Supporting Materials

Measuring and understanding parties' anti-elite strategies

Hauke Licht* Tarik Abou-Chadi† Pablo Barberá‡ Whitney Hua§

October 31, 2023

Contents

A Included countries and parties **3**
A.1 Countries 3
A.2 Parties 4

B Coding scheme development and validation **12**
B.1 Background: crowd-sourced measurements of populist rhetoric 12
B.2 First version: Discriminating between elite criticism and other valence attacks 14
B.3 Second version: Increasing conceptual depth 16
B.4 Third version: Reducing the cognitive load to facilitate reliable coding . . 18

C Generating our labeled dataset **21**
C.1 Sampling of tweets for the first round of crowd coding 21
C.2 Results of the first round of crowd-sourced elite criticism coding 25
C.3 Judgment aggregation: Inducing tweet-level labels from first-round crowd codings 29
C.4 Learning to detect elite criticism tweets based on the first sample 31
C.5 Sampling from machine-labeled tweets for the second round of crowd coding 34
C.6 Results of the second round of crowd-sourced coding 35
C.7 Judgment aggregation: Inducing tweet-level labels from pooled crowd codings 39

*University of Cologne. Corresponding author. Contact: hauke.licht@wiso.uni-koeln.de
†University of Oxford
‡University of Southern California
§University of Southern California

D Classifier training, evaluation, and validation **42**
D.1 Learning approaches 42
D.2 Training strategies 43
D.3 Classifier evaluation 44
D.4 Explaining the classifier's predictions 46

E Additional measurement validation **50**
E.1 Validation against CHES data 50
E.2 Comparison to dictionary-based measurements 50

F Regression analyses **54**
F.1 Data 54
F.2 Regression tables 56

G References **58**

A Included countries and parties

A.1 Countries

Table S.1 lists the countries included in our study along with their ISO 3166-1 alpha-2 and alpha-3 codes.

Table S.1: Country coverage. Alpha-2 and alpha-3 country codes according to ISO 3166-1.

	Country code	
Country name	alpha-2	alpha-3
Australia	AU	AUS
Austria	AT	AUT
Belgium	BE	BEL
Canada	CA	CAN
Denmark	DK	DNK
Finland	FI	FIN
France	FR	FRA
Germany	DE	DEU
Greece	GR	GRC
Ireland	IE	IRL
Italy	IT	ITA
Luxembourg	LU	LUX
Netherlands	NL	NLD
New Zealand	NZ	NZL
Norway	NO	NOR
Portugal	PT	PRT
Spain	ES	ESP
Sweden	SE	SWE
Switzerland	CH	CHE
United Kingdom	GB	GBR

A.2 Parties

Table S.2 reports the parties included in our analyses along with their Twitter handles, numeric identifiers (IDs), and total numbers of tweets. Data to identify parliamentary parties in our sample was drawn from the *Parliaments and Government* (ParlGov) database (Döring and Manow, 2019).

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (ibid.). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded.

Country	Party Abbr.	Party Name	Party ID	Twitter account	*N* tweets
AUS	AG	Australian Greens	751	@Greens	1646
	ALP	Australian Labor Party	1253	@AustralianLabor	2513
	CouLP	Country Liberal Party	215	@CountryLibs	570
	Katter	Katter's Australian Party	2258	@KAPteam	651
				@RealBobKatter	890
	LNPQ	Liberal National Party of Queensland	154	@LNPQLD	1199
	LPA	Liberal Party of Australia	1411	@LiberalAus	2119
	NCP|NPA	National (Country) Party | National Party of Australia	184	@The_Nationals	1056
	NXT	Nick Xenophon Team	2630	@centre_alliance	121
	PUP	Palmer United Party	2259	@PalmerUtdParty	576
				@UnitedAusParty*	112
AUT	FPO	Freedom Party of Austria	50	@FPOE_TV*	1203
				@HCStracheFP	10320
	Gruene	The Greens – The Green Alternative	1429	@Gruene_Austria	5036
	NEOS	NEOS – The New Austria	2255	@neos_eu	8345
	OVP	Austrian People's Party	1013	@volkspartei	2308
	SPO	Social Democratic Party of Austria	973	@SPOE_at	3415

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	*N* tweets
BEL	AGL-Gr	Agalev – Green	1594	@groen	3092
	CVP	Flemish Christian Peoples Party	723	@cdenv	3269
	Ecolo	Confederated ecologists for the organisation of original struggles	161	@Ecolo	3388
	FDF	Francophone Democratic Front	969	@defi_eu	1988
	MR	Reformist Movement	915	@MR_officiel	2656
	N-VA	New Flemish Alliance	501	@de_NVA	18996
	PA-PTB	Workers' Party of Belgium	256	@ptbbelgique	3846
				@pvdabelgie	3513
	PS	Socialist Party [Francophone]	1378	@PSofficiel	1491
	PSC-CDH	Francophone Christian Social Party – Humanist Democratic Centre	1192	@lecdh	1431
	PVV|VLD	Party of Liberty and Progress | Flemish Liberals and Democrats	1110	@openvld	2678
	Pp	People's Party	438	@LePeuple2	3439
	SP	Socialist Party	1029	@sp_a	3207
	VB	Flemish Block	993	@vlbelang	3313
CAN	BQ	Quebec Bloc	448	@BlocQuebecois	2388
	CCF|NDP	Co-operative Commonwealth Federation | New Democratic Party	296	@NDP*	12245
	CPC	Conservative Party of Canada	1255	@CPC_HQ	2347
				@PCC_HQ	1861
	GPC	Green Party of Canada	1259	@CanadianGreens	3093
	LP	Liberal Party of Canada	368	@liberal_party	3110

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	N tweets
CHE	BDP	Conservative Democratic Party of Switzerland	1213	@BDPSchweiz	1684
	EDU-UDF	Federal Democratic Union of Switzerland	1318	@EDUSchweiz	259
	EVP-PEP	Protestant Peoples Party	602	@evppev	1094
	FDP	FDP.The Liberals	26	@FDP_Liberalen	1019
	GPL-PVL	Green Liberal Party	308	@grunliberale	1594
	Grue	Greens	141	@GrueneCH	2537
	KK/CVP	Catholic Conservative / Christian Democratic Peoples Party	531	@CVP_PDC	3760
	LdT	Ticino League	1500	@LEGAdeiTicinesi	473
	MCR	Geneva Citizens' Movement	2599	@mcglecitoyen	27
	SP-PS	Social Democratic Party of Switzerland	35	@spschweiz	2853
	SVP-UDC	Swiss People's Party	750	@SVPch	1563
DEU	AfD	Alternative for Germany	2253	@AfD	2521
	B90/Gru	Alliance 90 / Greens	772	@Die_Gruenen	3257
	CDU+CSU	Christian Democratic Union / Christian Social Union	1727	@cducsubt	4676
	FDP	Free Democratic Party	543	@fdp	3368
	Li/PDS	The Left / PDS	791	@dieLinke	3204
	SPD	Social Democratic Party of Germany	558	@spdde	6206
DNK	A	The Alternative	2567	@alternativet_	5125
	DF	Danish Peoples Party	1418	@DanskDf1995	633
	En-O	Red-Green Alliance	306	@Enhedslisten	1903
	IA	Community of the People	1891	@IAtaqatigiit	43
	KF	Conservatives	590	@KonservativeDK	2699
	NLA	New-Liberal Alliance	376	@LiberalAlliance	2555
	RV	Danish Social Liberal Party	211	@radikale	3971
	SF	Socialist Peoples Party	1644	@SFpolitik	2354
	Sbf	Union Party (Faroe Islands)	1892	@Sambandsflokkur *	19
	Sd	Social Democrats	1629	@Spolitik	1176
	Si	Forward (Greenland)	74	@Siumut	31
	V	Liberal Party	1605	@venstredk	1441

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	*N* tweets
	AP-P	People's Alliance-Party	645	@ppopular *	96382
	BNG	Galician Nationalist Block	520	@obloque	3008
ESP	Bai	Yes	385	@NafarroaBai2011 *	1154
	C-PC	Citizens – Party of the Citizenry	2375	@CiudadanosCs	4337
	CA	Aragonese Council	1367	@chunta	40
	CC	Canary Coalition	234	@coalicion	4155
	CUP	Popular Unity Candidacy–For Rupture	2792	@cupnacional	979
	C|AV	Compromise | A la valenciana	2607	@compromis	8351
	EA	Basque Solidarity	845	@ealkartasuna	772
	ECP	In Common We Can	2606	@EnComu_Podem	5116
	EHB	Basque Country Unite	2603	@ehbildu	9698
	EM	In Tide	2604	@En_Marea	9725
	ERC	Republican Left of Catalonia	757	@Esquerra_ERC	14111
	JxCat	Together for Catalonia	2722	@JuntsXCat	6627
	MP	More Country	NaN	@MasPais_Es	133
	NA+	Sum Navarre	2723	@navarra_suma	900
	P	We Can	2376	@PODEMOS *	94026
	PCE|IU	Communist Party | United Left	118	@iunida	1796
	PNV	Basque Nationalist Party	1361	@eajpnv	6051
	PRC	Regionalist Party of Cantabria	NaN	@prcantabria	1072
	PSOE	Spanish Socialist Workers Party	902	@PSOE	12238
	TE	Teruel Exists	NaN	@TeruelExiste_	1235
	UPyD	Union, Progress and Democracy	551	@UPYD	1366
	Vox	Voice	2380	@vox_es	800
	DL|VAS	Democratic Union | Left Alliance	1292	@vasemmisto	2675
FIN	KD	Christian Democrats	1463	@KDpuolue	3468
	KESK	Centre Party	94	@keskusta	2908
	KOK	National Coalition Party	1118	@kokoomus	3475
	RKP-SFP	Swedish People's Party	585	@sfprkp	2646
	SP|P	Finnish Party | True Finns	200	@persut	3013
	SSDP	Social Democratic Party of Finland	395	@Demarit	3566
	UV|SIN	New Alternative | Blue Reform	2645	@SiniTulevaisuus	841
	VIHR	Green League	1062	@vihreat	3797

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	N tweets
	AC	Centrist Alliance	2078	@AllianceC	405
	DLR|DLF	Arise the Republic | France Arise	2399	@DLF_Officiel	6391
	FI	Unbowed France	2644	@FranceInsoumise	4731
FRA	FN	National Front	270	@RNational_off	5188
	NC	New Centre	1058	@LesCentristes_	544
	PCF	French Communist Party	686	@PCF	2303
	PRG	Radical Party of the Left	1492	@PartiRadicalG	234
	PS	Socialist Party	1539	@partisocialiste	4656
	REM	The Republic Onwards!	2643	@enmarchefr	1822
	UDF|MD	Union for French Democracy | Democratic Movement	509	@MoDem	772
	UDI	Union of Democrats and Independents	2273	@UDI_off	3327
	UMP|LR	Union for a Popular Movement | The Republicans	658	@lesRepublicains	3454
	V	Greens	873	@EELV	2309
	Con	Conservatives	773	@Conservatives	3622
	DUP	Democratic Unionist Party	319	@duponline	3212
GBR	GP	Green Party	467	@TheGreenParty	4129
	Lab	Labour	1556	@UKLabour	3668
	Lib	Liberals	659	@LibDems	3850
	Plaid	Plaid Cymru	311	@Plaid_Cymru	5742
	SDLP	Social Democratic and Labour Party	1023	@SDLPlive	2021
	SF	Sinn Fein	689	@sinnfeinireland	3866
	SNP	Scottish National Party	1284	@theSNP	3487
	UKIP	United Kingdom Independence Party	1272	@UKIP	1529
	UUP	Ulster Unionist Party	1210	@uuponline	1317
GRC	ANEL	Independent Greeks	2091	@anexartitoi	2696
	DIMAR	Democratic Left	2093	@dimokratiki	587
	EK	Union of Centrists	2165	@antidiaploki	2645
	KKE	Communist Party of Greece	614	@gt_kke *	1728
	LAOS	Popular Orthodox Rally	1179	@laos_Hellas	279
	LE	Popular Unity	2596	@LAE_epikoinonia	20
	ND	New Democracy	47	@neademokratia	2494
	PASOK	Panhellenic Socialist Movement	1338	@pasok	2603
	SYRIZA	Coalition of the Radical Left	1592	@syriza_gr	3695
	TP	The River	2346	@ToPotami	5492

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	*N* tweets
IRL	DS	Social Democrats	2619	@SocDems	1887
	FF	Fianna Fail	280	@fiannafailparty	3529
	FG	Fine Gael (Familiy of the Irish)	1393	@FineGael	3430
	Green	Green Party	1573	@greenparty_ie	2583
	IA	Independent Alliance	2622	@IndepAlliance	462
	IC	Independents 4 Change	2621	@independents4_C	2
	Lab	Labour Party	318	@labour	3572
	PBPA	People Before Profit Alliance	1804	@pb4p	896
	SF	Sinn Fein	2217	@sinnfeinireland	3866
	SP	Socialist Party	1014	@SocialistParty	815
ITA	AIE	Associative Italians Abroad	927	@maie_mondiale	42
	CD	Democratic Centre	2153	@ilCentroDem	1402
	FI-PdL	Go Italy – The People of Freedom	596	@forza_italia	3689
	FdI-CN	Brothers of Italy – National Centre-right	2154	@FratellidItalia	5131
	LN	North League	1436	@LegaSalvini	11208
	M5S	Five Star Movement	2155	@Mov5Stelle	3067
	PD	Democratic Party	382	@pdnetwork	3997
	R	Radicals	1296	@Radicali	1187
	SC	Civic Choice	2156	@scelta_civica	2094
	SVP	South Tyrol Peoples Party	1030	@SVP_Suedtirol	709
	UC	Union / Centre	226	@UdcIta	1136
	USEI	South American Union Italian Emigrants	2671	@useiufficiale	682
	UV	Valdotanian Union	974	@unionvaldotaine	474
LUX	AR|ADR	Action Committee Pensions | Alternative Democratic Reform Party	1582	@adr_lu	1480
	CSV	Christian Social People's Party	1234	@CSV_news	725
	DL	The Left	457	@dei_lenk	776
	DP	Democratic Party	967	@dp_lu	950
	Greng	The Greens	310	@deigreng	1062
	LSAP	Luxembourg Socialist Workers' Party	701	@lsap_lu	999
	Pi	Pirate Party Luxembourg	2256	@Piratepartei	540

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	N tweets
NLD	50+	50PLUS	2109	@50pluspartij	1538
	CDA	Christian Democratic Appeal	235	@cdavandaag	4908
	CU	ChristianUnion	1206	@christenunie	2388
	D66	Democrats 66	345	@D66	7004
	DENK	Think	2641	@DenkNL	1128
	FvD	Forum for Democracy	2642	@fvdemocratie	2888
	GL	GreenLeft	756	@groenlinks	3802
	PVV	Party for Freedom	1501	@PVVpers *	56
				@_pvv	7183
	PvdA	Labour Party	742	@PvdA	3542
	PvdD	Party for the Animals	990	@PartijvdDieren	4112
	SGP	Political Reformed Party	1251	@SGPnieuws	738
	SP	Socialist Party	357	@SPnl	3344
	VVD	People's Party for Freedom and Democracy	1409	@VVD *	16080
NOR	DNA	Norwegian Labour Party	104	@Arbeiderpartiet	3237
	Fr	Progress Party	351	@frp_no	835
	H	Conservative Party	1435	@Hoyre	1773
	KrF	Christian Democratic Party	1538	@KrFNorge	822
	MDG	Green Party	2254	@Partiet	4246
	RV	Red Electoral Alliance	1638	@Raudt	859
	SV	Socialist Left Party	81	@SVparti	3556
	Sp	Centre Party	702	@Senterpartiet	1009
	V	Liberal Party of Norway	647	@Venstre	3219
NZL	ACT	ACT New Zealand	617	@actparty	1907
	Greens	Green Party	1171	@NZGreens	3043
	LP	Labour Party	878	@nzlabour	2649
	MP	Maori Party	114	@Maori_Party	789
	NP	National Party	997	@NZNationalParty	2679
	NZFP	New Zealand First Party	891	@nzfirst *	796
	UFNZ	United Future New Zealand	1313	@UnitedFutureNZ	8
PRT	BE	Bloc of the Left	557	@BlocoDeEsquerda *	199
	CDS-PP	Democratic and Social Centre – People's Party	251	@_CDSPP	1670
	CDU	Unified Democratic Coalition	1295	@CDUPCPPEV	730
	CH	Enough!	2789	@PartidoCHEGA	323
	IL	Liberal Initiative	2788	@LiberalPT	1364
	L	Livre	2374	@LIVREpt	727
	PAN	Party for Animals and Nature	1781	@Partido_PAN	2308
	PS	Socialist Party	725	@psocialista	3991
	PSD	Social Democratic Party	1273	@ppdpsd	5148

(Continued on next page...)

Table S.2: Parties included in our dataset. Party abbreviations, names and IDs based on Döring and Manow (2019). Twitter account names link to Twitter pages (using accounts' user IDs). Twitter accounts marked with an asteriks (*) flag accounts we have updated in May and Jule 2023. The last column reports the number tweets recorded. *(continued)*

Country	Party Abbr.	Party Name	Party ID	Twitter account	*N* tweets
SWE	C	Centre Party	1461	@Centerpartiet	3044
	FP	People's Party	892	@liberalerna	2587
	KD	Christian Democrats	282	@kdriks	2951
	M	Moderate Party	657	@moderaterna	3987
	MP	Greens	1154	@miljopartiet	5959
	SAP	Social Democrats	904	@socialdemokrat	2918
	SD	Sweden Democrats	1546	@sdriks	1822
	V	Left Party (Communists)	882	@vansterpartiet	3750

}{*}{

SWE} & V & Left Party (Communists) & 882 & @vansterpartiet & 3750* \end{longtable}

Table S.3: Parties accounts researched and added in May and June 2023. Party abbreviations and IDs based on Döring and Manow (ibid.). Twitter account names link to Twitter pages (using accounts' user IDs). Comment "collected (not added)" means that we have found this account but none of its tweets were posted during the parliamentary gconfigurations for which we record entries for the given party.

Country	Party Abbr.	Party ID	Twitter account	Comment
AUS	PUP	2259	@UnitedAusParty	pooled with tweets from account @PalmerUtdParty
AUT	TS	2150	@TeamStronach	collected (not added)
AUT	FPO	50	@FPOE_TV	merged with tweets from account @HCStracheFP
BEL	FN	171	@DNatbe	collected (not added)
CAN	CCF|NDP	296	@NDP	added
DNK	Sbf	1892	@Sambandsflokkur	added
ESP	AP-P	645	@ppopular	replaces @ppopular_ account
ESP	Bai	385	@NafarroaBai2011	added
ESP	P	2376	@PODEMOS	replaces @Podemos_Unidos account
FRA	PR	401	@PartiRadical	collected (not added)
GRC	KKE	614	@gt_kke	replaces @KKEGreece account
ITA	AN	373	@ANazionaleTw	collected (not added)
NLD	VVD	1409	@VVD	replaces @VVDZwolle account
NLD	PVV	1501	@PVVpers	merged with tweets from @_pvv
NLD	LN	744	@LeefbaarNL	collected (not added)
NZL	NZFP	891	@nzfirst	added
PRT	BE	557	@BlocoDeEsquerda	replaces @EsquerdaNet account

B Coding scheme development and validation

We developed our coding scheme iteratively. On the one hand, this process featured deductive, theory-guided design choices. On the other hand, we made multiple changes to the draft coding scheme based on the evidence we gathered when testing and evaluating the coding decisions (judgments) a given version induced. The below subsections summarize this process, focusing on four key aspects:

1. the key design choices respectively revisions from one version to the other,
2. the motivation behind making these choices,
3. how we validated these choices, and
4. the results of these choices.

B.1 Background: crowd-sourced measurements of populist rhetoric

We already realized many of the constituent elements of our idea to measure parties' rhetorical appeals from their social media communications in a precursor to the present paper: Hua et al. (2018). This paper focused on crowd-sourcing populism measurements and was presented at the Midwest Political Science Association (MPSA) Meeting in March 2018 and at the European Political Science Association (EPSA) Meeting in June 2018.

B.1.1 Coding scheme

The coding scheme used in this precursor paper featured three substantive questions:

1. Does this tweet/post criticize or mention in a negative way the elites?
2. Does this tweet/post mention in a positive way or even praise the people (citizens of the country, the working class, the natives, ...) or the nation?
3. Does this tweet/post criticize minorities or specific groups of people (muslims, jews, LGBT people, poor people, ...)?

For each of these questions, there were only two answer categories: "Yes" and "No." In addition, we asked coders to indicate whether or not they could read and understand the text of the social media post they were currently seeing. When coders negated this filter question, we showed them to the next post. If answering positively, we asked them to answer the three questions above.

The main motivations behind including the three questions listed above in one coding task were conceptual completeness and cost-efficiency. Theory suggests that anti-elitism may come in many forms but a populist ideology necessarily combines anti-elitism with a

conception of the pure people that stands antagonistically against the corrupt elite (e.g., Canovan, 1999; Mudde, 2004). Similarly, populist rhetoric simplifies political problems by creating and highlighting an antagonism between the "pure" people and some culprit to blame for their misfortunes (cf. Bonikowski and Gidron, 2016b). Given these conceptual considerations, we required answers to all three questions to detect populist rhetoric for any given text item. However, devising one coding task for each conceptual subdimension seemed not very cost-efficient. Coders would need to read each post three times as often.

B.1.2 Evaluation

The final coding scheme, too, was developed iteratively with the support of three research assistants (RAs). The judgments and experiences of our RAs helped us to refine the coding instrument. Finally, we deployed the coding instrument on *CrowdFlower*.[1]

The results obtained with the populism coding scheme described above were very encouraging. However, we realized that the elite criticism question was likely insufficient to distinguish genuine elite criticism from other valance attacks, such as incumbent critique or negative campaigning targeted at selected political opponents (cf. Jung and Tavits, 2021; Nai, 2018; Walter, 2014). We found this was important to enable comparative research on anti-elite rhetoric, which is used by both populist and non-populist parties (cf. de Vries and Hobolt, 2020) Hence we wanted to add a component to our coding scheme that would allow taping into whether or not the criticism voiced in a social media post is generalizing and relatively broad in scope or whether it is rather limited and scope and focuses on a specific elite.

The following two tweets taken from our corpus exemplify this distinction between "general" and "specific" elite criticism. On March 15, 2019, the official Twitter account of the *Green Party of Canada* posted:[2]

> To fight climate change, you need to change the system.
> And you can't do that with the parties who have built and maintained the old system. It has worked just fine for them. Now we need what's best for the rest of us.
> #StrikeForClimate #ClimateStrike #Strike4Climate

In this tweet, the Canadian Greens criticize a lack of action against climate change and attribute blame to 'old system' parties. This is a prototypical generalizing elite criticism. Contrast this with the following tweet by the official Twitter account of the Swedish center-right *Moderaterna* posted on October 10, 2016.[3]

[1] Later *Figure Eight*, currently *Appen*: https://appen.com/
[2] see https://twitter.com/28370071/status/1106644168632238080
[3] English translation; for the original see https://twitter.com/19226961/status/785401301144403968

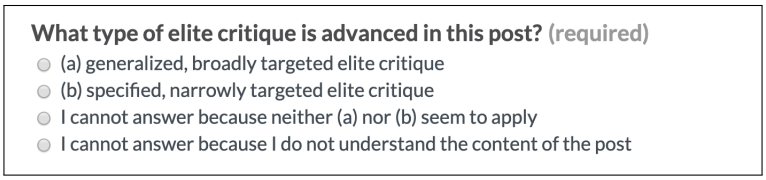
> Stefan Löfven closes his eyes to societal problems and lacks answers to break exclusion. @KinbergBatra [URL] #svpol

This criticism targets a single political elite, the then-acting Swedish prime minister. While we would agree that this tweet is a prototypical instance of a valence attack, we argue that it is *not* an instance of anti-elite rhetoric because it lacks the attribute of generalization.

B.2 First version: Discriminating between elite criticism and other valence attacks

In a first revision to the original three-fold coding scheme, we thus made two changes:

1. We focused the coding scheme on the elite criticism measurement.
2. We added questions that would allow tapping into the distinction between (a) criticism that targets a broader set of elites and (b) criticism that narrowly targets individual members of the elite or a specific elite group or organization.



(a) Elite criticism question

(b) Follow-up/clarification question.

Figure S.1: Questions in the first version of our elite criticism coding scheme. Note that the second question was displayed only if a coder selected the third answer category for the first question.

B.2.1 Coding scheme

Figure S.1 shows the resulting coding scheme. Its first two answer categories to the first question ("What type of elite critique is advanced in this post?") serve to discriminate between broadly and narrowly targeted elite criticism.[4] In a designated section of the coding instructions, we elaborate accordingly:

[4] We have later corrected the wording to say "criticism" instead of "critique."

There are generally differences in how one can criticize the elite. On the one hand, one can be specific and single out an individual elite or elite group. Take for example the following post:

> *... The PP is equally corrupt in Spain than in Galicia. There is no possible democratic regeneration with the corrupt party in government. ...*

It singles out the PP (a major political party in Spain) and alleges it of being corrupt. In selecting the PP as its only target, this critique is both specific and narrow. Posts like this one are therefore examples of specific, narrow-targeted elite critique.

On the other hand, a critique can criticize a very broadly conceived elite in a very generalizing and encompassing way. Take for example the following post:

> *The established power cartel consisting of ÖVP and SPÖ has driven the country to a standstill due to their endless discussions and complete inability to reach consensus.*

This post alleges two mainstream Austrian parties to form a cartel (implying collusion) and accuses it of generally hindering the country's progress. Critiques like this one are examples of broadly targeted, generalizing elite critique.

We then instructed coders to 'try to make [their] judgment on *what type of elite critique* [a post] advances and select either of the two corresponding answer categories.'

B.2.2 Evaluation

We have sampled 250 tweets from the original crowd-coded social media posts we collected with the populism coding scheme to evaluate this version of the coding scheme. We have then distributed these via *CrowdFlower* to three RAs for annotation with the new coding scheme. In addition, one member of our author team (Hauke) separately annotated the same set of posts. Another author (Tarik) then reviewed posts for which there was much disagreement between RAs and Haukes's judgments.

With this data at hand, the two authors (Hauke and Tarik) engaged in a qualitative review of the codings and disagreements. The goal of this process was to understand where the present version of the coding scheme was falling short to induce correct coding decisions. One thing that stood out was that focusing on the targeting aspect was biasing coding decisions because the target of an elite criticism was often left implicit. In such

cases, our RAs could not provide the correct coding because there was no appropriate answer category.

B.3 Second version: Increasing conceptual depth

This insight motivated a design choice in our second revision: we included a component that allowed coders to indicate when they thought that a tweet expressed elite criticism, but the specific target was left implicit.

Because this further increased the complexity of our coding task, a second decision in the second revision was to separate the coding process into two steps: We first collected codings to identify social media posts that likely featured elite criticism according to crowd coders' aggregate judgment (task 1). We then distributed this subset of social media posts for more granular annotation according to the explicit–implicit (targeting) and general–specific (scope) distinctions (tasks 2.1 and 2.2).

B.3.1 Coding scheme

Specifically, we arrived at the following coding scheme:

1. Does this post contain or approve of an implicit or explicit criticism of the elite? (yes, no, cannot answer)

2.1 Is it made explicit what actor is being criticized? (yes, no, cannot answer)

2.2 Is the criticism itself rather general or rather specific? (specific, general, cannot answer)

B.3.2 Evaluation

We evaluated this coding scheme (including its two-step annotation strategy) via AWS *SageMaker Ground Truth*. We sampled 582 tweets from the data set already used to crowd-source measurements with our original populism coding scheme for the first task. We distributed these tweets to both crowd workers and three RAs. We have collected three judgments per post from our RAs and five per post from the crowd.[5]

We aggregated these judgments at the tweet-level for each workforce using a Dawid-Skene annotation model (Dawid and Skene, 1979). Next, we distributed the tweets with positive induced labels again for coding to collect judgments for tasks 2.1 and 2.2.

Analyzing the data gathered in this evaluation showed that crowd coders were more likely to disagree on the coding of a post and that they had lower (estimated) true-positive detection abilities than our RAs. Because of substantial within-workforce levels of disagreement at the post level, there was at best moderate agreement between labels

[5] However, we could collect judgments from only two RAs for 226 out of 382 cases due to a backend bug.

induced from RA respectively crowd workers' codings. These findings raised the question of whether aggregating crowd workers' judgments would yield labels of inferior quality than when aggregating trained RAs' judgments on either of the three tasks.

We conducted an expert review of the coding decision made by both workforces to evaluate this question. The sample of posts annotated by both workforces comprised 582 tweets. We sampled 250 tweets from this set for review. There were 132 tweets in the review sample for which labels induced from our two workforces' judgments disagreed. A total of four experts (the entire author team) participated in the review. We randomly assigned our four experts into two teams and let each team review a sample of 150 tweets with an overlap of 50 tweets between teams' samples.

The results of this expert review showed that in the case of tweets for which labels induced from workforces' judgments disagreed (disagreement cases), there was overall more agreement between RA and expert-judgment induced labels than between crowd and expert-judgment induced labels. However, for task 1, revisions of either workforces' aggregate judgments occurred only for "Yes" labels and at similar rates across workforces. Thus, while our experts disagreed with the labels induced from either workforces' judgments on the first task, labels obtained from crowd workers' judgments were no more likely to be judged incorrect than those obtained from RAs' judgments. We took this as suggestive evidence that reducing the complexity of task 1 and separating it from tasks 2.1 and 2.2 was a good design choice.

In contrast, revisions of disagreement cases in the set of labels induced from crowd workers and RAs' judgments on tasks 2.1 and 2.1 occurred across answer categories. This provided evidence that making coding decisions for these tasks was challenging for coders of either workforce. Importantly, task 2.1—distinguishing between implicit and explicit criticism—was a task that also challenged our experts: Experts' estimated true-detection abilities for the valid answer categories of this task (i.e., all but "Cannot answer") were lower than on either of the two other tasks. Taken together, we took this as evidence that we should rephrase our task or omit the explicit–implicit coding task altogether.

The results from the review were more encouraging for cases for which the labels induced from crowd workers respectively RAs' judgments agreed: Not only agreed individual experts very frequently with induced labels. The labels induced from experts' judgments did so, too. There were thus many cases in which making the correct coding decision was relatively easy for all workforces. Another, smaller set of tweets seemed to provide much more ambiguous signals, however.

One thing that stood out when reviewing tweets for which labels agreed, however, was that among tweets labeled "No" on task 1, there were relatively many expressing no political point of view or electoral appeal whatsoever. Many tweets in this category, for example, were tweets that posted or retweeted links to events or news articles without commenting on them. This was already brought to our attention by our RAs in earlier

stages of the coding scheme development process.[6] In the codings review process, all members of our author team had seen many such examples, too.

Discussing this issue among authors led to two conclusions: First, removing such "spam" tweets before distributing tweets for (crowd) coding would make completing the task easier for coders and collecting annotations more cost-efficient. Second, such "non-political" tweets should not be in the denominator at all when estimating the prevalence of anti-elite rhetoric in parties' social media communications because, conceptually speaking, only political tweets may contain anti-elite rhetoric.

In sum, we concluded from these evaluations that making correct decisions for tasks 2.1 and 2.2 was more demanding than for task 1. The explicit–implicit criticism distinction, in particular, was too difficult to enable aggregating judgments into high-quality labels. Moreover, we decided to remove non-political posts ("spam") from the data before collecting elite criticism codings.

B.4 Third version: Reducing the cognitive load to facilitate reliable coding

The conclusions we drew from evaluating the second revision round motivated our design choices in the third and final round of coding scheme revisions:

1. easing coders' focus on the elite criticism task by pre-filtering non-political posts,
2. omitting coding of the implicit–explicit targeting distinction, and
3. condensing the remaining two tasks into one task.

The last design choice, in particular, was motivated by our goal to reduce the complexity of the preceding version but maintaining a coding scheme that would allow us to discriminate between specific and general elite criticisms.

B.4.1 Coding scheme

The two questions we asked coders to answer in the new coding scheme were:

> Does this statement contain or approve of a criticism of the elite?
> And, if yes, who is the target of this criticism? A specific elite actor *or* an elite group more generally?

The first part of the question corresponds to task 1 of the preceding version of the coding scheme. The second part ('if yes, . . . ') corresponds to task 2.2, and is designed to tap into the general–specific distinction outlined above.

Given these questions, we provided the following answer categories:

[6] In fact, this motivated including the filter question in the original populism coding scheme.

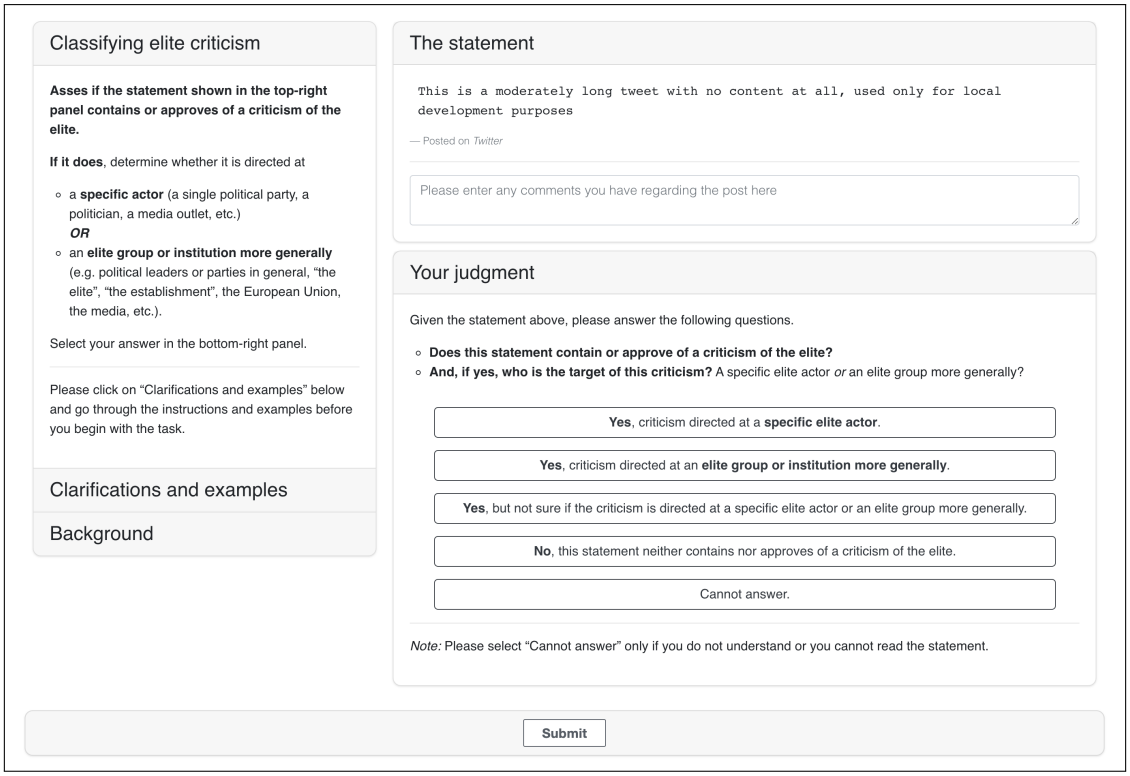


Figure S.2: User interface of the third version of our elite criticism coding scheme. The left panel provides a description of the coding task. The top-right panel displays tweets' text on at a time and allows coders to leave a comment. The bottom-right panel prompts coders to indicate their coding decisions.

1. Yes, criticism directed at a specific elite actor.
2. Yes, criticism directed at an elite group or institution more generally.
3. Yes, but not sure if the criticism is directed at a specific elite actor or an elite group more generally.
4. No, this statement neither contains nor approves of a criticism of the elite.
5. Cannot answer.

Figure S.2 shows a screenshot of the coding scheme. Note that we cannot provide the link to the online coding App here because it contains the authors' contact information and thus would reveal their identities to reviewers.

Neither the questions nor the answer categories explicitly offer the option to indicate that the target of a supposedly elite-critical message is left implicit. However, answer category 3 offers coders the possibility to state their uncertainty regarding the target. Such uncertainty should be, among other things, a function of whether or not the target of an elite criticism is made explicit. This design choice thus allowed us to obtain a coding scheme that is relatively concise compared to the second version). At the same time, it mitigates design-induced bias in coding decisions in case of implicit elite criticism—in contrast to the first version.

B.4.2 Evaluation

To evaluate our new coding scheme, we drew a balanced sample of 292 tweets from the set of tweets for which we had previously collected crowd, RA, and expert reviewer judgments.[7] Analyzing these data, we found that the new codings scheme enabled coders to make better judgments. The induced labels were of better quality than those obtained with the previous version of the coding scheme. Specifically, when removing judgments of low-quality coders and accounting for prior expectations of label class prevalence, annotation model-based labels were of generally high quality:

- they strongly correspond to labels induced when pooling crowd workers, RAs, and our experts' judgments on task 1 of the second version of the coding scheme;
- they strongly correspond to labels induced only from experts' judgments on this task; and
- a qualitative review of a sample of labeled tweets suggests few misclassifications.

These findings reassured that the third version of our coding scheme yielded crowd codings of sufficient quality to induce reliable labels.

[7] Specifically, we pooled task-1 judgments and fitted a Dawid-Skene annotation model. 146 tweets were assigned to the positive label class (elite criticism), and we sampled an equal amount of tweets with "negative" labels.

C Generating our labeled dataset

C.1 Sampling of tweets for the first round of crowd coding

C.1.1 Sampling eligibility criteria

Before drawing the first sample for crowd coding, we have subset our Twitter data to tweets featuring political messages for which there was an English translation or original text available from a prior project phase. Table S.4 cross-tabulates these two criteria and shows that a total of 144702 tweets were eligible for sampling into the first sample distributed for elite criticism crowd coding.

We selected these two sampling eligibility criteria for both methodological and practical reasons: We expected that tweets that contain elite criticism would likely be considered political by crowd coders. Because we also expected that elite criticism would be pretty rare in our data, we wanted to pre-filter tweets in a way that would eventually increase the number of positive (i.e., elite criticism) samples in our training data. Sampling from likely "political" tweets served this goal. And because we rely on MTurk for crowd coding, we needed to machine-translate sampled tweets to English. Given that we had already done so for a pretty balanced sample of tweets in an initial project phase, we did not want to generate extra costs but just went with sampling from this subset of the data.

To identify tweets that likely feature political messages, we have used a pre-trained ensemble classifier to predict tweets probabilities of belonging to the "political" class. This classifier has been trained by Licht (2020) on a sample of more than 8000 crowd-coded tweets drawn from the same tweets corpus as used in our study.. The coding scheme used in to obtain these annotations tasked crowd workers to classify tweets as "political" if a tweet meets at least one of the following criteria: (i) it states, quotes or comments on a political stance or point of view; (ii) it discusses politics or a concrete policy issue; and/or (iii) it contains a praise or criticism of elite actors or organizations.

Table S.4: Tweets in our data tabulated by the availability of an English-text version and whether their content was classified to be "political."

	available in English		
Political	no	yes	*Prop.*
yes	255645	144702	0.692
no	107010	71457	0.308
Prop.	0.627	0.373	

C.1.2 Semantic similarity-based sample stratification

As a second step, we stratified the tweets eligible for sampling according to their content. While we could have drawn a random sample, this would have ignored the information about tweets' semantic similarities. We thus adopt a stratified sampling strategy that leverages the information contained in tweet texts' multilingual embedding representations. Specifically, we have

1. taken the subset of tweets classified as political for which an English text version is available (samples in the upper-right cell of the above table),
2. used independent component representations of tweets' LASER embeddings (Artetxe and Schwenk 2019; cf. Licht 2023) to obtain 500 clusters using the k-means algorithm, and
3. obtained a sample of n tweets stratified by clusters and cluster characteristics (size and party account diversity).

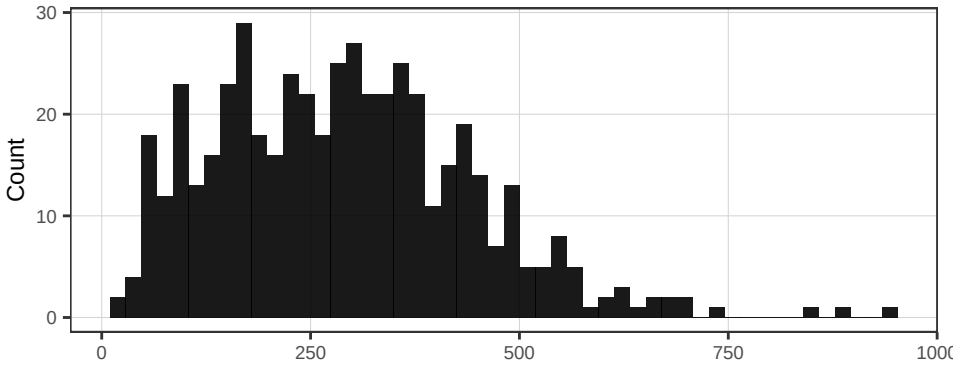


Figure S.3: Distribution of cluster sizes of 500 k-means clusters. Clusters obtained from 300 independent components of tweet LASER embedding representations.

For clustering, we have applied the k-means algorithm to the subset of tweets eligible for sampling. Specifically, we have obtained 500 clusters in iterations.

Figure S.3 reports the distribution of cluster sizes. Most clusters comprise many tweets.

Note that the clustering is hardly reducible to a lower-dimensional representation. Obtaining principal components (PCs) from clusters' LASER embedding centroids suggests, we find that it requires NA (NA) PCs to explain more than 60% (90%) of the differences in cluster centroids.

Moreover, Figure S.4 shows that while some clusters group in local neighborhoods, they overwhelmingly spread over the semantic space.[8] The fact that only a few clusters

[8] This figure depicts the two-dimensional representation of clusters' centroids in the LASER embedding space after applying the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm (Maaten

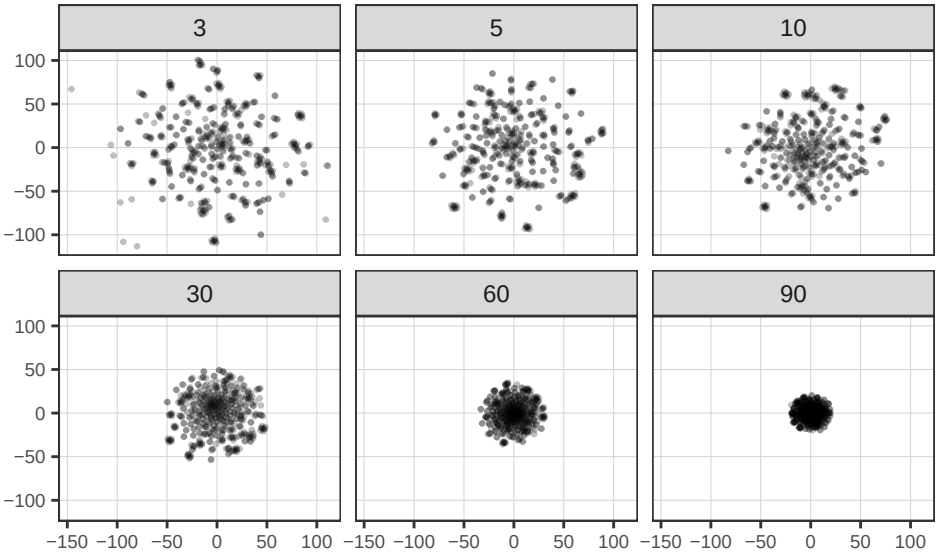


Figure S.4: Two-dimensional t-SNE representation of cluster centroids.Facets report results at different perplexity values.

group in local neighborhoods and the vast majority of clusters distribute evenly over the plane suggests that there is not much global structure in the higher-dimensional representation of the data.[9] What is more, unordered (high-entropy) data has high information value, independent of its dimensionality. The induced clustering thus yields a highly informative organization of tweets (multilingual) semantic similarities.

Figure S.5 plots the numbers of distinct party and country IDs contained in clusters. It shows that most clusters also comprise tweets authored by different party accounts from different countries. Clusters are thus also diverse in terms of tweets authors' countries and parties. This suggests that the semantic similarity-based clustering is not simply picking up cross-party or cross-country differences in tweets' contents.

C.1.3 Drawing a stratified sample of tweets

Provided with a clustering that reflects tweets' semantic, we sample tweets from each cluster while accounting for differences in their sizes and party diversity. Specifically, we want to sample tweets from all clusters but draw more tweets from larger clusters and

and Hinton, 2008) at different degrees of perplexity. t-SNE finds an "optimal" lower-dimensional representation of the data by minimizing the sum of the difference in conditional probabilities between data points' locations in the high- and low-dimensional distance/similarity-based graph representation of the data. *Perplexity* is the parameter that governs how many neighbors each data point ought to have. It thus determines how to balance the algorithm's attention between local and global aspects of your data. Lower values favor preserving local graph structure; higher values favor representation of global graph structure.

[9] Note that the relative "distance" of points *within* grouped neighborhoods (the size of the bounding box delineating local neighborhoods) does not represent distance in terms of similarity. Moreover, the distances *between* neighborhoods of clusters can only be interpreted comparatively across different values of perplexity.

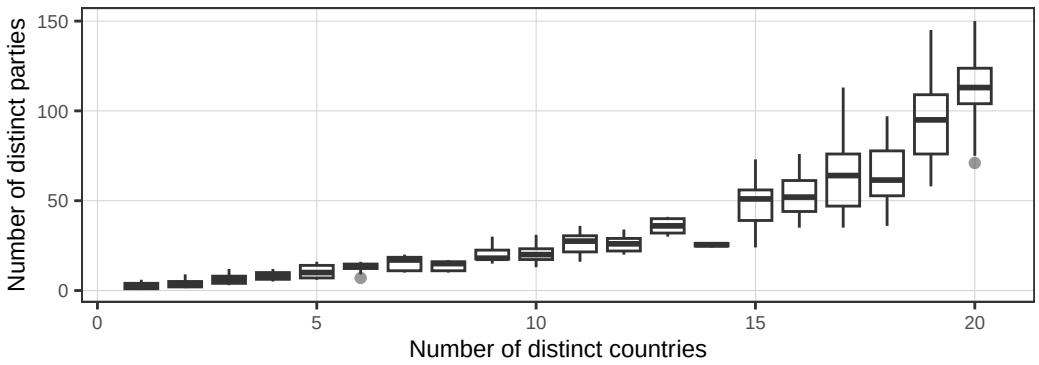


Figure S.5: Cluster composition in terms of country and party diversity.

fewer for low diversity clusters.

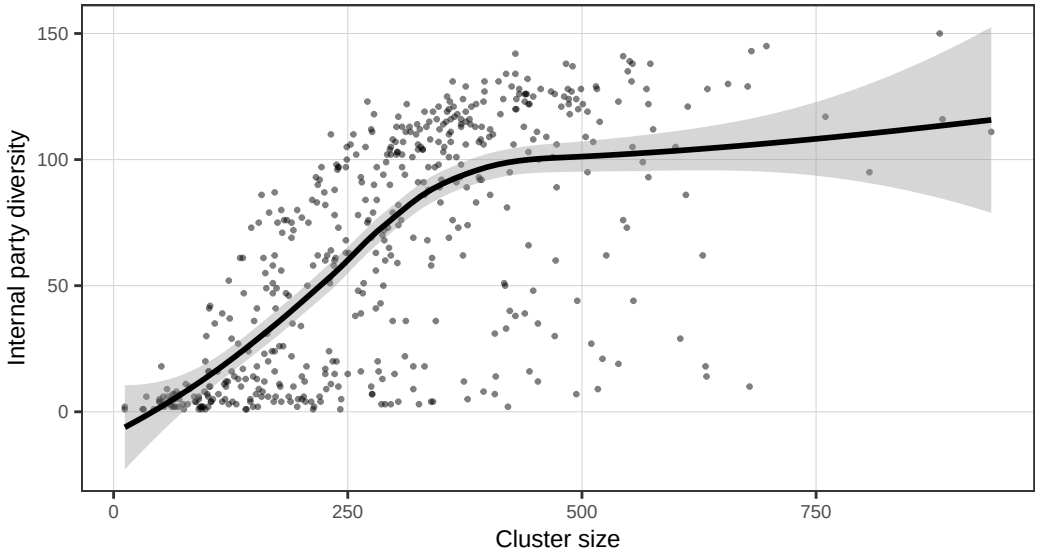


Figure S.6: Relation between clusters' sizes and their internal diversity in terms of party composition.

Inspecting the relationship between the number of party accounts represented in the tweets of a cluster and cluster size (Figure S.6), we see that clusters' party diversity first increases with cluster size and then levels off. At the same time, there are only a few large but low-diversity clusters.

We thus decided to use our measure of clusters' internal party diversity as a heuristic stratification criterion, and sampled fewer tweets from lower-diversity (and likely smaller-sized) clusters. Specifically, we have taken the ceiling of the $\log_2$ of party account counts and added one to obtain the number of tweets to be sampled from each cluster. The resulting distributions of cluster sample sizes at these sample sizes are reported in Figure S.7 and allow to confirm that they reflect cluster size in terms of tweets contained.

Given this rule-based sample size criterion, we have first drawn one tweet per party from each cluster. From these samples of party tweets, we have then drawn the designated

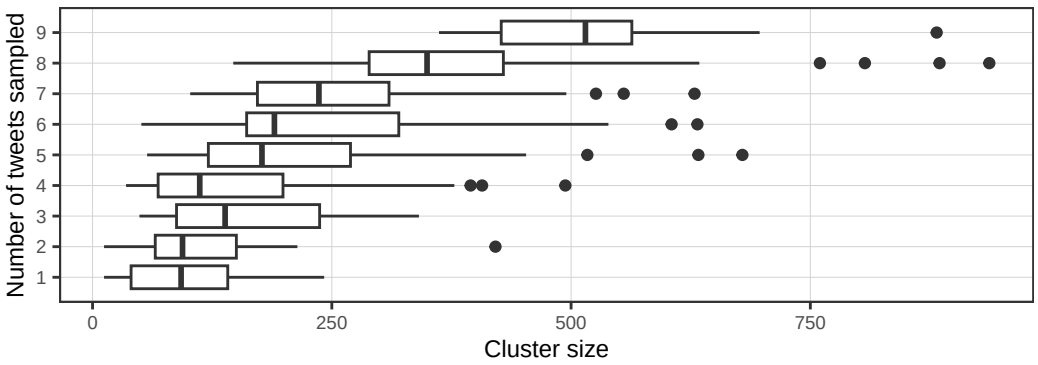


Figure S.7: Number of tweets sampled from cluster (vertical axis) against cluster size in terms of number of tweets (horizontal) axis.

number of tweets. This sampling strategy ensured that we sampled similar tweets from different parties for crowd coding in the first round.

C.2 Results of the first round of crowd-sourced elite criticism coding

We have distributed the 3270 tweets in the first round sample for crowd coding through AWS *SageMaker Ground Truth*, using a custom coding scheme.[10] Specifically, we collected six judgments for each tweet. In total, crowd workers contributed a total judgments. Table S.5 reports the frequencies of coders absolute label choices (judgments). Adding subcategories, % of judgments indicate some form of elite criticism.

Table S.5: Total judgment frequencies in first round of crowd coding.

Judgment	*N*	Proportion
No elite criticism ("No")	12803	0.653
General elite criticism ("General")	3131	0.160
Specific elite criticism ("Specific")	2036	0.104
Ambiguous elite criticism ("Unsure")	1338	0.068
Cannot answer ("Cannot answer")	312	0.016

C.2.1 Coder statistics

A total of crowd workers contributed judgments. However, as shown in Figure S.8, the average number of judgments varies across coders.

The high number of coders who only provided a few judgments is typical for crowd-sourcing tasks with no (unpaid) screening tasks. Not needing to pass screening tasks allows crowd workers to "shop around," dropping out as quickly as they dropped in.

[10] We cannot provide the link to the online coding App here because it contains the authors' contact information and thus would reveal their identities to reviewers.

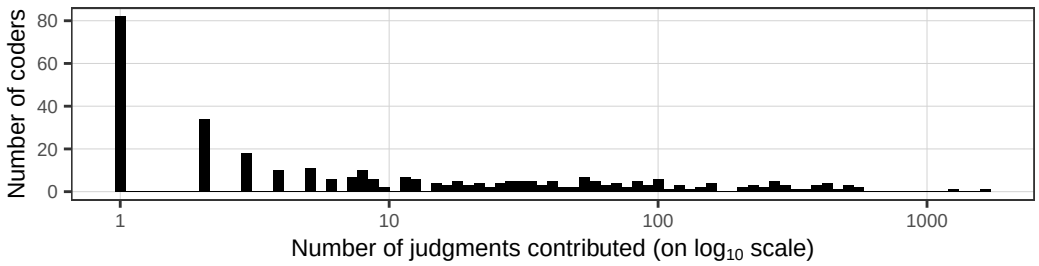


Figure S.8: Distribution of judgments contributed per coder in first round of crowd coding.

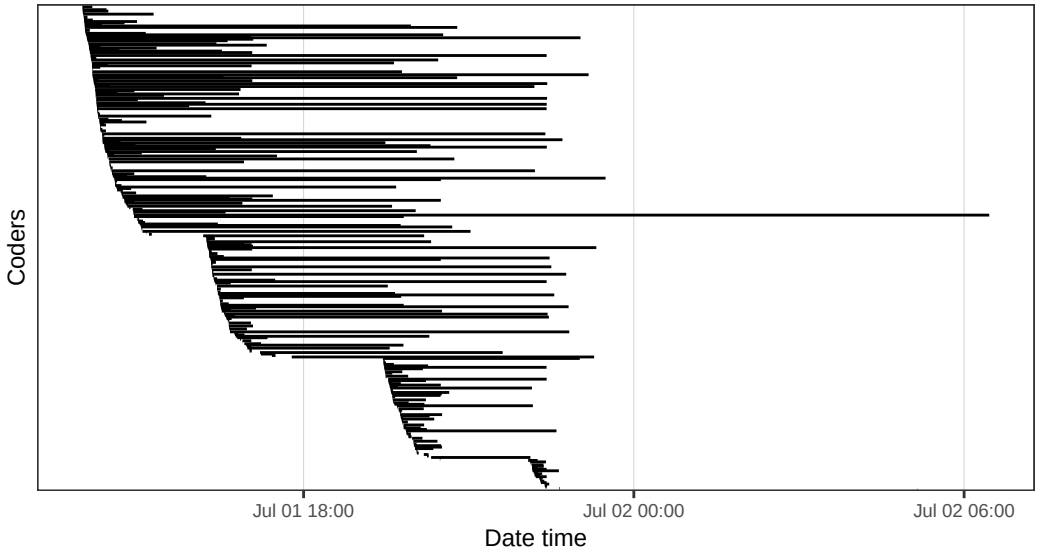


Figure S.9: Coder activity times and duration in first round of crowd coding. Each horizontal line represents the activity period and duration of a single coder.

Looking at the times during which individual coders contributed to the task and their time spent (i.e., the time passed between a coder's first and last judgment) supports this assessment. In Figure S.9 we see more clearly that only few coders spent more than 30 minutes or an hour on the task (NaN% and NaN%, respectively). Coders that contributed few judgments and/or only for a short time were likely on a "shopping tour" and may thus provide random or even adversarial judgments.

Note that the inter-quartile range of coders' median per-judgment duration did not contain the expected 10 seconds determining coders per-judgment pay. Instead, the median of median annotation durations is 22.654 seconds. We thus doubled the per-task pay in the second round of crowd coding.

In addition to these coder summary statistics, it is worth looking at the coder-level variability in the judgments they contributed. The intuition of this analysis is simple. We may expect a very low variability in the labels assigned by coders that provided only a few judgments (because it is not unlikely that the first few tweets a coder encounters all have the same "true" label). However, we want to see higher label variability as the

number of judgments a coder contributes increases. Figure S.10 assesses for which coders these expectations hold. It measures "label variability" by coder-level label set entropy (i.e., the entropy in labels in the set of all judgments provided by a coder).

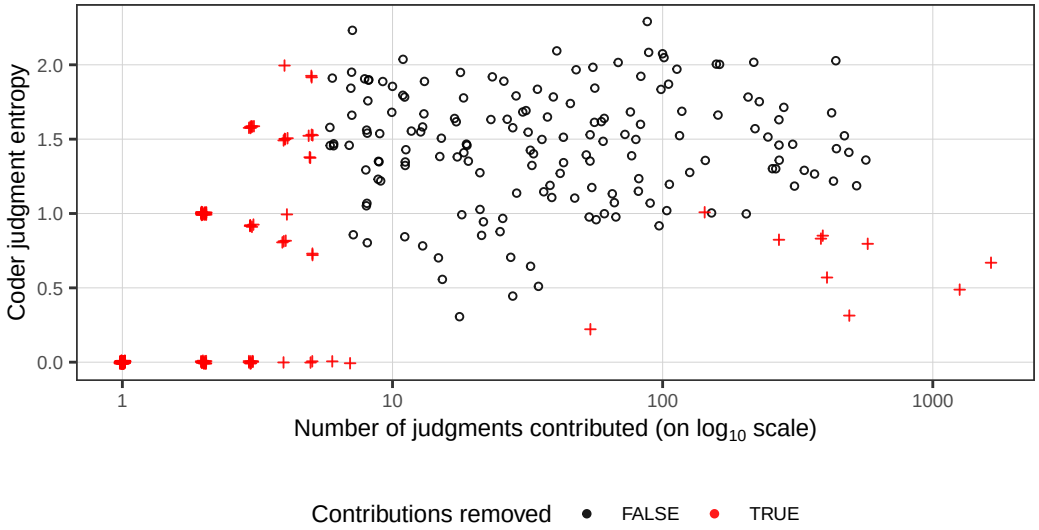


Figure S.10: Coders' numbers of judgments against judgment entropies in first round of crowd coding. Vertical and horizontal jitter of max. 1% added to avoid over-plotting.

Besides coders with low total numbers of judgments provided, there are about a dozen coders (located in the bottom-right plot area) whose label variability is comparatively low for the high numbers of judgments they contributed. We have manually inspected the contributions of coders who (a) contributed more than 50 judgments, but label set variabilities of less than 0.5, or (b) contributed more than 100 judgments, but label variabilities of less than 1.

This assessment confirmed our suspicion. Although these coders' judgments exhibit some level of label variability, it is relatively low compared to the number of judgments they have provided. Such behavior is indicative of low-quality contributions.

When inducing tweet-level labels from crowd codings (see below), we have thus removed the judgments of coders with (i) less than 10 contributions, (ii) more than 50 judgments and label set variability $< .5$, or (iii) more than 100 judgments and label set variability < 1. The coders affected by either of these removal criteria are colored in red in Figure S.10.

C.2.2 Tweet level statistics

Turning to descriptives at the tweet level, we further analyzed tweet-level judgment set variability, the diversity in the set of labels assigned to each tweet. Ideally, we want *low* degrees variability in label sets at the tweet level. When tweets are labeled consistently,

there is much agreement in coders' judgments of their "true" label class. Note that this contrasts with our assessment of coder-level label variability, where low variability may indicate spam coders if the number of judgments provided by the coder is high.

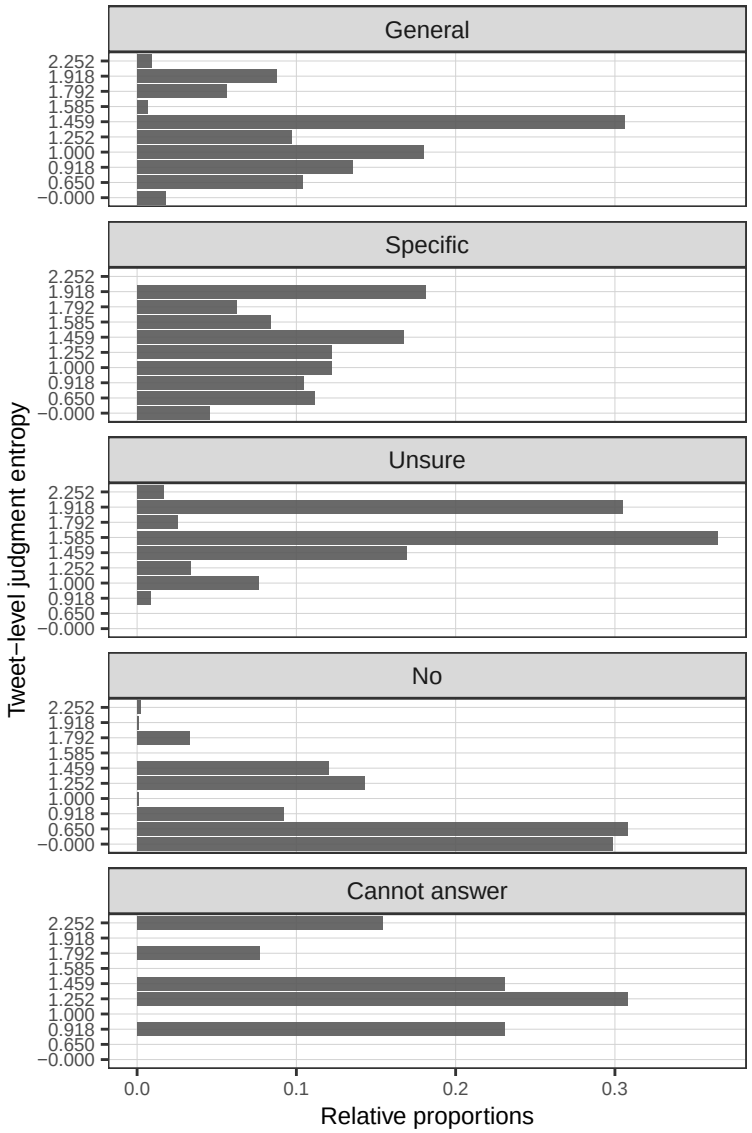


Figure S.11: Tweet-level variability in labels assigned (entropy) in the first round of crowd coding. Panel columns group tweets by labels that were most frequently assigned to them.

Figure S.11 plots how tweet-level label variability (again measured by entropy) varies across label classes. There are a few things that stand out from this figure. First, tweets for which "No" is the plurality judgment outnumber those for which other labels were assigned most frequently. Second, "Cannot answer" is very rarely the most frequent judgment. Inducing labelings by plurality voting would thus result in a very low proportion of "Cannot answer"-type tweets. Third, the number of tweets characterized by relatively inconsistent judgments is generally relatively low. This finding is strongly driven by the large number of tweets for which the plurality judgment is "No", however. Fourth, within

groups determined by the plurality label (plot panel rows), judgment variability is highest in the "Unsure" category, followed by the "Cannot answer" and "General" categories.

C.3 Judgment aggregation: Inducing tweet-level labels from first-round crowd codings

Because of the indications of variable annotation quality reported above, we have fitted a statistical annotation model to the crowd-sourced codings using an Empirical Bayes implementation of the traditional Dawid–Skene per-annotator model (Dawid and Skene, 1979). This model-based annotations aggregation method estimates tweet-level label class membership probabilities and coders' annotation abilities from the data. Thus, instead of taking coders' judgments at face value, this model-based approach takes coders' varying label class-specific detection abilities into account when estimating tweets class membership probabilities.

Note that before fitting this model, we have removed (i) coders indicated in red in S.10, (ii) all judgments made in less than four seconds, (iii) tweets for which in the remaining judgments "Cannot answer" was the plurality judgment, as well as (iv) all other remaining "Cannot answer" judgments. We argue concerning the latter two points that taking the wording of this answer category in our coding scheme literally, we should have little trust in coders who are 'unsure' or 'cannot answer' because they do not understand or cannot read the tweet text. If coders chose their answers according to their lack of understanding or their uncertainty about what would be the correct answer, removing these judgments is similar to remove "don't know" responses when summarizing responses to a survey.

After removing judgments meeting these criteria, the number of judgments per tweet is still high in many tweets. We have four or more judgments per tweet for 2521 tweets, three for 618 tweets, and less for only 131 tweets. Given that by fitting a per-annotator model, we borrow information from across coders' judgments to induce tweet-level labelings, these few cases with few(er) judgments will not be classified based on only these few judgments (in contrast to the plurality winner method).

Table S.6 reports model-based label prevalence estimates along with the proportions of labels induced by the annotation model respectively plurality voting. The largest difference arises in the proportions of tweets that the "Unsure" label.

Table S.6: Model-based label prevalence estimates and label proportions due to model-based aggregation respectively plurality voting (PV) in first sample of crowd codings.

		Label proportions	
Label	Est. prevalence	model	PV
"General"	0.178	0.176	0.169
"Specific"	0.109	0.105	0.089
"Unsure"	0.093	0.079	0.037
"No"	0.621	0.640	0.705

Table S.7: Comparing model-based and plurality winner labelings in the first sample of crowd codings.

	Plurality winner label			
Model-based labeling	"General"	"Specific"	"Unsure"	"No"
"General"	411	19	19	124
"Specific"	32	239	10	59
"Unsure"	61	8	70	119
"No"	46	23	21	1989

Table S.7 cross-tabulates the labelings induced by both judgment aggregation methods. With 83.354% the overall labeling agreement between aggregation methods is rather modest. This in turn indicates that it makes a real difference what aggregation method is used in our data. The label class-specific agreement of model-induced labelings with plurality voting (column-wise analysis) is 411:139 (74.727%) for the "General", 239:50 (82.699%) for the "Specific", 70:50 (58.333%) for the "Unsure", and 1989:302 (86.818%) for the "No" label class.

Figure S.12 and Table S.8 summarize coders' estimated label detection abilities. These data can be interpreted as follows. We have four label classes. Judgments of coders with an estimated ability above 0.25 are better than chance (because the odds of erring when guessing are 3:1). As ability estimates approach 1, judgment becomes more reliable. Judgments of coders with estimated ability *below* 0.25 can be considered adversarial—the more so, the closer their estimate is to 0. Figure S.12 and Table S.8 thus provide clear evidence that for all label classes but the "Unsure", coders' are are overwhelmingly non-adversarial and in many cases quite reliable.

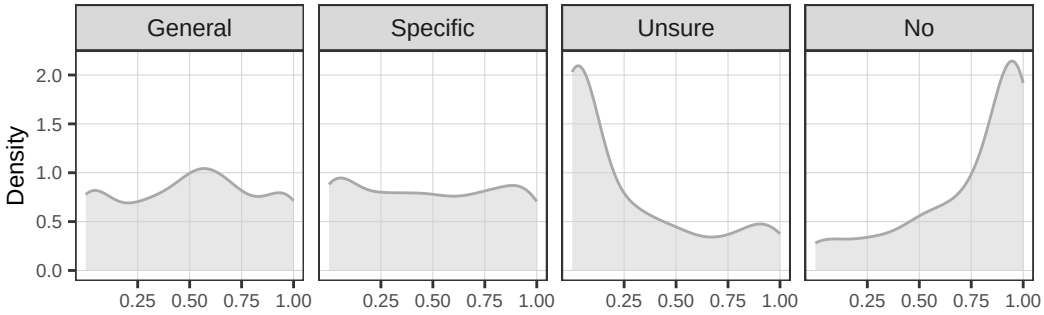


Figure S.12: Estimates of coders' true label detection abilities in the first sample of crowd codings.

Table S.8: Summary statistics of coders' true label detection abilities in the first sample of crowd codings.

	Moments			Quantiles				
Label class	Mean	Std. Dev.	Skewness	10%	25%	Median	75%	90%
"General"	0.498	0.334	-0.062	0.009	0.229	0.524	0.757	0.995
"Specific"	0.477	0.347	0.039	0.009	0.172	0.485	0.779	0.991
"Unsure"	0.265	0.330	1.032	0.003	0.007	0.070	0.479	0.985
"No"	0.735	0.298	-1.124	0.250	0.568	0.869	0.969	0.999

C.4 Learning to detect elite criticism tweets based on the first sample

Table S.9: Numbers of tweets by label class.

Label	Code	*N*	Cum. sum
"General"	yes-general	528	528
"Specific"	yes-specific	360	888
"Unsure"	yes-unsure	286	1174
"No"	no	1962	3136
"Cannot answer"	cannot-answer	`NaN`	`NaN`

The ultimate goal of the application presented in our paper is to train a supervised classifier of elite criticism with high recall and high precision. However, the class of primary interest, *general* elite criticism, is strongly underrepresented in the labelings induced from coder judgments collected in the first round of crowd codings (see Table S.9). Given low absolute counts, learning to classify minority label classes would prove challenging.

Although already expected on theoretical and empirical grounds, the class imbalance in our first sample was thus what motivated us to adopt an iterative strategy for the

construction of our training data. Our two-step strategic sampling strategy was thus aimed at alleviating the severe class imbalance in the first sample.

As an intermediate step, we decided to use the initial sample of labeled tweets to select additional unlabeled tweets for crowd coding. Specifically, we decided to over-sample tweets that are likely members of the three positive label classes ("General," "Specific," and "Unsure") in the second round of crowd annotations.

To allow for such selective sampling, we needed to know which of the unlabeled tweets were likely featuring some type of elite criticism. We have thus trained a GLM-Net classifier on the set of labeled tweets in our first sample. AS expected, it proved hard to obtain a good classifier with the first-round sample because of its strong class imbalance.

C.4.1 Training details

The task of the supervised learning classifier is to classify tweets into the four label classes present in our first sample:

- *positive classes*: "General", "Specific", and "Unsure" elite criticism
- *negative class*: "No" elite criticism

We used 300 independent components (ICs) obtained from tweets' LASER embeddings and, in addition, several other language-independent features created from tweets' metadata:

1. *Tweet metadata features*:
    - tweet "type" (indicator if the tweet is a direct tweet, reply or quote; no retweets included in the dataset),
    - user reactions counts (favorites, retweets, quotes and replies), and
    - an boolean indicator assuming the value 1 if there was media attached to the tweet and 0 otherwise.
2. *Tweet text-based, semantic-independent features*
    - (a) counts and (b) boolean indicators for mentions, hashtags, symbols, URLs, Emojis and quotes found in the tweet text;
    - tweet text character counts in- and excluding URLs, mentions, and/or hashtags (full character width, character width without URLs, and display text width according to Twitter API);
    - cumulated character widths of mentions, hashtags, Emojis and quotes, respectively;

- mentions, hashtags, Emojis, and quotes character widths relative to full character width, character width excluding URLs and display text width, respectively;
- character type-specific counts for alpha-numeric, punctuation, spacing and Unicode characters in full tweet text, and for alpha-numeric and punctuation characters in text without URLS; and
- character type ratios relative to full character width, character width excluding URLs, and display text width, respectively

Using metadata-based and semantic-independent features was inspired by Khandelwal et al. (2017).

Hyper-parameter tuning Because the GLM-Net is a regularized linear model, we adopted a 10-times repeated 10-fold cross-validation (CV) strategy to select optimal hyper-parameters on a random 80% training data split of labeled tweets. Table S.10 reports the distribution of label classes in the training and validation data split. As we have not applied any specific constraints when sampling into training and validation data splits, the proportions of labels are about equal to the distribution in the complete data.

Table S.10: Numbers of tweets by label class in the training data.

Label	N_{train}	N_{val}
"General"	410	118
"Specific"	285	75
"Unsure"	222	64
"No"	1587	375
"Cannot answer"	`NaN`	`NaN`

We have constructed the tuning grid to range over 5 values of α (the L1–L2 regularization mixture parameter), at each of which we have computed maximum-regularization values λ_{max} to obtain 5 equidistant values of λ at each value of α.

We have used the cross-class mean balanced accuracy (BA) as an evaluation metric to select the best-performing hyper-parameter values. The BA metric is the weighted mean of a classifier's specificity and sensitivity (i.e., recall). Weights correspond to positive and negative tweets' prevalence in the evaluation data.[11] Consequently, equal weight is given to classes independent from the prevalence in the training data.

[11] With more than two label classes, class-specific BA metrics are obtained by computing specificity and sensitivity values in a one-against-all setup.

Table S.11: Mean cross-validation performance of best GLM-Net model in training split of first sample of labeled tweets.

	Balanced Accuracy		F1		Precision		Recall		Specificity	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Mean	0.603	0.020	0.391	0.035	0.395	0.035	0.389	0.031	0.816	0.012
"General"	0.584	0.041	0.300	0.077	0.326	0.087	0.283	0.077	0.886	0.023
"Specific"	0.627	0.047	0.337	0.083	0.343	0.095	0.339	0.092	0.915	0.021
"Unsure"	0.541	0.039	0.163	0.071	0.168	0.082	0.159	0.078	0.924	0.019
"No"	0.657	0.030	0.759	0.027	0.744	0.033	0.776	0.034	0.539	0.051

The best model achieved a cross-class mean BA of 60.253%. This model is very performant in terms of classifying "No" tweets (last row). It is very precise (about 76% of tweets labeled "No" are correctly classified), and it has a high recall (about 84% of actual "No" tweets are correctly classified).

It performs far worse on the other label classes, however. Only about 39% of tweets *labeled* "General" and only 32% of *actual* "General" tweets are correctly classified (precision and recall, respectively). Only about 35% of tweets *labeled* "Specific" and only 29% of *actual* "Specific" tweets are correctly classified. And only about 13% of tweets *labeled* "Unsure" and only 9% of *actual* "Unsure" tweets are correctly classified.

To learn more about how exactly the model erred, we inspected the model's confusion matrices across CV held-out sets. This revealed that, on average, a little less than twice as many actual "General" tweets are misclassified as "No" tweets as are correctly classified. For actual "Specific" tweets, this ratio is a little higher than 1:1. And for actual "Unsure" tweets, it is about 1:1. Thus, the vast majority of positive classes' tweets tend to be classified into the "No" class and not the other, conceptually closer categories.

These findings replicated in the validation set. Hence, the classifier obtained from the first sample is strongly predictive of the *absence* of elite criticism, but it performs very poorly in detecting tweets of the positive label classes ("General", "Specific" and "Unsure"). It seems that the data of the first sample is too imbalanced to detect elite criticism tweets reliably.

C.5 Sampling from machine-labeled tweets for the second round of crowd coding

We nevertheless decided to sample tweets for the second round of crowd coding only from the subset of tweets the classifier assigns a label in the set of positive label classes—hence, tweets for which there are reasons to believe that they contain elite criticism. We decided so for three reasons. First, the classifier has high "No"-precision and recall. Hence, the proportion of tweets erroneously classified into the "No" class is relatively low (precision), and the number of actual "No" tweets correctly classified is already relatively high (recall).

Sampling some of the few false-"No" tweets would thus require to sample many tweets predicted to be "No" because many of these will be correctly classified. Accepting this bias is justifiable. For one, alleviating class imbalance is important to obtain a reliable classifier. For another, attempting to mitigate bias would require to again collect many codings for tweets predicted to be "No" tweets of which the vast majority will likely turn out to be true-"No"s,

Second, the proportion of actual "No" tweets among those tweets misclassified in the first three columns of the confusion matrix is high across positive label classes. Sampling only from tweets that are currently predicted to be members of one of these label classes would again have resulted in a prevalence of the "No" label between 40 and 50%. Even when selectively sampling only from unlabeled tweets predicted to feature some elite criticism, we can therefore only mitigate but likely not entirely resolve the class imbalance problem.

Third, the misclassified (actual) "No" tweets that would be contained in the second-round sample should help to improve supervised classifiers' positive-classes detection abilities.

Hence, we first used the GLM-Net model to predict label class membership of all tweets that were not crowd-coded in the first round. We then subset the data to tweets predicted to contain any type of elite criticism and for which an English-text version was available (sampling eligibility criteria). Finally, we applied the same cluster-specific sampling rule as for the first sample (using the same clusters). Overall, this rule implied sampling a total 2762 tweets for the second round of crowd coding.

C.6 Results of the second round of crowd-sourced coding

We have again distributed the 2762 tweets in the second round sample for crowd coding via AWS *SageMaker Ground Truth* using the same custom coding scheme. We collected six judgments for each tweet. In total, crowd workers contributed a total judgments. Table S.12 reports the frequencies of coders absolute label choices (judgments).

Table S.12: Total judgment frequencies in second round of crowd coding.

Judgment	*N*	Proportion
No elite criticism ("No")	6053	0.365
General elite criticism ("General")	4770	0.288
Specific elite criticism ("Specific")	3206	0.193
Ambiguous elite criticism ("Unsure")	2220	0.134
Cannot answer ("Cannot answer")	321	0.019
`invalid`	2	0.000

Taking together types, % of judgments suggest some form of elite criticism. Note that

the percent of "No" labels assigned comes very close to what we expected according to the false-positive rate of the GLM-Net classifier used to sample likely positive instances of elite criticism into the second-round sample.

C.6.1 Coder statistics

A total of crowd workers contributed their judgments. However, as shown in Figure S.13, the average number of judgments contributed again varied across coders.

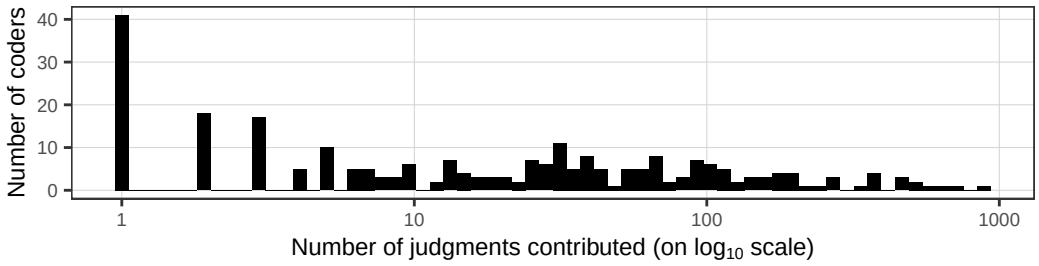


Figure S.13: Distribution of judgments contributed per coder in second round of crowd coding.

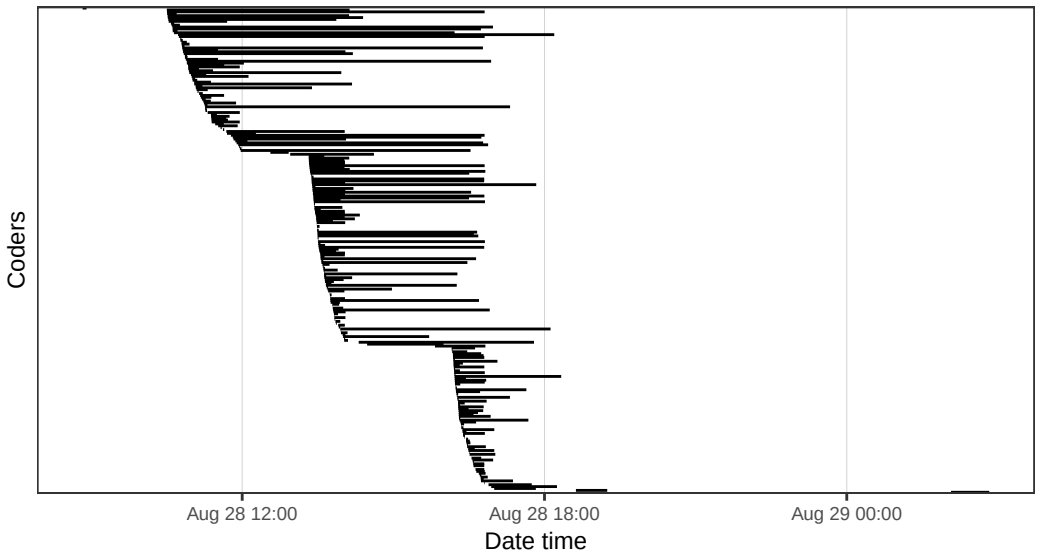


Figure S.14: Coder activity times and duration in the second round of crowd coding. Each horizontal line represents the activity period and duration of a single coder.

In Figure S.14 we see that, again, only few coders spent more than 30 minutes or an hour on the task (NaN% and NaN%, respectively). As in the first sample, we believe that coders that contributed few judgments and/or only for a short time were likely on a "shopping tour" and may thus provide random or even adversarial judgments.

Because we doubled the per-task pay in the second round, the inter-quartile range of coders' median per-judgment duration contains the expected 20 seconds determining

coders per-judgment pay. In fact, with 19.68 seconds, the median of medians is below that value.

Looking at Figure S.15 there is again indication that a couple of coders may have contributed poor-quality judgments.

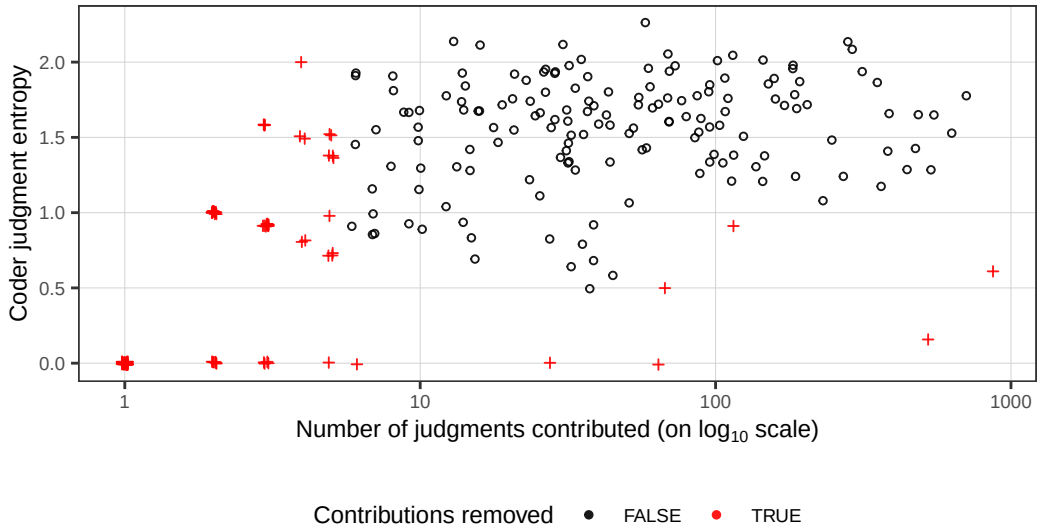


Figure S.15: Coders' numbers of judgments against judgment entropies in second round of crowd coding. Vertical and horizontal jitter of max. 1% added to avoid over-plotting.

C.6.2 Tweet-level statistics

Turning to descriptives at the tweet level, Figure S.16 shows that tweet-level label variability again varies across label classes.

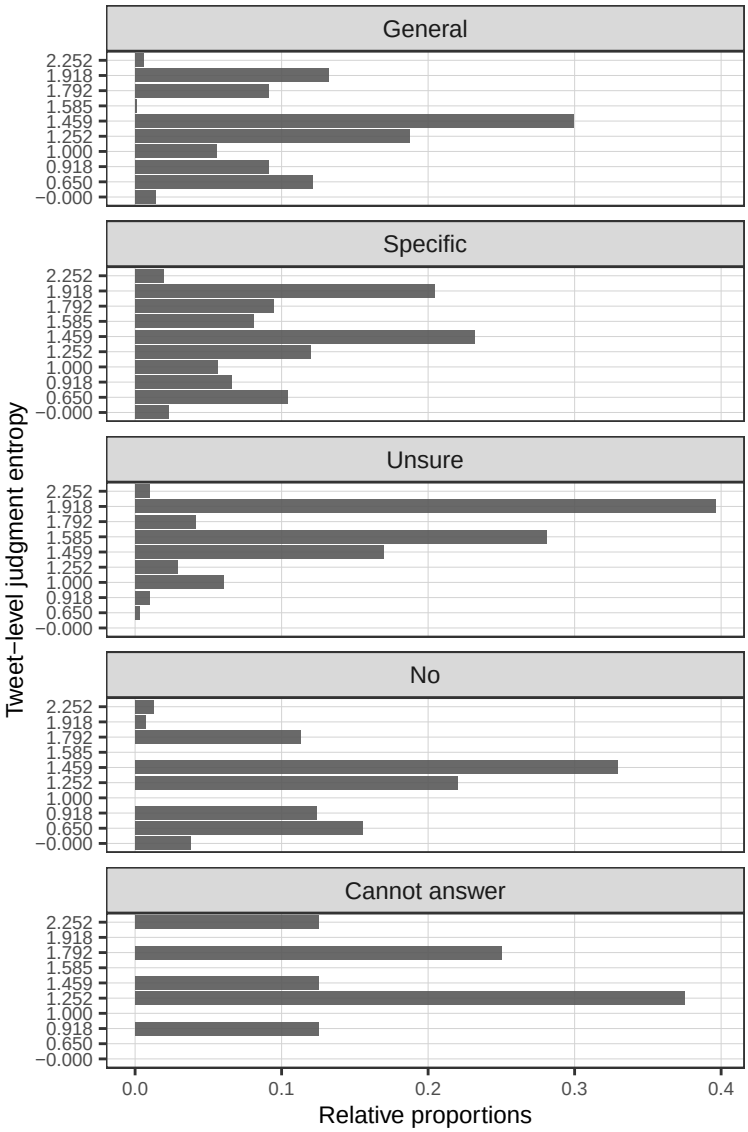


Figure S.16: Tweet-level variability in labels assigned (entropy) in the second round of crowd coding. Panel columns group tweets by labels that were most frequently assigned to them.

C.7 Judgment aggregation: Inducing tweet-level labels from pooled crowd codings

Table S.13: Sample descriptives and crowd-coding statistics. Rows report values for samples 1 and 2 collected in the first and second iteration of our selective sampling strategy. Column two reports the total number of tweets distributed for coding in each round sample. Columns grouped under the header "Crowd-coding statistics" report the number of codings collected per tweet, the total number of codings collected the total number of coders that have contributed to coding and the per-task pay. Columns grouped under the header "Median coding duration" report summary statistics of the coder-level median times spent for coding a tweet.

		Crowd-coding statistics				*Coder: Median coding duration*			
Sample	Tweets	per tweet	Codings	Coders	Task pay	Median	Mean	SD	Skew
1	3270	6	19620	329	$ 0.024	22.7	53.2	148	9.27
2	2762	6	16572	256	$ 0.048	19.7	45.7	109	9.43

Table S.14: Number of codings across categories retained at subsequent data cleaning steps by sample.

Removal steps	**Coding categories**						
	Cannot answer	No	Yes, general	Yes, specific	Yes, but unsure	`NA`	*Total*
Sample 1							
	312	12803	3131	2036	1338	0	19620
codings of "low-quality" coders	269	7751	2763	1736	1185	0	13704
judgments made in < 4 seconds	268	7697	2754	1733	1181	0	13633
majority-"Cannot answer" tweets	222	7684	2748	1726	1177	0	13557
other "Cannot answer" codings	0	7684	2748	1726	1177	0	13335
Sample 2							
	321	6053	4770	3206	2220	2	16572
codings of "low-quality" coders	220	5114	4574	3113	1661	0	14682
judgments made in < 4 seconds	219	5000	4469	2913	1649	0	14250
majority-"Cannot answer" tweets	188	4993	4463	2906	1644	0	14194
other "Cannot answer" codings	0	4993	4463	2906	1644	0	14006

Given these indications of coder and tweet-level differences in the quality of codings, we again adopted an annotation modeling approach to aggregate crowd codings into estimates of coded tweets' "true" labels. Specifically, we have taken the codings collected in both rounds (cf. Table S.13) and cleaned the annotations data as described in Table S.14. For example, in the first coding cleaning step, we have removed all judgments of coders marked red in Figure S.17.

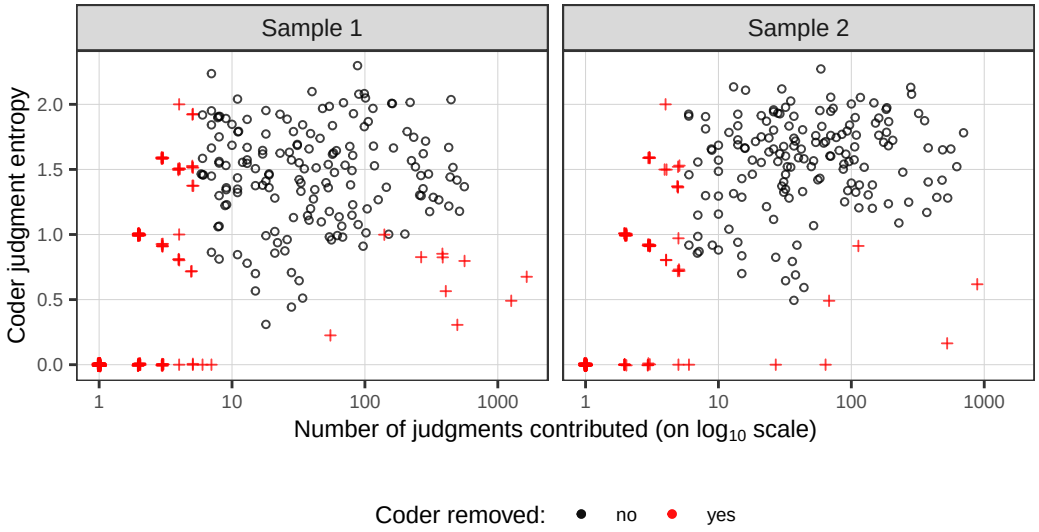


Figure S.17: Number of judgments against coder judgment entropy by sample. Vertical and horizontal jitter of max. 1% added to avoid over-plotting. Removed (retained) coders are shown as red crosses (hollow black circles). Note that values on the horizontal axis are reported on a $\log_{10}$-scale.

We have then fitted a Dawid-Skene per-annotator model to the annotations data retained after these cleaning steps. Table S.15 reports label classes' estimated prevalence and their proportions and counts resulting from this model-based judgment aggregation. Class imbalance in the labels estimated from the pooled and cleaned judgments data is still quite pronounced in favor of the "No" label. Our selective sampling strategy was effective in, however, the sense that class imbalance in induced labels is less severe in the second sample compared to the first one (see Table S.6).

Recall, however, that the objective of our two-step sampling procedure has been to obtain a sample of tweets that is (a) as class-balanced as possible and (b) maximally diverse in terms of country, party, and semantic composition. Hence, the label class prevalence estimates reported in Table S.15 are hardly good estimates of label classes' "true" prevalence in the entire corpus.

Table S.15: Model-based label prevalence estimates and label proportions and counts due to model-based aggregation from Dawid–Skene model fitted to the pooled codings in samples 1 and 2 retained after data cleaning.

		Induced labels	
Label class	Est. prevalence	Proportion	*N*
"General"	0.216	0.212	1271
"Specific"	0.131	0.127	760
"Unsure"	0.117	0.107	644
"No"	0.536	0.554	3322

Figure S.18 also shows that in the pooled and cleaned codings judgments data, most coders are estimated to be far better than chance in correctly labeling tweets (see also Table S.16). The variability in these estimates is non-negligible, however, and there is an indication that the Dawid–Skene model puts relatively much weight on a few coders' "Unsure" codings when estimating tweets label class membership probabilities.

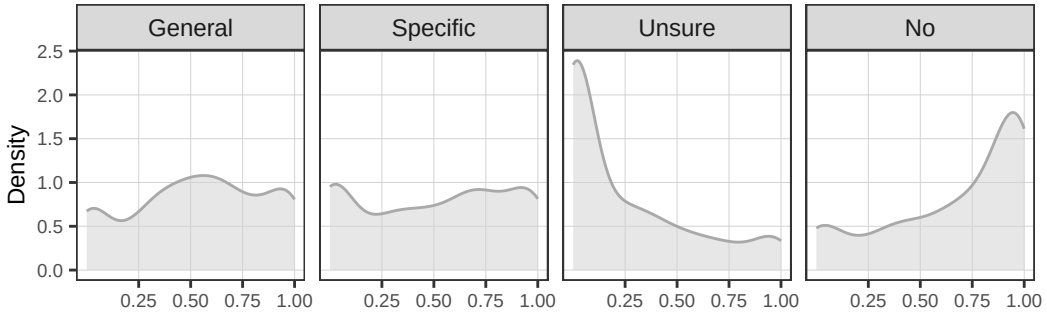


Figure S.18: Distribution of estimated coder label class detection ability parameters obtained by fitting a Dawid–Skene model to the pooled codings in samples 1 and 2 retained after data cleaning.

Table S.16: Summary statistics of coder label detection ability estimates of the Dawid–Skene model fitted to the pooled and cleaned codings.

	Moments			Quantiles					
	Mean	Std. Dev.	Skewness	10%	25%	Median	75%	90%	% > chance
"General"	0.531	0.318	-0.180	0.010	0.309	0.544	0.809	0.996	0.782
"Specific"	0.500	0.354	-0.102	0.005	0.182	0.532	0.794	0.995	0.700
"Unsure"	0.250	0.313	1.100	0.002	0.005	0.064	0.420	0.992	0.371
"No"	0.669	0.331	-0.784	0.084	0.432	0.785	0.960	0.999	0.847

D Classifier training, evaluation, and validation

The next step of our measurement strategy consists in training a supervised machine learning classifier with the task of discriminating between "General" and instances of other label classes in the set of labeled tweets. We have discarded all tweets the Dawid–Skene model fitted to the pooled annotations data has assigned "Unsure" labels, assuming that the they result from coders' occasional lack of confidence in their judgments regarding the general–specific distinction. Next, we have combined the "No" and "specific" elite criticism labels into a single class label. Following standard terminology for binary classification, below we refer to the collapsed category as the "negative" label class and to "General" elite criticism tweets as instances of the "positive" label class.

D.1 Learning approaches

To train our classifier, we have adopted a large language model (LLM) fine-tuning approach. Specifically, we have fine-tuned a classification head for a pre-trained "XLM-T" model, a cross-lingual Transformer-based neural language model Barbieri et al. (2022) have pre-trained on a large, multilingual Twitter corpus. LLM fine-tuning approach leverages the potential benefits of transfer learning, that is, it allows to build on the latent linguistic "knowledge'' learned by the LLM during pre-training on an external corpus when training a new classifier for our classification task: identifying tweets that contain anti-elite rhetoric.

To benchmark the performance of this classifier, we have also implemented two alternative learning approaches: First, supervised classification using tweet texts' multilingual sentence embeddings (MSEs) as inputs (cf. Licht, 2023). The MSE approach, too, hinges on the idea of transfer learning (cf. Artetxe and Schwenk, 2019; Reimers and Gurevych, 2020). However, we have used tweets' embeddings as static features. The embedding model's parameters are thus not updated ("frozen") when updating the classifiers parameters to optimize performance in the target corpus. This makes training and evaluation more time-efficient compared to the end-to-end fine-tuning of a large language model with a classification head. However, it also limits the classifier's ability to leverage the latent linguistic knowledge captured by the embedding model when trying to solve the target task. Moreover, it is important to stress that the embedding models we use have *not* been pre-trained on tweets or other social media data. Using them to embed tweets might thus reduce text representation quality. Both limitations might result in reduced classification performance.

The second benchmark approach we include in our analysis is classic bag-of-words (BoW) classification using machine translation (MT) to align tweets' text representations across languages. This MT+BoW approach is currently still the dominant classification

strategy in the existing political and communication science literature.

D.2 Training strategies

For each of these benchmark approaches, we have evaluated the performance of different algorithms. For the MSE approach, we have trained L2-regularized linear models as well as multilayer perceptrons (MLPs). For the MT+BoW approach, we have trained Naive Bayes, Support Vector Machine (SVM), L2-regularized linear models, and MLPs. This was thought to ensure that our assessment of benchmark classifiers' out-of-sample performance is not algorithm-dependent.

In addition, we have tuned critical hyper-parameters for all classifiers. Specifically, we have evaluated all classifiers on the same 80% of labeled tweets using 5-fold cross-validation (CV). In this CV process, we have performed grid search for each algorithm over a range of hyper-parameter values to identify the values that yielded the best average-CV out-of-sample performance.

In the case of the XLM-T model, we have varied only the class weighting scheme. Increasing a class's relative weight in the loss function increases the cost of misclassifying its instances and thus incentivizes a classifier to correctly label them. It is also a popular strategy to counter class imbalance because it leaves the label distribution in the training and validation data unchanged. Accordingly, we have cross-validated XLM-T classifiers that weighted the positive label class 1:1 to the negative label class that combines "no" and "specific" elite criticism instances (i.e., equal weight), 4:1 (inverse label prevalence), 8:1, and 16:1, respectively. We have only cross-validated the class weights since a single training run (i.e., training for several epochs on one CV training fold) took about 12 minutes on a Google Colab instance.[12] We thus fixed the learning rate at $2e^{-5}$, weight decay at 0.01, the batch size at 32, and the number of epochs at 5.

We have cross-validated the same class weighting scheme for the MSE and MT+BoW-based classifiers. In addition, we have varied critical text pre-processing choices, the choice of the learning algorithm, and a number of algorithm-specific hyper-parameters. The text pre-processing choice we have varied for the MSE approach is the choice of the pre-trained MSE model used to embed tweets, comparing two models: a knowledge-distilled XLM-R model (Reimers and Gurevych, 2020) and the LASER encoder (Artetxe and Schwenk, 2019). We have fed these models tweets' texts as-is (after removing URLs, hashtag and mention symbols). As learning algorithms, we have compared a L2-regularized linear model (Ridge regression or "Perceptron") using the hinge loss and mini-batch SGD for optimization to a multilayer perceptron (MLP) with one hidden layer of 100 units using the ReLU activation function and the ADAM optimizer. For these algorithms, we have cross-

[12] 5-fold CV of four class weighting schemes and training the final model thus took already about 5 hours.

validated various algorithm-specific hyper-parameters in addition to the class weights. For the Perceptron, we have cross-validated α (degree of regularization) $\in \{1e^{-3}, 1e^{-4}, 1e^{-5}\}$. For the MLP, we have cross-validated the batch size $\in \{64, 128, 256\}$, the initial learning rate $\in \{1e^{-2}, 1e^{-3}, 1e^{-4}\}$, and the drop out rate $\in \{0.25, 0.5\}$.

For the MSE approach, we have cross-validated two text-preprocessing choices: Whether to include only unigrams are also bigrams in the vocabulary. And whether or not to tf-idf transform the document-feature matrix.[13] As learning algorithms, we have compared a Naive Bayes classifier, a Support Vector Machine (SVM) with a linear kernel, a L2-regularized linear model ("Perceptron") using the hinge loss and mini-batch SGD for optimization, and a MLP using the ReLU activation function and the ADAM optimizer. For all but the Naive Bayes classifier, we have cross-validated various algorithm-specific hyper-parameters in addition to the class weights. For the Perceptron, we have cross-validated $\alpha \in \{1e^{-4}, 1e^{-5}, 1e^{-6}, 1e^{-7}\}$. For the SVM, we have cross-validated C (cost parameter) $\in \{1, 2, 4, 8\}$ And for the MLP, we have cross-validated the number of hidden layers $\in \{1, 2\}$, the number of hidden units ber layer $\in \{100, 200, 300\}$, the initial learning rate $\in \{1e^{-2}, 1e^{-3}\}$, and the drop out rate $\in \{0.25, 0.5, 0.725\}$.

For each approach and algorithm, we have selected the tuning parameter values that yield the best performance in terms of the average CV macro F1 score. We have used the macro F1 as a criterion because it combines the recall and precision metrics and thus quantifies success in correctly labeling positive instances at a low as possible the number of false-positive classifications.

D.3 Classifier evaluation

We have then trained the models using their best-performing hyper-parameters to the complete training set and evaluated them in the held-out 20% of samples (i.e., the test set). The result of this evaluation are reported in Table S.17.

[13] We have generally discarded features with a document frequency higher than .99 or lower than 0.001 and limited the number of features to 2500.

Table S.17: Performance of models trained with different approaches and alorithms. Models sorted by macro F1 score within approach.

Model	$F1_{\text{macro}}$	$F1_{\text{micro}}$	Precision	Recall	Specificity
LLM fine-tuning					
XLM-Twitter	0.745	0.815	0.644	0.581	0.893
MSE					
XLM-R + Ridge regression	0.685	0.737	0.482	0.660	0.763
XLM-R + MLP	0.639	0.777	0.604	0.317	0.931
LASER + Ridge regression	0.643	0.688	0.422	0.664	0.696
LASER + MLP	0.630	0.765	0.552	0.321	0.913
MTBoW					
Ridge regression	0.611	0.674	0.391	0.543	0.718
SVM	0.585	0.704	0.394	0.336	0.827
Naive Bayes	0.574	0.664	0.352	0.408	0.749
MLP	0.428	0.750	0.000	0.000	1.000

The fine-tuned XLM-T classifier performs best across all evaluation metrics but recall where the L2-regularized linear model trained using XLM-R sentence embeddings performs best. Overall, the XLM-T classifier achieves a macro F1 score of 0.745 and thus is 9% better then the second best classifier (the L2-regularized ("ridge") regression trained using XLM-R sentence embeddings). What is important to note is that the XLM-T classifier achieves a high level of precision and thus very often correct when predicting that a held-out tweet is a "general' ' elite criticism instance. The XLM-T classifier's low false-positive rate also shows in its relatively high specificity, that is, the proportion of negative instances it labeled correctly. However, our XLM-T classifier's recall is outperformed by 14% by the ridge regression trained on XLM-R sentence embeddings. However, the latter classifier exhibits relatively low precision and thus achieves lower micro and macro F1 scores.

Given that our data is multilingual and our classification task is hard, the performance of our XLM-T classifier is also relatively good compared to other classifiers presented in the existing political and communication science literature. Compare, for example, the results reported by Dai and Kustov (2022). Their best classifier achieves an accuracy (i.e. micro F1) of 91% in discriminating between populist and non-populist speech segments. Given that their data is monolingual (English), linguistically far less varied,[14] and their test set likely exhibits slightly stronger class imbalance,[15] the 82% accuracy (micro F1) our XLM-T model achieves is quite competitive. van Atteveldt et al. (2021, Table 2), in turn, report that their best multi-class sentiment classifier achieves an overall

[14] Their data is monolingual and language use in U.S. presidential campaign speeches are likely also simpler and more codified (cf. Benoit et al., 2019; Bischof and Senninger, 2018).

[15] The positive label class prevalence in their labeled corpus is 11.79% before and 20.78% after nine active learning iterations.

accuracy of 0.63 (lower than ours) while the F1 scores for their three label classes range between 0.56 and 0.66 and are thus comparable to the positive-class F1 of our classifier (0.62). This is notable since the labeled dataset analyzed by van Atteveldt et al. (2021) is monolingual (Dutch) and they report higher inter-coder reliability than our crowd coders achieve. Finally, compare the results presented by Widmann and Wich (2022). They have fine-tuned a transformer-based classifier to discriminate between eight emotion categories in a corpus of 10K labeled sentences sampled from German parliamentary speeches and politicians' Facebook posts and report a cross-class average (i.e., macro) F1 score of 0.67.[16] This suggests that our multilingual elite criticism classifier is quite competitive even in head-to-head comparison with a classifier trained using LLM fine-tuning.[17] We thus conclude that our classifier performs relatively reliably compared to other classifiers reported in the existing political and communication science literature.

In addition to assessing and comparing the average out-of-sample predictive performance of our classifier, it seems vital to address the concern that measuring the anti-elite strategies of parties from different countries — and thus across political cultures — with a single, multilingual classifier is not feasible because of the context dependence of anti-elite discourse. After all, it is very likely that anti-elite rhetoric manifests in different tropes and phrasesin different political cultures. Given that these tropes and phrases might not neatly translate across countries, cross-lingual measurement with a multilingual classifier might not be feasible.[18] Below, we address this concern by relating the classification decision of our classifier to patterns in text inputs. Specifically, we seek to explain which text features lead our classifier to assign positive labels to tweets posted by parties from different countries.

D.4 Explaining the classifier's predictions

How can we explain the predictions our XLM-T classifier makes? If we had trained our classifier using bag-of-words text representations as inputs, answering this question would have been relatively straightforward. Many standard machine learning algorithms allow assessing how important specific text tokens are for assigning samples into a given label class by directly inspecting learned model parameters or by computing feature importance metrics (cf. Kuhn and Johnson, 2016, chapter 18). However, since we use a deep neural network model to classify the tweets in our corpus, direct interpretation of learned model parameters is not an option since the relationship between text inputs and their representations in the embedding space is nonlinear and not intelligible.

[16] Own calculation based on Table 1 in Widmann and Wich (2022).

[17] The limitation of this comparison is that they train a multi-class classifier, not a binary classifier. However, note that they use almost twice as much labeled data for fine-tuning.

[18] We thank one of the anonymous reviewers for encouraging us to address this concern in greater detail.

Fortunately, direct inspection of model parameters or computing feature importance metrics is not the only way to make model outputs intelligible (Molnar, 2018). One alternative is to rely on feature extraction methods to identify the text tokens most strongly associated with classification into a given label class. One such method, proposed by Monroe et al. (2008), models text token counts as a function of some categorical variable p. Their model estimates standardized z-scores that indicate how distinctive a word or phrase is when comparing language use between groups defined by p. Given a text classifier, we can let p indicate texts' predicted label class and extract the most distinctive n-gram tokens from pair-wise comparisons between classes.

We have applied this approach to interpret the predictions of our classifier.[19] This analysis shows that the terms found to be most distinctive for tweets our XLM-T classifier predicts to features general elite criticism vary across countries and languages. This suggests that cross-national measurement of parties' anti-elite strategies with a single, multilingual classifier is feasible although anti-elite rhetoric is a context-dependent discursive phenomenon.

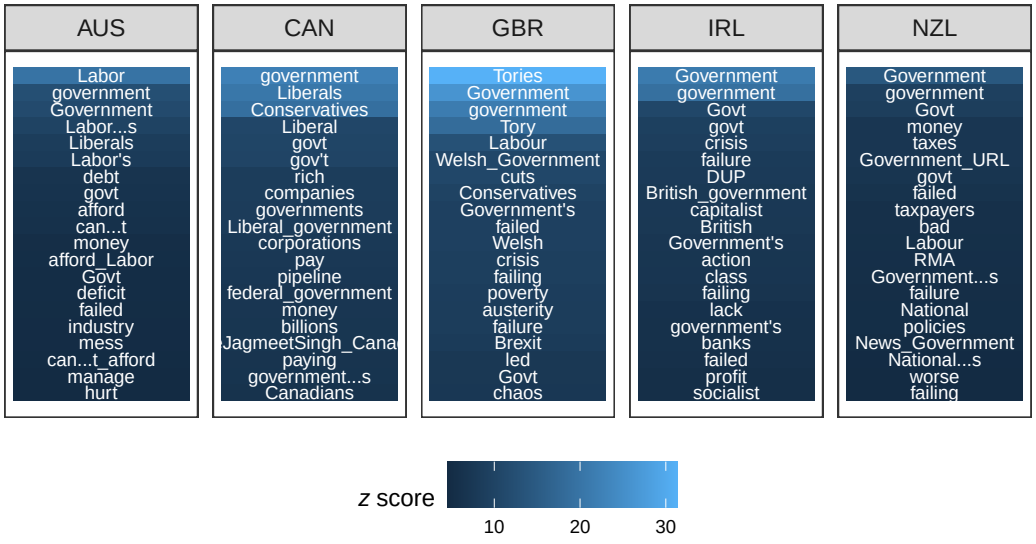


Figure S.19: Text features most distinctive among tweets labeled as general elite criticism instances by our XLM-T classifier in the five English-speaking countries in our sample. Plot panels list the 20 terms per country that are most distinctive among tweets predicted to contain general elite criticism. Color shading indicates z-scores (lighter colors indicate higher scores, i.e., higher distinctiveness). z-scores obtained by applying the feature extraction method proposed by Monroe et al. (ibid.).

Take for example, differences in the features across English-speaking countries (Figure S.19). References to the government and governing mainstream parties are much more

[19] Note that before fitting the model to extract most distinctive features, we have tokenized tweets into words; split words at hyphens; removed punctuation, numbers, symbols, separators and (language-specific) stop words; and extracted all unique uni-, bigram tokens. When estimating z-scores we have specified flat token count priors.

prevalent among anti-elite tweets than other tweets in all five English-speaking countries in our sample. Further, several words among most distinctive terms have a negative sentiment such as "failed", "mess", and "crisis." However, the issue-related terms that are most distinctive for tweets predicted to contain general anti-elite criticism differ across countries. In Australia it are terms relate to monetary issues ("debt", budgetary "deficit", "can't afford", and "money"). This holds, too, for the United Kingdom, Ireland, and New Zealand. In contrast, some of the distinctive issue-related terms in the United Kingdom relate to social and welfare issues like poverty. And in Irish parties' tweets, there are terms like "capitalist" and "profit" that indicate a left-wing connotation of the elite (March, 2011).

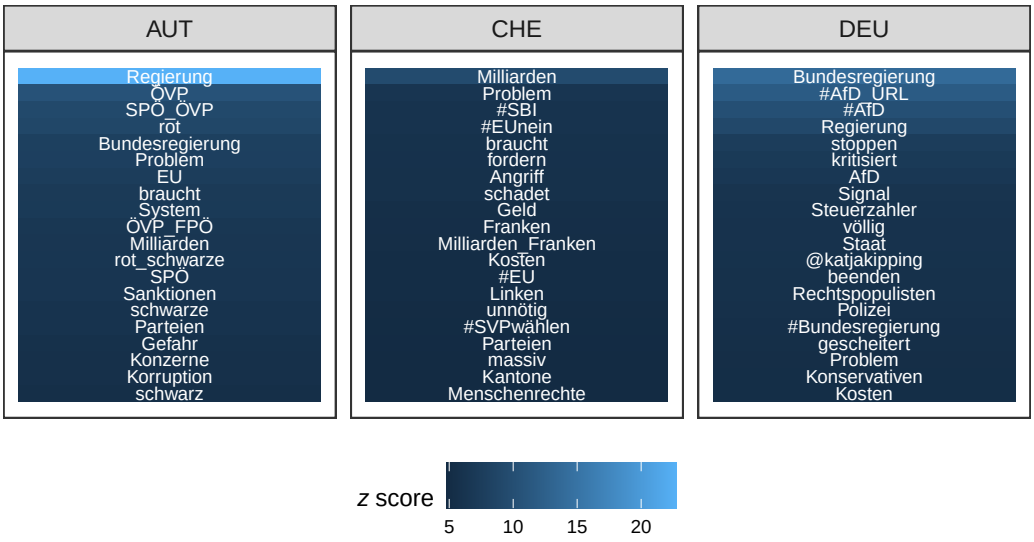


Figure S.20: Text features most distinctive among tweets labeled as general elite criticism instances by our XLM-T classifier in the German-speaking countries in our sample. Plot panels list the 20 terms per country that are most distinctive among tweets predicted to contain general elite criticism. Color shading indicates z-scores (lighter colors indicate higher scores, i.e., higher distinctiveness). z-scores obtained by applying the feature extraction method proposed by Monroe et al. (ibid.). *Note:* Results for Luxembourg omitted because of low number of anti-elite tweets.

Similar patterns can be found in the German-speaking countries in our sample (Figure S.20): Among the terms that are most distinctive for tweets predicted to feature general elite criticism, some refer to the government ("Regierung" and "Bundesregierung"), governing coalitions ("rot schwarze", "GroKo") or governing mainstream parties ("ÖVP", "SPÖ", and "SPÖ ÖVP"). Other terms have a strong negative sentiment like "Gefahr" (*danger*), "Problem" (*problem*), "Korruption" (*corruption*), Angriff" (*attack*), "schadet" (*harming*), "unnötig" (*unnecessary*), "gescheitert" (*failed*), or "Kosten" (*costs*).

Further, our XLM-T classifier also taps into systematic differences in anti-elite discourse in multilingual countries. In Spain, for example, there are notable differences

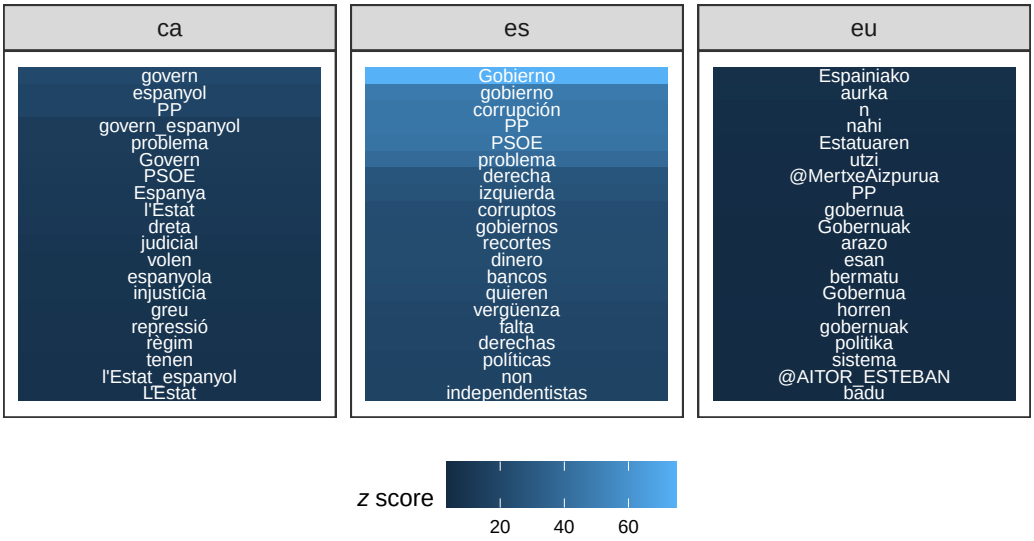


Figure S.21: Text features most distinctive among tweets labeled as general elite criticism instances by our XLM-T classifier in Spain. Plot panels list the 20 terms per language that are most distinctive among tweets predicted to contain general elite criticism. Color shading indicates z-scores (lighter colors indicate higher scores, i.e., higher distinctiveness). z-scores obtained by applying the feature extraction method proposed by Monroe et al. (2008). *Note:* Results for tweets written in Portuguese omitted due to the small sample size.

between the features most indicative of tweets predicted to containing anti-elite rhetoric depending on tweets' language (see Figure S.21). As a point in case, words and phrases used to refer to the Spanish government and central state are found to be most distinctively anti-elitist among Catalan tweets. Further, terms like "injustícia" (*injustice*) and "repressió" (*repression*) which call for freedom to political prisoners, indicate that generalized elite criticism is most frequently voiced in the context of tweets related to the issue of Catalan independence. In contrast, in Spanish-language tweets, terms to name the government are found to be very distinctively anti-elitist, too. But the issue-related terms that dominate are corruption ("corrupción"), money ("dinero"), and (financial) cuts ("recortes").

These examples show that our multilingual classifier is quite robust to cross-country and cross-lingual variation in the patterns of anti-elite discourse. We believe that from a methodological perspective, this should mitigate concerns that measuring the anti-elite strategies of parties from different countries — and thus across political cultures — with a single, multilingual classifier is not feasible because of the context dependence of anti-elite discourse.

E Additional measurement validation

E.1 Validation against CHES data

In the main paper, we compare our anti-elite emphasis estimates against those included in the Chapel Hill Expert Survey (CHES), waves 2014 and 2019 (Bakker et al., 2020; Polk et al., 2017). Table S.18 reports these correlations (a) by CHES wave and (b) as a function of the extent of the time window before the respective CHES field end dates we have used to compute our estimate.

Table S.18: Correlations of CHES anti-elite salience estimates with our anti-elite strategy estimates computed by aggregating varying numbers of tweets prior to CHES waves' field end dates. Our estimates of anti-elite strategies were obtained by aggregating parties' tweets in the 8, 6, 4, respectively 2 quarters prior to the field end data of a given CHES wave. Parties with less than 100 tweets in these date ranges were omitted.

	Prior quarters included			
Wave	8	6	4	2
2014	0.463	0.474	0.469	0.449
2019	0.304	0.347	0.366	0.306

E.2 Comparison to dictionary-based measurements

An alternative strategy to measure parties anti-elite strategies at scale would be to apply dictionary methods. As shown in Table S.19, there exist a few keyword lists for detecting instances of anti-elite rhetoric in political text and these all stem from populism literature.

Table S.19: Overview of existing populism dictionaries (with or without separate list of anti-elitism keywords). *Note:* Based on Gründl (2022).

Paper	Description	
Pauwels (2011)	languages	Dutch
	countries	BEL
	domain	party member magazine
	(separate) anti-elitism keywords	no
Rooduijn and Pauwels (2011)	languages	English, Dutch, German, Italian
	countries	DEU, GBR, ITA, NLD
	domain	election manifestos
	(separate) anti-elitism keywords	yes
Bonikowski and Gidron (2016a)	languages	English
	countries	EU countries
	domain	parliamentary speeches (EUP)
	(separate) anti-elitism keywords	no
Bonikowski and Gidron (2016b)	languages	English
	countries	USA
	domain	campaign speeches
	(separate) anti-elitism keywords	no
Gründl (2022)	languages	German
	countries	AUT, CHE, DEU
	domain	social media
	(separate) anti-elitism keywords	yes

However, these dictionaries have important limitations. Most existing populism dictionaries do not explicitly differentiate between the people-centrism and anti-elite subdimensions of populism by including separate anti-elitism keywords. The exceptions are Gründl (ibid.) and Rooduijn and Pauwels (2011).[20] A further limitation of the dictionaries that allow tapping into the anti-elitism dimension is there limited linguistic coverage. Gründl's dictionary records only German keywords. And Rooduijn and Pauwels' dictionary only keywords for (British) English, Dutch, German, and Italian. Last but not least, only Gründl's dictionary has been developed and evaluated for measurement in social media data. Applying Rooduijn and Pauwels' dictionary to our social media corpus, in contrast, implies cross-domain transfer (Osnabrügge et al., 2021).

Despite these limitations, Gründl's and Rooduijn and Pauwels' dictionaries allow us to triangulate the quality of our measurements. To do so, we use the CHES estimates as an external benchmark. Specifically, to compare our measures with those induced by Gründl's dictionary, we have first subset our corpus to German-language tweets from Austria, Germany, and Switzerland since Gründl's dictionary records only German-language keywords. Next, we have computed the prevalence of anti-elitism as indicated by his

[20] Rooduijn and Pauwels (2014) use the anti-elitism dimension to measure parties' populism.

dictionary and the labels predicted by our classifier, respectively, considering a tweet anti-elite if it contains at least one of th keywords listed in Gründl's dictionary. We again include tweets publisehd in the 12 month prior to the respective CHES waves field end dates. We have then replicated this procedure using Rooduijn and Pauwels' dictionary.

Table S.20: Correlations of CHES anti-elite salience estimates with measurements created with our XLM-T classifier, the dictionary compiled by Gründl (2022), and the dictionary compiled by Rooduijn and Pauwels (2011), respectively. Estimates computed on German-language tweets in Austrian, German, and Swiss parties tweets published in the 12 months prior to the respective CHES waves' field end date.

		Dictionary	
CHES wave	XML-T classifier (ours)	Gründl (2020)	Rooduijn and Pauwels (2011)
2014	0.491	0.199	0.341
2019	0.665	0.232	0.569

Table S.20 reports the results of this validation.[21] The correlations of anti-elite strategy estimates obtained with our methods with anti-elite salience estimates recorded in the CHES is highest for both waves. What is more the dictionary Pauwels and Rooduijn have developed for measurement in party manifestos converges more strongly with CHES indicators than the dictionary Gründl has developed for measurement in social media data. Further, Figure S.22 shows that our measurement strategy is less sensitive to changes in the number of quarters (3-month periods) used to obtain party-level measurements.

[21] It is important to note the that correlations reported in Table S.20 deviate from those reported in Gründl (2022). There are several reasons for these deviations: Gründl's social media data covers the period between 2014 to February 2020. He pools data from Facebook and Twitter, and across countries and CHES waves. He uses CHES waves 2014 and 2017 — not 2019 — and the CSES to cross-validate his measures. He only reports overall correlation when pooling these data sources. And he uses different time windows to aggregate tweets. (For the 2014 wave, the complete year — although the field time ended on November 30, 2014. And for the 2017 wave, the complete year — although the field time ended on January 15, 2018.

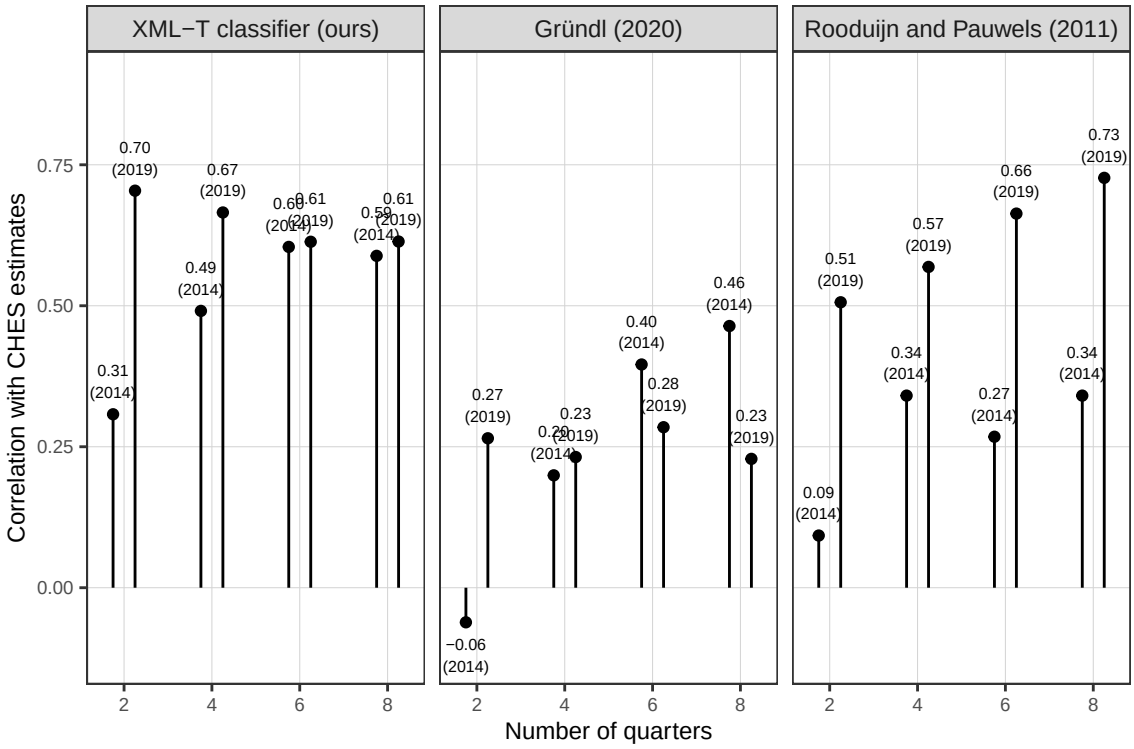


Figure S.22: Correlations of CHES anti-elite salience estimates with measurements created with our XLM-T classifier, the dictionary compiled by Gründl (2022), and the dictionary compiled by Rooduijn and Pauwels (2011), respecitvely, depending on how many quarters we have prior to CHES waves' field end dates we have included when computing party-level estimates. Estimates computed on German-language tweets in Austrian, German, and Swiss parties tweets.

F Regression analyses

F.1 Data

Our regression analyses model parties' anti-elite strategies as a function of their poll results respectively their coalition inclusion probabilities at t-1. The polling and coalition inclusion probabilities data comes from Kayser et al. (2022).

Since we measure parties' anti-elite strategies at the party–quarter level, we have aggregated the polls and coalition inclusion probability indicators at this level, too. Specifically, Kayser et al. (ibid.) provide polls and coalition inclusion probability indicators at a montly level and we have averaged these indicators at the level of quarters.

To extend the temporal coverage of Kayser et al.'s polls data, we have added polling data from POLITICO Europe (2020).[22] Specifically, we have obtained POLITICO Europe's aggregate poll-of-polls estimates that aggregate the polling results of different institutions (using a Kalman filter) at a daily resolution. To aggregate these data into quarterly indicators, we have averaged them at the party–qurater level. The resulting polls data covers all countries in our data but Australia, Canada, and New Zealand. As shown in Table S.21, the combined polls data has overall quite good coverage. The exception is France, for which there is only polling data available in our data sources between 2012 and 2017, and between mid 2012 and 2014 only for the *Parti socialiste*. Figure S.24 shows the distribution of these indicators by country.

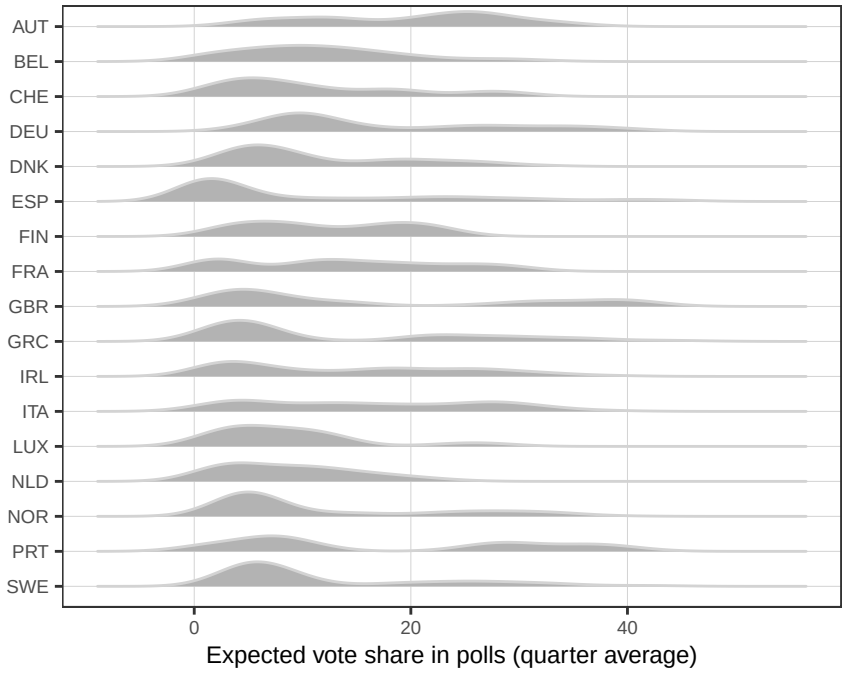


Figure S.23: Distribution of quarterly polls indicators by country.

[22] The data was kindly provided to us by Cornelius Hirsch, POLITICO Europe's Intelligence Analyst.

Table S.21: Distribution of number of party–quarter units for which polls data respectively coalition inclusion probability (CIP) estimates are available by country.

		Quarters per party				
Country	Data	Min.	1st Qu.	Median	3rd Qu.	Max.
AUT	CIP	25	28	44	45	45
	Polls	26	28	44	45	45
BEL	Polls	20	23	23	23	23
CHE	Polls	9	33	33	33	33
DEU	CIP	18	52	57	57	57
	Polls	18	52	57	57	57
DNK	CIP	19	39	43	43	55
	Polls	20	43	43	43	55
ESP	CIP	1	16	25	47	47
	Polls	1	9	25	36	47
FIN	CIP	8	39	51	51	51
	Polls	8	39	51	51	51
FRA	Polls	1	2	2	8	20
GBR	Polls	10	16	24	24	24
GRC	CIP	4	9	11	33	40
	Polls	2	12	21	30	42
IRL	CIP	6	22	47	47	51
	Polls	4	16	36	47	51
ITA	CIP	1	14	32	37	47
	Polls	4	20	32	41	49
LUX	Polls	22	38	38	38	39
NLD	CIP	12	37	52	52	52
	Polls	12	37	52	52	52
NOR	CIP	7	34	50	53	57
	Polls	17	34	52	55	57
PRT	CIP	1	1	1	35	60
	Polls	1	4	17	35	60
SWE	CIP	26	48	51	52	53
	Polls	46	52	53	53	53

The coverage of Kayser et al.'s coalition inclusion probabilities data is somewhat more limited. It only includes data for Austria, Denmark, Finland, Germany, Greece, Ireland, Italy, Netherlands, Norway, Portugal, Spain, and Sweden. Accordingly, our analysis using these data are based on a smaller number of party–quarter observations. Figure S.24 shows the distribution of these indicators by country.

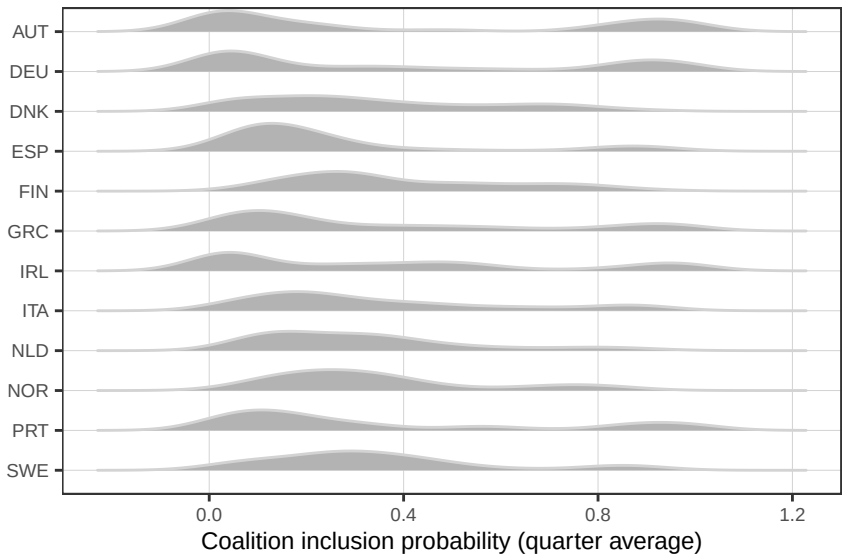


Figure S.24: Distribution of quarterly averages of coalition inclusion probability indicators by country.

F.2 Regression tables

Table S.22: Regression of parties' anti-elite strategies on their average polling results at $t-1$. All variables recorded at the party–quarter level. Models include the lag of the anti-elite strategy indicator and party fixed effects. Regression coefficients estimated using OLS with panel-corrected standard errors.

	Mainstream	Challenger
Lagged DV: Anti-elite strategy $(t-1)$	0.427^{***}	0.186^{***}
	(0.039)	(0.050)
Polling average $(t-1)$	-0.001^{**}	0.004^{***}
	(0.000)	(0.001)
Party FEs	Yes	Yes
R^2	0.183	0.058
Adj. R^2	0.154	0.008
Num. obs.	2546	1420

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table S.23: Regression of parties' anti-elite strategies on their coalition inclusion probability at $t-1$. All variables recorded at the party–quarter level. Models include the lag of the anti-elite strategy indicator and party fixed effects. Regression coefficients estimated using OLS with panel-corrected standard errors.

	Mainstream	Challenger
Lagged DV: Anti-elite strategy $(t-1)$	0.451^{***}	0.248^{***}
	(0.041)	(0.055)
Coalition Inclusion Probability $(t-1)$	-0.067^{***}	-0.039
	(0.015)	(0.039)
Party FEs	Yes	Yes
R^2	0.216	0.068
Adj. R^2	0.190	0.016
Num. obs.	1966	947

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

G References

Artetxe, Mikel and Holger Schwenk (2019). "Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond". In: *Transactions of the Association for Computational Linguistics* 7, pp. 597–610.

Bakker, Ryan, Liesbet Hooghe, Seth Jolly, Gary Marks, Jonathan Polk, Jan Rovny, Marco Steenbergen, and Milada Anna Vachudova (2020). *2019 Chapel Hill Expert Survey.*

Barbieri, Francesco, Luis Espinosa Anke, and Jose Camacho-Collados (2022). "XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond". In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference.* Ed. by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis. Marseille, France: European Language Resources Association, pp. 258–266.

Benoit, Kenneth, Kevin Munger, and Arthur Spirling (2019). "Measuring and Explaining Political Sophistication through Textual Complexity". In: *American Journal of Political Science* 63.2, pp. 491–508.

Bischof, Daniel and Roman Senninger (2018). "Simple Politics for the People? Complexity in Campaign Messages and Political Knowledge: SIMPLE POLITICS FOR THE PEOPLE?" In: *European Journal of Political Research* 57.2, pp. 473–495.

Bonikowski, Bart and Noam Gidron (2016a). "Populism in Legislative Discourse: Evidence from the European Parliament, 1999–2004". Unpublished manuscript. Harvard University, Cambridge MA.

— (2016b). "The Populist Style in American Politics: Presidential Campaign Discourse, 1952–1996". In: *Social Forces* 94.4, pp. 1593–1621.

Canovan, Margaret (1999). "Trust the People! Populism and the Two Faces of Democracy". In: *Political Studies* 47.1, pp. 2–16.

Dai, Yaoyao and Alexander Kustov (2022). "When Do Politicians Use Populist Rhetoric? Populism as a Campaign Gamble". In: *Political Communication* 39.3, pp. 383–404.

Dawid, Alexander Philip and Allan M Skene (1979). "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm". In: *Applied statistics* 28.1, pp. 20–28.

De Vries, Catherine E. and Sara B. Hobolt (2020). *Political Entrepreneurs: The Rise of Challenger Parties in Europe.* Princeton University Press.

Döring, Holger and Philip Manow (2019). *Parliaments and Governments Database (ParlGov): Information on Parties, Elections and Cabinets in Modern Democracies.*

Gründl, Johann (2022). "Populist Ideas on Social Media: A Dictionary-Based Measurement of Populist Communication". In: *New Media & Society* 24.6, pp. 1481–1499.

Hua, Whitney, Tarik Abou-Chadi, and Pablo Barberá (2018). "Networked Populism: Characterizing the Public Rhetoric of Populist Parties in Europe". Paper prepared for the 2018 MPSA Conference.

Jung, Jae-Hee and Margit Tavits (2021). "Valence Attacks Harm the Electoral Performance of the Left but Not the Right". In: *The Journal of Politics* 83.1, pp. 277–290.

Kayser, Mark A., Matthias Orlowski, and Jochen Rehmert (2022). "Coalition Inclusion Probabilities: A Party-Strategic Measure for Predicting Policy and Politics". In: *Political Science Research and Methods*, pp. 1–19.

Khandelwal, Ankush, Sahil Swami, Syed Sarfaraz Akhtar, and Manish Shrivastava (2017). "Classification Of Spanish Election Tweets (COSET) 2017: Classifying Tweets Using Character and Word Level Features." In: *IberEval@ SEPLN*, pp. 49–54.

Kuhn, Max and Kjell Johnson (2016). *Applied Predictive Modeling*. Springer. 600 pp.

Licht, Hauke (2020). "Mitigating the Undue Influence of Domain-Irrelevant Content on Measurements of Elite Political Behavior When Using Social Media Data". In: *Digital Democracy Workshop*. Zurich, Switzerland.

— (2023). "Cross-Lingual Classification of Political Texts Using Multilingual Sentence Embeddings". In: *Political Analysis* 31.3, pp. 366–379.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing Data Using T-SNE". In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605.

March, Luke (2011). *Radical Left Parties in Europe*. Routledge. 259 pp.

Molnar, Christoph (2018). *Interpretable Machine Learning*. Leanpub.

Monroe, Burt L., Michael P. Colaresi, and Kevin M. Quinn (2008). "Fightin' Words: Lexical Feature Selection and Evaluation for Identifying the Content of Political Conflict". In: *Political Analysis* 16.4, pp. 372–403.

Mudde, Cas (2004). "The Populist Zeitgeist". In: *Government and Opposition* 39.4, pp. 542–563.

Nai, Alessandro (2018). "Going Negative, Worldwide: Towards a General Understanding of Determinants and Targets of Negative Campaigning". In: *Government and Opposition*, pp. 1–26.

Osnabrügge, Moritz, Elliott Ash, and Massimo Morelli (2021). "Cross-Domain Topic Classification for Political Texts". In: *Political Analysis* First view, pp. 1–22.

Pauwels, Teun (2011). "Measuring Populism: A Quantitative Text Analysis of Party Literature in Belgium". In: *Journal of Elections, Public Opinion & Parties* 21.1, pp. 97–119.

POLITICO Europe (2020). *POLITICO Poll-of-Polls*.

Polk, Jonathan, Jan Rovny, Ryan Bakker, Erica Edwards, Liesbet Hooghe, Seth Jolly, Jelle Koedam, Filip Kostelka, Gary Marks, Gijs Schumacher, Marco Steenbergen, Milada Vachudova, and Marko Zilovic (2017). "Explaining the Salience of Anti-Elitism

and Reducing Political Corruption for Political Parties in Europe with the 2014 Chapel Hill Expert Survey Data". In: *Research & Politics* 4.1, pp. 1–9.

Reimers, Nils and Iryna Gurevych (2020). "Making Monolingual Sentence Embeddings Multilingual Using Knowledge Distillation". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Empirical Methods in Natural Language Processing (EMNLP), pp. 4512–4525.

Rooduijn, Matthijs and Teun Pauwels (2011). "Measuring Populism: Comparing Two Methods of Content Analysis". In: *West European Politics* 34.6, pp. 1272–1283.

Van Atteveldt, Wouter, Mariken A. C. G. van der Velden, and Mark Boukes (2021). "The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms". In: *Communication Methods and Measures* 15.2, pp. 121–140.

Walter, Annemarie S (2014). "Choosing the Enemy: Attack Behaviour in a Multiparty System". In: *Party Politics* 20.3, pp. 311–323.

Widmann, Tobias and Maximilian Wich (2022). "Creating and Comparing Dictionary, Word Embedding, and Transformer-Based Models to Measure Discrete Emotions in German Political Text". In: *Political Analysis*, pp. 1–16.