
	x6Gy6C8fzdk	Як Росія захопила Крим? Спецпроект | Крим.Реалії	Радіо Свобода	120
	xSpczh5DqW8	Почему Украина потеряла Крым? Опрос киевлян	Страна.ua	120

In [80]:
```
first_sample=df.loc[df['watcher'].isin(['User 1', 'User 2', 'User 3', 'User 4', 'User 6', 'User 7'])]
```

In [81]:
```
len(first_sample)
```

Out[81]:
```
1697
```

In [82]:
```
second_sample=df_3.loc[df_3['watchedVideoId'].isin(['0h1-iJLxi10', 'rOakGH-nvYs', 'IX62FV9TCSU', '35X8XzKvCWo', 'oHiH9IjaM1w',
 'wIg6WkwMd9k', 'HCRKQ5YmRfU', 'x6Gy6C8fzdk', 'K6WJu5hbL04', 'PNuPLtM7jwU', 'VTnBfzRcC_4', 'RikohxTV6y4', 'BJgzVCIqVAg',
 'L7XcwUGw17s', 'ZlUhntshLpw', 'vgOTFcmqosc', 'uAKuFx9VZsc', 'gIkQB456CC8'])]
```

In [83]:
```
len(second_sample)
```

Out[83]:
```
2155
```

In [84]:
```
data = {'dataset': ['first', 'first_sample', 'second', 'third', 'second_sample'], 'amount of start videos': [15, 15, 18, 44,18],
comparison = pd.DataFrame(data=data)
comparison['percentage unique to total of recommended']=(comparison['unique videos']/comparison['total recommended'])*100
comparison['percentage unique to total of recommended']=comparison['percentage unique to total of recommended'].round(1)
```

In [85]:
```
comparison
```

Out[85]:

	dataset	amount of start videos	hours of data collection	total recommended	unique videos	max of unique recommendations per video	percentage unique to total of recommended
0	first	15	1033	2954	933	60	31.6
1	first_sample	15	980	1697	682	33	40.2
2	second	18	124	2040	413	84	20.2
3	third	44	48	5273	1019	146	19.3
4	second_sample	18	46	2155	550	64	25.5

Comparing the percentage values in the last column, we see a range between 19% and 40%. The first dataset and test sample show the largest values, presumably because of the longer period of data collection. In the cases of the second and third datasets, collected within a shorter period, having six users watching the same list of videos shows the same relation between the numbers of unique recommendations in relation to the total recommended.

The comparison between the first dataset and the test sample demonstrates that the decrease in the number of users leads to more disparate recommendations. Let's inspect how the number of users affects the relation between unique and total of recommended in each dataset.

In [86]:
```
data_first_ds = {'n_user': [1,2,3,4,5,6,7,8,9,10], 'total_recommended': [279,579,839,1139,1379,1677,1937,2193,2453,2713],'unique_
first_ds_table = pd.DataFrame(data=data_first_ds)
```

In [87]:
```
first_ds_table['percentage_first_ds']=(first_ds_table['unique_recommended']/first_ds_table['total_recommended'])*100
```

In [88]:
```
first_ds_table['percentage_first_ds']=first_ds_table['percentage_first_ds'].round(1)
```

In [89]:
```
data_second_ds = {'n_user': [1,2,3,4,5,6], 'total_recommended': [340,680,1020,1379,1700,2040],'unique_recommended': [175,242, 284
second_ds_table = pd.DataFrame(data=data_second_ds)
```

In [90]:
```
second_ds_table['percentage_second_ds']=(second_ds_table['unique_recommended']/second_ds_table['total_recommended'])*100
```

In [91]:
```
second_ds_table['percentage_second_ds']=second_ds_table['percentage_second_ds'].round(1)
```

In [92]:
```
data_third_ds = {'n_user': [1,2,3,4,5,6], 'total_recommended': [879,1759,2636,3515,4395,5273],'unique_recommended': [433,615,744,
third_ds_table = pd.DataFrame(data=data_third_ds)
```

In [93]:
```
third_ds_table['percentage_third_ds']=(third_ds_table['unique_recommended']/third_ds_table['total_recommended'])*100
```

In [94]:
```
third_ds_table['percentage_third_ds']=third_ds_table['percentage_third_ds'].round(1)
```

In [95]:
```
percentage_table=first_ds_table[['n_user','percentage_first_ds']].set_index('n_user').join(second_ds_table[['n_user','percentage_
    third_ds_table[['n_user','percentage_third_ds']].set_index('n_user'), lsuffix='n_user')
```

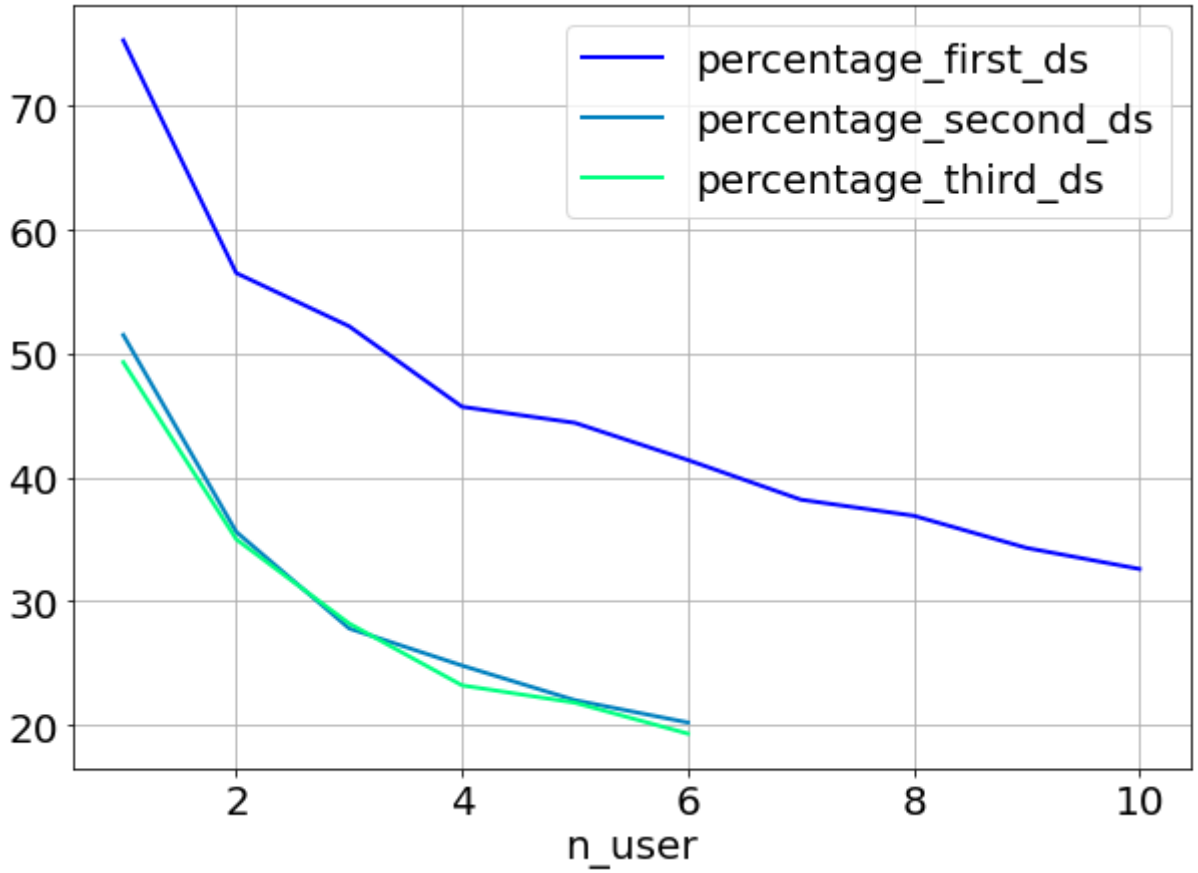
In [96]:
```
percentage_table
```

Out[96]:

n_user	percentage_first_ds	percentage_second_ds	percentage_third_ds
1	75.3	51.5	49.3
2	56.5	35.6	35.0
3	52.2	27.8	28.2
4	45.7	24.8	23.2
5	44.4	22.0	21.8
6	41.4	20.2	19.3
7	38.2	NaN	NaN
8	36.9	NaN	NaN
9	34.3	NaN	NaN
10	32.6	NaN	NaN

In [97]:
```
percentage_table=percentage_table.reset_index()
```

In [98]:
```
plt.rc('font', size=20)
percentage_table.plot(x='n_user',figsize=(10, 7), grid=True, colormap='winter', linewidth=2.0)
plt.legend(bbox_to_anchor=(1.0, 1.0))
plt.savefig('p_unique_recommended_to_user_num.jpeg')
```



In all datasets, the relation of unique recommended to the total of recommended remains stable. In case of second and third dataset, which were collected within a shorter peroid of time, there are more compact sets of recommended. However, even in the first dataset, with longer period of data collection, there is a similar relation at a higher percentual level. This demonstrates how the data collections are affected by the number of users and duration of time collection. Six users seem to be a reasonable compromise to get a base for comparative analysis without accumulating too much noise. A longer duration seems also to be preferable to reduce the possibility of influencing the working of the algorithm through repetitive actions. However, this small test on methodology of data collection is exemplatory, there are more tests on different topics and conditions needed.

Annotation of second and third dataset

For the annotation of second and thrid dataset, the annotated videos of the first dataset are used in the first place. For the rest, the relevance was estimated based on presence of keyword in hashtags and titles of videos. Video information was downloaded via Youtube-DL (on 13.07.2021). Since many videos are not accessible since then, I provide the file with video infos as part of dataset.

In [99]:
```
full_video_infos=pd.read_excel('all_datasets_df_with_tags.xlsx')
del full_video_infos['Unnamed: 0']
```

In [100…
```
full_video_infos['recommendedVideoId'].nunique()
```

Out[100]:
```
1977
```

In [101…
```
full_video_infos.shape
```

Out[101]:
```
(10245, 33)
```

In [102…
```
annotation=df[['recommendedVideoId', 'evaluation', 'relevance']]
```

In [103…
```
annotation=annotation.drop_duplicates(subset='recommendedVideoId')
```

In [104…
```
annotation=annotation.set_index('recommendedVideoId')
```

In [105…
```
full_video_infos=full_video_infos.set_index('recommendedVideoId')
```

In [106…
```
full_video_infos=full_video_infos.join(annotation, lsuffix='recommendedVideoId')
```

In [107…
```
full_video_infos['relevance']=full_video_infos['relevance'].fillna('unknown')
```

In [108…
```
full_video_infos.relevance.value_counts()
```

Out[108]:
```
unknown    5392
yes        2442
no         2411
Name: relevance, dtype: int64
```

In [109…
```
full_video_infos['evaluation']=full_video_infos['evaluation'].fillna('unknown')
```

In [110…
```
full_video_infos.evaluation.value_counts()
```

Out[110]:
```
unknown       5392
no            2413
positive      1507
negative       741
ambivalent     192
Name: evaluation, dtype: int64
```

In [111…
```
full_video_infos['tags']=full_video_infos['tags'].str.strip('[]')
```

In [112…
```
full_video_infos['watchedTags']=full_video_infos['watchedTags'].str.strip('[]')
```

In [113…
```
full_video_infos['tags']=full_video_infos['tags'].explode().dropna().groupby(level=0).agg(list)
```

In [114…
```
full_video_infos['watchedTags']=full_video_infos['watchedTags'].explode().dropna().groupby(level=0).agg(list)
```

In [115…
```
full_video_infos=full_video_infos.loc[full_video_infos['tags'].notnull()]
```

In [116…
```
full_video_infos=full_video_infos.loc[full_video_infos['watchedTags'].notnull()]
```

In [117…
```
len(full_video_infos)
```

Out[117]:
```
9560
```

In [118…
```
full_video_infos['intersection'] = [list(set(a).intersection(set(b))) for a, b in zip(full_video_infos.tags, full_video_infos.wat
```

In [119…
```
full_video_infos['N_intersection']=full_video_infos['intersection'].str.len()
```

In [120…
```
full_video_infos['N_intersection'].value_counts()
```

Out[120]:
```
0    9441
1     119
Name: N_intersection, dtype: int64
```

In [121…
```
full_video_infos.groupby('relevance').N_intersection.value_counts().to_frame()
```

Out[121]:

		N_intersection
relevance	**N_intersection**	
no	**0**	2124
	1	6
unknown	**0**	5006
	1	36
yes	**0**	2311
	1	77

The table above demostrates that absence of common tags seems to be a good indicator of irrelevant videos; however, the presence of common tags does not necessary indicate relevancy. This estimation was used to construct the graphs of the second and third dataset presented below.

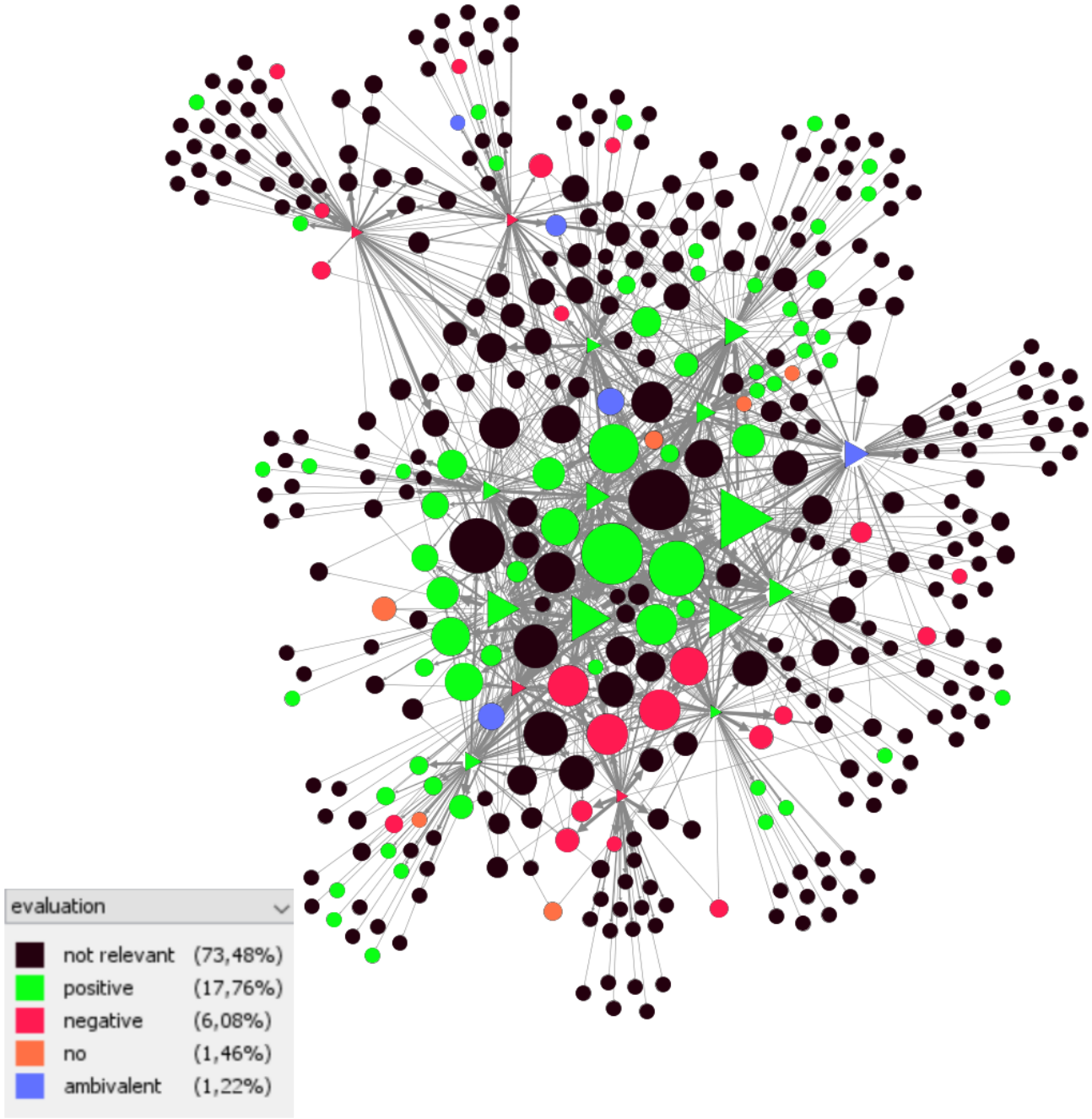
With the help of the annotation of the first dataset, I annotated 23% of unique videos, 13% of which are relevant. After this, I excluded videos with upload date before the event of the annexation (18 March 2014). This way I eliminated 7% of irrelevant recommendations. Next, I divided recommendations in two categories according to the presence of unique videos with shared tags: 48% with no shared tags and 52% with shared tags. Manual annotation of these recommendations proved, that among the first recommendations, 88% are not relevant. Hence, the absence of common tags seems to be a good indicator for irrelevance. Among the second, 33% are relevant, showing that the presence of shared tags does not clearly indicate relevant videos. However, this procedure facilitated the annotation: 49 new unique relevant videos (12% of unique recommended) together with the repetitions from the old dataset resulted in 25% relevant recommendations across the second dataset. On the graph below these videos are mostly located in the centre of the graph.

Gephi graph of the second dataset: layout ForceAtlas 2, noverlap, nodes ranking after in-degree between 10 and 50, watched videos as triangles, size ranked after in-degree, colours after annotated evaluation of Crimea's annexation in watched and recommended videos.

In [122…

```
Image(filename='evaluation_second_ds.png', width=700, height=700)
```

Out[122]:

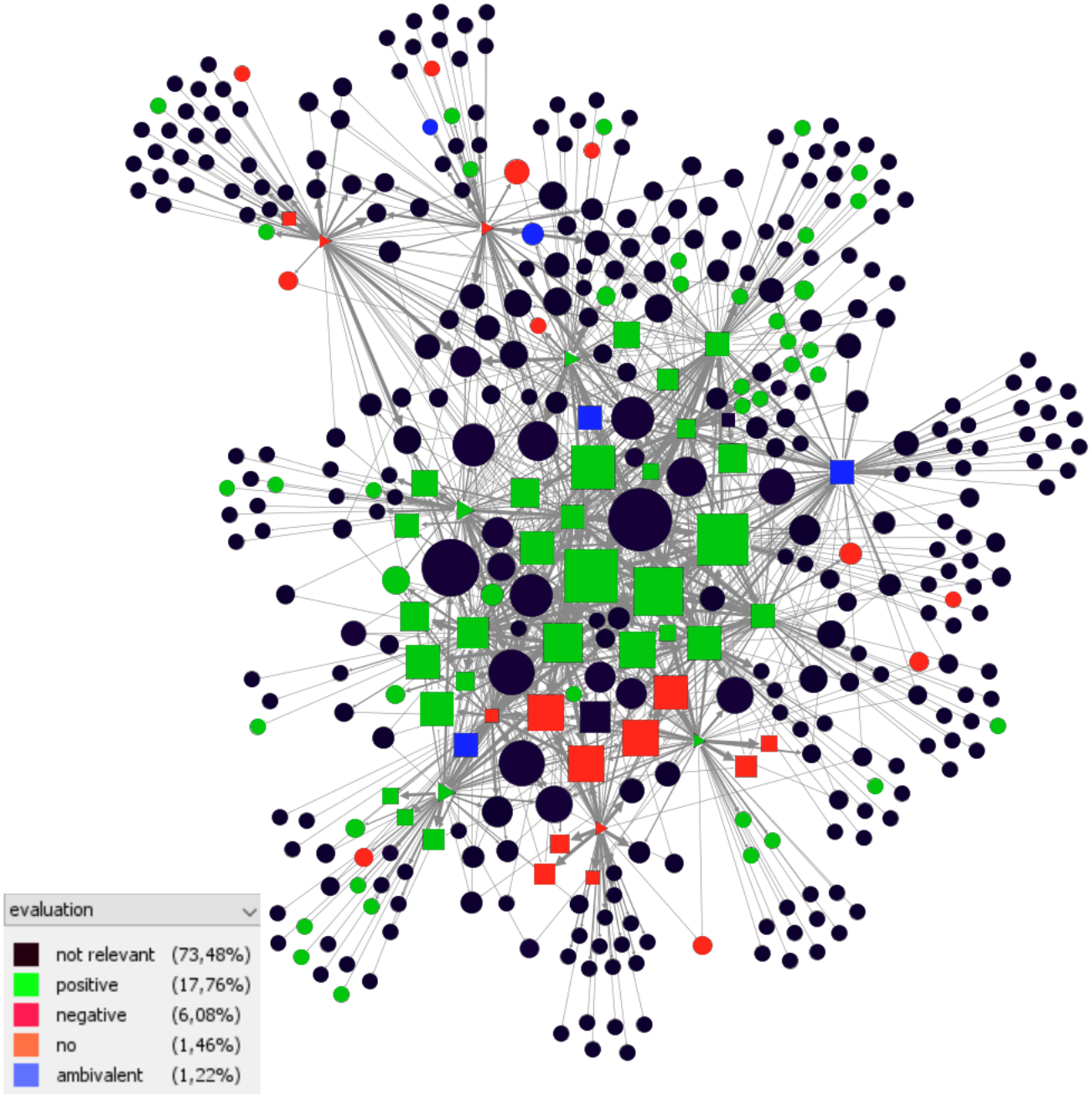

evaluation
not relevant (73,48%)
positive (17,76%)
negative (6,08%)
no (1,46%)
ambivalent (1,22%)

In the graph, positive evaluation of the Crimea annexation prevails and videos with a positive attitude towards the annexation are recommended after viewing contra-annexation videos as well. For the third data collection, the pragmatic decision was to explore the limits of this bubble by collecting recommendations for the top of relevant recommendations (44 videos, including several watched videos from the second data collection as they were frequently recommended again). The next graph shows this selection in the form of square nodes.

In [123…

```
Image(filename='selection_second_ds.png', width=700, height=700)
```

Out[123]:



Next visualisation shows the third dataset, annotated in the same way as the second dataset. Inside of the graph, watched videos build clusters,

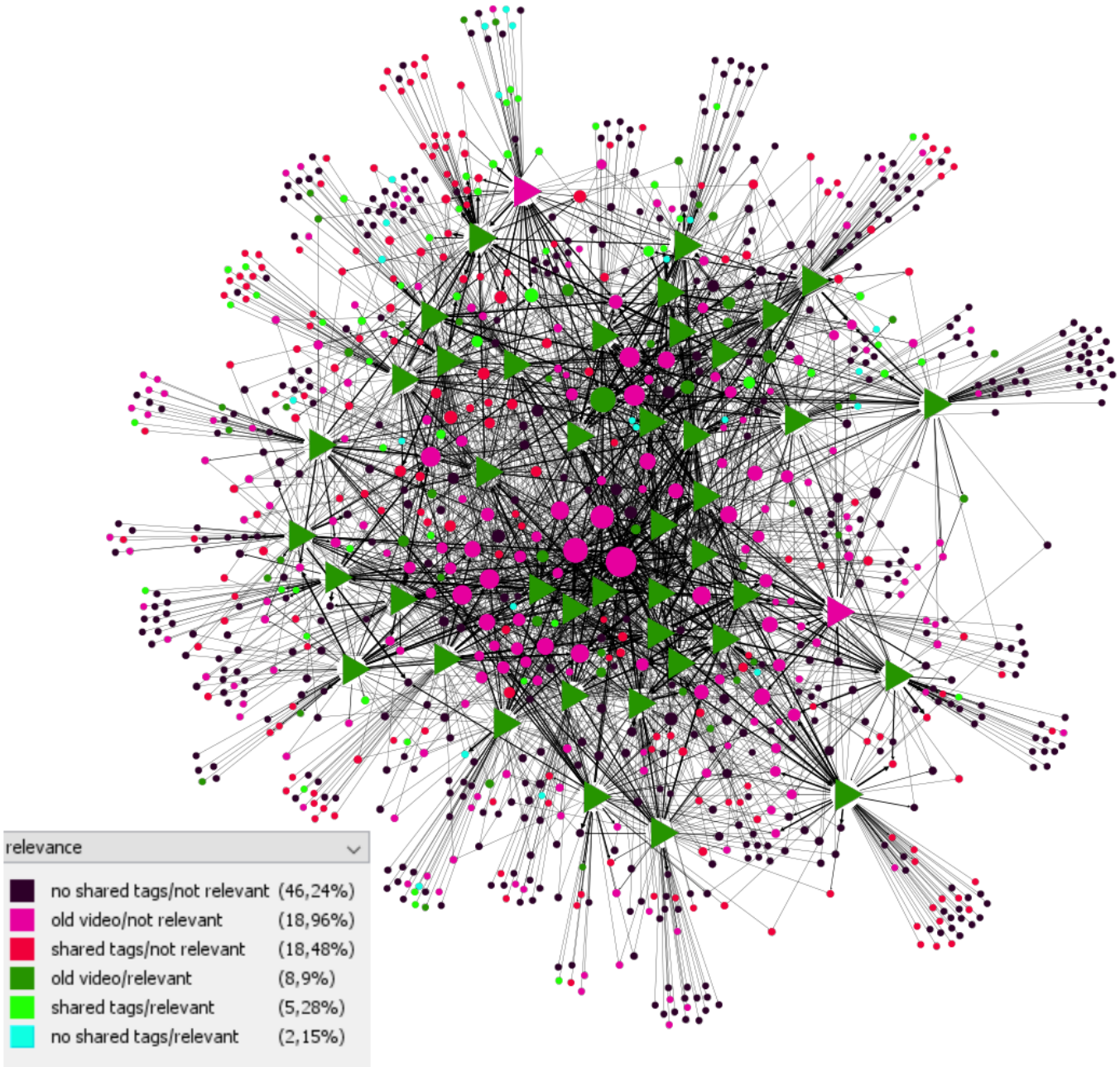
directly relating to each other or sharing mutual recommendations. However, these mutual recommendations are mostly not new. Among new ones, irrelevant recommendations prevail. New relevant videos are often found on the periphery of the graph, having been recommended once or twice.

Gephi graph of the third dataset: layout ForceAtlas2, Scaling=70, noverlap, nodes weights after in-degree between 10 and 50, watched videos as triangles with size 50, colours after coincidence of shared tags and annotated relevance.

In [124…

```
Image(filename='relevance_third_ds.png', width=700, height=700)
```

Out[124]:



The consolidation of three datasets in one makes an evaluation of most robust relations possible. After building and filtering the Gephi graph, we

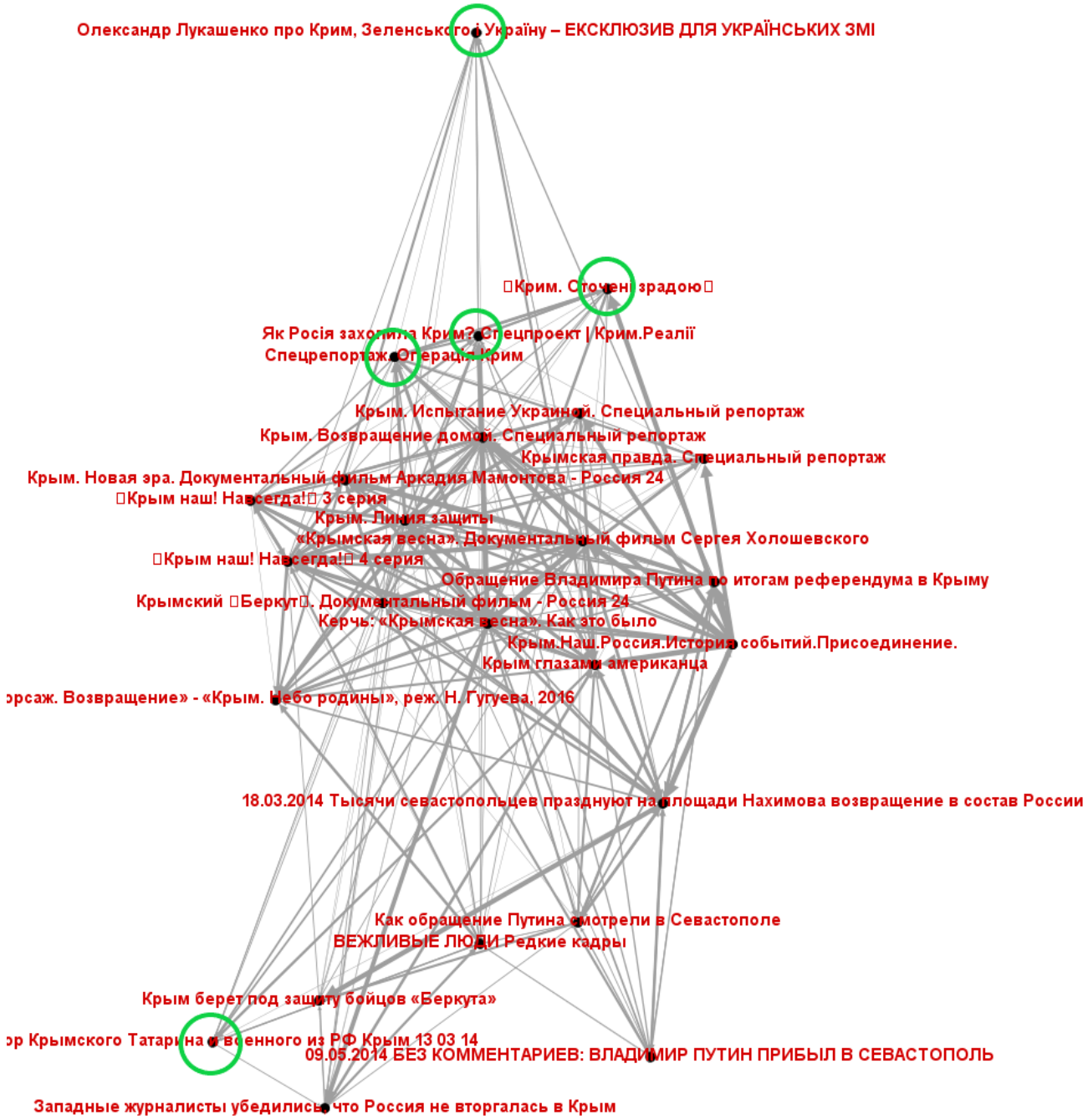
can easily recognise nodes with strong reciprocal correspondence.

Gephi graph of consolidated dataset from three data collections, filtered after mutual degree, range 3-13. Nodes with negative or ambivalent view on annexation are marked by green circles.

In [125…

```
Image(filename='mutual_relations_network.png', width=700, height=700)
```

Out[125]:


Олександр Лукашенко про Крим, Зеленського і Україну – ЕКСКЛЮЗИВ ДЛЯ УКРАЇНСЬКИХ ЗМІ
Крим. Сточені зрадою
Як Росія захопила Крим? Спецпроект | Крим.Реалії
Спецрепортаж. Операція "Крим
Крым. Испытание Украиной. Специальный репортаж
Крым. Возвращение домой. Специальный репортаж
Крымская правда. Специальный репортаж
Крым. Новая эра. Документальный фильм Аркадия Мамонтова - Россия 24
Крым наш! Навсегда! 3 серия
Крым. Линия защиты
«Крымская весна». Документальный фильм Сергея Холошевского
Крым наш! Навсегда! 4 серия
Обращение Владимира Путина по итогам референдума в Крыму
Крымский Беркут. Документальный фильм - Россия 24
Керчь: «Крымская весна». Как это было
Крым.Наш.Россия.История событий.Присоединение.
Крым глазами американца
орсаж. Возвращение» - «Крым. Небо родины», реж. Н. Гугуева, 2016
18.03.2014 Тысячи севастопольцев празднуют на площади Нахимова возвращение в состав России
Как обращение Путина смотрели в Севастополе
ВЕЖЛИВЫЕ ЛЮДИ Редкие кадры
Крым берет под защиту бойцов «Беркута»
ор Крымского Татарина военного из РФ Крым 13 03 14
09.05.2014 БЕЗ КОММЕНТАРИЕВ: ВЛАДИМИР ПУТИН ПРИБЫЛ В СЕВАСТОПОЛЬ
Западные журналисты убедились что Россия не вторгалась в Крым