Election Violence Newspaper Article Selection Process Pilot

August 6, 2018

1 Summary

The document below provides code to implement the keyword selection process (and later the plan is that it will describe the whole workflow of article selection).

The main point so far is to use classifier suggestions to expand keywords so that we capture cases of election violence. The highly predictive keyword lists can be found in Section 8.

Should we use highly predictive keywords (which might reinforce patterns we have found) or should we use words whose predictivity is changing between first and second stage classification.

As a first attempt (not particularly principled) I suggest taking:

one from highly predictive in stage 1 classification rough

one from highly predictive in stage 2 classification mob

one from becomes highly predictive from stage 1 to stage 2 stone

Note: Here the highly predictive keywords are taken only from the description (not the full ocr). If we want to use the King method then this is necessary because we won't have the full ocr for unclassified cases in general. However, if we use the first stage classifier we could use the full ocr. Lists from first stage classifier with full ocr can be found in Section 10

2 Packages used

```
library(durhamevp)
library(tidyverse)
library(quanteda)
```

3 Download Classified Cases

There is an initial set of cases which represent the concept (in this case 'election violence'). We also have a set of cases which do not represent the concept (Note: King et al do not have the non-election violence cases).

I also add indicators that this material is classified (`classified==1`) and that it is not unclassified (`unclass==0`) which will be useful when added to subsequent material.

```
# In fact we have some classified docs non-EV cases here King's R set doesn't contain any no
# I think we should use our more extensive information here

classdocs<-durhamevp::get_classified_docs()

classdocs<-classdocs %>%
  dplyr::mutate(std_url = sub("download/", "", url)) %>%
  dplyr::mutate(unclass=0, classified=1)
```

4 Download Initial 1832 Cases

We run three searches on the BNA. election riot, election incident, election disturbance. The code below returns these search results (including the description text) as stored on the database.

A (slightly adjusted) version of the url is used to detect and remove material which is in the classified set above.

I also add indicators that this material is not classified (`classified==0`) and that it is unclassified (`unclass==1`) which will be useful when added to the classified material above.

(Note: infact these searches have been run multiple times in 1832 probably for experimentation reasons - presumably because they were run very close in time to each other with no new material added the results are identical - for this reason I have just chosen one instance of each of the three searches).

```
all_searches<-get_archivesearches()

## Warning:  package 'bindrcpp' was built under R version 3.4.4
## Warning in value[[3L]](cond):  Object of class MySQLConnection could
not be destroyed properly, but was successfully removed from pool.
Error message:  internal error in RS_DBI_getConnection:  corrupt connection
handle
## Warning:  It wasn't possible to activate and/or validate the object.
Trying again with a new object.

initial_1832_searches<-all_searches%>%
  dplyr::select(id, search_text, archive_date_start, archive_date_end) %>%
```

```
  filter(archive_date_start>lubridate::ymd("1832-01-01"), archive_date_start<lubridate::ymd(
  filter(search_text %in% c("election riot", "election incident", "election disturbance"))

# a number of precisely duplicated searches here which return precisely duplicated results -
# election riot - id 73
# election disturbance - id 81
# election incident - id 85
res_i_1832<-get_archivesearchresults(archive_search_id = c(73, 81, 85)) %>%
  left_join(all_searches, by=c("archive_search_id"="id")) %>%
  mutate(std_url = sub("download/", "", url))

is_classified <-res_i_1832$std_url %in% classdocs$std_url

unclass_i_1832 <- res_i_1832[!is_classified,] %>%
  mutate(unclass=1, classified=0)

all_docs<- bind_rows(classdocs,
                     unclass_i_1832)
all_docs$fakeid<-1:nrow(all_docs)
```

5 First stage classification

Here I use the classified non-election violence cases which we have. King et al do not have the equivalent - so they use the unclassified cases as non-cases.

```
all_corpus_d<-corpus(all_docs[c("fakeid", "description", "EV_article", "classified")], text_

all_dfm_d <- preprocess_corpus(all_corpus_d, stem=TRUE, min_termfreq=20, min_docfreq = 20)

class_dfm_d<-quanteda::dfm_subset(all_dfm_d, quanteda::docvars(all_dfm_d, "classified")==1)
class_nb <- quanteda::textmodel_nb(class_dfm_d, y=quanteda::docvars(class_dfm_d, "EV_article
first_stage_keywords<-nb_keywords(class_dfm_d, "EV_article")
```

6 Second stage classification

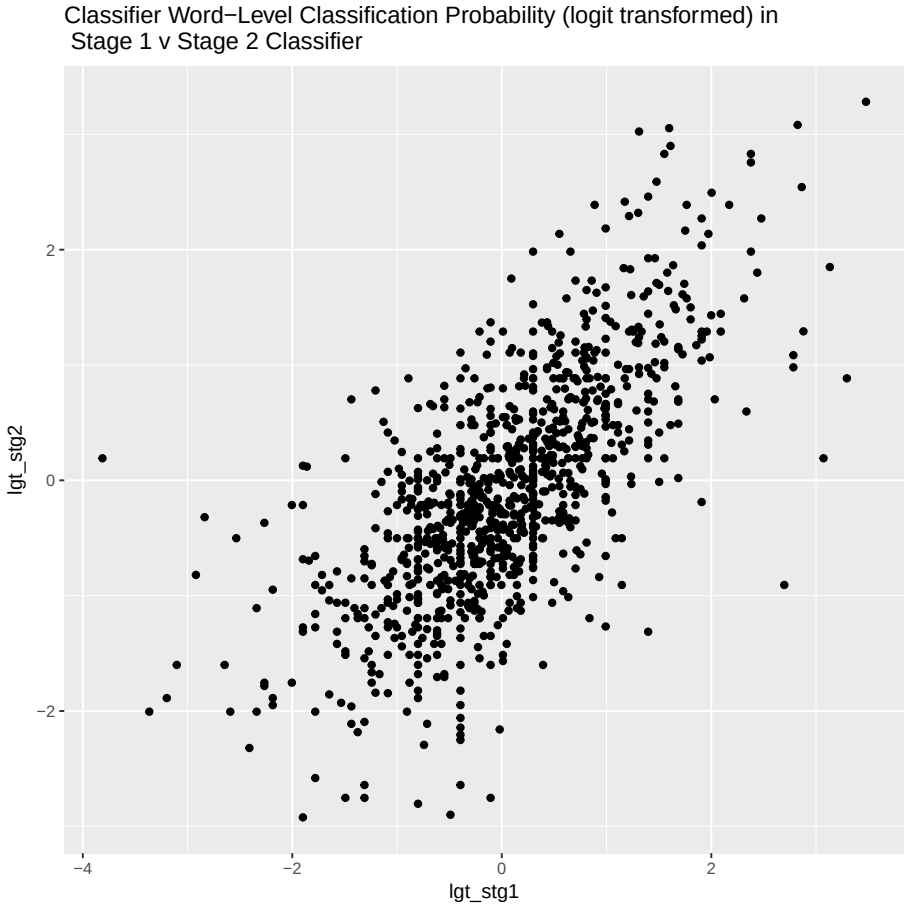
```
S_dfm <- quanteda::dfm_subset(all_dfm_d, quanteda::docvars(all_dfm_d, "classified")==0)
quanteda::docvars(S_dfm, "T")<-predict(class_nb, newdata = S_dfm, type="class")
second_stage_keywords<-nb_keywords(S_dfm, "T")
```

7 Connect results

Connect results from the two classification stages.

```
names(second_stage_keywords)<-c("keyword", "stg2_0", "stg2_1", "id")
names(first_stage_keywords)<-c("keyword", "stg1_0", "stg1_1", "id")
p_change<-left_join(first_stage_keywords, second_stage_keywords, by=c("keyword", "id")) %>%
  mutate(lgt_stg1 =log(stg1_1/stg1_0), lgt_stg2=log(stg2_1/stg2_0)) %>%
  mutate(abs_lgt_chng=abs(lgt_stg1-lgt_stg2)) %>%
  mutate(lgt_chng=lgt_stg2-lgt_stg1) %>%
  arrange(lgt_chng)

p_change%>%
  ggplot(aes(lgt_stg1, lgt_stg2))+
  geom_point() +
  ggtitle("Classifier Word-Level Classification Probability (logit transformed) in \n Stage
```



8 Keyword suggestions

From the first stage classifier (words most predictive of election violence articles) From the second stage classifier (words most predictive of predicted election violence articles) From the difference between the first and second stage classifers (words which become much more and much less predicitive of election violence articles between the two classification stages).

```
knitr::kable(head(first_stage_keywords, 30),
             digits = 2)
```

	keyword	stg1_0	stg1_1	id
949	assault	0.03	0.97	949
976	fight	0.04	0.96	976
971	outrag	0.04	0.96	971
963	durham	0.04	0.96	963
920	riot	0.05	0.95	920
951	rough	0.05	0.95	951
657	window	0.06	0.94	657
977	hill	0.06	0.94	977
981	ruffian	0.06	0.94	981
973	troubl	0.06	0.94	973
914	mob	0.08	0.92	914
957	throw	0.08	0.92	957
309	serious	0.08	0.92	309
970	warwick	0.08	0.92	970
974	lloyd	0.08	0.92	974
478	aris	0.09	0.91	478
967	robert	0.09	0.91	967
923	rioter	0.10	0.90	923
932	injur	0.11	0.89	932
953	band	0.11	0.89	953
903	club	0.12	0.88	903
960	wakefield	0.12	0.88	960
594	polic	0.12	0.88	594
602	labour	0.12	0.88	602
614	wednesday	0.12	0.88	614
402	afternoon	0.12	0.88	402
637	inquest	0.13	0.87	637
804	scene	0.13	0.87	804
909	cambridg	0.13	0.87	909
972	protect	0.13	0.87	972

```
knitr::kable(head(second_stage_keywords, 30),
             digits=2)
```

	keyword	stg2_0	stg2_1	id
949	assault	0.04	0.96	949
657	window	0.04	0.96	657
297	adjourn	0.05	0.95	297
918	riotous	0.05	0.95	918
302	stone	0.05	0.95	302
889	verdict	0.06	0.94	889
974	lloyd	0.06	0.94	974
970	warwick	0.06	0.94	970
630	summon	0.07	0.93	630
951	rough	0.07	0.93	951
960	wakefield	0.08	0.92	960
878	militari	0.08	0.92	878
394	dread	0.08	0.92	394
735	shot	0.08	0.92	735
808	struck	0.08	0.92	808
923	rioter	0.08	0.92	923
118	read	0.09	0.91	118
754	kill	0.09	0.91	754
914	mob	0.09	0.91	914
978	frome	0.09	0.91	978
988	stamford	0.10	0.90	988
926	crowd	0.10	0.90	926
614	wednesday	0.11	0.89	614
664	gaol	0.11	0.89	664
637	inquest	0.12	0.88	637
882	accompani	0.12	0.88	882
309	serious	0.12	0.88	309
937	shout	0.12	0.88	937
617	confus	0.13	0.87	617
737	wound	0.13	0.87	737

```
become_predictive<- p_change %>%
  filter(stg2_1>.8) %>%
  arrange(-lgt_chng)

become_less_predictive<- p_change %>%
  filter(stg1_1>.8) %>%
  arrange(lgt_chng)
knitr::kable(head(become_predictive, 30),
             digits = 2)
```

keyword	stg1_0	stg1_1	id	stg2_0	stg2_1	lgt_stg1	lgt_stg2	abs_lgt_chng	lgt_chng
riotous	0.21	0.79	918	0.05	0.95	1.31	3.02	1.71	1.71
accompani	0.43	0.57	882	0.12	0.88	0.30	1.98	1.68	1.68
front	0.48	0.52	753	0.15	0.85	0.09	1.75	1.66	1.66
gaol	0.37	0.63	664	0.11	0.89	0.55	2.14	1.59	1.59
struck	0.29	0.71	808	0.08	0.92	0.89	2.39	1.50	1.50
adjourn	0.17	0.83	297	0.05	0.95	1.60	3.05	1.46	1.46
shout	0.34	0.66	937	0.12	0.88	0.66	1.98	1.33	1.33
stone	0.17	0.83	302	0.05	0.95	1.61	2.90	1.29	1.29
verdict	0.17	0.83	889	0.06	0.94	1.55	2.83	1.28	1.28
dread	0.24	0.76	394	0.08	0.92	1.17	2.42	1.24	1.24
southern	0.43	0.57	948	0.18	0.82	0.30	1.53	1.23	1.23
stamford	0.27	0.73	988	0.10	0.90	0.99	2.18	1.19	1.19
summon	0.19	0.81	630	0.07	0.93	1.48	2.59	1.11	1.11
kill	0.23	0.77	754	0.09	0.91	1.22	2.29	1.08	1.08
militari	0.20	0.80	878	0.08	0.92	1.40	2.46	1.06	1.06
prison	0.33	0.67	348	0.15	0.85	0.70	1.73	1.03	1.03
read	0.21	0.79	118	0.09	0.91	1.30	2.32	1.02	1.02
nottingham	0.35	0.65	942	0.17	0.83	0.62	1.58	0.96	0.96
concern	0.30	0.70	25	0.15	0.85	0.86	1.73	0.87	0.87
friday	0.31	0.69	64	0.16	0.84	0.81	1.65	0.84	0.84
special	0.29	0.71	618	0.16	0.84	0.90	1.63	0.72	0.72
wm	0.27	0.73	853	0.16	0.84	0.99	1.67	0.68	0.68
constabl	0.24	0.76	834	0.14	0.86	1.16	1.84	0.68	0.68
nomin	0.31	0.69	946	0.19	0.81	0.78	1.44	0.66	0.66
shot	0.15	0.85	735	0.08	0.92	1.77	2.39	0.62	0.62
took	0.23	0.77	378	0.14	0.86	1.23	1.83	0.60	0.60
death	0.30	0.70	202	0.19	0.81	0.87	1.47	0.60	0.60
mid	0.31	0.69	728	0.20	0.80	0.81	1.40	0.59	0.59
confus	0.20	0.80	617	0.13	0.87	1.40	1.93	0.53	0.53
chairman	0.27	0.73	887	0.18	0.82	0.99	1.51	0.52	0.52

```
knitr::kable(head(become_less_predictive, 30),
             digits=2)
```

keyword	stg1_0	stg1_1	id	stg2_0	stg2_1	lgt_stg1	lgt_stg2	abs_lgt_chng	lgt_chng
troubl	0.06	0.94	973	0.71	0.29	2.70	-0.91	3.60	-3.60
durham	0.04	0.96	963	0.45	0.55	3.07	0.19	2.88	-2.88
section	0.20	0.80	986	0.79	0.21	1.40	-1.31	2.71	-2.71
fight	0.04	0.96	976	0.29	0.71	3.29	0.88	2.41	-2.41
oppos	0.13	0.87	975	0.55	0.45	1.91	-0.19	2.10	-2.10
ruffian	0.06	0.94	981	0.27	0.73	2.78	0.98	1.80	-1.80
aris	0.09	0.91	478	0.36	0.64	2.34	0.60	1.74	-1.74
hill	0.06	0.94	977	0.25	0.75	2.78	1.09	1.70	-1.70
whig	0.16	0.84	861	0.50	0.50	1.69	0.02	1.67	-1.67
riot	0.05	0.95	920	0.22	0.78	2.88	1.29	1.59	-1.59
freedom	0.18	0.82	851	0.50	0.50	1.50	-0.01	1.52	-1.52
met	0.17	0.83	965	0.45	0.55	1.55	0.19	1.36	-1.36
club	0.12	0.88	903	0.33	0.67	2.03	0.70	1.33	-1.33
outrag	0.04	0.96	971	0.14	0.86	3.13	1.85	1.28	-1.28
wa	0.20	0.80	693	0.45	0.55	1.40	0.19	1.21	-1.21
free	0.16	0.84	551	0.38	0.62	1.69	0.49	1.19	-1.19
radic	0.17	0.83	933	0.38	0.62	1.61	0.48	1.13	-1.13
buri	0.18	0.82	847	0.40	0.60	1.50	0.41	1.09	-1.09
came	0.20	0.80	897	0.42	0.58	1.40	0.31	1.09	-1.09
sit	0.20	0.80	561	0.41	0.59	1.40	0.35	1.05	-1.05
termin	0.16	0.84	815	0.34	0.66	1.69	0.68	1.00	-1.00
norfolk	0.16	0.84	773	0.33	0.67	1.69	0.70	0.98	-0.98
labour	0.12	0.88	602	0.26	0.74	1.99	1.07	0.92	-0.92
tavistock	0.20	0.80	983	0.38	0.62	1.40	0.50	0.90	-0.90
citizen	0.13	0.87	980	0.26	0.74	1.91	1.04	0.87	-0.87
disgrac	0.16	0.84	604	0.31	0.69	1.66	0.82	0.84	-0.84
band	0.11	0.89	953	0.22	0.78	2.09	1.29	0.80	-0.80
ion	0.20	0.80	760	0.36	0.64	1.40	0.60	0.80	-0.80
session	0.19	0.81	117	0.34	0.66	1.45	0.68	0.77	-0.77
robert	0.09	0.91	967	0.17	0.83	2.31	1.58	0.74	-0.74

9 Some descriptives of search returns and overlaps

The 1832 searches with the three terms (election riot, election disturbance, election incident) returned a total of 2744 unique articles. Below are the numbers and proportion of total returns from combinations of the initial search terms and from each search term independently.

```
res_32_des<-res_i_1832 %>%
  group_by(url) %>%
  mutate(search_text=sub(" ", "_", search_text)) %>%
  select(url, search_text) %>%
```

```
  mutate(value=1) %>%
  spread(search_text, value, fill=0) %>%
  mutate(tot_art=nrow(.))

knitr::kable(res_32_des %>%
  group_by(election_riot, election_incident, election_disturbance) %>%
  summarize(n=length(tot_art), prop=n/tot_art[1])%>%
    arrange(-prop), digits = 2)
```

election_riot	election_incident	election_disturbance	n	prop
1	0	0	1395	0.51
0	0	1	708	0.26
1	0	1	352	0.13
0	1	0	198	0.07
0	1	1	35	0.01
1	1	1	29	0.01
1	1	0	27	0.01

```
knitr::kable(res_32_des %>%
               group_by(election_riot) %>%
               summarize(n=length(tot_art), prop=n/tot_art[1]) %>%
               arrange(-election_riot),
             digits = 2)
```

election_riot	n	prop
1	1803	0.66
0	941	0.34

```
knitr::kable(res_32_des %>%
               group_by(election_incident) %>%
               summarize(n=length(tot_art), prop=n/tot_art[1])%>%
               arrange(-election_incident),
             digits = 2)
```

election_incident	n	prop
1	289	0.11
0	2455	0.89

```
knitr::kable(res_32_des %>%
               group_by(election_disturbance) %>%
               summarize(n=length(tot_art), prop=n/tot_art[1])%>%
               arrange(-election_disturbance),
             digits = 2)
```

election_disturbance	n	prop
1	1124	0.41
0	1620	0.59

10 First stage classification on full OCR

```
all_corpus_o<-corpus(all_docs[c("fakeid", "ocr", "EV_article", "classified")], text_field =
all_dfm_o <- preprocess_corpus(all_corpus_o, stem=TRUE, min_termfreq=20, min_docfreq = 20)
class_dfm_o<-quanteda::dfm_subset(all_dfm_o, quanteda::docvars(all_dfm_d, "classified")==1)
class_nb_ocr <- quanteda::textmodel_nb(class_dfm_o, y=quanteda::docvars(class_dfm_o, "EV_art
first_stage_keywords_ocr<-nb_keywords(class_dfm_o, "EV_article")

knitr::kable(head(first_stage_keywords_ocr, 30), digits = 2)
```

	rowname	0	1	id
8827	truncheon	0.03	0.97	8827
8474	smash	0.05	0.95	8474
8824	p.c	0.06	0.94	8824
8649	missil	0.07	0.93	8649
7160	rioter	0.08	0.92	7160
8261	riotous	0.08	0.92	8261
8670	brickbat	0.10	0.90	8670
8179	affray	0.11	0.89	8179
8823	someon	0.12	0.88	8823
8108	quell	0.13	0.87	8108
6733	bludgeon	0.13	0.87	6733
8813	suffragist	0.14	0.86	8813
8440	cricket	0.14	0.86	8440
3551	mob	0.15	0.85	3551
8715	dump	0.16	0.84	8715
8148	rabbl	0.16	0.84	8148
6955	infuri	0.17	0.83	6955
2958	riot	0.17	0.83	2958
8630	volley	0.17	0.83	8630
8702	desist	0.17	0.83	8702
8705	polio	0.17	0.83	8705
6761	stave	0.17	0.83	6761
7360	mauric	0.18	0.82	7360
6396	baird	0.18	0.82	6396
8540	partizan	0.18	0.82	8540
8822	rosett	0.18	0.82	8822
8817	telegram	0.19	0.81	8817
8668	sewel	0.19	0.81	8668
8593	sandon	0.19	0.81	8593
8687	tumultu	0.19	0.81	8687