
Assessing the epidemiological utility of public
discourse on Twitter during the COVID-19
pandemic

Student:

Tristan Naidoo

02169026

Supervisor:

Prof. Neil Ferguson

Co-supervisors:

Prof. Jose Camacho-Collados

Jameel Institute,
Department of Infectious Disease Epidemiology,
School of Public Health,
Imperial College London

Submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy of Imperial College London

December 5, 2025

Statement of Originality

I certify that this report, and the research to which it refers, are the product of my own work, conducted during the current year of my PhD at Imperial College London. Any ideas or quotations from the work of other people, published or otherwise, or from my own previous work are fully acknowledged in accordance with the standard referencing practices of the discipline.

Copyright

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC). Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose. When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes. Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Data ethics statement

The datasets used in this work were collected using the Twitter API, in full compliance with its terms and services agreement, which continue to apply. I ensured that the privacy of Twitter users was preserved by anonymising all non-verified user mentions and removing potentially identifying URLs. Annotators were shown this anonymised version of the dataset. Given that some tweets were offensive, a warning was included in the annotator guidelines.

Abstract

In the context of epidemiology, accurately evaluating adherence and protective behaviours remains a substantial challenge. To explore whether social media data can provide behavioural insight during disease outbreaks, I examined whether Twitter discussions related to COVID-19, and their associated sentiment, were linked to epidemiological outcomes.

I began by presenting a novel multilingual dataset of 5.15 million tweets collected between January 2020 and January 2022 across eight countries and languages, using 843 disease-, policy-, and vaccine-related search terms.

Using an annotated subset of this dataset, I conducted an empirical evaluation to identify the most effective multilingual sentiment classification approach. I compared lexicon-based models, encoder language models (encoder LMs), and large language models (LLMs). Despite advances in natural language processing (NLP), many public health studies continue to rely on traditional lexicon-based methods. To support the adoption of more accurate, modern approaches, I have provided practical recommendations for model selection.

I applied the best-performing model from the multilingual sentiment classification to the curated dataset. Using the result, I described the temporal evolution of sentiment-stratified tweet volumes and identified the topics that dominated Twitter discussion.

Next, I assessed associations between tweet variables and epidemiological and policy variables using two approaches. First, I used Kendall's τ to test in-sample correlations at zero lag and cross-correlations across a 14-day window. Second, I employed time series modelling to capture non-linear relationships and longer temporal dependencies, using four classical machine learning and seven deep learning models.

Looking ahead, my thesis has demonstrated that integrating social media analytics into outbreak response holds considerable promise for generating timely behavioural insights. This potential is reinforced by advances in NLP that have greatly enhanced our ability to analyse large-scale text data, and my work provides a foundation for future research to build on.

Acknowledgements

I am deeply grateful to everyone who has helped me along the way of whom there are many. To prevent my acknowledgements from getting too long, I will only name a select few.

A big thank you to my supervisors. To Neil, for promoting scientific curiosity and for your invaluable input. Being able to engage with someone with expertise across such a wide range of topics has been a highlight of my PhD and set a marker to strive toward. To Jose, for teaching me about language models from the ground up, for your patience along the way, and for all your support and encouragement. You have both made me a better and more confident researcher, for which I am extremely grateful.

To my interim examiners Anne and Seb, for helping keep me on track and to my final examiners Lucy and Lorenzo, thank you for assessing my work and sorry that there are so many pages. To Chloe, Patrick, Rob, Thom, and especially Ruth, thank you proofreading my work; your insight and corrections greatly improved the clarity of this thesis. To Anna, Anne, Chloe, Christian, Emma, Iwona, Katharina, Mara, Nieves, Nina, Paula, Rhys, Sabine, and Sara thank you for translating search terms and annotating tweets.

To everyone in VB11, and the extended VB11 on the “cool floor”, thank you for your support, the laughs we shared, and for making the office such an enjoyable place to come to. A special thank-you to Patrick, Sara, and Thom for your friendship, support, and consistent kindness. To Rhys, who became my partner in crime (and suffering), thank you for all your encouragement during the final stretch. Finally, thank you to both Christian and Ruth, for being so caring and always offering advice and encouragement.

Thank you to my friends who have made London feel much closer to home. To Alec, Byron, Jaaki, Juan, and Rayno, for keeping me sane with countless nights of Catan and, more generally, for giving me reasons to get away from my computer. To Rayno, for also being my weekend study buddy and sharing some of my most productive days at the British library (followed by Senate House when things were extra bad). To Sean, thank you for backing me throughout my PhD, for dusting off the cobwebs

of your stats brain to listen to me drone on about my work, and for providing much-needed Frisbee breaks.

To my partner Chloe, thank you for your endless support and patience, and for always being willing to help and listen. You might know just as much about my thesis as I do at this point. You have made the last two years of my PhD far easier than they would otherwise have been.

To my family back home, to my brother, sister-in-law, and nephews, JP, Sarah, Hunter, and Luca, thank you for making trips back to South Africa a yearly highlight and for making them restorative and joyful. Finally, and most importantly, thank you to my parents, Mavis and Patrick, for teaching me the value of education, for encouraging me to pursue this PhD, and for your unwavering support and love throughout.

Contents

1	Introduction	1
1.1	Behaviour in an epidemiological context	3
1.1.1	The definition of behaviour and why it is important	3
1.1.2	The use of behaviour in epidemic modelling and informing public policy	4
1.1.3	The challenge of collecting behavioural data	6
1.2	The relationship between sentiment and behaviour	7
1.2.1	Defining and measuring sentiment	7
1.2.2	Linking affective states to behaviour	8
1.2.3	Incorporating affective states in an epidemiological context . .	8
1.3	Measuring protective behaviours and adherence during the COVID-19 pandemic	9
1.3.1	Survey-based measurement of protective behaviours and adher- ence	9
1.3.2	Quantification of protective behaviours and adherence using pas- sive sources	11
1.4	The role of social media in COVID-19	12
1.5	Twitter sentiment during the COVID-19 pandemic	13
1.6	Using models to learn from data	15
1.6.1	Statistical learning versus machine learning	15
1.6.2	The learning problem	17
1.7	The evolution of natural language processing	18
1.8	Introduction to time series modelling	20
1.8.1	The evolution of time series modelling	21
1.9	Thesis aims and structure	24
2	Sentiment analysis methods	26
2.1	Overview	26

2.2	Introduction	26
2.3	Lexicon approaches	27
2.4	Machine learning approaches	28
2.5	Attention	31
2.6	Transformers	32
2.7	Encoder language models	36
2.8	Classification heads	39
2.9	Large language models	43
2.10	Fine-tuning large language models	46
2.10.1	Quantised low rank adaptation	46
2.10.2	The quantised low rank adaptation learning problem	47
2.11	Summary	49
3	Time series modelling	50
3.1	Overview	50
3.2	Introduction	50
3.3	Modelling approaches	51
3.4	Time series learning problem	53
3.5	Model evaluation	56
3.6	Hyperparameter tuning	57
3.7	Machine learning models	59
3.7.1	Least Absolute Shrinkage and Selection Operator	59
3.7.2	Regression trees	60
3.7.3	Random forest	62
3.7.4	CatBoost	63
3.7.5	Support Vector Regression	65
3.8	Deep Learning	67
3.8.1	Training	67
3.9	Recurrent Neural Networks	68
3.9.1	Vanilla Recurrent Neural Networks	68
3.9.2	Gated cells	69
3.10	Dilated Recurrent Neural Networks	70
3.11	Temporal Fusion Transformer	71
3.12	Temporal Convolutional Network	73

3.13	Neural Basis Expansion Analysis for Time Series with exogenous variables	76
3.14	Time series Dense Encoder	78
3.15	Komolgorov-Arnold Network	80
3.16	Summary	82
4	Curating a dataset of tweets	83
4.1	Overview	83
4.2	Introduction	83
4.3	Search queries	85
4.4	Downloading tweets	86
4.5	Filtering tweets	89
4.5.1	Reference tweets	89
4.5.2	Bots	90
4.5.3	Languages	94
4.6	Creating an analysis dataset	94
4.6.1	Plurals	95
4.6.2	Overlapping terms	96
4.6.3	Catch-all terms	97
4.7	Final dataset	97
4.8	Discussion	98
5	Identifying the best approach to multilingual sentiment classification and practical recommendations	101
5.1	Overview	101
5.2	Introduction	101
5.3	Related work	103
5.3.1	Social media sentiment in public health	103
5.3.2	Modern sentiment analysis	104
5.4	Data	105
5.5	Methods	107
5.5.1	Lexicon	107
5.5.2	Encoder	108
5.5.3	Large language models	109
5.5.4	Model specifications	111

5.5.5	Evaluation	112
5.6	Results: the best performing model	114
5.7	Results: variation in LLM performance by type of prompt	117
5.7.1	General performance	117
5.7.2	Relative prompt performance	118
5.7.3	Impact of example choice	120
5.7.4	Generated response length	120
5.7.5	Comparison between prompt length and runtime	121
5.8	Recommendations for the non-expert user	123
5.8.1	Accuracy	124
5.8.2	Runtime	124
5.8.3	Ease of use	125
5.8.4	Versatility	126
5.8.5	Data requirements	126
5.8.6	Data protection	126
5.8.7	Cost	127
5.9	Discussion	127
5.10	Conclusion	131
6	Temporal evolution of tweet variables and their correlation with epidemiological and public policy variables	132
6.1	Overview	132
6.2	Introduction	132
6.3	Related work	133
6.4	Data	135
6.4.1	Tweet variables	135
6.4.2	Epidemiological and public policy variables	135
6.5	Methods	138
6.6	Results	144
6.6.1	Temporal evolution of attention and sentiment polarity	144
6.6.2	Correlation of tweet volume with epidemiological and public pol- icy variables	150
6.7	Discussion	159
6.7.1	The evolution of attention and sentiment polarity	159

6.7.2	Correlation of tweet volume with epidemiological variables and public policy variables	161
6.8	Conclusion	164
7	Time series modelling to quantify associations between tweet, epidemiological, and public policy variables	166
7.1	Overview	166
7.2	Introduction	166
7.3	Related work	168
7.3.1	The context of time series modelling in COVID-19	168
7.3.2	Applications of time series modelling in COVID-19	169
7.4	Methods	172
7.4.1	Evaluation	175
7.4.2	Smoothing	176
7.5	Results	178
7.5.1	Experiment design choices	178
7.5.2	Assessing the utility of exogenous variables	181
7.5.3	Model lags	193
7.6	Discussion	197
7.7	Future work	202
7.8	Conclusions	204
8	Discussion	206
8.1	Summary of thesis	206
8.2	Limitations	208
8.2.1	Data limitations	208
8.2.2	Analytical limitations	210
8.3	Future work	210
8.3.1	Expanding behavioural measures	210
8.3.2	Disaggregated and expanded modelling approaches	211
8.4	Conclusion	214
	Bibliography	216
A	Appendix: Time series methods	252
A.1	Least Absolute Shrinkage and Selection Operator	252

A.2	Regression trees	254
A.3	Support Vector Regression	255
A.4	Vanilla Recurrent Neural Network	256
A.5	Gated cells mechanisms	257
B	Appendix: curating a dataset of tweets	260
B.1	Pydantic model	260
B.2	Creating mapping files	261
B.2.1	Bot filtering	261
B.2.2	Language composition	264
B.3	Tweet composition	264
B.4	Search terms	265
B.4.1	France	265
B.4.2	Germany	266
B.4.3	Italy	268
B.4.4	Netherlands	270
B.4.5	Poland	271
B.4.6	Portugal	272
B.4.7	Spain	274
B.4.8	United Kingdom	275
B.5	Plural mapping	277
B.5.1	France	277
B.5.2	Germany	278
B.5.3	Italy	278
B.5.4	Netherlands	279
B.5.5	Poland	280
B.5.6	Portugal	280
B.5.7	Spain	281
B.5.8	United Kingdom	282
B.6	Catch-all terms	283
C	Appendix: identifying the best approach to multilingual sentiment classification and practical recommendations	284
C.1	Annotation	284
C.1.1	Annotator details	284

C.1.2	Guidelines	284
C.2	Prompts	286
C.2.1	Zero-shot	286
C.2.2	Few-shot	287
C.2.3	Annotation	288
C.2.4	Prompts per fold	290
C.3	Run details	292
C.4	Hyperparameter tuning	292
C.4.1	Quantisation configuration	292
C.4.2	Run time	292
C.4.3	Encoder results	293
C.4.4	LLM results	293
C.5	Cost estimate	293
C.5.1	Encoder LMs and LLMs	293
C.5.2	ChatGPT	295
C.6	Extended model results	295
C.6.1	All model results	295
C.6.2	Best model results without Dutch and Polish	297
C.7	Model details	299
D	Appendix: temporal evolution of tweet variables and their correlation with epidemiological and public policy variables	300
D.1	Most popular terms	300
D.2	Extended correlation results	303
D.2.1	Epidemiological variables and vaccination uptake	303
D.2.2	Policy stringency variables	304
D.3	Sentiment of excluded tweets	305
D.4	Variant Data	307
D.5	Vaccination uptake	307
D.6	Policy stringency variables	309
E	Appendix: Time series modelling to quantify associations between tweet, epidemiological, and public policy variables	310
E.1	Hyperparameters spaces	310
E.1.1	Lasso	310

E.1.2	CatBoost	310
E.1.3	Random forest	311
E.1.4	General Deep Learning (DL) training	311
E.1.5	DilatedRNN	312
E.1.6	LSTM	312
E.1.7	KAN	312
E.1.8	N-Beatsx	313
E.1.9	TCN	313
E.1.10	TFT	313
E.1.11	TiDE	314
E.2	Hyperparameters results	315
E.2.1	Lasso	315
E.2.2	CatBoost	317
E.2.3	Random forest	319
E.2.4	SVR	321
E.2.5	DilatedRNN	323
E.2.6	KAN	324
E.2.7	LSTM	325
E.2.8	N-BEATSx	327
E.2.9	TCN	328
E.2.10	TFT	330
E.2.11	TiDE	331

List of Figures

1.1	Valence-arousal decomposition	7
1.2	Schematic of learning from data	17
2.1	Traditional ML sentiment classification flowchart	29
2.2	Diagram of the transformer architecture	33
2.3	Encoder LM sentiment classification flowchart	39
2.4	Illustration of a frozen encoder with classification head	39
2.5	Diagram of an MLP and a single neuron	40
2.6	Illustration of 5-fold cross-validation	42
2.7	LLM sentiment classification flowchart	45
3.1	Hierarchy of ML (A) and DL (B) models by class	52
3.2	Illustration of different time series forecasting approaches	54
3.3	Illustration of different time series evaluation strategies	56
3.4	Diagram of a regression tree and its corresponding partitioned predictor space	61
3.5	Diagram of an SVR hyperplane	65
3.6	Diagram of an unrolled RNN	69
3.7	An illustration of dilated causal convolution	74
3.8	Diagram of the N-BEATSx architecture	76
3.9	Diagram of the TiDE architecture	79
3.10	An illustration of a KAN	81
4.1	Flowchart of filtering and creating an analysis dataset	83
4.2	Schema of the flattened tabular dataset	88
4.3	Schematic of process to create a dataset from Twitter	89
4.4	Flowchart of filtering and creating a raw dataset	89
4.5	Histogram of author-day frequencies across similar tweets by posting frequency group	93

4.6	Flowchart of creating an analysis dataset	95
4.7	Examples of tweets with multiple search terms (A) and a single search term (B)	96
4.8	Seven-day MA of daily tweet count by search term category and country	99
5.1	Sentiment classification F1 scores by country and class for each model	115
5.2	English versus country-specific language prompt performance by model and prompt type	119
5.3	Prompt performance relative to zero-shot prompts	119
5.4	Mean LLM response length per model by language, prompt, and model type	122
5.5	Prompt length versus runtime by model, model type, and prompt type	122
5.6	Execution time for each model applied to 2,530 tweets	124
6.1	Raw values and seven-day moving averages of (A) cases and (B) hospitalisations	141
6.2	Raw values and seven-day moving averages of (A) deaths and (B) raw vaccination uptake	142
6.3	(A) Raw and seven-day MA of Government Response index and (B) corresponding positive, negative, and combined changes	143
6.4	Distribution of sentiment polarity by country	144
6.5	Seven-day MA of (A) tweet counts and (B) proportions by country and sentiment, with dominant variant periods shaded	146
6.6	Box plots of proportions of responses and likes per day, stratified by (A and C) sentiment polarity and (B and D) search term category	149
6.7	Kendall's τ between tweet volume and epidemiological variables or vaccination uptake with 95% CIs, coloured by country	151
6.8	Kendall's τ between tweet volume and epidemiological variables and vaccination uptake, with 95% CIs, coloured by country and stratified by (A) sentiment and (B) search term category	153
6.9	Kendall's τ between tweet volume and policy stringency variables with 95% CIs, coloured by country	154
6.10	Kendall's τ between tweet volume and policy stringency variables, with 95% CIs, coloured by country, stratified by (A) sentiment and (B) search term category	155

6.11 Comparison of Kendall's τ cross-correlations between tweet volume and (A) epidemiological variables and (B) policy stringency variables by country	157
7.1 Illustrative example of different smoothing approaches	177
7.2 Boxplots of grouped model performance across all countries per out- come, stratified by model type and structure	183
7.3 Boxplots of individual model performance across all countries per out- come, stratified by model type	184
7.4 Boxplots of country performance across all models per outcome	185
7.5 Heatmaps of model performance for epidemiological variables across models and countries	189
7.6 Heatmap of model performance for tweet volume across models and countries	190
7.7 Average difference in MAPE across countries between models without exogenous variables and best exogenous models	192
7.8 Comparison of input sequence length per DL model across epidemiolog- ical variables	194
7.9 Comparison of input sequence length per DL model for tweet volume .	195
A.1 Diagram of different RNN cells	257
B.1 Pydantic model for Twitter API v2 responses	260
B.2 Dendrogram of 60 tweets generated using the bot detection process . .	262
B.3 Histogram of author-day frequencies (50 tweets) across similar tweets at two distance thresholds (0.1 and 0.25)	262
D.1 Maximum Kendall's τ cross-correlation between tweet volume and epi- demiological variables or vaccination uptake, with 95% CIs, coloured by country, stratified by (A) sentiment and (B) search term category . . .	303
D.2 Maximum Kendall's τ cross-correlation between tweet volume and pol- icy stringency variables with 95% CIs, coloured by country, stratified by (A) sentiment and (B) search term category	304
D.3 Distribution of sentiment polarity of (A–B) excluded problematic tweets and (C) excluded vaccine rows by country	306
D.4 Dominant variant by country over time	307

D.5 Vaccination profiles in Spain from ECDC (weekly), OWID (raw and interpolated), with the problematic period highlighted	308
D.6 Absolute change of the Government Response index	309

List of Tables

4.1	Statistics of the groups used in the bot detection process	92
4.2	Summary of the distinct dataset after filtering, stratified by country .	95
4.3	Summary of the analysis dataset after deduplication, stratified by country	98
5.1	Summary of the annotated dataset by country	107
5.2	Hyperparameter search space for encoder models.	109
5.3	Hyperparameter search space for Large Language Models (LLMs). . .	110
5.4	Prompt combinations by instruction, tweet language, and context type	111
5.5	Details of model language support and parameter counts	112
5.6	Macro-averaged sentiment classification metrics across models on the annotated dataset	114
5.7	Predicted classification distribution for each model using the best prompt	116
5.8	Comparison of the standard deviation in F1 scores for dynamic and static prompts	120
5.9	RAG ratings of model classes across accuracy, runtime, ease of use, versatility, data needs, protection, and cost	123
6.1	Outcomes and sources of hospitalisation data	137
6.2	Sources of the vaccination uptake data	137
6.3	Policies included in the Gouvernement Response index	138
6.4	Negative-to-positive tweet ratio across categories and most frequent term proportions	148
6.5	Distribution of tweet engagements across likes and responses	148
7.1	Variable combinations used across models	173
7.2	Lag variable time step options in ML models	174
7.3	The training, validation, and test periods	175
7.4	Experiment design choices	176
7.5	MAPE of smoothed variables by model class	179

7.6	MAPE of C-spline variables by model class and refitting frequency . . .	180
7.7	MAPE of logging and lag choices	181
7.8	MAPE of model classes by exogenous variable combinations	182
7.9	MAPE (%) results for the best exogenous variable combination per model	187
7.10	MAPE of individual models with optimal exogenous combinations, showing median and IQR across countries	191
7.11	Comparison of the optimal lag combinations	196
B.1	Composition of clusters in the bot detection example	263
B.2	Summary of distinct tweets dataset with reference tweets removed, strat- ified by country	264
B.3	Proportion of tweets in the analysis dataset by number of search terms	264
B.4	Search terms used for tweets geotagged to France in the analysis dataset	266
B.5	Search terms used for tweets geotagged to Germany in the analysis dataset	268
B.6	Search terms used for tweets geotagged to Italy in the analysis dataset	270
B.7	Search terms used for tweets geotagged to the Netherlands in the anal- ysis dataset	271
B.9	Search terms used for tweets geotagged to Portugal in the analysis dataset	274
B.10	Search terms used for tweets geotagged to Spain in the analysis dataset	275
B.11	Search terms used for tweets geotagged to the United Kingdom in the analysis dataset	277
B.12	Plural-to-singular mapping for tweets geotagged to France	278
B.13	Plural-to-singular mapping for tweets geotagged to Germany	278
B.14	Plural-to-singular mapping for tweets geotagged to Italy	279
B.15	Plural-to-singular mapping for tweets geotagged to the Netherlands . .	280
B.16	Plural-to-singular mapping for tweets geotagged to Poland	280
B.17	Plural-to-singular mapping for tweets geotagged to Portugal	281
B.18	Plural-to-singular mapping for tweets geotagged to Spain	282
B.19	Plural-to-singular mapping for tweets geotagged to the United Kingdom	282
B.20	Catch-all terms used during categorisation	283
C.1	Set of UK tweets used in each fold for dynamic prompts	291
C.2	Run configuration across the different prompting configurations	292

C.3	The total wall time in hours, minutes, and seconds to fine-tune each model	292
C.4	The best hyperparameter tuning configuration per encoder LM	293
C.5	The best hyperparameter tuning configuration for each LLM per fold	293
C.6	The cost of running each model using AWS compute	294
C.7	Average ChatGPT 4o-mini cost per prompt across 2,530 tweets	295
C.8	Performance metrics of applying each model and prompt to the annotated dataset	297
C.9	Performance metrics of best models excluding Dutch and Polish	298
C.10	Links and citations of the models used	299
D.1	Top five search terms by tweet volume per country before and on or after 9 November 2020, with negative to positive ratios and proportions	302
D.2	Sentiment polarity distribution of excludes tweets	305
E.1	Hyperparameter search space for lasso models	310
E.2	Hyperparameter search space CatBoost models	310
E.3	Hyperparameter search space for Random forest models	311
E.4	Hyperparameter search space for SVR.	311
E.5	Hyperparameter search space for general DL training hyperparameter	311
E.6	Hyperparameter search space for DilatedRNN models	312
E.7	Hyperparameter search space for LSTM models	312
E.8	Hyperparameter search space for KAN models.	312
E.9	Hyperparameter search space for N-BEATSx models	313
E.10	Hyperparameter search space for TCN models	313
E.11	Hyperparameter search space for TFT models	313
E.12	Hyperparameter search space for TiDE models	314
E.13	Lasso hyperparameter specifications for epidemiological responses	315
E.14	Lasso hyperparameter specifications for tweet volume response	316
E.15	CatBoost hyperparameter specifications for epidemiological responses	317
E.16	CatBoost hyperparameter specifications for tweet volume response	318
E.17	Random forest hyperparameter specifications for epidemiological responses	319
E.18	Random forest hyperparameter specifications for tweet volume response	320
E.19	SVR hyperparameter specifications for epidemiological responses	321

E.20 SVR hyperparameter specifications for tweet volume response	322
E.21 DilatedRNN hyperparameter specifications for epidemiological responses	323
E.22 DilatedRNN hyperparameter specifications for tweet volume response	324
E.23 KAN hyperparameter specifications for epidemiological responses . . .	324
E.24 KAN hyperparameter specifications for tweet volume response	325
E.25 LSTM hyperparameter specifications for epidemiological responses . .	326
E.26 LSTM hyperparameter specifications for tweet volume response	326
E.27 N-BEATSx hyperparameter specifications for epidemiological responses	327
E.28 N-BEATSx hyperparameter specifications for tweet volume response .	328
E.29 TCN hyperparameter specifications for epidemiological responses . . .	329
E.30 TCN hyperparameter specifications for tweet volume response	329
E.31 TFT hyperparameter specifications for epidemiological responses . . .	330
E.32 TFT hyperparameter specifications for tweet volume response	331
E.33 TiDE hyperparameter specifications for epidemiological responses . . .	331
E.34 TiDE hyperparameter specifications for tweet volume response	332

Acronyms and Abbreviations

AR Autoregressive

ARIMA Autoregressive Integrated Moving Average

API Application Programming Interface

BERT Bidirectional Encoder Representations from Transformers

BoW Bag-of-words

BPE Byte-Pair Encoding

CDC Centers for Disease Prevention and Control

CLM Causal Language Modelling

CNN Convolutional Neural Network

DL Deep Learning

ECDC European Centre for Disease Prevention and Control

GISAID Global Initiative on Sharing All Influenza Data

GRU Gated Recurrent Unit

GNN Graph Neural Network

EI Expected Improvement

EU European Union

IQR Interquartile Range

KAN Komolgorov-Arnold Network

GLU Gated Linear Unit

GRN Gated Residual Network

LM Language Model

LLM Large Language Model

LoRA Low-Rank Adaptation

lasso Least Absolute Shrinkage and Selection Operator

LSTM Long Short-Term Memory

MA Moving Average

MAE Mean Absolute Error

MAPE Mean Absolute Percentage Error

ML Machine Learning

MLP Multilayer Perceptron

MLM Masked Language Modelling

MSE Mean Squared Error

N-BEATS Neural Basis Expansion Analysis for Time Series

N-BEATSx N-BEATS with Exogenous Variables

NLP Natural Language Processing

NPI Non-pharmaceutical Intervention

OWID Our World in Data

OxGRT Oxford COVID-19 government response tracker

QLoRA Quantised Low-Rank Adaptation

RevIN Reversible Instance Normalisation

RLHF Reinforcement Learning from Human Feedback

RNN Recurrent Neural Network

RoBERTa Robustly Optimized BERT Pretraining Approach

SL Statistical Learning

SLM Small Language Model

SVR Support Vector Regression

TCN Temporal Convolutional Network

TiDE Time Series Dense Encoder

TF-IDF Term Frequency–Inverse Document Frequency

TFT Temporal Fusion Transformer

TESSy The European Surveillance System

TLM Translation Language Modelling

TPE Tree-structured Parzen Estimator

UK United Kingdom

UKHSA United Kingdom Health Security Agency

USA United States of America

VADER Valence Aware Dictionary and sEntiment Reasoner

VAR Vector Autoregression

WHO World Health Organisation

XLM Cross-lingual Language Model

XLM-R XLM-RoBERTa

XLM-T XLM-Twitter

1 Introduction

Disease outbreaks are a recurring feature of human history. Documented epidemics span from antiquity, such as the Justinian plague, through the medieval era, most notably the Black Death, to the early 20th century with the 1918 Spanish flu (Sam-path et al., 2021). In more recent decades, events such as the 2003 SARS outbreak, the 2009 H1N1 influenza pandemic, the 2014–2016 Ebola epidemic, the 2015–2016 Zika virus outbreak, repeated MERS outbreaks since 2012, and the most recent 2020–2023 COVID-19 pandemic, tested modern public health systems and shaped outbreak response. While the pathogens, contexts, and impacts of these events differed, one constant was the need for a timely and effective response. Over time, what has changed is the way such responses can be supported, with advances in epidemiology, computing, and data availability enabling real-time analytical input into public health decision-making during outbreaks.

Analytical modelling has grown in prominence both for improving epidemic forecasts and for enhancing understanding of transmission dynamics and the impact of control measures. These approaches are data-driven and depend not only on epidemiological variables but also on wider information about disease mechanisms and behavioural factors. Such data are not always available, in which case assumptions are made or alternative sources are used (Watson et al., 2021). For instance, when direct measurements of contact rates are unavailable or impractical to obtain at scale, mobile phone mobility data can serve as a proxy, allowing population-level contact patterns to be inferred in near real time (Grantz et al., 2020; Nouvellet et al., 2021).

However, while mobility streams can capture movement and mixing, comparable passive measures of other behaviours remain elusive. The adoption of protective behaviours and adherence to interventions can strongly influence epidemic dynamics, yet are difficult to measure at scale (Funk, Salathé, and Jansen, 2010; Funk et al., 2015; Bedson et al., 2021). While surveys, particularly self-report questionnaires, are the primary means of measuring these behaviours and their determinants, logistical factors such as time lags in data collection and processing, slow response rates, cost limitations, and participant attrition can make them challenging to implement and sustain (Davies et al., 2022). They are also prone to reporting biases and fatigue,

especially when surveys are longitudinal (Egleston, Miller, and Meropol, 2011; Dan et al., 2025). Researchers have therefore explored alternative signals that might reflect public behaviour or attitudes, but with mixed success. An early and widely noted example is Google Flu Trends, a model that aimed to estimate the prevalence of influenza-like illness (ILI) by using aggregated Google search query volumes to approximate the proportion of doctor visits for ILI. Although it initially performed well, its poor performance during the 2009 H1N1 pandemic and its substantial overestimation of the 2012–2013 U.S. influenza season, predicting more than twice the actual peak proportion of ILI visits, has left its legacy as a cautionary tale in using online search data as an alternative source and “big data hubris” for epidemiological surveillance (Butler, 2013; Lazer et al., 2014).

Despite these mixed results, there are pragmatic reasons to reevaluate alternative behavioural signals. Social media adoption and engagement have expanded markedly between the 2009 H1N1 pandemic and the COVID-19 era. During the 2009 H1N1 pandemic, social media was still in its infancy. In 2008, Facebook, the world’s most popular social media platform, reached only about 1.5% of the global population, a figure that rose to around 30% by 2018 (Ortiz-Ospina, 2019). By 2021, the number of social media users worldwide had grown to an estimated 3.91 billion¹, representing nearly half of the world’s population (Statista, 2025; United Nations Department of Economic and Social Affairs, Population Division, 2024). COVID-19 was therefore the first pandemic in which social media platforms served as a widespread channel for real-time public discourse and information exchange. The World Health Organisation (WHO) has characterised the excess of information being disseminated, including both disinformation and misinformation, that causes health-harming behaviours as an “infodemic” (WHO, 2022). It has also launched formal infodemic-management initiatives, underscoring the salience of social media during COVID-19.

Against this backdrop, in this Thesis, I explored public discourse as a potential source of behavioural insight, focusing specifically on public attention to a topic, stratified by sentiment. My hypothesis was that high levels of attention, when paired with a particular sentiment, may serve as a proxy for certain behavioural tendencies. While sentiment alone offers only a broad attitudinal measure and is less nuanced than

¹Note that this figure may include multiple accounts belonging to the same individual.

detailed assessments of beliefs or intentions, its simplicity enables large-scale, cross-country, and cross-language assessment. The COVID-19 pandemic provides an ideal case study, being unprecedented in scale, marked by considerable variation in policy response and public reaction, and accompanied by widespread social media engagement. At the same time, advances in Natural Language Processing (NLP), particularly Transformer-based architectures now make it possible to classify sentiment from text with a level of accuracy and scale not previously achievable.

My overarching aim was to evaluate the utility of public discourse on Twitter² during COVID-19 as a source of behavioural insight for outbreak response. To address this aim, an interdisciplinary approach was required, integrating large-scale data acquisition, NLP-driven sentiment classification, and epidemiological analysis in a multilingual and multinational context.

Throughout the remainder of this Chapter, I first discuss the importance of behaviour in an epidemiological context, the relationship between sentiment and behaviour, and the ways in which behaviour is measured in such settings. I then present the broad framework, formulated as a learning problem, that describes how models are fit to data. Finally, I outline the two domains to which this framework is applied, namely NLP and time series modelling.

1.1 Behaviour in an epidemiological context

1.1.1 The definition of behaviour and why it is important

In infectious disease epidemiology, human behaviour shapes multiple parameters that determine epidemic trajectories (Funk, Salathé, and Jansen, 2010; Weston, Hauck, and Amlôt, 2018). It affects the risk of exposure and transmission by influencing how often and with whom people interact (contact rates) and the probability of transmission during those interactions. Behaviour also modifies the course of infection, influencing the timeliness of case detection and isolation, the duration of infectiousness, and the pace of recovery. At the individual level, relevant practices include mask-wearing, hand hygiene, and safe sex (exposure and transmission); routine screening, symptom reporting, early testing, and self-isolation (detection and infectious period);

²Although the platform is now called “X”, I have continued to refer to it as Twitter, since that was its name at the outset of my Thesis.

and adherence to prescribed medication (recovery) (Davies et al., 2022). Understanding human behaviour in this context is critical for epidemic models to accurately capture disease dynamics within a population, which is subsequently important for improving the support provided to policymakers (Ferguson, 2007).

Many reviews have considered how behavioural factors could be incorporated into mathematical models (Funk, Salathé, and Jansen, 2010; Funk et al., 2015; Verelst, Willem, and Beutels, 2016; Weston, Hauck, and Amlôt, 2018; Bedson et al., 2021). The review by Funk, Salathé, and Jansen (2010), in particular, examined the incorporation of human behaviour into infectious disease models and categorised the drivers of behaviour into two groups. The first was intrinsic, meaning self-initiated and voluntary actions based on internal beliefs or cultural norms. The second was extrinsic, where external factors shape behaviour. Within the extrinsic category, there was a further distinction between self-informed factors, such as perceived risk based on the number of known cases in one's social circle, and mandated interventions, including lockdowns. In this Thesis, building on this categorisation, I classified behaviour into two types: protective behaviours, which may be intrinsic or extrinsic but are self-informed; and adherence, defined as compliance with mandated or recommended measures.

1.1.2 The use of behaviour in epidemic modelling and informing public policy

A common approach to modelling infectious diseases is the use of dynamic transmission models, which incorporate known or assumed biological characteristics of the disease (Grassly and Fraser, 2008). These mechanistic models are widely used because they explicitly represent the hypothesised causal structure of disease spread, for example, transmission pathways, through equations or simulations grounded in biological, physical, or social mechanisms. Given their design, such models are particularly useful for scenario exploration and counterfactual policy evaluation. Incorporating human behavioural responses is challenging, as this requires an explicit structural understanding of behaviour that is often difficult both to specify and to validate. Nevertheless, behavioural effects have been introduced in various ways, for example, through explicit behavioural compartments, changes to contact matrices, or functions linking intervention efficacy to behaviour (Bedson et al., 2021; Epstein, Hatna, and Crodelle, 2021; Saad-Roy and Traulsen, 2023; Gozzi, Perra, and Vespignani, 2025;

Liu et al., 2021; Gimma et al., 2022; Banholzer et al., 2022; Reitenbach et al., 2024). Rather than specifically representing behaviour change, a more common approach has been to estimate a time-varying transmission rate, which can indirectly absorb the combined influence of behaviour, policy, and other unmeasured factors (Flaxman et al., 2020; Brauner et al., 2021; Perez-Guzman et al., 2023). However, this implicit approach is coarse and may not adequately capture changes in behaviour, making explicit behavioural inputs preferable when they are available (Funk et al., 2015; Bedson et al., 2021).

The sensitivity of model outputs to behavioural assumptions was evident in the final stage of the United Kingdom (UK) Government’s 2021 Roadmap modelling, with these Roadmaps serving as strategic plans that used models to project possible future epidemic scenarios under varying assumptions. The Scientific Pandemic Influenza Group on Modelling, Operational sub-group (SPI-M-O) identified future behaviour as a key source of uncertainty in modelling results. In the absence of empirical behavioural data, modellers varied the assumed post-intervention time-varying reproduction number (R_t) across a plausible range to reflect potential changes in contact rates (SPI-M-O, 2021). Keeling et al. (2022) reviewed the use of the Warwick SARS-CoV-2 transmission and COVID-19 disease model across different Roadmaps. They found that, despite some inaccuracies, the model predicted broad trends with sufficient accuracy to inform policy. However, they also identified behavioural responses and vaccine uptake as the two most significant sources of uncertainty. This example underscores the need for continued efforts to enhance quantifiable measures of these values.

In contrast to dynamic transmission models, time series models typically aim to capture predictive relationships between variables, rather than providing an explicit mechanistic representation of biological processes. In these models, the set of input variables, including behavioural indicators, can change depending on data availability and the purpose of the analysis. The way behaviour influences outcomes does not need to be pre-specified; if behavioural inputs are informative, the model can learn the associations directly. Modern DL methods, discussed further in Section 1.8.1, have advanced this capability by learning complex, non-linear relationships between behavioural and epidemiological variables, thereby pushing the boundary of what unstructured or weakly structured models can capture. This flexibility can be an

advantage when the mechanisms linking behaviour to outcomes are uncertain or complex to formalise. However, it comes at the cost of interpretability and may require more extensive data.

Beyond modelling, behavioural data can directly inform policy. Understanding whether behaviours align with expectations enables policymakers to adjust interventions in real time. For example, deviations from expected mask-wearing or distancing patterns could prompt targeted campaigns. A randomised controlled trial in Bangladesh demonstrated that community-level mask distribution and promotion increased mask usage, illustrating that timely behavioural insights can shape policy implementation during an outbreak (Abaluck et al., 2022).

1.1.3 The challenge of collecting behavioural data

As noted above, collecting behavioural data at the necessary scale and frequency is challenging. Some measures, such as vaccine uptake, are recorded in administrative datasets with sufficient spatial and demographic granularity to serve as robust inputs for both modelling and operational planning (Mathieu et al., 2021). By contrast, other behaviours, such as mask-wearing and hand hygiene, are more difficult to quantify and typically require surveys, which pose challenges as noted previously (Davies et al., 2022). I expand on how behaviour was captured during the COVID-19 pandemic in Section 1.3.

When direct behavioural data are unavailable, an alternative approach is to consider observable proxies and infer the latent affective states that determine behaviour (Bish and Michie, 2010; Davies et al., 2022). These proxies may be integrated into psychological frameworks of behaviour, which can then inform the design of epidemiological models and guide policy implementation (Williams, Rhodes, and Conner, 2019). Alternatively, they can be incorporated more directly, for instance by parametrising models with behavioural measures fitted to the observed proxy data (Jirsa et al., 2022).

1.2 The relationship between sentiment and behaviour

1.2.1 Defining and measuring sentiment

To identify a suitable latent affective state to proxy in my analysis, I drew on the circumplex model of affect, a widely used framework in affective science for classifying emotions (Russell, 1980). The model posits that emotions arise from two independent neurophysiological dimensions: valence (the level of pleasantness or unpleasantness associated with a feeling) and arousal (the level of activation associated with that feeling). Valence can also be understood as the extent to which a person feels positively or negatively toward something, with neutrality indicating indifference. An example of this decomposition is shown in Figure 1.1.

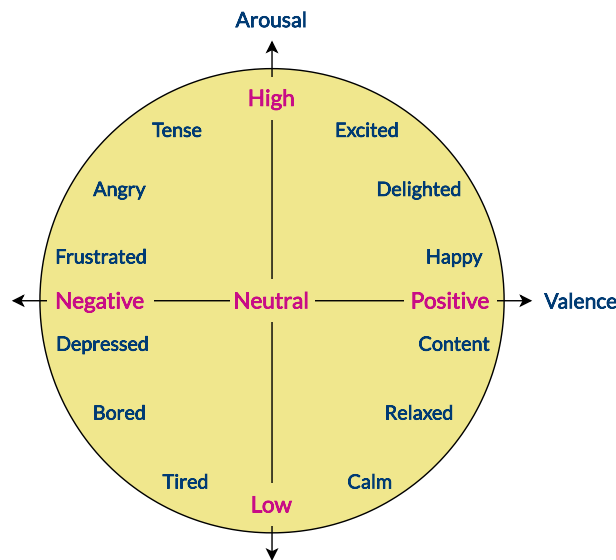


FIGURE 1.1: A decomposition of emotions into a two dimensional valence-arousal space. Source (adapted): Praveen, Cardinal, and Granger (2023).

To measure an affective state, observable outputs such as speech, actions, or written text are used. Although emotions can be mapped to the valence–arousal space, reliably distinguishing discrete categories is challenging (Schuller, 2018; Braun, Goudbeek, and Krahmer, 2022). Measuring valence or arousal entails inferring, respectively, evaluative polarity or activation. In text-only data, valence is more readily cued by lexical choice, whereas arousal is often better captured by acoustic signals (Pang, Lee, et al., 2008; Schuller, 2018). This modality contrast may explain why NLP

has primarily concentrated on valence, with large-scale resources and models targeting sentiment polarity far more often than arousal (Wankhade, Rao, and Kulkarni, 2022). Given these considerations, I adopted a pragmatic approach, using sentiment polarity derived from social media text as an observable proxy for valence. I argue that this proxy remains theoretically meaningful for the study of behavioural outcomes, given its role in the valence–arousal framework; the next Section elaborates on this using theoretical models of behaviour.

1.2.2 Linking affective states to behaviour

To illustrate how affective states map onto theoretical models of behaviour, I present three examples of established psychological frameworks. These examples demonstrate that the influence of affect, particularly its evaluative dimension, is evident across multiple perspectives in behavioural science. The Extended Parallel Process Model (EPPM) links negative valence (e.g., fear) with perceived efficacy to explain whether individuals adopt adaptive “danger control” behaviours or maladaptive “fear control” responses (Witte, 1992). The Health Belief Model (HBM) incorporates affect into perceptions of risk, severity, and self-efficacy, which in turn guide intentions to act (Rosenstock, 1974). The Affect Infusion Model (AIM), which is the example most explicitly focused on valence, focuses on how overall evaluative tone shapes the style and speed of decision-making, regardless of specific emotion labels (Forgas, 1995).

1.2.3 Incorporating affective states in an epidemiological context

In epidemiology, these psychological frameworks can be operationalised in two ways. First, by informing the design of mathematical models, such as specifying structures or influence–response functions that describe how behavioural variables change in response to emotional or perceptual inputs, and second, by defining constructs based on these frameworks that can be directly parameterised and that subsequently influence the dynamics of the system (Jirsa et al., 2022; Reitenbach et al., 2024). Funk, Salathé, and Jansen (2010) note that, although many models incorporate constructs such as fear, risk perception, and protective motivation, these are often based on common-sense reasoning or anecdotal evidence rather than quantitative validation. They also highlight that, despite extensive sociological, psychological, and public health research on behavioural responses to disease, few studies have systematically examined how these responses, as characterised in existing behavioural frameworks, shape epidemic

dynamics. Measuring sentiment from large-scale communication data offers the potential to parameterise behavioural models in a data-driven manner, reducing reliance on unvalidated assumptions.

In addition to informing model parameterisation, affective-behavioural frameworks can also guide policy design in principle. For example, the EPPM predicts that rising negative valence combined with low efficacy should prompt interventions to boost perceived efficacy (Witte, 1992). The AIM, in turn, suggests that shifts in the overall evaluative tone of public discourse can influence both the speed and style of decision-making (Forgas, 1995). However, these are theoretical predictions, and their practical relevance cannot be determined without empirical evidence on the prevailing affective state of the public. This dual role, providing quantitative inputs for modelling and qualitative guidance for policy, highlights the potential of sentiment data to strengthen epidemic projections and support targeted public health interventions.

1.3 Measuring protective behaviours and adherence during the COVID-19 pandemic

1.3.1 Survey-based measurement of protective behaviours and adherence

On 1 December 2019, the WHO was informed of the first cases of what would later be identified as COVID-19 in Wuhan City, China. By 30 January 2020, the outbreak had been declared a Public Health Emergency of International Concern, and by 11 March 2020, it was classified as a pandemic (World Health Organization Regional Office for Europe, 2025). It was the first global health crisis of this magnitude in the twenty-first century, entailing unprecedented lockdowns and widespread disruption to daily life worldwide for nearly two years.

In the initial phase, before vaccines were available, control relied on Non-pharmaceutical Interventions (NPIs) (Hale et al., 2021). The effectiveness of NPIs depended heavily on public adherence (Singh et al., 2021; Davies et al., 2022; Rader et al., 2024). NPIs included government-mandated policies, such as lockdowns, as well as broader measures that exerted extrinsic influence on behaviour, such as information campaigns and public signage about handwashing. When vaccines were developed and rolled out later in the pandemic (from late 2020), uptake remained strongly dependent on

behavioural factors. Vaccination became a highly polarised issue, with hesitancy limiting coverage across many countries (Lazarus et al., 2023; Henkel et al., 2023).

Measuring protective behaviours and adherence proved challenging for the reasons discussed earlier in this Chapter. The global scale of the pandemic added complexity through substantial geographical heterogeneity, shaped by differing epidemic trajectories, culturally driven variations in risk-avoidance behaviours, and variations in the timing and nature of policy interventions (Mathieu et al., 2020b; Yin and Lui, 2024; Mula et al., 2021; Hale et al., 2021).

During the pandemic, many efforts aimed to characterise and understand public protective behaviours and adherence, as well as to track policy interventions globally. The Oxford Supertracker exemplified the latter, providing a comprehensive directory of COVID-19 policy trackers and surveys (Daly et al., 2020). To measure adherence, researchers most often relied on surveys designed to capture self-reported behaviours and associated drivers, such as risk perception.

Most studies were cross-sectional or repeated cross-sectional (Margraf, Brailovskaia, and Schneider, 2021; Smith et al., 2022). The longitudinal studies identified covered short periods and, compared with the cross-sectional ones, had smaller sample sizes (Nivette et al., 2021; Brooks-Pollock et al., 2023). However, it is worth noting that large longitudinal studies have been conducted (Petherick et al., 2021; Wright et al., 2022). A potential explanation for the fewer longitudinal studies was that more resources and longer follow-up times were required with participants (Smith et al., 2022). The studies reviewed varied in their aims, with many considering differences in protective behaviours. Despite contrasting in their results of how protective behaviour changed, all studies concluded that, within their study parameters, behaviour did change. Furthermore, it was uncommon for these studies to consider the association between their survey results and disease outcomes of interest.

In addition to the above studies, two larger repeated cross-sectional studies, both in terms of duration and cohort size, were conducted. The first was a study by Smith et al. (2022), conducted as part of the COVID-19 Rapid Survey of Adherence to Interventions and Responses (CORSAIR) study. This survey ran for two years in the UK, starting in March 2020, and investigated engagement with protective behaviours through questions on hand hygiene, out-of-home activity, and mask use. The second was the CoMix study by Gimma et al. (2022), which reported weekly data in the

UK for one year, also starting in March 2020, adopting the same survey design as the POLYMOD study (Mossong et al., 2008). The study focused on determining the impact of control measures on social contact patterns. Additionally, it posed questions related to risk perception, status, and mitigation. In assessing contact patterns, this study aimed to quantify the epidemiological impact of behavioural change, specifically its effect on transmission rates through social contacts. Both the preceding studies were associated with governmental responses, which may be a plausible reason for their comparatively larger sample sizes and longer study durations.

In addition to behavioural data collected for specific studies, there were also broader attempts to collect this data either in collaboration with or by public companies (Kreuter et al., 2020; Astley et al., 2021; Jones, 2020; Ipsos and Axios, n.d.). For example, the YouGov platform, an online opinion polling company, created a COVID-19 survey to capture a range of behavioural responses to the pandemic (Jones, 2020). This dataset has been used to quantify vaccine hesitancy and vaccine confidence (Ng and Tan, 2023; Mahase, 2021). Furthermore, it has also facilitated the exploration of less common questions; for example, Atchison et al. (2021) explored the impact of NPIs on mental health. The need for the involvement of public companies again emphasises the magnitude of the task of administering surveys to a large sample.

1.3.2 Quantification of protective behaviours and adherence using passive sources

The literature reviewed in the preceding Section actively sought to measure behavioural outcomes through surveys. By contrast, researchers also drew on passive sources, meaning information collected for unrelated purposes but subsequently repurposed to provide insight into behaviour. A typical example was mobility data, which, as previously mentioned, can help estimate contact rates and, in turn, influence patterns of disease transmission. This data is typically passively collected through the background geolocation services of smartphones and other connected devices, which generate aggregated, anonymised mobility datasets (Wesolowski et al., 2016; Grantz et al., 2020). Many NPIs were expected to reduce population movement, and mobility datasets collected by Google, Meta, Apple, and others were used to assess their impact (Google LLC, n.d.; Meta (Facebook) Data for Good, n.d.; Apple Inc., 2020; Ilin et al., 2021a; Snoeijs et al., 2021). Studies at varying levels of granularity examined

how these interventions influenced mobility patterns (Gibbs et al., 2021; Ilin et al., 2021b).

This type of data was also used directly in modelling approaches (Deng et al., 2024). As an illustrative example, Gozzi, Perra, and Vespignani (2025) compared three mechanistic compartmental models, one of which estimated a contact reduction parameter using Google mobility data. This form of data, however, is not without limitations (Chang et al., 2021). For instance, mobility metrics are typically reported only in aggregated form, which complicates efforts to disaggregate analyses by socio-demographic factors (Kostandova et al., 2024). This contrasts with conventional individual-level epidemiological data, such as seroprevalence surveys or risk-factor studies, which are often sufficiently granular to support stratified analyses.

1.4 The role of social media in COVID-19

Beyond mobility indicators, another passive source is social media, which captured the wider information environment during the pandemic. It served the dual role of both providing a platform for public expression and facilitating the circulation of information (Terry et al., 2023; Tsao et al., 2021).

In a scoping review, Terry et al. (2023) investigated the role of social media in relation to disease outbreaks, identifying three main themes, namely surveillance monitoring, risk communication, and disease control. Surveillance monitoring was the most prominent, involving retrospective and real-time analysis of social media data to determine prevalence, assist with predictions, and capture public sentiment. Risk communication focused on the circulation and seeking of information during crises, including official health advice, information-seeking behaviour, misinformation and infodemics, and the growth of telehealth and online support. Disease control examined how social media supported government responses, encouraged behaviour change, and enhanced health education efforts. Similarly, Tsao et al. (2021) conducted a scoping review focused explicitly on COVID-19, covering studies published between November 2019 and November 2020, and identified three comparable themes: social media for surveillance and monitoring, disease control, and as an information contagion and vector.

Social media, therefore, provides an information-rich environment on COVID-19, offering a broad view of content and opinions circulating in the public domain. It also

reflects a wide range of perspectives, including those that may be harder to capture through traditional surveys. However, although social media is a passive source, deriving quantitative insight from it is more challenging because it is primarily text-based. This contrasts with mobility data, which are comparatively straightforward to analyse in numerical form. Overcoming this challenge requires methods from NLP, which allow text to be transformed into quantitative indicators such as sentiment. Across reviews applying NLP methods to social media during outbreaks, including sentiment analysis, Twitter was the most frequently used platform (Tang et al., 2018; Alamoodi et al., 2021; Terry et al., 2023; Pilipiec, Samsten, and Bota, 2023).

Twitter is a microblogging platform that allows users to post short messages (up to 280 characters) and interact with others' posts (Kwak et al., 2010). Its microblogging format produces concise, easily parsed text, and the historic availability of the academic Application Programming Interface (API) provided broad collaborative access to large-scale data streams. These features are potential reasons for its popularity, and, together with its central role in the COVID-19 infodemic, make Twitter particularly well-suited for investigating public discourse during the pandemic.

While Twitter has been widely used, a brief aside is warranted regarding the platform itself. In early 2023, it was acquired and rebranded as X. As noted previously, throughout this Thesis, I continue to refer to it as Twitter, reflecting its name at the outset of my PhD. Shortly after the acquisition, the platform discontinued its free academic API without offering a suitable replacement. This policy shift disrupted not only the present work but also the wider research ecosystem that had long relied on Twitter's data, and it now represents a significant limitation for the reproducibility of research based on the platform (Blakey, 2024).

1.5 Twitter sentiment during the COVID-19 pandemic

There is an extensive literature that has classified sentiment and topics from COVID-19-related Twitter data (Alamoodi et al., 2021; Tsao et al., 2021; Braig et al., 2023). A common approach in these studies has been first to collect tweets referencing COVID-19, then apply topic modelling to identify themes, and finally analyse the sentiment associated with them. In this Section, I highlight an illustrative subset of this work.

Studies varied in whether they were location-specific or not, though all were conducted in English, reflecting a language limitation in the literature (Braig et al., 2023). Among location-agnostic studies, Boon-Itt and Skunkan (2020) analysed 107,990 tweets related to COVID-19 between December 2019 and March 2020, finding that 78% expressed negative sentiment. Expanding the scope, Chandrasekaran et al. (2020) examined 13.9 million tweets and traced the temporal evolution of sentiment across ten topics, one of which focused on government response (including travel restrictions, financial measures, and lockdown regulations). Over the first five months of 2020, they observed that sentiment was initially negative but gradually became more positive. Wang and Chen (2023) conducted a narrower study centred specifically on vaccine-related discourse. Although they analysed fewer tweets (304,292), their dataset spanned a longer period, from March 2020 to March 2021. Using topic modelling, they identified seven themes, with vaccine advocacy and factual information becoming increasingly dominant compared to vaccine hesitancy. Across these topics, sentiment was generally more positive than negative, though vaccine hesitancy consistently attracted more negative sentiment.

In addition to location-agnostic approaches, several studies adopted country-specific perspectives. For example, Xie et al. (2023) analysed four million tweets between March 2020 and May 2021, finding significant spatiotemporal differences in both public awareness and sentiment across the United States. Similarly, Cheng et al. (2021) investigated 33,641 mask-related tweets in the UK between February and August 2020 and observed sentiment spikes corresponding to government announcements, highlighting public responsiveness. A broader cross-country perspective was taken by Wang et al. (2022a), who analysed 654 million geotagged Twitter and Weibo posts from over 100 countries between January and May 2020. They reported a sharp global decline in sentiment at the onset of the pandemic, followed by a slower, asymmetric recovery. As part of this study, the authors also conducted a subanalysis focused explicitly on the public policy of lockdowns. However, they found that lockdown policies had a moderate to no effect on sentiment.

Across these studies, sentiment, and in some cases emotion, was most often analysed using lexicon-based approaches such as Valence Aware Dictionary and sEntiment Reasoner (VADER) (Cheng et al., 2021; Lwin et al., 2020; Cabezas et al., 2021; Boon-Itt and Skunkan, 2020; Cheng, Han, and Liu, 2023). However, some studies

instead employed transformer-based models (Wang et al., 2022a; Lamsal, Harwood, and Read, 2022). Xie et al. (2023) compared both approaches, showing that Bidirectional Encoder Representations from Transformers (BERT) outperformed VADER for COVID-19 sentiment classification. More broadly, from an NLP perspective, transformer-based models have increasingly been recognised as more effective for sentiment classification, with their superiority consistently demonstrated in benchmarking tasks (Haddaoui et al., 2023). Despite the availability of more effective methods, much of the literature has continued to rely on simpler lexicon-based approaches. Lexicon-based and transformer-based models are examined in greater detail in Chapter 2 and compared in Chapter 5.

In addition to sentiment evolution and topic discovery, several studies have examined the association between tweet volume, sentiment, and COVID-19 variables, which are reviewed in Chapter 6. Many reported correlations or associations between tweet-based variables and epidemiological or public policy variables (Doogan et al., 2020; Cheng et al., 2021; Cheng, Han, and Liu, 2023; Turiel, Fernandez-Reyes, and Aste, 2021; Lamsal, 2021). Taken together, the literature reviewed here and in these studies provides a strong foundation for this Thesis.

1.6 Using models to learn from data

1.6.1 Statistical learning versus machine learning

Models approximate real-world processes using available data to make predictions about future events and to draw inferences about past events or unobserved states. Across disciplines, a wide range of approaches exists for learning from data, with key decisions involving how models are specified and how they are fitted. Two overlapping paradigms in this context are Statistical Learning (SL)³ and Machine Learning (ML) (Abu-Mostafa, Magdon-Ismail, and Lin, 2012; Hastie, Tibshirani, and Friedman, 2001; Goodfellow, Bengio, and Courville, 2016; James et al., 2021).

SL typically focuses on identifying a suitable functional form to represent the relationship between response and predictors, often under strong distributional assumptions.

³Note that SL is distinct from Statistical Learning Theory, which provides the theoretical foundations for model construction and for deriving bounds on generalisation error (for example, using measures of capacity such as VC dimension and Rademacher complexity, or distribution-dependent approaches such as PAC-Bayes bounds) (McAllester, 1998; Bartlett and Mendelson, 2002; Vapnik, 2013).

It often prioritises interpretability, inference, and uncertainty quantification, with a focus on understanding the relationship between variables (Ij, 2018). Estimation commonly relies on (penalised) likelihood or Bayesian inference, with diagnostics and model checking integral to the process. Some methods, such as boosting, are notable exceptions and considerably less interpretable, reflecting cases where out-of-sample prediction is the primary objective.

In contrast, ML adopts a more algorithmic and empirical stance, prioritising out-of-sample prediction through optimisation techniques rather than closed-form solutions. It generally imposes fewer assumptions on the data-generating process, often trading interpretability for flexibility and predictive performance (Ij, 2018). At the same time, there is growing interest in interpretability, with post-hoc explainability methods increasingly employed (Ribeiro, Singh, and Guestrin, 2016; Lundberg and Lee, 2017). DL, a branch of ML based on multi-layer neural networks, has become especially prominent in recent years, and its success has contributed to the wider visibility of ML as a paradigm (LeCun, Bengio, and Hinton, 2015).

Growing interest in predictive modelling across disciplines has increasingly blurred the boundary between SL and ML. Shared computational tools reinforce this convergence, as automatic differentiation frameworks and optimisers are now used to fit maximum-likelihood and penalised-likelihood models, as well as deep neural networks (Baydin et al., 2018). These advances have also made approximate Bayesian inference feasible for models that were previously computationally intractable, with Hamiltonian Monte Carlo (HMC) providing one example (Chen, Fox, and Guestrin, 2014). Building on this progress, probabilistic programming has matured, with new languages and libraries emerging (Murray and Schön, 2018; Bingham et al., 2019; Phan, Pradhan, and Jankowiak, 2019). Variational inference further bridges the gap by enabling approximate Bayesian uncertainty in complex machine-learning models, although the reliability of its uncertainty estimates and the flexibility of its approximations, particularly at extreme scale, remain limited (Zhang et al., 2018).

While the contrast between SL and ML highlights important differences, in practice, there is a spectrum of methods that do not fit neatly into either paradigm. Many approaches combine elements of both, borrowing statistical modelling tools alongside algorithmic optimisation techniques, and are applied flexibly depending on context. In this Thesis, I draw on both paradigms as appropriate. I adopt a pragmatic approach

in my explanations, combining elements of SL and ML guided by the problem at hand and the most common framing in the literature.

1.6.2 The learning problem

At the core of both paradigms is the same task of using models to learn from data. Although the specific models may differ, the underlying intuition about how they learn is consistent. To build intuition, in this Section, I adopt a pedagogical framing from Abu-Mostafa, Magdon-Ismail, and Lin (2012) and use it to outline the process of learning, which I similarly refer to as the learning problem.

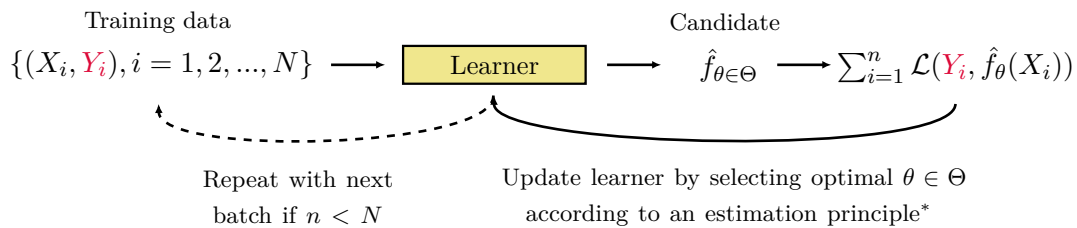


FIGURE 1.2: A schematic of process of learning from data. The presence or absence of $\mathbf{Y}_i$ distinguishes supervised from unsupervised problems. The dashed line, which indicates repetition, applies only to certain optimisation algorithms. *Examples of estimation principles include minimising loss in empirical risk minimisation and maximising posterior probability in Bayesian inference.

A learner (or model) is a parameterised function that specifies how input variables are combined to produce an output. Its specification defines a functional space, the set of all functions it can represent. Learners with more parameters typically span a larger functional space, which allows them to approximate a broader range of relationships. These learners are considered more complex because of their higher parameter count, and more flexible because their larger functional space enables adaptation to diverse data patterns. A learner whose functional space is sufficiently rich to represent any function is referred to as a universal approximator.

Learning problems are defined by the data available. Often, they involve input–target pairs (X, Y) , gathered by observing (or simulating) a process that potentially relates X to Y . In many cases, this process reflects an underlying functional relationship between inputs and outputs. However, observational data rarely capture this relationship perfectly and typically include an irreducible level of noise. A learner is updated using these data to obtain the optimal function within its functional space, to approximate the observed relationship as accurately as possible. This setup is referred to as

supervised learning, since the response is used to supervise the learning of the function. When only X is available, the problem is unsupervised learning. Supervised and unsupervised learning are the most common types, although other paradigms, such as semi-supervised, self-supervised, and reinforcement learning, also exist.

The learning process is illustrated in Figure 1.2. A learner produces a candidate output function based on its current state, $\hat{f}_{\theta \in \Theta}$, which may be initialised randomly or intentionally chosen. This output is compared to the observed data using an evaluation criterion; in a supervised setting, this is given by $\mathcal{L}(Y_i, \hat{f}(X_i))$. The choice of evaluation criterion is usually tied to an estimation principle, such as maximum likelihood estimation, Bayesian estimation, or numerical optimisation, each of which is associated with a subset of (possibly overlapping) optimisation methods. Within this pedagogical framing, the optimisation method acts as the teacher, guiding the learner on how to update its parameters in response to the evaluation, with the evaluation serving as the learning material. In some optimisation methods, this process is repeated iteratively, with only a batch of the data used at each step, as indicated by the dashed line in the figure.

Sections 2.8 and 2.10.2 describe specific learning problems in sentiment analysis, while Section 3.4 focuses on time series modelling. The remaining Sections provide additional domain context

1.7 The evolution of natural language processing

In this Section, I describe the evolution of NLP, contextualising the emergence of the current era of LLM. Chapter 2 provides a more comprehensive explanation, with a specific focus on the models applied in my Thesis.

Early NLP systems were dominated by rule-based methods, including lexicon- and semantics-based approaches grounded in syntax and handcrafted linguistic rules (Cambria and White, 2014). With the rise of statistical modelling, these gave way to approaches that relied on deliberately engineered linguistic features, such as part-of-speech tags, n-grams, and syntactic parses, fed into models like hidden Markov models, conditional random fields, and maximum entropy classifiers (Cambria and White, 2014). The subsequent emergence of bespoke DL architectures, which I expand more

on in Section 1.8.1, shifted the emphasis from manual feature engineering to representation learning (LeCun, Bengio, and Hinton, 2015). More recently, the introduction of the attention mechanism (Bahdanau, Cho, and Bengio, 2016) and, subsequently, transformers (Vaswani et al., 2017) has become the dominant paradigm, catalysing rapid advances in NLP research and applications.

Building on the transformer architecture, a new generation of pretrained Language Models (LMs) emerged, with BERT representing a landmark advance by achieving state-of-the-art results across numerous benchmarks (Devlin et al., 2019). This shifted the field away from bespoke architectures toward general-purpose models trained on massive corpora and fine-tuned for downstream tasks. Subsequent refinements further improved performance, most notably Robustly Optimized BERT Pretraining Approach (RoBERTa), which optimised BERT’s pretraining strategy (Liu et al., 2019).

Given the success of these models in English, multilingual extensions soon followed. Conneau and Lample (2019) introduced a cross-lingual pretraining approach that led to the Cross-lingual Language Model (XLM), while the methodological advances of the RoBERTa were combined with cross-lingual training to produce XLM-RoBERTa (XLM-R), a state-of-the-art multilingual masked LM (Conneau et al., 2020). Building on these foundations, domain-specific adaptations have been proposed, including several multilingual models trained specifically on Twitter data (Barbieri, Espinosa Anke, and Camacho-Collados, 2022; Nguyen, Vu, and Tuan Nguyen, 2020; DeLucia et al., 2022; Zhang et al., 2023).

With a view toward generality and task-agnostic models, there was a concurrent movement toward developing promptable LLMs. For instance, (Brown et al., 2020) introduced GPT-3, with 175 billion parameters. A pivotal development was introduced by OpenAI with Instruct-GPT (Ouyang et al., 2022). Building on earlier work, they applied Reinforcement Learning from Human Feedback (RLHF) to align the model’s responses with human preferences (Ziegler et al., 2020; Wu et al., 2021b). OpenAI was not alone in this endeavour; Bai et al. (2022) from Anthropic released comparable work in the same year.

The simplicity of prompting has facilitated the widespread integration of LLMs into day-to-day applications and is encouraging broader interdisciplinary adoption of NLP tools. However, the computational demands required to train such models remain a significant barrier, leading to their development primarily within industry. This trend

has resulted in a proliferation of closed-source hosted models, including ChatGPT, Claude, and Grok. Nevertheless, there have been notable open-source releases from industry as well-such as Meta’s LLaMA suite and Mistral’s models (Touvron et al., 2023b; Touvron et al., 2023a; Grattafiori et al., 2024; Jiang et al., 2023; Jiang et al., 2024).

Although LLMs are general performers, further gains can often be achieved via fine-tuning. Unfortunately, the resources required for full fine-tuning place is often impossible with consumer hardware. To address this, Hu et al. (2021a) proposed the Low-Rank Adaptation (LoRA) method. Dettmers et al. (2023) then improved the memory efficiency of this approach by first quantising the pretrained LM.

The emergence of large pretrained LMs provides new opportunities to retrospectively examine COVID-19 sentiment. While prior work has primarily leveraged BERT, broader applications of LLMs remain underexplored, in part because effective prompting of such models had not yet become prominent during the pandemic. Moreover, recent advances in fine-tuning methods offer additional motivation for their application in this domain.

1.8 Introduction to time series modelling

In Section 1.1.2, I outlined two main approaches to modelling infectious diseases, namely mechanistic models and time series models. In this Thesis, I focus on the latter because they learn associative relationships directly from historical data rather than relying on structural assumptions about behavioural influences on disease dynamics, which remain poorly understood.

My interest lies in whether tweet variables, as proxies for public response, contain predictive information about COVID-19 variables of interest, and conversely, whether these variables influence public response. Traditionally, such questions would have been addressed using classical statistical time series models.

However, as outlined in the next Section, the past decade has seen rapid advances in more powerful time series models. While many of the models I use were designed for forecasting, my primary aim is not to generate forecasts but to use out-of-sample predictive performance as a stronger measure of association than within-sample correlation. Developing a clearer understanding of how tweet variables relate to COVID-19

variables provides a foundation for the more effective use of social media data in outbreak response.

1.8.1 The evolution of time series modelling

This Section outlines the evolution of time series modelling and is divided into two parts. The first addresses classical statistical and ML models, and the second focuses on DL models. In Section 1.6.2, I introduced ML as a paradigm of learning, which is conceptually distinct from classical ML models.

Classical statistical and machine learning models

Classical statistical models such as Autoregressive (AR), Moving Average (MA), and autoregressive moving average (ARMA), and later Autoregressive Integrated Moving Average (ARIMA) models with their extensions, dominated much of the twentieth century (Yule, n.d.; Wold, 1938; Box and Jenkins, 1976; Engle, 1982; Bollerslev, 1986; Hamilton, 2020). These approaches typically assume stationarity and were primarily developed to capture structured, often linear, dependencies over time. They remain widely used as a starting point for time series analysis. Towards the end of the century, ML models began to appear in the literature, and in more recent decades they have matured into practical alternatives, particularly valuable for data characterised by strong non-linear or non-stationary behaviour (Kim et al., 2025).

Advances in computing power during the 1990s period made it feasible to train more computationally demanding algorithms and was a key driver of the shift toward ML methods. New methods such as AdaBoost, which implemented the theory of boosting proposed a few years earlier, the Least Absolute Shrinkage and Selection Operator (lasso), Support Vector Regression (SVR), and later random forests demonstrated strong predictive performance with fewer assumptions about the underlying data distribution (Freund, Schapire, et al., 1996; Schapire, 1990; Freund, 1995; Tibshirani, 1996; Vapnik, Golowich, and Smola, 1996; Breiman, 2001).

Initially, these successes were confined primarily to tabular datasets. When ML models were applied to time series, forecasting was typically reframed as supervised regression on a tabularised representation of the series, constructed from explicitly defined lag inputs. This allowed these models to relax many parametric assumptions, scale to large numbers of covariates, and capture moderate non-linearity. Nonetheless,

their performance depended heavily on manual feature engineering, and uncertainty was generally only approximated (e.g. random forests, SVR).

Boosting methods, in particular, gained attention for their ability to combine many weak learners into a single strong predictor. Since 2010, boosting techniques have advanced further with the development of gradient boosting frameworks such as XGBoost, LightGBM, and CatBoost (Friedman, 2001; Ke et al., 2017; Prokhorenkova et al., 2019). These achieved state-of-the-art results in electricity demand prediction, traffic flow forecasting, epidemiological case forecasting, and broader tabular-data competitions.

ML models are flexible learners but, in many cases, not inherently interpretable. Some, such as random forests, offer built-in proxies like variable importance measures, while the lasso’s sparsity promotes feature selection and often yields an interpretable linear structure. However, in most cases, interpretability is achieved post-hoc, often via model-agnostic tools such as Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) (Ribeiro, Singh, and Guestrin, 2016; Lundberg and Lee, 2017). While useful for feature-level explanations, these methods face important limitations in time series modelling. They break temporal correlation by treating each forecast step independently, typically explaining only one step of the forecast horizon, and describe influences on predictions rather than revealing the model’s internal sequential dynamics.

Deep Learning models

DL has its practical roots as far back as the 1950s, with the perceptron introduced by as a single-layer linear classifier, and while multilayer networks existed as early as the 1960s, training deep Multilayer Perceptrons (MLPs) was computationally infeasible (Rosenblatt, 1958; Ivakhnenko, 1971). The seminal development of the backpropagation algorithm and its application provided a practical and generalisable method for training multilayer networks (Werbos, 1981; Rumelhart, Hinton, and Williams, 1986). This made it feasible to optimise MLP architectures that, for the time, were considered deep, catalysing a new era in DL.

Soon after, Recurrent Neural Networks (RNNs) were proposed as a method for modelling sequential data (Jordan, 1986; Elman, 1990). They were trained using backpropagation through time, which extended the principles of backpropagation to

unfolded temporal sequences (Williams and Zipser, 1989). The Long Short-Term Memory (LSTM) cell provided a solution to the vanishing gradient problem that had limited the ability of earlier RNNs to learn long-term dependencies (Hochreiter and Schmidhuber, 1997). Convolutional Neural Networks (CNNs) also emerged, with LeNet-5 demonstrating state-of-the-art performance on handwritten digit recognition in the MNIST dataset (Lecun et al., 1998). Given their sequential structure, RNNs and LSTM became natural choices for time series modelling and NLP.

Initially, RNNs and LSTM saw limited use in time series modelling because they were difficult and resource-intensive to train. As computational resources improved in the 2000s to the early 2010s, they began to appear in time series applications. However, they were typically applied to one-step-ahead predictions, with longer forecast horizons generated recursively; an approach prone to compounding errors over time. They also faced computational challenges in handling long input sequences and extended forecast horizons.

The emergence of sequence-to-sequence models, and more specifically the encoder–decoder paradigm, marked a shift in how DL time series models were designed. This addressed the limitations of earlier RNN- and LSTM-based time series modelling pipelines. While the encoder–decoder paradigm was a breakthrough in NLP, its adoption in time series was more gradual. The core idea was that past observations could be encoded into a temporally rich embedding vector. This vector could then be used (decoded) to produce forecasts for the target horizon.

In contrast to NLP, the decoder in time series modelling does not need to be autoregressive (such as an RNN). In many applications, including those in this Thesis, the decoder is implemented as an MLP. This offers several advantages, including parallel predictions over the entire forecast horizon, reduced error accumulation relative to step-by-step recurrent decoding, and greater flexibility. In particular, when the encoder already captures temporal dependencies, an autoregressive prediction is often unnecessary. This allows the decoder to focus solely on mapping encoded features to output forecast values. The specific DL models employed in this Thesis are described in Section 3, but most follow this general encoder–decoder paradigm.

The modularity of the encoder–decoder paradigm allowed for rapid exploration of alternative encoder structures in time series models. In this Thesis, I used six models which adopt this approach across different architecture styles (Hochreiter and

Schmidhuber, 1997; Chang et al., 2017; Bai, Kolter, and Koltun, 2018; Lim et al., 2021; Olivares et al., 2023; Das et al., 2024). In addition, I also explored an alternative DL approach based on the Komolgorov-Arnold theorem (Liu et al., 2025).

Time series methods have broadened considerably in recent decades, with classical ML and DL models emerging to complement classical statistical approaches, particularly in addressing non-linear and non-stationary dynamics. The specific ML and DL models employed in this Thesis are introduced in Chapter 3 and they are applied in Chapter 7.

1.9 Thesis aims and structure

Based on the motivation presented in this Chapter, the aims of my Thesis are given below:

1. Construct a multilingual dataset of COVID-19-related tweets, encompassing general disease, public policy, and vaccination search terms, collected across eight European countries and languages over a two-year study period.
2. Using an annotated subset of this dataset, evaluate and identify the most accurate and computationally feasible NLP method for classifying sentiment polarity from tweets.
3. Apply the selected NLP method to characterise the temporal evolution of tweet volume and sentiment polarity.
4. Examine whether tweet variables are associated with epidemiological and public policy variables, considering both lagged and leading relationships.

Below is the structure of my Thesis which addressed these aims.

In the following two chapters, I outline the methods used in my analyses. Because the problems I addressed span two distinct domains, Chapter 2 focuses on NLP, while Chapter 3 covers time series modelling.

In Chapter 4, I describe the construction of my Twitter dataset, from initial downloading to the processing steps that made it suitable for analysis.

The following three chapters present the analyses and results. In Chapter 5, I evaluated multiple NLP models on an annotated subsample to identify the most effective for sentiment classification.

In Chapter 6, I applied this model to the full dataset, analysed sentiment-stratified public attention descriptively, and examined associations between these attention tweet, epidemiological, and public policy variables.

In Chapter 7 I extended the analysis of associations using formal time series models. I investigated two directions of temporal association: first, whether tweet volume or policy stringency improves predictions of epidemiological variables; and second, whether tweet volume is more strongly associated with public policy or with epidemiological variables.

Finally, Chapter 8 synthesises the findings and reflects on the epidemiological utility of public discourse on Twitter during the COVID-19 pandemic.

2 Sentiment analysis methods

2.1 Overview

In this Chapter, I introduce both lexicon-based and ML approaches to classify sentiment polarity from the dataset described in Chapter 5. I trace the evolution of ML for sentiment analysis, with particular emphasis on two pivotal innovations, namely attention mechanisms and transformers. I explain the distinction between encoder and decoder transformer architectures and, within decoders, describe the developments that enabled LLMs, highlighting the key innovation that makes fine-tuning such large models feasible.

I also formalise the two learning problems addressed in Chapter 5. The first involves training a classification head for an encoder model, where MLPs predict sentiment (Section 2.8). The second involves training two reduced-rank matrices for a quantised LLM (Section 2.10).

2.2 Introduction

NLP is the broad field associated with the computational processing of text. In this Thesis I focused on sentiment analysis, a specific task within the subfield of Natural Language Understanding. I considered two approaches to sentiment analysis, namely lexicon-based and ML-based.

In the next Section, I briefly discuss lexicon-based approaches, while the remainder of this Chapter is dedicated to ML-based methods. I outline the general process of sentiment classification from tweets and then focus on two seminal advancements, namely the introduction of the attention mechanism and the subsequent development of the transformer architecture. Finally, I present the models used in this Thesis that have emerged from transformers.

2.3 Lexicon approaches

Lexicon-based approaches provide a rule-based method for sentiment analysis. They rely on a sentiment lexicon, an inventory of words, phrases, or even characters/punctuation (including emojis), to which sentiment polarity scores are assigned. These scores are then used to determine the overall semantic orientation of an input.

The way the sentiment lexicon is constructed varies. Broadly, the two main approaches are corpus-based and dictionary-based. In a corpus-based approach, a small set of manually annotated seed words forms the basis of the lexicon. This lexicon is then expanded using co-occurrence statistics or semantic similarity, such as synonyms and antonyms (Gupta and Agrawal, 2020). In contrast, a dictionary-based approach starts with an initial set of seed words manually annotated with sentiment scores (by experts or crowdsourcing). The lexicon can be automatically extended by exploring semantic relationships in existing lexical databases such as WordNet Fellbaum, 1998. However, the exact methodology for lexicon construction can substantially impact the accuracy of these approaches (Taboada et al., 2011). In this Thesis, I applied a dictionary-based approach, described in more detail below.

A typical pipeline first tokenises the text, splitting the input into individual elements, such as words, phrases, or symbols, using a process called a tokeniser. Each token in the lexicon adjusts for modifiers (such as negators like “not” or intensifiers like “extremely”) and finally aggregates the weighted scores, often by summation or by applying rules that give extra weight to the sentiment of the final clause (especially in long texts). To adjust for varying input lengths, the aggregate score is usually normalised and mapped to a final sentiment category (positive, neutral, negative) using a predetermined threshold.

It is important to note that sentiment polarity and lexical associations can be highly domain- and language-specific. For example, the word “superspreader” may carry a negative connotation in epidemiology but might not be perceived negatively in general discourse. Social media adds further complexity, as it often features vernacular-specific language, polysemous terms, and domain-specific slang. To address these unique challenges, specialised sentiment lexicons have been developed for particular domains or languages.

One such resource is the VADER, a sentiment lexicon specifically designed for social media and microblog-like content (Hutto and Gilbert, 2014). Given its focus on the characteristics of online language, I used VADER as the dictionary-based approach in Chapter 5. It contains 7,520 terms, including emojis, with sentiment scores derived from ten independent human annotators. VADER also accounts for social media-specific features, such as the treatment of emojis as modifiers. The compound sentiment score is normalised using a saturation function to fall between -1 and 1, and sentiment is categorised using the following thresholds:

$$f(\text{score}) = \begin{cases} \text{negative} & \text{if } \text{score} \leq -0.05 \\ \text{neutral} & \text{if } -0.05 < \text{score} < 0.05 \\ \text{positive} & \text{if } \text{score} \geq 0.05 \end{cases}$$

Lexicon-based approaches are interpretable, as one can trace the contribution of each component to the final score. However, their static nature is a limitation; the lexicon is independent of the input and may not adapt well across different domains. VADER, while tailored for social media, does not dynamically adjust for sub-domains or contexts. Furthermore, in multilingual settings, lexicon translation is often non-trivial due to the lack of one-to-one mappings and may require further rule modification. In such cases, translating the input text can be more straightforward than translating the lexicon. Lexicon-based approaches are most effective when sentiment expressions are not domain-specific, a condition that is difficult to guarantee. The next Section examines ML approaches, which provide a more data-driven alternative.

2.4 Machine learning approaches

The initial process of classifying sentiment from text using ML is shown in Figure 2.1. While more modern alternatives, presented in later Sections of this Chapter, have since emerged, the figure provides a high-level overview in the context of sentiment classification. The same workflow applies to many other NLP tasks, with the final block labelled “Sentiment Classification” replaced more generally by “Output”.

Starting with a tweet¹, there are two main steps to generating sentiment, where each

¹Initial preprocessing may be required, for example replacing links and user references.

step may involve many sub-steps depending on the approach used. The first step is to convert a tweet into a feature vector. As suggested by its name, a feature vector, is a numeric vector of features extracted from the text that can be used as input for further downstream applications.

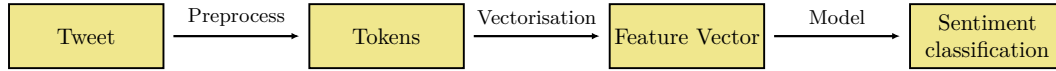


FIGURE 2.1: Traditional ML flowchart showing the steps required to generate a sentiment classification from a tweet.

To create a feature vector, the text must first be preprocessed. General preprocessing steps include converting the text to lowercase and removing punctuation. For tweets, additional steps are applied, such as replacing website links with `http` and removing unverified user mentions. Next, the text is broken into individual tokens by a tokeniser. Tokens are typically words or parts of words. These tokens may then be further processed through, for example, stemming or lemmatising, and removing stop words, but this depends on the exact vectoriser used. These steps are aimed at simplifying the words to a single form so that they can be grouped together. The result is then passed to a vectoriser, which converts the tokenised text into a numerical feature vector.

To better understand the process described above, below is a simple example using a Bag-of-words (BoW) vectoriser (Salton, Wong, and Yang, 1975). Consider the following corpus of two tweets:

Tweet 1: I am a student writing a thesis related to public health.

Tweet 2: My examiners are interested in public health.

The text preprocessing pipeline involves converting all text to lowercase, removing punctuation, tokenising with a word tokeniser, eliminating common stop words (e.g., am, a, to, my, are), and applying lemmatisation (e.g., writing $\rightarrow$ write). This results in the following tweet tokens:

Tweet 1 tokens = [i, student, write, thesis, relate, public, health]

Tweet 2 tokens = [examiner, interest, public, health]

The vocabulary ($\mathcal{V}$) includes all unique tokens from both tweets:

$\mathcal{V} = \{\text{i, student, write, thesis, relate, public, health, examiner, interest}\}$

Using this vocabulary, each tweet is represented as a count vector $\mathbf{v} \in \mathbb{R}^{|\mathcal{V}|}$, where each entry counts the occurrences of the corresponding token. Each feature vector is shown below:

Token	i	student	write	thesis	relate	public	health	examiner	interest
Tweet 1	1	1	1	1	1	1	1	0	0
Tweet 2	0	0	0	0	0	1	1	1	1

These feature vectors can then be used in downstream tasks. For example, if there are sentiment polarity labels for each tweet, a (weak) sentiment classifier can be trained. I provide a more detailed explanation of the models used for sentiment classification later, covering both encoder-based LMs and LLMs.

The purpose of the example above is to provide a simple and interpretable illustration of how a feature vector is constructed, using the BoW approach. This method is applied in Chapter 4 to vectorise text for a Twitter bot filtering task. Researchers first introduced BoW in NLP through the work of Salton, Wong, and Yang (1975). This was later extended by Salton and Buckley (1988), who proposed the Term Frequency–Inverse Document Frequency (TF-IDF) vectoriser. These methods, known as count-based vectorisers, generate sparse representations by encoding how frequently tokens appear in a document.

Count-based methods remained standard in NLP until a shift toward dense word embeddings, which more effectively encode semantic meaning. These dense representations operate in a lower-dimensional space compared to count vectors, which span the entire vocabulary, and can be used as learned feature vectors in downstream tasks. The Word2Vec model by Mikolov et al. (2013) played a pivotal role in this transition. In practice, this means that semantically similar words are located close together in the embedding space. A commonly cited example from Mikolov et al. (2013) shows that pairs such as king and queen, or boy and girl, appear near each other in such a space. In a COVID-19-specific example, words like quarantine, lockdown, and taste would likely cluster together. Later approaches in this category, such as GloVe and FastText, further advanced the use of dense word embeddings (Pennington, Socher, and Manning, 2014; Bojanowski et al., 2017).

However, while word embeddings marked a significant advancement, a key limitation is that they are static, assigning the same vector to a word regardless of its context.

This restricts their ability to capture syntactic structure and to disambiguate polysemous words. The introduction of the attention mechanism, and subsequently the transformer architecture, addressed these limitations by enabling models to generate contextual embeddings that dynamically adapt to the surrounding text.

2.5 Attention

Early NLP tasks typically used a feature vector (or hand-crafted features) as input and produced a single output, such as a class label or the next word in a sequence. In DL NLP approaches, RNNs and their extension using LSTM cells were commonly applied, particularly for classification tasks such as part-of-speech tagging or sentiment analysis, as well as for next-word prediction. Machine translation, however, posed a different challenge, as it required generating an entire output sequence from a given input sentence. Traditionally, this problem was addressed using rule-based systems or statistical machine translation techniques, such as those employing hidden Markov models (HMMs). In 2014, neural machine translation (NMT) emerged as a new paradigm, using neural networks to learn translation end-to-end.

Early neural machine translation models followed an encoder-decoder architecture (Cho et al., 2014b). This architecture, proposed separately and almost simultaneously by Cho et al. (2014a) and Sutskever, Vinyals, and Le (2014), paired two RNNs: one to encode the input sentence into a fixed-length feature vector, and another to decode it into a variable-length target sequence. Sutskever, Vinyals, and Le (2014) used LSTM cells, while Cho et al. (2014a) proposed Gated Recurrent Units (GRUs) as a simpler alternative. These networks were trained jointly, allowing both feature extraction and output generation to be optimised together. This approach marked a shift toward sequence-to-sequence modelling, a term coined in this context by Sutskever, Vinyals, and Le (2014). This Section omits detailed explanations of RNNs and LSTM and GRU cells to focus on Transformers and their resulting architectures; however, these components are described in more detail in the context of time series modelling in Sections 3.9.1 and 3.9.2.

An issue with the original encoder-decoder model was the difficulty of encoding an entire sentence into a fixed-length vector, especially when input sentences at test time were longer than those seen during training. To address this, Bahdanau, Cho, and Bengio (2016) proposed an extension in which the encoder produces a sequence of

bidirectional hidden-state (context) vectors, capturing information from both past and future tokens, unlike the unidirectional RNN or LSTM states in earlier models, which encoded only past context. At each decoding step, an alignment model generates a weighted combination of these vectors, selecting the information most relevant to predicting the next word. As the authors noted, “this implements a mechanism of attention in the decoder”, allowing the model to focus on different parts of the input as needed rather than relying on a single, fixed representation. Thus, while the original encoder–decoder reused the same fixed-length vector for every decoding step, the attention mechanism constructs a new context vector for each step, selectively incorporating only the information needed for the current prediction. This principle of attention would later form the basis of the Transformer architecture.

2.6 Transformers

The Transformer architecture, introduced by Vaswani et al. (2017), replaced the recurrent components of earlier encoder–decoder models with a fully attention-based design. In a recurrent encoder–decoder architecture, each step depends on the computations from the previous step, making the process inherently sequential. In contrast, the Transformer processes inputs and outputs using only attention mechanisms, allowing dependencies to be modelled in a global context.

To achieve an attention-only design, Transformers use a mechanism called scaled dot-product attention (a form of multiplicative attention), applied in two different ways within the network. For each token in the input sequence, this mechanism defines three vectors: a query vector, which encodes what information the token is seeking from others; a key vector, which encodes what information the token can provide; and a value vector, which contains the actual information to be shared. For a given token, its query is compared, via a dot product, with the keys of all tokens in the sequence to determine how well the information those tokens can offer matches what the current token is seeking. These scores are divided by the square root of the vector size to prevent excessively large values, then passed through a softmax to produce weights that sum to one. These weights are then multiplied by the corresponding value vector whose key vector was compared with the query to produce the weight. Notably, there will be a self-comparison where the token’s own key vector will be compared to its query, and this will be weighted in the final output. The resulting

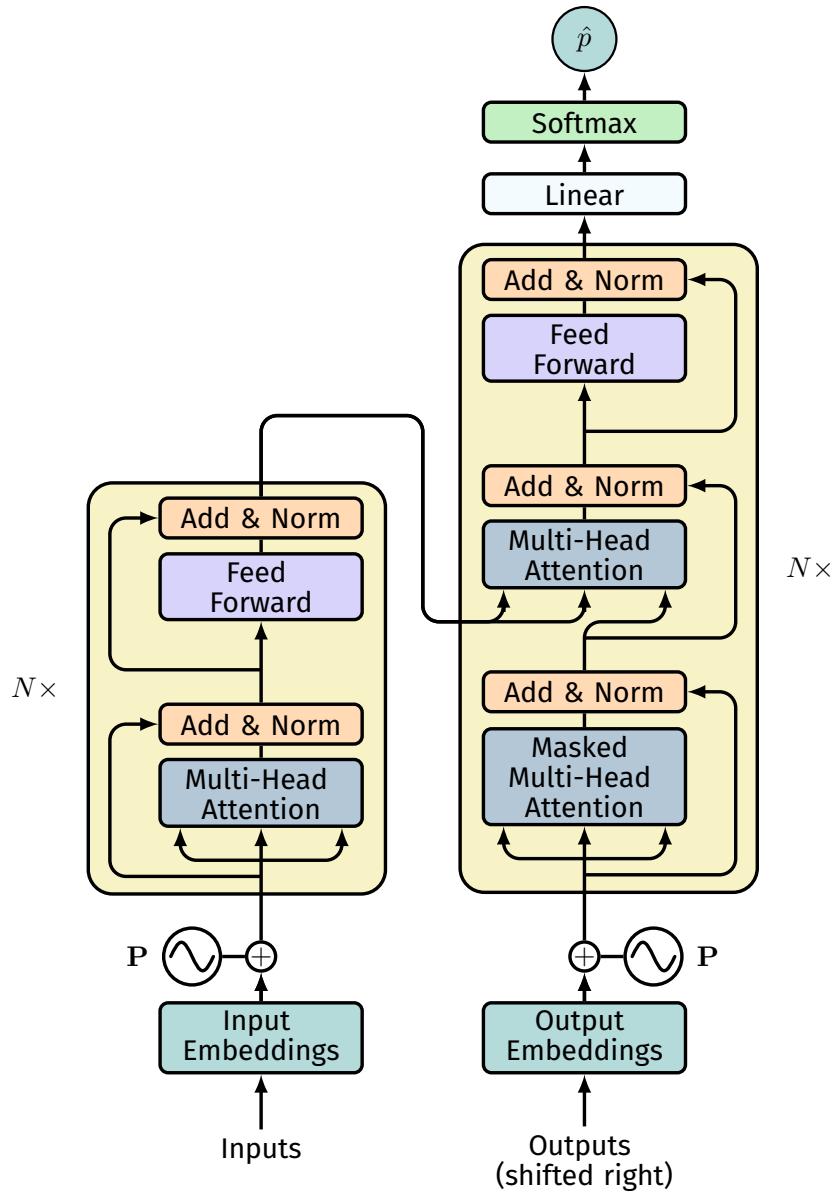


FIGURE 2.2: A digram of the transformer architecture. The encoder is shown on the left and the decoder is shown on the right. Source: Vaswani et al. (2017)

weighted sum is the output of the attention function. Therefore, the output of this attention can be thought of as a weighted combination that likely includes the token's own information and all past and future token values in the sequence that its query vector indicates are relevant.

Each of these components is learnable, with separate weight matrices used to generate the queries, keys, and values from the input embeddings. Through training on the target task, the model learns what information each token should seek (query), offer (key), and contribute (value). To capture multiple types of relationships in parallel,

the Transformer applies this mechanism across multiple scaled dot-product attention heads. The outputs of these heads are concatenated and combined to produce the final representation, a process known as multi-head attention.

The Transformer architecture, like earlier sequence-to-sequence architectures, includes both an encoder and a decoder block. The architecture is shown in Figure 2.2. For context throughout the explanation, it is helpful to consider a language translation task (the original purpose of the architecture). As a specific example consider a translation from English to French. The input tokens relate to an English sentence and the output tokens relate to the corresponding French translation.

Before entering the encoder, the input tokens are mapped to continuous vector representations via an embedding layer. As explained previously, this represents the vocabulary in a dense, learned space. The encoder's first attention mechanism is self-attention, in which each token attends to every other token in the same sequence. Because self-attention is position-agnostic, learnable positional encodings are added to the token embeddings to inject information about word order. This enables the model to consider token position when computing attention, producing contextualised embeddings in which each token's representation incorporates information from the entire sentence. Notably, by removing the sequential dependency of recurrent models, this approach allows for parallel computation across all tokens.

Once the output from the multi-head attention layer is generated, it undergoes further processing to produce the final output of the encoder block. At this stage, the Transformer applies architectural tools that also appear in other methods discussed in this Thesis, namely, skip connections and normalisation layers. First, a residual skip connection is used to add the original input to the output of the multi-head attention layer, followed by layer normalisation. Layer normalisation normalises feature values for each token independently, stabilising activations and accelerating convergence (Ba, Kiros, and Hinton, 2016). More generally, normalisation techniques in DL networks, such as batch, layer, or weight normalisation, are designed to support training in these ways. Skip connections, introduced by He et al. (2015) in the context of computer vision, are now widely adopted across deep learning. They work by adding the output from a previous layer to the output of a later layer. If the current layer is not useful, its output can be made very small, allowing the network to preserve earlier context. This mechanism enables the training of deeper networks,

as unhelpful layers can be effectively skipped without harming overall performance.

The normalised output is then fed to a position-wise MLP network, where an MLP is applied independently to each token. Intuitively, attention determines which context is relevant for the current timestep, and the MLP decides how to transform this into a useful feature. A second residual connection and layer normalisation wrap the MLP sublayer. This encoder block is repeated (six layers in the original Transformer), and the final encoder representations are passed to the decoder.

At each time step, the decoder starts by taking the output sequence available so far as input. During training, this target sequence is shifted one position to the right, with a special start token ([START]) placed at the beginning; this prevents the decoder from directly seeing the ground-truth word it is supposed to predict at the current position. Given that generation is autoregressive, the decoder uses masked self-attention (a causal mask) so each position can attend only to earlier positions in the output sequence. During training, this prevents the model from accessing future ground-truth tokens. However, during prediction, only the previously generated tokens are available, but the causal mask is still applied to remain consistent with the training procedure. This attention is applied multiple times and the block is known as masked multi-head attention.

In addition to self-attention, the decoder also includes encoder–decoder attention, the second type of attention in the architecture. This allows the decoder to attend to the output of the final encoder layer, enabling it to use both the context from the generated sequence and the original input. In the context of this Section’s translation example, this corresponds to determining which parts of the English sentence are most relevant for the current step in generating the French translation, similar to the attention used in earlier encoder–decoder architectures. A skip connection and normalisation layer are used to add the output of the masked multi-head attention to the attended encoder output. In this context, the skip connection allows the model to ignore the encoder context if the decoder’s current context is sufficient for generating the next token. This output is then passed through an MLP with a final skip and normalisation layer, similar to the encoder. This stack is repeated multiple times (six in the original architecture).

The output of the final decoder layer is passed through a linear layer that maps it

to a vector of logits over the model’s vocabulary². A softmax converts these logits into probabilities for every possible token in the vocabulary. At inference, a decoding strategy (e.g., argmax, beam search, or sampling) selects the next token; during training, the lagged true token is provided instead (known as teacher forcing). This process is repeated until the entire output sequence is generated.

Some models retain the complete encoder–decoder architecture, such as T5 and BART, which are used for sequence-to-sequence tasks involving output generation conditioned on an input (for example, translation (as discussed above) and summarisation) (Raffel et al., 2020; Lewis et al., 2020). Other approaches, however, split the Transformer into its constituent components. Encoder-only models are particularly effective for producing bidirectional contextual embeddings, which can be applied to a range of downstream tasks. In contrast, decoder-only models are well-suited to tasks requiring generative output, such as text generation. I use these constituent models in this Thesis. The following Sections examine each model class in more detail and take a broader view of their development; however, the architectural intuition remains the same as described in this Section.

2.7 Encoder language models

Following the introduction of Transformers, NLP underwent a significant shift. The focus changed from developing bespoke models for individual tasks to creating large-scale pretrained LMs that can be adapted to a variety of downstream tasks. These models, which produce contextual embeddings, exhibit a statistical understanding of language and aim to learn generalised language representations. A pivotal milestone in this shift was the development of BERT (Devlin et al., 2019), a large-scale pretrained Transformer encoder-only model designed for deep bidirectional contextual representation.

Built upon the encoder component of the original Transformer architecture, BERT emphasised large-scale pretraining to enhance the generalisability and transferability of its learned representations; it was trained on massive corpora (~3.3B words). To train on such a large dataset, BERT employed a novel self-supervised training

²In practice, the vocabulary is fixed before training, often by collecting the most frequent tokens from a large corpus and mapping each to an integer ID. Modern implementations often use subword tokenisation methods, such as Byte-Pair Encoding (BPE), to keep the vocabulary size manageable while still representing rare words.

approach that combined Masked Language Modelling (MLM) and next-sentence prediction (NSP). MLM involves randomly replacing a portion of tokens with the special token [MASK] and training the model to predict these masked tokens based on their surrounding context, enabling it to learn bidirectional relationships. NSP involves presenting the model with pairs of sentences, where in half of the cases the second sentence follows the first one in the original text (labelled *IsNext*). In the other half, it is a random sentence from the corpus (labelled *NotNext*). The model is trained to predict whether the second sentence logically follows the first, encouraging it to learn sentence-level coherence. Together, these objectives enabled BERT, the seminal encoder LM, to develop rich contextual representations at both the word and sentence levels, improving performance across a wide range of downstream NLP tasks.

With the transformer architecture firmly established, subsequent work shifted towards training larger and more effective models. One key development was RoBERTa (Liu et al., 2019), an enhanced version of the original BERT model that incorporated a more robust pretraining scheme and additional data. RoBERTa introduced several key modifications: dynamic masking, in which masking patterns are generated at runtime rather than during data preprocessing; the removal of the NSP objective, replaced by training on full sequences of contiguous sentences up to 512 tokens; larger mini-batch sizes³; extended training time; and the adoption of a byte-level BPE tokeniser instead of a character-level approach. Interestingly, the shift to byte-level tokenisation did not lead to performance gains in their experiments.

Along with the original BERT models, Devlin et al. (2019) also released a multilingual version called mBERT. This model was trained on a mix of Wikipedia articles from 104 languages using a shared WordPiece vocabulary. Although it was not explicitly designed to align languages, mBERT still showed a strong ability to transfer knowledge across languages in various tasks (Wu and Dredze, 2019). Building on this, Conneau and Lample (2019) introduced Translation Language Modelling (TLM), a new training method that explicitly encourages cross-lingual alignment. TLM extends MLM by concatenating parallel sentences from two languages and randomly masking tokens across both, allowing the model being trained to exploit information from both languages when one alone is insufficient. To make this possible, position embeddings restart at the boundary between the two sentences. This helps the model

³Mini-batches are explained in Section 2.8.

align languages within a shared embedding space. However, TLM requires parallel corpora, meaning it relies on labelled pairs of sentences which is a limitation.

After the release of RoBERTa, Conneau et al. (2019) improved on XLM and introduced an improved version called XLM-R. They observed that both XLM and multilingual BERT (mBERT) had been under-tuned in prior experiments and demonstrated that, under the right training conditions, using only MLM was sufficient to achieve strong cross-lingual performance. This approach allowed them to extend coverage to 100 languages. Leveraging 2.5 TB of newly curated, clean CommonCrawl data across these languages, they trained XLM-R, which achieved significant improvements over previous state-of-the-art results on a variety of benchmark tasks. Like BERT, XLM-R is a large-scale pretrained cross-lingual Transformer encoder model designed to be fine-tuned for diverse downstream applications.

Pretrained encoder LMs can produce general-purpose language representations, but domain-specific applications often require further adaptation. Certain domains have unique vocabulary, syntax, and semantic associations that general models are not optimised to handle. This is analogous to developing domain-specific lexicons, as discussed in Section 2.3. For example, Twitter data is characterised by informal language, abbreviations, and platform-specific vernacular. Consequently, in this Thesis, I used pretrained models that have been domain-adapted for Twitter. Broadly, there are two primary approaches to domain adaptation, and I applied one of each. The specific models I used are described below, with further details on size, language coverage, and exact versions used provided in Chapter 5.

The first approach fine-tunes an existing pretrained model on domain-specific data, updating its embedding space to reflect the new linguistic distribution. This fine-tuning can be understood as shifting the embedding space toward the domain language. For instance, Barbieri, Espinosa Anke, and Camacho-Collados (2022) employed this method to develop XLM-Twitter (XLM-T), a multilingual variant of XLM-R fine-tuned on millions of tweets across more than thirty languages.

The second approach is to train a model from scratch on domain-specific text. An example of this is Bernice, a RoBERTa-based model trained using the MLM objective on a large corpus of tweets (DeLucia et al., 2022). In addition to domain-specific pretraining, the authors constructed a custom Twitter-specific tokeniser, allowing

the model to handle platform-specific linguistic features more effectively by tailoring subword segmentation to the Twitter domain.

Twitter-specific encoders, such as XLM-T and Bernice, generate embedding vectors that capture meaningful information about each input tweet. These embeddings can then be used for downstream tasks, reducing some NLP tasks to training only a downstream model on the generated embeddings. This process is illustrated for classifying sentiment polarity in Figure 2.3, which is almost identical to the traditional pipeline shown in Figure 2.1, except that an encoder replaces the vectoriser.

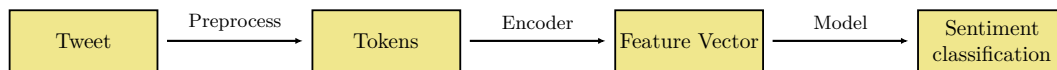


FIGURE 2.3: Flowchart showing the steps required to generate a sentiment classification from a tweet using an encoder LM to generate the feature vector.

For classification tasks, the model typically used on top of the encoder embeddings is an MLP. The encoder LM remains frozen, while only the MLP, often referred to as the classification head, is fine-tuned. This configuration is illustrated in Figure 2.4.

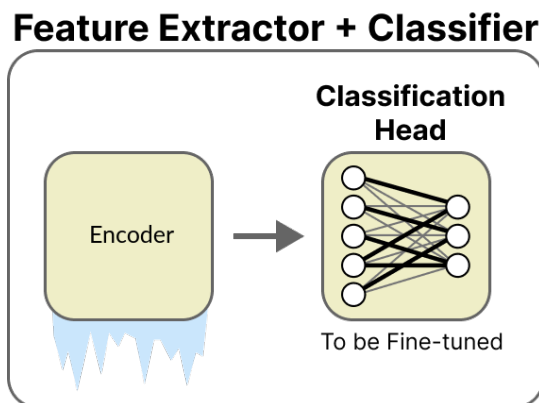


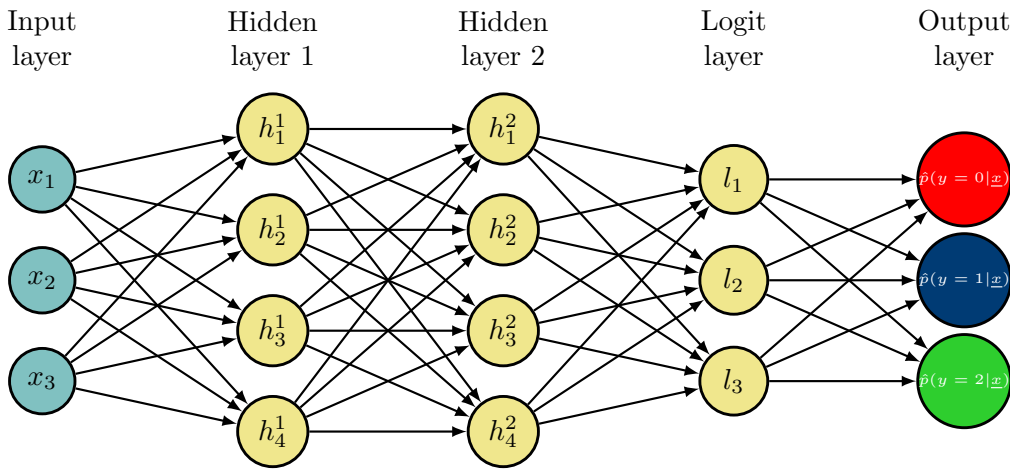
FIGURE 2.4: Illustration of a frozen encoder with a classification head to be fine-tuned for a downstream task. Source (Edited): Alammam and Grootendorst (2024)

2.8 Classification heads

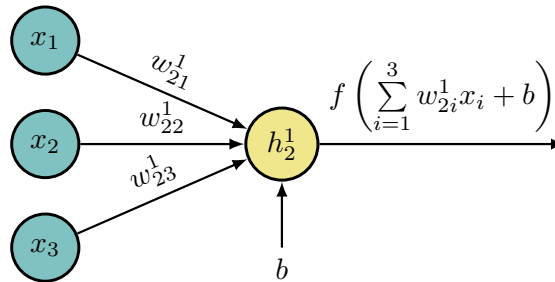
The first learning problem in this Chapter is training a classification head for sentiment analysis. Using labelled tweets, the model is trained to estimate the probability that each tweet expresses a negative, neutral, or positive sentiment. These probabilities are then used to assign the final sentiment label. The process, including how

the MLP classification head generates these probabilities, is described in more detail below.

Figure 2.5a illustrates an MLP designed for this specific classification problem. The input layer receives an embedding vector produced by the encoder LM, with each tweet represented by its own vector. Each x_i corresponds to an element in the embedding vector for a given tweet; in the example, the embedding vector has a dimension of three, but in reality, it is much larger. The input then passes through two hidden layers, which classify useful features. Each hidden state h is a latent feature that is a function of the previous layer's outputs and, by extension, of the original input. The model generates these features through a linear transformation of the outputs from the previous layer, combined with a bias term. To introduce non-linearity, it applies an activation function f to the result. The weights and biases are the learnable parameters of the model. Figure 2.5b shows this process for a single neuron.



(A) A complete MLP.



(B) A single neuron from the MLP above.

FIGURE 2.5: Combined visualisation of an MLP neural network used for a multi-class classification problem and a single neuron from it.

To generate class probabilities, the model passes the logits from the final hidden layer through a softmax activation function, introduced earlier in the context of creating

attention weights from the key and query vectors. In binary classification, a sigmoid function is used instead. The softmax function normalises the logits into a discrete probability distribution over the input classes and is defined for a three-class classification problem in Equation 2.1. The index i corresponds to the negative, neutral, and positive classes, respectively.

$$\text{softmax}(\hat{p}(y_i)) = \frac{e^{l_i}}{\sum_{j=1}^3 e^{l_j}}, \quad i = 1, 2, 3 \quad (2.1)$$

In this Thesis, I applied the learning paradigm outlined in Section 1.6.2 to train the model's weights and biases. To optimise these parameters, I used the Adam optimiser (described in more detail in Section 3.8.1), a gradient-based optimisation algorithm. I used categorical cross-entropy loss as the training objective, which quantifies the discrepancy between the predicted probabilities and the true class labels. This loss function is defined as follows:

$$\mathcal{L}(\mathbf{y}, \hat{p}(\mathbf{y})) = - \sum_{i=1}^3 y_i \log(\hat{p}(y_i)) \quad (2.2)$$

Assuming the labels are one-hot encoded, the loss simplifies to the negative log of the predicted probability for the true label, $\log(\hat{y}_{true})$. This means that the higher the predicted probability of the correct label, the lower the loss, and vice versa. The loss is computed only for the true label of each tweet. Using the chain rule, the error is then backpropagated through the network, and an optimiser updates the learnable parameters by following the gradient of the loss surface.

Training is typically performed on smaller subsets of the dataset, known as mini-batches, rather than on the entire dataset at once (full-batch training). Using mini-batches reduces computation time, improves convergence stability, and efficiently utilises hardware such as GPUs. This update process is repeated across the entire training set multiple times, with each complete pass referred to as an epoch. After training is complete, the model outputs the class with the highest predicted probability as its prediction, formally expressed as: $\hat{y} = \arg \max_i \text{softmax}(\hat{p}(y_i))$.

Some parameters, known as hyperparameters, are not learned during training and must be specified beforehand. These can be model-specific, for example, the choice of activation function, or related to the training process itself, such as batch size.

To select optimal hyperparameters, I used k -fold cross-validation. An example of this is shown in Figure 2.6. First, a search space is defined over the hyperparameters. The dataset is then divided into k splits. For each hyperparameter configuration, the model is trained k times, with one split held out for validation in each run while training occurs on the remaining splits. The model’s performance is evaluated on the held-out split, and the results are averaged over the k runs. This average serves as a proxy for the model’s generalisation ability. While I have introduced k -fold cross-validation in the context of hyperparameter tuning, it can also be used as a general performance measure when no independent test set is available, as is frequently the case with limited data.



FIGURE 2.6: Illustration of 5-fold cross-validation.

This process is repeated for multiple configurations, and each individual iteration of the process is referred to as a trial. The specific configurations used during cross-validation depend on the chosen method for hyperparameter sampling. After all runs are complete, the configuration yielding the lowest cross-validation error is selected. The number of folds n can vary, with an extreme case being leave-one-out cross-validation, where n equals the number of data points. Following this, the model is typically retrained on the entire training set and then tested on a separately held-out test set. Keeping this test set separate ensures an unbiased estimate of the model’s out-of-sample generalisation.

2.9 Large language models

While BERT was the first large-scale encoder-only LM and inspired an entire line of research on bidirectional contextual representations, a parallel development emerged in decoder-only architectures. In a similar role, GPT-1, introduced by Radford and Narasimhan (2018), served as a seminal model that laid the foundation for generative pretraining approaches.

In GPT-1, the goal was to create a single pretrained model that could be adapted to a variety of supervised NLP tasks. To achieve this, the authors first trained a large-scale decoder-only model on a substantial corpus of unlabelled text using the Causal Language Modelling (CLM) objective. As discussed in Section 2.6, CLM employs a causal mask to restrict the decoder’s attention to previously generated tokens, enabling autoregressive generation in which each token is predicted solely from prior context. After this general pretraining stage, the model was adapted to specific tasks such as (multiple choice) question answering or sentence similarity through supervised fine-tuning (SFT). For each task, the structured inputs were converted into a single token sequence, and a linear-plus-softmax layer was trained on top of the decoder’s final hidden states, analogous to the classification heads used in encoder-based models, albeit with a different approach to generating the feature vector. This approach required separate fine-tuning for each task, as the model could not switch tasks dynamically at inference time.

GPT-2 marked a shift away from per-task fine-tuning (Radford et al., 2019). Instead of training on specific labelled tasks, it was trained purely with causal language modelling (CLM) on a massive, diverse corpus. Although not explicitly optimised for instruction following, GPT-2 exhibited an emergent ability to infer the intended task from the text prefix. In effect, the prompt served as a task specification, enabling the same set of parameters to perform a wide range of tasks without further parameter updates.

GPT-3 amplified this effect through a dramatic increase in scale, 175 billion parameters trained on hundreds of billions of tokens, combined with the same CLM objective over a broad and diverse dataset (Brown et al., 2020). This diversity, rather than task-specific fine-tuning, enabled the model to infer and perform new tasks from context alone. The authors demonstrated strong performance in zero-shot settings, as

well as further gains in few-shot settings by including a small number of input–output examples in the prompt. Each successive GPT model represented roughly an order-of-magnitude increase in parameter count over its predecessor. Together, these advances shifted the adaptation strategy from supervised fine-tuning toward controlling model behaviour through carefully constructed prompts.

While these works showed that early LLMs had the capacity to generalise across a variety of tasks, there were still limitations to their utility. At times, the output of these models was undesirable; without constraints, they could generate biased, toxic, untruthful, or nonsensical responses. Ouyang et al. (2022) argued that this misalignment was due to a mismatch between the training objective (CLM on a broad corpus) and the goal of following “the user’s instructions helpfully and safely”. To address this, they introduced InstructGPT, which reintroduced SFT in a different form. Instead of labelled task datasets, it used curated prompt–completion pairs spanning many instruction types. During training, the loss was applied only to completion tokens, with prompt tokens loss-masked, so the model learned to treat the prompt purely as context. In effect, the model’s output was conditioned on the input prompt and scored solely on how it responded using this context. This was followed by reward modelling and RLHF, explicitly aligning generations with human preferences (Stiennon et al., 2022; Christiano et al., 2023). These steps are explained in more detail:

1. Supervised fine-tuning: A team of screened annotators wrote responses to a wide range of prompts previously submitted to InstructGPT by real users via OpenAI’s online platform. Annotators also contributed their prompts and completions. This supervised dataset was then used to fine-tune GPT-3.
2. Reward modelling: The fine-tuned model then generated multiple candidate responses for a wider set of prompts. Human annotators ranked these responses according to their preference. These rankings were used to train a reward model that allocates a reward based on human preference.
3. Reinforcement learning: Using reinforcement learning, the reward model was used to guide further updates to the fine-tuned GPT-3 model, thereby improving alignment between the model’s output and human expectations.

Bai et al. (2022) extended this approach by introducing multi-objective reward models

that optimised not only for helpfulness but also for reduced harmfulness. These efforts marked a key transition toward instruction-following language models, often referred to as aligned LLMs, which I refer to simply as LLMs in this Thesis. I use the term foundation model or base model for the underlying pretrained model (e.g., GPT-3) from which instruction-tuned variants are derived. However, terminology across domains, and even within the NLP community, varies and lacks strict consensus.

The compute and cost required to train these models have results in LLMs primarily being developed by industry. In some cases, these models are closed-source, meaning that the underlying model is not publicly available and can only be accessed through an API or web interface. In other cases these models are open-source with the model code and weights available. These models are shared through HuggingFace, a version control platform for model sharing analogous to GitHub (or other git platforms) (Wolf et al., 2020). In this Thesis, I predominantly used open-source LLMs, with a single closed-source model included for comparison. As with the encoder models, the exact LLMs used are described in Chapter 5, in the context of their application. However, all LLMs used were developed following the general methodology outlined in this Section.

The ability to prompt LLMs has created a new workflow in NLP. Rather than explicitly updating a model’s parameters, a prompt is used as context to switch between tasks. This process is illustrated in Figure 2.7. Both the prompt and the tweet input are tokenised and passed to the LLM, which processes them with its decoder to generate a free-text output. This output may then need to be parsed to classify the sentiment classification. In contrast to previous approaches, no explicit task-specific training is required. However, while prompting allows LLMs to perform new tasks without modifying their parameters, there are situations where this approach is insufficient. In these cases, the LLM may need to be fine-tuned.

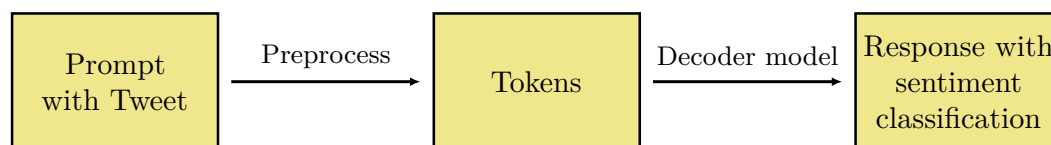


FIGURE 2.7: Flowchart showing the steps required to generate a sentiment classification from a tweet using an LLM.

2.10 Fine-tuning large language models

2.10.1 Quantised low rank adaptation

The strategy for adapting LLMs through fine-tuning has evolved, as illustrated in the previous Section. Earlier decoder-only models such as GPT-1 followed a more conventional supervised fine-tuning paradigm, training a classification head on labelled datasets, similar to the encoder fine-tuning process described in Section 2.8. However, modern instruction-following LLMs are optimised for producing natural-language responses to prompts rather than for direct use with classification heads. As a result, classification tasks are typically reformulated as text generation problems, where labels are expressed as text and the model’s output is constrained or post-processed to map back to a class.

Instruction-following LLMs are usually adapted in one of two ways, either by tuning the prompt (no parameter updates) or by updating the model’s weights. Prompt tuning, also known as prompt engineering, is illustrated in Chapter 5 with the specific prompts used in this Thesis. Updating the weights corresponds to supervised fine-tuning on curated prompt–completion pairs. However, because modern LLMs are large, full fine-tuning (updating all parameters) is computationally expensive and often infeasible on standard consumer hardware. To address these computational and memory constraints, Hu et al. (2021a) introduced LoRA, which learns small low-rank adapter matrices while keeping the base model frozen.

Their approach builds on earlier findings suggesting that, although LLMs have extremely high-dimensional parameter spaces, their effective parameters lie on a much lower-dimensional manifold (Li et al., 2018; Aghajanyan, Zettlemoyer, and Gupta, 2020). In other words, the subspace capturing the essential structure for strong performance is of lower dimension; in this sense, the LLM has a low intrinsic dimension.

LoRA leverages this insight by replacing full weight updates with small, trainable low-rank matrices added to selected weight matrices. These adapters define a compact subspace for adaptation, allowing the model to learn task-specific behaviour while keeping the original weights fixed. After training, the low-rank updates can be merged back into the base model, providing a parameter-efficient method that maintains performance. Conceptually, this is like steering the model’s behaviour by adjusting a few influential internal components rather than reconfiguring the entire system.

To further improve fine-tuning efficiency and reduce memory usage, Dettmers et al. (2023) proposed Quantised Low-Rank Adaptation (QLoRA). This approach quantises the pretrained LLM’s weights to 8-bit or 4-bit precision for storage, while keeping the QLoRA adapter weights in 16-bit precision to preserve the quality of updates. Despite the lower storage precision of the base model, all optimisation (including forward and backward passes) are still performed in 16-bit precision. The 4-bit variant uses a custom NormalFloat data type for optimal information storage. Additional procedural optimisations include double quantisation, which compresses the quantisation constants themselves to save further memory, and paged optimisers, which automatically offload data to CPU RAM when GPU memory is full to avoid memory errors.

While QLoRA is mainly similar to LoRA in how the low-rank updates are applied, it differs in the scope of layers updated. In the original LoRA formulation, low-rank adapters are typically inserted only into selected weight matrices, such as the query and value projection matrices in the attention layers, to limit memory usage. QLoRA, by contrast, can update all linear layers in the model. This is possible because the base model weights are stored in low precision, which frees enough memory to place adapters in every layer without exceeding hardware limits. Dettmers et al. (2023) showed that updating all linear layers yields stronger performance than the selective placement used in standard LoRA.

2.10.2 The quantised low rank adaptation learning problem

As outlined in the previous Section, QLoRA first quantises the LLM and then, following the LoRA approach, initialises low-rank weight matrices to enable task-specific adaptation. The second learning problem introduced in this Chapter is how to optimally learn these matrices in order to improve the performance of LLMs on tweet sentiment classification.

Formally, let the pretrained weight matrix be defined as $W \in \mathbb{R}^{d_{in} \times d_{out}}$. Two learnable matrices are introduced: $A \in \mathbb{R}^{d_{in} \times r}$ and $B \in \mathbb{R}^{r \times d_{out}}$. The first matrix, A , is typically initialised with a normal distribution centred at zero and projects the input down to a lower rank r . The rank r is a tunable hyperparameter that sets the dimensionality of the low-rank subspace in which the weight update lives, reflecting the assumption that

a smaller subspace is needed to adapt the model. The second matrix, B , is initialised as zeros and then projects the output of A back up to the original dimension.

To control the influence of the new, low-rank weights, an additional hyperparameter α is introduced; the scaling factor $\frac{\alpha}{r}$ determines the relative contribution of BA to the original weight matrix. During training, the model effectively uses the following: $W' = W + \frac{\alpha}{r}BA$, where this update is applied during each optimisation step. Because B is initialised as zero, the model initially matches the original pretrained model, which serves as a starting baseline.

Note that this approach implies that a fine-tuned model can be uniquely identified by $\frac{\alpha}{r}BA$, assuming the original weights are available. Consequently, rather than sharing entire fine-tuned models, which demand substantial storage, it is sufficient to share the adapters, which can be combined with the same base model to reproduce task-specific behaviour.

The input to this problem are prompt-completion pairs, where a prompt along with a tweet and a corresponding sentiment label is provided. From this there are two possible objectives. The first approach corresponds to the standard CLM task, where the loss is calculated over both the prompt and the completion. This is useful when the model needs to learn from the prompt, for example to capture specialised instructions or domain-specific vocabulary, while also learning how best to respond. The second approach fine-tunes the model for the completion only, as in InstructGPT. As explained earlier, this approach masks the input prompt when calculating the loss, so the model is evaluated only on how well it generates the expected response for a given prompt. This method is appropriate when the goal is to teach the model how to respond to instructions, assuming it already understands the prompt. The key difference in both these cases is that the pretrained weight matrix W is frozen, and only the low-rank matrices A and B are updated during training.

The learning problem reduces to minimising the categorical cross-entropy loss for each generated token, whether this is computed over the entire input or only over the completion tokens, where the total number of possible output values is equal to the vocabulary size V . For an input sequence with T tokens, consider the process over a complete sentence (to illustrate the effect of causal masking). For each token position t , the conditional probability of the t^{th} token taking the value of the i^{th} vocabulary entry is computed given the preceding tokens, the current low-rank weights BA , and

the frozen pretrained weights W^{frozen} . To make explicit that the pretrained weights are frozen, the parameter space is defined as $\Phi = (A, B, W^{\text{frozen}})$.

Assuming the true token is one-hot encoded, this probability is written as $\hat{p}_{\Phi}(y_{ti} \mid x_{<t})$, where the distribution over all tokens in the vocabulary is computed using a softmax activation. The cross-entropy loss for this problem is given in Equation 2.3. As noted in Section 2.8, given the one-hot encoding, this expression reduces to evaluating the probability the model assigns to the actual token, $\hat{p}_{\Phi}(y_{t,\text{true}} \mid x_{<t})$, as shown in Equation 2.4. Using this loss, QLoRA fine-tuning reduces to estimating the low-rank weight matrices that maximise the probability of predicting the correct token at each step.

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{p}}_{\Theta}) = -\frac{1}{T} \sum_{t=1}^T \sum_{i=1}^V y_{ti} \log(\hat{p}_{\Theta}(y_{ti} \mid x_{<t})) \quad (2.3)$$

$$= -\frac{1}{T} \sum_{t=1}^T \log(\hat{p}_{\Theta}(y_{t,\text{true}} \mid x_{<t})) \quad (2.4)$$

2.11 Summary

In this Chapter, I presented both lexicon-based and ML approaches to sentiment extraction. Within the class of ML models, I explained how they can be fine-tuned for the sentiment analysis task addressed in this Thesis. Models from both approaches are employed in Chapter 5, where I provide details on the specific models and the data used for fine-tuning.

3 Time series modelling

3.1 Overview

This Chapter is divided into two parts. In the first part, I focus on time series modelling in general. I begin by explaining its key components and discussing what makes it distinct. I then outline central challenges, including forecasting beyond a single step and strategies for hyperparameter tuning, which rely on appropriate validation approaches, before examining how evaluation metrics are used to assess performance. I conclude this part by formalising the overarching learning problem addressed in Chapter 7, considering two model classes, namely ML and DL.

In the second part, I turn to the specific models within these two classes. I describe their defining characteristics and emphasise the broader differences in how they are trained, thereby providing the foundation for their application in the subsequent analysis.

3.2 Introduction

The goals of time series analysis are similar to those of other regression problems, in that observed data are used either to explain events or to predict future outcomes. What distinguishes time series, however, is their ordered nature. A time series consists of observations recorded over time, usually at equal intervals, with the length of these intervals referred to as the period of the series. Such data violate the common assumption of independent and identically distributed observations, since values are typically autocorrelated. This means that current or future values depend on past ones. A common first step in addressing this dependence is to use historical values of the response variable as inputs. In this setting, the problem becomes autoregressive, as future values are predicted from past values of the same variable¹. A key challenge in time series modelling is to determine the lags at which such dependencies operate.

¹This approach parallels the CLM problem discussed in Section 2.9.

In this Thesis, the focus is on non-stationary time series, and I therefore considered two broad classes of models rather than classical statistical approaches, namely classical ML and DL. As explained in Section 1.6.1, DL is a subfield of ML; for brevity, I will omit the qualifier classical when referring to ML models, but note that this differs from the broader field. These models are not constrained by assumptions of stationarity and, rather than relying on manual transformations such as differencing, can instead learn such patterns directly from non-stationary data, as discussed in Section 1.8.1. Similar to developments in NLP, DL models in particular have been increasingly adopted for time series because of their ability to capture complex temporal dependencies. The capacity of both classes of models to learn directly from the data makes them well-suited to highly non-linear series (Makridakis, Spiliotis, and Assimakopoulos, 2022; Li and Law, 2024; Kim et al., 2025).

In the ML model class, I used a linear model, two tree-based models, and a kernel-based model. Within the DL category, the models I used fall into four categories: MLPs, CNNs, RNNs, and transformer-based architectures. In addition, I included a Komolgorov-Arnold Network (KAN), which falls outside conventional neural architectures. The specific models used in each class are shown in Figure 3.1.

In this Thesis, I used time series models to assess the association between tweet variables and COVID-19 variables of interest. I evaluate out-of-sample forecasting accuracy with and without exogenous variables, treating this as a more informative test of association than the in-sample correlation measures employed in Chapter 6. Although these models are inherently predictive and involve generating forecasts, the objective is not forecasting accuracy but rather an understanding of the potential temporal relationships between variables. In this Chapter, I present time series models and highlight the components related to forecasting, while keeping in mind that the focus is on their use for examining associations rather than for prediction alone.

3.3 Modelling approaches

When applying time series models, a sequence of predictions is typically generated over a specified number of steps, known as the forecast horizon. Broadly, when the forecast horizon extends beyond a single step, two main approaches can be used, namely the direct method and the recursive method. The columns in Figure 3.2 illustrate these methods. In each example, the task is to forecast values three time

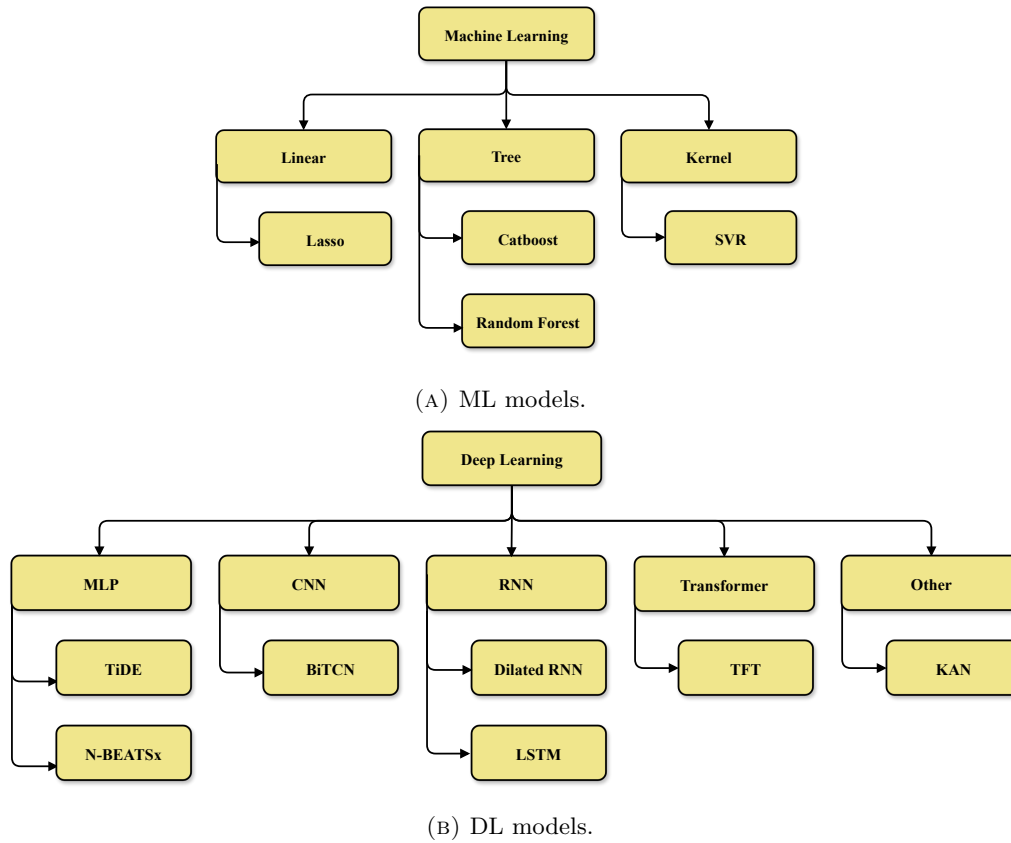


FIGURE 3.1: Hierarchy of (A) ML and (B) DL models used, organised by model class.

steps ahead using five observations (the previous four days and the current day). The key difference between the two approaches, as shown in each column, is in their treatment of inputs. Recursive forecasting reuses earlier predictions for subsequent steps, while direct forecasting relies only on observed data.

In addition, ML and DL models utilise historical values in different ways. The first row of Figure 3.2 illustrates each approach for ML models. These models can process sequences by replicating the same operation across multiple time steps. During model specification, the length of historical data to incorporate must be manually specified or determined through a feature selection process. In the examples shown, the models predict a forecast horizon of three, using today's value (t) and the value from two days ago ($t-2$).

Figure 3.2a illustrates recursive forecasting for ML models, in which a single model is trained and its predictions are reused as inputs when required. The orange blocks in the figure indicate reused predicted values. For example, the prediction for time $t+2$ uses the prediction from $t+1$ and the actual value from $t-1$ as inputs. In contrast,

for direct ML forecasting, shown in Figure 3.2b, a separate model is trained for each forecast step. Specifically, three models are trained, corresponding to $t+1$, $t+2$, and $t+3$, each of which uses the same input.

As discussed earlier, many DL models are capable of performing sequence-to-sequence modelling; that is, they can ingest and return entire sequences. These models take as input a specified number of past observations, which I refer to as the input sequence length. In the examples shown, this sequence length is five.

Figure 3.2c illustrates recursive DL forecasting. Here, a single model is trained with a forecast horizon of one, producing a single prediction at each time step. The input sequence is then updated to include this prediction, while the earliest observation is discarded to maintain a consistent input length. In contrast, direct DL forecasting, shown in Figure 3.2d, uses a fixed-length input sequence and directly outputs forecasts for the entire horizon.

Exogenous variables, which are predictors external to the response variable, can be included in both types of models. I use the term exogenous to distinguish them from AR terms. There are three classes of exogenous variables: historic, referring to values at or before the current time step; future, referring to values beyond the current time step; and static, which remain constant over time.

Incorporating exogenous variables into recursive forecasting can be challenging, as some model specifications require future values of these variables. This necessitates either training a separate model to forecast the exogenous inputs or employing multivariate models. I favoured direct forecasting in this Thesis because the models used are predominantly univariate and no additional forecasting models were available for the exogenous variables.

3.4 Time series learning problem

In line with the direct forecasting approach, the general form of the ML and DL models I used is defined below.

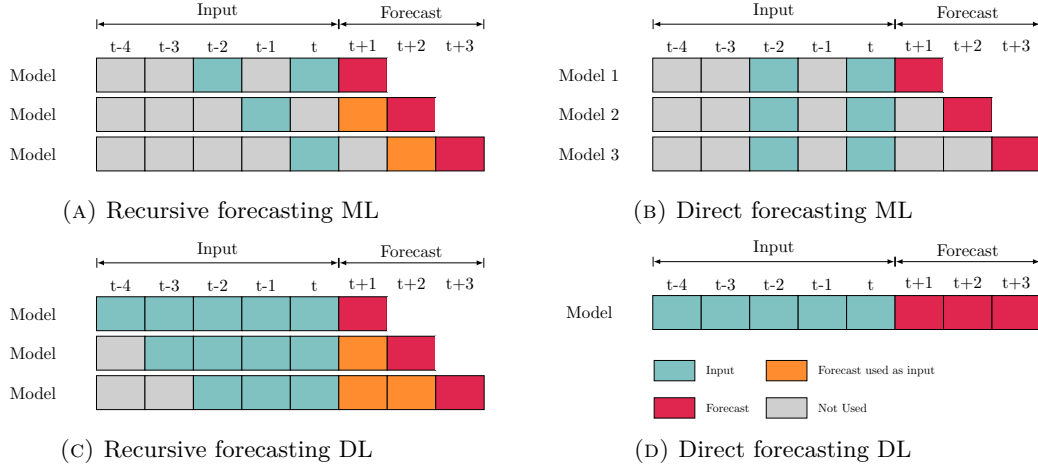


FIGURE 3.2: Illustration of different time series forecasting approaches (columns) for ML and DL models (rows), shown for a forecast horizon of three.

Machine Learning models

$$\hat{\mathbf{y}}_{t+h} = \begin{bmatrix} \hat{y}_{t+1} \\ \hat{y}_{t+2} \\ \vdots \\ \hat{y}_{t+h} \end{bmatrix} = \begin{bmatrix} \hat{f}_1(\mathbf{y}_l, \mathbf{X}_l) \\ \hat{f}_2(\mathbf{y}_l, \mathbf{X}_l) \\ \vdots \\ \hat{f}_h(\mathbf{y}_l, \mathbf{X}_l) \end{bmatrix} \quad (3.1)$$

where:

- $\mathbf{y}_l$ is a vector of l AR values, with maximum possible time step = t .
- $\mathbf{X}_l$ is a $l \times p$ matrix of p exogenous variables and each with l historic values, with maximum possible time step = t . Note that these values may be different across all inputs, but are notated as the same for simplicity.
- h is the forecast horizon

Deep learning models

$$\hat{\mathbf{y}}_{t+h} = \begin{bmatrix} \hat{y}_{t+1} \\ \hat{y}_{t+2} \\ \vdots \\ \hat{y}_{t+h} \end{bmatrix} = \hat{f}(\mathbf{y}_{is}, \mathbf{X}_{is}) \quad (3.2)$$

where:

- $\mathbf{y}_{is}$ is a vector of input sequence (is) length historic values, spanning a time period from $t - is$ to t
- $\mathbf{X}_{is}$ is a $is \times p$ matrix of p exogenous variables and each with input sequence length historic values, spanning a time period from $t - is$ to t .
- h is the forecast horizon

Based on the above formulations, the learning problem I aimed to solve is to minimise the distance between the model predictions, $\hat{\mathbf{Y}}_h$, and the true values for the same period, $\mathbf{Y}_h$, as measured by the loss function $\mathcal{L}(\mathbf{Y}_h, \hat{\mathbf{Y}}_h)$. To assess this difference, I used the Mean Absolute Error (MAE) loss function, which is defined for a single forecast horizon in Equation 3.3. Two notable exceptions are the lasso and SVR, and I discuss their specific loss functions in Sections 3.7.1 and 3.7.5. Note that taking the average over a complete dataset (e.g., a test set) is equivalent to averaging over a set of horizon values. Consequently, h can be replaced by T , where T denotes the number of samples in the evaluated dataset.

$$\mathcal{L}(\mathbf{y}_h, \hat{\mathbf{y}}_h) = \frac{1}{h} \sum_{i=1}^h |y_{t+i} - \hat{y}_{t+i}| \quad (3.3)$$

Many of the response variables modelled in Chapter 7 vary in scale over time. My preference for the MAE loss function reflects this consideration. Because MAE measures absolute differences, it is less susceptible to disproportionately weighting larger values compared with a squared-error loss such as Mean Squared Error (MSE). For example, when modelling case counts, a shift in the mean due to a new variant could result in prediction errors for that variant receiving greater weight than those for previous variants, even if the relative error is similar. MAE mitigates this effect by penalising all deviations linearly rather than quadratically.

From a statistical perspective, MAE can be interpreted as the likelihood for a homoskedastic Laplace error model, implying that minimisation targets the conditional median rather than the mean (Koenker and Bassett Jr, 1978). In this Thesis, however, I applied MAE in a distribution-free sense, without requiring residuals to follow any particular distribution, while still targeting the conditional median.

3.5 Model evaluation

In a predictive context, it is essential to evaluate how well a model generalises. Additionally, some models include hyperparameters that require tuning. In Section 2.8, I explained how k -fold cross-validation supports hyperparameter tuning, and how randomly held-out test sets help assess out-of-sample performance. In time series settings, this process becomes more complex due to the sequential dependencies within the data, which need to be preserved.

Consequently, different approaches are adopted for time series tasks. Unlike typical machine learning problems, datasets are split sequentially rather than randomly, in order to preserve temporal order. Additionally, the model may be refit during the evaluation process to incorporate the most recent data; this approach is also known as online learning. For each prediction, the task can be approached using different strategies. Two such approaches are illustrated in Figure 3.3. The first row depicts the expanding window approach, while the second row shows the sliding window approach.

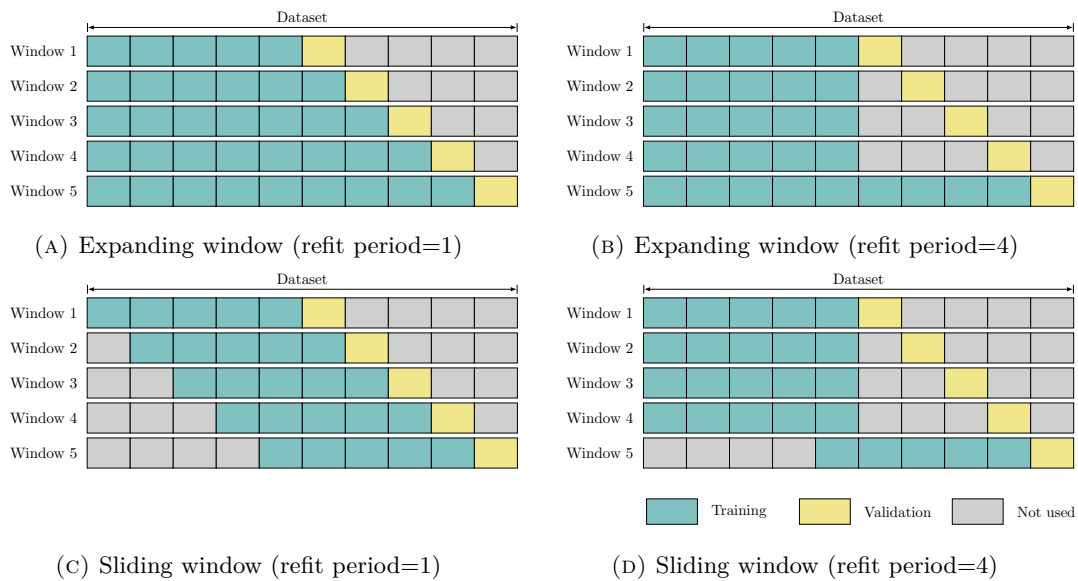


FIGURE 3.3: Illustration of different time series evaluation strategies (rows) and refit periods (columns). In each panel, blocks correspond to windows of h time steps, where h denotes the prediction horizon. In this illustration, the step size and prediction horizon are identical.

In the expanding window approach, the model is refit using all available data at periodic intervals, referred to as the refitting period. Figure 3.3a illustrates this with a refitting period of 1, where the model is retrained after every prediction window.

In contrast, Figure 3.3b presents a refitting period of 4, where the model is updated only after every fourth window, using all data available up to that point.

An alternative is the sliding window approach, in which the input size remains fixed and the window advances over time. The refitting period specifies how often the model is retrained. After each refitting period, the input window shifts forward by the same number of steps, and the model is refit accordingly. Examples of this approach with refitting periods of one and four are shown in Figures 3.3c and 3.3d, respectively. In my work, I adopted the expanding window approach, due to the limited size of my dataset and the assumption that incorporating additional data at each step improves generalisation.

I used a percentage-based evaluation metric to provide an intuitive and easily comparable measure of model performance. Loss functions are typically defined with properties that make them suitable for optimisation; for example, differentiability or convexity may be desirable. However, such functions can lack intuitive interpretability. Consequently, for evaluation purposes, I used Mean Absolute Percentage Error (MAPE), which reformulates the MAE as a percentage error relative to the true value, as defined in Equation 3.4. In addition to its interpretability, an advantage of this approach is that it enables comparisons across different series, which is desirable given the experiments I conducted in Chapter 7.

$$MAPE(\mathbf{y}_h, \hat{\mathbf{y}}_h) = \frac{1}{h} \sum_{i=1}^h \left| \frac{y_{t+i} - \hat{y}_{t+i}}{y_{t+i}} \right| \quad (3.4)$$

3.6 Hyperparameter tuning

To select hyperparameters, I trained models multiple times using the expanding window approach across a range of configurations. As in Section 2.8, this process is repeated for each configuration, with an iteration referred to as a trial. Several strategies have been proposed to guide hyperparameter selection across trials. A common starting point is a grid search or random sampling. However, many of the models I used involve large hyperparameter spaces. In such cases, a grid search becomes computationally expensive, while random sampling may result in poor coverage.

To address this, I adopted a more informed strategy that proposes hyperparameter configurations likely to yield lower values of an objective function. Unlike traditional loss functions, this function does not need to satisfy properties such as differentiability or convexity. Since I evaluate performance using MAPE, I also defined it as the objective function for tuning. I specifically used a Tree-structured Parzen Estimator (TPE), a Bayesian optimisation method based on Parzen estimators, also known as kernel density estimators (KDEs).

Standard Bayesian optimisation methods to hyperparameter tuning model the objective function conditioned on the hyperparameters. They use a surrogate model, such as a Gaussian Process, to approximate this function. An acquisition function then selects the next configuration to try by deciding how to sample the hyperparameter space. A common choice is Expected Improvement (EI), which estimates the expected gain over the current best result. However, EI can be expensive to compute.

The TPE takes a different approach. Instead of modelling the objective function, it models the distribution of hyperparameters conditioned on the objective. It uses these conditional probabilities as surrogate models and approximates EI from them without needing to calculate it explicitly. This approach is more efficient and avoids the computational cost of directly evaluating EI.

To express this more formally, let $\mathbf{h} \in \mathbb{R}^d$ represent a d -dimensional hyperparameter configuration with an associated MAPE score y . The TPE begins with an initial set of random trials. Based on the outcome of these trials, it defines the following partitioned conditional distribution:

$$p(\mathbf{h} \mid y) = \begin{cases} l(\mathbf{h}), & \text{if } y < y^* \\ g(\mathbf{h}), & \text{if } y \geq y^* \end{cases} \quad (3.5)$$

where $l(\mathbf{h})$ and $g(\mathbf{h})$ are density estimates, y are the observed objective values, y^* is the γ -quantile of observed objective values, and γ is a hyperparameter.

More precisely, the TPE estimates $l(\mathbf{h})$ using all hyperparameter configurations with MAPE values below a given threshold, and $g(\mathbf{h})$ using those with values at or above that threshold. As a result, $l(\mathbf{h})$ represents promising configurations, while $g(\mathbf{h})$ captures less favourable ones. The sampler aims to propose configurations that have

high probability under $l(\mathbf{h})$ and low probability under $g(\mathbf{h})$. At the end of each trial, after the initial random search phase, the algorithm selects multiple candidates and evaluates them using the ratio $\frac{l(\mathbf{h})}{g(\mathbf{h})}$, which approximates EI². For further details, I refer the reader to Bergstra et al. (2011).

3.7 Machine learning models

In the following Sections, I describe the ML models I used to address the time series learning problem outlined in Section 3.4. Each model defines a functional space, from which the most appropriate one must be selected.

For each model, I provide the exact functional form of the prediction for a single time step, along with details on how the model parameters are estimated. These models have explicit, structured representations that aid intuition. In the cases of the lasso and SVR, I also include their respective loss functions, which, as noted in Section 3.5, differ from MAE.

3.7.1 Least Absolute Shrinkage and Selection Operator

The lasso, introduced by Tibshirani (1996), is similar to linear regression in that it is a simple model that assumes a linear relationship between the predictors and the response variable. However, it extends linear regression by introducing a shrinkage term that constrains the parameter space.

A prediction using the lasso at time $t + 1$ is defined in Equation 3.6. In the Sections that follow, each machine learning model, when predicting at time $t + 1$, is assumed to include l AR terms, $\mathbf{y}_t \in \mathbb{R}^l$, and a single exogenous variable with p historic values, $\mathbf{x}_t \in \mathbb{R}^p$. Only one exogenous variable is included for simplicity, although in practice there may be several, including static variables.

For notational convenience, $\mathbf{z}_t \in \mathbb{R}^{l+p}$ is defined as the concatenation of $\mathbf{y}_t$ and $\mathbf{x}_t$. Since elements of $\mathbf{z}_t$ may correspond to the same time step, they are differentiated using z^y and z^x to indicate AR and exogenous features, respectively. The notation t_i is used to indicate the lag corresponding to the i^{th} z variable. This updated notation is shown in Equation 3.7.

²This ratio is inverted in practice to cast the task as a minimisation problem.

$$\hat{y}_{t+1} = \hat{f}_{t+1}(\mathbf{y}_t, \mathbf{x}_t) = \hat{\beta}_0 + \hat{\beta}_1 y_t + \cdots + \hat{\beta}_l y_{t-l} + \hat{\beta}_{l+1} x_t + \cdots + \hat{\beta}_{l+p} x_{t-p} \quad (3.6)$$

$$= \hat{f}_{t+1}(\mathbf{z}_t) = \hat{\beta}_0 + \hat{\beta}_1 z_t^y + \cdots + \hat{\beta}_l z_{t-l}^y + \hat{\beta}_{l+1} z_t^x + \cdots + \hat{\beta}_{l+p} z_{t-p}^x \quad (3.7)$$

In Chapter 7, I used the standard implementation of the lasso, which optimises MSE as the loss function plus an ℓ_1 regularisation term (Tibshirani, 1996; Pedregosa et al., 2011). The regularised loss used for the lasso for a single forecast horizon is shown in Equation 3.8.

$$\mathcal{L}(\mathbf{y}_h, \hat{\mathbf{y}}_h) = \frac{1}{h} \sum_{i=1}^h (y_{t+i} - \hat{y}_{t+i})^2 + \lambda \sum_{j=1}^{p+l} |\beta_j| \quad (3.8)$$

A more comprehensive justification of the lasso, including an explanation of the bias–variance trade-off and how using an ℓ_1 penalty can lead to variable selection, is provided in Appendix A.1. This Appendix also outlines the assumptions of the lasso. In time series applications, the assumption of independent error terms is violated. As a result, inferential measures such as confidence intervals and p-values are unreliable. However, this violation does not affect the coefficient point estimates or the fitted values. Since the primary objective is an assessment of models based on these point estimates, the lasso remains an appropriate choice, offering a valid linear representation despite the violated assumption.

3.7.2 Regression trees

Both random forests and CatBoost are ensembles of regression trees. An ensemble model is a collection of models where the final prediction is a function of the individual model predictions. Accordingly, I begin with a brief explanation of regression trees in this Section and then describe each tree ensemble in the two subsequent Sections.

A regression tree aims to partition the predictor space into distinct regions using a set of decision rules (Breiman et al., 1984). An illustrative example of this process, using two predictors y_t and y_{t-1} , is shown in Figure 3.4. Figure 3.4a presents a diagrammatic representation of the tree. The first node is called the root, and each subsequent node has a single incoming edge. Intermediate nodes are referred to

as internal nodes, and the final nodes are called leaf nodes. The total number of levels in a tree is known as its depth. Based on this terminology, the diagrammatic representation is inverted, with the root at the top and the leaves at the bottom. Every node corresponds to a rule that stratifies the predictor space and defines a half-plane relative to that node; that is, each split takes into account all previous splits.

The partitioned predictor space is shown in Figure 3.4b. The model assigns a constant value c_m to each region, where c_m is a function of the response values of all training samples within that region. These samples are indicated by the blue points in the Figure. To make predictions, the model determines which region a new sample falls into and assigns the corresponding c_m as the prediction.

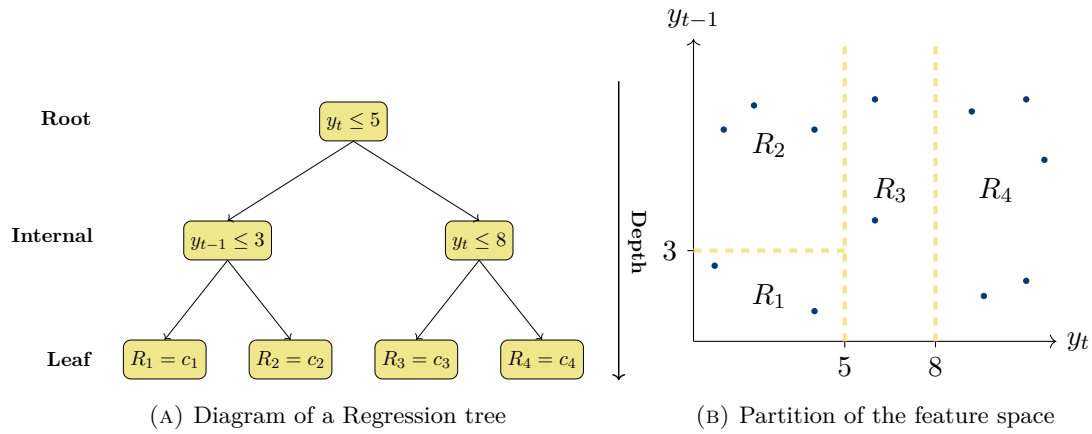


FIGURE 3.4: A diagram of a regression tree and its corresponding partitioned predictor space. The blue dots in (B) represent samples. The prediction for any point falling within a region is a function of the response values of all training samples in that region.

Using the notation given in Section 3.7.1, A prediction at time $t + 1$ is defined in Equation 3.6 below.

$$\hat{y}_{t+1} = \hat{f}(\mathbf{z}_t) = \sum_{m=1}^M c_m \mathbf{1}_{R_m}(\mathbf{z}_t) \quad (3.9)$$

where the indicator function $\mathbf{1}_{R_m}$ specifies whether $\mathbf{z}_t$ falls within region R_m , and is defined as:

$$\mathbf{1}_{R_m}(\mathbf{z}_t) = \begin{cases} 1 & \text{if } \mathbf{z}_t \in R_m \\ 0 & \text{otherwise} \end{cases} \quad (3.10)$$

The learning problem for a regression tree involves finding, for each region m , a constant value c_m that minimises the loss function $\mathcal{L}(\mathbf{y}_m, c_m)$ based on the training response values within that region. Using an MAE loss, the optimal solution for c_m node is the median of the values that fall in a terminal node. Since each c_m depends on how the predictor space is partitioned, the task reduces to identifying decision rules that produce regions where the assigned constant minimises the overall loss.

To grow a regression tree, a greedy splitting procedure is employed. At each node, the best split is chosen locally rather than optimising over all possible trees, which would be computationally infeasible in practice. The process is repeated until the tree reaches a pre-specified depth. Choosing an appropriate stopping depth is non-trivial, as it requires balancing model complexity against the bias–variance trade-off. Deeper trees are more complex and generally achieve lower bias, but this comes at the expense of higher variance.

A common strategy is therefore to begin by growing a large tree that may overfit the data, and then prune it back using cost-complexity pruning. This method merges nodes from the leaves upwards, with the extent of pruning controlled by a regularisation parameter α . Larger values of α yield simpler trees with fewer leaves, while $\alpha = 0$ corresponds to the fully grown tree. The formal details of this procedure, together with the splitting approach and estimation of c_m , are provided in Appendix 3.7.2.

3.7.3 Random forest

A random forest is an ensemble of bootstrap aggregated (also known as bagging) trees that includes additional steps to decorrelate these bagged trees (Breiman, 1996; Breiman, 2001).

In the absence of multiple training datasets, bagging creates them by sampling with replacement from the original training set. A separate regression tree is then fitted to

each bootstrapped dataset. The intuition is that bagged trees reduce variance while maintaining the same level of bias (complexity) as individual trees.

An issue with bagging is that the trees being averaged can be similar, especially when some features dominate. This correlation among trees reduces the benefit of variance reduction. To address this, a random forest uses only a subset of predictors when building each tree. Using a subset decorrelates the trees, making a random forest similar to bagged trees but with the important distinction that each tree is constructed using only a subset of the available predictors.

Let Q trees grow using this approach, where the q^{th} tree, built from a bootstrapped sample and a subset of features, is represented by f^q (in the same form as Equation 3.9). A prediction at time $t + 1$ can then be defined as:

$$\hat{y}_{t+1} = \hat{f}(\mathbf{z}_t) = \frac{1}{Q} \sum_{q=1}^Q f^q(\mathbf{z}_t) \quad (3.11)$$

3.7.4 CatBoost

CatBoost is a tree-based ensemble model that, unlike random forests which rely on bagging, uses boosting to build trees sequentially (Prokhorenkova et al., 2019).

In boosting, the process begins by fitting a simple tree to the training data. A second tree is then fit to the residuals of the first, and this new tree is added to the ensemble. The procedure is repeated iteratively. Each tree is typically shallow, with only a few terminal nodes and contributes a small correction to the overall model.

In contrast to bagging, which averages a set of independent models, boosting builds a sequence of typically weak, dependent models, each correcting the errors of the previous one. The contribution of each new tree is scaled by a factor η , known as the learning rate. Smaller values of η result in slower learning, with each tree having a smaller influence on the final prediction.

Assuming the same notation as in the previous Section, and denoting Q boosted trees and the first tree as f^0 , where each tree has the functional form given in Equation 3.9, a prediction at time $t + 1$ is given by:

$$\hat{y}_{t+1} = \hat{f}(\mathbf{z}_t) = f^0 + \eta \sum_{q=1}^Q f^q(\mathbf{z}_t) \quad (3.12)$$

CatBoost follows the general principles of boosting but incorporates several key implementation choices that enhance its performance. Four main features are highlighted. First, similar to a random forest, it optionally subsamples a fraction of the training data (without replacement) when growing each tree, which helps reduce variance.

Secondly, CatBoost uses a specific type of boosting known as gradient boosting (Friedman, 2001). Unlike standard boosting, which fits trees to residuals, gradient boosting fits each subsequent weak learner to the gradient of the loss function; these gradients act as pseudo-residuals. This approach enables the use of stochastic gradient descent to minimise the loss function efficiently. Gradient boosting is a common approach to improving computational efficiency across other modern boosting techniques.

Thirdly, to control tree complexity, Catboost introduces an ℓ_2 regularisation term into the loss function instead of relying on post-hoc pruning. Since boosting naturally encourages shallow trees, pruning is unnecessary. Specifically, Catboost adds the term $\lambda \sum_{m=1}^M c_m$ to the optimisation objective defined in Equation A.4.

Finally, CatBoost also handles categorical features efficiently using an approach called ordered target statistics. This approach is similar to mean encoding, which assigns a category the mean of its target values. However, to avoid target leakage, this approach only uses target values from earlier observations in an ordered sample to compute these encodings. Optionally, it may generate multiple permutations of the training data to reduce variance in the encodings. However, this is not done for time series tasks as it would break the temporal structure. For full implementation details, I refer the reader to Prokhorenkova et al. (2019)

I selected Catboost over other popular boosting methods such as LightGBM and XGBoost due to its effective handling of categorical variables, which are included in my experiments presented in Chapter 7. This choice is further supported by the results of Prokhorenkova et al. (2019), who demonstrated that Catboost consistently outperformed these alternatives across a range of benchmark tasks.

3.7.5 Support Vector Regression

SVR is an approach that aims to fit a hyperplane to the training data (Vapnik, Golowich, and Smola, 1996; Müller et al., 1997). However, SVR introduces additional conditions, detailed below, that allow small tolerances and specify how the hyperplane is fit. An important feature of SVR is its ability to fit non-linear functions by transforming the input into a higher-dimensional space. Applying a suitable transformation ($\phi(z)$) to the input data can make the relationship between the predictors and the response approximately linear in the transformed predictor space, allowing a hyperplane to be fit more effectively. For example, if the true relationship between the predictor and response is parabolic, one might apply a transformation like $\phi(z) = z^2$ to linearise the pattern.

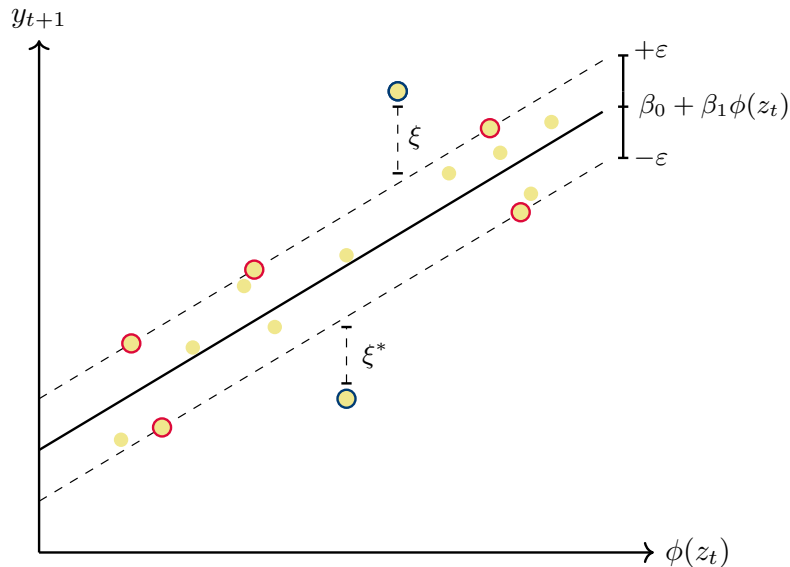


FIGURE 3.5: A diagram of an SVR hyperplane in a transformed predictor space.

To build intuition, an example of an SVR hyperplane in the transformed space is shown in Figure 3.5. To determine this hyperplane, a tube of width ϵ within which points incur zero loss is defined. The exact ϵ lines are known as the margin and are shown by the dotted lines. Smaller values of ϵ make the model less tolerant of errors, leading it to fit the data more closely. In contrast, larger values of ϵ increase the tolerance for error, meaning fewer data points influence the model fit, which can increase bias.

In addition, a tolerance is allowed for points that fall outside the tube, this is referred

to as slack, denoted by ξ and ξ^* for points above and below the ε margin, respectively. The extent of slack can be controlled by regularisation. Allowing greater slack introduces more flexibility beyond the ε -tube, which helps reduce variance across different training sets. Within these constraints, SVR fits the optimal hyperplane.

The dots circled in red or blue are the support vectors. The red dots lie exactly on the margin and are referred to as margin support vectors. The loss used in SVR is only non-zero for these support vectors, and they uniquely determine the final fitted function. This idea is expanded upon later in this Section.

SVR uses a specific loss function used known as the ε -insensitive loss and is defined as:

$$L_\varepsilon(y_{t+1}, \hat{y}_{t+1}) = \max(0, |y_{t+1} - \hat{y}_{t+1}| - \varepsilon). \quad (3.13)$$

Using this loss, together with constraints involving ε and slack variables, an optimisation problem can be specified. Since the primal problem is difficult to solve directly, it is reformulated in dual form using Lagrange multipliers. The resulting solution has two noteworthy characteristics. First, it depends only on the support vectors. Second, it can be expressed entirely in terms of kernel evaluations, which avoids the need for explicit feature mappings and makes the approach efficient even in high-dimensional spaces.

Consequently, given the reformulation, a prediction at time $t + 1$ is defined as:

$$\hat{Y}_{t+1} = f(\mathbf{z}_t) = \sum_{t_i=1}^T (\hat{\alpha}_i - \hat{\alpha}_i^*) K(\mathbf{z}_{t_i}, \mathbf{z}_t) + \beta_0 \quad (3.14)$$

where α_{t_i} and $\alpha_{t_i}^*$ are the dual variables and $K(\mathbf{z}_{t_i}, \mathbf{z}_{t_j})$ is the kernel function, representing the inner product in the transformed feature space.

Based on this formulation, predictions are obtained as weighted kernel combinations of the support vectors, with weights determined by the coefficients $\hat{\alpha}_t$ and $\hat{\alpha}_t^*$, together with an intercept term. Notably, making a prediction requires computing similarities between the input and all support vectors. Consequently, the complete set of support vectors, which form a subset of the training data, must be retained at prediction time.

A more detailed explanation of the primal and dual formulations of SVR is provided in Appendix A.3.

3.8 Deep Learning

In the previous Sections, I introduced various ML models and provided explicit functional forms for their predictions at time $t + 1$. The following Sections turn to DL architectures. For these models, explicit functional forms are not provided, as they are less straightforward due to their multiple interconnected components and more complex structures. However, all models considered here conform to the general form outlined in Equation 3.2. Several components of these architectures have already been introduced, such as the MLP in Section 2.8 and the Transformer architecture in Section 2.6. In Section 3.9.1, I expand on RNNs and their associated memory cells, previously discussed in Section 2.5, providing more detailed explanations because they serve as building blocks for various architectures. The remaining Sections then proceed by showing how these components are combined to form specific architectures and by justifying the design choices behind them.

3.8.1 Training

The DL models I used share a common approach to parameter estimation, in contrast to the ML methods introduced earlier. Before describing the individual models, I first outline the general training procedure applied to all DL models in Chapter 7. This Section provides a high-level summary; for a more detailed discussion of optimisation in DL, I refer the reader to Chapter 8 of Goodfellow, Bengio, and Courville (2016).

To train models, I optimised MAE using the Adam optimiser (Kingma and Ba, 2017). Adam is a variant of stochastic gradient descent with momentum that incorporates adaptive learning rates per parameter (Bottou, 2010). The learning rate determines the step size across the loss surface, while the gradient defines the direction of the update. I also employed a linear learning-rate scheduler that linearly decays the learning rate over time. Optimisation is performed on mini-batches, where each batch is referred to as a training step. Rather than specifying the number of epochs, I instead defined a maximum number of training steps for each model.

I applied window-based scaling to all inputs, meaning each window within a batch is scaled independently. This process aims to homogenise the data, addressing both temporal distribution shifts and differences in scale across series.

The scaling method I used is Reversible Instance Normalisation (RevIN) (Kim et al., 2021). RevIN standardises each window independently by subtracting its mean and dividing by its standard deviation. While this normalises the scale and distribution, the model must learn the reverse transformation to map the outputs back to the original data space. Standardisation removes non-stationary information, such as trends and changing volatility, which may be crucial for accurate modelling. Learning the reverse transformation can therefore be non-trivial.

RevIN addresses the issue of reversal by explicitly denormalising the final predictions using the same mean and standard deviation applied during input normalisation. This simple step allows the model to benefit from scale-invariant input distributions while recovering the original scale and dynamics of the target series. The RevIN procedure is applied at the input layer, with denormalisation occurring at the output layer after the final prediction is made. It is implemented as a trainable layer, with an optional learnable affine transformation applied to the normalised input. These parameters are optimised jointly with the model during training.

3.9 Recurrent Neural Networks

3.9.1 Vanilla Recurrent Neural Networks

I introduced the concept of an RNN in Section 2.5 in the context of NLP and encoder-decoder sequence-to-sequence models (Jordan, 1986; Elman, 1990). Additionally, I introduced the idea of attaching a classification head to an encoder model. RNNs used in time series modelling follow a similar idea to attaching an MLP to an encoder.

An example of an RNN with a hidden layer dimension of two is shown in Figure 3.6; a dimension of two is used for ease of illustration, although in practice this dimension d can be much larger. An RNN is used to encode an input sequence, where each input carries temporal context from the previous inputs. At each step in the sequence, both the current input and the previous hidden state are used to determine the current value of the hidden layer. Each hidden layer encodes temporal information, so by including the hidden layer from the previous step, temporal context is captured when

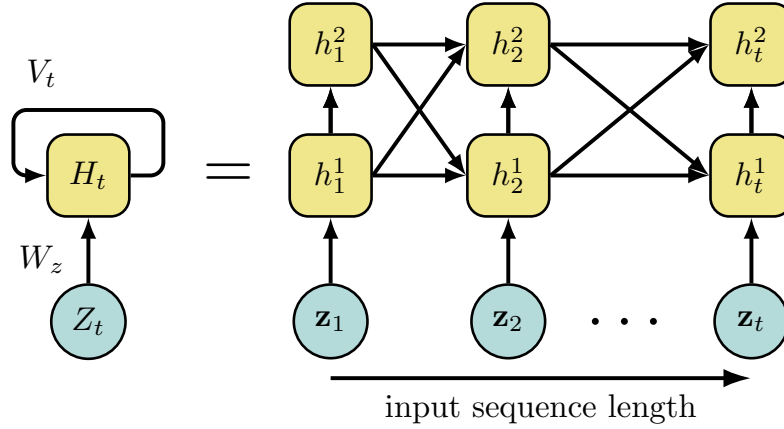


FIGURE 3.6: Diagram of an unrolled RNN. Z_t and H_t represent matrices corresponding to the input and hidden states for the entire system at time step t , while $\mathbf{z}$ and $\mathbf{h}$ denote the input and hidden state at each individual timestep. The unrolled version, shown to the right, omits weights for clarity.

processing the current input. Each time step is processed independently, and there are input vectors, $\mathbf{z}_t$, corresponding to each time step in the input sequence length.

The encoded hidden states are then used as input to an MLP to predict the forecast horizon. Specifically, for a given prediction horizon h , the last h hidden layers are used as input to an MLP with the structure shown in Figure 2.5a. However, instead of the logit layer, the output layer has several nodes equal in number to the forecast horizon. Furthermore, rather than using a softmax activation, a linear activation function is used in the output layer.

Importantly, the parameters of both the RNN encoder and the MLP are trained jointly. Each line in the unrolled RNN has a parameter attached to it. This joint training means that during training, the encoder learns to extract features at each step that capture information from both the current and previous steps, optimising these representations to be most useful for the MLP in minimising the MAE of its predictions. As a result, the entire model is optimised end-to-end.

3.9.2 Gated cells

One implication of this structure is that, when optimising an RNN using gradient descent, the loss must be backpropagated through time (Williams and Zipser, 1989). For long sequences, this can lead to exploding or vanishing gradients. To address these

issues, modifications are made to the structure of each hidden layer using memory cells. Two commonly used gated cells are the LSTM and the GRU.

Each approach regulates the amount of information retained within a hidden unit. In Figure 3.6, every hidden unit is replaced by one of these cells. Their role is to control the flow of temporal information through the hidden layers, thereby mitigating gradient issues in RNNs and enabling longer sequences to be processed more effectively. Both approaches achieve this through gating mechanisms, which determine how information is stored, updated, or discarded over time. LSTM cells employ three gates (input, forget, and output) together with a dedicated memory cell (Hochreiter and Schmidhuber, 1997). In contrast, GRU cells are simpler, using only two gates (update and reset) and a single state that combines the memory and hidden state (Cho et al., 2014a). Full mathematical details and illustrations of the gating mechanisms are provided in Appendix A.5.

In this Thesis, I employed an RNN with an LSTM cell. In addition, I used a variant of the RNN, introduced in the following Section, in which the choice of cell type is treated as a hyperparameter.

3.10 Dilated Recurrent Neural Networks

I used a variant of the RNN called a DilatedRNN (Chang et al., 2017). Similar to a standard RNN, the DilatedRNN follows the encoder-MLP approach to generate forecast predictions, with the entire network optimised jointly. The key difference is that the DilatedRNN introduces skip connections between hidden units.

A DilatedRNN adopts skip connections across the temporal dimension, with a slightly different motivation. In Figure 3.6, each hidden layer is a function of the previous hidden layer, that is, $\mathbf{h}_t = f(\mathbf{z}_t, \mathbf{h}_{t-1})$. In a DilatedRNN, this strict dependency is broken and connections between hidden layers can be skipped, that is, $\mathbf{h}_t = f(\mathbf{z}_t, \mathbf{h}_{t-s})$, where s is the dilation factor. Additionally, there are no cross-connections between layers³. Multiple RNN layers are stacked, each with a different skip connection. For example, in the network shown, layer one may have a dilation of one and remain unchanged, while layer two may have a dilation of two, meaning the output for h_1^2 would go directly

³In the case where a dilation of 1 is not used, an additional connection from the final hidden layer from the previous timestep is provided when generating the final output for the current timestep.

to h_3^2 . This approach allows for deeper networks, with each RNN layer capturing different temporal context; layers with higher dilations capture broader context. By using skip connections, the model can achieve a deeper temporal context without suffering from exploding or vanishing gradients and is more parameter-efficient.

An additional difference between the DilatedRNN and the LSTM, which may be specific to the implementation I used, lies in how inputs are handled. Rather than feeding the h (forecast horizon length) hidden states directly into the network, a linear layer generates an embedding vector of length h , combining information from all hidden states across the input sequence.

3.11 Temporal Fusion Transformer

The only transformer-based architecture I used is the Temporal Fusion Transformer (TFT) (Lim et al., 2021). Most available transformer models, such as the Autoformer and Informer, accept only future exogenous variables (Zhou et al., 2021; Wu et al., 2021a). The TFT is one of the few architectures that also accepts historic exogenous variables; incorporating external context was a central focus of the authors and was my motivation for selecting this model. The authors also emphasised interpretability. However, given the focus of Chapter 7, a detailed discussion of interpretability is unnecessary beyond the role of interpretable multi-head attention. For further details and an illustration of the architecture, I refer the reader to the original work Lim et al. (2021). In the remainder of this Section, I provide an explanation of each component of the architecture.

Gated Residual Networks (GRNs) are the basic building blocks of the TFT. The GRN takes an input vector and, when available, a static feature vector. The concatenated inputs first flow through an MLP with an Exponential Linear Unit (ELU) activation (Clevert, Unterthiner, and Hochreiter, 2016). The MLP output passes through another dense layer with a linear activation, followed by dropout; dropout randomly sets a fraction of activations to zero during training, which discourages co-adaptation among neurons and improves generalisation (Hinton et al., 2012). A Gated Linear Unit (GLU), functionally similar to the gates in GRUs and LSTMs, applies a learned gate element-wise to the output of the dropout layer (Dauphin et al., 2017). The GLU controls how much of the transformed signal is retained. A skip

connection adds the gated output to the original input, after which layer normalisation is applied. In effect, the gate decides how strongly the current block contributes to the overall representation; if the contribution is unhelpful, it can be suppressed to near zero.

Unlike conventional variable selection, where entire variables are dropped, selection here is performed on an instance-wise basis; that is, for a given time step, a variable selection network is applied. The variable selection network first transforms inputs by applying a linear transformation for continuous inputs and entity embeddings for categorical features (Gal and Ghahramani, 2016). Each variable is then passed through its own GRN, with weights shared across time steps. These variables are separately concatenated with a static feature vector (if available) and passed through another GRN with an additional softmax layer at the end. The softmax layer bounds the GRN outputs between zero and one; these act as variable selection weights. A dot product of these variable selection weights and the processed variables generates a single value for all inputs at a given time step.

An important component of the TFT is its ability to encode static variables. Static inputs are first passed through a variable selection network. The selected outputs are then passed to four GRNs, each producing a distinct static feature vector for use in different parts of the architecture: one vector is used for temporal variable selection; two vectors are used for the local processing of temporal features; and one vector enriches temporal features with static information. These static feature vectors are integrated throughout the network to provide relevant static information where useful.

Once static variables have been encoded and the variable selection network has processed past inputs, the next step is to encode temporal context into a set of hidden states. This is accomplished using a sequence-to-sequence approach with an LSTM encoder-decoder structure, where the decoder generates hidden state vectors for the forecast horizon. The first LSTM in the encoder is initialised using static feature vectors (one for the hidden state and one for the memory state). The hidden state outputs of both the encoder and decoder, which represent temporal features, are then fed to a GLU. A skip connection adds the initial feature vector input to the hidden state. To enrich the temporal features with static metadata, the static enrichment feature vectors are concatenated to this input and passed through a GRN.

The hidden states are updated using a temporal self-attention layer, which adds context from previous time steps and encoder hidden states. Instead of standard multi-head attention, the authors implement an interpretable variant by replacing multiplicative attention (see Section 2.6) with additive attention. Additionally, weights are shared with the input feature selection component; each head thus learns distinct temporal patterns while jointly attending to features identified as applicable across all heads.

All static-enriched temporal features are grouped and attended to at each forecast time step, creating a self-attended feature for every step in the forecast horizon⁴. These features are passed through a GLU with a skip connection, then through a final GRN with shared weights. Another GLU with a skip connection from the original sequence-to-sequence layer is applied, allowing the transformer block to be skipped if unnecessary.

To generate a forecast, the final vector produced by the preceding components (of length equal to the forecast horizon) is passed element-wise through a forecast-step-specific dense linear layer to produce a prediction for each time step. As with the RNN models, the entire model is optimised jointly, where optimality is determined by MAE.

3.12 Temporal Convolutional Network

The use of transformers and RNNs followed naturally, given their success in NLP and their inherent ability to capture temporal context. However, the models presented in the following four Sections break from this trend and consider other architectures for time series modelling.

With this in mind, Bai, Kolter, and Koltun (2018) challenged the common association of temporal tasks with recurrent or transformer models. They conducted empirical comparisons between recurrent and convolutional models, and introduced the Temporal Convolutional Network (TCN), which they applied across tasks⁵

CNNs were initially developed for image processing. The general idea is similar for sequence data; convolutional filters encode local context to produce feature vectors,

⁴Decoder masking, as in CLM tasks, prevents data leakage.

⁵In their paper, Bai, Kolter, and Koltun (2018) described the TCN as a broad family of architectures but also provided a specific implementation. This Section refers to that implementation.

which can then be used in an MLP for downstream tasks. As a full discussion of CNNs is unnecessary for describing the TCN, I instead refer the reader to Chapter 9 of Goodfellow, Bengio, and Courville (2016) for a comprehensive introduction.

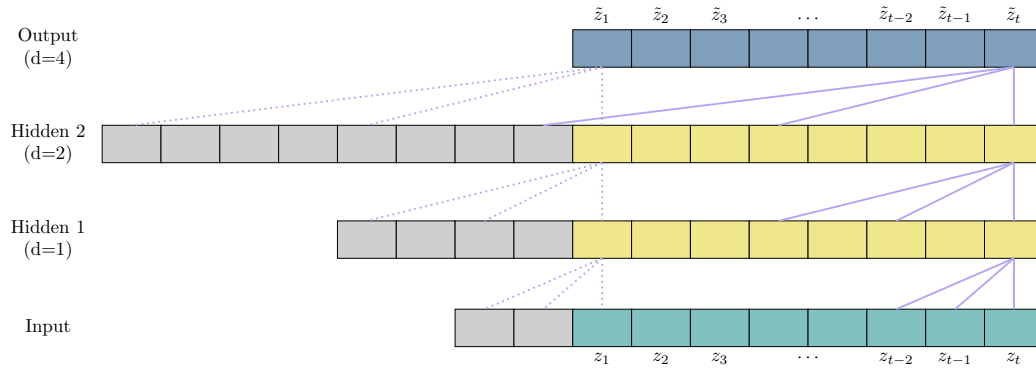


FIGURE 3.7: An illustration of dilated causal convolutions with dilation factors of 1, 2, and 4. The kernel consists of three learnable weights per layer, represented by lines between layers. Although these connections apply to every cell, for visual clarity, only those at two representative positions (1 and t) are shown. The connection at time step 1 is further distinguished with a dotted line, but the same kernel weights are used for this and all other positions in the layer. The variable z denotes either an exogenous or an AR feature. Grey blocks indicate zero-padding, which ensures that the number of entries remains constant across layers.

The TCN is a fully convolutional architecture, where each layer produces an output of the same length as the input. In the time series context, a one-dimensional convolutional kernel slides over the input series; conceptually, this is similar to the sliding window approach shown in Figure 3.3c, but with learnable weights. The kernel's width is called the kernel size (or filter size), and its role is to summarise the data points it covers. To prevent information leakage from future time steps, the TCN uses causal convolutions, which rely only on current and past values, analogous to causal masking in transformers. Each kernel also has a channel dimension, enabling different channels to capture different types of information. To ensure output sequences remain the same length as the input, the input to each layer is padded with zeros. This architecture closely resembles the earlier time delay neural network proposed by Waibel et al. (2013), but the TCN introduces two key additions: dilations and skip connections.

The first addition is the use of dilations, introduced earlier in the context of the DilatedRNN (Section 3.10). A dilation factor d means the convolutional kernel skips d time steps between inputs; this enlarges the receptive field, a concept from biological

vision, that can intuitively be thought of as how much of the input a model can “see” at once. By expanding the receptive field without increasing the number of parameters, the TCN captures broader temporal context and gains a longer memory. TCNs stack layers with base two exponentially increasing dilations (i.e. 1, 2, 4, ...), so every time step is included at least once, while deeper layers capture progressively broader context. Stacking layers with wider dilations at each layer captured a broader context in a parameter-efficient way. To preserve sequence length, zero-padding of $(\text{kernel size} - 1) \times d$ is added at each layer. An illustration of a stack of dilated causal convolutional layers is illustrated in Figure 3.7, with length-preserving zero-padding shown as gray blocks.

Although stacking dilated convolutions efficiently widens temporal coverage, it can lead to very deep networks, especially when long input sequences and large receptive fields are required. To address this, the TCN introduces skip connections through residual blocks. Each residual block stacks two dilated convolutions; each convolution is followed by weight normalisation (which parametrises weights as a direction vector and a scaling factor, each updated separately to encourage faster convergence), a Rectified Linear Unit (ReLU) activation, and spatial dropout. Spatial dropout, unlike standard dropout (previously described in Section 3.11), drops entire channels, where each channel represents a feature. This sequence is repeated twice to enable the network to learn more complex features. A skip connection links the input and final output of the residual block. If the input and output differ in the number of channels, a 1×1 convolution is applied to match the channel dimensions while preserving the sequence length.

As is a recurring theme across the DL models presented thus far, to generate forecasts, the TCN adopts the encoder-MLP structure. It creates an embedding vector that represents temporal features, which an MLP then uses to generate forecasts, with the whole architecture jointly optimised.

I chose the TCN because it is simpler than many other convolutional architectures, such as those described in Oord et al. (2016). A noteworthy advantage of the TCN is its efficiency. Given that each time-step can be convolved independently, the whole layer can be computed in parallel (unlike an RNN). For a complete discussion of the advantages and disadvantages, I refer the reader to Bai, Kolter, and Koltun (2018).

3.13 Neural Basis Expansion Analysis for Time Series with exogenous variables

While many recent state-of-the-art time series models rely on recurrent or convolutional structures, the Neural Basis Expansion Analysis for Time Series (N-BEATS) architecture has shown that models based on MLPs can also achieve strong predictive performance (Oreshkin et al., 2020). This architecture was subsequently extended to N-BEATS with Exogenous Variables (N-BEATSx), which incorporates exogenous variables⁶ (Olivares et al., 2023).

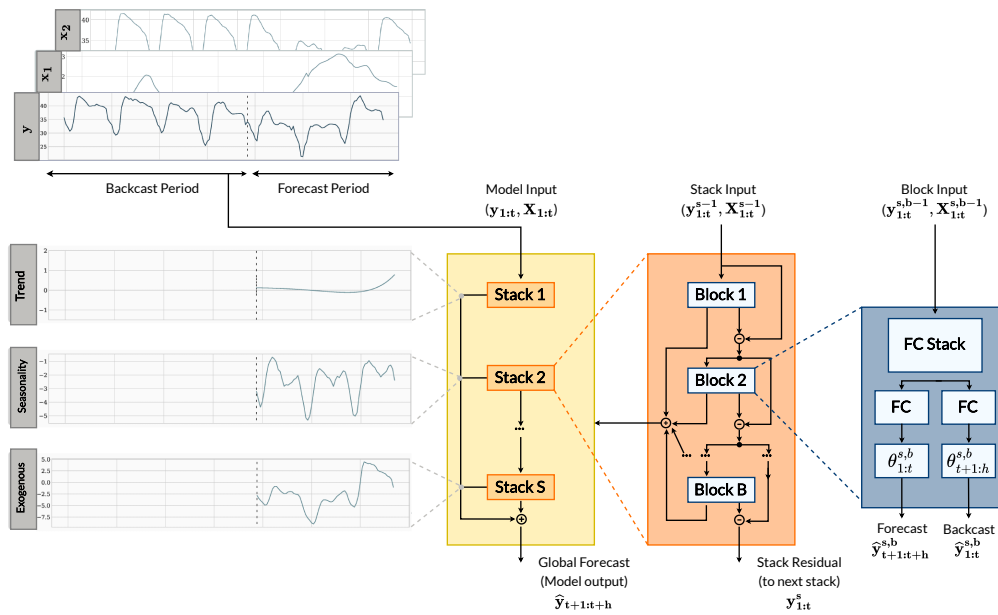


FIGURE 3.8: Diagram of the N-BEATSx architecture. From left to right: a block, a stack, and a composition of stacks are shown. The rightmost column illustrates the backcast and forecast periods, as well as the interpretable components generated when the model is used in interpretable mode. Source (adapted): Olivares et al. (2023).

The N-BEATSx architecture, illustrated in Figure 3.8, hierarchically combines individual modules to form the complete model. The fundamental building block, shown in the blue rectangle in the figure, makes two predictions: a forecast for the prediction horizon and a reconstruction of the historical response (the “backcast”) covering the input sequence length. Each output is generated in two steps. First, a fully connected layer linearly transforms the final hidden states to produce coefficients for a

⁶N-BEATSx includes a TCN in certain cases, so it is no longer purely MLP-based; however, it remains predominantly MLP-driven.

basis expansion. Second, these learned coefficients are used in the basis expansion to generate the forecast and backcast.

There are different choices in N-BEATSx for this basis expansion, and the choice of basis function determines the block name. A trend block is used to model the trend, for which Oreshkin et al. (2020) suggest a low-degree polynomial basis to capture slow-changing trends. A seasonal block captures seasonality, and the same authors propose a Fourier basis given its cyclical nature. The generic block does not use a known basis expansion and instead uses a linear layer to learn the best way to encode the context in the hidden layer. Intuitively, in the context of the basis functions, a block is allowed to represent its contribution to the overall forecast. In this way, it is similar to some of the previous encoding models presented. To address exogenous variables in generic blocks, exogenous outputs are concatenated to the AR input, as shown in Figure 3.8, and this is passed through the different block types.

With a view toward interpretability, which is discussed shortly, N-BEATSx also proposed a new block type called the exogenous block. In the exogenous block, the usual process is followed. However, rather than multiplying the predicted basis coefficients by a basis, or an identity matrix, the exogenous inputs are encoded into a feature vector using a TCN (explained in Section 3.12) and these are multiplied by the predicted coefficients. This process captures the important components of the input that can be directly used to determine the output. Olivares et al. (2023) refer to this process as decoding the encoded input.

These blocks are then arranged into stacks, with each stack consisting of blocks of the same type. Within each stack, double skip connections (Huang et al., 2018) are used. The first skip connection subtracts the input from the backcast prediction, yielding a residual that serves as the input to the next block; this is analogous to boosting, but focused on input components not yet used to explain the response. The second skip connection adds each block's forecast to the cumulative forecast. The total forecast for a stack is the sum of its blocks' predictions. This process is repeated at the stack level, where the residuals from each stack are passed as input to the next, and the forecasts from all stacks are additively combined to generate the final prediction. This repetition, additive combination, and the use of skip connections are illustrated by the orange and cream blocks in Figure 3.8.

The N-BEATSx architecture supports two modes, namely interpretable and generic.

The interpretable mode uses structured blocks arranged as trend, seasonality, and exogenous stacks. Each stack sequentially removes its corresponding component, and the outputs are additively combined, yielding an intuitive decomposition of the final prediction (illustrated in Figure 3.8). This mirrors classical approaches such as the Seasonal-Trend decomposition using LOESS (STL decomposition), where time series are separated into constituent parts. The generic mode, in contrast, does not provide this decomposition but offers greater depth and flexibility, which can improve performance. For this reason, I adopted the generic form⁷ in Chapter 7.

3.14 Time series Dense Encoder

Similar to N-BEATSx, with a focus on showing that MLP approaches could perform as well as other architectures in long-term times series modelling Das et al. (2024) proposed the Time Series Dense Encoder (TiDE) architecture. TiDE’s architecture comprises two sections: an encoder and a decoder, both built from a residual block. The full architecture and residual block are shown in Figure 3.9.

The residual block includes an MLP with one hidden layer and a ReLU activation, although the number of hidden nodes can be adjusted. Its output passes through a dropout layer. Separately, the original input is passed through a linear layer and then added to the MLP’s output via a skip connection. This sum is then processed by a layer normalisation.

The encoder uses this residual block in two ways. First, dynamic covariates are passed through a residual block to reduce the dimension of exogenous variables from p to $\tilde{p} \ll p$. This reduction, known as the temporal width, addresses the challenge of flattening a high-dimensional input over a long sequence length. The reduced-dimension exogenous variables are then flattened and concatenated with the AR feature vector (of input sequence length) and any static features. This combined vector is passed to the encoder, a sequence of residual blocks, which embeds all available information into a single embedding vector ($\mathbf{e}$ in the figure). Each residual block in the encoder has the same structure.

The embedding vector is then passed to the decoder block, which may consist of a different number of residual blocks, each with the same structure as those in the encoder.

⁷The generic form in Olivares et al. (2023), called N-BEATSx-G, uses only generic and exogenous stacks, whereas my approach mixes block types.

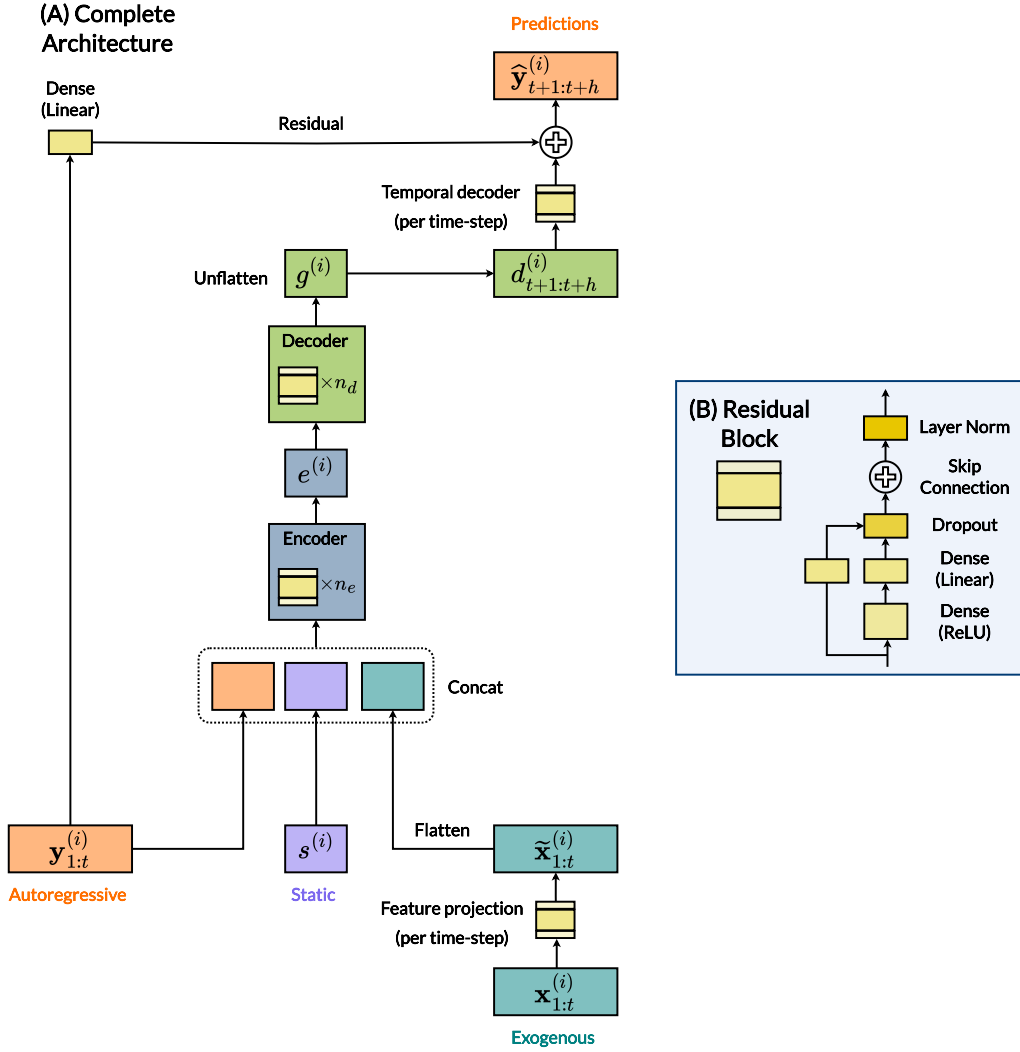


FIGURE 3.9: Diagram of the complete TiDE architecture (A), with the individual residual block shown (B). The linear projection of the AR input and the skip connection allow a linear solution to be found more easily when optimal. Source (adapted): Das et al. (2024).

The decoder transforms the embedding into a decoded vector ($\mathbf{g}$ in the Figure), which can be viewed as extracting the information most relevant for forecasting from the encoded representation. This process is conceptually similar to the exogenous block in N-BEATSx described in the previous Section. This decoded vector is a flat vector of dimension $h \cdot d_o$, where d_o is a tunable output decoder dimension. The vector is reshaped into an $h \times d_o$ matrix, with each row corresponding to the decoded context for a specific forecast time step. (If future exogenous variables are present, they are concatenated to this matrix; however, this step is omitted in my case.)

Each row of the decoded matrix is passed to a temporal decoder, implemented as a residual block with output size one, so each output yields a single prediction for

its corresponding time step. The final prediction is produced by adding a linear skip connection, where a linear layer generates a prediction using only the AR input sequence, which is then combined with the decoded output. This results in the final TiDE forecast.

The inclusion of a linear skip connection is a notable innovation of TiDE, reflecting its emphasis on the effectiveness of simpler models. By explicitly incorporating a linear component, TiDE can recover purely linear solutions when optimal by reducing the encoder-decoder weights to near zero and relying on the linear projection. Conversely, if the linear component is not helpful, its weights can be set to near zero, minimising its influence. This design choice is motivated by findings from Zeng et al. (2022), who observed that in some long-term forecasting tasks, simple linear models outperformed more complex transformer-based approaches.

3.15 Komolgorov-Arnold Network

Although the preceding models differ in architecture, they all rely on a standard design; non-linear modelling is achieved by applying fixed activation functions to learned linear combinations of the inputs. In contrast, the KAN takes a fundamentally different approach, drawing its inspiration from the Kolmogorov-Arnold Representation Theorem (Liu et al., 2025).

This theorem states that any multivariate continuous function on a bounded domain can be represented as a finite composition of continuous univariate functions and the binary operation of addition. Building on this idea, KANs model complex relationships not by applying a fixed nonlinearity after a linear projection, but by composing learnable univariate functions in a flexible, data-driven manner. Although this theorem served as a theoretical foundation for network design in the past, practical implementations fell short because they used only a single layer of univariate function composition. Representing complex functions with such shallow architectures would potentially require the constituent functions to be intricate for complex problems, making them difficult to learn. A key innovation introduced by KANs is the increased depth of function composition, which significantly enhances the network's expressive power while reducing the complexity required per layer; that is, the network has increased compositional complexity. An illustration of a KAN is shown in Figure 3.10.

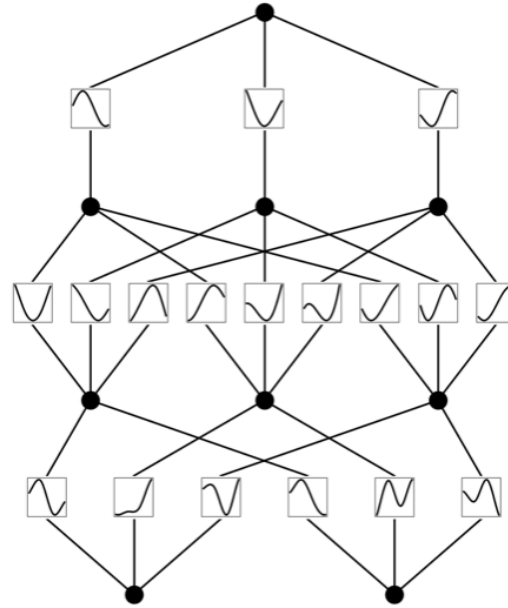


FIGURE 3.10: An illustration of a KAN, with learnable activation function as edges which are summed at the nodes. Source: Liu et al. (2025)

To create learnable activation functions, KANs use basis functions. Most commonly, a B-spline basis is used, as suggested by the authors, though other bases are possible. The control points of these splines are learnable parameters, allowing the network to approximate a wide variety of univariate functions flexibly. The key intuition is that KANs learn how to represent the response variable by directly shaping these functions, rather than applying a fixed nonlinearity.

The individual learned functions can be directly extracted and inspected, especially in shallow (simple) networks. This transparency allows users to analyse how the model forms its predictions, improving interpretability. In some cases, these extracted functions can even be matched to known symbolic forms. In such cases, after an initial round of training, a user can substitute a learned component with a specific symbolic function, retrain the network with this constraint, and ultimately obtain a final, fully symbolic representation of the model. While symbolic extraction is a possible byproduct, the core interpretability of KANs stems from their ability to reveal their internal mechanisms.

To further promote interpretability and model simplicity, KANs incorporate a regularisation approach that encourages the network to use a simpler representation when possible. Specifically, two regularisation terms are applied to the learned activation

functions: an ℓ_1 norm that encourages many functions to go to zero and an entropy penalty that discourages redundancy among the remaining functions. Both are controlled by a single hyperparameter, λ . Regularisation also helps address uncertainty regarding network shape, as the optimal configuration of depth and width depends on the complexity of the target function. Since this is generally unknown in advance, the network is initially grown deeper with the composite penalty; then, analogous to tree-based models, unhelpful nodes can be pruned to obtain a simpler structure.

Beyond interpretability, KANs offer several practical advantages. When training on new data, the local support property of each B-spline activation means that adjusting control points only perturbs the portion of the curve overlapping with the new input. If inputs extend beyond the original range, KAN implementations enlarge the grid by adding new control points rather than modifying existing ones. As a result, earlier regions of the function are largely preserved, providing a degree of memory and stability that ordinary MLPs lack, although the effectiveness of this memory is task-specific, as shown by Lee et al. (2025). However, this same property makes KANs poor at extrapolation, as predictions for inputs outside the training domain may be unreliable. Furthermore, I note that many advantages of KANs, including interpretability, may diminish as task complexity increases.

An additional drawback is that KANs are generally slower to train. The authors report that KANs are typically an order of magnitude slower than MLPs with a comparable number of parameters. Nevertheless, KANs can, in some cases, approximate the same target function as a larger MLP using a smaller and more interpretable network. The authors also found that KANs can be especially effective at learning partial differential equations (PDEs) and other challenging mathematical structures, and that smaller KANs can sometimes be more efficient for such tasks.

3.16 Summary

In this Chapter, I presented the key components of time series modelling and justified the methodological decisions that are applied in Chapter 7. I also described the specific ML and DL models that are used, with specific details on the variable structure provided in Chapter 7.

4 Curating a dataset of tweets

4.1 Overview

To apply the methodology presented in Chapter 2, a text-based dataset of tweets is required. In this Chapter, I outline the process I followed to compile a list of search terms and to download, parse, filter, and construct an analytical dataset of tweets retrieved using these search terms. A schematic of both the process for filtering raw tweets and generating an analysis dataset is shown in Figure 4.1. Each step included in Figure is explained in more detail in this Chapter. While my focus in this Chapter is on generating a cleaned dataset for analysis, my approach to parsing the Twitter data is also useful for further analyses.

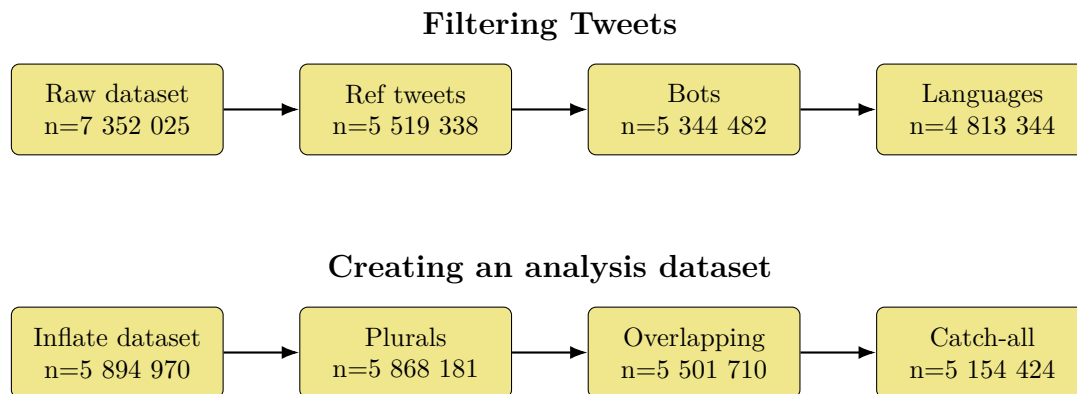


FIGURE 4.1: Flowchart displaying the process of filtering and creating both a raw and an analysis dataset.

4.2 Introduction

In this work, I focused on geotagged tweets. These are tweets where the author has explicitly specified their location. Using geotagged tweets enables comparison with location-specific epidemiological data, which varies across geographic regions. The exact location specified may vary in granularity, ranging from a country to an exact latitude and longitude. Specifically, I examined tweets from eight European countries and their national official languages: France (French), Germany (German), Italy (Italian), the Netherlands (Dutch), Poland (Polish), Portugal (Portuguese), Spain

(Spanish), and the UK (English)¹. I assumed that using the official language of each country would capture a broader and more representative user base, including both urban and rural populations, than using English alone across all countries (Sohail et al., 2025). I selected these countries because they represented diverse geographic regions, languages, and pandemic experiences, and for practical reasons such as the availability of annotators, the availability of epidemiological data, and the fact that each have a single official national language.

My decision to curate a custom dataset, rather than rely on existing ones, was motivated by the need to define a broader and more tailored set of search terms. Publicly available datasets typically used a narrow set of terms, often limited to synonyms of disease names (for example, `covid-19`, `coronavirus`, `sars-cov-2`) and common interventions (for example, `lockdown`, `mask`, `vaccine`).

A common approach to expanding topic coverage when using a limited set of search terms is to apply topic modelling techniques. For example, Lyu, Han, and Luli (2021) used 13 search terms, and Bogdanowicz and Guan (2022) used 12, both applying topic modelling to datasets composed almost entirely of synonyms related to COVID-19. In this context, topic modelling is an unsupervised machine learning method that identifies latent themes or topics within a collection of tweets. Each topic is represented by a set of keywords, and the broader topic may be manually labelled; for instance, a topic related to symptoms might include words such as cough, no taste, and no smell. However, I argue that explicitly specifying search terms provides greater precision. Moreover, I argue that due to limited overlap between these terms and the ones relevant to my analysis, many tweets relevant to my analysis are excluded from those datasets; I explore this in more detail in Sections 4.6.2 and 4.6.3.

A further motivation is that many larger datasets lack geotagged tweets. Previously, location has been inferred by parsing the user’s bios or other location data in the tweet. For example, Qazi, Imran, and Ofi (2020) adopted this approach and created a dataset with location information of more than 524 million multilingual tweets, but only 5.9 million of these are geotagged while the remainder of locations are inferred. While useful, inferred locations are less reliable than user-specified geotags (Sloan and Morgan, 2015). Although users may obscure their location with VPNs or by providing

¹For the UK and Germany, English and German function as the de facto official languages (Geographical Names, 2024; Kamocki et al., 2023), while in the remaining countries the official languages are de jure.

false information, I assumed such cases were minimal, since potentially problematic accounts were filtered out as detailed in Section 4.5.2.

However, while accurate geolocation is necessary for the purpose of this Thesis, I do acknowledge a major limitation of geotagged data. According to Twitter (2023), the prevalence of geo-tagged tweets is between $\sim 1\%$ - 2% . This low prevalence leads to demographic biases in the type of user who chooses to enable geolocation, as shown by Sloan and Morgan, 2015.

4.3 Search queries

Initially, in discussion with my supervisor, a subject matter expert on progression and policies implemented during the COVID-19 pandemic in the UK, we identified 92 search terms. Native speakers from Imperial’s Department of Public Health, with epidemiological expertise and local knowledge of the countries involved, then translated these terms. They translated only those terms they deemed relevant and contributed additional terms where appropriate. In addition, I used ChatGPT to augment the search term list with the 30 most commonly used COVID-19-related public policy terms in each country (OpenAI, 2022). The exact prompt used is given in below:

What were the 30 most commonly used public policy terms specific to {{COUNTRY}} used during the COVID-19 pandemic?

where {{COUNTRY}} was replaced by one of France, Germany, Italy, the Netherlands, Poland, Portugal, Spain, or the United Kingdom.

Many of the suggested terms were already included in the search list. The version of the LM (GPT-3.5) used by ChatGPT was trained on a wide array of publicly available data up to the end of January 2022, coincident with the end of the study period in this Thesis. Therefore the use of ChatGPT in this context can be considered analogous to searching a semantic database.

Search terms refer to the specific expressions of interest within tweets. These terms had to be formatted into search queries for the Twitter Academic API (v2). The queries I used followed the general structure: “({{SEARCH TERM}} OR #{{SEARCH TERM}}) place_country:{{COUNTRY}}”, where {{SEARCH TERM}} was replaced with one of the curated search terms and {{COUNTRY}} was replaced with the International Organization for Standardization (ISO) 3166-1 alpha-2 code of the relevant

country. If a term had spaces, a third # was included without any spaces. Using the search term `social distancing` as an example for the UK, the following search query is constructed: “(social distancing OR #social distancing OR #socialdistancing) place_country UK”. For every search term, I downloaded all tweets between 2020-01-01 and 2022-01-30 (inclusive) at a daily resolution.

In total, I used 863 unique search queries. Of these, eight search queries returned no tweets. Additional queries were subsequently excluded during filtering and cleaning, as detailed in Sections 4.5 and 4.6. To capture different dimensions of COVID-19 discourse, the search terms underlying these queries were categorised into three thematic groups: disease, policy, and vaccine. The disease category is a general category that includes both terms related to the disease and synonyms of the disease’s name. A complete list of categorised search terms present in the final dataset is given in Appendix B.4.

4.4 Downloading tweets

Twitter data offers utility beyond just the tweet text, with each tweet accompanied by useful metadata. I used `twarc`, a Python library, to interact with the Twitter Academic v2 Representational State Transfer (REST) API (Summers et al., 2023). Based on the JSON payload returned, and, by extension, Twitter’s structuring of the data, I parsed five main objects: `Author`, `Geo`, `Tweet`, `RefTweet`, and `NestedRefTweet`.

Each of these objects plays a distinct role in capturing the contextual and structural information of a tweet and is explained in more detail below. `Author` contains the information of the user that made the post and includes a sub-object, `AuthorPublicMetrics`, which describes details such as follower count and tweet count. `Geo` contains geographic information related to the tweet, and includes two sub-objects: `SubGeo`, which describes a bounding box, and `SubCoordinates`, which details an exact longitude and latitude. `Tweet` describes the tweet itself and includes two sub-objects: `PublicMetrics`, indicative of engagement; and `EditControls`, which indicates whether a tweet has been or can be edited. `EditControls` may be relevant if the same tweet is accessed at a later stage from the platform, as there is a possibility that the tweet has been edited.

The final two objects, `RefTweets` and `NestedRefTweets`, capture tweets that are

referenced by tweets returned from my search queries. Specifically, each tweet that matches a search term may reference another tweet. Furthermore, a referenced tweet may itself reference another tweet, creating nested references.

There are two types of reference tweets, namely replies and quote tweets. A reply is a response to another tweet, while a quote tweet is a tweet that quotes an existing tweet with added commentary. **RefTweets** captures the tweet that the **Tweet** object either replies to or quotes, while **NestedRefTweets** captures reference tweets related to the **RefTweets** object. A single tweet may reply to only one tweet, but it may quote multiple tweets, including in a reply. The **RefTweet** itself is a **Tweet** object, which I parse and add to the Tweet table. However, this tweet does not necessarily contain one of my search terms and is not necessarily geotagged. **NestedRefTweet** only contains an ID.

One challenge when parsing such a complex JSON payload is that empty fields are omitted entirely, which complicates the creation of a consistent tabular dataset. To address this, I defined a Pydantic data model that automatically assigns default values when fields are missing (Colvin et al., 2023). The structure of each of these objects, their subclasses, and the default values assigned during parsing are shown in Figure B.1. Using the Pydantic-parsed data, a final flat-file data structure is created, shown in Figure 4.2. It follows the same structure as the Pydantic model but flattens all sub-objects into their parent objects. Additionally, two new tables are created: a search query table (**search_query**) and a mapping table that links each tweet to its corresponding search query (**search_query_mapping**). The **search_query_mapping** table also indicates which reference tweets are linked to a specific search query; note that a reference tweet does not necessarily contain the search term related to the query for which it is returned. No language table is created, as each language ID is already represented as an ISO 639-1 code. All files are deduplicated so that each row in each table is unique.

My choice to create flat files was partly motivated by the goal of maintaining a distinct and portable data store. Rather than building an in-storage OLAP (Online Analytical Processing) database, I saved each table as a separate **.parquet** file and use DuckDB as an in-process OLAP system to query these individual files. An alternative approach would have been to use a NoSQL database, such as MongoDB (Tanner et al., 2023). However, I argue that deduplicating flat files is simpler, with this approach offering

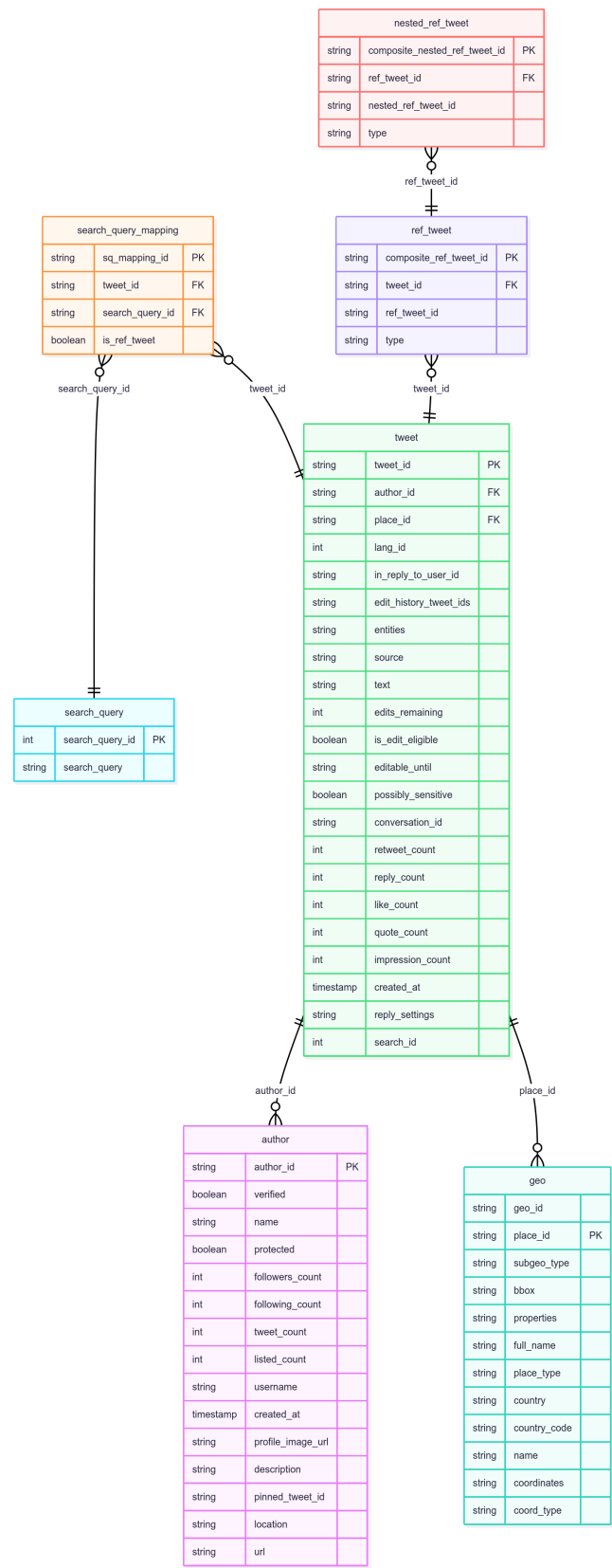


FIGURE 4.2: Schema of the flattened tabular dataset constructed from a Pydantic model of Twitter API v2 JSON responses.

comparable, and in some cases better, performance (particularly when indexes are not optimised). The whole data download process is illustrated in Figure 4.3. The Figure shows a `.csv` file as the final output to emphasise that a human-readable tabular dataset can be created using this process; however, I used `.parquet` files as a more efficient final format.

Using the Twitter Academic API v2, I collected a total of 7,352,025 unique tweets and their associated metadata between the period of 2020-01-01 and 2022-01-30, across 855 search queries.

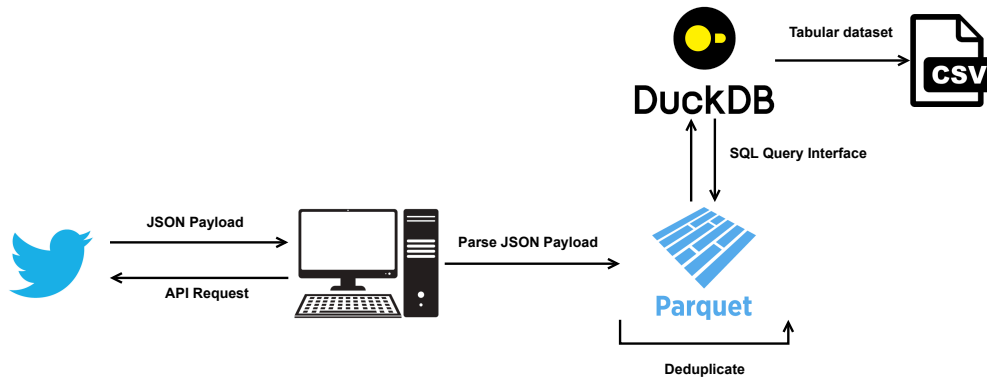


FIGURE 4.3: Schematic of process to download, deduplicate, and create a tabular dataset from Twitter.

4.5 Filtering tweets

In this Section I describe how I filtered tweets, the first of two broad processing steps I applied to the downloaded tweet data. I applied three main filtering steps, namely filtering reference tweets, filtering bots, and filtering by language. The impact of each step is shown in Figure 4.4 and is discussed in more detail in the following Sections.

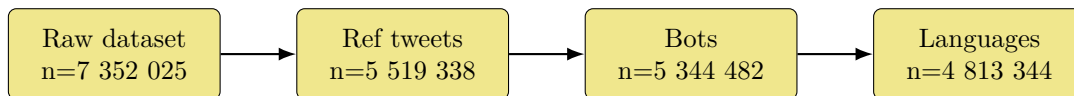


FIGURE 4.4: Flowchart displaying the process of filtering the dataset of raw tweets.

4.5.1 Reference tweets

I removed all reference tweets that did not match one of the search queries, either because they did not originate from one of the countries in this Thesis, did not contain one of the specified search terms, or both. In total, 25% of tweets fell into this category.

These removed reference tweets cannot be distinctly allocated to a specific country, as tweets from different countries may reference the same tweet. Consequently, a per-country breakdown of removed reference tweets is not provided.

4.5.2 Bots

Following the removal of reference tweets, I next removed problematic users. I performed this step prior to language filtering to ensure that accounts operating across multiple languages are excluded. I only considered users who are not verified (98.25% of users). These problematic accounts are assumed to be predominantly bots and are referred to as such throughout, although I acknowledge that they may also include human users exhibiting bot-like behaviour.

As bot detection is not the focus of this Thesis, I adopted a relatively simple approach. However, I recognise that more comprehensive classification methods have been proposed in the literature (Orabi et al., 2020). Using an agglomerative clustering approach, I aimed to identify tweets that are similar to one another, per author, per day (author-day). If the number of similar tweets exceeds a given threshold, I considered the account problematic. My approach to detecting similar tweets and removing problematic ones depended on the number of tweets an author had posted per day. Both aspects are described in detail below.

I considered daily sets of tweets where an author had posted more than once. For each set, I applied the following process. First, I cleaned the tweets by removing links, user mentions, numbers, stop words, and punctuation. I then applied stemming. Next, I constructed a simple BoW and vectorised each tweet using the resulting vocabulary. From these feature vectors, I computed the cosine similarity for each pairwise combination. Both feature vectors and the BoW count vectoriser are discussed in more detail in Section 2.4. Cosine similarity between tweets i and j is defined as: $S_C(\mathbf{t}_i, \mathbf{t}_j) = \frac{\mathbf{t}_i \cdot \mathbf{t}_j}{\|\mathbf{t}_i\| \|\mathbf{t}_j\|}$, where $\mathbf{t}$ is feature vector for a tweet. That is, the dot product of the feature vectors divided by the product of their norms. Applying this across all tweet pairs produces a similarity matrix per user per day.

To generate clusters, I converted the similarity matrix into a distance matrix by computing $1 - S_C(\mathbf{t}_i, \mathbf{t}_j)$ for each element. I then applied agglomerative hierarchical clustering using the Unweighted Pair Group Method with Arithmetic Mean (UPGMA)

method, which iteratively merges the pair of clusters with the smallest average pairwise distance to determine cluster linkage (Sneath and Sokal, 1973). I used a low distance threshold to define cluster membership, representing a high degree of similarity. Finally, I calculated the proportion of tweets that belong to clusters with more than one member. Algorithm 1 summarises the process.

Algorithm 1 Tweet Clustering Process

Require: Two or more tweets for a single author on a single day

```

1: for each author do
2:   for each day do
3:     Collect tweets for the author on that day
4:     Remove links, user mentions, numbers, stop words, and punctuation
5:     Apply stemming
6:     Construct a BoW vocabulary
7:     Vectorize the tweets using the BoW
8:     Compute pairwise cosine similarity ( $S_C$ ) between tweet vectors to create a
       similarity matrix
9:     Convert similarity matrix to distance matrix ( $1 - S_C$ )
10:    Perform agglomerative hierarchical clustering
11:    Use distance threshold to cut the dendrogram and form clusters
12:    Determine the proportion of tweets in clusters with more than one member
13:  end for
14: end for
15: return Clusters and proportion of tweets in multi-point clusters

```

I applied this process to three groups of users, categorised based on the total number of tweets per author-day. This classification operates under the assumption that accounts with higher daily tweet volumes are more likely to exhibit suspicious behaviour and should therefore be subject to stricter filtering criteria. The first group included users who posted 50 or more times per day. This threshold was based on the work of Bradshaw et al. (2020), who analysed approximately 22 million tweets related to political conversations during the 2016 U.S. presidential election and the 2018 State of the Union address. They considered any account posting more than 49.5 times in a single day using identified political hashtags to be unlikely to be human. Such accounts were classified as amplification accounts, which they defined as accounts that deliberately increase either the volume of tweets on particular topics or the attention those topics receive. The remaining two groups included users who post between 10 and 49 times per day and those who post between 2 and 9 times per day².

²Initially, I grouped users by ranges of ten tweets per day. However, due to the relatively sparsity in the 10–49 range compared to the 2–9 range, I opted to combine this range into a single category.

Cluster attribution was determined using a distance threshold, which was adjusted based on the group classification. A higher threshold requires less similarity between tweets for them to be considered part of the same cluster. I applied a stricter threshold of 0.25 to the group with 50 or more tweets per day, based on the assumption that these accounts were more likely to be suspicious and should therefore be subject to tighter clustering constraints. In contrast, I applied a more relaxed threshold of 0.1 to the remaining groups. Note that, in the context of clustering, a higher threshold is generally regarded as more relaxed since it allows tweets at larger distances to be grouped together. In this instance, however, because a greater number of similar tweets increases the likelihood of exclusion, a higher threshold functions as a stricter criterion. Details for each group are provided in Table 4.1, and a complete example of the process applied to the ≥ 50 per day group is included in Appendix B.2.1.

Group	Size	Authors	Tweets	Distance
tweets ≥ 50	425	97	41 110	0.25
$10 \leq \text{tweets} < 50$	12 612	3 416	189 473	0.1
$2 \leq \text{tweets} < 10$	681 047	151 810	1 827 038	0.1

TABLE 4.1: Groups to which the bot detection process was applied. Column description: Size - the count of unique user–day combinations associated with a tweet count specified by their group membership, and Distance - the distance threshold, points with distances below this threshold are grouped together.

Following the clustering process, I applied the group-specific distance thresholds to identify accounts exhibiting problematic behaviour. Figure 4.5 presents the resulting distribution of author–days by the proportion of similar tweets under these thresholds. The x-axis indicates, for each author on a given day, the proportion of tweets classified as similar, while the y-axis represents the relative frequency of author–day observations across groups. Notably, there was a marked difference in the number of highly similar tweets (90%–100%) across the groups. Users who posted fewer than 50 tweets per day exhibited a lower proportion of highly similar tweets compared to those in the group of 50 or more tweets per day. Conversely, they exhibited a higher proportion of dissimilar tweets (0%–10%).

The similarity profile of the ≥ 50 per day group is interpreted as indicative of automated or bot-like behaviour. As a result, the divergence in similarity distributions across groups suggests that detecting bots becomes substantially more challenging among users who post fewer than 50 tweets per day.

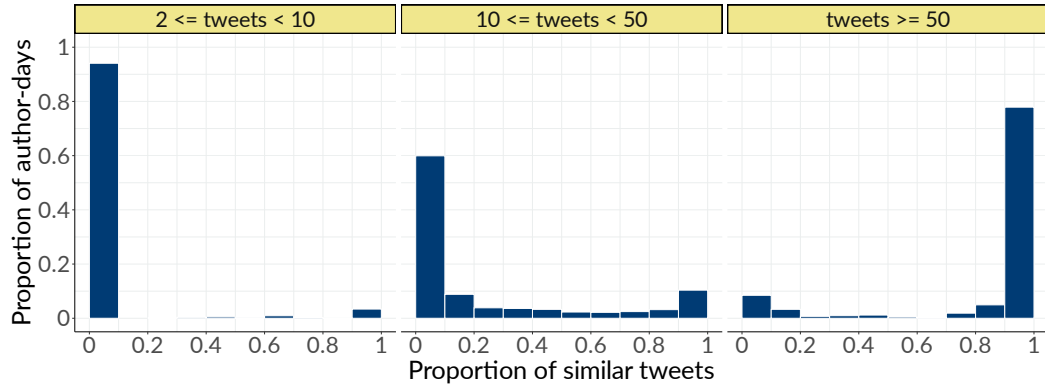


FIGURE 4.5: Histogram of the relative frequency of author-days across similar tweets, by posting frequency group.

To test whether the higher distance threshold applied to the ≥ 50 per day group biased the results, I repeated the analysis using a uniform threshold of 0.1 across all groups. The overall pattern of dissimilarity persisted under this stricter setting, as shown in Figure B.3, which compares the two thresholds. With the uniform threshold, the proportion of authors in the 90–100% similarity range decreased from 78% to 69%, with most of the difference shifting to the 80–89% range.

Given the differences in author proportions and tweet similarity distributions across groups, I adopted group-specific filtering strategies. For the ≥ 50 per day group, I removed all tweets from any account exhibiting a non-zero proportion of similar tweets. This approach removed 96 authors (99%) from this group, resulting in the deletion of 88,498 tweets. When applying the lower threshold of 0.1, five fewer authors and 4,137 fewer tweets were removed.

For users who posted fewer than 50 tweets per day and had two or more similar tweets, I deduplicated tweets by retaining only one representative per cluster. I prioritised tweets that were not replies and that were written in one of the official languages under consideration. While this approach may retain some bot-generated tweets, it represents a more conservative strategy that minimises the risk of discarding human-authored content. After removing tweets from amplification accounts, I reduced the number of duplicate tweets from 104,149 to 40,775 distinct tweets (a 60% reduction) in the 2–9 tweets per day group, and from 32,523 to 6,059 distinct tweets (an 81% reduction) in the 10–49 tweets per day group.

4.5.3 Languages

The final filtering step I applied was to remove all non-official languages. For each country, I collected tweets based only on search terms in its official language, as noted in Section 4.2. However, this did not restrict the returned tweets to a single language, particularly since some language-agnostic terms such as `covid-19` were used. The language composition of the raw dataset after removing reference tweets, is shown in Table B.2.

Although I ultimately used only the official language, I did consider including additional languages. English emerged as the leading candidate, as it was the most common secondary language across all non-English countries except Spain, where Catalan occupied this position. This constituted a special case, and I recognise this exclusion as a limitation. However, incorporating additional languages would have necessitated translating the search term list multiple times for each country and required translation expertise, since some terms are not easily rendered in other languages. Moreover, restricting collection to a single language per country simplified the development of sentiment classifiers, as it avoided the complexity of multilingual training. The process of training sentiment classifiers is described in detail in Chapter 5.

In total, this step removed 531,138 tweets. Additionally, filtering languages removes 12 search queries. These correspond to terms with only a small number of associated tweets. The combined impact of all three filtering steps reduced the total dataset size by 35%, resulting in 4,813,344 unique tweets made by 626,122 distinct authors across 843 search queries.

A breakdown of the number of distinct tweets and those removed, by country, is provided in Table 4.2. The second processing step is to create an analysis dataset, which is discussed in the next Section.

4.6 Creating an analysis dataset

My goal in creating the analysis dataset was to capture the level of public attention paid to a topic, represented by a search term, which could then be stratified by sentiment. Using the final collection of distinct tweets, I matched each tweet to a search query. Notably, a single tweet could contain multiple search terms and,

Country	SQs	Authors	Tweets	BT removed	BT deduplicated	Languages
United Kingdom	113	310 256	2 426 693	24 080	35 582	93 797
Spain	132	133 594	985 629	23 798	19 256	153 399
France	109	70 795	484 251	8 195	9 419	62 180
Italy	115	43 140	356 369	9 904	11 655	78 088
Germany	120	32 295	239 491	1209	5 271	74 261
Netherlands	101	25 285	198 680	21 310	3 509	45 743
Portugal	89	13 250	74 096	2	1 020	13 233
Poland	76	6 108	48 135	-	1 136	10 437
Total	843	626 122	4 813 344	88 498	86 358	531 138

TABLE 4.2: Summary of the distinct dataset after the filtering process, stratified by country. Note that the total row displays the distinct counts and proportions across the entire dataset and, consequently, may not equal the sum of individual country values.

Column description: SQs - search queries; BT removed - count of tweets removed due to being assumed to be a bot account; BT deduplicated - count of tweets deduplication due to being similar to one or more tweets on a given data, possibly originating from a bot; and Languages - tweets removed due to not being in the official language of the country.

consequently, could match multiple search queries. This matching process led to the intentional duplication of some tweets in the dataset. Applying this process inflated the distinct dataset by 22%, resulting in a total of 5,894,970 rows.

However, this inflation also led to unintentional duplication in the form of plurals, subset terms, and catch-all terms. A schematic of each type of duplication and the total amount of tweets removed is shown in Figure 4.6.

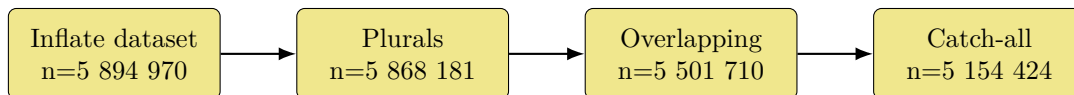


FIGURE 4.6: Flowchart displaying the process of creating an analysis dataset.

Furthermore, an example of each type of duplication is shown in Figure 4.7a, which I use to guide the discussion of each case and my approach to addressing it in the subsequent Sections.

4.6.1 Plurals

The first type of duplication I addressed was plurals. As shown in Figure 4.7a, both vaccine and vaccines are matched by this tweet. To address this, I mapped all plural search terms to their corresponding singular forms. In cases where a tweet contained both the singular and plural forms, as in the example, I removed the plural attribution rather than mapping it. In some instances, this process was not possible because

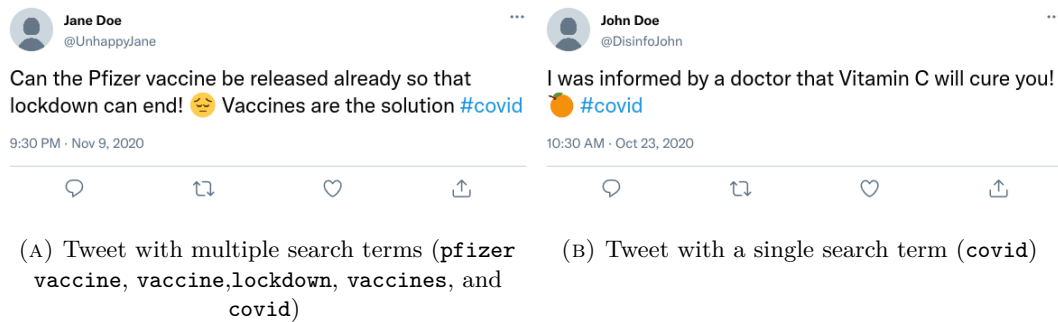


FIGURE 4.7: Examples of tweets with multiple search terms (A) and a single search term (B).

either the plural or the singular form was omitted from the dataset. This omission may have occurred because it was not part of the original search, no tweets were returned for the search query, or it was filtered out during the process described in Section 4.5.

In total, 128 plural terms were mapped. The complete list of mapped terms is given in Appendix B.5. In a small percentage of cases (0.05%), both the singular and plural term were contained in the same tweet. This resulted in the removal of 26,789 tweets.

4.6.2 Overlapping terms

The next type of duplication I addressed was the use of overlapping terms. These occurred when a shorter term, which was part of a more detailed phrase or compound word, matched the same tweet. It also included cases where terms only differed by punctuation (of which compound words are a special case). In such cases, I attribute the tweet to the more detailed search term and remove the attribution to the shorter term. In Figure 4.7a, an example of overlapping terms is shown with **pfizer vaccine** and **vaccine**.

In total, I identified 76 terms that overlapped with 134 other terms. In only one instance, was a search term completely contained in another: the Portuguese term **variante delta** → **variante delta+**. Overlapping terms were more common compared to plurals, with 6.2% of tweets overlapping³. This process resulted in the removal of 366,741 tweets.

³This percentage was calculated relative to the number of tweets remaining after the removal of plurals, not the initial inflated dataset.

4.6.3 Catch-all terms

The final type of duplication I address was catch-all terms. I used the term catch-all to refer to general synonyms for the name of the disease, such as `covid-19` and `coronavirus`. The complete list of catch-all terms is provided in Table B.20. These terms were intended to capture tweets that did not reference a more specific search term.

Consequently, a tweet was only attributed to a catch-all term if it did not include any other non-catch-all search terms. Figure 4.7 shows two example tweets that include the search term `covid`, illustrating how it serves as a catch-all in one case (B) but not in the other (A). In Panel 4.7a, the presence of other, more specific (non-catch-all) search terms means that the tweet would not be attributed to the `covid` catch-all term. In contrast, in Panel 4.7b, where `covid` is the only matched search term, the tweet would be attributed to it. In total, similar to overlapping terms, 6.3% of tweets included both a catch-all and non-catch-all term. This resulted in the removal of 347,286 tweets.

As an additional output, after filtering catch-all terms, I examined the extent of overlapping search terms. Across all countries, 93.5% of tweets matched a single term, 5.9% matched two terms, and the remaining tweets matched between three and seven terms. This pattern was consistent at the country level. Exact percentages per country are shown in Table B.3. When inspecting the nature of these overlaps, I found that 56% were due to tweets containing multiple catch-all terms. While it would be more accurate to remove these duplications, they account for only 59,255 tweets (a small fraction of the dataset). As such, I do not expect this to have a notable impact on the overall analysis.

4.7 Final dataset

In summary, matching the distinct dataset to all remaining search queries inflated the number to 5,894,970 tweets. Deduplication in the form of mapping plurals, overlapping terms, and catch-all terms removed 740,546 tweets and 128 search queries. This results in a final dataset of 5,154,424 tweets, corresponding to a distinct set of 4,813,344 tweets, 626,122 authors, and 714 search queries.

A breakdown of the final dataset is shown in Table 4.3. Additionally, Figure 4.8 presents the seven-day MA of tweet counts by search term category. Vaccine-related tweets are included from the beginning of the observation period; however, note that tweets mentioning vaccines prior to 2020-11-09⁴ may not specifically relate to COVID-19. The grey dashed line in the Figure indicates this date. Note that, across all countries, there is a spike in vaccine-related tweets following this date. Where relevant, I indicate how these tweets are filtered in subsequent analyses. Except Chapter 5, this final filtered and categorised dataset forms the basis of the analyses presented in this Thesis.

Country	SQs	Authors	Tweets	Plurals	Overlapping	Catch-all
United Kingdom	102	310 256	2 580 095	14 297	236 802	180 147
Spain	103	133 594	1 071 545	5 367	12 990	69 990
France	94	70 795	516 705	3 817	48 183	34 970
Italy	101	43 140	381 396	608	30 301	22 983
Germany	99	32 295	262 595	913	15 364	21 201
Netherlands	79	25 285	211 845	1 254	9 785	11 925
Portugal	72	13 250	75 909	299	7 807	2 519
Poland	64	6 108	54 334	234	5 239	3 551
Total	714	626 122	5 154 424	26 789	366 471	347 286

TABLE 4.3: Summary of the analysis dataset after deduplication, stratified by country. Note that the total row displays the distinct counts and proportions across the entire dataset and, consequently, may not equal the sum of individual country values.

Column description: SQs - total number of search terms, Plurals - count of tweets removed with both a plural and singular term, Overlapping - count of tweets removed with overlapping search terms, and Catch-all - count of tweets removed with both a catch-all and non-catch-all search term.

4.8 Discussion

In this Chapter, I outlined my approach to constructing a dataset of tweets that reflects public attention to specific topics. I described the structure of the objects returned by the Twitter API and my method for reliably parsing its complex, nested format. To accurately capture public attention, I expanded a distinct set of tweets by matching them to relevant search terms, and carefully deduplicated the resulting dataset where appropriate.

My primary motivation for creating a custom dataset, rather than relying on existing ones, was the ability to define my own search terms. The low proportion of overlap

⁴The date that the Pfizer-BioNTech COVID-19 vaccine received FDA approval.

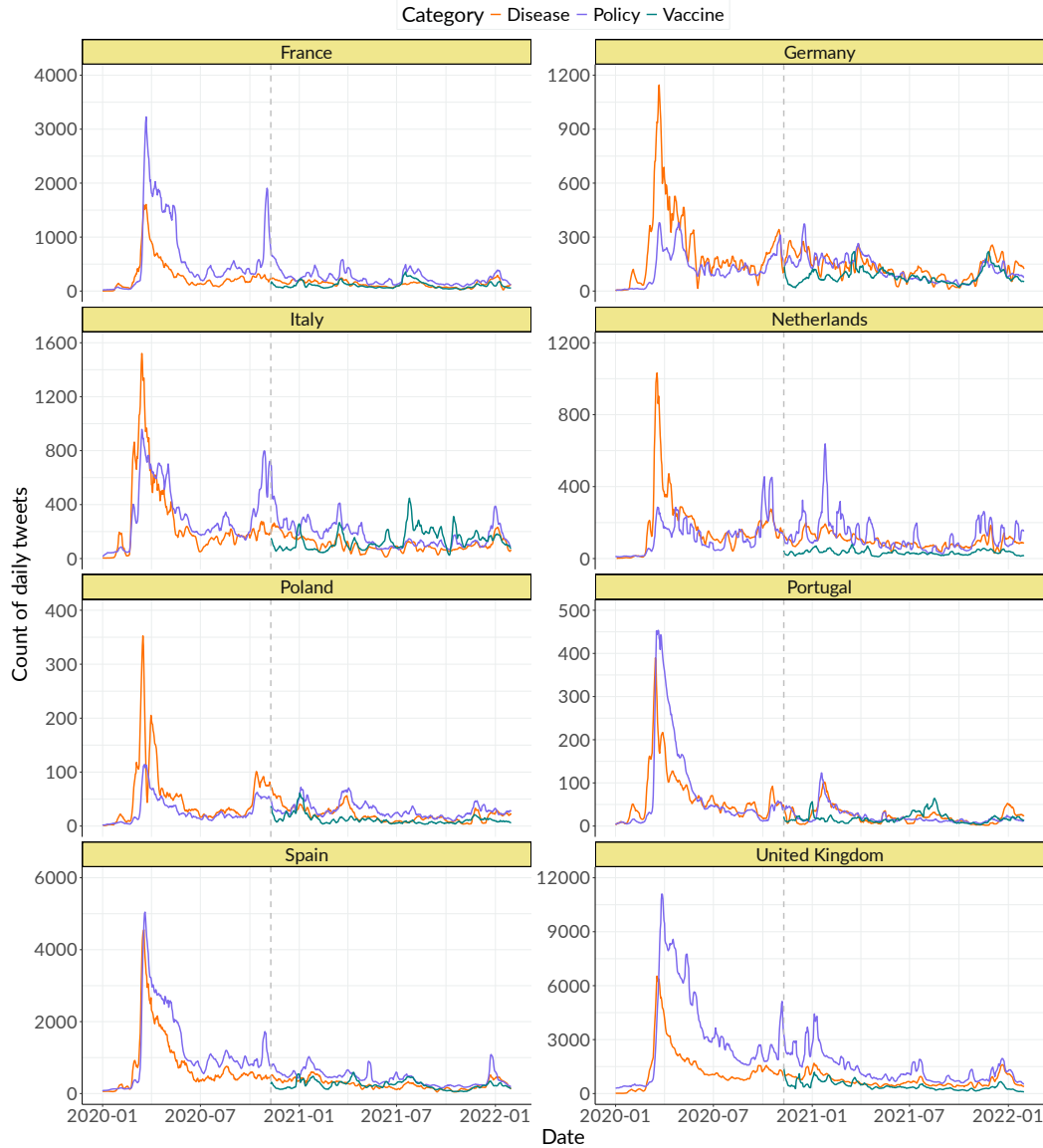


FIGURE 4.8: Seven-day MA of daily tweet count stratified by search term category per country.

between both overlapping terms and catch-all terms supports my argument that many of the search terms I used are not included in other datasets, which used a smaller subset of terms. The limited scope of search terms in many available datasets highlights a central theme of this Thesis: interdisciplinarity is essential to avoid missing domain-specific depth and the valuable insights it provides.

By creating my own dataset, I was able to develop an efficient method for interacting with tweet data. Instead of relying on a NoSQL approach, I parse the data into `.parquet` files and query them using DuckDB. This use of an in-process OLAP database offers advantages for reproducibility, as a central versioned store can be

queried directly with simple SQL scripts to recreate raw datasets, avoiding the need to duplicate data. This approach is particularly valuable in settings, where computational resources may be limited. For large-scale datasets, it offers a practical and efficient alternative to more complex infrastructure.

A significant challenge in curating the Twitter dataset was the discontinuation of the Academic Twitter API. Twitter has long been a popular platform for researchers, owing to its accessibility for data collection and analysis (Tsao et al., 2021; Pilipiec, Samsten, and Bota, 2023). This openness was exemplified during the COVID-19 pandemic, when Twitter launched the COVID-19 stream and substantially relaxed rate limits, enabling the creation of datasets at the billion-tweet scale (Adam, 2020). However, in February 2023, the API was officially discontinued⁵, preventing any further augmentation of my dataset, including the correction of issues such as an incorrectly specified end date (Blakey, 2024).

Nonetheless, the methods developed in this Thesis remain relevant. I anticipate that new platforms supportive of research, such as Bluesky, will continue to emerge (Bluesky Documentation, 2025). Moreover, the techniques described here apply more broadly to any context involving complex, nested data structures, beyond the scope of social media alone.

⁵The announcement was made in 2023, with access phased out throughout the year.

5 Identifying the best approach to multilingual sentiment classification and practical recommendations

5.1 Overview

Sentiment analysis has demonstrated potential for inclusion in public health research, especially during dynamic events such as the COVID-19 pandemic. Despite recent advances in NLP, many public health social media analyses still rely on traditional methods such as lexicons. In this Chapter, I compare three classes of sentiment analysis models, lexicons, encoder LMs, and LLMs, using a curated multilingual dataset of 2,530 annotated tweets spanning eight languages. Within the LLM class, I explore a range of prompting strategies and evaluate their performance using the F1 score.

5.2 Introduction

Transformers have accelerated growth in NLP. The evolution of NLP and recent developments were reviewed in Section 1.7 and expanded on in Chapter 2. These developments have led to a new paradigm of general-purpose pretrained LMs that can be fine-tuned for specific tasks. This began with BERT-based encoder LMs, but has since evolved into the era of LLMs (Devlin et al., 2019). Until recently, LLMs required considerable resources and technical expertise to use. However, the ability to prompt a model and align its output with human expectations has ushered in new possibilities for interdisciplinary research, where an optimal balance between performance and ease of use may now be achievable.

Sentiment analysis is a task with a wide range of interdisciplinary applications, often with a focus on social media data. However, advancements have not necessarily propagated across domains, where more straightforward and more accessible approaches are still common (Rodríguez-Ibáñez et al., 2023). In their opinion piece on interdisciplinary sentiment analysis practices, Öhman (2021) focused on lexicon-based

approaches and emphasises that accuracy is not the sole criterion. Indeed, in interdisciplinary contexts, there is often the need to balance ease of use and performance.

During the COVID-19 pandemic, social media served as a tool for expressing opinion and disseminating information (Karhu, Suoheimo, and Häkkinen, 2021; Delogu et al., 2025; Szeto, Au, and Cheng, 2024). This increased interest in the utility of social media in a public health context, with a myriad of applications (Terry et al., 2023; Tsao et al., 2021), including sentiment analysis. As will be discussed in Section 5.3.1, simple approaches are often still used, with LLMs underexplored. Given the globally unified focus and the extent of social media engagement during the pandemic, COVID-19 social media data is a strong candidate for comparing the latest techniques in multilingual sentiment analysis.

Using a manually curated dataset of COVID-19-related sentiment, I empirically evaluated a range of models that fall into the following model classes: lexicons, encoder LMs, and LLMs. I considered these classes either due to their strong performance on social media benchmark tasks or their previous application in public health (Braig et al., 2023; Antypas et al., 2023). Additionally, within LLMs, I explored optimal prompting strategies for sentiment analysis, including whether prompts should be given in English or the language of the input data.

In this Chapter, I aimed to identify the most accurate model for classifying sentiment to apply to the larger analysis dataset described in Chapter 4. Given the continued reliance on simpler sentiment classification methods in the interdisciplinary literature, I also aimed to provide practical guidance and recommendations on model selection and usage for non-expert users, distinguishing these from expert NLP users, drawing on my experience applying these models throughout the analysis. Specifically, I addressed the research questions listed below.

1. Which models are best for classifying multilingual Twitter sentiment across the following model classes: lexicons, encoder LMs, and LLMs (both open-source and closed-source)?
2. What is the most effective prompting approach for multilingual sentiment analysis using LLMs?
3. What recommendations can help non-expert users understand key considerations when choosing between models, and make these models easier to use and

more practical to apply?

By addressing these questions, my research contributes to the literature by comparing different model classes for sentiment analysis in a multilingual setting, with a particular focus on LLMs which remain underexplored in this context. It also contributes by offering recommendations for non-expert users, outlining key considerations to support practical applications and guide the choice of the most appropriate model for a given task.

5.3 Related work

5.3.1 Social media sentiment in public health

In their systematic review of 28 papers examining sentiment analysis across various infectious diseases from 1 January 2010 to 30 June 2020, Alamoodi et al. (2021) found that 36% of the studies employed lexicon-based methods, including four of the six COVID-19-related papers they reviewed. The remainder used either ML methods, hybrid approaches blending ML with lexicon-based methods, or fell into a residual category. Similarly, Braig et al. (2023), focusing specifically on COVID-19, reviewed 40 papers published between October 2019 and January 2022. They found that 43% used deep learning and 40% used ML-based methods, with the remaining 17% used lexicon-based methods. Furthermore, they reported that 95% of the studies focused on English-language text. Although there are examples of multilingual sentiment analysis studies in the public health literature, they remained limited (Machado Júnior et al., 2023).

The systematic reviews considered above predated the widespread adoption of LLMs, which likely explains their absence. However, even when extending the scope to literature published after these reviews, only one comprehensive application of LLMs focused on COVID-19 was found. Espinosa and Salathé (2024) classified public stances toward vaccination using LLMs and demonstrated their superior performance compared to VADER (introduced in Section 2.3), which is widely used in public health social media applications, as highlighted by Alamoodi et al. (2021). In a related public health application, Elmitwalli et al. (2024) showed that LLMs accurately classified sentiment in synthetically generated tweets related to tobacco and e-cigarette use.

5.3.2 Modern sentiment analysis

While the application of LLMs to sentiment analysis in public health remains sparse, there have been many general investigations, with the specific task varying in complexity (Zhong et al., 2023; Wang et al., 2024).

Topal, Bozanta, and Basar (2024) evaluated both off-the-shelf and QLoRA fine-tuned versions of three open-source LLMs against encoder LMs on a variety of benchmark tweet classification tasks, using zero- and few-shot prompting. They found that Mistral-7B and LLaMA3-8B delivered the strongest performance, and that QLoRA-based models performed comparably to those that underwent full fine-tuning.

In a broader investigation of the different types of sentiment analysis tasks, Zhang et al. (2024) conducted a systematic review and compared LLMs to naive baselines and domain-specific fine-tuned Small Language Models (SLMs). Their findings indicated that LLMs performed well in zero-shot settings for simpler tasks and, within the confines of their experiment, consistently outperformed SLMs in few-shot settings. However, they noted the limitation that all datasets in their review were in English, and called for more multilingual and multicultural investigations to improve understanding of LLM performance.

While sentiment classification tasks in English have dominated the field, multilingual studies have also been conducted. Some research has empirically compared LLMs and assessed their capacity to be fine-tuned for other languages (Filip, Pavlíček, and Sosík, 2024; Buscemi and Proverbio, 2024). Furthermore, broader investigations have examined the general performance of LLMs in multilingual contexts. For example, Vajjala and Shimangaud (2025) examined both zero- and few-shot prompting schemes across a diverse set of tasks and compared LLMs to various models, including encoder LMs. Their experiments were conducted on a multilingual dataset spanning multiple tasks. For sentiment classification, they found that zero-shot prompting performed best within the constraints of their study. In contrast, few-shot prompts and fine-tuned models yielded stronger performance on other tasks.

Zhu et al. (2024) compared the cross-lingual transfer capabilities of LLMs, both open- and closed-source, with SLMs (which included encoder LMs) across four languages, in the context of sentiment classification on human conversational transcript data. Their study included both English-only and multilingual prompts. They observed

that SLMs outperformed open-source, English-centric LLMs in the zero-shot setting, although the latter adapted well under few-shot prompting. Moreover, they found that variants of ChatGPT, the closed-source LLM used in their study, performed best in the zero-shot setting, but were outperformed by fine-tuned open-source models in the few-shot setting.

Similarly, Kanayama et al. (2024) constructed multilingual prompts consisting of English instructions and input in the target language. To support the prompt, they augmented it with syntactic and lexical cues by adding a hint when a polar word was detected in the input; this was determined using an external sentiment classification tool (Kanayama and Iwamoto, 2020).

Collectively, these studies underscore the additional complexity of multilingual sentiment classification, particularly the extra care required in prompt design. Moreover, in many cases, this literature demonstrates the potential of LLMs as sentiment classifiers and highlights the need for further work applied to diverse datasets as support.

5.4 Data

To evaluate model performance, I manually curated an ad hoc annotated dataset from the larger collection of 4,813,344 filtered, distinct tweets. The larger dataset is described in more detail in Section 4.5 and its country-stratified composition is given in Table 4.2.

I adopted a stratified sampling approach in which tweets were sampled according to the distribution of search term volume per country in the larger dataset. As a consequence of this approach, not all search terms were included in this dataset. All search terms included are highlighted in search term tables in Appendix B.4. The resulting dataset was then annotated, and this is described in more detail in the next Section.

Initially, I sampled 463 English tweets from the UK, selecting 0.025% of the total tweets per search term. These tweets were annotated by three native English speakers, each with varying levels of knowledge about the COVID-19 pandemic. The annotators followed guidelines and categorised the tweet polarity to one of either negative, neutral, or positive.

All annotators agreed that three tweets were not in English. Of the remaining 460 tweets, complete agreement on sentiment categories was reached for 59% of those considered. Interrater reliability was measured using Fleiss’s κ , where $\kappa = 1$ reflects perfect agreement, $\kappa = 0$ agreement at chance level, and $\kappa < 0$ agreement worse than chance (negative values are rare in practice) (McHugh, 2012). For all three annotators, overall κ was 0.57, indicating moderate agreement. Agreement varied by class, with κ values of 0.70 for negative, 0.38 for neutral, and 0.62 for positive sentiment.

The preceding measures reflects agreement within specific annotator pairs or among all three annotators across the full set of tweets. However, when considering agreement regardless of which annotators agreed, agreement was observed for 97% of tweets. I considered each tweet as in agreement if at least two annotators selected the same category, and the sentiment label was assigned by the corresponding majority vote. This left 14 tweets with no agreement, where all three annotators assigned different labels (negative, neutral, and positive). In addition, annotators were asked to label whether a tweet was related to COVID-19 and whether it was UK-related, which led to further disagreement on 16 tweets. Because the annotators were unable to discuss their choices due to logistical constraints, the 30 tweets for which there was no agreement were excluded from the analysis.

For the remaining seven languages, only a single annotator per language was available to assist. In most cases, the annotators were the same as those who translated search terms. Where this was not the case, the annotators had the same profile as those who translated, as described in Section 4.3, namely native speakers with public health expertise and local knowledge of the countries involved. Given the time constraints faced by the annotators, I chose to sample a fixed set of 300 tweets per language, rather than fixing the percentage of tweets per search term. See Appendix C.1.2 for the instructions and details of the annotators.

In total, there are 2,530 tweets, 430 English tweets, and 300 per remaining language. The composition of the annotated dataset is shown in Table 5.1.

Language	Country	NT	ST	Neg	Neu	Pos
Germany	German	300	28	0.46	0.28	0.26
France	French	300	36	0.47	0.30	0.24
Italy	Italian	300	38	0.40	0.51	0.09
Netherlands	Dutch	300	29	0.38	0.40	0.22
Poland	Polish	300	21	0.52	0.27	0.22
Portugal	Portuguese	300	23	0.49	0.38	0.13
Spain	Spanish	300	28	0.46	0.28	0.26
United Kingdom	English	430	34	0.39	0.27	0.34
Total	-	2530	197	0.45	0.33	0.22

TABLE 5.1: Summary of the annotated dataset by country. Note that the total column shows the distinct number of search terms. Column descriptions from NT (left to right): number of tweets; number of search terms; and proportion of negative, neutral, and positive tweets.

5.5 Methods

To investigate which model is best suited for classifying multilingual Twitter sentiment and to provide practical recommendations for model selection, I considered three broad classes of models: lexicon-based approaches, encoder LMs, and LLMs. Within each class, I considered off-the-shelf models and, for encoder LMs and LLMs, I also considered fine-tuned versions. In the literature I reviewed, both in Section 1.5 and Section 5.3 these model classes are widely applied in sentiment analysis, with lexicon-based approaches and encoder LMs more strongly represented than LLMs in public health research.

5.5.1 Lexicon

I employed a single lexicon model, VADER, using the same classification thresholds as the original authors (Hutto and Gilbert, 2014). However, I acknowledge that others, such as Espinosa and Salathé (2024), have used different thresholds, and that this is something that could be optimised in future work. For non-English tweets, I used an English lexicon and translated the tweets into English using the Google Translate API.

5.5.2 Encoder

Model details

In Chapter 2, I introduced the two social media-specific encoder LMs used here, Bernice and XLM-T, which represent the fine-tuned and trained-from-scratch domain-specific settings found in the literature. I used two versions of XLM-T. The first, which I denote as XLM-T_{SENT}, is an XLM-R model trained on approximately 198 million tweets and fine-tuned for sentiment analysis (Loureiro et al., 2022). The sentiment fine-tuning included six of the eight languages I consider (English, French, German, Italian, Portuguese, and Spanish). The second, which I denote as XLM-T_{LARGE}, has the same base but was retrained on over 1 billion tweets from various languages up until December 2022. Bernice is also a multilingual XLM-R model. However, rather than fine-tuning an existing LM, it was trained from scratch on 2.5 billion tweets, using a tweet-specific tokeniser. To use Bernice and XLM-T_{LARGE} in a sentiment classification setting, I fine-tuned the models using a classification head (as described in Section 2.8).

Fine-tuning details

To optimise classification performance, I used RayTune for hyperparameter tuning, implementing a 5-fold cross-validation scheme, with an 80:20 train and validation split per fold, to identify the learning rate, batch size, and number of epochs that maximised the validation F1 score (Liaw et al., 2018). A random search was employed as the sampling algorithm from the search space given in Table 5.2. I conducted 25 trials across this search space per fold. For comparison, I used the optimal 5-fold cross-validation scores for these models. Additionally, I fine-tuned the existing classification head of XLM-T_{SENT} using the same approach. Given the pre-existing classification head, this model serves as an off-the-shelf baseline against which I compared the effect of fine-tuning. For these models, I used a softmax activation function at the final layer of the classification head, and select the argmax of this output as the sentiment classification.

Parameter	Values
Learning rate	$\log \mathcal{U}(1e^{-6}, 1e^{-3})$
Batch size	[2, 4, 8, 16, 32, 64]
Epochs	[1, 15]
Random Seed	3141

TABLE 5.2: Hyperparameter search space for encoder models.

5.5.3 Large language models

Model details

In the LLM model class, I considered both open-source models and, due to cost-constraints, a single closed-source model. Among the open-source models, I evaluated three models: Llama 3 8B, Llama 3.1 8B, and Mistral v0.3 7B (Grattafiori et al., 2024; Jiang et al., 2023). For the latter, I chose ChatGPT with a `gpt-4o-mini-2024-07-18` base model as this was most comparable in size to the other LLMs used (Zeff, 2024). As LLMs outputs free text, I applied a post-processing rule to map responses to a sentiment class. I searched for the first match of either negative, neutral, or positive (across all languages), and if no match was found, I classified the tweet as neutral. To fine-tune the open-source LLMs, I employed QLoRA (Dettmers et al., 2023). I denote both all fine-tuned models by adding the superscript ^{FT} to their names.

Fine-tuning details

I adopted the following LLM fine-tuning scheme. Due to computational constraints, I limited my scope to a single Llama model (Llama 3 8B) as it performed better off-the-shelf on average compared to its successor. I again utilised RayTune for hyperparameter tuning, implementing the 5-fold cross-validation scheme (Liaw et al., 2018). I identified parameters specific to three broad groups: general optimisation (learning rate, batch size, and number of epochs); LORA (dropout, alpha, and R); and generation (LLM generation temperature¹). Given my downstream task of classifying sentiment polarity, I optimised the mean evaluation classification accuracy, favouring this over mean token accuracy due to the token overlap between my output words (both positive and negative share the same suffix across many languages). I determined the optimal hyperparameter configuration from the search space shown

¹The generation temperature controls the randomness of outputs from an LLM. Values range from 0 to 1, with lower values reducing diversity and creativity and consequently producing more deterministic results.

in Table 5.3 using a TPE, described in Section 3.6. I used the TPE implementation provided in the Optuna Python package (Akiba et al., 2019). Given the computational cost per trial due to the size of these models, I conducted only five trials per fold. The quantisation configuration is provided in Appendix C.4.1.

Parameter	Values
Learning rate	$\mathcal{U}(1e^{-9}, 1e^{-5})$
Batch size	[2, 4, 8]
Epochs	[5;15]
Lora Dropout	[0.0, 0.05, 0.1]
Lora Alpha	[8, 16, 32, 64, 128, 256]
Lora R	[8, 16, 32, 64, 128, 256]
Temperature	$\mathcal{U}(0, 1)$
Random Seed	3141

TABLE 5.3: Hyperparameter search space for LLMs.

Prompting strategy

These models were applied using three prompting strategies: zero-shot, few-shot, and a longer prompt matching the instructions given to the human annotators. I refer to this longer prompt as the **annotation** prompt. While the prompts varied in the amount of context provided, the general task for each model was to classify tweets as either negative, neutral, or positive. More details about the prompting strategies and the prompts themselves can be found in Appendix C.2.

To better understand the impact of tweet selection in few-shot prompts and to ensure a fair comparison with the encoder cross-validated accuracy, I adopted the following few-shot prompting scheme. Using the same cross-validation datasets employed for the encoder LMs, I selected country-specific negative, neutral, and positive tweets from the training set in each fold. I evaluated the model on the corresponding test set. These tweets were country-specific both in language and in context. Final performance metrics were computed as the average across test set results. I refer to these prompts as **dynamic**, as they rely on country-specific tweets that change across folds. As an illustrative example, the UK tweets used in the dynamic prompts for each fold are provided in Appendix C.2.4.

In the multilingual setting, I examined the impact of prompt language. For each prompt, the instructions were provided in both English and the country-specific language of the tweet being evaluated. In dynamic prompts, the example tweets always

remained in the country-specific language. Consequently, to disentangle the effect of instruction language from the effect of country-specific tweet content, I created a second version of the few-shot and annotation prompts. In this version, I used a fixed set of tweets from the UK, identical to those shown to the annotators. I refer to these prompts as **static**, since the same UK tweets were used for all countries. For the country-specific static prompts, the entire prompt (including the UK example tweets) was translated into the language of the tweets being evaluated. The English prompt variants are indicated by appending [en] to the prompt name or by adding it as a superscript. The complete set of prompts and their differences are shown in Table 5.4. Differences in run configurations for each prompt are provided in Appendix C.3.

Prompt name	Instruction	Tweet	Context
Zero-shot	country	-	-
Zero-shot [en]	English	-	-
Few-shot (static)	country	country	UK
Few-shot (static)[en]	English	English	UK
Few-shot (dynamic)	country	country	country
Few-shot (dynamic)[en]	English	country	country
Annotation (static)	country	country	UK
Annotation (static)[en]	English	English	UK
Annotation (dynamic)	country	country	country
Annotation (dynamic)[en]	English	country	country

TABLE 5.4: The different prompt combinations used. Instruction and Tweet refer to the language of each component, and context refers to whether the tweet used in the example is generic or country-specific.

As an additional assessment, I evaluated how the choice of example tweets affected model performance in the dynamic prompts. To contextualise these results and account for both fold-level variation and the inherent stochasticity of LLM generation, I used static prompts as a baseline. To generate a comparable result set, static prompts were run five times, with one fold sequentially selected from each run to create a static dataset of five folds. It should be noted that this static fold dataset was used only to examine the impact of example choice.

5.5.4 Model specifications

The number of parameters and supported languages for each model are listed in Table 5.5, which also illustrates the size disparity between LLMs and encoder models. Parameter counts are omitted for ChatGPT 4o-mini, as it is closed-source, and for

VADER, whose lexicon-based design has no trainable parameters. Language support for Mistral v0.3 is not documented. ChatGPT is reported to support more than 50 languages, though no exact figure is provided. For both Llama models, I have reported the supported languages, noting that the releases mention training on small samples of additional languages. For XLM-T_{LARGE}, the first number denotes the languages used to fine-tune the model, while the second reflects the base model’s coverage. For XLM-T_{SENT}, the first number refers to the languages used to train the classification head, and the second and third numbers correspond to the fine-tuned and base model coverage, respectively. Links to the exact implementations of each model are provided in Appendix C.7.

Model name	Language support	Number of parameters
Llama 3.1	8	8,701,349,888
Llama 3	English	8,597,540,864
Mistral v0.3	Unspecified	7,403,212,800
XLM-T _{LARGE}	30 100	559,893,507
Bernice	66	277 749 507
XLM-T _{SENT}	8 30 100	278,045,955
ChatGPT 4o-mini	50+	-
VADER	English	-

TABLE 5.5: Details of the models used. For fine-tuned multilingual models, vertical separators in the “Language Support” column indicate the base model’s coverage, with the leftmost value showing the final fine-tuned model’s coverage.

5.5.5 Evaluation

I evaluated model performance using accuracy, precision, recall, the F1 score, and the proportion of cross-over misclassifications between negative and positive classes. Each metric is defined below.

For a given class c (negative, neutral, or positive), precision, recall, and the F1 score are defined as follows:

$$\text{Precision}_c = \frac{TP_c}{TP_c + FP_c}, \quad \text{Recall}_c = \frac{TP_c}{TP_c + FN_c}, \quad F1_c = \frac{TP_c}{TP_c + \frac{1}{2}(FP_c + FN_c)}$$

where TP_c (true positives) is the number of tweets correctly predicted as class c , FP_c (false positives) is the number of tweets incorrectly predicted as class c , and FN_c (false negatives) is the number of tweets from class c incorrectly predicted as another class. Across these metrics, I report macro-averages, calculated as the unweighted

mean across the negative, neutral, and positive classes. By contrast, micro-averages weight classes by frequency and are dominated by the majority class, making them less suitable for my imbalanced dataset (see Table 5.1).

Using the notation above, and with N denoting the total number of tweets and neg , neu , and pos representing the negative, neutral, and positive classes respectively, accuracy is defined as:

$$\text{Accuracy} = \frac{TP_{neg} + TP_{neu} + TP_{pos}}{N}$$

In addition, I reported the proportion of cross-over misclassification between negative and positive classes. I denote the number of negative tweets misclassified as positive as $FN_{neg \rightarrow pos}$, and the number of positive tweets misclassified as negative as $FN_{pos \rightarrow neg}$. I used the entire sample as the denominator to enable comparisons across models. Based on this, the cross-over is defined as:

$$\text{Cross-over} = \frac{FN_{pos \rightarrow neg} + FN_{neg \rightarrow pos}}{N}$$

Given the small number of samples per country, I did not create a separate hold-out test set; instead, I used 5-fold cross-validated error as a proxy for generalisation (k -fold cross-validation was introduced in Section 2.8). This procedure is applied to dynamic prompts and to fine-tuned models. Static prompts were evaluated on the entire dataset², and averaged across five runs (conducted to assess variability in LLM performance, as described in the previous Section).

For fine-tuned models, k -fold cross-validation served both to tune hyperparameters and to estimate performance. Within this context, two standard implementations are commonly used, non-nested and nested. In the non-nested form, the same folds are used for both tuning and error estimation, whereas, in the nested form, an outer loop provides a dedicated test fold while the inner loop is used solely for tuning. Nested CV yields less biased estimates of generalisation error, but at substantially higher computational cost (Raschka, 2020; Varma and Simon, 2006). For example, in my setting, with 25 hyperparameter trials and five folds repeated five times, complete nesting would require 625 runs per model. I therefore adopted a pragmatic compromise, tuning hyperparameters within each fold (as described in Section 5.5.2)

²The English tweet examples used are not included in the dataset.

Model name	Prompt	F1	Acc	Prec	Rec	C-o
XLM-T _{LARGE} ^{FT} 560M	-	0.71	0.75	0.73	0.71	0.05
Mistral v0.3 ^{FT} 7B	Annotation (dynamic)[en]	0.70	0.72	0.71	0.71	0.04
Bernice ^{FT} 280M	-	0.70	0.74	0.71	0.70	0.06
ChatGPT 4o-mini	Annotation (dynamic)	0.69	0.73	0.74	0.69	0.02
XLM-T _{SENT} ^{FT} 280M	-	0.67	0.72	0.68	0.68	0.07
Llama 3.1 8B	Annotation (dynamic)[en]	0.67	0.71	0.70	0.68	0.05
Llama 3 ^{FT} 8B	Annotation (dynamic)[en]	0.66	0.68	0.67	0.69	0.06
Llama 3 8B	Annotation (dynamic)	0.65	0.68	0.69	0.68	0.05
Mistral v0.3 7B	Annotation (dynamic)[en]	0.63	0.66	0.65	0.65	0.04
XLM-T _{SENT} 280M	-	0.59	0.63	0.62	0.60	0.07
VADER	-	0.48	0.50	0.51	0.52	0.18

TABLE 5.6: Macro-averaged sentiment classification metrics across all runs of applying each model to the annotated dataset, sorted by F1. For each LLM, only results for the best performing prompt are shown. A “-” indicates that encoder LMs and VADER are not prompted. Columns from Acc (left to right): accuracy, precision, recall, and proportion of cross-over misclassification between negative and positive.

and reported the average validation performance across folds. This procedure may yield optimistic estimates, but as it is applied consistently across fine-tuned models, comparisons remain valid.

5.6 Results: the best performing model

My first research question examined which model was most accurate, as judged by a range of metrics, in classifying multilingual Twitter sentiment across different modelling approaches. The metrics shown are averaged across all runs or folds as applicable. Table 5.6 presents the top-performing models in descending order of their F1 scores. For clarity, only the best-performing prompt per prompting model are included in this comparison. Differences in performance across various prompting strategies are discussed in Section 5.7.

Overall, LLMs (F1 = 0.67) and encoder LMs (F1 = 0.67) performed similarly, with encoder LMs achieving an F1 score only 0.002 higher than that of LLMs. By contrast, VADER, the lexicon-based approach, was the poorest performer (F1 = 0.48). Its weak performance was mainly attributable to an overestimation of positive tweets, as it classified 44% as positive, double the true proportion.

To better understand the relative strengths of encoder LMs, I examined differences in

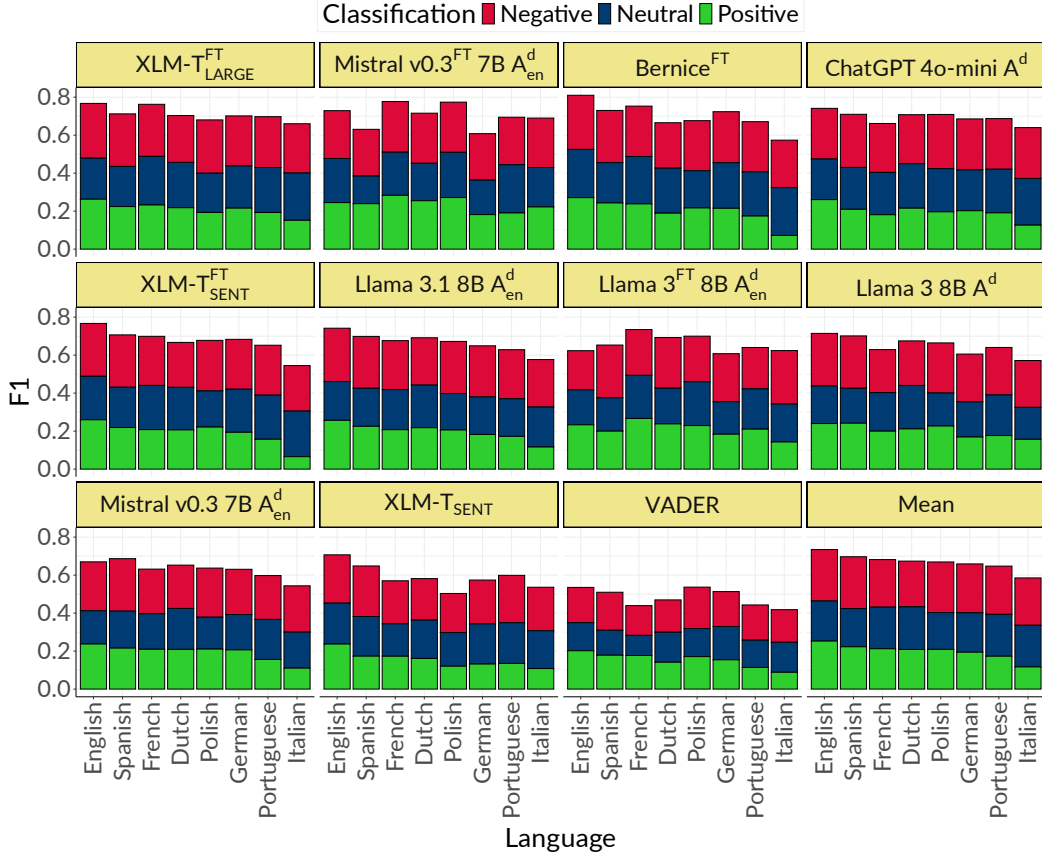


FIGURE 5.1: Macro-averaged sentiment classification F1 scores stratified by country and constituent class score for each model. For each LLM, only results for the best performing prompt is shown. Prompt legend: A_d^{en} : Annotation (dynamic)[en] and A_d : Annotation (dynamic).

performance across model sizes and training strategies. Larger encoder LMs outperformed the smaller XLM-T_{SENT}. Overall, XLM-T_{LARGE}^{FT} was the best-performing model, with comparably high performance across all countries. Bernice performed similarly but was less precise and recalled fewer correct examples. To assess the impact of fine-tuning, I compared the XLM-T_{SENT} model with its fine-tuned variant. Fine-tuning a classification head resulted in an 8% increase in F1 score. Without fine-tuning, LLMs outperformed XLM-T_{SENT}, the only off-the-shelf encoder LM.

Across all LLMs, the fine-tuned version of Mistral v0.3 7B was the top-performing model, with ChatGPT 4o-mini achieving a similar overall performance. In contrast, ChatGPT 4o-mini had the lowest proportion of misclassification between the negative and positive classes. While this metric is not directly factored into the F1 score, it provides additional context for model behaviour. Among the other models, performance was generally comparable, with the exception of VADER, which recorded a

misclassification rate approximately 2.5 times higher than the next closest model.

The impact of fine-tuning on performance varied across models. For the Llama suite, I chose to fine-tune Llama 3, as it performed better on average across all prompts (F1=0.61 compared with 0.59 for Llama 3.1). The optimally prompted off-the-shelf Llama 3 model (F1=0.65) was then narrowly outperformed by its fine-tuned counterpart (F1=0.66). However, this was not consistent across all prompt types and in many cases the fine-tuned model performed worse (see Table C.6.1). In contrast, fine-tuning had a substantial effect on Mistral v0.3 7B, yielding an 11% increase in F1 score compared to its best off-the-shelf result.

With the exception of VADER, model predictions were, on average, aligned with the distribution of the annotated dataset. However, four models exhibited a drift towards particular classes compared to the true distribution. Llama 3.1 toward the negative class, Llama 3^{FT} toward positive, and both XLM-T_{SENT} and ChatGPT 4o-mini toward neutrality (see Table 5.7 for proportions). However, this general alignment did not translate to equal performance across countries.

Model name	Prompt	Negative	Neutral	Positive
XLM-T _{LARGE} ^{FT}	-	0.47	0.33	0.20
Mistral v0.3 ^{FT} 7B	Annotation (dynamic)[en]	0.42	0.34	0.23
Bernice ^{FT}	-	0.47	0.33	0.20
ChatGPT 4o-mini	Annotation (dynamic)	0.43	0.43	0.14
XLM-T _{SENT} ^{FT}	-	0.45	0.34	0.21
Llama 3.1 8B	Annotation (dynamic)[en]	0.50	0.31	0.19
Llama 3 ^{FT} 8B	Annotation (dynamic)[en]	0.39	0.32	0.29
Llama 3 8B	Annotation (dynamic)	0.44	0.33	0.22
Mistral v0.3 7B	Annotation (dynamic)[en]	0.41	0.37	0.21
XLM-T _{SENT}	-	0.41	0.42	0.17
VADER	-	0.33	0.24	0.44
<i>True distribution</i>	-	<i>0.45</i>	<i>0.33</i>	<i>0.22</i>

TABLE 5.7: Predicted classification distribution for each model. For each LLM, only results for the best performing prompt are shown.

Results by country A more detailed presentation of model results is provided in Figure 5.1, which shows macro-averaged F1 scores stratified by its constituent classes across countries. Overall, models performed best at identifying negative tweets (F1, mean (μ): 0.75; standard deviation (σ): 0.08) followed by neutral (μ : 0.62; σ : 0.09) and positive (μ : 0.59; σ : 0.14). When stratified by country, model performance was lowest on tweets in Italian, Portuguese, and German, with average F1 scores of 0.58

(σ : 0.19), 0.63 (σ : 0.12), and 0.64 (σ : 0.11), respectively. In contrast, performance was highest on tweets in English, Spanish, and French, with average F1 scores of 0.71 (σ : 0.11), 0.67 (σ : 0.12), and 0.67 (σ : 0.12).

There was substantial misclassification of the positive class for Italian tweets. Notably, Bernice and the fine-tuned version of XLM-T_{SENT} misclassified 79% and 74% of positive tweets, respectively. Comparing the two versions of XLM-T, there was a 40% increase in tweets classified as neutral following fine-tuning. Furthermore, there was a 56% increase in positive tweets misclassified as neutral. This pattern may be attributed to the distribution of sentiment labels in the Italian subset of the annotated dataset, which contains a high proportion of neutral and relatively few positive tweets (see Table 5.1). Such class imbalance likely biased the fine-tuned model toward predicting neutrality.

These results indicate that although overall model performance can be strong, substantial differences exist across model architectures. In the next Section, I turn specifically to LLMs, examining how prompt design and instruction language affect performance.

5.7 Results: variation in LLM performance by type of prompt

To understand which prompt is best, I evaluated prompts across a range of experiments: their relative performance, differences based on prompt language, sensitivity to the specific tweet example used, and model responses to each prompt. For each prompt, I tested both an English and country-specific language version, as described in Section 5.5.

5.7.1 General performance

Figure 5.2 compares model performance by prompt. Except for ChatGPT 4o-mini, all off-the-shelf models performed better with English zero-shot prompts. On average, English prompts exceeded country-specific language prompts by 4% in F1 score ($\sigma = 0.07$). ChatGPT 4o-mini instead favoured country-specific language prompts, which outperformed English by 2%. The largest contributor to the overall English advantage was Llama 3.1 ($\mu = 0.15$). Llama was also the most variable model in this comparison

($\sigma = 0.11$), whereas Mistral v0.3 7B and ChatGPT 4o-mini were the most consistent ($\sigma = 0.02$)

The distinction between static and dynamic prompts was particularly noteworthy. In few-shot settings, dynamic prompts consistently favoured English over country-specific instructions, though to varying degrees. A similar but less pronounced pattern was observed with dynamic annotation prompts. However, within annotation prompts this trend did not hold for all models, as Llama 3 and ChatGPT 4o-mini performed better with country-specific instructions.

In contrast, for static prompts, language had less impact. With static annotation prompts, an equal number of models favoured English and country-specific instructions, though those favouring English did so more strongly. For few-shot static prompts and zero-shot prompts, the differences across languages were minor. Among the open-source LLMs, most favoured English; Llama 3.1 was a notable outlier, showing the strongest preference for zero-shot English prompts, consistent with its overall tendency to favour English. ChatGPT, by contrast, maintained a preference for country-specific languages. Although fine-tuning did not alter the overall trend between English and country-specific prompts, it amplified the English advantage for dynamic prompts, an effect particularly pronounced for Llama 3, while reducing language differences in static few-shot and zero-shot prompts.

5.7.2 Relative prompt performance

To assess prompt performance relative to one another, I used the zero-shot prompt as a baseline and calculated the percentage increase in F1 score (Figure 5.3 below). For off-the-shelf models, a clear monotonic pattern emerged in which annotation-style prompts consistently outperformed few-shot prompts, which in turn outperformed zero-shot prompts. In general, dynamic prompts led to better performance than static ones. An exception was Mistral v0.3 7B, where static few-shot prompts performed better than their dynamic counterpart. Interestingly, once models were fine-tuned, the distinction between dynamic and static prompts became more important; static prompts performed comparably to the zero-shot baseline, suggesting that in my application fine-tuning reduces the benefit of prompt complexity, but the benefit of country-specific examples remains.

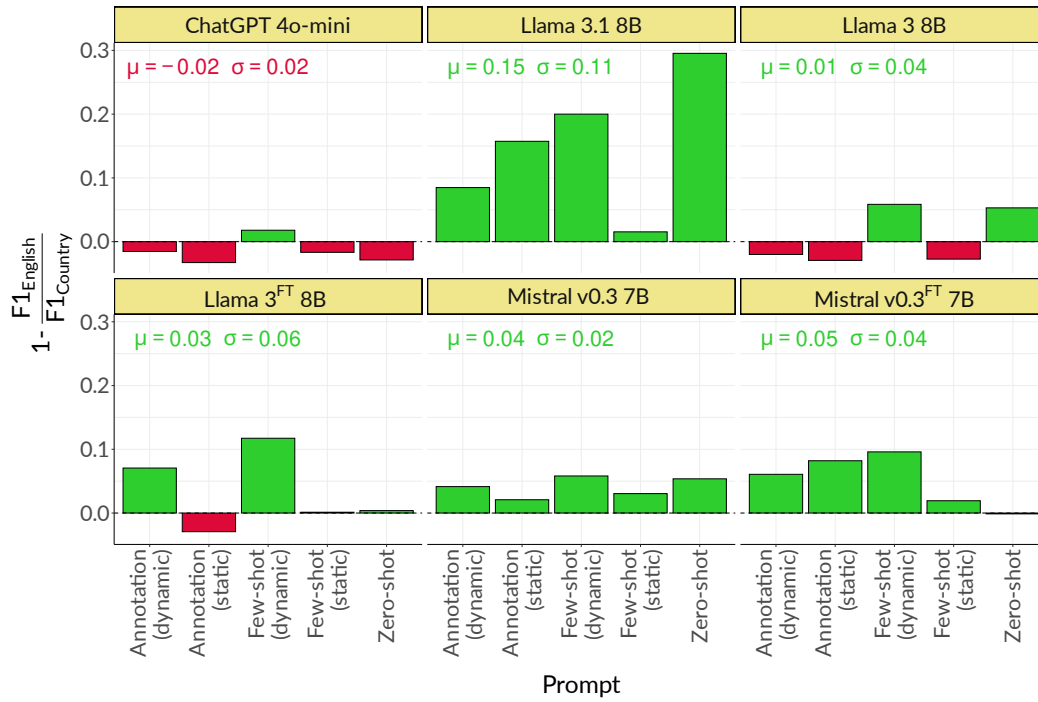


FIGURE 5.2: Comparison of mean F1 performance across all countries (excluding the UK), contrasting English and country-specific language prompt variants, stratified by prompt type for each model. Overlaid text shows the mean and standard deviation across all prompts. Colour indicates which variant performed better: green for English, red for the country-specific language.

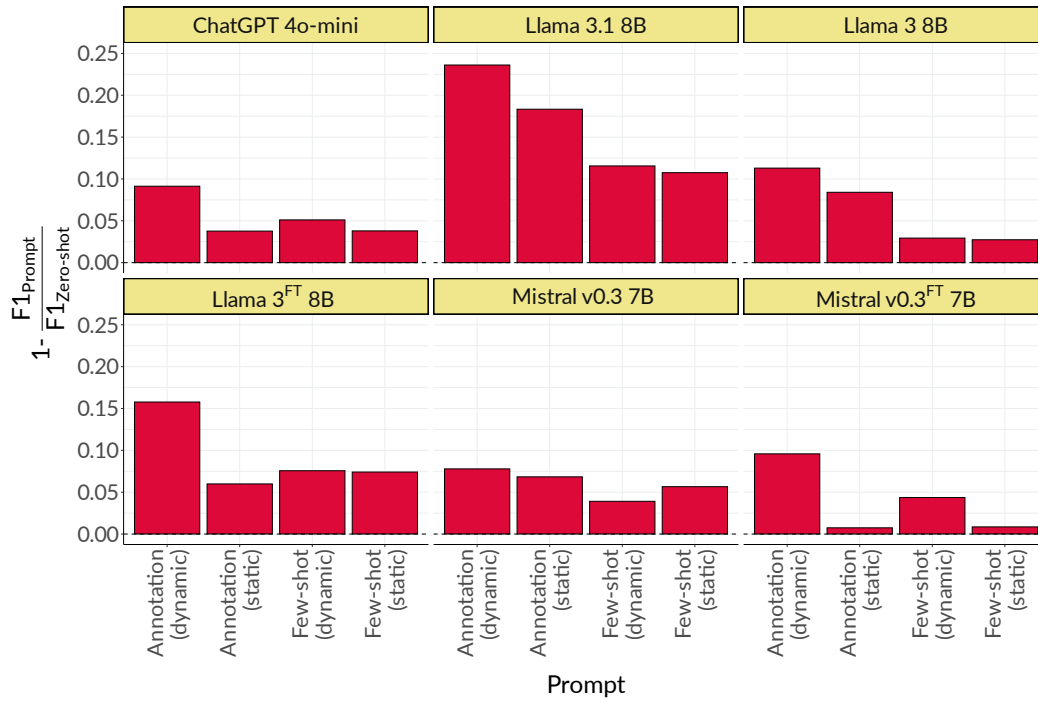


FIGURE 5.3: Comparison of the mean F1 performance of each prompt relative to the zero-shot prompt.

5.7.3 Impact of example choice

In addition to prompt complexity and language, I examined the impact of example choice in few-shot setups. To assess variability across examples, I calculated the standard deviation of F1 scores across folds³. The results are shown in Table 5.8. The largest difference in standard deviation was observed for Llama 3 8B (0.09%). However, this difference was not large enough to indicate that the specific tweet used as an example significantly affects performance. Thus, the inclusion of a country-specific example was important, but the exact tweet itself was not.

Model name	Dynamic	Static
ChatGPT 4o-mini	0.064	0.062
Llama 3 8B	0.065	0.054
Llama 3.1 8B	0.065	0.064
Mistral v0.3 7B	0.059	0.061

TABLE 5.8: Comparison of the standard deviation in F1 scores across the five dataset folds for dynamic and static prompts. Zero-shot prompts are excluded from this analysis.

5.7.4 Generated response length

A challenge in using LLMs for classification is that their outputs are unconstrained free text, rather than fixed labels. Assigning a sentiment class, therefore, required the post-processing mapping outlined in Section 5.5.3. To evaluate the potential influence of this step and better characterise model outputs, I examined both the length and content of the generated responses.

I observed that all prompts produced responses of a similar length (mean character length, $\mu = 8$). Fine-tuning generally led to an increase in response length, with a more notable increase observed for country-specific language prompts. This difference cannot be explained by the classification labels themselves, as the average label length is consistent across both country-specific languages and English. For off-the-shelf models, the variation in response length across languages was less pronounced.

To examine the influence of explicit statements in the prompt to restrict output, I tested a variant of the annotation prompt. This version, referred to as the unrestricted prompt, omitted the phrase: “Provide only a single word, the relevant category, as your response.” This resulted in an average response length of 156 characters, with

³For static prompts, I used the static fold dataset described in Section 5.5.3.

a similar standard deviation. Notably, ChatGPT produced shorter and more consistent responses in both scenarios, which I attribute to its ability to infer the desired response format from the prompt. My qualitative inspection of model output revealed that the additional characters in the unrestricted responses were typically explanations added by the model, despite not being explicitly requested. In ChatGPT's case, the additional length was due to inclusion of the phrase "label:", which aligned with my prompt format. I also observed that English prompts tended to elicit longer explanations than country-specific language prompts. A detailed breakdown of response lengths across all models is provided in Figure 5.4.

In addition to the response length, I also considered the content of the response. Across all models, a classification label was found from one of the eight languages in 99.5% of the responses. Of the remaining responses, 0.23% did not match an exact classification, but contained a classification label as part of the word returned. An example, is *neutrale* which contains the label *neutral*. *Neutrale* is an Italian word for neutral; this is in addition to *neutro*, the classification label I used for neutral Italian tweets. The final 0.27% were responses in which a classification label was not given. In many cases, this was because the model returned a related word, for example, "hopeful" or "sad". In other cases, the model refused to classify a tweet due to safety mechanisms. For instance, the following was a response from Llama 3: "i cannot classify a tweet that contains hate speech. can i help you with anything else?". To prevent bias, I classified all cases as neutral. However, if response types can be reliably determined beyond the defined classification labels, performance can likely be improved by incorporating response reasoning, for example by treating hate speech as negative.

5.7.5 Comparison between prompt length and runtime

The preceding Sections show that annotation prompts tend to achieve higher F1 scores. However, their increased context length also leads to longer runtimes. To better understand the trade-off between prompt length and runtime, I measured the runtime of different LLMs under the prompt configurations used in my experiments. Details of my compute setup are provided in Section 5.8.2.

The results are presented in Figure 5.5, with fine-tuned models indicated by dashed lines. Runtime is defined as the wall-clock time, measured in seconds, from program

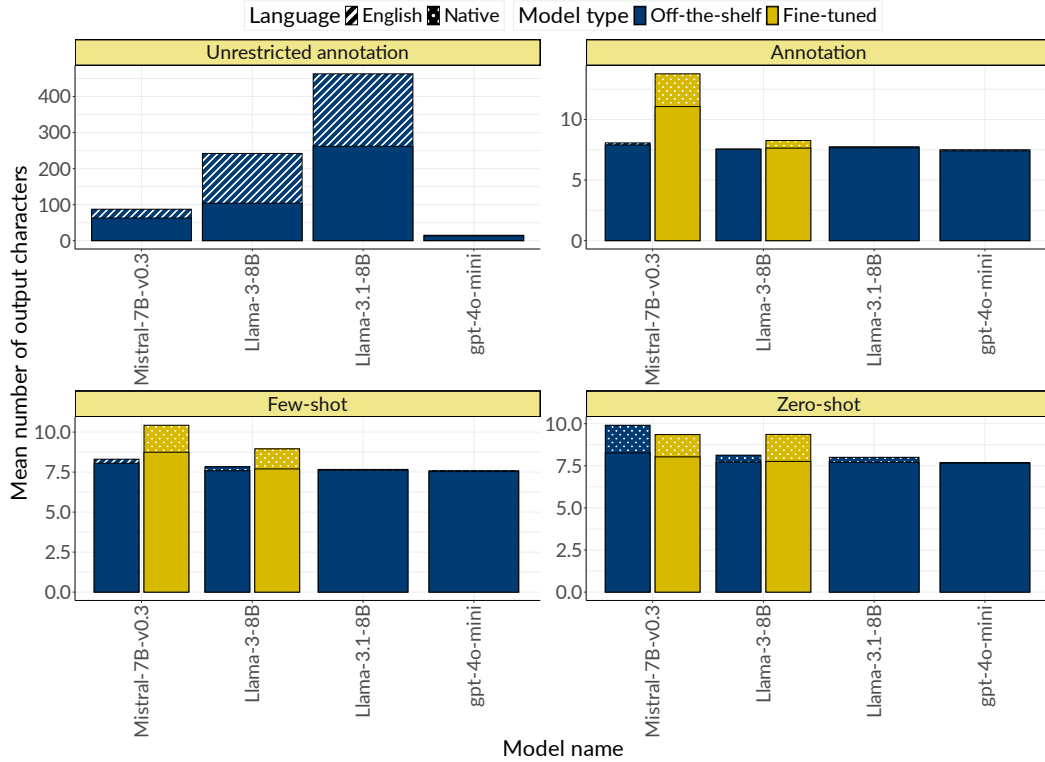


FIGURE 5.4: Mean number of output characters per model stratified by language, prompt, and model type for each prompt class. The pattern shaded on the bar indicates the additional output length for the prompt language type.

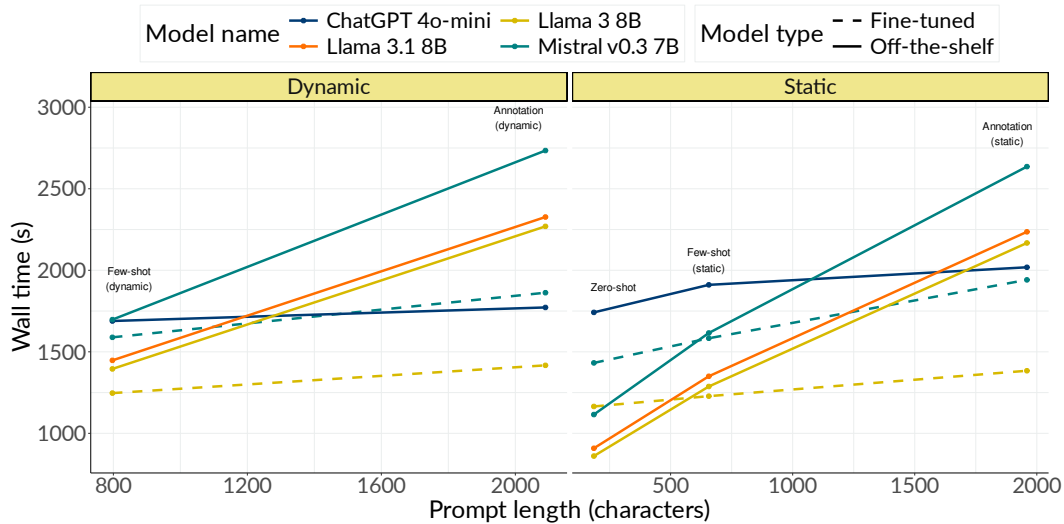


FIGURE 5.5: Scatter plot of prompt length compared to runtime for 2,530 tweets, with trend lines overlaid, stratified by model, for each prompt type.

start to completion. Although CPU and GPU execution times would more precisely capture computational workload, wall time provides a practical and representative metric for comparison. Dynamic and static prompts are reported separately because

their preprocessing pipelines differ, introducing minor overheads prior to model execution.

I found that ChatGPT-4o-mini exhibited an approximately constant runtime profile, which favours longer prompts. This is likely a consequence of its hosted setting, where prompt length introduces negligible overhead in my examples. I expect this constant profile to hold only up to a certain threshold, beyond which prompt length may impact processing time depending on the compute infrastructure used by OpenAI. For all other models, runtime scaled approximately linearly with prompt length. The gradient of increase was lower for fine-tuned models, which were initially slower than off-the-shelf models due to the overhead of adapter merging, but eventually became faster as the prompt length increased.

5.8 Recommendations for the non-expert user

A core goal of this Chapter, using my curated COVID-19 dataset as a case study, is to share my experience and provide recommendations for conducting sentiment analysis for non-expert users. I considered non-expert user as a broad spectrum, which can run from someone with knowledge of sentiment analysis to someone who is able to code at a basic level (either Python or a similar language), although not necessarily experienced with NLP techniques. Moreover, while my analysis was specific to classifying sentiment polarity, many of these insights and recommendations can be extended to other NLP classification tasks, in particular for social media analysis.

Model class	Acc	Time	Ease	Vers	Data R	Data P	Cost
Lexicon	✗	✓	✓	✗	✓	✓	✓
Encoder	!	✓	✓	✗	✓	✓	✓
Encoder ^{FT}	✓	✓	!	!	✗	✓	✓
LLM o-s	!	!	!	✓	✓	✓	!
LLM ^{FT} o-s	✓	!	✗	!	✗	✓	!
LLM c-s	✓	!	✓	✓	✓	✗	✗

TABLE 5.9: Red-Amber-Green (RAG) ratings and symbols (✗: not recommended, !: choose with caution, and ✓: recommended) applied to each model class, reflecting my quantitative and qualitative assessment across a range of categories. Columns from left to right: accuracy, runtime, ease of use, versatility, data requirements, data protection and cost. Model class legend: c-s: closed-source, o-s: open-source, and ft: fine-tuned.

Table 5.9 summarises my experiences of applying each model class across the following categories: accuracy, runtime, ease of use, versatility, data requirements, privacy, and

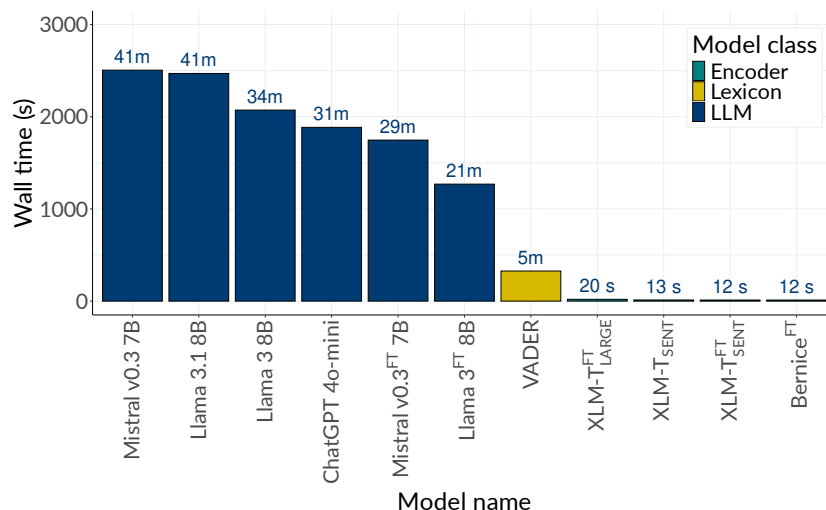


FIGURE 5.6: Execution time for each model applied to 2,530 tweets. For LLMs, the time for the prompt that yields the highest F1 score is shown.

cost. Each category is accompanied by a quantitative or qualitative justification or description based on my practical experience. I stratified the model classes by fine-tuning status and, where relevant, additionally by open- or closed-source status. This table offers a nuanced perspective that can be more useful than a fixed list of recommendations, as it allows users to focus on the relevant categories based on their specific requirements, making it easier to select the most appropriate model.

5.8.1 Accuracy

Accuracy is based on the general performance across a range of metrics defined in Section 5.5.5. Fine-tuned encoder LMs were the best performing according to F1, with the closed-source LLM and fine-tuned LLMs yielding similar scores. Not all open-source LLMs performed well and the worst performers were similar to the off-the-shelf encoder-based sentiment model. Finally, the lexicon approach performed the worst by a large margin ($F1=0.48$). For more information on this category, refer to Section 5.6 where it is discussed in detail.

5.8.2 Runtime

Runtime was introduced in Section 5.7.5 in the context of different prompts. In this Section, I extend the comparison to all models. The runtime is measured on the classification of 2,530 tweets; for LLMs, the value reported corresponds to the prompt yielding the highest F1 score. All models, with the exception of VADER and

ChatGPT, were run on a server equipped with three dedicated NVIDIA RTX A5000 GPUs (24GB vRAM each), an AMD EPYC 7542 32-core CPU, and 504 GB DDR4 RAM. VADER was run on a laptop equipped with an 11th Gen Intel(R) Core(TM) i7-1185G7 @ 3.00 GHz processor and 15.3 GB of DDR4 RAM, with the API calls for ChatGPT made from the same machine. I assumed that processing time was not a bottleneck for either model and, consequently, that running on a laptop did not affect runtime.

The overall model runtimes are shown in Figure 5.6. Among the models, LLMs took the longest to run. For ChatGPT, which was hosted on OpenAI's servers, an additional constraint was API rate limiting, which caps the number of requests (and tokens) per minute as well as the level of concurrency; these limits vary by account tier. Throughput can be improved with strategies such as batching inputs and managing concurrency, but the bottleneck cannot be eliminated. Fine-tuned models were generally quicker than their off-the-shelf counterparts due to quantisation. For VADER, translation was included and constitutes almost all of the processing time. Encoder LMs were the fastest overall. For all models, the time required for fine-tuning (detailed in Table C.3) was excluded. For LLMs, I also excluded the time required to engineer an appropriate prompt, as it was difficult to measure accurately. Nevertheless, this overhead should be considered, as it can increase implementation time in practice.

While LLMs perform comparably in terms of accuracy, they require substantially longer runtimes. For example, the second-best performing LLM, the fine-tuned Mistral model, would take approximately 38 days to process the larger dataset of 4,813,344 tweets, assuming a constant per-tweet runtime. Although runtime may not scale linearly and performance could be improved with better compute resources, this nonetheless represents a notable limitation.

5.8.3 Ease of use

Ease of use refers to the ease with which a model can be interacted with and deployed, considering the technical knowledge, setup complexity, and infrastructure requirements involved. I found that both encoder LMs and open-source LLMs required similar levels of expertise, particularly when it came to setup and environment configuration. However, open-source LLMs generally required substantially more compute

resources unless a quantised version was used, and fine-tuning them was considerably more challenging than fine-tuning encoder LMs. This increased complexity stemmed from both the technical skills required to properly fine-tune LLMs and the extended runtimes. In contrast, VADER and ChatGPT did not present these barriers. Because ChatGPT is a hosted solution, no model setup was required. This makes it far more accessible for practitioners with limited technical backgrounds. Importantly, this benefit is not inherently tied to being a closed-source model, but rather to being hosted. Nevertheless, some data and prompt preprocessing was still needed before issuing API calls, so using ChatGPT was not as simple as typing in a prompt directly into the interface. I assume that a similar level of expertise is required to prompt ChatGPT and use VADER.

5.8.4 Versatility

Versatility refers to how easily a model can be adapted to another classification task. For LLMs, adaptation is straightforward, as the user only needs to change the prompt. For encoder models that require fine-tuning, adaptation is also relatively simple, since only the classification head must be updated, although this assumes the availability of domain-specific data. In contrast, the off-the-shelf encoder classification model and the lexicon model used here were developed specifically for sentiment analysis and therefore cannot be readily adapted to other tasks, limiting their versatility.

5.8.5 Data requirements

Data requirements closely relate to model versatility. Off-the-shelf models offer the advantage of requiring little to no annotated data, making them appealing for quick deployment. While prompting may require some annotated examples - particularly in few-shot setups, this number is typically small, as demonstrated in my experiments. In contrast, fine-tuning a model necessitates a substantial volume of annotated data. Compiling this data represents a significant effort. Every time a model needs to be adapted for a new classification task, a new dataset must be collected and annotated.

5.8.6 Data protection

A crucial consideration related to data is the issue of data protection. Many applications require that data remain confidential throughout processing. When running models locally, data remains private since all computation is performed within the

local environment. By contrast, closed-source hosted models such as ChatGPT require user data to be transmitted externally to the provider’s servers. This reliance on third-party infrastructure introduces potential privacy concerns, the severity of which depends on how the hosting company stores, processes, and utilises user data.

5.8.7 Cost

The final category is cost. An important consideration for closed-source hosted models is that each API call incurs a usage fee. The optimal dynamic annotation prompt cost 0.24 USD to process 2,530 tweets. Across all prompts, the total cost was 1.39 USD, and the cumulative cost across all runs was 6.95 USD (see Appendix C.5.2 for details). Open-source LLMs and encoder models, by contrast, can be run locally, although they require access to graphics processing units (GPUs). Encoder models are less compute-intensive and can often run on consumer hardware, whereas LLMs typically require larger GPUs. As described in Section 5.8.2, I ran these on an in-house server. While this setup carries associated costs (e.g., amortised capital costs, depreciation, and energy), these are difficult to quantify precisely. Consequently, I do not report server costs directly. Instead, for cases where in-house compute is unavailable, rough cost estimates for running encoder LMs and LLMs on cloud resources are provided in Appendix C.5.1.

5.9 Discussion

In this Chapter, I compared three classes of models, LLMs, encoder LMs and a lexicon-based approach, using an annotated dataset of 2,530 tweets across eight languages. The evaluation considered both accuracy and, more broadly, the intention of informing practical recommendations for model selection. For the LLMs, this involved testing a range of prompts that varied in the amount of information provided and the language used, combining country-specific and English formulations.

I found that fine-tuned encoder LMs were the best-performing models. The superior performance of encoder LMs over lexicons was consistent with the literature reviewed. Moreover, I found that most off-the-shelf and fine-tuned LLMs performed comparably to the encoder LMs, although their performance is prompt-dependent.

Across the range of prompt configurations, prompts with more context performed better and the optimal language of instruction changed based on the amount of context provided. Furthermore, dynamic prompts performed best across most models. These results varied between fine-tuned and off-the-shelf LLMs. Across all prompts, responses were similar in length and were easy to parse for classification. However, LLMs require explicit instructions to restrict their responses.

Overall, with two exceptions, the best-performing prompts were dynamic with English instructions. The first exception was ChatGPT 4o-mini, which is expected given its multilingual support. The second was Llama 3, which is interesting as its successor natively supports multiple languages. The superior performance of cross-lingual prompts in a few-shot settings supports Zhu et al.'s (2024) finding that the English-centric LLMs they tested were able to adapt target language with limited data. Moreover, in line with Zhu et al. (2024), I also found that while the closed-source model show superior performance in zero-shot setting, this advantage diminishes in settings when open-source LLMs are exposed to task data (that is, fine-tuning and few-shot settings).

Although it is difficult to determine a priori what level of accuracy is sufficient for a model to be useful in a downstream task, it is nevertheless important to reflect on both prediction quality and accuracy, particularly with a view to applying the best-performing model to the larger dataset in Chapter 6. Assuming a uniformly random guess across all three classes, an accuracy of 33.33% could be expected. My best-performing model surpassed this by a large margin, achieving an accuracy of 75%. When a model was incorrect, I was also concerned with the nature of the errors, with misclassifications that bypass the neutral class regarded as more severe. The best-performing model, XLM-TLARGE^{FT}, misclassified around 5% of tweets across the negative-positive boundary, which was notably higher than ChatGPT 4o-mini's proportion on this metric (2%). Nonetheless, given XLM-TLARGE^{FT}'s superior overall accuracy, higher F1 score, and faster runtime, it remained the more appropriate choice for my purposes.

My analysis was applied to a manually curated dataset of tweets. Securing expert annotators proved challenging, resulting in a single annotator for languages other than English. This made it difficult to disentangle differences in polarity distributions across countries arising from annotator bias versus cultural variation. This limitation

reflects a common issue when working across multiple languages in an English-centric environment, and is likely to be even more pronounced for more complex annotation tasks. While this presents a constraint, it underscores the need for reliable automatic solutions to analyse large volumes of data, as is often required in social media research.

Although my annotated sample is small relative to the larger dataset, I argue that its diversity may allow for a more balanced reflection of public sentiment across a broader range of pandemic-related issues. I adopted a stratified sampling approach to retain the distribution of search terms from the larger corpus, which public health experts had curated. By maintaining this diversity within my smaller annotated subset, I aimed to ensure that the dataset remained representative of the broader discourse. In contrast, many existing datasets focus heavily on a narrow subset of topics (for example, lockdowns, masks, or vaccines), which can introduce topic-specific bias. For instance, Lamsal, Harwood, and Read (2022) and Banda et al. (2021) used 13 and 14 search terms, respectively, many of which overlapped, illustrating this limited topic scope.

The benefits of constructing my dataset extend beyond ensuring topic diversity. The process of curating an annotated dataset is a crucial step when adapting models to new domains, enabling a consistent evaluation framework. Undertaking this process myself informed my evaluation of model dimensions beyond accuracy, including runtime, ease of use, and versatility. These dimensions are not well documented and are often not considered together in NLP research, despite their importance for real-world applications.

One challenge in using LLMs for classifying public data is the potential risk that the models may have been trained on portions of the same data later used for evaluation. While I cannot entirely exclude the possibility that the LLMs or encoder LMs used here were exposed during training to tweets from my dataset, existing research suggests that any such overlap would have only a minimal effect on model performance (Ushio and Camacho-Collados, 2024; Agarwal and Nenkova, 2022). Moreover, by curating my own annotated dataset, I ensured that the models would not have had access to the specific sentiment labels assigned, thereby mitigating the risk of label leakage.

Considering the non-expert user, it is important to reflect on the specific tasks that are most important to them. My work has focussed on classifying sentiment polarity.

However in conjunction with advancements in NLP, the scope of sentiment analysis has expanded. This trend is evident in the literature, where the range of benchmark tasks has expanded to include, for example, aspect-based sentiment analysis and stance detection (Zhang et al., 2024; Antypas et al., 2023). I note that some of these tasks may be more useful in a public health context. For instance, in the case of COVID-19 vaccination, stance and aspect-based sentiment classification could provide more valuable insights than general sentiment polarity, as these methods offer a deeper understanding within a topic-specific context (Bruin et al., 2022). While the general nature of my task might limit its immediate utility for more specialised downstream applications, the recommendations I have developed extend more broadly.

Overall, encoder LMs were the quickest to run but required additional time for fine-tuning depending on their size. These models also required technical knowledge, compute resources, and annotated data to fine-tune. Off-the-shelf models such as XLM-T_{SENT} were available, but as shown, they did not always perform well on bespoke datasets. In cases where data are sparse, an LLM can be a better option. Although these models are easier to interact with, they still require technical knowledge to host and, at larger scales (e.g., in the millions), can be prohibitively slow. For users unfamiliar with fine-tuning or hosting LLMs, a hosted model is recommended. However, both cost and data privacy requirements should be carefully considered. While I only evaluated a closed-source hosted LLM, it is important to note that other approaches exist to improve usability. For example, Camacho-collados et al. (2022) provide a Python API that offers a straightforward way to interact with and fine-tune encoder LMs, including XLM-T used in this work.

A limitation of my recommendations is the scope of models considered. Within each category, I restricted the scope to a subset of models; for example, I did not include larger open-source LLMs. The same limitation applies to closed-source models, where I could have examined larger variants, reasoning-focused models, and model families from other providers such as Anthropic’s Claude and xAI’s Grok. Finally, an additional class of models worth considering would be traditional ML approaches, such as support vector machines (SVMs) with TF-IDF or lexicon-based features. These approaches have also been used in public health applications and may perform better than lexicon-based methods while remaining simpler than encoder LMs and LLMs (Braig et al., 2023).

5.10 Conclusion

In conclusion, smaller, domain-specific encoder LMs are reliable for multilingual sentiment analysis of social media data. In my experiments, they proved the most accurate, outperforming both LLMs and a single lexicon-based approach. However, given their balance between performance and ease of use, LLMs hold significant promise in interdisciplinary contexts. To improve the performance of LLMs, further research is needed into optimal prompting strategies, particularly in multilingual settings. Based on my findings, I argue that simpler approaches, commonly used in both public health and more broadly, should be replaced by more modern transformer-based techniques.

To support non-expert users in selecting a suitable approach, I have provided recommendations informed by both my qualitative insights and quantitative results. While my experiments focused on sentiment polarity, the guidance offered is broadly applicable to similar classification tasks. My findings encourage the adoption of more accurate and modern approaches, and my recommendations serve as a foundation for future work to build upon.

6 Temporal evolution of tweet variables and their correlation with epidemiological and public policy variables

6.1 Overview

Chapter 5 showed that XLM-T_{LARGE}^{FT} was the best-performing model, achieving the highest F1 score and ranking among the fastest in runtime. In this Chapter, I apply this model to classify sentiment from the analysis dataset introduced in Chapter 4. Using this dataset, I investigate the evolution of sentiment-stratified tweet volume and examine the topics that dominated COVID-19-related public discussion. I then use Kendall's τ to assess associations between tweet volume, both overall and stratified by sentiment and search term category, and epidemiological and public policy variables.

6.2 Introduction

As discussed in Chapter 1, behaviour plays a crucial role in epidemiological dynamics by directly influencing the trajectory of disease spread. In particular, adherence to and adoption of protective behaviours affect multiple aspects of disease dynamics, and during the COVID-19 pandemic their role in mediating the effectiveness of NPIs and vaccination uptake was especially relevant (Lison et al., 2023).

Measuring the public's adherence and adoption of protective behaviour is important, but difficult. During the COVID-19 pandemic, it was typically captured through surveys, which were often expensive and time-consuming to conduct, and also challenging to administer quickly in real-time (Wang et al., 2022a). Beyond surveys, researchers also relied on proxy measures that may relate to disease transmission, with mobility data being a typical example (Alessandretti, 2022; Hu et al., 2021b). Analogously, social media may provide another such proxy.

In this Chapter, motivated by the idea of using sentiment as a proxy for valence as a measurable behavioural determinant, I investigated Twitter sentiment. The COVID-19 pandemic was the first in which social media played a major role both in the

dissemination of information and as a medium for public expression. The volume of tweets on specific topics served as a barometer of public attention, and I focused on how sentiment evolved alongside topic prominence. To capture heterogeneity in both policies and public responses, I examined a wide range of topics across multiple countries. I also assessed how tweet volume, both overall and stratified by sentiment and search term category, correlated with epidemiological and public policy variables. My analysis was conducted over two-year period beginning on 1 January 2020.

With the above in mind, I aimed to answer the following research questions:

1. How did COVID-19 related discussion on Twitter evolve over the pandemic?
2. How did tweet variables correlate with epidemiological variables (cases, hospitalisations, and deaths), vaccination uptake, and policy stringency?
3. To what extent were the evolutions and correlations described above consistent across countries?

While previous studies have examined sentiment and social media in the context of COVID-19 (see Section 6.3), in this Chapter I present one of the first cross-country examinations of pandemic-related sentiment over an extended period and across a broad set of search terms. This analysis highlights both the dynamics of sentiment-stratified tweet volume and its associations with epidemiological and public policy variables. In doing so, it provides a more holistic view of COVID-19-related tweet volume, sentiment, and their potential correlates.

6.3 Related work

Building on Chapter 5 and Sections 1.5 and 5.3.1, where I reviewed sentiment classification and broader applications of social media in public health, I now review studies that examined how tweet and search volumes correlated with COVID-19 epidemiological and public policy variables. The studies reviewed vary across several dimensions, including how they correlate time series (no lag, fixed lag, or cross-correlation), the length and start of the study period, and the variables examined.

Studies without a temporal shift measured correlations at the same time point. For example, Doogan et al. (2020) calculated the Pearson correlation between the three-day MA of tweet volume related to NPIs and new COVID-19 cases across six countries

over four months beginning 1 January 2020. Using the date of the first confirmed case as the starting point, they reported strong correlations above 0.7 in Australia and New Zealand, but a much weaker, non-significant correlation of 0.25 in the UK. Similarly, Cheng, Han, and Liu (2023) used Spearman's rank correlation to examine the relationship between tweet activity and vaccination uptake in the UK during three phases of the vaccine rollout (1 January 2021 to 1 April 2022). After classifying sentiment and finding that roughly half of the tweets were positive, they restricted their analysis to this subset and observed moderate, consistent associations across all three phases (0.53–0.54).

By contrast, Turiel, Fernandez-Reyes, and Aste (2021) introduced a single fixed shift, intending to develop an early warning system. They examined whether early peaks in Twitter activity (21 January to 21 March 2020) preceded mortality peaks by one month in Italy, Spain, and the United States of America (USA). Tweet volume was defined as the number of unique users posting COVID-19-related tweets, normalised by a regional population metric. Using Spearman's rank correlation, they reported associations ranging from 0.31 to 0.66 between active daily users and log mortality. However, the study did not clearly specify the exact time windows or the series of observations used to calculate these correlations, introducing uncertainty into the interpretation of the results.

Moving beyond single fixed shifts, some studies have examined correlations across a range of temporal lags. Cheng et al. (2021) explicitly investigated cross-correlations between tweet volume and epidemiological variables. They analysed five major UK cities to determine the temporal lag at which the absolute Spearman's rank correlation between tweet volume (for general disease terms) and COVID-19 deaths was maximised during the period 1 February to 19 August 2020. They found strong correlations ranging from 0.7 to 0.95, with lags of 6 to 27 days. Complementing this correlation-based approach, Lamsal, Harwood, and Read (2022) explicitly accounted for temporal dependence between series by using Granger causality; both this concept and the study are discussed further in Chapter 7.

A final noteworthy study, which examined a range of lags, used Google Trends data (introduced in Chapter 1). Although Google Trends is not strictly a form of social media, it also reflects public attention and has previously been used to study associations with epidemiological variables. Effenberger et al. (2020) analysed relative

search volume indices for “Coronavirus” across 12 countries (including the UK, Germany, and Italy) over three months beginning 31 December 2019. Using Pearson cross-correlations over a 35-day window, they found that correlations with new cases peaked on average 11.5 days before incidence, with the UK specifically showing a 12-day lag. Lags for Germany and Italy were not reported. Across all countries, correlation magnitudes were presented only in a figure, often with values near or above 0.5, but exact values were not provided. Further applications of Google Trends in time series modelling are discussed in Chapter 7

6.4 Data

6.4.1 Tweet variables

In this Chapter, I measured tweet volume using the deduplicated analysis dataset of 5,154,424 tweets, the composition of which is shown in Table 4.3. To stratify volume by sentiment, I applied the best-performing model from Chapter 5, XLM-T_{LARGE}^{FT}, to the distinct set of tweets in this dataset.

In addition to tweet volume, I also considered engagement with individual tweets, stratified into two categories. The first category was likes, indicating the number of times a tweet was liked. The second category was responses, aggregating the number of replies, retweets, and quotes a tweet received.

6.4.2 Epidemiological and public policy variables

I considered two classes of variables. The first comprised epidemiological variables, namely, case incidence, reported deaths, and hospital admissions and occupancy. The second comprised public policy variables, namely policy stringency and vaccination uptake.

For all EU member states, I sourced data from the European Centre for Disease Prevention and Control (ECDC) whenever possible. These data originated either from national submissions to The European Surveillance System (TESSy) or, when TESSy data were incomplete, from official national sources. Further details on the TESSy reporting protocol, including how countries submitted data and how the ECDC collected and processed them, are provided in European Centre for Disease Prevention and Control (2022b) and European Centre for Disease Prevention and Control

(2022a). When these data were incomplete or only available at a weekly resolution, I sourced data from Our World in Data (OWID). For the UK, I obtained data from the United Kingdom Health Security Agency (UKHSA). Each outcome is described in more detail below.

Cases and deaths

For European Union (EU) countries, I obtained daily data from the ECDC, where each country reports the number of newly confirmed cases and newly reported deaths. There were three instances in France and one in Italy where the reported values for cases and deaths, respectively, were negative. In these instances, I replaced the values with the mean of the previous seven days. Exact details on data sources, as shown for hospitalisations in Table 6.1, were not available. I obtained daily data for the UK from the UKHSA, where cases refers specifically to Pillar 2 positives for individuals over the age of 25, and deaths refers to deaths occurring within 28 days of a positive test.

Hospitalisations

I primarily sourced hospital admissions data from the ECDC. When these data were incomplete, I used OWID as an alternative source. For Poland and Portugal, admission data were unavailable and I relied on hospital occupancy data instead. Throughout the remainder of this Thesis, I broadly refer to hospitalisations, which includes occupancy in the context of these two countries. For the UK, I obtained data from the UKHSA. The exact outcome and data source for each country are provided in Table 6.1.

Vaccination uptake

I assessed vaccine uptake using the total number of daily vaccinations administered in each country, irrespective of dose number, with data sources from OWID (Mathieu et al., 2021). The sources for the data provided by OWID is given in Table 6.2. For EU countries, I used data from OWID because they were available at a daily resolution. In contrast, data from the ECDC were only available at a weekly resolution (Mathieu et al., 2020a). OWID reported a seven-day rolling average of the daily number of vaccinations. For countries that did not report daily data, they assumed that the number of doses changed uniformly over any period with missing data and linearly

Country	Outcome	Source
France	Weekly new admissions per 100k	TESSy RESPISEVERE
Germany	Weekly new admissions per 100k	ECDC database ⁺
Italy	Weekly new admissions per 100k	TESSy RESPISEVERE
Netherlands	Weekly new admissions per 100k	TESSy RESPISEVERE
Poland	Daily number of patients in hospital	ECDC* (OWID)
Portugal	Daily number of patients in hospital	General Directorate of Health# (OWID)
Spain	Weekly new admissions per 100k	Surveillance
UK	Daily new admissions within the past 24 hours	UKHSA

TABLE 6.1: Outcomes and sources of hospitalisation data. All outcomes are COVID-19 specific. TESSy RESPISEVERE denotes data submitted to the ECDC by each country. *Polish occupancy data was obtained from OWID, which cites the ECDC as the source; however, data for Poland were not available in the ECDC dataset version used in this study. ⁺compiled from public online sources. #via Data Science for Social Good Portugal.

interpolated the data. The result was smoothed using a seven-day rolling average based on the previous six days. For the UK, I obtained data from the UKHSA.

Country	Source
France	Public Health France
Germany	Robert Koch Institut
Italy	Extraordinary commissioner for the COVID-19 emergency
Netherlands	ECDC, via Marino van Zelst
Poland	Ministry of Health
Portugal	WHO
Spain	Ministry of Health
United Kingdom	UKHSA

TABLE 6.2: Sources of the vaccination uptake data

Anomalies were observed in OWID data for Spain between 5 December 2021 and 5 January 2022, when compared with uptake outside this period in the country. Additionally, Spain tracked the ECDC vaccination profile except for this period. To address this, I linearly interpolated the weekly ECDC values to a daily resolution. I then divided these values by the sum of the ECDC reported values for the same period and used this ratio as a weight to redistribute the total number of vaccinations according to OWID. These results are shown in Figure D.5.

Policy stringency

The Oxford COVID-19 government response tracker (OxGRT) quantified the level of policy restrictions implemented over time worldwide during the COVID-19 pandemic (Hale et al., 2021). It provided four indices measuring government response: Government Response, Containment and Health, Stringency, and Economic Support. For this study, I focused on the Government Response index because it was the most comprehensive and incorporated policies visible to the general public. This composite index comprised 16 indicators covering containment and closure measures, health system policies, and economic interventions, as listed in Table 6.3. Each indicator ranged from 0 (most lenient) to 100 (most stringent). Some policy restrictions varied by vaccination status, resulting in different versions of the index. I used the simple average version, which calculated the average index value across vaccination groups. Further details on individual policies and index construction are provided in Hale et al. (2023). Note that throughout this Thesis, the term stringency refers specifically to the level of policy stringency as measured by the Government Response index, and should not be confused with the Stringency index.

Policy class	Policy name
Containment and Closure	School closing
	Workplace closing
	Cancel public events
	Restrictions on gathering size
	Close public transport
	Stay at home requirements
	Restrictions on internal movement
	Restrictions on international travel
Economic Response	Income support
	Debt/contract relief for households
Health Systems	Public information campaign
	Testing policy
	Contact tracing
	Facial coverings
	Vaccination Policy
	Protection of elderly people

TABLE 6.3: Policies included in the Government Response index

6.5 Methods

To examine the relationships between tweet variables and epidemiological and public policy variables, presented in the previous Section, I applied Kendall's τ (specifically

τ_b). I chose this measure over Pearson’s correlation because the data were non-linear, and over Spearman’s ρ because it handles ties more effectively (Alvo and Cabilio, 1985). The significance level is set at 0.05, adjusted for multiple testing using a Bonferroni correction. In the results to follow, the number of tests used to adjust α is shown in the plot legend where applicable.

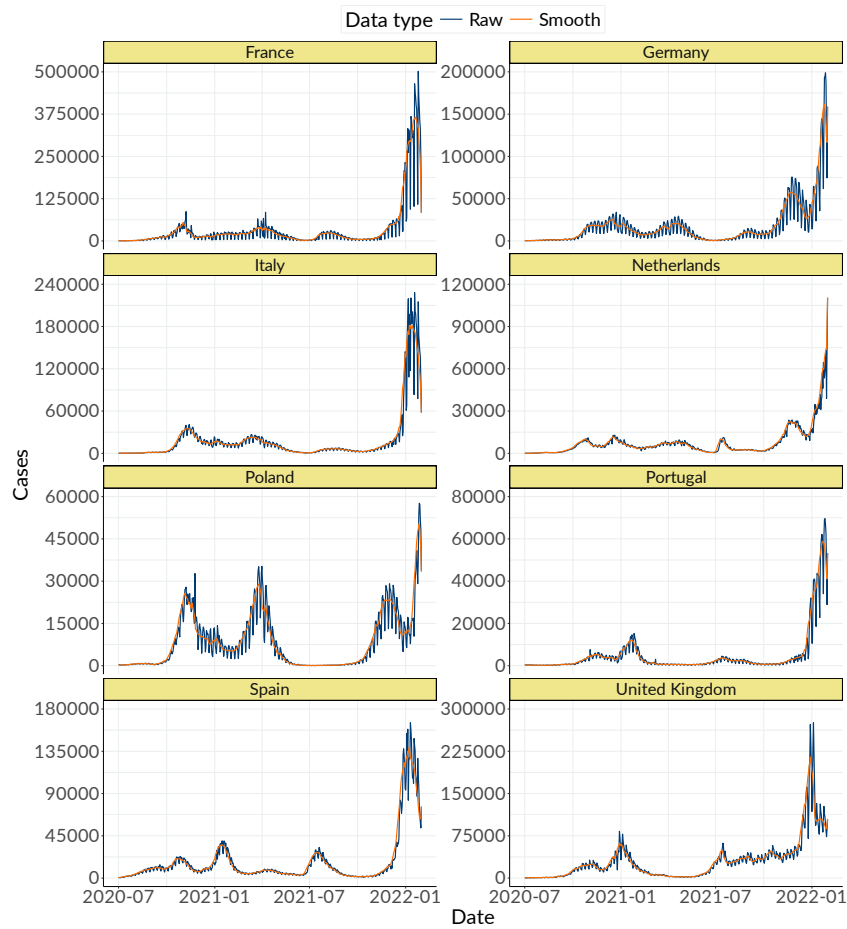
I filtered all data to the period from 1 January 2020 to 30 January 2022, with additional outcome-specific adjustments. For cases, I restricted the data to those reported after 1 July 2020, when more widespread testing had begun across all countries in the analysis (Hasell et al., 2020). For vaccine-related comparisons, I restricted the data to tweets on or after 9 November 2020, a date motivated by both Pfizer Inc. (2020) and the spike in vaccine-related activity shown in Figure 4.8. This removed a further 55,981 entries from the inflated dataset. As a sub-analysis, I also applied XLM-T_{LARGE}^{FT} to both these excluded tweets and the potential bot tweets filtered earlier (see Section 4.5.2), with results presented in Appendix D.3. More generally, for each outcome, I used the earliest available reporting date while applying the relevant outcome-specific filters.

I considered five policy stringency variables. The first is the raw stringency level, which allowed me to examine whether tweet volume and sentiment were associated with the degree of restriction itself. The remaining four variables captured changes in stringency. I began with the general change in stringency, calculated as the one-day lagged difference in the raw stringency level, before introducing three measures derived from this change to assess whether particular types of change elicit stronger public responses. Specifically, I separated the change into positive and negative components, capturing the implementation and easing of restrictions, respectively. Each variable is assigned a value of zero when no change of the corresponding sign occurs. Finally, I considered absolute changes, calculated as the absolute value of the general change, to assess how users respond to a change in policy regardless of its direction.

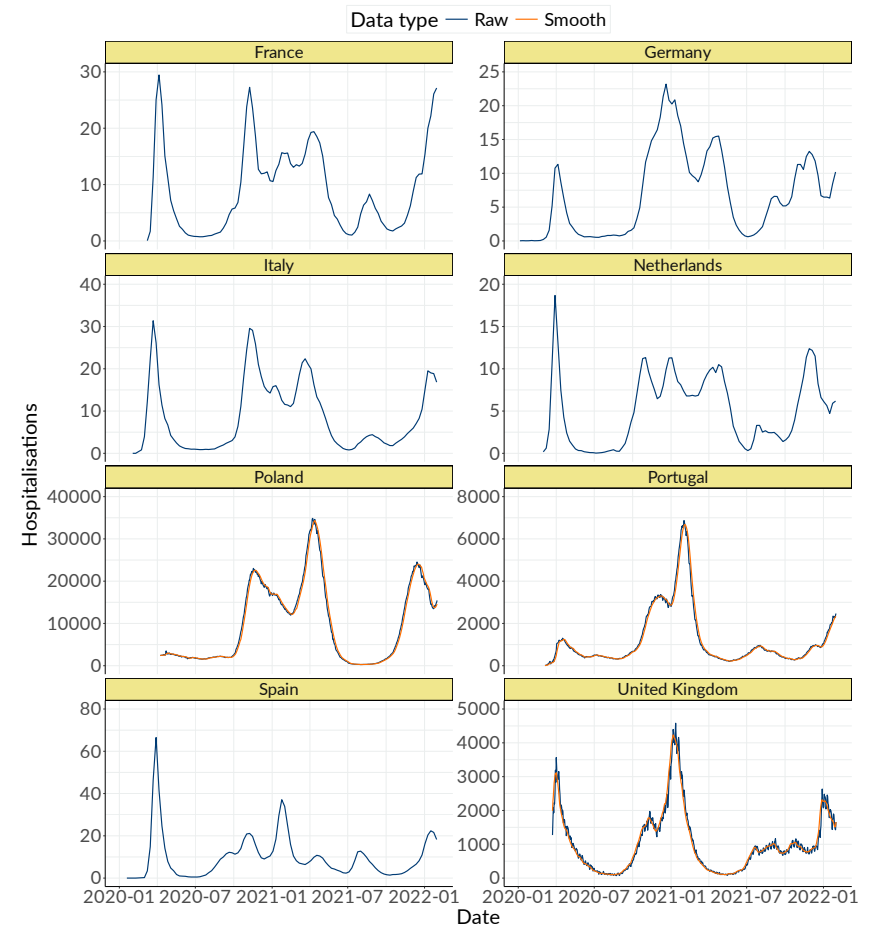
When considering the utility of tweets, it is important to determine the optimal time window for analysis, specifically the lag at which the maximum correlation between tweet variables and variables of interest occurs. To assess this, I extended the analysis by computing cross-correlations over a two-week window, shifting the tweet-volume series forward or backwards by up to 14 days relative to each variable of interest and recomputing Kendall’s τ at each shift.

To account for day-of-the-week effects in both tweet volume and outcome reporting, I applied a seven-day MA to all variables before comparison. When hospitalisation data were reported weekly, no smoothing was applied; instead, tweets were aggregated into weekly counts with aligned dates. The motivation for smoothing stringency differed. Since stringency is a step function, changes in stringency form a discrete impulse function. This makes comparison to tweet volume difficult because only a small subset of days are compared. By applying smoothing, the change is extended over several days and consequently allows tweet volume a few days before and after a change to be considered. Smoothing was applied to the raw stringency variable, and the subsequent stringency change variables were calculated from it.

The raw data for each profile, along with the seven-day MA, are shown next. Figure 6.1 presents cases and hospitalisations, and Figure 6.2 presents deaths and vaccination uptake. Figure 6.3 presents the policy stringency variables. The first panel (Figure 6.3a) shows the raw index alongside the index smoothed using a seven-day MA. The second panel (Figure 6.3b) presents three of the four change components introduced earlier: the overall change in stringency, as well as the positive and negative changes. The absolute change is omitted for visual clarity; this variable comprises the positive component together with the absolute value of the negative component. It is instead shown in Appendix Figure D.6.

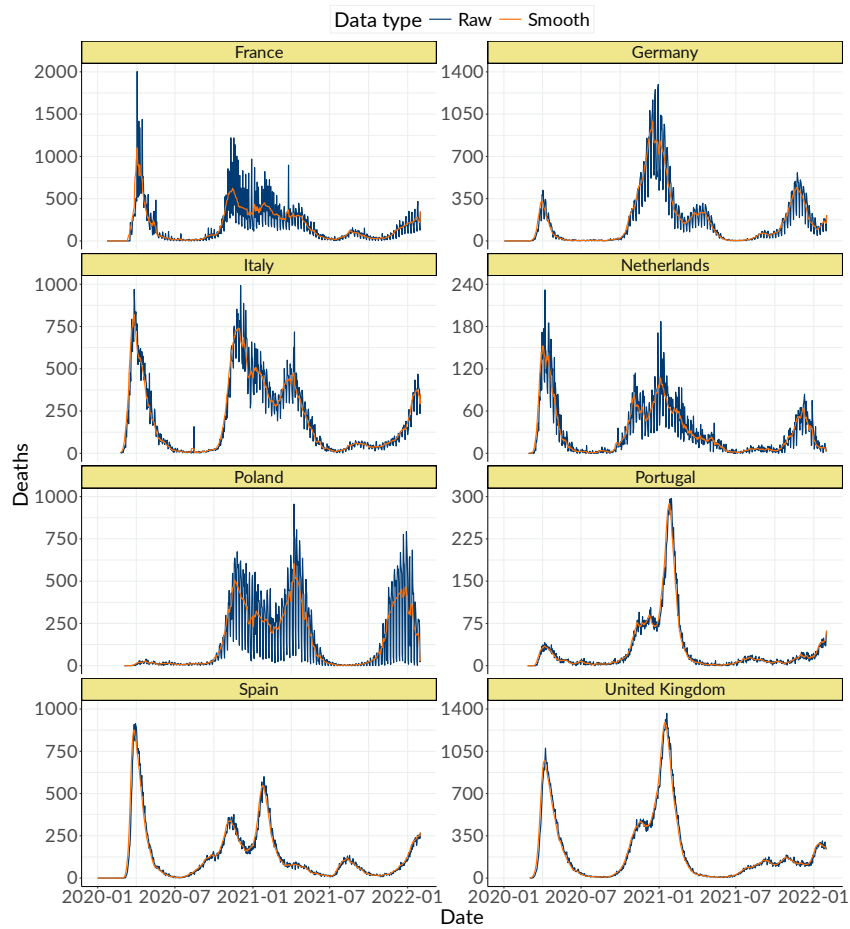


(A) Cases

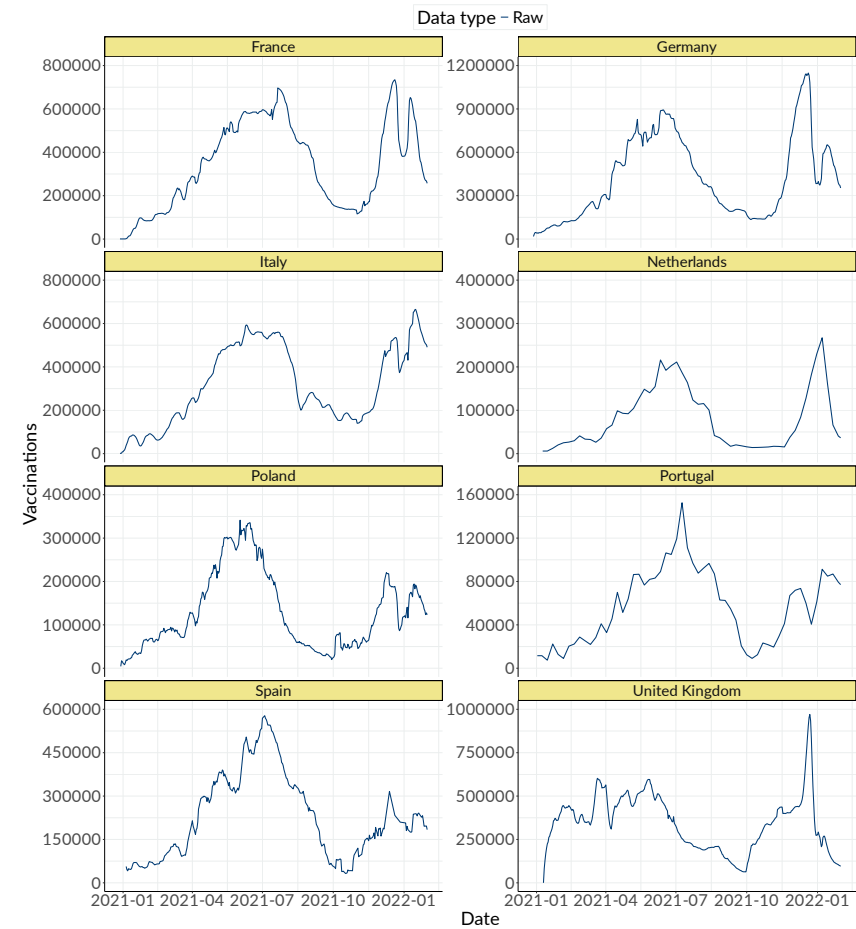


(B) Hospitalisations

FIGURE 6.1: Raw values and seven-day MA of (A) cases and (B) hospitalisations. Where hospitalisations are reported weekly, no smoothing is applied. Please see Table 6.1 for exact outcomes and clarification of the y-axis scale.

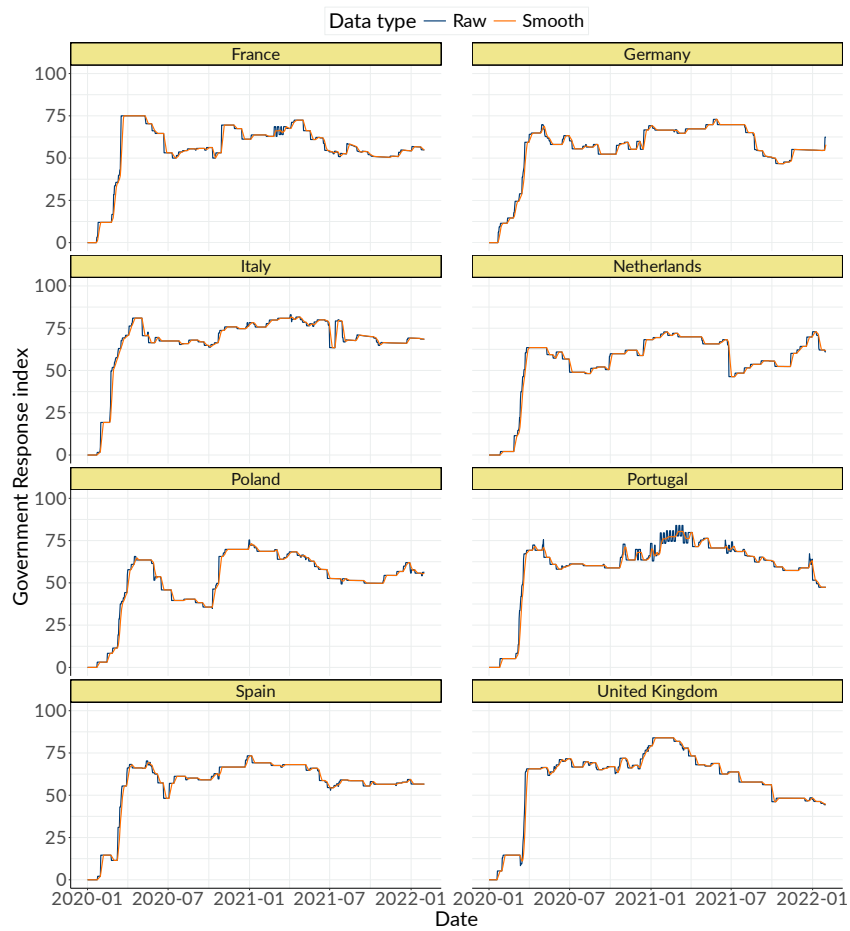


(A) Deaths

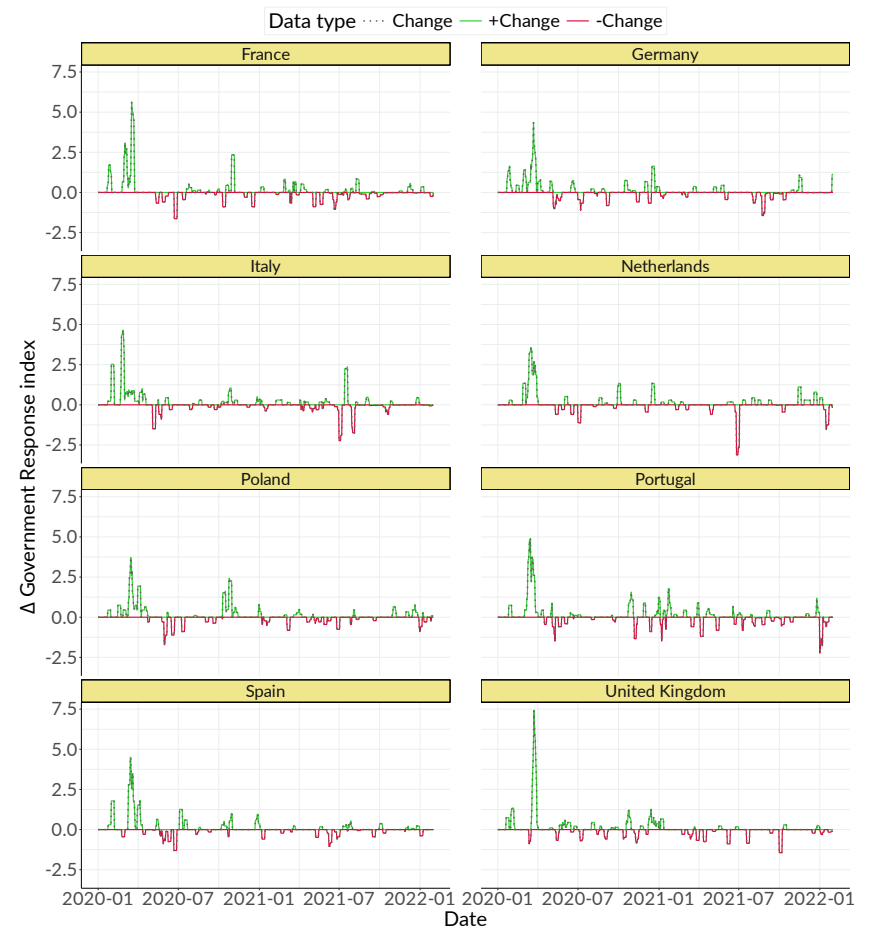


(B) Vaccination uptake

FIGURE 6.2: (A) Raw values and seven-day moving averages of deaths, and (B) raw values of vaccination uptake. Vaccination uptake are already smoothed at source, so no further smoothing is applied.



(A) Government Response index



(B) Changes in Government Response index

FIGURE 6.3: The (A) raw and seven-day MA of the Government Response index, along with (B) the corresponding one-day lagged difference of the seven-day MA. The positive, negative, and combined change components are shown.

6.6 Results

6.6.1 Temporal evolution of attention and sentiment polarity

Overall sentiment polarity

All countries had a similar distribution of negative, neutral, and positive tweets overall. The distribution of tweets across all countries (mean (standard deviation)) is as follows: 53% (3.7%) negative, 30% (7.8%) neutral, and 17% (5.9%) positive. Two outlying countries were the UK, which had the largest proportion of positive tweets, and Italy, which had more neutral tweets than negative and a much smaller proportion of positive tweets. The exact distribution of tweet polarity per country is shown in Figure 6.4.

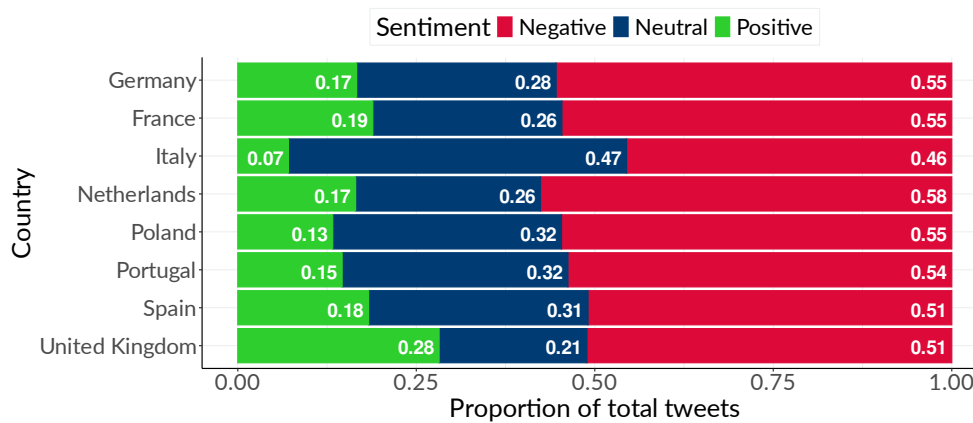


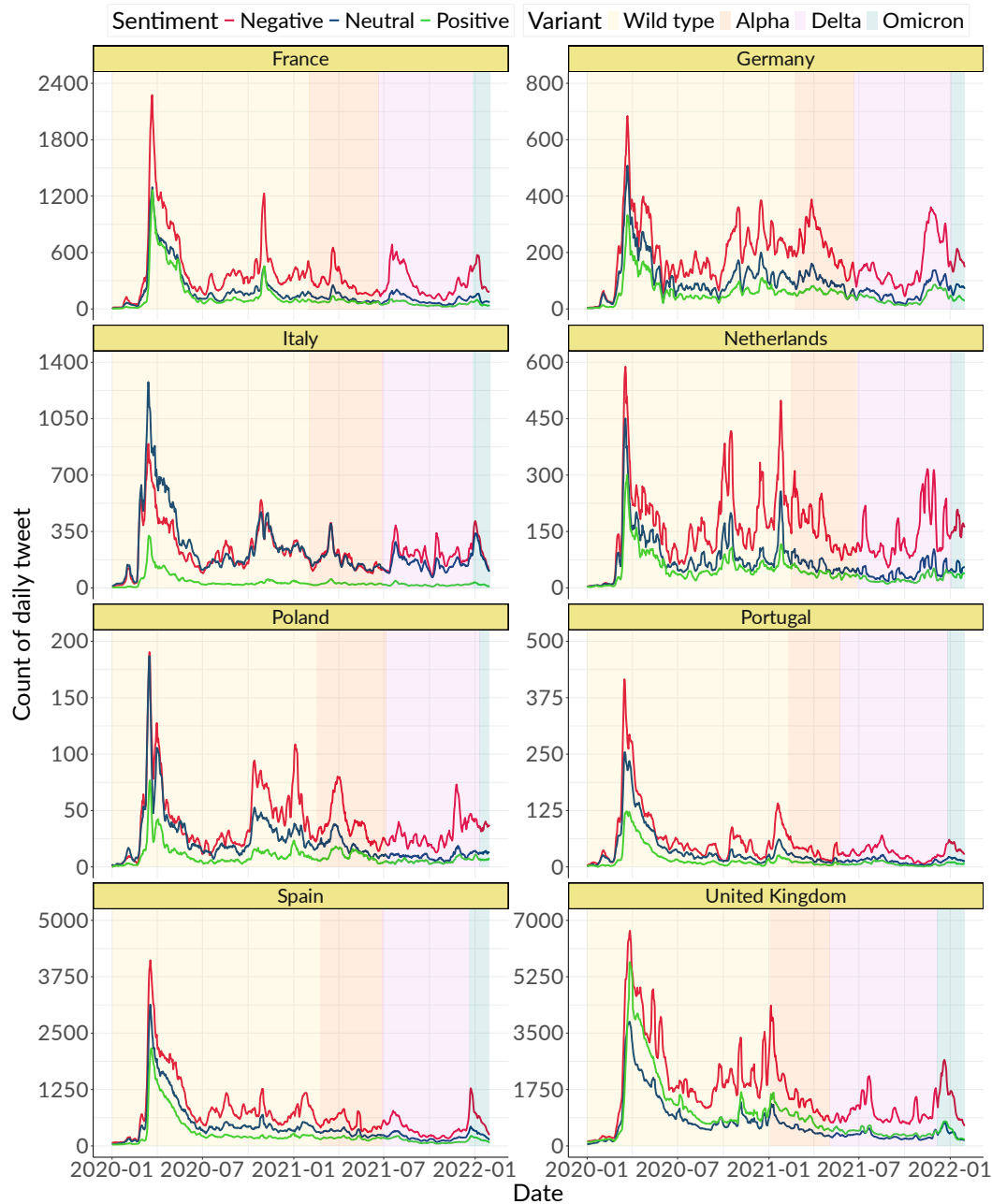
FIGURE 6.4: Distribution of sentiment polarity from 1 January 2020 to 30 January 2022 across all countries.

Sentiment polarity over time

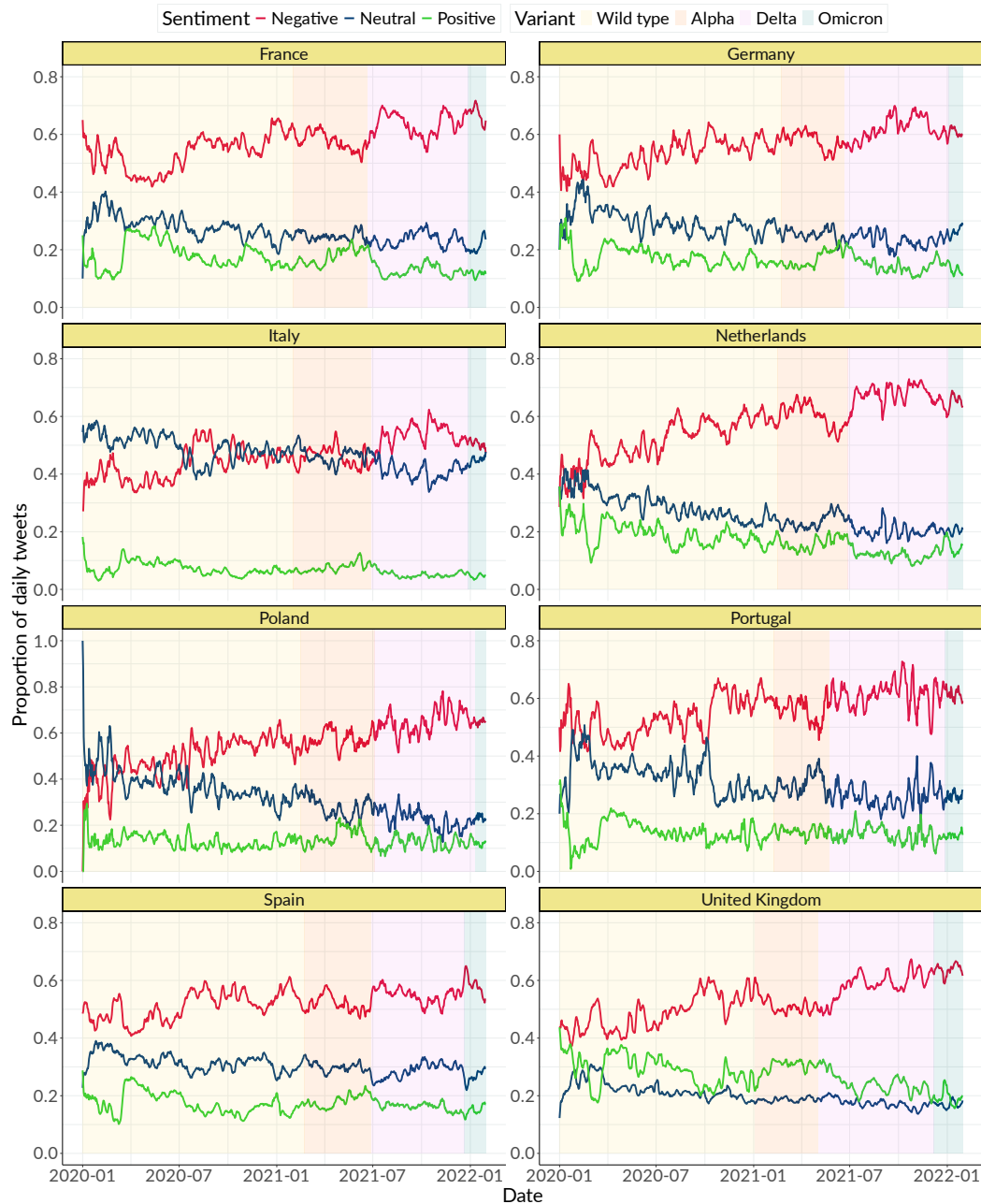
Tweet volume varied over the study period. The profiles per country, stratified by sentiment, are shown in Figure 6.5a. Across all countries, there was a pronounced peak in attention, which declined over time. However, this decline was more rapid for positive tweets and, in some instances, for neutral tweets. As the pandemic progressed, the difference between negative tweets and the other sentiment categories became more pronounced.

An alternative view is provided in Figure 6.5b, which shows daily proportions rather than raw counts. From this perspective, the contrast between sentiment classes is more apparent, particularly the trend of an increasing proportion of negative tweets.

An exception is Italy, where the dominant sentiment fluctuated between neutral and negative. Additionally, Spain showed a more clear separation between sentiment classes over the entire period. This view also highlights that the proportion of positive tweets is primarily driven by a relatively high share at the start of the pandemic, as illustrated by the UK.



(A) Daily count of tweets



(B) Daily proportion of tweets

FIGURE 6.5: Seven-day MA of the daily (A) count of tweets and (B) proportion of tweets, per country and stratified by sentiment. Background shading denotes periods of variant dominance.

Search term sentiment polarity

To understand the drivers of tweet distribution, I examined the most popular terms and their associated ratio of negative to positive tweet volume. I excluded the catch-all terms from this analysis as previously described in Section 4.6.3. The comparison was performed before and on or after 9th November 2020, which corresponds to Pfizer's

public announcement of positive Phase 3 trial results for its COVID-19 vaccine. The results are shown in Appendix D.1.

Before 9 November 2020, the top five most popular terms in almost all countries¹ included one or both of **lockdown** and **quarantine**, and all included **mask**, with each term appearing either in its original form or in translation. From 9 November 2020 onward, **vaccine** became the most popular term in every country, accounting for between 11% and 22%² of the total volume of tweets posted between 9 November 2020 and 1 January 2022. The exception was the UK, where **lockdown** remained the most popular term, although its share of total tweet volume fell from 25% to 16%.

Although I considered a wide range of search terms, a small subset captured a large proportion of public attention. The top five search terms throughout the entire study period accounted for an average of 41% (range: 32%–54%) of the total tweet volume. Catch-all terms contributed a further 37% (range: 27%–47%). The exact proportions and ratio of negative to positive tweets (negative-to-positive ratio) for each country are presented in Table 6.4. In all countries, the negative-to-positive ratio exceeded one, indicating more negative than positive tweets, but remained below two. Similarly, the negative-to-positive ratio for the catch-all terms (not presented in the table) ranged from 1.18 to 1.96.

The “Range Neg:Pos” column of Table 6.4 shows the range of negative-to-positive ratios across all search terms. I excluded terms that were only positive or only negative, as these typically correspond to search terms with a low tweet count. The results show considerable variation across countries and demonstrate that, although specific topics were highly polarised, they did not dominate the main Twitter discourse. Nonetheless, these polarising terms contribute to the overall high proportion of negative tweets observed. Moreover, the fact that catch-all terms account for a high proportion of tweet volume suggests that, despite using a wide range of search terms, there were potentially relevant terms that I did not capture.

I further investigated sentiment towards vaccines specifically. The negative-to-positive ratio of tweets posted on or after 9 November 2020 is shown in the “Vac Neg:Pos” column of Table 6.4. Across countries, values ranged from 1.19 to 3.97, indicating a consistent negative skew compared to positive tweets. The exact content of these

¹The Netherlands was the exception.

²I distinguished between **vaccine** and **vaccination**; however, grouping these two terms increased the range from 13% to 28%.

tweets is unknown; for example, they may have referred to vaccine availability rather than uptake. The UK showed a ratio notably lower than that of other countries, whereas Italy and the Netherlands were the most negatively skewed.

Country	Top 5 prop	Top 5 Neg:Pos	Range Neg:Pos	Vac Neg:Pos
France	0.51	1.14	[0.50; 18.60]	1.64
Germany	0.33	1.15	[0.29; 19.00]	2.07
Italy	0.33	1.63	[0.67; 44.00]	3.11
Netherlands	0.32	1.42	[0.50; 24.00]	3.97
Poland	0.39	1.94	[0.50; 15.00]	2.22
Portugal	0.48	1.69	[0.50; 13.00]	1.61
Spain	0.37	1.08	[0.59; 8.55]	2.03
United Kingdom	0.54	1.02	[0.73; 10.00]	1.19

TABLE 6.4: Negative-to-positive tweet ratio across categories, and the proportion of tweets attributed to the five most frequent terms. Columns (left to right) - Top 5 prop: proportion of tweets represented by the top five terms; Top 5 Neg:Pos: negative-to-positive (tweet) ratio among these terms; Range Neg:Pos: range of negative-to-positive ratios across all terms; Vac Neg:Pos: negative-to-positive ratio of vaccine-related tweets posted on or after 9 November 2020

Engagement

In the previous Sections, I discussed the polarity of tweets. However, because Twitter is inherently social, with users posting content for others to consume and interacting with one another, I now focus on engagement. Engagement was stratified into likes and responses, as previously described in Section 6.4.1. Responses were dominated by retweets, which on average accounted for 61% of all responses (range: 47%–73%), followed by replies at 32% (range: 22%–47%). Quote tweets formed a small minority, contributing only 7% (range: 5%–8%).

Country	Responses			Likes		
	μ	σ	Top 10% prop	μ	σ	Top 10% prop
France	1.9	19	0.95	7.1	84	0.92
Germany	4.3	88	0.96	8.8	196	0.93
Italy	1.4	11	0.94	3.6	29	0.90
Netherlands	4.5	44	0.97	13.5	112	0.94
Poland	4.7	144	0.95	6.8	121	0.92
Portugal	4.2	137	0.94	8.7	373	0.91
Spain	1.9	21	0.97	5.5	64	0.95
United Kingdom	2.6	125	0.94	7.7	261	0.91

TABLE 6.5: The mean (μ), standard deviation (σ), and proportion of total tweet engagements accounted for by the upper decile, across likes and responses.

Table 6.5 reports the mean, standard deviation, and the proportion of total engagement accounted for by the upper decile of posts. The Top 10% proportion columns show that at least 90% of all engagements were concentrated in the most popular 10% of posts. This skewed engagement reflects highly overdispersed distributions, with the extent of overdispersion varying across countries. The mean and standard deviation further illustrate this pattern, with extremely high standard deviations relative to the mean resulting in large coefficients of variation. The extreme skewness is further emphasised by the fact that the median number of likes and responses per post across all countries was zero.

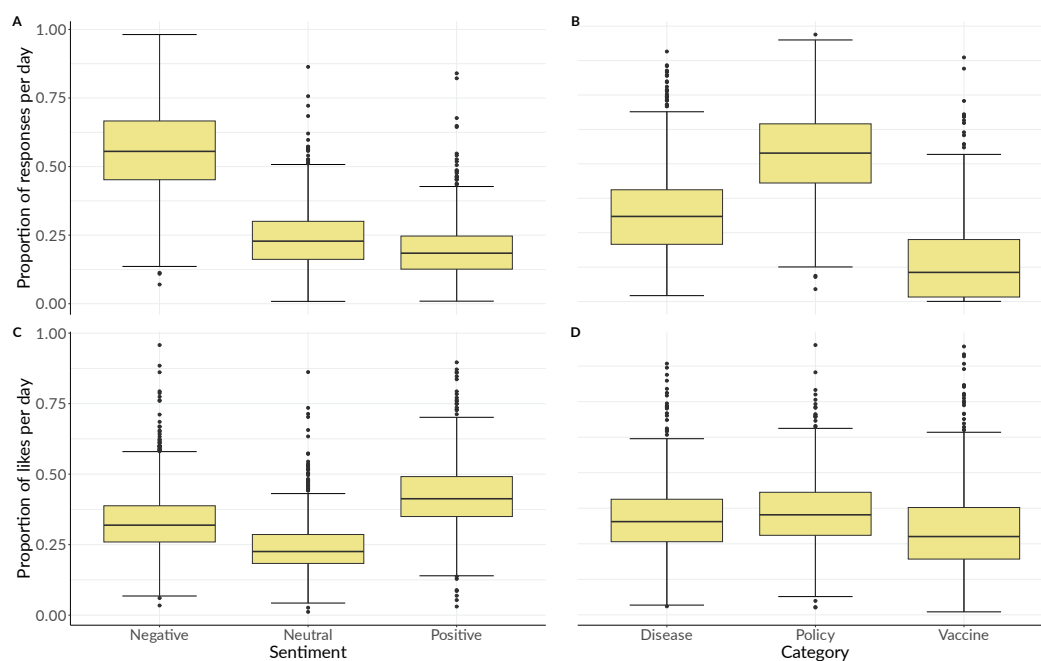


FIGURE 6.6: Box plots showing the median and Interquartile Range (IQR) of proportions of responses and likes per day, stratified by (A and C) sentiment polarity and (B and D) search term category. Proportions are obtained by normalising responses or likes per tweet, aggregating to group-level daily totals, and dividing by the corresponding overall daily total. Outliers are determined using Tukey's criterion ($< Q1 - 1.5 \times IQR$ or $> Q3 + 1.5 \times IQR$), where Q refers to quartiles.

Figure 6.6 presents the distribution of engagement across sentiment polarity and search term category. The first row corresponds to responses, and the second to likes. For each tweet, the number of responses or likes is normalised by tweet volume to account for differences across sentiment polarities and search term categories. These normalised values were then aggregated by day to obtain group-level totals, which were subsequently divided by the corresponding overall daily total to yield daily proportions for each group. Expressing engagement as proportions adjusts for changing

activity levels over the study period, allowing the plots to highlight general trends.

The first column of Figure 6.6 (Panels A and C) displays daily engagement proportions stratified by sentiment polarity. For responses (Panel A), negative tweets recorded the highest median (M) of 0.56 and the fewest outliers, while neutral and positive tweets recorded lower medians of 0.23 and 0.18. Panel C shows a different pattern from responses, with positive tweets having the highest median proportion of likes (M = 0.41), followed by negative tweets (M = 0.32) and neutral tweets (M = 0.23). The second column (Panels B and D) displays these proportions stratified by search term category. Policy-related tweets dominated responses (M = 0.54), with relatively few outliers, followed by disease-related tweets (M = 0.31) and vaccine-related tweets (M=0.11). For likes (Panel D), the same ordering across categories was observed, but the median values are closer together (0.35, 0.33, and 0.28 for policy-, disease-, and vaccine-related tweets, respectively), suggesting that engagement with tweets was more uniform across categories. In both sentiment and search term category plots, there were many outliers, suggesting that on some days, tweets related to specific topics and with a specific sentiment accounted for a disproportionately large share of total engagement.

6.6.2 Correlation of tweet volume with epidemiological and public policy variables

Epidemiological variables and vaccination uptake

In this Section, although vaccination uptake is a public policy variable, I group it with epidemiological variables for ease of presentation. Across all variables, I observed moderate to low correlations with tweet volume as shown in Figure 6.7. In the following results, the mean (μ) Kendall's τ across all countries was calculated using absolute values ($|\tau|$), which removed differences in the direction of association so that the mean reflected only the magnitude of association. In contrast, the standard deviation (σ) was calculated from the signed τ values, providing a better measure of variability across countries. Cases (mean (μ): 0.29, standard deviation (σ)=0.18), hospitalisations (μ =0.31, σ =0.09), and deaths (μ =0.33, σ =0.08) showed similar levels of correlation, while vaccination uptake had a lower average correlation (μ =0.10, σ =0.13). Deaths exhibited the least variation across countries, whereas cases showed the most. Correlations for all epidemiological variables were statistically significant,

but for vaccination uptake, only Italy, the Netherlands, and the UK showed significant associations.

The strength and direction of correlations between tweet volume and variables varied by country. The UK had a negative but low correlation between tweet volume and cases, while Italy and Poland showed the highest correlations for this outcome. For hospitalisations, countries using weekly data exhibited wider confidence intervals, though overall heterogeneity was lower. Poland again had the strongest correlation, similar to its results for cases, representing the highest association across all variables. Portugal and the UK had lower correlations and were distinct from the other countries. For deaths, Spain and the Netherlands showed the highest correlations, while Portugal, Poland, and the UK had the lowest. Correlations with vaccination uptake were weak in most countries, with the UK, Italy, and the Netherlands as notable exceptions.

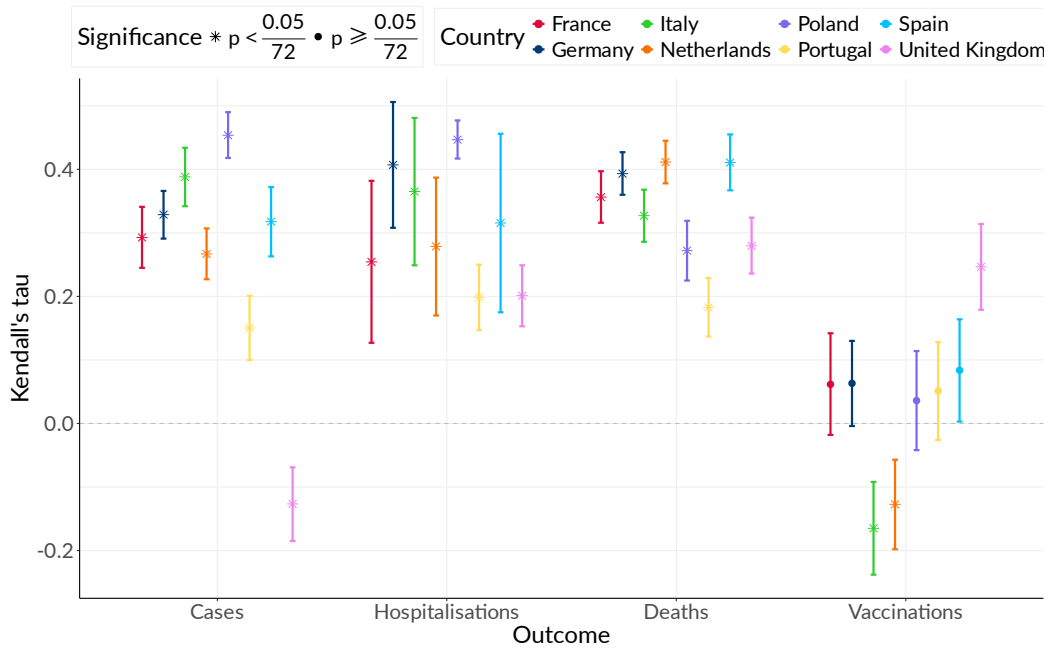
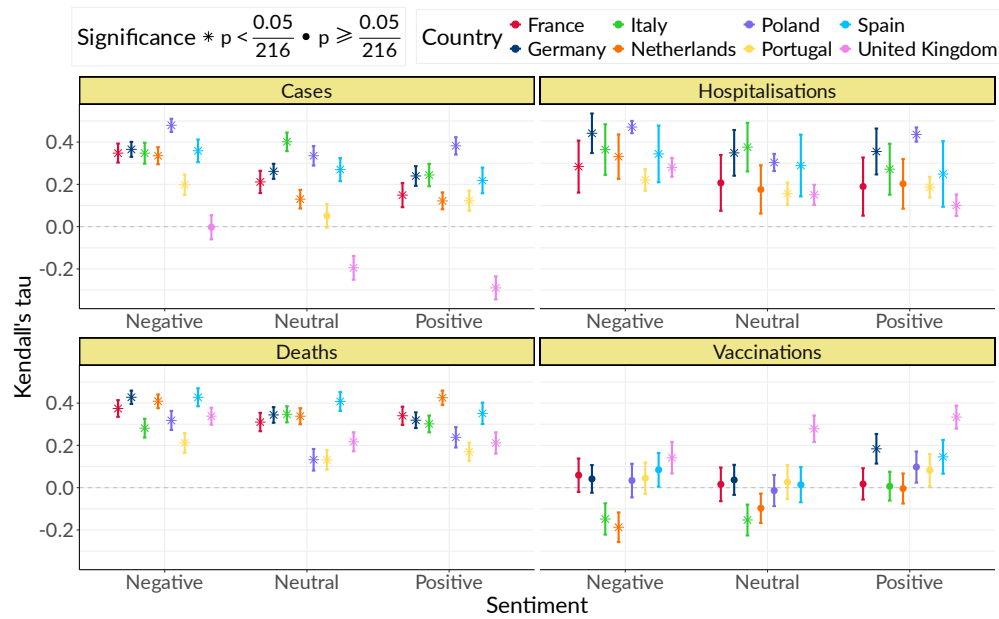


FIGURE 6.7: Kendall's τ capturing the correlation between tweet volume and epidemiological variables, with 95% confidence intervals, coloured by country.

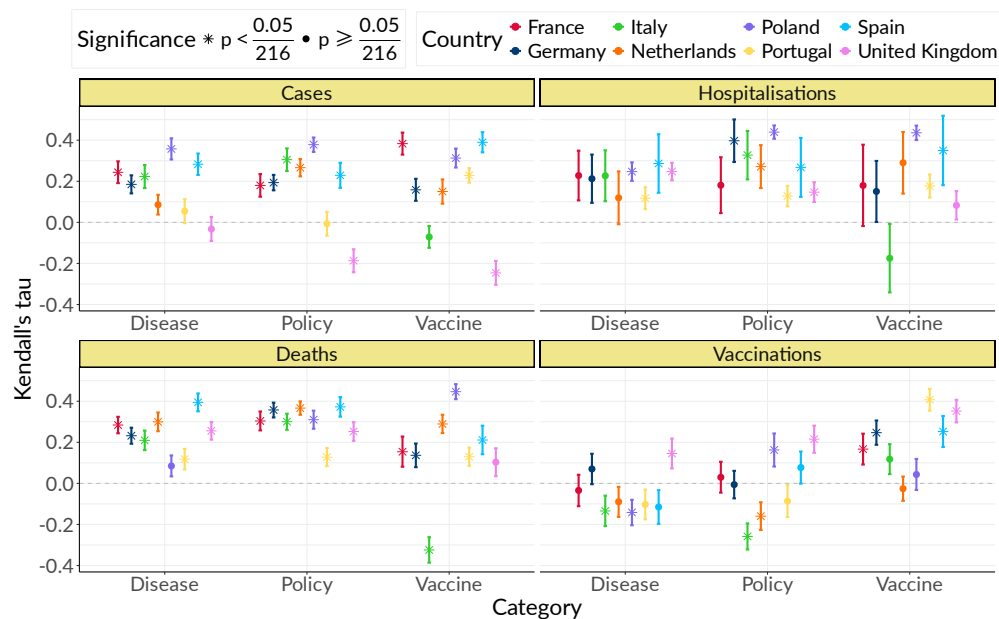
Figure 6.8a presents correlations stratified by sentiment. Across variables, differences between sentiment categories were heterogeneous, but for epidemiological variables a consistent tendency emerged, with negative tweets showing higher correlations than neutral or positive ones. The mean correlation across countries was 0.30 for negative tweets ($\sigma=0.15$), compared with 0.23 ($\sigma=0.20$) for neutral and 0.22

($\sigma=0.19$) for positive tweets. For hospitalisations, negative tweets again correlated more strongly ($\mu=0.34$, $\sigma=0.08$) than neutral ($\mu=0.25$, $\sigma=0.089$) or positive tweets ($\mu=0.25$, $\sigma=0.10$), although the country-specific estimates were associated with larger confidence intervals. For deaths, the gap narrowed but remained in the same direction (negative: $\mu=0.35$, $\sigma=0.08$ compared to neutral $\mu=0.28$, $\sigma=0.11$ and positive: 0.29 , $\sigma=0.08$). Vaccination uptake, by contrast, showed broadly similar means across sentiment, with positive tweets only slightly higher ($\mu=0.09$, $\sigma=0.11$) than neutral ($\mu=0.08$, $\sigma=0.13$) and negative ($\mu=0.09$, $\sigma=0.12$). Notable differences in correlation across sentiment were observed in the Netherlands and the UK for several variables, while France showed marked divergence between positive and negative tweets for cases, Poland displayed lower correlations for neutral tweets in hospitalisations and deaths, and Germany and Italy exhibited greater divergence between positive and negative tweets for vaccination uptake.

Figure 6.8b presents correlations across countries stratified by search term categories. For epidemiological variables, the differences across tweet categories were smaller compared to sentiment. Cases showed higher correlations with vaccination tweets ($\mu=0.24$, $\sigma=0.22$) and policy tweets ($\mu=0.22$, $\sigma=0.18$) than with general disease tweets ($\mu=0.18$, $\sigma=0.13$). Hospitalisations were more highly correlated with policy tweets ($\mu=0.27$, $\sigma=0.11$) compared to vaccine tweets ($\mu=0.23$, $\sigma=0.19$) and disease tweets ($\mu=0.21$, $\sigma=0.06$). Deaths followed a similar but more pronounced pattern, with policy tweets showing the strongest correlation ($\mu=0.30$, $\sigma=0.08$), followed by disease tweets ($\mu=0.23$, $\sigma=0.10$) and vaccine tweets ($\mu=0.22$, $\sigma=0.22$). For vaccination uptake, differences across tweet categories were clearer, as expected, with vaccine tweets showing higher correlations ($\mu=0.20$, $\sigma=0.15$) than policy ($\mu=0.12$, $\sigma=0.16$) or disease tweets ($\mu=0.10$, $\sigma=0.10$). Vaccination uptake showed the most variability, with the highest standard deviation among the epidemiological variables. At the country level, Italy consistently stood out with larger differences across categories, driven by greater separation between vaccine-related correlations and those of other categories. Portugal, France, Germany, Poland, and the UK exhibited stronger outcome-specific variation, with notable examples including Portugal's strong association with vaccine tweets and Poland's divergence in deaths.



(A) Sentiment



(B) Category

FIGURE 6.8: Kendall's τ between tweet volume and epidemiological variables or vaccination uptake, with 95% confidence intervals, coloured by country and stratified by (A) sentiment and (B) search term category.

Policy stringency variables

The correlation between tweet volume and policy stringency variables varied, as shown in Figure 6.9. The highest correlation was observed with raw stringency ($\mu=0.30$, $\sigma=0.10$). Poland and France had the highest correlation, followed closely by Spain and

the UK. Portugal and Italy had lower correlations compared to the other countries considered. The remaining variables which look at some form of change in stringency, showed, on average, lower correlations compared to the raw stringency. Poland stood out with a higher correlation in many of the stringency change variables considered. Both the change in stringency ($\mu=0.15$, $\sigma=0.06$) and the absolute change in stringency ($\mu=0.13$, $\sigma=0.05$) were similar. The positive change (implementation of restrictions) ($\mu=0.20$, $\sigma=0.07$) had a higher average correlation compared to the negative change (easing of restrictions) ($\mu=0.09$, $\sigma=0.07$).

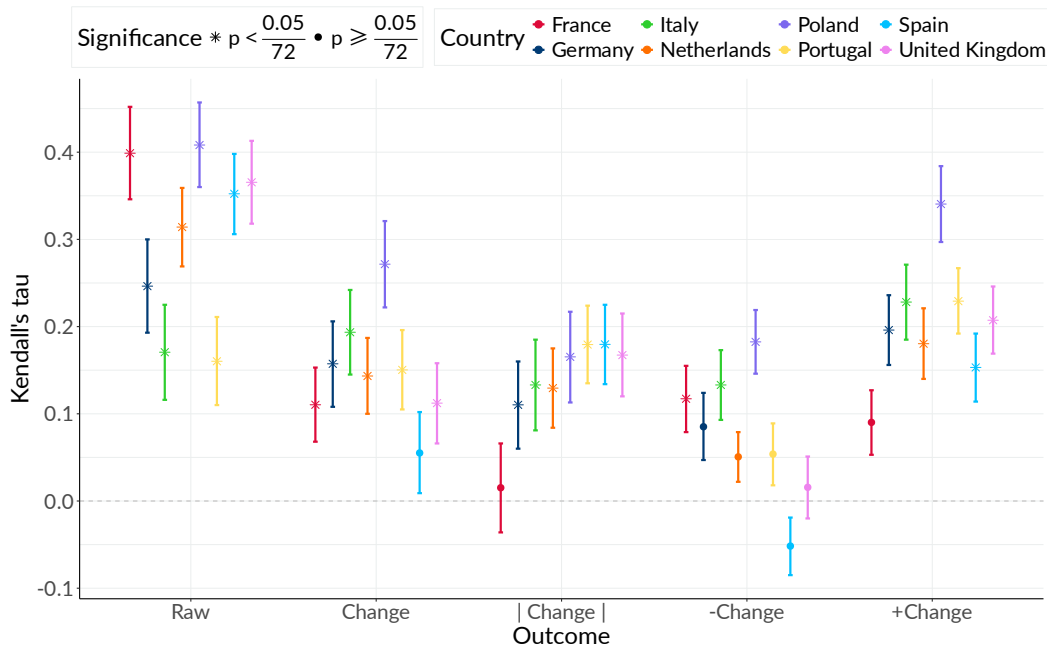
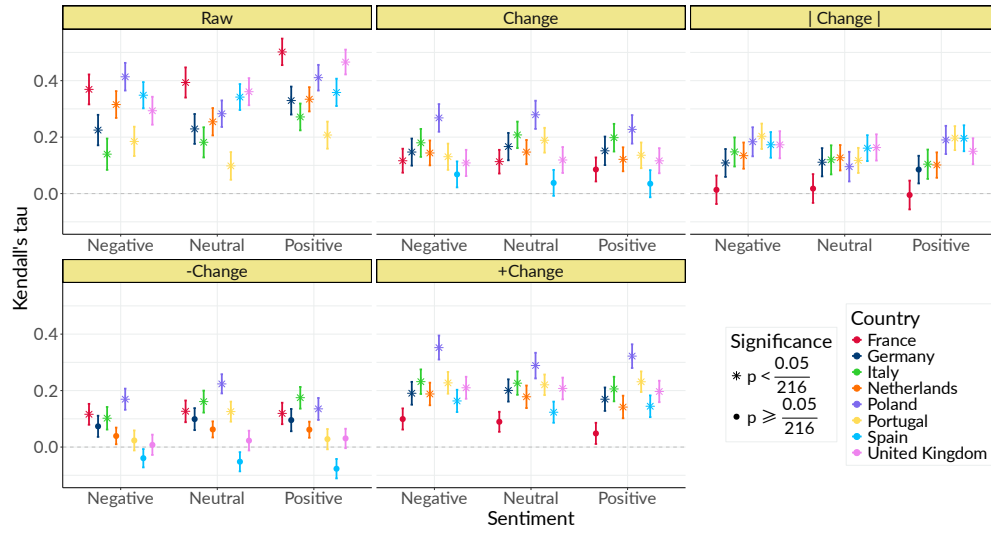


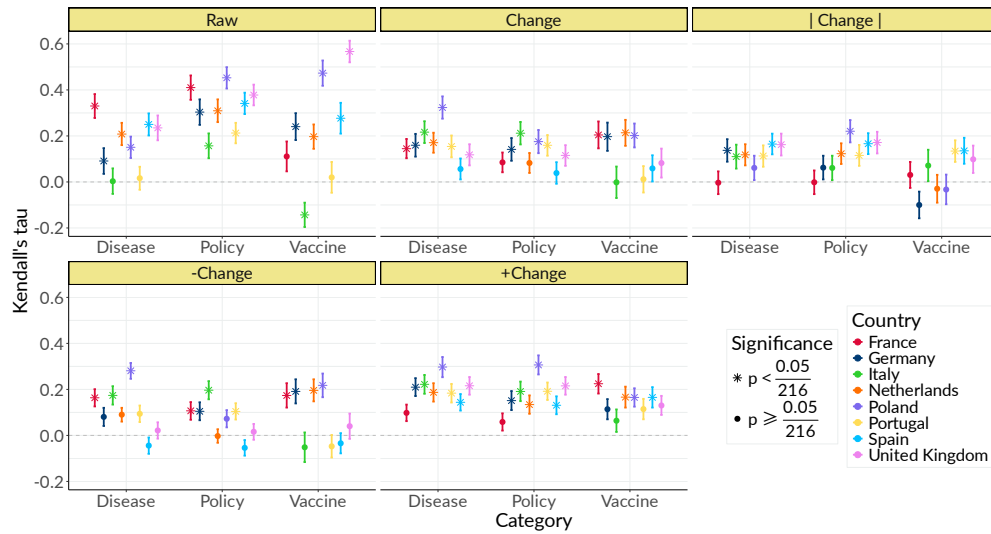
FIGURE 6.9: Kendall's τ between tweet volume and policy stringency variables, with 95% confidence, intervals coloured by country. **Raw** stringency refers to the seven-day MA-smoothed variable. **Change** denotes the one-day lagged difference of the smoothed profile, while **|Change|** represents its absolute value. **-Change** and **+Change** correspond to the negative and positive components of change, respectively, with values set to zero in the absence of signed changes.

Analogously to the epidemiological variables, I further stratified correlations by sentiment and search term category. Figure 6.10a presents correlation stratified by sentiment. Compared to the epidemiological variables and vaccination uptake, the average differences across sentiment were even smaller. The raw stringency variable displayed the largest divergence across classes, with higher correlations for positive tweets ($\mu=0.36$, $\sigma=0.10$), followed by negative ($\mu=0.29$, $\sigma=0.10$) and neutral tweets ($\mu=0.27$, $\sigma=0.10$). France, Germany, Italy, and the UK exhibited the most notable differences between positive and negative tweets for this variable. The remaining

variables exhibited greater consistency, with only minor differences across sentiment. This is partly attributable to their lower overall correlations, with mean differences between classes ranging from 0.01 to 0.04. Moreover, the per-country correlations observed for each class mirrored the patterns identified for total tweet volume.



(A) Sentiment



(B) Category

FIGURE 6.10: Kendall's τ between tweet volume and policy stringency variables with 95% confidence intervals coloured by country, stratified by (A) sentiment and (B) search term category. **Raw** stringency refers to the seven-day MA-smoothed variable. **Change** denotes the one-day lagged difference of the smoothed profile, while **|Change|** represents its absolute value. **-Change** and **+Change** correspond to the negative and positive components of Change, respectively, with values set to zero in the absence of signed changes.

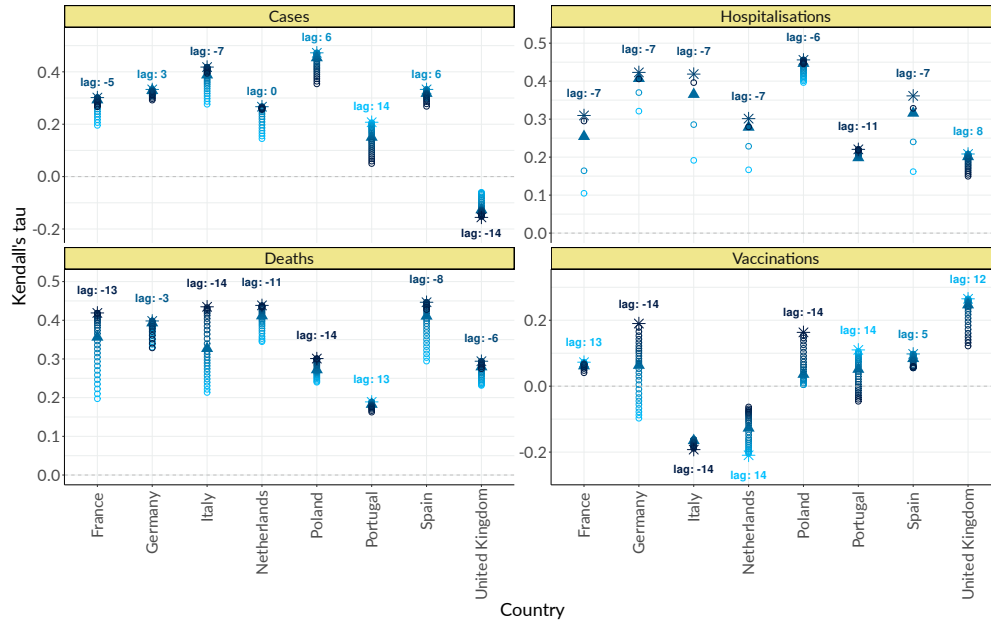
Stratifying by search term category, as shown in Figure 6.10, revealed greater heterogeneity. The raw stringency variable correlated most strongly with policy-related

tweets ($\mu=0.32$, $\sigma=0.10$), followed by vaccine tweets ($\mu=0.25$, $\sigma=0.23$) and disease tweets ($\mu=0.16$, $\sigma=0.12$). The high standard deviation for vaccine tweets reflected comparatively high positive correlations in Poland and the UK, in contrast to a weaker negative correlation in Italy. While still minimal, differences across categories were larger than those observed for sentiment. The change variable was most correlated with disease-related tweets ($\mu=0.17$, $\sigma=0.08$), compared with policy ($\mu=0.13$, $\sigma=0.06$) and vaccine tweets ($\mu=0.12$, $\sigma=0.09$). In contrast, absolute change was more correlated with policy ($\mu=0.12$, $\sigma=0.07$) and disease tweets ($\mu=0.11$, $\sigma=0.06$) than with vaccine tweets ($\mu=0.08$, $\sigma=0.09$). A similar pattern held for positive change, where correlations were highest for policy ($\mu=0.19$, $\sigma=0.06$), followed by disease ($\mu=0.17$, $\sigma=0.07$) and vaccine tweets ($\mu=0.14$, $\sigma=0.05$). Negative change, by contrast, correlated equally with disease ($\mu=0.12$, $\sigma=0.10$) and vaccine tweets ($\mu=0.12$, $\sigma=0.12$), but less with policy tweets ($\mu=0.08$, $\sigma=0.08$). Country-level variability was also evident, with larger differences particularly in Italy (raw stringency, change, and negative change), France (raw stringency and positive change), Poland (raw and absolute change), Germany (absolute change), and the UK (raw stringency).

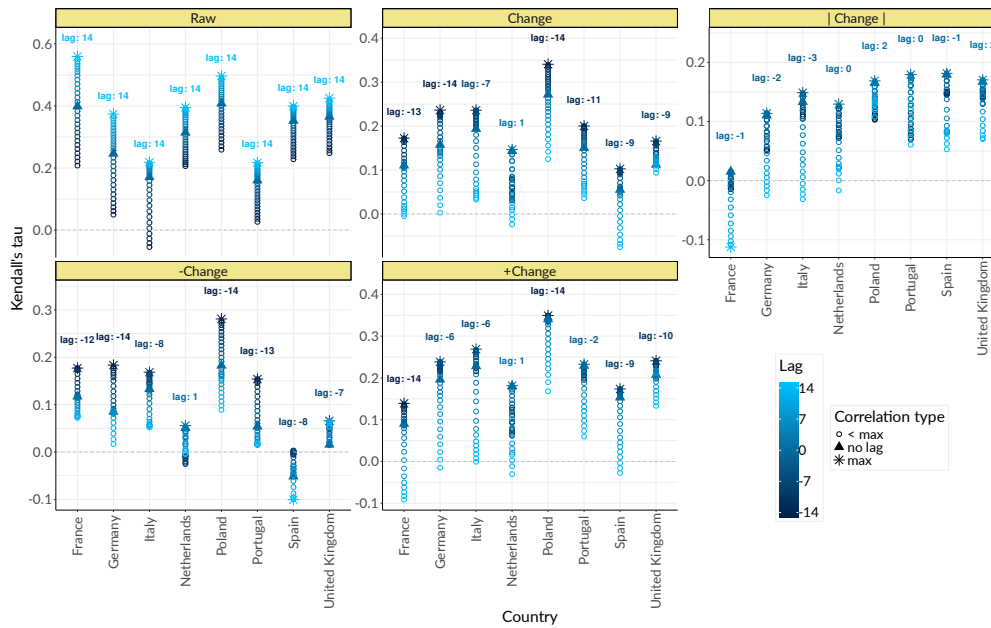
Cross-correlations of variables of interest with tweet volume

To assess the temporal relationships between tweet volume and each variable of interest, I used cross-correlations. For each country–outcome pair, 25 correlations were computed: 14 where tweet volume lagged (preceded) the variable of interest, one zero-lag correlation (as presented previously), and 14 where tweet volume led (followed) it. The lag at which the (absolute) maximum correlation was achieved is annotated for each country. Negative values indicate that the maximum occurred when tweet volume preceded the variable. In contrast, positive values indicate that the maximum occurred when tweet volume followed the variable. While these results do not imply causation, they highlight potential predictive associations. Results for epidemiological variables and vaccination uptake are shown in Figure 6.11a and for policy stringency variables in Figure 6.11b.

Figure 6.11a illustrates that shifting tweet volume almost always increased correlations, although the magnitude of this effect varied across variables of interest and countries. For cases, tweet volume preceded the outcome in half of the countries, and vice versa in the other half. The Netherlands was the only instance across all epidemiological variables and vaccination uptake where the maximum correlation occurred



(A) Epidemiological variables



(B) Policy stringency variables

FIGURE 6.11: Comparison of Kendall's τ cross-correlations between tweet volume and (A) epidemiological variables and (B) policy stringency variables, coloured by tweet lag (between -14 and 14) and stratified by country. Darker shades represent lags and lighter shades leads. A wider spread between symbols reflects greater variation across lags. Raw stringency refers to the seven-day MA-smoothed variable. Change denotes the one-day lagged difference of the smoothed profile, while $|\text{Change}|$ represents its absolute value. $-\text{Change}$ and $+\text{Change}$ correspond to the negative and positive components of Change, respectively, with values set to zero in the absence of signed changes.

at no lag. Correlation sensitivity to lagging varied across countries for cases but was often low in terms of the direction of the maximum correlation. For hospitalisations and deaths, the maximum correlation almost always occurred when tweets preceded the outcome, except for UK hospitalisations and Portuguese deaths. Hospitalisations typically peaked at a lag of seven days, reflecting the use of weekly data. In contrast, deaths peaked at longer lags, often exceeding one week, and showed greater cross-country sensitivity, particularly in France, Italy, and Spain. For vaccination uptake, the optimal lag direction was less consistent, with tweet volume lagging uptake in three countries and uptake lagging tweet volume in five countries. In most instances, however, the maximum correlation occurred at or near the 14-day limit. Spain was the exception, reaching its maximum at five days, although correlations there remained relatively stable across lags. Like deaths, vaccination uptake showed higher sensitivity to lagging. However, in contrast, the lowest sensitivity was observed for France, Italy, and Spain.

Overall, policy stringency variables were more sensitive to lags compared to epidemiological variables and vaccination uptake, with larger increases relative to the no-lag correlation, as shown in Figure 6.11b. For the raw variables, the maximum correlation in every country was observed when the tweet volume followed the outcome by 14 days. The change variable and its positive and negative constituents also reached their maximum correlations at longer lags, often at seven days or more. However, in contrast to the raw variable, these peaks typically occurred when tweet volume preceded the outcome. The Netherlands was a notable exception, with the maximum correlation observed when the tweet volume followed the outcome by only one day. In line with this, absolute change also behaved differently, with correlations peaking at much lower lags (between -3 and 2 days) and showing no consistent direction, with three countries achieving the maximum correlation at no lag.

The results of cross-correlations, stratified by sentiment and search term category, are presented in Appendix D.2.1 for epidemiological variables and vaccination uptake, and in Appendix D.2.2 for policy stringency variables. These appendices mirror Figures 6.8 and 6.10, but report the maximum correlation observed across all lags and the time shift at which it was achieved. In general, cross-correlation increased the strength of associations, but no single sentiment or search term category exhibited a notable improvement. Moreover, the lag at which the maximum correlation occurred varied

across variables of interest and countries.

6.7 Discussion

6.7.1 The evolution of attention and sentiment polarity

In a step toward understanding the utility of Twitter in a pandemic setting, I investigated how COVID-19 topics evolved on the platform, and how this discourse was associated with both epidemiological and public policy variables.

Across all countries, the onset of the pandemic was accompanied by a surge in interest. Once this initial interest waned, attention was punctuated by smaller subsequent spikes, though the extent of decline varied between countries. Overall, half or more of all tweets were negative. Italy was an exception to this trend, characterised by a higher proportion of neutral tweets than negative ones, and a very low share of positive tweets. The literature reports mixed perspectives on the overall distribution of public opinion, ranging from positive to negative majorities (Cheng et al., 2021; Wang et al., 2022a; Boon-Itt and Skunkan, 2020; Xie et al., 2023; Cheng, Han, and Liu, 2023). However, direct comparison across studies is difficult, given differences in search strategies, study periods, sentiment classification methods, and geographic contexts.

In addition to similarities in the distribution of sentiment and tweet volume, there was also a similarity among the popular terms during the pandemic. I found that **lockdown**, **quarantine**, and **mask** were in the top five terms in all countries before the introduction of vaccines. Once vaccines were introduced, the term **vaccine** became the most popular in every country, apart from the UK, where **lockdown** remained the most popular. I found that a small subset of terms accounted for a large portion of the total tweet volume, indicating that certain restrictions garnered substantial attention. However, none of these terms were drivers of the larger proportion of negative tweets, as there were similar numbers of negative and positive tweets. This result suggests that specific terms provoked a strongly negative response. Similarly, I found that engagements were highly disproportionate, with 90% of total engagements concentrated in the top 10% of posts. Within this engagement pattern, negative tweets and policy-related tweets accounted for a larger median share of daily interactions, suggesting that such content was more likely to provoke responses. By contrast, positive tweets

captured a larger median share of daily likes, indicating that although they generated fewer overall responses, they tended to elicit more supportive reactions.

The fact that tweet volume followed a declining trajectory after the initial surge is consistent with the patterns described in the issue-attention cycle (Downs, 1972; Yin and Lui, 2024). In this context, the impact of waning attention manifests in two ways: first, public focus on long-standing policy restrictions diminishes over time, and second, attention to the pandemic as a whole recedes. Certain events were able to reignite interest temporarily, for example, the emergence of vaccines became the most popular search term following their introduction. However, these surges occurred within a broader trajectory of declining attention, with no single event capable of fully restoring attention to the levels seen at the onset of the pandemic.

Alongside this cyclical decline, it is important to consider the role of agenda-setting in shaping public attention. The agenda-setting theory suggests that the media do not directly tell people what to think, but rather influence what issues they consider important, through the topics they highlight and the ways in which these are framed (McCombs and Shaw, 1972). Terms such as “new normal”, propagated through news channels, undoubtedly contributed to public acceptance of the situation and framed the level of attention it received (Corpuz, 2021; Graso, 2022). However, in the age of social media, where the influence of traditional mass media is fragmented, agenda-setting must also be understood in a participatory context, where the public itself generates and amplifies content (Feezell, 2018). In such an environment, echo chambers, gatekeeping, and filter bubbles may substantially shape not only the public agenda but also sentiment and its persistence (Flaxman, Goel, and Rao, 2016; Ross Arguedas et al., 2022). Against this backdrop, the persistence of negativity within the dataset is particularly noteworthy. Although overall attention declined, negative tweets decreased at a slower rate and came to account for a larger proportion of total tweet volume as the pandemic progressed. They also exhibited sharper peaks during periods of high tweet activity. This pattern may, at least in part, reflect the reinforcing dynamics of this environment in sustaining and amplifying negative sentiment.

An additional important contribution of this study is its focus on multilinguality, which necessarily entails engagement with the multicultural contexts embedded in language (Imai, Kanero, and Masuda, 2016). Because sentiment is expressed through

language and situated within cultural context, differences in sentiment distributions across countries may partly mirror these cultural variations (Mula et al., 2021; Bavel et al., 2020). At the same time, such cultural influences are difficult to disentangle from methodological considerations. As discussed in Section 5.9, one challenge is determining how accurate a classifier must be for downstream tasks. In this Chapter, the model used classified many Italian tweets as neutral, mirroring an outlying case in its training dataset, where Italian tweets were disproportionately labelled neutral. This raises a further question warranting future investigation, which is to what extent do observed cross-country differences in sentiment distribution reflect cultural context, as conveyed through language, versus residual effects of classifier performance?

6.7.2 Correlation of tweet volume with epidemiological variables and public policy variables

Following the investigation of sentiment across countries, I also considered correlations between tweet volume and epidemiological variables, vaccination uptake, and policy stringency variables, using Kendall's τ . I found similar associations across most epidemiological variables and tweet volume, suggesting comparable responses regardless of outcome severity. By contrast, correlations with vaccination uptake exhibited the greatest variance across countries.

Policy stringency variables were also associated with tweet volume, though the strength of these associations varied across variables of interest. The highest correlation was observed with the raw stringency variable, suggesting that public attention was more closely aligned with the overall level of restrictions than with changes in them. Across the change variables, correlations were highest for the implementation of restrictions and lowest for their easing. One caveat is that the change in stringency was coded by the date of implementation rather than the date of announcement. Future work should therefore examine policy announcements directly, for example, using the Global Database of Events, Language, and Tone (GDELT) dataset, as in the study by Owuor and Hochmair (2023).

Stratifying correlations by sentiment and search term category revealed relatively minor differences, but some tendencies emerged. Negative tweets correlated more strongly with epidemiological variables, while vaccination uptake showed little difference across sentiments and was marginally higher for positive tweets. For policy

stringency variables, sentiment differences were generally minor, with only raw stringency showing a tendency toward positive tweets. On average, increases in epidemiological variables elicited stronger negative responses. By category, policy-related tweets most frequently displayed stronger correlations, although vaccine- and disease-related tweets at times surpassed them. As expected, vaccine-related tweets were most strongly associated with vaccination uptake.

Extending this analysis by incorporating temporal lags revealed additional patterns. Allowing for lags generally increased correlations, though the extent of improvement varied. This was true for both total tweet volume and analyses stratified by search term category or sentiment. However, no particular search term category or sentiment consistently produced stronger correlations than the others. Tweet volume most often preceded hospitalisations and deaths, with peaks typically emerging at weekly or longer intervals. By contrast, cases and vaccination uptake showed less consistent patterns, with tweets sometimes leading and sometimes lagging these variables of interest, although maximum correlations were usually observed at longer shifts. One possible explanation is that more severe outcomes, such as hospitalisations and deaths, are highly visible and emotionally salient, making people more responsive to increases in these outcomes. As these epidemiological variables begin to rise, people are prompted to tweet about them, causing the peak in tweet volume to occur before the peak in the epidemiological variables. In contrast, cases and vaccination uptake are less directly observable and therefore associated with more variable patterns of attention.

Policy stringency variables showed a different dynamic. Tweet volume consistently followed raw stringency by two weeks, suggesting that public attention accumulated after restrictions had been in place. In contrast, changes in stringency often exhibited negative lags, with tweet volume lagging behind the outcome. This may reflect responses to announcements of upcoming adjustments. However, it could also be confounded by the fact that restrictions remain constant until altered, meaning reactions are directed at the ongoing level of stringency itself, consistent with the positive lag observed for the raw index.

The strength of association found here is weaker than findings in the literature discussed in Section 6.3, which considered similar tweet variable correlates (cases, vaccination uptake, and deaths) (Doogan et al., 2020; Cheng et al., 2021; Turiel, Fernandez-Reyes, and Aste, 2021; Cheng, Han, and Liu, 2023; Effenberger et al., 2020). As noted in the discussion on sentiment, direct comparison across related work is complicated by design choices. The wide range of search terms used here, while capturing greater breadth, may have introduced noise. Although these terms accounted for only a small proportion of the total tweet volume, their inclusion contrasts with studies using narrower sets of terms that reported higher correlations. The observation that maximum correlations often occurred between historical tweet volume and hospitalisations or deaths over the extended study period is promising. This result is consistent with previous work, which showed that tweet volume tends to lag behind deaths (Turiel, Fernandez-Reyes, and Aste, 2021; Cheng et al., 2021). In the UK, a maximum correlation was achieved when tweet volume preceded reported cases by 14 days, comparable to the lag observed by Effenberger et al. (2020) based on relative search volume indices from Google Trends.

A further design consideration is the extended study period. Across all the studies considered in Section 6.3, except for Cheng, Han, and Liu (2023), the study period was six months or less around the start of 2020. As the pandemic progressed, both restrictions and behavioural responses evolved (Bruin et al., 2022; Haktanir et al., 2022; Hale et al., 2021). The dynamic nature of a pandemic, coupled with changes in the underlying distributions, complicates comprehensive retrospective analyses; it may be more appropriate to examine specific periods in isolation, such as by variant. The Omicron wave presents an unfair comparison, as it contrasts a sharp increase in cases with a waning level of attention on social media across all countries. This was particularly evident in the UK, where a negative correlation was observed. During the Omicron period, tweet volume decreased substantially compared with earlier in the pandemic, driven primarily by a decline in positive and neutral tweets, as reflected in the sentiment-stratified correlations for cases.

Beyond design decisions, limitations in the underlying data may also have influenced the results. There was substantial variance in tweet volumes across countries, raising concerns about comparability. This issue is highlighted by reliance on geotagged data, as Twitter estimates that only $\sim 1\text{--}2\%$ of tweets are geotagged (Twitter, 2023). These

tweets reflect only a subset of users, for whom no demographic information is available. The absence of demographic detail further restricts alignment with epidemiological variables, which are often stratified by age or other factors. Combined with the possibility that users who geotag differ systematically from those who do not, these factors introduce the risk of sampling bias (Sloan and Morgan, 2015; Huang and Carley, 2020). Importantly, these challenges are not unique to this study but extend more broadly to research using Twitter data.

Limitations in epidemiological reporting further compound these issues. Early in the pandemic, coverage gaps and reporting delays in hospitalisations and deaths likely skewed comparisons with higher tweet volumes. In Poland, for instance, the lower correlation between deaths and tweet volume may be attributable to under-reporting (Walkowiak and Walkowiak, 2022). Hospitalisations in Poland were similarly affected but, because data were truncated and missed part of the initial peak in tweet volume, its correlation was not as low, supporting this interpretation. Beyond incomplete coverage, revisions and backfills, weekend reporting cycles, and changes in testing or reporting rates also affect the reliability of these series (Badker et al., 2021; Alvarez et al., 2023). Smoothing can partially mitigate the influence of day-of-the-week effects and retrospective revisions, but it cannot resolve issues of incomplete coverage or contextualise the extent of reporting. To address these limitations, future work should extend the analysis to incorporate additional measures such as excess mortality, testing rates, or the effective reproduction number (R_t), which would provide a more comprehensive perspective (Msemburi et al., 2023; Hasell et al., 2020). Moreover, considering growth rates rather than raw levels may be more informative for capturing how changes in the trajectory of the pandemic align with fluctuations in public attention.

6.8 Conclusion

From a pandemic preparedness perspective, it is essential to understand how to utilise social media more effectively, which is now deeply embedded in the zeitgeist of modern society. During the COVID-19 pandemic, tweet volume was responsive to both epidemiological variables and related public policy, with the majority of these responses being negative across countries. This responsiveness highlights the value of tweet volume as a barometer for gauging public focus and sentiment.

Tweet volume was low to moderately associated with these epidemiological variables and public policy variables. The observation that maximum correlations often occurred between historical tweet volume and epidemiological variables is particularly promising, as it suggests that the level of disease-related Twitter discourse may serve as a predictive indicator of disease trajectory. Moreover, the consistent two-week delay between maximum correlations of tweet volume and the level of policy stringency provides a useful benchmark for policymakers, helping to distinguish when responses reflect current restrictions and enabling a retrospective assessment of public reaction.

As a concluding remark, while this study focused on Twitter, it lays a foundation that can be extended more broadly to similar social media platforms, especially given the recent cessation of the Twitter API.

7

Time series modelling to quantify associations between tweet, epidemiological, and public policy variables

7.1 Overview

Chapter 6 demonstrated a moderate correlation between tweet volume, epidemiological variables, and policy stringency. In this Chapter, I extend that analysis using time series models, to assess if they can capture the dynamic relationships between variables more robustly than Kendall's τ . To maximise the chances of detecting a relationship, I apply a range of models of varying complexity, spanning four ML models and seven DL models, across multiple experimental designs. Several of these DL architectures, to the best of my knowledge, have not previously been applied in this context. I evaluate both directions of association, examining whether historic tweet and policy stringency relate to epidemiological variables and conversely whether epidemiological variables and policy stringency relate to tweet volume.

7.2 Introduction

In this Chapter, I applied time series models to build on the correlative associations established in Chapter 6. While these models are typically employed to maximize forecasting accuracy, my emphasis here was different, as outlined in Chapter 3. I used them as a framework to examine the extent of association between variables, evaluated by the extent to which one variable improves the forecast of another.

This approach mirrors the logic of Granger causality, which asks whether past values of one series improve predictions of another beyond what is achievable from its history (Granger, 1969; Shojaie and Fox, 2022). Formally, Granger causality is usually assessed within a Vector Autoregression (VAR) model by testing whether the lagged coefficients of one variable are jointly zero in the equation for another, using procedures such as Wald or F-tests. Because each variable has its own equation in a VAR model, such tests allow predictive content to be assessed in both directions.

For example, one can ask whether tweet volume or policy stringency can predict epidemiological variables, and conversely, whether epidemiological variables or policy stringency can predict tweet volume. This provides a rigorous, in-sample test of predictive associations.

In contrast to Granger causality, I assessed each direction using out-of-sample forecasting, an approach better suited to non-linear and non-stationary time series. Rather than statistical inference in the Granger sense, the emphasis was on whether adding exogenous variables to an AR-only model produced meaningful improvements in predictive performance. The underlying rationale was that such improvements indicate a predictive association between the exogenous variable and the response.

In examining the temporal dynamics of tweet volume, I focused on two directions of influence. First, I assessed whether historic tweet volume and policy stringency were associated with epidemiological variables. Second, I explored the reverse relationship, considering whether epidemiological variables or policy stringency were associated with changes in tweet volume. I addressed these questions through a range of model configurations to ensure robust results. As stratifying tweet volume by either sentiment or search term category did not yield meaningfully different correlations in Chapter 6, I focused only on overall tweet volume in this Chapter. Finally, because vaccines were introduced approximately six months into the analysis period, vaccination uptake was omitted.

With this context in mind, I addressed the following research questions in this Chapter:

1. To what extent could COVID-19 epidemiological variables (cases, hospitalisations, and deaths) be predicted using historical tweet volume and policy stringency?
2. To what extent could tweet volume be predicted using historical epidemiological variables and policy stringency?
3. How consistent were any predictive associations across models and countries?

In addressing these questions with time series modelling, I built on the results of Chapter 6 and contributed further by examining more closely the epidemiological utility of tweet volume. In addition, I employed newer DL architectures that, to my

knowledge, had not previously been applied in this context, thereby introducing them into this area of research.

7.3 Related work

7.3.1 The context of time series modelling in COVID-19

Time series models are particularly well suited to infectious diseases with strong seasonal and environmental drivers, such as arboviruses (e.g., dengue, Zika) and endemic seasonal respiratory viruses (e.g. influenza), having shown strong performance in these areas (Lim et al., 2023; Leung et al., 2023). In such settings, periodic incidence patterns and consistent surveillance data (e.g., cases, hospitalisations) allow these models to learn directly from historic trends without requiring detailed mechanistic parameters, information which is often unavailable or highly uncertain.

By contrast, the early stages of the COVID-19 pandemic lacked stable seasonal periodicity. Factors such as rapid variant emergence, changing public health interventions, shifting population behaviour, and vaccination rollout fundamentally altered transmission dynamics over short timescales. These drivers produced highly non-stationary time series that differed markedly from the regular cycles seen in seasonal diseases, as is common with outbreaks. While this remains a potential constraint for time series modelling, advances in the models discussed in Chapter 3 suggest that modern approaches may be increasingly capable of capturing complex, shifting temporal patterns directly from observed data

Most applications of time series models for forecasting during the COVID-19 pandemic relied on publicly available data sources. These data were essential not only for supporting public health modelling but also for engaging the broader research community. Building such datasets in real time required rapid consensus on case definitions, harmonisation of disparate reporting systems, and the establishment of reliable, regularly updated data streams. Institutions such as Johns Hopkins, OWID, the WHO, and the ECDC played a central role by maintaining open-access surveillance platforms (Dong, Du, and Gardner, 2020; Mathieu et al., 2020c; Allan et al., 2022; Organization, 2025; Disease Prevention and Control, 2025).

Despite the availability of these open data resources, COVID-19 modelling still faced

substantial data-related challenges. Surveillance systems had to contend with underreporting (e.g. due to asymptomatic infections, limited testing, and unclassified deaths), delayed reporting, and abrupt changes in testing and case definitions, all of which introduced discontinuities into data. These limitations further motivated the investigation of exogenous variables, which, as evidenced in the subsequent reviewed studies, were drawn from diverse sources, including surveys, mobility data, and internet search queries.

Alongside open data, benchmark tasks further facilitated time series modelling by providing a common framework for model evaluation. These tasks are standardised exercises that enable fair model comparison by fixing data, targets, evaluation metrics, and, in time series settings, forecast horizons, sometimes across multiple lead times for the same outcome. During the pandemic, the Centers for Disease Prevention and Control (CDC) and ECDC Forecast Hubs implemented COVID-19 benchmark tasks that provided the submission protocols and infrastructure needed to integrate contributions from governmental and independent teams (European Centre for Disease Prevention and Control (ECDC), 2021; Cramer et al., 2022).

Despite these regional initiatives, the absence of a global benchmark has limited comparability across studies. This absence likely reflects the difficulty of standardising datasets, case definitions, and reporting practices worldwide. In research conducted without benchmark datasets, inconsistencies in forecasting horizons, analysis periods, and the omission of naïve baselines have hindered the systematic assessment of forecasting approaches.

7.3.2 Applications of time series modelling in COVID-19

Most forecasting studies focused on maximising predictive accuracy for COVID-19 variables. In this Section, I review studies that incorporated exogenous variables, spanning both benchmark and non-benchmark settings. All benchmark-based studies included here were validated against CDC Forecast Hub data. In contrast, studies using ECDC Forecast Hub benchmarks focused on mechanistic and classical models and did not provide relevant ML/DL applications, hence their exclusion. Although primarily aimed at predictive accuracy, the studies reviewed provided an opportunity to evaluate the role of exogenous variables in COVID-19 time series modelling.

Considering the CDC COVID-19 Forecast Hub context, a notable study was conducted by Wang et al. (2022b), who employed the lasso with Google Trends data, vaccination rates, and historical COVID-19 cases and admissions to forecast two-week-ahead new hospital admissions at USA national and state levels. Building on earlier influenza modelling by Yang, Santillana, and Kou (2015), which addressed limitations of the Google Flu Trends system, they incorporated 11 search terms (e.g., “loss of smell”) optimally lagged to align with reported outcomes. They evaluated a lag range of 4 to 35 days and found that for 10 of the 11 terms the optimal lag fell between 4 and 9 days, with the exception of “sinus”, which peaked at 21 days. Benchmarking against CDC Forecast Hub submissions, their approach achieved superior performance in 2021, particularly for state-level forecasts.

With a focus on DL, Du et al. (2023) proposed an LSTM model to forecast weekly COVID-19 cases and deaths for each USA state over horizons of 1–4 weeks. Their model integrated epidemiological, mobility, survey, climate, demographic, and variant frequency data, notably using the Facebook COVID-19 symptom survey (Salomon et al., 2021). Using a recursive forecasting approach without future exogenous inputs, they benchmarked their method against the CDC COVID-19 Forecast Hub ensemble. They found consistent improvements, particularly at longer horizons, and highlighted the time-varying importance of predictors.

Using a newer architecture, Tulchinsky et al. (2025) augmented the N-BEATS architecture with a TCN to encode exogenous covariates and added a variance-forecasting component for probabilistic predictions. In the CDC COVID-19 Forecast Hub’s national-level, 4-week-ahead hospitalisation benchmark, their model reduced MAE by 34.0% and 37.0% compared with performance-weighted and unweighted Hub ensembles, respectively, with ablations confirming the critical role of exogenous predictors (Cramer et al., 2022).

Beyond benchmark datasets, other studies pursued independent forecasting tasks. Li et al. (2023) forecasted eight-day cumulative USA COVID-19 cases using a range of models, including a TCN, MLP, LSTM, CNN, and an ARIMA baseline. They also explored an ensemble approach combining TCN predictions with those from models using exogenous variables. All DL models outperformed ARIMA, with the TCN achieving the best performance and outperforming the ensemble. However, they noted that DL methods required substantial amounts of data, posing a limitation in

data-scarce contexts.

Similarly, Ribeiro et al. (2020) compared four ML approaches, specifically Cubist regression, ridge regression, SVR, alongside an ARIMA model, and developed an ensemble using a Gaussian Process meta-learner. They considered five contiguous lagged variables, selected via a grid search, but did not specify the search range for lags. Forecasting cumulative confirmed COVID-19 cases one, three, and six days ahead in ten Brazilian states with high daily incidence, they found symmetric mean absolute percentage errors (sMAPE) ranging from 0.87% to 6.90% across horizons, with SVR performing best overall.

Zbezhkhovska and Chumachenko (2025) applied four models (LSTM, TFT, XGBoost, and LightGBM) to weekly COVID-19 case data from Ukraine, Bulgaria, Slovenia, and Greece, using 12- and 24-week forecast horizons. They highlighted the noisy nature of outbreak data, consistent with the irregular temporal dynamics of outbreaks discussed earlier. Smoothing methods are often used to address this issue (Babanejaddehaki, An, and Papagelis, 2025). However, the authors noted that the interaction between smoothing and model choice is rarely examined. To address this, they compared four smoothing approaches: a seven-day symmetric MA, an exponentially weighted moving average (EWMA), a Kalman filter, and seasonal–trend decomposition using Loess (STL). For their boosting models, they used fixed lags of 1, 2, 7, and 14 days, while for their DL models they used a 48-week input sequence length. They considered forecast horizons of 12 and 24 weeks, which are much longer than in most comparable studies. Using MAPE for evaluation, they found that smoothing generally improved stability, particularly for DL models. STL was the most effective at stabilising errors across models. LSTM and TFT, when combined with STL or the rolling mean, performed best on short-term forecasts; in contrast, XGBoost and LightGBM were more robust at longer horizons. However, no single model–smoothing combination consistently dominated across countries, underscoring substantial cross-country heterogeneity. A noteworthy limitation of this work was the use of a seven-day symmetric MA, which introduced look-ahead bias and may have influenced their results.

An example from the literature that aligned more closely with my emphasis on exogenous utility was the study by Lamsal, Harwood, and Read (2022). Using a fine-tuned

BERT-based model, they estimated sentiment polarity from Twitter data in the Australian context, extracting 18 topics along with the sentiment polarity associated with each. Granger causality was applied as a feature selection mechanism to identify variables with predictive value, considering lags of up to 14 days for each candidate variable. Using these models they found that an ARIMAX (ARIMA with exogenous variables) model which included the social media data fit to Australian case data better than an ARIMA model, where fit was determined using the Akaike Information Criterion (AIC). They additionally found that including these social media variables improved forecasting, but a major limitation was their reliance on only a nine-day test set. This undermines their claim that the derived sentiment–topic features were reliable predictors of Australian COVID-19 cases.

Overall, prior work indicated that, despite the complex and non-stationary dynamics of COVID-19, time series models have performed well in a predictive context and remained a viable approach for evaluating the utility of exogenous variables in this Chapter. The literature highlighted the heterogeneity within the field; there was no standardised approach to benchmarking, forecast horizons, or a dominant model that consistently performed best. Model construction itself, including lag and exogenous variable specification, was also heterogeneous. In addition, many modern forecasting architectures are yet to be applied in public-health settings, which, in part, motivated my choice of models for comparison.

7.4 Methods

In this Chapter, I used tweet volume to capture attention to COVID-19 topics on Twitter, epidemiological variables (cases, hospitalisations, and deaths), and the OxGRT Government Response index to measure policy stringency, across the same eight countries as in previous chapters. This dataset is detailed in Chapter 6. Note that throughout this Chapter, for brevity, I use OxGRT to refer specifically to the OxGRT Government Response index.

Based on my research questions, I considered the variable combinations presented in Table 7.1. Experiments with epidemiological variables as the response addressed research question one, while those with tweet volume as the response addressed research question two. When referring to epidemiological variables, the analysis was repeated separately for cases, hospitalisations, and deaths.

Response	Exogenous variable
Epidemiological variables	None Tweet volume OxGRT Tweet volume and OxGRT
Tweet volume	None Epidemiological variables OxGRT Epidemiological variables and OxGRT

TABLE 7.1: Variable combinations used across models. Epidemiological variables refer to cases, hospitalisations, and deaths, with each outcome assessed separately.

I presented the time series modelling problem and models I applied to the aforementioned variable combinations in Chapter 3. Specifically, within the ML class, I considered the lasso, random forest, CatBoost, and SVR. Within the DL class, I considered the DilatedRNN, LSTM, TCN, N-BEATSx, TiDE, TFT, and KAN. For clarity in figures and tables throughout this Chapter, I denote the random forest model as RF; all other models are referred to as previously defined. For a naïve baseline, I used a drift method (Hyndman and Athanasopoulos, 2018).

The drift method assumes that, for a given time series with T values, the forecast is equal to the last observed value (y_T) plus the average of the historical changes between consecutive equally spaced periods. To account for the length of the forecast horizon (h), this change is multiplied by h . The forecast is identical across all time points within a given horizon, as defined in Equation 7.1. This reduces more simply to Equation 7.2, which shows the change as the slope of the secant line between the first observed value (y_1) and the current value (y_T).

$$\hat{y}_{T+h|T} = y_T + h \cdot \frac{1}{T-1} \sum_{t=2}^T (y_t - y_{t-1}) \quad (7.1)$$

$$= y_T + h \cdot \frac{y_T - y_1}{T-1} \quad (7.2)$$

Importantly, as mentioned previously, for ML models the lag structure needed to be explicitly specified. To avoid overspecification and maintain interpretability, either two or four lags per feature were used and these were allowed to differ across variables. I treated the lags as a hyperparameter and tuned them during training.

Setup	Lag	Options
Two lags	Lag 1 (l_1)	t-1, t-2, t-3, t-4, t-5, t-6, t-7
	Lag 2 (l_2)	t-8, t-9, t-10, t-11, t-12, t-13, t-14
Four lags	Lag 1 (l_1)	t-1, t-2, t-3, t-4
	Lag 2 (l_2)	t-5, t-6, t-7, t-8
	Lag 3 (l_3)	t-9, t-10, t-11
	Lag 4 (l_4)	t-12, t-13, t-14

TABLE 7.2: The different time step options for each lag variable included in the ML model.

A more flexible approach would have been to specify an overall input sequence length and treat both the number of lags and their positions as hyperparameters. This, however, would have required dynamic hyperparameter spaces, as each lag selection would have depended on those already chosen. As an illustrative example, consider a search space that includes lags one to seven. If the first lag selected is one, the valid options for the second lag would be two to seven. If, instead, three is chosen first, the valid subspace would be different and include all lags except the third. Because I used a TPE sampler (described in Chapter 3), such dynamic dependencies could not be represented, since its EI approximation required a fixed hyperparameter space. While alternative strategies, such as random sampling, would have allowed for this flexibility, I retained the TPE sampler to ensure consistency across experiments. To accommodate the restriction imposed by the TPE sampler, I partitioned the 14-day lag window into equal segments (with earlier segments absorbing the modulus, if any). A single lag was chosen for each lag parameter from its assigned segment. This design was analogous to the dilations used in DilatedRNN and TCN architectures. The lag options used for each configuration are shown in Table 7.2.

Based on these options, and assuming two exogenous variables in addition to the AR terms, the functional form of a single-step forecast can be expressed as follows:

Two lag setup: $\hat{f}_1(y_{l_1}, y_{l_2}, x_{l_1}^1, x_{l_2}^1, x_{l_1}^2, x_{l_2}^2)$

Four lag setup: $\hat{f}_1(y_{l_1}, y_{l_2}, y_{l_3}, y_{l_4}, x_{l_1}^1, x_{l_2}^1, x_{l_3}^1, x_{l_4}^1, x_{l_1}^2, x_{l_2}^2, x_{l_3}^2, x_{l_4}^2)$

where each term represents a lagged variable in the regression, with the possible lags specified in Table 7.2.

I implemented my experiments in Python (v3.10.11) using the Nixtla framework, which provides packages for time series modelling. The drift method was implemented using StatsForecast (v2.0.0) (Garza et al., 2022). For ML models, I used MLForecast

(see fork details below for version), which uses model implementations from scikit-learn (v1.2.2) and the official CatBoost library (v1.2.7) (Pedregosa et al., 2011; Nixtla, 2025; Prokhorenkova et al., 2018). For DL models, I used NeuralForecast (v3.0.2, including commit c174c92, which enables in-sample prediction of static variables) (Olivares et al., 2022). To automate the generation of ML lag features, I extended MLForecast (v1.0.2). This extension is available in my GitHub fork of Nixtla to ensure reproducibility of the experiments presented here; my aim is to add it to the official package: <https://github.com/tristan-myles/mlforecast>.

7.4.1 Evaluation

To ensure comparability across experiments, I adopted the same training, validation, and test periods for each model. I excluded the Omicron period between December 2021 and January 2022 due to its markedly higher case counts compared with earlier variants. The exact periods are given in Table 7.3. Across all periods and experiments a forecast horizon of 7 days was used.

Type	Duration (days)	Period
Training	243	[2020-07-01; 2021-03-01)
Validation	153	[2021-03-01; 2021-08-01)
Test	122	[2021-08-01; 2021-11-30]

TABLE 7.3: The training, validation, and test periods.

To determine optimal hyperparameters, I performed 30 trials using the training set, with candidate configurations suggested by the TPE sampler. The exact hyperparameter spaces are given in Appendix E.1. I adopted an expanding window evaluation scheme with a step size equal to the forecast horizon (7 days). All experiments were conducted globally, meaning that models were fit jointly across all countries. To account for country-level differences, I included a static categorical variable indicating country membership as a hyperparameter; this allowed for country-specific adjustments where beneficial. After identifying the optimal hyperparameter configuration, I retrained the model using the combined training and validation period (10 July 2020 to 1 August 2021) and then performed the final evaluation using the test set.

To further address cross-country differences, I scaled the data. For DL models, I used the RevIN method described in Section 3.8.1. For ML models, I standardised the data to zero mean and unit variance by subtracting the mean and dividing by the standard deviation of the corresponding training data. During validation, standardisation used

statistics from the training set, while during testing the statistics were recalculated using the combined training and validation set. It is worth highlighting that, although evaluation sets were excluded from the calculation of scaling statistics to prevent data leakage, they were nonetheless scaled using those statistics.

Beyond hyperparameter tuning, I assessed key experimental design choices, namely, lag structure, refitting frequency, log transformation, and smoothing. I tested three refitting intervals: one, four, and eight forecast horizon windows. In this setup, less frequent refitting acts as a form of regularisation. For smoothing, I initially applied a seven-day MA, consistent with the correlation analysis in Chapter 6. However, because this approach performed poorly in preliminary experiments (as shown in Section 7.5), I also tested four alternatives: no smoothing (raw), a linear spline (D1-spline, where D denotes degree), a quadratic spline (D2-spline), and a cubic cyclic spline (C-spline). These smoothing approaches are described in more detail in Section 7.4.2. My experimental design choices are summarised in Table 7.4.

Choice	Options
Smoothing	Raw, seven-day MA, D1-spline, D2-spline, C-spline
Refit frequency	1, 4, 8
Transformations	log
ML Lags	2, 4

TABLE 7.4: Experiment design choices.

To reduce the number of experiments, I adopted a greedy procedure for evaluating design choices. Specifically, I began by selecting the most appropriate smoothing method with a refit frequency of eight, since this was the least computationally demanding interval I considered. Next, with smoothing fixed, I evaluated different refitting frequencies. Finally, with these two decisions set, I assessed the impact of log transformation across both lag options for the ML models. To isolate the effect of these design choices, I conducted the experiments using AR terms, introducing exogenous variables only once the optimal design had been determined.

7.4.2 Smoothing

The additional smoothing approaches were chosen to produce responses smoother than a seven-day MA. Despite the seven-day MA, I found that the series still exhibited excessive volatility, which was especially challenging given my short training windows and the atypical dynamics of these time series. My decision to explore stronger

smoothing was motivated by Zbezhkivska and Chumachenko (2025), whose study, introduced in Section 7.3, demonstrated the value of smoothers beyond the moving average. Moreover, given the previously noted caveat of look-ahead bias, it is plausible that the true advantage of the alternative smoothers was diluted and that their performance may have been even stronger relative to the moving average. Finally, to assess the full impact of smoothing, models were also fit to the raw, unsmoothed series.

I adopted three spline-based smoothing approaches, previously introduced, each of which imposes continuity constraints to produce smoother curves than direct interpolation. Specifically, the D1- and D2-splines enforce continuity at knot points, while the C-spline imposes additional constraints by requiring continuity of the first and second derivatives and enforcing periodicity at the series endpoints. This additional structure makes the cyclic spline particularly well suited to capturing seasonal effects. I applied these smoothing methods across all variables, except for hospitalisations in countries where values had already been interpolated using a linear spline (as detailed in Chapter 6); in those cases, I did not apply any additional smoothing.

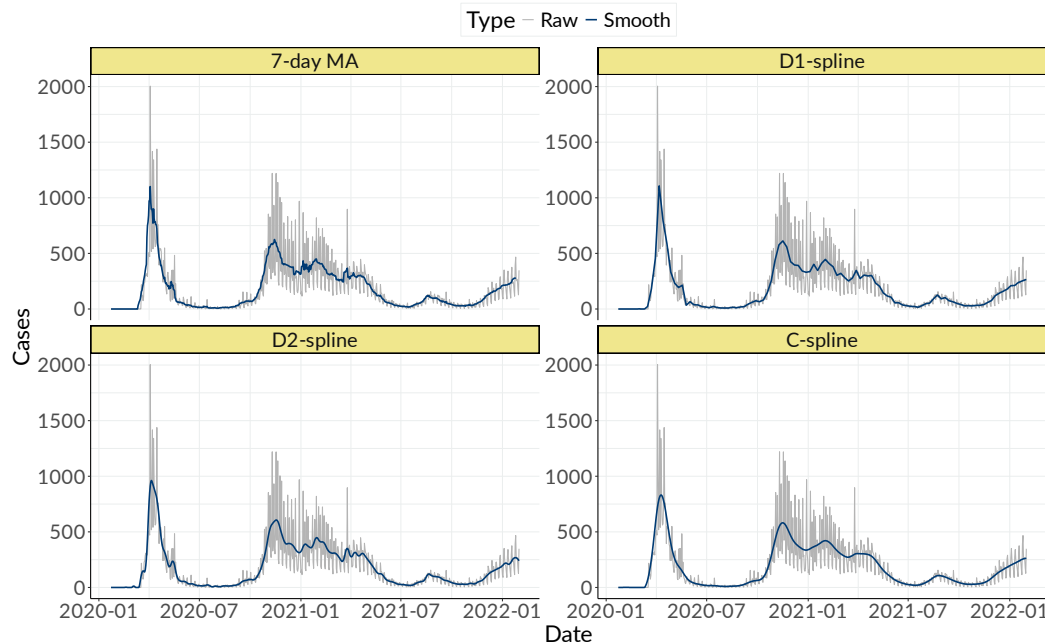


FIGURE 7.1: Different smoothing options shown for COVID-19 deaths in France as an illustrative example. The Raw time series is shown in gray, with the smooth curve overlaid.

The effects of these smoothing methods are illustrated in Figure 7.1, which shows

smoothed series for COVID-19 deaths in France. I selected this outcome for illustration because it exhibits a particularly pronounced day-of-week effect. This Figure shows that all spline approaches result in a smoother time series compared to the seven-day MA. The D1-spline and D2-spline yield similar results, though the D1-spline retains sharper peaks due to its linear structure. The C-spline is the smoothest, but its additional constraints dampen peak magnitudes relative to the other approaches.

7.5 Results

I begin the results by establishing the optimal experimental design using AR-only models (i.e., the model with AR terms only), before presenting the results of analysing the utility of exogenous variables using this design. All results presented correspond to optimal models that were hyperparameter-tuned across 30 trials, as described in Section 7.4.1. Due to the large number of models evaluated, optimal hyperparameter configurations are provided only for the models presented in Section 7.5.2 and are reported in Appendix E.2. Given their central role in determining how much historical information each model incorporated, particular emphasis was placed on lag-related hyperparameters, which are the focus of Section 7.5.3.

7.5.1 Experiment design choices

Smoothing

The first element of experiment design concerned the choice of smoothing approach. Table 7.5 reports the median and IQR of MAPE values on the test set for each model class and outcome, with the baseline result (drift method) shown in gray. The median and IQR are presented rather than the mean because model performance varied substantially within classes, with tree-based ML methods showing particularly large deviations for lower levels of smoothing. The lowest values within each model class are highlighted in bold for each outcome, and the lowest overall value per outcome is highlighted in green. Where median values are tied, the result with the lower IQR is favoured. All subsequent tables follow the same conventions.

The C-spline was most often the best-performing smoothing approach, and was particularly advantageous for DL models (Table 7.5). In general, the median MAPE

Outcome	Model class	Raw	Seven-Day MA	D1-spline	D2-spline	C-spline
Cases	Baseline	32 (29)	12 (3)	13 (4)	13 (4)	13 (4)
	ML	70 (97)	12 (4)	11 (5)	16 (57)	8 (5)
	DL	26 (15)	11 (7)	12 (8)	7 (4)	3 (3)
Hospitalisations	Baseline	12 (3)	12 (4)	12 (4)	12 (4)	12 (4)
	ML	11 (4)	10 (4)	10 (6)	11 (7)	11 (7)
	DL	11 (5)	8 (5)	10 (8)	9 (8)	8 (7)
Deaths	Baseline	54 (24)	10 (6)	14 (4)	14 (4)	12 (6)
	ML	62 (38)	17 (9)	24 (14)	22 (10)	15 (8)
	DL	41 (38)	18 (7)	14 (7)	14 (8)	8 (8)
Tweets	Baseline	28 (5)	14 (4)	13 (3)	16 (2)	13 (2)
	ML	71 (30)	46 (31)	35 (31)	21 (4)	18 (6)
	DL	28 (7)	17 (4)	18 (5)	12 (6)	12 (8)

TABLE 7.5: MAPE (%) from applying each model class to smooth variables. Values show the median with the IQR in parentheses across models and countries. The best result per model class and outcome is shown in **bold**; the best overall model class-smoothing combination per outcome is shown in **green**.

decreased with increased smoothing for DL models. The exception was hospitalisations, which formed a special case because only Poland, Portugal, and the UK varied across smoothing variables. Although the C-spline was most often optimal, the D2-spline achieved the same median performance with a lower IQR for tweet volume. For ML models, this pattern did not hold, as the seven-day MA often outperformed one or both of the D1- and D2-splines for epidemiological variables. For the raw outcome, ML models performed poorly, with hospitalisations again the exception, since this outcome was already relatively smooth, as noted previously. This performance may reflect the sparse lag specification, which provided insufficient information for the models to capture day-of-the-week effects. In contrast, DL models were still able to fit the data and likely learned day-of-the-week effects in much the same way as models learn seasonality. They achieved lower, or at least comparable, values to the baseline across all raw variables.

Across all variables, the IQR was high relative to the median, indicating variability in how models responded to smoothing. Although it is plausible that certain models may have benefitted more from specific smoothing approaches, this cannot be established with certainty without more granular investigation. Moreover, adopting different smoothing schemes for individual models would have increased complexity and, more importantly, reduced comparability across models. Consequently, to maintain consistency and meaningful comparison, a single general scheme was applied, with the C-spline used across all variables due to its superior average performance.

Refitting frequency

Table 7.6 reports the results of applying different refitting frequencies using C-spline-smoothed variables. A refitting frequency of one had the strongest effects for ML models fitted to hospitalisations and for both model classes applied to tweet volume. For the remaining variables, differences across frequencies were minimal for both classes. Overall, a refitting frequency of one yielded the best performance across both model classes and was adopted.

Outcome	Model class	Refit frequency		
		1 week	4 weeks	8 weeks
Cases	Baseline	13 (4)		
	ML	9 (7)	8 (5)	8 (5)
	DL	2 (2)	3 (1)	3 (3)
Hospitalisations	Baseline	12 (4)		
	ML	8 (6)	10 (7)	11 (7)
	DL	7 (6)	7 (8)	8 (7)
Deaths	Baseline	12 (6)		
	ML	15 (8)	15 (8)	15 (8)
	DL	8 (7)	8 (7)	8 (8)
Tweets	Baseline	13 (2)		
	ML	16 (5)	18 (5)	18 (6)
	DL	7 (4)	10 (7)	12 (8)

TABLE 7.6: MAPE (%) from applying each model class to C-spline smoothed variables with different refitting frequencies. Values show the median with the IQR in parentheses across models and countries. The best refit period per model class and outcome is shown in **bold**; the best overall model class-refitting frequency combination per outcome is shown in **green**.

Model structure

The final design component concerned the number of lags to include in the ML models, and whether to log the outcome. Table 7.7 presents the results of these choices applied to C-spline-smoothed variables with a refitting frequency of one. For DL models, logging had little impact on epidemiological variables, although for tweet volume, the unlogged outcome performed best. In contrast, for ML models, logging improved performance for epidemiological variables, while for tweet volume, the logged outcome performed considerably worse. In most cases, four lags outperformed two. The only exception was hospitalisations, where the median MAPE with two lags was slightly lower but accompanied by a higher IQR. On this basis, I adopted unlogged variables

for DL models, while for ML models I used four lags and logged epidemiological variables, retaining the unlogged outcome only for tweet volume.

Outcome	Model class	Lags	C-spline R1	Log(C-spline R1)
Cases	Baseline	-	13 (4)	
	ML	2	11 (6)	10 (8)
	ML	4	9 (7)	8 (6)
	DL	-	2 (2)	2 (2)
Hospitalisations	Baseline	-	12 (4)	
	ML	2	13 (15)	7 (8)
	ML	4	8 (6)	8 (5)
	DL	-	7 (6)	8 (6)
Deaths	Baseline	-	12 (6)	
	ML	2	16 (9)	11 (7)
	ML	4	15 (8)	10 (4)
	DL	-	8 (7)	8 (5)
Tweets	Baseline	-	13 (2)	
	ML	2	18 (7)	53 (51)
	ML	4	16 (5)	52 (44)
	DL	-	7 (4)	11 (6)

TABLE 7.7: MAPE (%) results assessing whether logging an outcome improves performance and whether four or two lags are preferable for an ML model. Values show the median with the IQR in parentheses across models and countries. For each outcome, the best result within each model class is shown in **bold**, and the best overall model class-smoothing combination in **green**.

7.5.2 Assessing the utility of exogenous variables

All exogenous variables

Building on the preceding results, the optimal experimental design was determined as follows: a C-spline with a refitting frequency of one, unlogged variables for DL models, and four lags with logged variables for ML models (with tweet volume remaining unlogged). This design was applied to results presented in this Section.

Table 7.8a presents the results of applying different combinations of exogenous variables to epidemiological variables. Overall, DL models achieved the best performance. None of the exogenous combinations reduced the median MAPE compared with the AR-only model. However, the IQR was narrower for cases when OxGRT was included, and for hospitalisations when either OxGRT alone or in combination with tweet volume is included. For deaths, including tweet volume, yielded the same median MAPE but a higher IQR. For ML models, the AR-only specification consistently performed best. Nevertheless, when comparing the exogenous combinations to one

another, tweet volume performed best for cases, performed similarly to OxGRT for deaths, and yield the same median but higher IQRs than the tweet volume–OxGRT combination for hospitalisations.

Outcome	Model class	None	OxGRT	Tweets	Tweets + OxGRT
Cases	Baseline	13 (4)			
	ML	8 (6)	9 (7)	8 (4)	11 (8)
	DL	2 (2)	2 (1)	3 (3)	3 (1)
Hospitalisations	Baseline	12 (4)			
	ML	8 (5)	10 (8)	8 (8)	8 (6)
	DL	7 (6)	7 (5)	8 (5)	7 (4)
Deaths	Baseline	12 (6)			
	ML	10 (4)	12 (6)	12 (6)	14 (10)
	DL	8 (7)	9 (8)	8 (9)	10 (10)

(A) Cases, Hospitalisations, Deaths

Outcome	Model class	None	OxGRT	Cases	Cases + OxGRT	Hosps	Hosps + OxGRT	Deaths	Deaths + OxGRT
Tweets	Baseline	13 (2)							
	ML	16 (5)	18 (7)	18 (9)	23 (10)	18 (7)	20 (10)	17 (4)	27 (52)
	DL	7 (4)	9 (4)	9 (4)	9 (4)	9 (4)	10 (5)	10 (5)	9 (6)

(B) Tweets

TABLE 7.8: MAPE (%) results from model classes across exogenous variable combinations. Values show the median with the IQR in parentheses across models and countries. For each model class and outcome, the lowest median and IQR combination is shown in **bold**, and the best overall per outcome in **green**.

Consistent with the results for epidemiological variables, Table 7.8b shows that tweet volume also did not benefit from the inclusion of exogenous variables. Overall, DL models performed approximately twice as accurately as ML models and improved on the baseline, whereas ML models did not. Across all settings, the model without exogenous variables performed best. Among the exogenous variables considered, performance was similar across combinations for DL models, with the strongest results observed when OxGRT, cases, OxGRT and cases, and hospitalisations were used. For ML models, including only deaths yielded the largest improvement, while OxGRT, cases, and hospitalisations performed comparably but with higher IQRs.

Figure 7.2 presents an expanded view of the MAPE scores shown in Table 7.8 across all countries. The Figure again highlights the superior performance of DL models. Although exogenous variables did not improve performance for this class of models, they did not substantially degrade it either. By contrast, differences in performance between exogenous and AR-only models were more pronounced for ML models. These

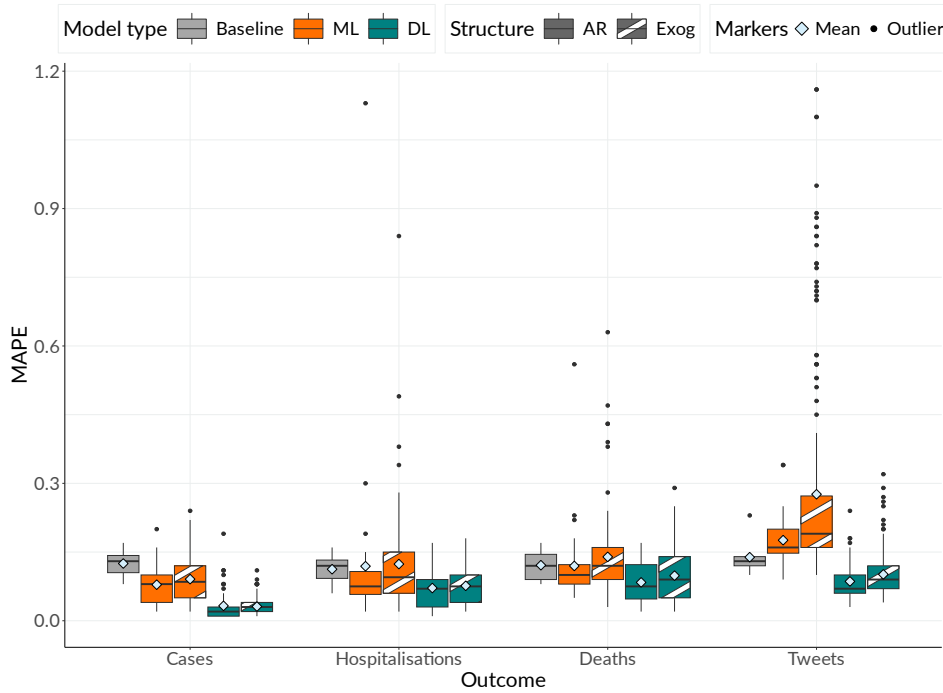


FIGURE 7.2: Boxplots of grouped model performance across all countries presented per outcome and stratified by model type and structure. The baseline refers to the baseline model (drift method). The AR structure refers to models that contain only AR terms, while the Exog structure includes all models that incorporate exogenous variables. Outliers are determined using Tukey's criterion ($< Q1 - 1.5 \times IQR$ or $> Q3 + 1.5 \times IQR$).

models had similar medians but larger IQRs, as well as a greater number of outliers compared to both AR-only models and to DL models. In particular, when exogenous variables were included with the tweet response, the number of outliers increased substantially for both ML and DL models.

Overall, the Figure highlights the heterogeneity in performance, both in the AR-only context and when exogenous variables were incorporated. This suggests that specific models or countries may have benefitted from exogenous information. As discussed in Chapters 3 and 6, both model structures and country responses vary in their relationship to exogenous variables. To explore this further, model- and country-specific results are presented next.

Figure 7.3 presents performance across individual models for all countries and exogenous variable combinations. While DL models tended to perform best overall, achieving the lowest median and IQR MAPE scores, no single model consistently dominated. Within the ML group, poor performance was primarily driven by the random forest and CatBoost, both of which exhibited wide ranges of MAPE scores.

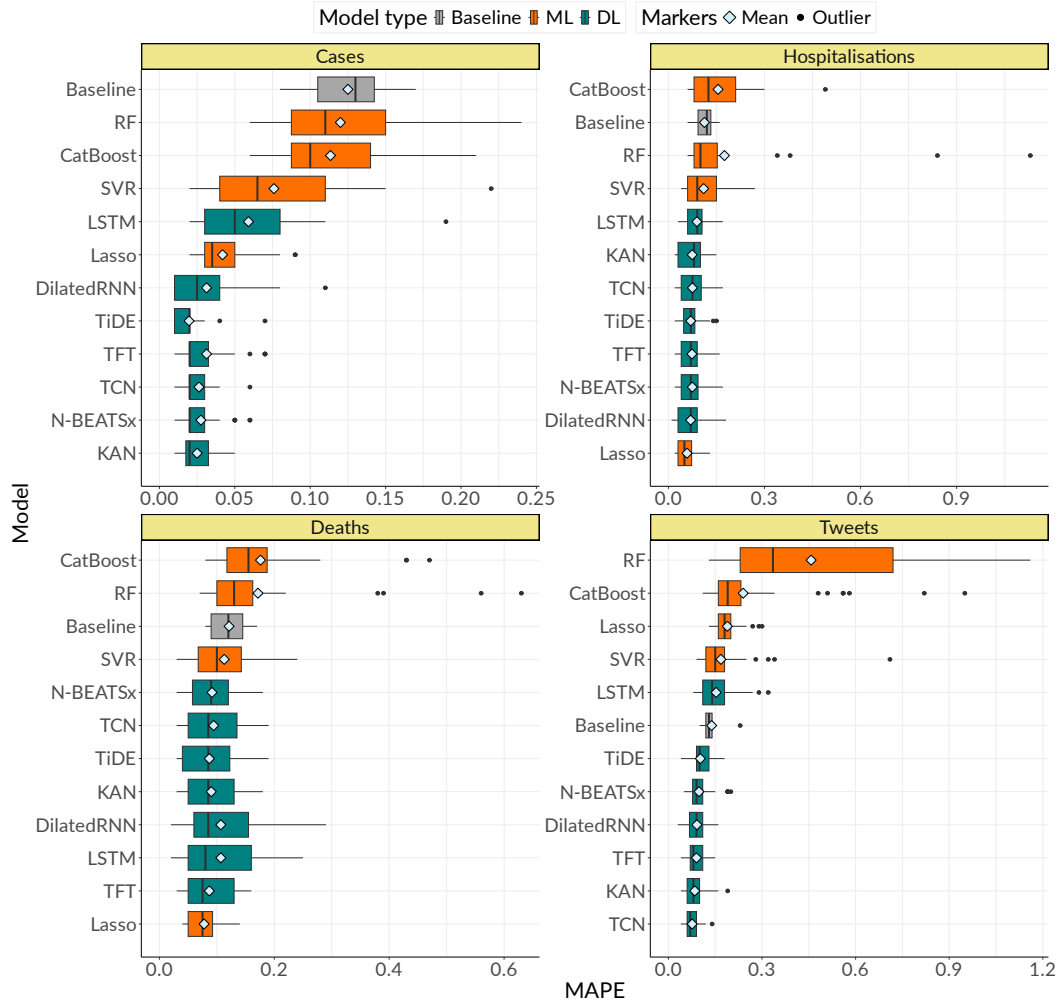


FIGURE 7.3: Boxplots of individual model performance across all countries presented per outcome and stratified by model type. Outliers are determined using Tukey's criterion ($< Q1 - 1.5 \times IQR$ or $> Q3 + 1.5 \times IQR$).

SVR showed mixed results, whereas the lasso performed strongly, surpassing the DL models for deaths and hospitalisations and outperforming the LSTM for cases. For tweet volume, however, all ML models performed worse than the baseline, while DL models generally improved upon it. Additionally, most of the outliers observed in Figure 7.2 stemmed from the random forest and CatBoost among the ML models, and from the LSTM among the DL models, as evident in Figure 7.3.

Across countries, similar heterogeneity across variables was observed, as shown in Figure 7.4 which presents the difference in MAPE scores compared with the baseline for each country. Because each country had its own baseline, a single baseline boxplot, as used in previous figures, is not meaningful in this context. Numerous outliers above the upper quartile indicate poor performance for specific exogenous variable

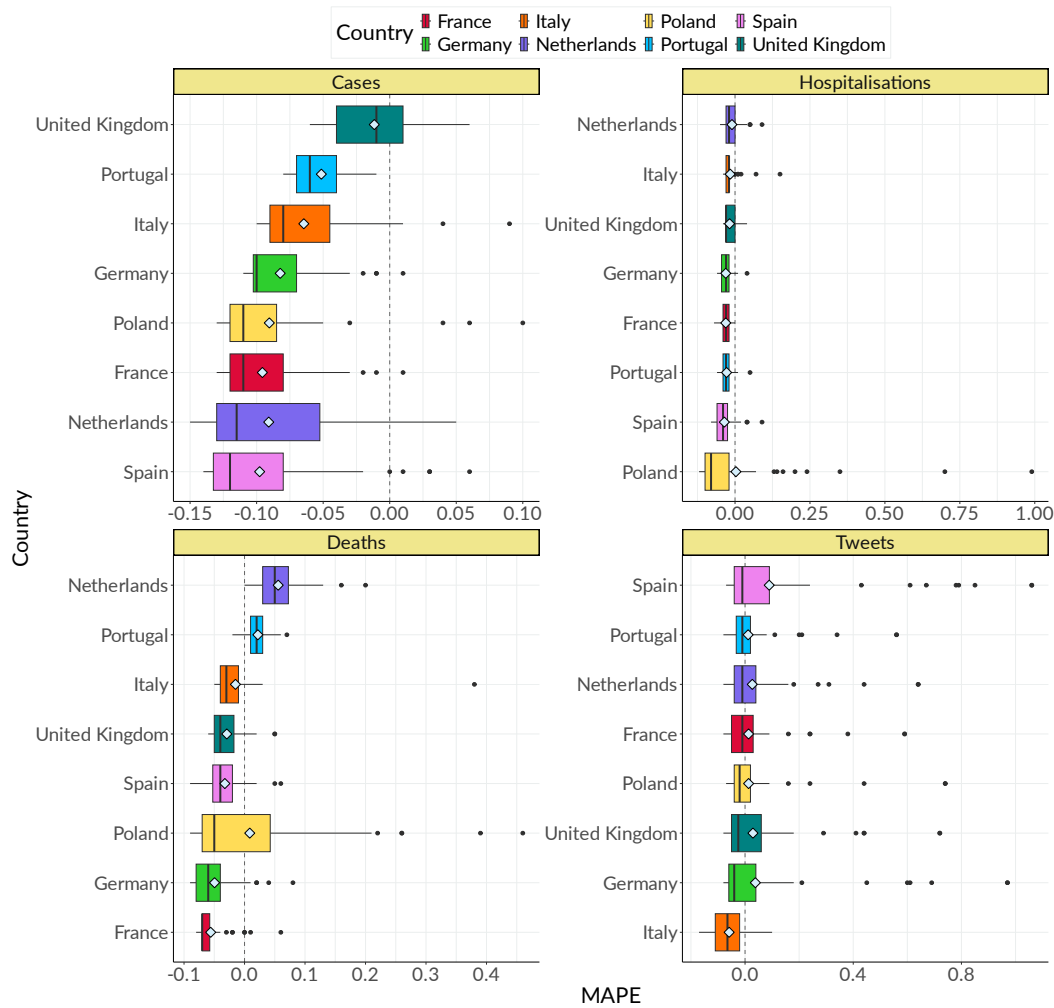


FIGURE 7.4: Boxplots of individual model performance (difference from MAPE baseline) across all models presented per outcome. Outliers are determined using Tukey's criterion ($< Q1 - 1.5 \times IQR$ or $> Q3 + 1.5 \times IQR$).

and model combinations, which were again primarily driven by tree-based ML models.

Performance differed substantially across countries. In the Netherlands, models performed poorly overall, with the highest medians for hospitalisations and deaths, and the third-highest median for tweet volume, for which they also recorded the second-worst mean. By contrast, performance was comparatively strong for cases, where it ranked second-best. Notably, for deaths in both the Netherlands and Portugal, the distribution of model scores had a median above the baseline, the only epidemiological variables where this occurred. The UK also underperformed, with the highest median for cases, where more than a quarter of scores lay above the baseline, the second-highest for hospitalisations, and mid-range performance for the remaining variables. Poland displayed a comparatively wide range of MAPE scores for deaths and

hospitalisations, achieving the lowest median but highest mean for hospitalisations, and the third-highest mean and largest IQR for deaths.

Outcome-specific patterns were also apparent across countries. Deaths and hospitalisations tended to have narrower IQRs but exhibited more outliers than cases, reflecting inconsistent model behaviour. Tweet volume encountered the greatest number of outliers and were most often worse than the baseline. However, an exception was Italy, where more than three quarters of models outperformed the baseline and no outliers were present. These findings highlighted the heterogeneity of model performance across both variables and countries, motivating the need for more detailed, disaggregated analysis.

The best exogenous fit

The previous results compared all exogenous variable combinations collectively with AR-only models. In the following results, the focus shifts to identifying the best-performing exogenous variables, at two levels of disaggregation. First, at the model level and, second, at the intersection of individual countries and models.

Table 7.9 reports the best-performing exogenous variable combinations at the model level. Specifically, it shows, for each model and outcome, the exogenous combination (or combinations, in the case of ties) that yielded the lowest median score and IQR when applied consistently across all countries. Across all outcomes, the model without exogenous variables performed best in 64%¹ of comparisons. Considering epidemiological variables together, OxGRT was included the least (12%), tweet volume was included the most (18%), followed closely by the combination of tweet volume and OxGRT (15%). No model consistently performed best with a single exogenous combination across outcomes, with exogenous variables improving performance for at most two outcome per model.

In most cases, the best-performing model included no exogenous variables. Exceptions were observed for the lasso, where including tweet volume as an exogenous variable tied with the AR-only model when predicting hospitalisations. For cases, TiDE also performed the same as the AR-only model when including tweet volume, while the TCN performed best when including OxGRT, although many models achieved similar

¹Percentages are reported relative to fixed denominators that are multiples of 11, corresponding to the number of models. For example, since all variables are considered here, the denominator is 44. This avoids inflation from duplicates.

Models	Cases		Hospitalisations		Deaths		Tweets	
	MAPE	Best model	MAPE	Best model	MAPE	Best model	MAPE	Best model
Baseline	13 (4)		12 (4)		12 (6)		13 (2)	
CatBoost	9 (1)	□	8 (3)	+	12 (4)	+	16 (4)	◆
Lasso	2 (2)	■	5 (4)	□, +	7 (4)	+	16 (1)	+
RF	10 (3)	□	9 (6)	+	11 (4)	+	18 (8)	+
SVR	4 (1)	+	6 (5)	+	9 (8)	□	11 (4)	+
DilatedRNN	2 (1)	+	6 (4)	◆	6 (8)	+	5 (1)	+
KAN	2 (1)	+	8 (5)	■	8 (8)	■	6 (3)	+
LSTM	3 (1)	◆	8 (5)	+	8 (11)	+	11 (3)	◆
N-BEATSx	2 (1)	+	6 (4)	+	8 (6)	□	6 (2)	+
TCN	2 (1)	◆	6 (3)	■	8 (6)	+	6 (3)	+
TiDE	2 (1)	□, +	6 (3)	◆	7 (6)	+	6 (3)	+
TFT	2 (1)	+	6 (4)	■	6 (8)	+	8 (2)	▲

TABLE 7.9: MAPE (%) results for the best exogenous variable combination per model. If multiple combinations yield the same median and IQR, all such combinations are shown. Values represent the median with the IQR in parentheses across countries. For each outcome, the lowest median–IQR combination is highlighted in **green**. ML and DL models are displayed above and below the dashed line, respectively. Each symbol in the table denotes the set of exogenous variables used in the model achieving the lowest median-IQR MAPE, with None indicating the AR model. A legend for the symbols is as follows: + None, ◆ OxGRT, □ Tweets, ■ Tweets + OxGRT, ▲ Hospitalisations + OxGRT.

performance for this outcome. At the outcome level, further heterogeneity is evident, with tweet volume most often included for cases (27%), the combination of tweet volume and OxGRT for hospitalisations (27%), and tweet volume alone for deaths when exogenous variables were selected (18%). For tweet volume as an outcome, models generally performed best with the AR-only specification (73%), similar to deaths, although when exogenous variables were preferred, OxGRT performed best (18%).

Figure 7.5 provides a more detailed view by disaggregating performance to the intersection of individual countries and models. Unlike Table 7.9, this presentation does not impose the same exogenous combination across all countries but instead highlights which combinations performed best within each country. It is important to emphasise, however, that models are still fit globally, as outlined in Section 7.4.1. Although performance is displayed by country, each symbol in the Figure represents the same globally fit model, rather than a separate country-specific. Each square in Figure 7.5 shows the model with the lowest difference in MAPE compared to the

baseline across all exogenous variable combinations, including the case with no exogenous variables. In the case of ties, all best-performing combinations are shown, except when the tie includes the model without exogenous variables. In that situation, only the symbol for the AR-only model is displayed. This contrasts with Table 7.9, where all ties are reported. The motivation for this choice was to improve clarity of presentation while drawing attention to instances where exogenous variables provide an advantage over AR-only models.

For cases, most models improved on the baseline. Across models, DL approaches most often performed best without exogenous variables. The LSTM was an exception, where, for all countries except France, the best model included only tweet volume as an exogenous variable. In France, the combination of tweet volume and OxGRT performed best, and this combination was also the joint best for Italy and Poland. Among ML models, all approaches except SVR tended to perform best when exogenous variables were included. The lasso most frequently selected a combination of tweet volume and OxGRT, the random forest typically included OxGRT alone or together with tweet volume, and CatBoost alternated between including OxGRT and tweet volume. The UK stood out as the most challenging country to predict; tree-based ML models performed slightly worse than the baseline, and both ML and DL models more often included exogenous variables. Across all models, exogenous variables were selected in similar proportions: 18%² included OxGRT, 17% included tweet volume, 16% included their combination, and 58% included no exogenous variables.

For hospitalisations, models generally improved on the baseline, although the gains were not as substantial as for cases. Models performed similarly on average across countries for this outcome. However, Poland showed comparatively greater variability, with tree-based models recording the worst performance and DL models the best performance across all countries. Considering model preferences, the TCN and TFT models included exogenous variables most often. In contrast to the other variables and to DL models, ML models seldom included exogenous variables. Selection percentages were similar to those for cases: 12% included OxGRT, 11% included tweet volume, 19% included their combination, and 65% included no exogenous variables.

²Percentages are reported relative to a fixed denominator of 88, which avoids inflation due to ties; percentages reported therefore do not necessarily sum to 100%.

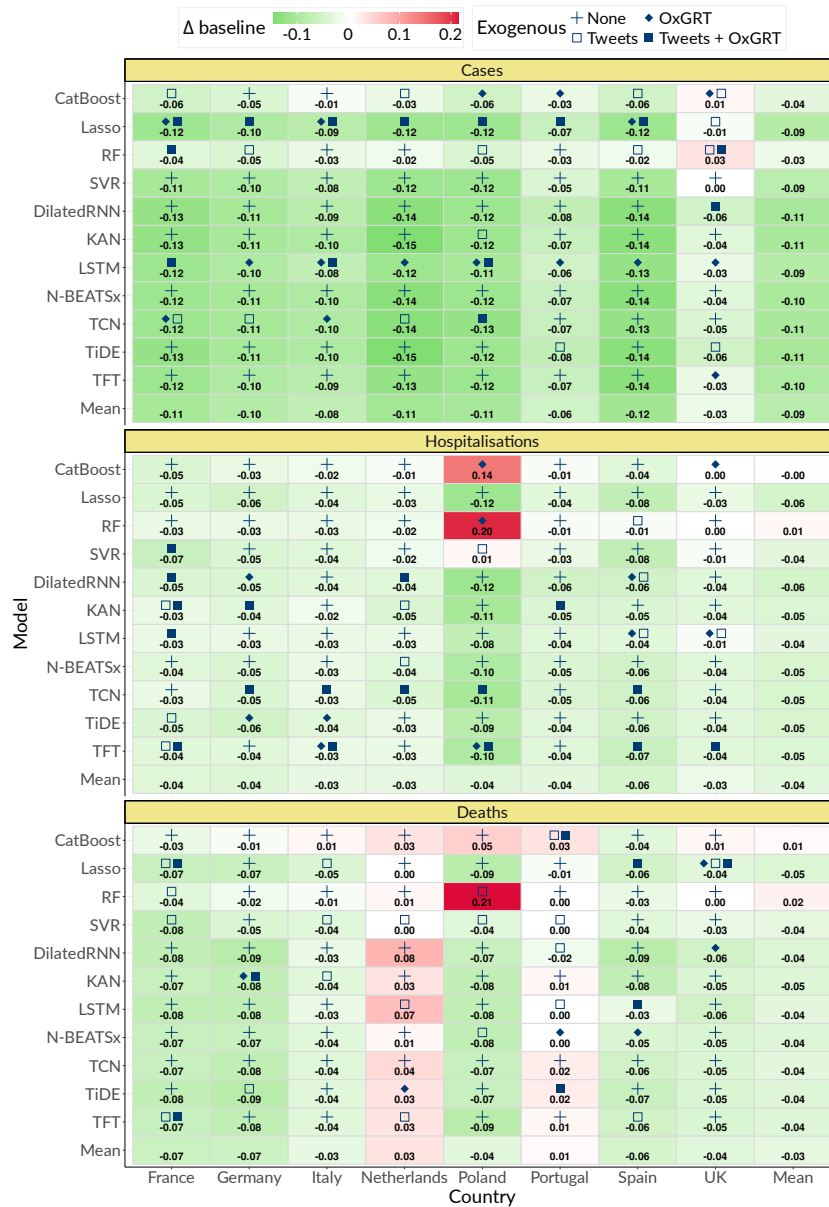


FIGURE 7.5: Heatmap of model performance compared to the baseline for epidemiological variables across models and countries. Each square shows the difference between the baseline model (drift method) and the best-performing exogenous variable combination, indicated by the shape. In the case of ties, all best-performing combinations are shown, except when an AR-only model is included, in which case only None (+) is displayed. Positive values (red) indicate worse performance than the baseline, zero (white) indicates no change, and negative values (green) indicate improvement.

For deaths, improvements over the baseline were less pronounced, with models struggling in particular for the Netherlands and Portugal. Similar to the UK for cases, models most often performed best in these countries when exogenous variables were included. Tree-based methods again performed poorly, especially for Poland. There was no consistent trend in models including specific variables. Notably, SVR, which

included no exogenous variables for cases, incorporated tweet volume for five countries. OxGRT was least included (7%), tweet volume was most frequently included (23%), a combination of the two was included in 9% of models, and 68% included no exogenous variables.

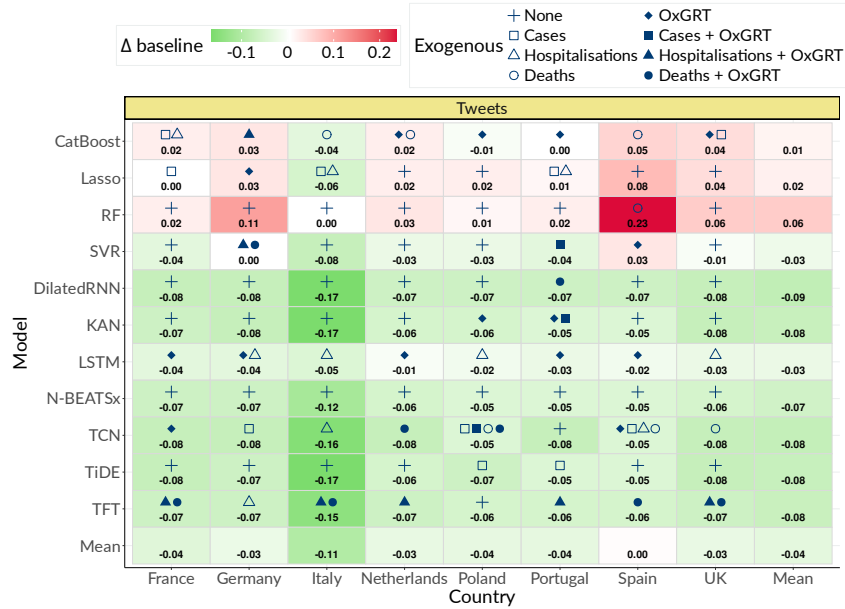


FIGURE 7.6: Heatmap of model performance compared to the baseline for tweet volume across models and countries. Each square shows the smallest difference between the baseline model (drift method) and the best-performing exogenous variable combination, indicated by the shape. In the case of ties, all best-performing combinations are shown, except when an AR-only model is included, in which case only None (+) is displayed. Positive values (red) indicate worse performance than the baseline, zero (white) indicates no change, and negative values (green) indicate improvement.

Figure 7.6 is analogous to Figure 7.5 but with tweet volume as the outcome and with exogenous variable combinations of epidemiological variables and OxGRT. Overall, ML models performed poorly, with the exception of SVR, which improved on the baseline in all countries except Spain and Germany, where performance matched the baseline. By contrast, DL models perform strongly across countries, with the best results observed for Italian tweet volume, where even ML models achieve strong performance. The LSTM and TCN included exogenous variables in every country, whereas the TFT did so in all but Poland. For the LSTM, this pattern is consistent with its behaviour on the case outcome. The distribution of exogenous variables was relatively balanced across model classes. OxGRT was most frequently selected (17%), followed by hospitalisations (11%) and cases (11%). Other variables appear at comparable rates: cases + OxGRT (3%), hospitalisations + OxGRT (8%), deaths

(8%), and deaths + OxGRT (9%). AR-only terms were still included in the majority of models (51% of country–model combinations).

The mean country results in these heatmaps are consistent with, and provide additional context to, Figure 7.4. Despite these heatmaps presenting a smaller subset of only the best-performing models, the same patterns appeared as in Figure 7.4. Consequently it is reasonable to infer that the lower-performing models followed similar trends. To provide a more comprehensive model-level overview, and in line with the results presented earlier in this Chapter, Table 7.10 summarises the heatmap results by reporting the median and IQR of MAPE scores across the best exogenous-combination models for each country and outcome.

Model	Cases	Hospitalisations	Deaths	Tweets
Baseline	13 (4)	12 (4)	12 (6)	13 (2)
CatBoost	9 (1)	8 (3)	12 (7)	16 (1)
Lasso	2 (1)	5 (4)	4 (4)	16 (2)
RF	10 (3)	9 (6)	11 (5)	18 (8)
SVR	4 (1)	6 (3)	8 (7)	11 (3)
DilatedRNN	2 (1)	6 (5)	6 (7)	5 (1)
KAN	2 (1)	8 (7)	5 (4)	6 (3)
LSTM	3 (1)	8 (4)	8 (9)	9 (2)
N-BEATSx	2 (1)	6 (4)	8 (6)	6 (2)
TCN	2 (1)	6 (5)	8 (6)	6 (2)
TiDE	1 (1)	6 (3)	7 (7)	6 (2)
TFT	2 (1)	7 (4)	5 (6)	6 (2)

TABLE 7.10: MAPE (%) results from different individual models. Values show the median and (IQR) across countries for the optimal exogenous combination. ML and DL models are displayed above and below the dashed line, respectively. Per outcome, the best ML and DL model is in **bold** and the best overall model is in **green**.

The results in Table 7.10 show differences in performance across models, both within and between the ML and DL classes. Among the ML models, the lasso performed best on epidemiological variables and achieved the lowest error overall for hospitalisations and deaths. However, it performed poorly on tweet volume, likely due to their volatility, in which case SVR was the best-performing ML model. Among DL models, performance varied by outcome, with TiDE performing best on cases and hospitalisations, KAN on deaths, and the DilatedRNN on tweet volume. Overall, TiDE and the DilatedRNN were consistently the best or close to the best-performing models, though differences across DL models remained small.

Comparing exogenous and AR-only models

The preceding results showed that the impact of including exogenous variables varies across models. To more clearly assess this impact, I compared the best-performing exogenous model with its AR-only counterpart for each country, focusing on whether the exogenous model performed better on average. This contrasts with the previous results in this Section, which examined whether the best overall model improved on the baseline. These results are presented in Figure 7.7.

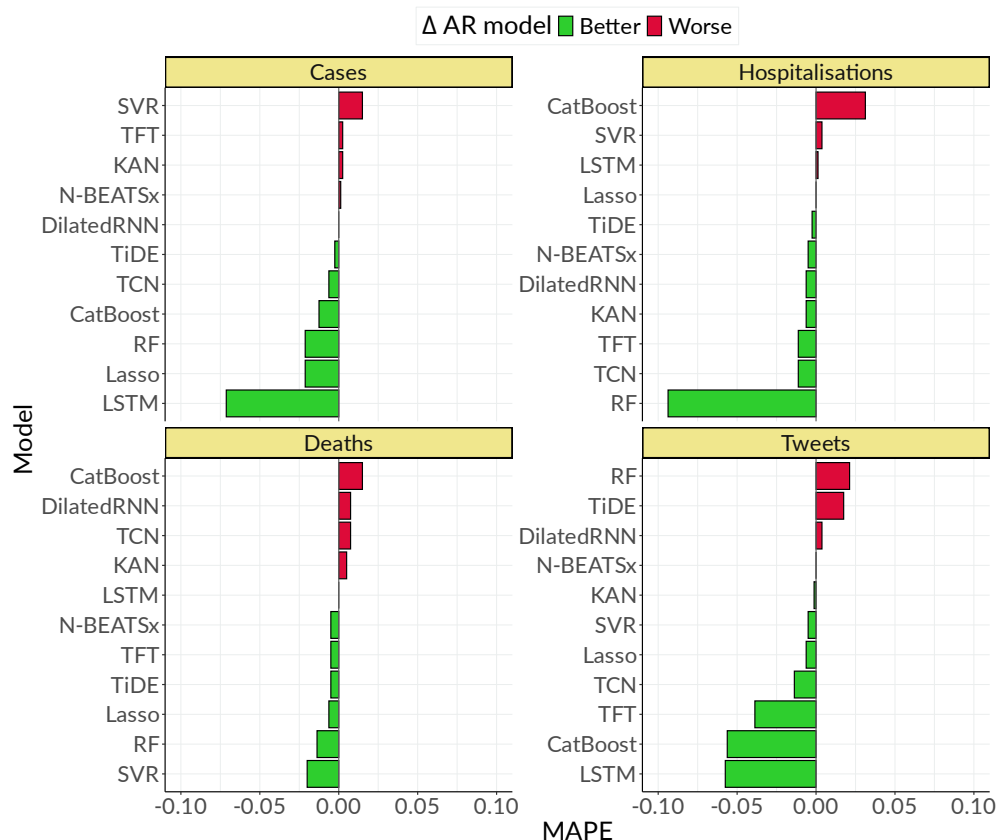


FIGURE 7.7: Average difference in MAPE scores across countries between the model without exogenous variables and the exogenous model with the lowest MAPE score for each outcome. Positive values (red) indicate that including exogenous variables performs worse on average than the AR-only model, whereas negative values (green) indicate an improvement.

The models that benefited from exogenous variables and the extent of those benefits varied across variables, as shown in Figure 7.7. The LSTM showed the largest gains for cases and tweet volume, yet exhibited no change for deaths and performed marginally worse for hospitalisations. A similar contrasting pattern was observed for the random forest and CatBoost, which benefited from exogenous variables for some variables but not others, with inconsistent gains across deaths, hospitalisations, and tweet volume.

For hospitalisations, the random forest achieved the greatest improvement; however, Figure 7.5 showed that the random forest only included exogenous variables in two instances, and in one of these it performed notably worse than the baseline. Across DL models, differences were modest, with the largest degradations occurring when deaths were modelled alongside exogenous variables. The TFT benefited across variables except cases, while the TCN similarly benefited except when predicting deaths. Two notable outliers were the LSTM’s improvement when predicting cases with exogenous variables and TiDE’s degradation in tweet volume prediction under these conditions.

7.5.3 Model lags

This Section focuses on the lag-related hyperparameters selected for the final tuned models presented in Section 7.5.2, namely the input sequence length for DL models and the set of four lags for ML models. The following results examine differences in the selected lags across models, exogenous variable setups, and variables.

Figure 7.8 shows the results for epidemiological variables. Despite being able to select input sequence lengths of up to 56 days, most models used shorter lengths. On average, deaths used the shortest input sequence length ($\mu=12$ days, $\sigma = 4$ days), followed by cases ($\mu=13$ days, $\sigma=9$ days), and hospitalisations with the longest ($\mu=16$ days, $\sigma=6$ days).

Within groups, models incorporating tweet volume as the sole exogenous variable exhibited the longest average sequence lengths for cases and deaths. However, differences in average input sequence length across exogenous-variable combinations were minor for deaths. For cases, the longer sequence length associated with tweet volume was primarily driven by the LSTM, which had an outlying length of 55 days. Hospitalisations exhibited the greatest within-group variability, with sporadic spikes across variables but no consistent pattern. Nonetheless, most models relied on approximately two weeks of data for this outcome. Among individual models, the LSTM used the most data on average ($\mu = 20$ days) and also exhibited the highest variability ($\sigma = 13$ days). In contrast, the TCN used the least data ($\mu = 10$ days) with the second-lowest variability ($\sigma = 2$ days), while the KAN also required little data ($\mu = 10$ days) and exhibited the lowest variability ($\sigma = 1$ day).

The corresponding breakdown for tweet volume is shown in Figure 7.9. On average, models used 14 days of data ($\sigma = 9$), similar to epidemiological variables, but with

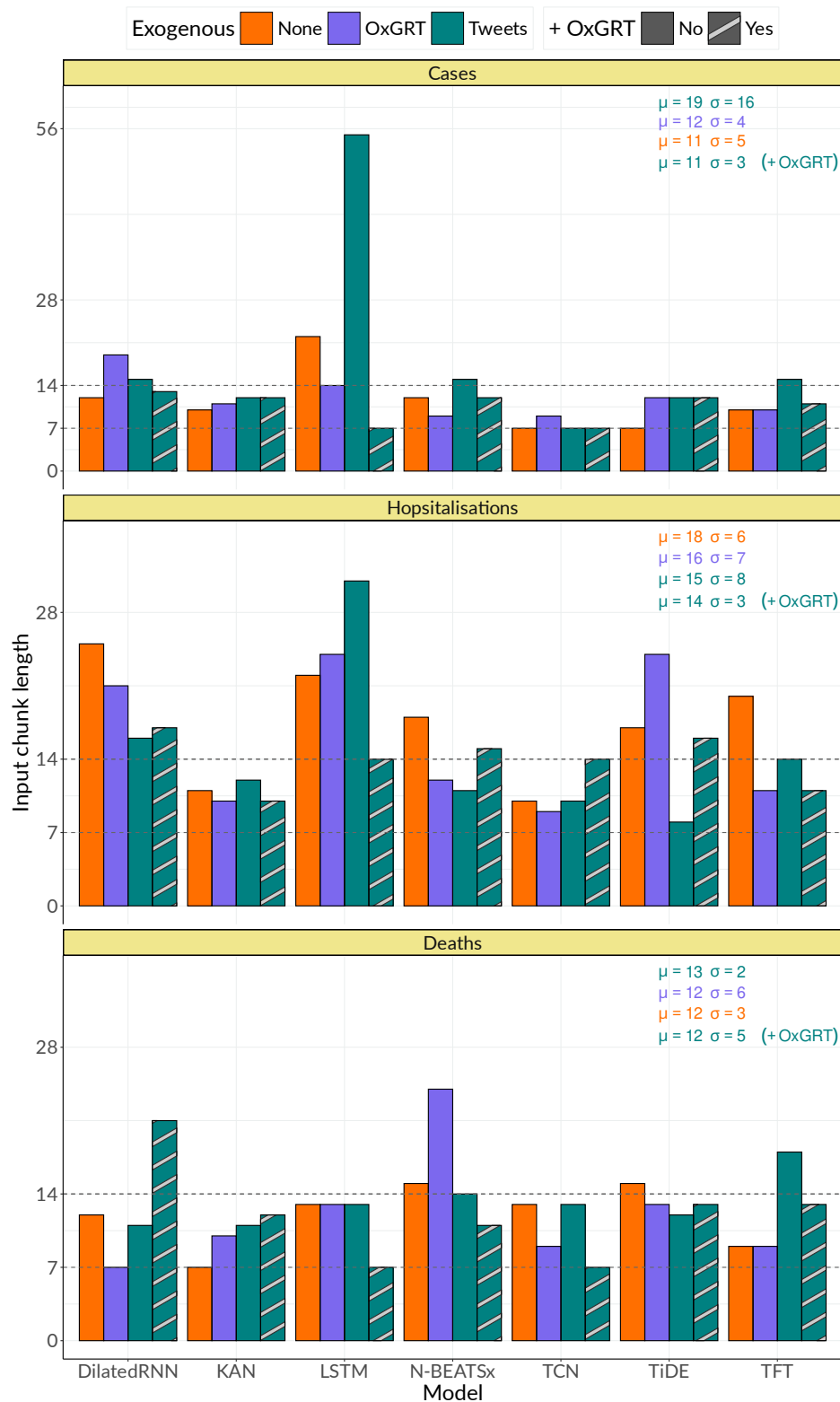


FIGURE 7.8: Comparison of input sequence length per DL model across epidemiological variables, stratified by exogenous variable combination. The mean and standard deviation across countries for each combination are superimposed.

greater heterogeneity across models and exogenous combinations. The DilatedRNN, the best-performing model on tweet volume, used the most data (25 days, $\sigma = 3$). In contrast, N-BEATSx, which performed comparably, was the most consistent, using 7 days across all combinations except the AR-only case, where it used 10 days. The KAN and TCN likewise relied on little data ($\mu = 9$ days, $\sigma = 1.4$). On average, models without exogenous variables used the most data, followed by those with included either deaths or hospitalisations. The higher averages for these exogenous combinations were driven by the TFT. Across all models, however, there was no consistent change associated with exogenous variables.

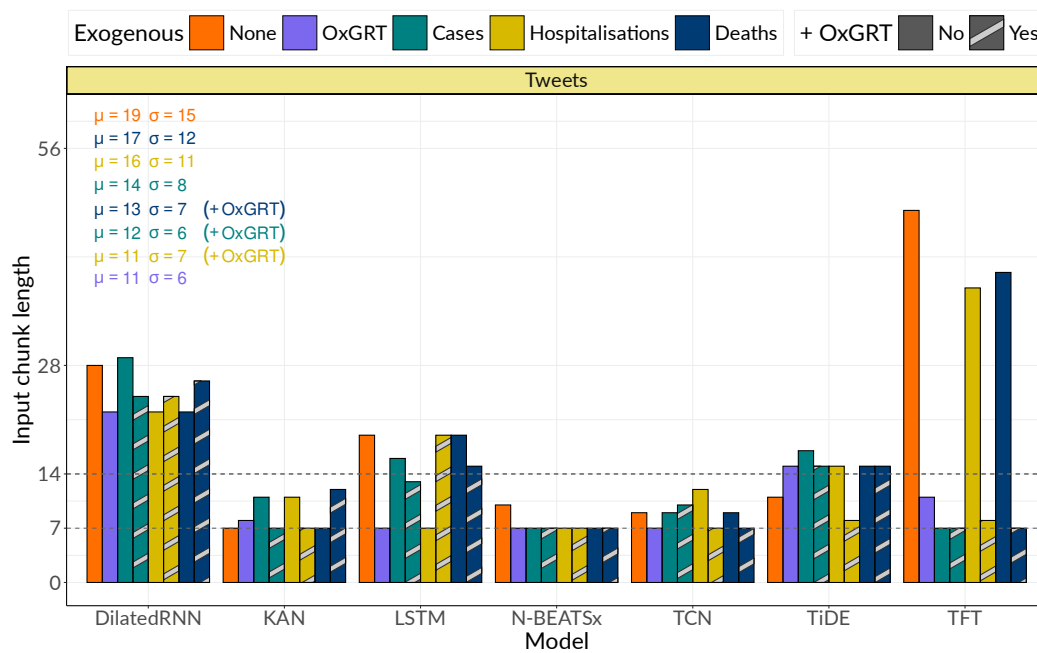


FIGURE 7.9: Comparison of input sequence length per DL model for tweet volume, stratified by exogenous variable combination. The mean and standard deviation across countries for each combination are superimposed.

The lag structure for the ML models was more complex and challenging to present, particularly since the same features appear across multiple combinations. For example, every model includes lags of the response, and there was overlap between specifications that include OxGRT. To provide a succinct comparison, the results are summarised as shown in Table 7.11. Results were averaged across the lasso and SVR, excluding tree-based models due to their poor performance. For each lag, variables are reported by exogenous variable, with the available lag options listed in the table header. Differences for other models that included the same variable were shown in brackets; where a variable appeared in more than one exogenous combination, the

average difference is reported.

As an illustration of how to interpret the Table, consider Lag V1 for cases. The model without exogenous variables selected a lag of one day (i.e., value at time t when predicting $t + 1$). On average, including exogenous variables (OxGRT, tweet volume, and tweet volume + OxGRT) increased this lag by 0.2 days. OxGRT and tweet volume used lags of three and three and a half days, respectively, while in the combined OxGRT + tweet volume model, each lag was reduced by one day.

Outcome	Exogenous	Lag V1	Lag V2	Lag V3	Lag V4
Lag options		[1, 2, 3, 4]	[5, 6, 7, 8]	[9, 10, 11]	[12, 13, 14]
Cases	None	1 (0.2)	7 (-1)	10 (-0.2)	13.5 (-1.2)
	OxGRT	3 (-1)	8 (-1)	10 (1)	13 (0.5)
	Tweets	3.5 (-1)	7.5 (0)	10.5 (-0.5)	13 (-0.5)
Hospitalisations	None	1.5 (0)	6.5 (-0.5)	9.5 (0.7)	13 (-0.5)
	OxGRT	3.5 (-2)	7.5 (-0.5)	10 (0)	13 (0.5)
	Tweets	3.5 (-1)	7.5 (0)	10 (0)	13.5 (-1)
Deaths	None	1.5 (-0.3)	6.5 (0.2)	9.5 (0.7)	13.5 (-0.7)
	OxGRT	2.5 (-0.5)	7.5 (0)	10 (1)	13 (0.5)
	Tweets	4 (-1.5)	6.5 (1)	10 (0)	13 (-0.5)
Tweets	None	1 (0.4)	6 (0.3)	10.0 (-0.1)	13.5 (-0.1)
	OxGRT	1 (1)	6.5 (-0.7)	10.5 (-1)	12.5 (0.5)
	Cases	2.5 (0.5)	5.5 (1)	10.5 (-0.5)	12.5 (1)
	Hospitalisations	2 (1)	5.5 (2.5)	11 (-1)	13 (0.5)
	Deaths	1 (2)	7.5 (-0.5)	10 (1)	13.5 (0)

TABLE 7.11: Comparison of the optimal lag, averaged across SVR and the lasso, for each lag variable and exogenous combination. Differences relative to the combined variants of the exogenous combinations are shown in brackets.

Considering Lag V1 (days 1–4), AR inputs were between 1 and 1.5 days. For epidemiological variables that incorporated tweet volume, later lags within the block (around 3–4 days) were often preferred, but these were systematically reduced to approximately 2.5 days when OxGRT was also included. When multiple exogenous features were combined, shorter lags were generally selected for epidemiological variables. For tweet variables, shorter lags were typically selected, although the inclusion of cases represented a notable exception. In contrast, when multiple exogenous features were combined for tweet variables, longer lags were more likely to be selected.

Across the subsequent blocks, lag selection was highly consistent. Lag V2 was typically chosen between 6 and 7 days, Lag V3 between 9 and 10 days, and Lag V4 between 13 and 14 days. This stability was broadly similar across cases, deaths, and hospitalisations, indicating a shared lag structure for epidemiological variables. The

inclusion of exogenous variables had mixed effects but did not alter the tendency to avoid the earliest option in each block. By contrast, tweet variables displayed much greater variability, with no clear systematic pattern emerging across exogenous inputs.

Overall, lag selection was stable for epidemiological variables, clustering around 7- and 14-day cycles that may reflect residual weekly reporting effects despite smoothing. Tweet volume, however, showed more variable lag structures, likely reflecting stronger interactions between exogenous inputs.

7.6 Discussion

In this Chapter, I assessed whether tweet volume and policy stringency were linked to epidemiological variables, and conversely, whether these variables and policy stringency drove tweet volume. I applied time series modelling to address this, comparing AR-only models with models that incorporated exogenous variables, and evaluated performance using out-of-sample prediction as a stronger indicator of association than the in-sample correlations examined in Chapter 6.

To refine the experimental design, I first tested how different modelling choices affected performance. Smoothing had a clear impact, with smoother series performing best and the C-spline proving most effective. This aligned with Zbezhkhovska and Chumachenko (2025), who found that STL smoothing improved both ML and DL performance. Weekly refitting compared to less frequent refitting also enhanced results, particularly boosting ML performance on hospitalisations and both model classes on tweet volume. Response transformations further shaped variables, with DL models performing best with unlogged responses, while ML models performed better with logged epidemiological variables but unlogged tweet volume. Finally, including four lag variables improved results on average.

The lack of standardisation in forecasting studies complicates comparison across results. Many studies that were not compared against a benchmark task also did not include a naïve baseline such as drift method, even though such methods have often been shown to perform as well as, or better than, more advanced approaches (Ribeiro et al., 2020; Li et al., 2023; Zbezhkhovska and Chumachenko, 2025; Makridakis, Spiliotis, and Assimakopoulos, 2020). This inconsistency, combined with the

limited application of many of the models I consider, makes it difficult to situate my design decisions within the broader literature and, more broadly, to contextualise the findings presented here.

Nevertheless, in this design, DL models generally performed best and were especially effective for cases. The impact of exogenous variables varied. For cases and hospitalisations, the same median but a lower IQR was achieved when including policy stringency and a combination of policy stringency and tweet volume, respectively, compared to AR-only models. For deaths, including no exogenous variables, yielded the best performance. ML models consistently performed worse with exogenous variables and performed poorly overall. In general, no exogenous variable notably improved performance, but in most instances, it did not notably degrade performance either.

Stratifying performance by model type clarified the sources of variation across methods and variables. Poor results from tree-based methods, reflecting their sensitivity to noise and distributional shifts, largely drove the weak performance of ML models. In contrast, the lasso performed strongly, achieving the best results for hospitalisations and deaths, and remaining competitive with DL models for cases. Its success likely reflects the structure of these series, which featured fewer, more clearly defined peaks and extended flat periods that the linear model could capture effectively. Previous work has shown the lasso to perform well in forecasting dengue, a seasonal infectious disease (Shi et al., 2016; Chen et al., 2018). My findings extend this evidence to an outbreak context, demonstrating its utility for COVID-19 hospitalisations and deaths. However, all ML models, including lasso, performed poorly on tweet volume, where high volatility and short peaks may have contributed to weaker results. Closer examination of model behaviour at peaks, motivated by the divergence observed in ML models between epidemiological variables and tweet volume, could provide useful insight into performance.

DL models were generally more robust across exogenous variable combinations, consistently outperforming the baseline except for the LSTM on tweet volume. Among them, TiDE and the DilatedRNN achieved the best results, although differences across architectures were modest. The strong results of both lasso and TiDE highlight the continued importance of linear dynamics in forecasting. For TiDE, the contribution of its linear sub-model cannot be isolated, but its strong performance on

tweet volume, where lasso struggled, shows that the value of linear methods is context dependent. That TiDE succeeded where lasso failed underscores the advantage of combining linear and nonlinear elements (Zeng et al., 2022).

Performance varied substantially across countries. Models performed particularly poorly when predicting deaths in the Netherlands and Portugal, while tree-based methods struggled with hospitalisations and deaths in Poland. In contrast, models achieved notably strong results on Italian tweet volume, likely reflecting Italy's relatively stable tweet volume following the initial peak period of public attention. These patterns may reflect distributional differences across the training, validation, and test sets, including lower hospitalisation and death counts as well as reduced tweet volume during the test period compared with earlier stages of the pandemic.

More broadly, cross-country variability is not unique to COVID-19 but is a recurring challenge when modelling infectious diseases across countries (Kretzschmar et al., 2022). In an Ebola forecasting challenge, where mechanistic models were primarily used, differences in surveillance systems, case definitions, and reporting artefacts created similar difficulties to those observed here (Viboud et al., 2018). Such heterogeneity underscores the importance of outcome definitions. Models here were fit to reported cases, hospitalisations, and deaths, but national testing strategies, case ascertainment, and reporting practices all shaped these measures (Verity et al., 2020; Russell et al., 2020; Hasell et al., 2020). Moreover, because hospitalisation data were difficult to obtain, outcome definitions varied across countries. Alternative measures, such as excess mortality, rates of change, or estimates of the time-varying reproductive number, may provide more harmonised signals across settings and reduce artefactual variability introduced by country-specific surveillance practices (Msemburi et al., 2023).

At the country-model level, some improvements were observed, but most models still performed better with no exogenous variables. When predicting epidemiological variables, between 58% and 68% of the best performing models used no exogenous predictors, while 11%–23% favoured tweet volume, 7%–18% OxGRT, and 9%–19% their combination. When predicting tweet volume, OxGRT (15%), hospitalisations (11%), and cases (11%) were the most commonly selected predictors, but AR-only models still dominated, accounting for 52% of the best performing models. These patterns were consistent when restricting models to a single exogenous combination

across all countries, with tweet volume most often included for epidemiological variables (18%). In comparison, OxGRT was most often included when predicting tweet volume (18%).

Overall, these results suggest possible predictive links, though they remain challenging to disentangle. For epidemiological variables, no single exogenous variable consistently performed best, and each may have a predictive association with specific variables. For tweet volume, the level of policy stringency, followed by hospitalisations and cases, showed the strongest predictive associations of the exogenous variables considered. However, because only a small proportion of models selected these predictors, support for these links remains limited. This variability highlights the importance of local data characteristics when evaluating the role of exogenous predictors and also highlights that model architecture may influence whether an exogenous variable proves useful.

The models included in this Chapter were selected to capture a diverse range of approaches, both to increase the chance of identifying associations and to provide a more comprehensive test of the utility of exogenous variables. However, as outlined in Chapter 3, these architectures differ substantially in how they handle exogenous inputs. Particularly for DL models, many designated as state-of-the-art are optimised to maximise predictive accuracy on benchmark tasks such as the Makridakis competitions (Makridakis, Spiliotis, and Assimakopoulos, 2018; Makridakis, Spiliotis, and Assimakopoulos, 2020). The emergence of many recent architectures that do not accommodate exogenous variables further emphasise this forecast-oriented view (Cao et al., 2020; Zeng et al., 2022; Nie et al., 2023; Chen et al., 2023; Wu et al., 2023; Ansari et al., 2024). This distinction was evident in my findings, with models such as the LSTM and TFT often benefiting from exogenous inputs, whereas others, including N-BEATSx and TiDE, seldom did. My choice of models may therefore have biased results, with some architectures inherently less suited to leveraging exogenous information.

Furthermore, despite evidence of potential predictive links, the use of tweet volume breaks the connection to valence, making the behavioural value of this association less clear. Nevertheless, prior studies have shown that both historical Google Trends attention and tweet volume correlated with or even predicted specific epidemiological variables, and my findings add further support to this literature (Yang, Santillana, and Kou, 2015; Doogan et al., 2020; Turiel, Fernandez-Reyes, and Aste, 2021; Cheng

et al., 2021; Wang et al., 2022b; Cheng, Han, and Liu, 2023). What remains unclear, however, is whether internet-based measures of public attention can act as proxies for determinants of behaviour. A clearer understanding and formalisation of this potential relationship would allow these variables to be considered for inclusion more explicitly in broader analytical contexts, for example in mechanistic models.

Beyond overall performance, differences also emerged in how models used input sequence lengths and lag structures. DL models typically performed better with short input sequences of two to three weeks, with deaths requiring the least input data, followed by cases and then hospitalisations. No consistent changes in input length were observed when specific exogenous variables were included. The similarity in input length across all three variables is notable, given that hospitalisations and deaths are dynamically linked to cases, with hospitalisations typically lagging cases and deaths in turn lagging hospitalisations (Ward and Johnsen, 2021). These nested dynamical processes produce progressively smoother trajectories, as each outcome reflects a delayed subset of the preceding one. The relatively short forecast horizon of seven days that was used likely reduced the importance of modelling the full lag structure, since much of the progression from cases to hospitalisations and deaths extends beyond a single horizon. By contrast, shorter horizons were sufficient to track the faster-moving dynamics of cases, which may help explain the comparatively strong performance of DL models on this outcome.

ML models similarly showed consistent lag preferences. In particular, lags between 6-7 and 13-14 days, performed best within two of the four lag blocks, likely reflecting reporting cycles. Exogenous inputs shifted these lag preferences slightly but did not alter the overall structure. Tweet volume displayed the most variable lag patterns, suggesting stronger interactions with exogenous inputs. Interpreting these results for ML models requires consideration of the lag structure, which may have constrained model performance. More flexible specifications, including dilated lags over a longer window of lag options, could have enabled richer comparisons. However, it is worth noting that the two-week window used here falls within the one-to-three-week range reported in broader evidence for Google Trends and tweet-related variables (Effenberger et al., 2020; Wang et al., 2022b; Lamsal, Harwood, and Read, 2022), which also supports the input sequence lengths found for the DL models. Despite this potentially restrictive window, the strong performance of the lasso under a sparse

specification of dilated lags is encouraging.

7.7 Future work

The previous Section considered a wide range of results, but several extensions could further enhance the scope and validity of the comparisons as discussed in this Section.

Design choices and forecast horizons were key limitations of this Chapter. I aimed to create a standardised setting for model comparison, conducting many experiments to determine an appropriate design. This was a greedy approach adopted for parsimony, but exogenous models may have performed better under alternative settings. I also focused only on a single, relatively short forecast horizon. Extending analyses to longer horizons would be more valuable for policy planning in outbreak response, while also providing a stronger basis for comparing model classes and assessing the contribution of exogenous variables, including a more detailed analysis of lag structures. While AR inputs may be sufficient in the short term, this assumption may not hold when forecasting further ahead.

Alternative data structures could also improve robustness. Smoothing improved performance in my setting, but using weekly data could have resolved this issue while also allowing analysis of model behaviour under data-sparse conditions, which are particularly relevant in early outbreak scenarios. If models can perform well with less data, it would be worthwhile to repeat the analysis stratified by epidemic wave to identify if there are periods where the variables I consider are more informative. Such a design would also permit the inclusion of variant and vaccination data, the former of which was shown to be particularly useful by Du et al.'s (2023) in their modelling pipeline for forecasting US COVID-19 cases and deaths.

Another useful direction would be to repeat experiments with the AR component excluded. When included, the question becomes whether exogenous variables add predictive value beyond the existing temporal dependence in these dynamical processes. Excluding it, by contrast, would allow a more precise assessment of how exogenous variables relate to the response. This could also shed light on lag structures, where predictors may be associated with variables at longer horizons and thus act as earlier indicators than an outcome's own past values.

It would also be valuable to assess whether sentiment- and search term category-based stratifications of tweet volume offer stronger predictive value. Although they showed similar correlations in Chapter 6, their predictive contribution may differ and could be better captured by time series models. Incorporating sentiment in particular would re-establish the behavioural link to valence. Investigating this stratification is further supported by Lamsal, Harwood, and Read (2022), who showed that a subset of COVID-19 topics on Twitter, when expressed with particular sentiments, Granger-caused COVID-19 cases in Australia, as discussed in Section 7.3.

Heterogeneity observed across countries highlights the need for both local modelling and interpretability. Future work may benefit from fitting models locally to better capture relationships between predictors and variables, while also allowing country-specific lags to emerge. At the same time, focusing on interpretable architectures could help clarify why exogenous variables are excluded and, when included, how they are used. Although I included some interpretable designs in this Chapter (e.g. TFT and N-BEATSx), this dimension was not fully leveraged here. Given the similar performance observed across models, a smaller, interpretable subset may be more valuable to pursue in future work, both for comparing global versus country-specific models and for understanding the drivers of the heterogeneous results.

Finally, although a diverse set of model classes were used in this Chapter, it was not possible to include all potentially relevant approaches, given the practical constraints on scope and feasibility. Graph Neural Networks (GNNs) have recently been proposed for infectious-disease modelling, with their use in epidemiology further supported by Liu et al.'s (2024) recent review and the epidemiologically focused package `epilearn`, which implements GNNs as well as many of the models used in this Chapter (Cao et al., 2020; Liu et al., 2024b). LLMs have likewise been explored as time series encoders, though their utility in outbreak analytics remains uncertain because their training on data extending beyond 2020 complicates evaluation on retrospective outbreaks (Gruver et al., 2023; Jin et al., 2024; Ansari et al., 2024). As noted in Chapter 5, this limitation may be less pronounced in NLP, but its impact in time series contexts is difficult to assess. Mamba, a state-space alternative to transformers, also represents a promising direction for future investigation (Gu and Dao, 2024).

7.8 Conclusions

In this Chapter, I set out to evaluate whether tweet volume and policy stringency were associated with future epidemiological variables, and conversely, whether epidemiological variables and policy stringency were associated with future tweet volume. To address this, I compared a range of time series models, including recently proposed DL models, some of which, to the best of my knowledge, have not previously been applied in this setting.

Performance varied across countries and models. Overall, DL models achieved lower MAPE scores than ML models and performed consistently well across variables, particularly for cases. Within the DL class, TiDE and the DilatedRNN were among the top performers, though differences across architectures were modest. In contrast, ML performance was more heterogeneous, with tree-based models performing poorly while the lasso was competitive for hospitalisations and deaths. However, like other ML models, the lasso performed poorly on tweet volume, where volatility and the absence of clear peaks posed challenges. At the country level, all models tended to succeed or struggle in the same settings relative to their own performance, underscoring the challenges that local data and epidemic dynamics pose for cross-country analysis and the application of global models.

Across variables, no single exogenous variable or combination systematically improved performance, though results varied when disaggregated by country and model. At model level, tweet volume was most often included for epidemiological variables, while policy stringency was most often included when predicting tweet volume, although such cases were limited across all responses. At the country level, there were also instances where exogenous variables improved forecasts, but the best-performing variable differed by outcome, reinforcing the challenge of cross-country variability. Furthermore, models varied in the extent to which they benefitted from exogenous inputs, with the TFT, the LSTM, and the lasso showing the greatest gains among better-performing models.

Among the hyperparameters considered, lag variables were focussed on given their importance in relating past information to the response. For DL models, which could use up to eight weeks of data, two to three weeks of input data performed best on average across variables. For ML models, which were allowed to choose four lags from

consecutive blocks within a two-week window, the selected lags were often around 6–7 days and 13–14 days, with greater variability in the remaining choices. The two-week window specification for ML performed well but could be extended, particularly when using dilated lag structures.

This variability in the results of how models responded to exogenous variables underscores the need for more granular investigation. Future studies could focus on interpretable models, including those tailored to incorporating exogenous variables, and on interpretable components within DL models to better characterise these relationships. Moreover, longer forecast horizons and country-specific models could help clarify whether there are specific conditions under which tweet volume and its extensions provide value.

In conclusion, this Chapter provides evidence of predictive associations between tweet volume, policy stringency, and epidemiological variables. These findings suggest that, in a future pandemic, social media data could help predict disease trajectory and provide a barometer of public attention to policy stringency. However, these relationships are context-dependent, and their practical value requires further investigation. To strengthen pandemic preparedness, future research should seek to characterise the mechanisms and conditions that shape these relationships. This work establishes a time series modelling foundation on which such research can build.

8 Discussion

8.1 Summary of thesis

Motivated by the rapid expansion of social media and recent advances in NLP, my Thesis examined the potential of social media data as a source of information during outbreaks. COVID-19, unprecedented in scale and accompanied by widespread online engagement, provided a unique case study. In this context, I considered sentiment-stratified tweet volume as a potential proxy for public valence, a latent behavioural determinant. I examined the entire process, from curating a dataset and classifying sentiment to analysing how the resulting tweet variables related to epidemiological and public policy variables.

In Chapter 4, I introduced a novel dataset of tweets designed to capture diverse discussions related to the COVID-19 pandemic. Unlike existing collections, which often relied on narrow query sets, I constructed this dataset using a broad range of search queries, which was developed in discussion with a subject matter expert and translated to other languages by colleagues with expertise in infectious disease epidemiology. These queries spanned three categories, namely public policy, vaccination as a specific subset, and general disease-related terms. Data were gathered across eight countries and languages, yielding 4,813,344 tweets. I then developed an approach to inflate the raw dataset for analysis, allowing the same tweet to be matched to multiple search terms. To the best of my knowledge, such an approach has not been applied in other tweet-level analyses and represents an important contribution of this Thesis. Applying this method expanded the dataset to 5,154,424 tweets. In accordance with Twitter's terms of service, only tweet IDs can be shared, which users with API access can download through a process known as rehydration. Since the cessation of the Twitter Academic API, however, the utility of such ID lists has become extremely limited. Nonetheless, the sentiment assignments presented in Chapter 7, together with the search terms, will be made publicly available and remain a valuable output of this Thesis.

Assessing the existing literature on sentiment analysis in public health and in applied settings more broadly, I found that, despite recent methodological advances in NLP,

many studies have continued to rely on simpler approaches, likely because of their ease of use. From an NLP perspective, research has often emphasised methodological innovation, prioritising accuracy while overlooking practical considerations of deployment, which may further contribute to the lack of broader adoption. LLMs, however, now present an accessible option that can match or even surpass previous state-of-the-art approaches. They only rose to prominence toward the end of the COVID-19 pandemic and remain underexplored in this context. To address this gap, in Chapter 5 I conducted an empirical study comparing the performance of three classes of methods, lexicons, encoder LMs, and LLMs, for sentiment polarity classification, using a multilingual dataset of 2,530 tweets across eight languages drawn from the larger collection. For LLMs, I explored a range of prompting strategies varying in both complexity and language of instruction. The results showed that fine-tuned encoder models achieved the highest performance, with $\text{XLM-T}_{\text{LARGE}}^{\text{FT}}$ performing best overall, while LLMs were competitive when prompted effectively. To encourage the adoption of more accurate approaches in multilingual public health contexts, I evaluated each model class across criteria beyond accuracy. I provided practical recommendations to support non-specialist practitioners in selecting appropriate methods. These findings highlight the trade-off between simplicity and accuracy. While fine-tuned encoder models currently offer the best performance, LLMs provide a pragmatic alternative that may lower barriers to adoption in future outbreak responses.

Building on the findings in Chapter 5, I applied $\text{XLM-T}_{\text{LARGE}}^{\text{FT}}$ in Chapter 6 to classify sentiment from the larger analysis dataset, enabling an investigation of how Twitter discourse evolved during COVID-19 and how it related to epidemiological and public policy variables. Attention surged early in the pandemic before waning, with smaller spikes around key events such as vaccine rollout. Across countries, most tweets were negative, with vaccines, lockdowns, quarantines, and masks dominating discussion, though negativity was not tied to specific topics. Engagement was also highly skewed toward a minority of posts. Using Kendall's τ , I found similar associations across most epidemiological variables and tweet volume, suggesting comparable responses regardless of outcome severity. Tweet volume was most strongly correlated with policy stringency, particularly overall restriction levels and their implementation, while vaccine-related discourse showed the greatest cross-country variation. Stratifying by sentiment or search term category did not alter associations, indicating that the overall attention to COVID-19-related topics, rather than its composition, is the primary

correlate in this setting.

In Chapter 7, I extended the analysis of associations using time series models, which capture non-linear relations and longer temporal dependencies to provide a more structured assessment. I focused on tweet volume alone, given the similarity in correlations between sentiment- and search term category-stratified measures; this focus allowed a more detailed analysis of a single variable within available computing resources. I assessed two directions of association: whether tweet volume and policy stringency improve predictions of epidemiological variables, and conversely, whether epidemiological variables and policy stringency improve predictions of tweet volume. Comparing AR-only models with those including exogenous variables, I found that the latter did not improve performance overall. DL models generally outperformed ML, however, the lasso was competitive for epidemiological variables but performed poorly on the more volatile tweet volume series. DL models preferred short input windows of 2–3 weeks, while ML models were restricted by design to 14 days, which should be expanded in future work. Moreover, I applied a parsimonious dilated lag structure specification for ML models, under which the lasso performed well. There was heterogeneity in performance across countries and models, and in some cases, exogenous variables improved forecasts. Although rare, these cases suggest that associations may exist in specific contexts, and warrants future work. This work should emphasise interpretable models, longer horizons, local fits, and harmonised variables to determine whether stronger associations emerge and under what conditions tweet volume, and potentially its stratification, provides reliable value.

8.2 Limitations

There are two main classes of limitation in this Thesis, namely those related to the data used and generated and those related to the analyses conducted with the resulting data.

8.2.1 Data limitations

The first, and perhaps most substantial, data limitation of this work arose from the platform used. When I began this project in 2022, Twitter was widely used in computational social science, including public health applications, in part attributable to the comprehensive Academic API made available to researchers (Tsao et al., 2021;

Pilipiec, Samsten, and Bota, 2023). However, in the second year of my PhD, this API was discontinued, with no analogous replacement (Blakey, 2024). This had two notable consequences. Firstly, it prevented expansion of the dataset to additional countries, the incorporation of refinements identified after my initial analysis, and the ability to better leverage larger datasets already collected, particularly since I found that a few popular search terms captured by other datasets accounted for a large proportion of the total volume. Secondly, it limits the utility of the curated dataset and the methods described in Chapter 4.

More broadly, the removal of the API reflects a wider trend among established social media platforms of increasingly restricting access to their data, which also undermines the reproducibility of historic work (Freelon, 2018; Acker and Kreisberg, 2020; Bruns, 2021; Poudel and Weninger, 2024). In contrast, newer platforms are emerging that may provide continuity for this line of work. BlueSky is a notable example, which unlike Twitter, is decentralised, openly shares data, and offers a free API (Bluesky Documentation, 2025). Early studies have already begun characterising the platform and datasets derived from it, suggesting potential pathways for future research (Quelle and Bovet, 2025; Balduf et al., 2024; Jeong et al., 2024; Rostami et al., 2025).

A second limitation is the potential for demographic bias in the dataset. Twitter does not provide demographic information, making it impossible to assess the representativeness of its users. Moreover, only a small fraction of tweets ($\sim 1\text{--}2\%$) are geotagged (Twitter, 2023). These challenges are compounded by high variance in sample sizes across countries, likely age-related biases in the Twitter user base, and systematic differences in who chooses to geotag their posts (Sloan and Morgan, 2015; Huang and Carley, 2020). If demographic information were available in a privacy-preserving format, it would enhance the utility of social media data by enabling more representative analyses and closer alignment with epidemiological variables, which are typically stratified by age or other factors. As with mobility data, reliance on aggregated, non-representative samples limits the extent to which results can address the same questions as purpose-designed surveys.

A third limitation relates to my focus on sentiment polarity as the primary outcome. While pragmatic and broadly generalisable, polarity may obscure behavioural determinants by collapsing emotions with distinct implications into a single valence dimension. As outlined in Chapter 1, the valence–arousal framework highlights that

emotions such as fear and anger, though both negative in valence, differ in arousal and may drive opposite behaviours, fear may increase adherence to protective measures while anger may reduce it (Choi and Kim, 2022; Erhardt et al., 2021). Polarity was chosen to enable a standardised multilingual analysis, yet even with this simplification, curating such datasets remained challenging. As explained in Chapter 5, I was only able to obtain a single annotator for non-English languages.

8.2.2 Analytical limitations

In this Thesis, I found suggestive evidence of associations among the variables considered. Cross-correlations and time series modelling provide some indication of directionality, but these approaches cannot establish causality. It remains unclear whether the variables influence one another directly or are jointly shaped by external factors. This limitation is typical of associative research and is captured by the familiar cautionary maxim that correlation does not imply causation.

Additional analytical limitations concern the structure of the experiments themselves. Specifically, the analyses in Chapters 6 and 7 lack temporal disaggregation, and the latter is further constrained by short forecast horizons and limited country–model-level detail. As these issues align closely with avenues for future research, they are elaborated in Section 8.3.2.

8.3 Future work

8.3.1 Expanding behavioural measures

Building on the limitation of sentiment polarity, future work should explore alternative measures that may act as stronger determinants of behaviour and show greater association with the variables examined. Two dimensions in particular warrant attention. The first is granularity. I assessed sentiment attached to a specific search term at the tweet level, whereas a more refined approach could assess terms locally or focus on outcomes associated with the terms themselves. For example, aspect-based sentiment may provide more accurate signals around a specific term (Nazir et al., 2020).

The second aspect concerns the outcome itself. Emotion may be a more informative measure, as discussed in the previous Section. A challenge, however, is that emotion

involves many more categories compared to polarity, several of which are difficult to distinguish reliably. Consequently, it may be preferable to use a subset of emotions that are most strongly associated with behaviour according to existing frameworks (Lerner and Keltner, 2000; So, Kuang, and Cho, 2016).

In addition, it is useful to consider measures that are not directly related to sentiment. Three illustrative examples are discussed below. First, is the widespread disinformation circulated on social media during the COVID-19 pandemic (WHO, 2022). Second, proxies of prosociality merit consideration. According to Batson and Powell (2003), prosociality encompasses a range of actions intended to benefit others without necessarily considering one's own interests. Such proxies may serve as indicators of whether the public will adhere to policy interventions and adopt protective behaviours (Jordan, Yoeli, and Rand, 2021; Campos-Mercade et al., 2021; Fang et al., 2022; Abel and Brown, 2022). Third, quantitative measures of both government trust and social trust would be valuable. Trust is an important factor in public health crises, shaping how individuals perceive risk, whether they adopt protective behaviours, and the extent to which they adhere to policy measures (Bish and Michie, 2010; Siegrist and Zingg, 2014; Devine et al., 2021; Liu, Shahab, and Hoque, 2022; Adhikari, Yeong Cheah, and von Seidlein, 2022).

While social media is a platform for expressing opinions, it is also a source from which people seek information; in this context, Twitter functions as a tool that allows the public to produce, disseminate, and receive information (Tsao et al., 2021; Terry et al., 2023). Related to this, an expanded analysis of engagement with posts would better capture dissemination and reception, thereby enhancing the utility of disinformation measures.

8.3.2 Disaggregated and expanded modelling approaches

My analyses of tweet volume and policy stringency were conducted at an aggregate level across all search terms. An alternative approach would be to restrict the analysis to terms most directly related to the specific measures included in the policy stringency variable. This may clarify the relationship between the two variables at a higher level of aggregation. Further disaggregating at the policy level would also reveal which specific measures attracted the most attention on Twitter, serving as a proxy for broader public salience.

To support the suggestion above, while I considered a broad range of search terms, additional terms could have been included that were more directly tied to the specific policies captured in the stringency index. For example, considering containment and closure, search terms related to workplace and school closures, gathering-size restrictions, and limitations on domestic and international movement might have been especially relevant. A key challenge in expanding the set of terms, however, is that results may become less COVID-19 specific. Still, given the heightened focus on the pandemic during this period, it is plausible that most tweets would remain relevant. This trade-off highlights an important consideration both for my chosen set of terms and for any potential extensions.

The temporal nature of social media associations should be more carefully disaggregated, as their relationship with epidemiological and policy variables is unlikely to remain constant over the course of a pandemic. For example, terms such as `no smell` or `no test` are more closely tied to earlier variants and therefore likely to correlate more strongly with cases during those periods (Whitaker et al., 2022). More broadly, the dynamics of association between social media and other variables are likely to evolve, as suggested by the waning tweet volume observed in my work and more broadly by the issue-attention cycle (Downs, 1972). Such disaggregation would also allow the incorporation of covariates such as variant prevalence and vaccination data, which are particularly relevant in the context of Chapter 7. This approach, however, reduces the available data and may therefore be challenging when applied to DL time series models.

A more granular and expanded analysis represents a key area for future work. To ensure comparability, I adopted a parsimonious design, but exogenous models may have performed better under alternative specifications. Extending analyses to longer forecasting horizons would be particularly valuable, especially from a policy planning perspective, where short horizons such as seven days are often too limited to inform meaningful decisions. Longer horizons also create greater scope to examine when exogenous variables contribute information and to analyse lag structures in more detail, including the role of dilated lag periods. Alternative data structures, such as weekly aggregation, would smooth fluctuations caused by day-of-the-week effects and, in data-sparse settings, reduce the need for interpolation, while also facilitating longer forecast horizons. At the same time, focusing on interpretable architectures

would help clarify why exogenous variables are excluded and, when included, how they are used. Although I tested some interpretable designs (e.g. TFT and N-BEATSx), this dimension was not fully leveraged, and given the broadly similar performance observed across models, a smaller and more interpretable subset may be more valuable for understanding the drivers of heterogeneous results.

Beyond refining existing models, future work could also expand the scope of models considered. In Chapter 7, a broad set of ML and DL approaches were evaluated, but other promising classes were excluded due to scope and feasibility constraints. Examples include GNNs, LLMs as time series encoders, and state-space alternatives such as Mamba (Liu et al., 2024a; Liu et al., 2024b; Cao et al., 2020; Gruver et al., 2023; Jin et al., 2024; Ansari et al., 2024; Gu and Dao, 2024). Similarly, in Chapter 5, only a subset of models could feasibly be compared, narrowing the breadth of comparison both within the classes considered and more broadly, as traditional but potentially relevant ML approaches, such as SVMs with TF-IDF features, were omitted.

In addition to heterogeneity across models, performance varied systematically by country, with models displaying similar cross-country patterns. From a modelling perspective, developing country-level models may help address some of these issues by allowing local relationships and lags to emerge. From a data perspective, augmenting the analysis with additional indicators could also account for these differences. Incorporating testing rates, estimates of case ascertainment, and excess mortality would be particularly valuable (Verity et al., 2020; Russell et al., 2020; Hasell et al., 2020; Msemburi et al., 2023).

Beyond traditional epidemiological indicators, considering the mechanisms of transmission themselves may provide further insight. Just as Google mobility data have been used as proxies for contact rates, social media variables may also reflect shifts in population-level contact patterns, particularly during periods of heightened public attention such as the onset of the pandemic. As a first step, given the difficulty of incorporating social media directly into mechanistic models to generate R_t , a practical alternative is to use R_t estimates from other approaches (e.g., EpiEstim) and test their correlation with these variables (Nash, Nouvellet, and Cori, 2022). Moreover, analysing growth rates rather than levels may provide additional insight into whether such measures capture or anticipate meaningful changes. Although these analyses

would maintain the limitation of not being able to infer causality, they could provide a stronger indication of meaningful association.

8.4 Conclusion

In this Thesis, I adopted an interdisciplinary approach that integrated epidemiology, computational social science, and recent advances in NLP and time series modelling. From this perspective, I examined the role of social media in outbreak analytics, with a particular focus on sentiment in Twitter during the COVID-19 pandemic across multiple countries and languages. This approach offered a holistic view of how these tools can be applied in combination, in contrast to most existing work, which has treated them in isolation. My contribution is valuable given the growing availability of real-time data, albeit tempered by reduced access to social media sources, and the rapid evolution of NLP methods, particularly LLMs, whose potential in public health and related fields remains underexplored.

In particular, in this Thesis I have made the following contributions. I have provided practical guidance on constructing datasets from social media, including methods for interacting with nested data structures and strategies for managing overlapping search terms. Building on this foundation, I have compared a wide range of sentiment analysis methods in a multinational and multilingual setting, an area rarely addressed in applied NLP and public health research. From this comparison, I have outlined key model selection considerations that encourage the uptake of more recent NLP models, thereby helping to bridge the methodological gap between technical advances and their practical application. I have examined associations between tweet variables, including sentiment outputs derived from these models, and epidemiological as well as public policy variables, thereby contributing to ongoing efforts to assess the role of social media data in this context. Finally, by finding some evidence of such associations, I have provided a promising basis for further research to clarify their robustness, evaluate their relevance as behavioural determinants, and assess their practical utility.

Given the demonstrated utility of mobility data during the COVID-19 pandemic, parallels can be drawn for how social media data may be used in future outbreaks. Although access to social media data has become more restricted, it still holds considerable potential. While it would be ideal for platforms to remain open and for

research to continue transparently, a more realistic path forward may be the provision of aggregated metrics during crises. This approach is not without precedent, as seen in COVID-19-specific support from Twitter (although under different leadership) and Meta (Adam, 2020; Jin and McGorman, 2020). These illustrative examples demonstrate both the feasibility and value of platform-supported data sharing, and they highlight the potential for social media to contribute meaningfully to outbreak analytics in the future. One advantage of this approach is standardisation, as researchers would not need to construct these measures themselves and could instead focus on outbreak analytics rather than the technical challenges of NLP. As noted throughout this Thesis, NLP methods are advancing rapidly, yet interdisciplinary applications have not necessarily kept pace.

While social media data was available, researchers still had to generate their own measures to use it analytically. A key question that remains is which measures would be most useful in future outbreaks. As I highlighted in the introduction of this Thesis, unlike mobility data, extracting quantitative measures from text is more complex. My findings suggest that simple search term volume, similar to Google Trends, are a useful starting point. However, additional measures may prove valuable, such as the prevalence of emotions linked to behavioural determinants, stance toward policy and protective behaviours, levels of trust, and prosociality. This list is not exhaustive, and the growing capabilities of LLMs suggest that even richer indicators could be developed. To fully realise the potential of such measures, continued research from behavioural perspectives is essential to clarify how these indicators relate to the dynamical processes that drive outbreaks and to guide the identification of the most informative ones.

Looking ahead, integrating social media analytics into outbreak response holds considerable promise for delivering timely behavioural insights that can strengthen preparedness and response. In the era of LLMs, the ability to extract meaningful signals from text has increased dramatically, and this Thesis presents a systematic framework for conducting such analyses. By demonstrating how social media variables can be linked to behaviour, my work lays the groundwork for extending analyses to richer behavioural and epidemiological variables. Such extensions could enable integration with traditional mechanistic models to inform interventions, as well as provide real-time feedback to improve policy efficacy.

Bibliography

- Abaluck, Jason et al. (2022). “Impact of community masking on COVID-19: a cluster-randomized trial in Bangladesh”. In: *Science* 375.6577, eabi9069.
- Abel, Martin and Willa Brown (2022). “Prosocial behavior in the time of COVID-19: The effect of private and public role models”. In: *Journal of Behavioral and Experimental Economics* 101, p. 101942. ISSN: 2214-8043. DOI: <https://doi.org/10.1016/j.socec.2022.101942>.
- Abu-Mostafa, Yaser S, Malik Magdon-Ismail, and Hsuan-Tien Lin (2012). *Learning from data*. Vol. 4. AMLBook New York.
- Acker, Amelia and Adam Kreisberg (2020). “Social media data archives in an API-driven world”. In: *Archival Science* 20.2, pp. 105–123.
- Adam, Tornes (Apr. 2020). *Enabling Study of the Public Conversation in a Time of Crisis*. Accessed: 2025-09-17. URL: https://blog.x.com/developer/en_us/topics/tools/2020/covid19_public_conversation_data.
- Adhikari, Bipin, Phaik Yeong Cheah, and Lorenz von Seidlein (2022). “Trust is the common denominator for COVID-19 vaccine acceptance: A literature review”. In: *Vaccine: X* 12, p. 100213. ISSN: 2590-1362. DOI: <https://doi.org/10.1016/j.jvacx.2022.100213>.
- Agarwal, Oshin and Ani Nenkova (Sept. 2022). “Temporal Effects on Pre-trained Models for Language Processing Tasks”. In: *Transactions of the Association for Computational Linguistics* 10, pp. 904–921. ISSN: 2307-387X. DOI: [10.1162/tacl_a_00497](https://doi.org/10.1162/tacl_a_00497).
- Aghajanyan, Armen, Luke Zettlemoyer, and Sonal Gupta (2020). *Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning*. URL: <https://arxiv.org/abs/2012.13255>.

- Akiba, Takuya et al. (2019). “Optuna: A Next-Generation Hyperparameter Optimization Framework”. In: *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631.
- Alammar, Jay and Maarten Grootendorst (2024). *Hands-on large language models: language understanding and generation*. ” O’Reilly Media, Inc.”
- Alamoodi, A.H. et al. (2021). “Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review”. In: *Expert Systems with Applications* 167, p. 114155. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2020.114155>.
- Alessandretti, Laura (2022). “What human mobility data tell us about COVID-19 spread”. In: *Nature Reviews Physics* 4.1, pp. 12–13.
- Allan, Maya et al. (2022). “The world health organization COVID-19 surveillance database”. In: *International journal for equity in health* 21.Suppl 3, p. 167.
- Alvarez, Elizabeth et al. (2023). “Limitations of COVID-19 testing and case data for evidence-informed health policy and practice”. In: *Health Research Policy and Systems* 21.1, p. 11.
- Alvo, Mayer and Paul Cabilio (1985). “Average rank correlation statistics in the presence of ties”. In: *Communications in Statistics-Theory and Methods* 14.9, pp. 2095–2108.
- Ansari, Abdul Fatir et al. (2024). “Chronos: Learning the Language of Time Series”. In: *Transactions on Machine Learning Research*. Expert Certification. ISSN: 2835-8856.
- Antypas, Dimosthenis et al. (Dec. 2023). “SuperTweetEval: A Challenging, Unified and Heterogeneous Benchmark for Social Media NLP Research”. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 12590–12607. DOI: [10.18653/v1/2023.findings-emnlp.838](https://doi.org/10.18653/v1/2023.findings-emnlp.838).
- Apple Inc. (Apr. 2020). *Apple makes mobility data available to aid COVID-19 efforts*. Accessed: 2025-09-03. URL: <https://www.apple.com/uk/newsroom/2020/04/apple-makes-mobility-data-available-to-aid-covid-19-efforts/>.

- Astley, Christina M. et al. (2021). “Global monitoring of the impact of the COVID-19 pandemic through online surveys sampled from the Facebook user base”. In: *Proceedings of the National Academy of Sciences* 118.51, e2111455118. DOI: [10.1073/pnas.2111455118](https://doi.org/10.1073/pnas.2111455118).
- Atchison, Christina et al. (Jan. 2021). *Early perceptions and behavioural responses during the COVID-19 pandemic: a cross-sectional survey of UK adults*. DOI: [10.1136/bmjopen-2020-043577](https://doi.org/10.1136/bmjopen-2020-043577).
- Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. URL: <https://arxiv.org/abs/1607.06450>.
- Babanejaddehaki, Ghazaleh, Aijun An, and Manos Papagelis (Feb. 2025). “Disease Outbreak Detection and Forecasting: A Review of Methods and Data Sources”. In: *ACM Trans. Comput. Healthcare* 6.2. DOI: [10.1145/3708549](https://doi.org/10.1145/3708549).
- Badker, Rinette et al. (2021). “Challenges in reported COVID-19 data: best practices and recommendations for future epidemics”. In: *BMJ global health* 6.5, e005542.
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2016). *Neural Machine Translation by Jointly Learning to Align and Translate*. URL: <https://arxiv.org/abs/1409.0473>.
- Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun (2018). *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. URL: <https://arxiv.org/abs/1803.01271>.
- Bai, Yuntao et al. (2022). *Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback*. URL: <https://arxiv.org/abs/2204.05862>.
- Balduf, Leonhard et al. (2024). “Looking AT the Blue Skies of Bluesky”. In: *Proceedings of the 2024 ACM on Internet Measurement Conference*. IMC ’24. Madrid, Spain: Association for Computing Machinery, pp. 76–91. ISBN: 9798400705922. DOI: [10.1145/3646547.3688407](https://doi.org/10.1145/3646547.3688407).
- Banda, Juan M. et al. (2021). “A Large-Scale COVID-19 Twitter Chatter Dataset for Open Scientific Research—An International Collaboration”. In: *Epidemiologia* 2.3 (3), pp. 315–324. ISSN: 2331-8422. DOI: [10.3390/epidemiologia2030024](https://doi.org/10.3390/epidemiologia2030024).

- Banholzer, Nicolas et al. (2022). “The methodologies to assess the effectiveness of non-pharmaceutical interventions during COVID-19: a systematic review: N. Banholzer et al.” In: *European Journal of Epidemiology* 37.10, pp. 1003–1024.
- Barbieri, Francesco, Luis Espinosa Anke, and Jose Camacho-Collados (June 2022). “XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond”. In: *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 258–266.
- Bartlett, Peter L and Shahar Mendelson (2002). “Rademacher and gaussian complexities: Risk bounds and structural results”. In: *Journal of machine learning research* 3.Nov, pp. 463–482.
- Batson, C. Daniel and Adam A. Powell (2003). “Altruism and Prosocial Behavior”. In: *Handbook of Psychology*. John Wiley Sons, Ltd. Chap. 19, pp. 463–484. ISBN: 9780471264385. DOI: <https://doi.org/10.1002/0471264385.wei0519>.
- Bavel, Jay J.Van et al. (May 2020). *Using social and behavioural science to support COVID-19 pandemic response*. DOI: [10.1038/s41562-020-0884-z](https://doi.org/10.1038/s41562-020-0884-z).
- Baydin, Atilim Gunes et al. (2018). “Automatic Differentiation in Machine Learning: a Survey”. In: *Journal of Machine Learning Research* 18.153, pp. 1–43.
- Bedson, Jamie et al. (2021). “A review and agenda for integrated disease models including social and behavioural factors”. In: *Nature human behaviour* 5.7, pp. 834–846.
- Bergstra, James et al. (2011). “Algorithms for Hyper-Parameter Optimization”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor et al. Vol. 24. Curran Associates, Inc.
- Bingham, Eli et al. (2019). “Pyro: Deep Universal Probabilistic Programming”. In: *J. Mach. Learn. Res.* 20, 28:1–28:6.
- Bish, Alison and Susan Michie (2010). “Demographic and attitudinal determinants of protective behaviours during a pandemic: A review”. In: *British journal of health psychology* 15.4, pp. 797–824.

- Blakey, Elizabeth (2024). “The Day Data Transparency Died: How Twitter/X Cut Off Access for Social Research”. In: *Contexts* 23.2, pp. 30–35.
- Bluesky Documentation (2025). *Get Started*. Accessed: 2025-08-25. URL: <https://docs.bsky.app/docs/get-started>.
- Bogdanowicz, Alexander and ChengHe Guan (2022). “Dynamic topic modeling of Twitter data during the COVID-19 pandemic”. In: *Plos one* 17.5, e0268669.
- Bojanowski, Piotr et al. (2017). “Enriching Word Vectors with Subword Information”. In: *Transactions of the Association for Computational Linguistics* 5. Ed. by Lillian Lee, Mark Johnson, and Kristina Toutanova, pp. 135–146. DOI: [10.1162/tac1_a_00051](https://doi.org/10.1162/tac1_a_00051).
- Bollerslev, Tim (1986). “Generalized autoregressive conditional heteroskedasticity”. In: *Journal of econometrics* 31.3, pp. 307–327.
- Boon-Itt, Sakun and Yukolpat Skunkan (Oct. 2020). “Public perception of the COVID-19 pandemic on twitter: Sentiment analysis and topic modeling study”. In: *JMIR Public Health and Surveillance* 6. ISSN: 23692960. DOI: [10.2196/21978](https://doi.org/10.2196/21978).
- Bottou, Léon (2010). “Large-scale machine learning with stochastic gradient descent”. In: *Proceedings of COMPSTAT’2010: 19th International Conference on Computational Statistics Paris France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer, pp. 177–186.
- Box, George and GM Jenkins (1976). “Analysis: Forecasting and Control”. In: *San francisco*.
- Bradshaw, Samantha et al. (2020). “Sourcing and Automation of Political News and Information over Social Media in the United States, 2016-2018”. In: *Political Communication* 37.2, pp. 173–193. DOI: [10.1080/10584609.2019.1663322](https://doi.org/10.1080/10584609.2019.1663322).
- Braig, Niklas et al. (2023). “Machine learning techniques for sentiment analysis of COVID-19-related twitter data”. In: *IEEE access* 11, pp. 14778–14803.
- Braun, Nadine, Martijn Goudbeek, and Emiel Krahmer (2022). “Lost in transmission? Self-and other-annotation of emotional words”. In: *Acta Psychologica* 229, p. 103713.

- Brauner, Jan M et al. (2021). “Inferring the effectiveness of government interventions against COVID-19”. In: *Science* 371.6531, eabd9338.
- Breiman, L. et al. (1984). “Classification and Regression Trees”. In: *Biometrics* 40, p. 874.
- Breiman, Leo (1996). “Bagging predictors”. In: *Machine learning* 24.2, pp. 123–140.
- (2001). “Random forests”. In: *Machine learning* 45.1, pp. 5–32.
- Brooks-Pollock, Ellen et al. (Dec. 2023). “Voluntary risk mitigation behaviour can reduce impact of SARS-CoV-2: a real-time modelling study of the January 2022 Omicron wave in England”. In: *BMC Medicine* 21 (1). ISSN: 17417015. DOI: [10.1186/s12916-022-02714-5](https://doi.org/10.1186/s12916-022-02714-5).
- Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*. URL: <https://arxiv.org/abs/2005.14165>.
- Bruin, Marijn de et al. (2022). “Behavioural insights and the evolving COVID-19 pandemic”. In: *Eurosurveillance* 27.18, p. 2100615.
- Bruns, Axel (2021). “After the ‘APIcalypse’: Social media platforms and their fight against critical scholarly research”. In: *Disinformation and data lockdown on social platforms*, pp. 14–36.
- Buscemi, Alessio and Daniele Proverbio (2024). *ChatGPT vs Gemini vs LLaMA on Multilingual Sentiment Analysis*. URL: <https://arxiv.org/abs/2402.01715>.
- Butler, Declan (2013). “When Google got flu wrong: US outbreak foxes a leading web-based method for tracking seasonal flu”. In: *Nature* 494.7436, pp. 155–157.
- Cabezas, Javier et al. (July 2021). “Detecting emotional evolution on twitter during the COVID-19 pandemic using text analysis”. In: *International Journal of Environmental Research and Public Health* 18 (13). ISSN: 16604601. DOI: [10.3390/ijerph18136981](https://doi.org/10.3390/ijerph18136981).
- Camacho-collados, Jose et al. (Dec. 2022). “TweetNLP: Cutting-Edge Natural Language Processing for Social Media”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed.

- by Wanxiang Che and Ekaterina Shutova. Abu Dhabi, UAE: Association for Computational Linguistics, pp. 38–49. DOI: [10.18653/v1/2022.emnlp-demos.5](https://doi.org/10.18653/v1/2022.emnlp-demos.5).
- Cambria, Erik and Bebo White (2014). “Jumping NLP curves: A review of natural language processing research”. In: *IEEE Computational intelligence magazine* 9.2, pp. 48–57.
- Campos-Mercade, Pol et al. (2021). “Prosociality predicts health behaviors during the COVID-19 pandemic”. In: *Journal of public economics* 195, p. 104367.
- Cao, Defu et al. (2020). “Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., pp. 17766–17778.
- Chandrasekaran, Ranganathan et al. (Oct. 2020). “Topics, Trends, and Sentiments of Tweets about the COVID-19 Pandemic: Temporal Inveillance Study”. In: *Journal of Medical Internet Research* 22 (10). ISSN: 14388871. DOI: [10.2196/22624](https://doi.org/10.2196/22624).
- Chang, Serina et al. (2021). “Mobility network models of COVID-19 explain inequities and inform reopening”. In: *Nature* 589.7840, pp. 82–87.
- Chang, Shiyu et al. (2017). “Dilated Recurrent Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc.
- Chen, Si-An et al. (2023). *TSMixer: An All-MLP Architecture for Time Series Forecasting*. URL: <https://arxiv.org/abs/2303.06053>.
- Chen, Tianqi, Emily Fox, and Carlos Guestrin (2014). “Stochastic Gradient Hamiltonian Monte Carlo”. In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Beijing, China: PMLR, pp. 1683–1691.
- Chen, Yirong et al. (2018). “Neighbourhood level real-time forecasting of dengue cases in tropical urban Singapore”. In: *BMC medicine* 16.1, p. 129.
- Cheng, I. Kit et al. (2021). “Evaluation of Twitter data for an emerging crisis: an application to the first wave of COVID-19 in the UK”. In: *Scientific Reports* 11 (1). ISSN: 20452322. DOI: [10.1038/s41598-021-98396-9](https://doi.org/10.1038/s41598-021-98396-9).

- Cheng, Tao, Baoyan Han, and Yunzhe Liu (2023). “Exploring public sentiment and vaccination uptake of COVID-19 vaccines in England: a spatiotemporal and sociodemographic analysis of Twitter data”. In: *Frontiers in Public Health* 11. ISSN: 22962565. DOI: [10.3389/fpubh.2023.1193750](https://doi.org/10.3389/fpubh.2023.1193750).
- Cho, Kyunghyun et al. (Oct. 2014a). “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. DOI: [10.3115/v1/D14-1179](https://doi.org/10.3115/v1/D14-1179).
- Cho, Kyunghyun et al. (Oct. 2014b). “On the Properties of Neural Machine Translation: Encoder–Decoder Approaches”. In: *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Ed. by Dekai Wu et al. Doha, Qatar: Association for Computational Linguistics, pp. 103–111. DOI: [10.3115/v1/W14-4012](https://doi.org/10.3115/v1/W14-4012).
- Choi, Jounghwa and Kyung-Hee Kim (2022). “The Differential Consequences of Fear, Anger, and Depression in Response to COVID-19 in South Korea”. In: *International Journal of Environmental Research and Public Health* 19.11. ISSN: 1660-4601. DOI: [10.3390/ijerph19116723](https://doi.org/10.3390/ijerph19116723).
- Christiano, Paul et al. (2023). *Deep reinforcement learning from human preferences*. URL: <https://arxiv.org/abs/1706.03741>.
- Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2016). *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. URL: <https://arxiv.org/abs/1511.07289>.
- Colvin, Samuel et al. (Sept. 2023). *pydantic/pydantic: V1.10.13 2023-09-27*. Version v1.10.13. DOI: [10.5281/zenodo.8384033](https://doi.org/10.5281/zenodo.8384033).
- Conneau, Alexis and Guillaume Lample (2019). “Cross-lingual Language Model Pre-training”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc.
- Conneau, Alexis et al. (2019). “Unsupervised Cross-lingual Representation Learning at Scale”. In: *CoRR* abs/1911.02116.

- Conneau, Alexis et al. (July 2020). “Unsupervised Cross-lingual Representation Learning at Scale”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 8440–8451.
- Corpuz, Jeff Clyde G (2021). “Adapting to the culture of ‘new normal’: an emerging response to COVID-19”. In: *Journal of public health* 43.2, e344–e345.
- Cramer, Estee Y et al. (2022). “The United States COVID-19 forecast hub dataset”. In: *Scientific data* 9.1, p. 462.
- Daly, Mary et al. (2020). “Oxford supertracker: The global directory for COVID policy trackers and surveys”. In: *Department of Social Policy and Intervention*.
- Dan, Shozen et al. (2025). “Addressing survey fatigue bias in longitudinal social contact studies to improve pandemic preparedness”. In: *Scientific Reports* 15.1, p. 17935.
- Das, Abhimanyu et al. (2024). *Long-term Forecasting with TiDE: Time-series Dense Encoder*. URL: <https://arxiv.org/abs/2304.08424>.
- Dauphin, Yann N et al. (2017). “Language modeling with gated convolutional networks”. In: *International conference on machine learning*. PMLR, pp. 933–941.
- Davies, Rachel et al. (2022). “A systematic review of observational methods used to quantify personal protective behaviours among members of the public during the COVID-19 pandemic, and the concordance between observational and self-report measures in infectious disease health protection”. In: *BMC Public Health* 22.1, p. 1436.
- Delogu, Franco et al. (2025). “The impact of the COVID-19 pandemic on the use of social media: A cross-national comparison”. In: *Acta Psychologica* 255, p. 104888. ISSN: 0001-6918. DOI: <https://doi.org/10.1016/j.actpsy.2025.104888>.
- DeLucia, Alexandra et al. (Dec. 2022). “Bernice: A Multilingual Pre-trained Encoder for Twitter”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 6191–6205.

- Deng, Yang et al. (2024). “Trending on the use of Google mobility data in COVID-19 mathematical models”. In: *Advances in Continuous and Discrete Models* 2024.1, p. 21.
- Dettmers, Tim et al. (2023). *QLoRA: Efficient Finetuning of Quantized LLMs*. URL: <https://arxiv.org/abs/2305.14314>.
- Devine, Daniel et al. (2021). “Trust and the coronavirus pandemic: What are the consequences of and for trust? An early review of the literature”. In: *Political Studies Review* 19.2, pp. 274–285.
- Devlin, Jacob et al. (2019). “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186.
- Disease Prevention, European Centre for and Control (2025). *Download COVID-19 data*. Accessed: 14 September 2025. URL: <https://www.ecdc.europa.eu/en/covid-19/data>.
- Dong, Ensheng, Hongru Du, and Lauren Gardner (2020). “An interactive web-based dashboard to track COVID-19 in real time”. In: *The Lancet infectious diseases* 20.5, pp. 533–534.
- Doogan, Caitlin et al. (Sept. 2020). “Public Perceptions and Attitudes Toward COVID-19 Nonpharmaceutical Interventions Across Six Countries: A Topic Modeling Analysis of Twitter Data”. In: *J Med Internet Res* 22.9, e21419. ISSN: 1438-8871. DOI: [10.2196/21419](https://doi.org/10.2196/21419).
- Downs, Anthony (1972). “Up and Down with Ecology-the Issue-Attention Cycle”. In: *The Public Interest* 28, p. 38.
- Du, Hongru et al. (2023). “Incorporating variant frequencies data into short-term forecasting for COVID-19 cases and deaths in the USA: a deep learning approach”. In: *Ebiomedicine* 89.
- Effenberger, Maria et al. (June 2020). *Association of the COVID-19 pandemic with Internet Search Volumes: A Google Trends™ Analysis*. DOI: [10.1016/j.ijid.2020.04.033](https://doi.org/10.1016/j.ijid.2020.04.033).

- Egleston, Brian L, Suzanne M Miller, and Neal J Meropol (2011). “The impact of misclassification due to survey response fatigue on estimation and identifiability of treatment effects”. In: *Statistics in medicine* 30.30, pp. 3560–3572.
- Elman, Jeffrey L (1990). “Finding structure in time”. In: *Cognitive science* 14.2, pp. 179–211.
- Elmitwalli, Sherif et al. (2024). “Enhancing sentiment and intent analysis in public health via fine-tuned Large Language Models on tobacco and e-cigarette-related tweets”. In: *Frontiers in Big Data* 7, p. 1501154.
- Engle, Robert F (1982). “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation”. In: *Econometrica: Journal of the econometric society*, pp. 987–1007.
- Epstein, Joshua M, Erez Hatna, and Jennifer Crodelle (2021). “Triple contagion: a two-fears epidemic model”. In: *Journal of the Royal Society Interface* 18.181, p. 20210186.
- Erhardt, Julian et al. (2021). “The emotional foundations of political support: How fear and anger affect trust in the government in times of the Covid-19 pandemic”. In: *Swiss political science review* 27.2, pp. 339–352.
- Espinosa, Laura and Marcel Salathé (Oct. 2024). “Use of large language models as a scalable approach to understanding public health discourse”. In: *PLOS Digital Health* 3.10, pp. 1–15. DOI: [10.1371/journal.pdig.0000631](https://doi.org/10.1371/journal.pdig.0000631).
- European Centre for Disease Prevention and Control (June 2022a). *How ECDC Collects and Processes COVID-19 Data*. Page last updated June 23, 2022; accessed July 15, 2025. URL: <https://www.ecdc.europa.eu/en/covid-19/data-collection>.
- (2022b). *Protocol for Reporting Surveillance of Respiratory Viruses*. Accessed: 2025-07-15. URL: <https://www.ecdc.europa.eu/sites/default/files/documents/respiratory-virus-surveillance-reporting-protocol.pdf>.
- European Centre for Disease Prevention and Control (ECDC) (Apr. 22, 2021). *Forecasting COVID-19 cases and deaths in Europe – new hub will support European pandemic planning*. Accessed: 2025-08-13. URL: <https://www.ecdc.europa.eu/en/news-events/forecasting-covid-19-cases-and-deaths-europe-new-hub>.

- Fang, Ximeng et al. (2022). “Prosociality predicts individual behavior and collective outcomes in the COVID-19 pandemic”. In: *Social Science & Medicine* 308, p. 115192.
- Feezell, Jessica T. (June 2018). “Agenda Setting through Social Media: The Importance of Incidental News Exposure and Social Filtering in the Digital Era”. In: *Political Research Quarterly* 71 (2), pp. 482–494. ISSN: 1938274X. DOI: [10.1177/1065912917744895](https://doi.org/10.1177/1065912917744895).
- Fellbaum, Christiane (1998). *WordNet: An electronic lexical database*. MIT press.
- Ferguson, Neil (2007). “Capturing human behaviour”. In: *Nature* 446.7137, pp. 733–733.
- Filip, Tomáš, Martin Pavlíček, and Petr Sosík (2024). *Fine-tuning multilingual language models in Twitter/X sentiment analysis: a study on Eastern-European V4 languages*. URL: <https://arxiv.org/abs/2408.02044>.
- Flaxman, Seth, Sharad Goel, and Justin M. Rao (Mar. 2016). “Filter Bubbles, Echo Chambers, and Online News Consumption”. In: *Public Opinion Quarterly* 80.S1, pp. 298–320. ISSN: 0033-362X. DOI: [10.1093/poq/nfw006](https://doi.org/10.1093/poq/nfw006).
- Flaxman, Seth et al. (2020). “Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe”. In: *Nature* 584.7820, pp. 257–261.
- Forgas, Joseph P (1995). “Mood and judgment: the affect infusion model (AIM).” In: *Psychological bulletin* 117.1, p. 39.
- Freelon, Deen (2018). “Computational research in the post-API age”. In: *Political Communication* 35.4, pp. 665–668.
- Freund, Yoav (1995). “Boosting a weak learning algorithm by majority”. In: *Information and computation* 121.2, pp. 256–285.
- Freund, Yoav, Robert E Schapire, et al. (1996). “Experiments with a new boosting algorithm”. In: *icml*. Vol. 96. Bari, Italy, pp. 148–156.
- Friedman, Jerome H (2001). “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics*, pp. 1189–1232.

- Funk, Sebastian, Marcel Salathé, and Vincent AA Jansen (2010). “Modelling the influence of human behaviour on the spread of infectious diseases: a review”. In: *Journal of the Royal Society Interface* 7.50, pp. 1247–1256.
- Funk, Sebastian et al. (2015). “Nine challenges in incorporating the dynamics of behaviour in infectious diseases models”. In: *Epidemics* 10. Challenges in Modelling Infectious Disease Dynamics, pp. 21–25. ISSN: 1755-4365. DOI: <https://doi.org/10.1016/j.epidem.2014.09.005>.
- Gal, Yarin and Zoubin Ghahramani (2016). “A theoretically grounded application of dropout in recurrent neural networks”. In: *Advances in neural information processing systems* 29.
- Garza, Federico et al. (2022). *StatsForecast: Lightning fast forecasting with statistical and econometric models*. PyCon Salt Lake City, Utah, US 2022. URL: <https://github.com/Nixtla/statsforecast>.
- Geographical Names, United Kingdom Permanent Committee on (2024). *Toponymic Guidelines for Map and Other Editors, United Kingdom of Great Britain and Northern Ireland*. UK Government website. First edition February 2009; updated May 2017; most recent revision May 2024. URL: <https://www.gov.uk/government/publications/toponymic-guidelines/toponymic-guidelines-for-map-and-other-editors-united-kingdom-of-great-britain-and-northern-ireland--2>.
- Gibbs, Hamish et al. (July 2021). “Detecting behavioural changes in human movement to inform the spatial scale of interventions against COVID-19”. In: *PLoS Computational Biology* 17 (7). ISSN: 15537358. DOI: [10.1371/journal.pcbi.1009162](https://doi.org/10.1371/journal.pcbi.1009162).
- Gimma, Amy et al. (Mar. 2022). “Changes in social contacts in England during the COVID-19 pandemic between March 2020 and March 2021 as measured by the CoMix survey: A repeated cross-sectional study”. In: *PLoS Medicine* 19 (3). ISSN: 15491676. DOI: [10.1371/journal.pmed.1003907](https://doi.org/10.1371/journal.pmed.1003907).
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep learning*. MIT press.

- Google LLC (n.d.). *Google COVID-19 Community Mobility Reports*. Accessed: 2025-09-03. URL: <https://www.google.com/covid19/mobility/>.
- Gozzi, Nicolò, Nicola Perra, and Alessandro Vespignani (2025). “Comparative evaluation of behavioral epidemic models using COVID-19 data”. In: *Proceedings of the National Academy of Sciences* 122.24, e2421993122.
- Granger, Clive WJ (1969). “Investigating causal relations by econometric models and cross-spectral methods”. In: *Econometrica: journal of the Econometric Society*, pp. 424–438.
- Grantz, Kyra H et al. (2020). “The use of mobile phone data to inform analysis of COVID-19 pandemic epidemiology”. In: *Nature communications* 11.1, p. 4961.
- Graso, Maja (2022). “The new normal: Covid-19 risk perceptions and support for continuing restrictions past vaccinations”. In: *PloS one* 17.4, e0266602.
- Grassly, Nicholas C and Christophe Fraser (2008). “Mathematical models of infectious disease transmission”. In: *Nature Reviews Microbiology* 6.6, pp. 477–487.
- Grattafiori, Aaron et al. (2024). *The Llama 3 Herd of Models*. URL: <https://arxiv.org/abs/2407.21783>.
- Gruver, Nate et al. (2023). “Large language models are zero-shot time series forecasters”. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS ’23. New Orleans, LA, USA: Curran Associates Inc.
- Gu, Albert and Tri Dao (2024). *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*. URL: <https://arxiv.org/abs/2312.00752>.
- Gupta, Neha and Rashmi Agrawal (2020). “Application and techniques of opinion mining”. In: *Hybrid computational intelligence*. Elsevier, pp. 1–23.
- Haddaoui, Bousselham El et al. (June 2023). “Sentiment analysis in SemEval: a review of sentiment identification approaches”. In: *International Journal of Electrical and Computer Engineering* 13 (3), pp. 3322–3338. ISSN: 20888708. DOI: [10.11591/ijece.v13i3.pp3322-3338](https://doi.org/10.11591/ijece.v13i3.pp3322-3338).
- Haktanir, Abdulkadir et al. (2022). “Do we experience pandemic fatigue? current state, predictors, and prevention”. In: *Current Psychology* 41.10, pp. 7314–7325.

- Hale, Thomas et al. (2021). “A global panel database of pandemic policies (Oxford COVID-19 Government Response Tracker)”. eng. In: *Nature human behaviour* 5.4, pp. 529–538. ISSN: 2397-3374.
- Hale, Thomas et al. (June 2023). *Variation in Government Responses to COVID-19*. Tech. rep. BSG-WP-2020/032. Version 15. Accessed: 2025-07-18. Blavatnik School of Government, University of Oxford.
- Hamilton, James D (2020). *Time series analysis*. Princeton university press.
- Hasell, Joe et al. (2020). “A cross-country database of COVID-19 testing”. In: *Scientific data* 7.1, p. 345.
- Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.
- He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*. URL: <https://arxiv.org/abs/1512.03385>.
- Henkel, Luca et al. (Feb. 2023). “The association between vaccination status identification and societal polarization”. In: *Nature Human Behaviour* 7 (2), pp. 231–239. ISSN: 23973374. DOI: [10.1038/s41562-022-01469-6](https://doi.org/10.1038/s41562-022-01469-6).
- Hinton, Geoffrey E. et al. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. URL: <https://arxiv.org/abs/1207.0580>.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780.
- Hodcroft, Emma B. (2021). *CoVariants: SARS-CoV-2 Mutations and Variants of Interest*. URL: <https://covariants.org/>.
- Hu, Edward J. et al. (2021a). *LoRA: Low-Rank Adaptation of Large Language Models*. URL: <https://arxiv.org/abs/2106.09685>.
- Hu, Tao et al. (2021b). “Human mobility data in the COVID-19 pandemic: characteristics, applications, and challenges”. In: *International Journal of Digital Earth* 14.9, pp. 1126–1147.

- Huang, Binxuan and Kathleen M. Carley (2020). “A large-scale empirical study of geotagging behavior on Twitter”. In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ASONAM '19. Vancouver, British Columbia, Canada: Association for Computing Machinery, pp. 365–373. ISBN: 9781450368681. DOI: [10.1145/3341161.3342870](https://doi.org/10.1145/3341161.3342870).
- Huang, Gao et al. (2018). *Densely Connected Convolutional Networks*. URL: <https://arxiv.org/abs/1608.06993>.
- Hutto, Clayton and Eric Gilbert (2014). “Vader: A parsimonious rule-based model for sentiment analysis of social media text”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 8, pp. 216–225.
- Hyndman, Rob J and George Athanasopoulos (2018). *Forecasting: principles and practice*. OTexts.
- Ij, H (2018). “Statistics versus machine learning”. In: *Nat Methods* 15.4, p. 233.
- Ilin, Cornelia et al. (2021a). “Public mobility data enables COVID-19 forecasting and management at local and global scales”. In: *Scientific reports* 11.1, p. 13531.
- Ilin, Cornelia et al. (Dec. 2021b). “Public mobility data enables COVID-19 forecasting and management at local and global scales”. In: *Scientific Reports* 11 (1). ISSN: 20452322. DOI: [10.1038/s41598-021-92892-8](https://doi.org/10.1038/s41598-021-92892-8).
- Imai, Mutsumi, Junko Kanero, and Takahiko Masuda (2016). “The relation between language, culture, and thought”. In: *Current Opinion in Psychology* 8. Culture, pp. 70–77. ISSN: 2352-250X. DOI: <https://doi.org/10.1016/j.copsyc.2015.10.011>.
- Ipsos and Axios (n.d.). *Axios-Ipsos Coronavirus Index*. Accessed: 2025-09-03. URL: <https://www.ipsos.com/en-us/news-polls/axios-ipsos-coronavirus-index>.
- Ivakhnenko, A. G. (1971). “Polynomial Theory of Complex Systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-1.4, pp. 364–378. DOI: [10.1109/TSMC.1971.4308320](https://doi.org/10.1109/TSMC.1971.4308320).

- James, Gareth et al. (2021). *An Introduction to Statistical Learning: With Applications in R*. 2nd. Berlin: Springer. DOI: [10.1007/978-1-0716-1418-1](https://doi.org/10.1007/978-1-0716-1418-1).
- Jeong, Ujun et al. (2024). “Descriptor: A Temporal Multi-network Dataset of Social Interactions in Bluesky Social (BlueTempNet)”. In: *IEEE Data Descriptions* 1, pp. 71–79. DOI: [10.1109/IEEEDATA.2024.3474640](https://doi.org/10.1109/IEEEDATA.2024.3474640).
- Jiang, Albert Q. et al. (2023). *Mistral 7B*. URL: <https://arxiv.org/abs/2310.06825>.
- Jiang, Albert Q. et al. (2024). *Mixtral of Experts*. URL: <https://arxiv.org/abs/2401.04088>.
- Jin, KX and Laura McGorman (Apr. 2020). *Data for Good: New Tools to Help Health Researchers Track and Combat COVID-19*. Accessed: YYYY-MM-DD. URL: <https://about.fb.com/news/2020/04/data-for-good/>.
- Jin, Ming et al. (2024). *Time-LLM: Time Series Forecasting by Reprogramming Large Language Models*. URL: <https://arxiv.org/abs/2310.01728>.
- Jirsa, Viktor K et al. (2022). “Integrating psychosocial variables and societal diversity in epidemic models for predicting COVID-19 transmission dynamics”. In: *PLOS Digital Health* 1.8, e0000098.
- Jones, Sarah P (2020). “Imperial college London big data analytical unit and yougov plc. 2020”. In: *Imperial College London YouGov Covid Data Hub, v1. 0, YouGov Plc*.
- Jordan, Jillian J, Erez Yoeli, and David G Rand (2021). “Don’t get it or don’t spread it: Comparing self-interested versus prosocial motivations for COVID-19 prevention behaviors”. In: *Scientific reports* 11.1, p. 20222.
- Jordan, Michael I (1986). “Attractor dynamics and parallelism in a connectionist sequential machine”. In: *Proceedings of the Annual Meeting of the Cognitive Science Society*. Vol. 8.
- Kamocki, Pawel et al. (2023). *Legal status and regulation of the German language in the Federal Republic of Germany*. DOI: [10.1080/14664208.2023.2296787](https://doi.org/10.1080/14664208.2023.2296787). URL: <https://doi.org/10.1080/14664208.2023.2296787>.

- Kanayama, Hiroshi and Ran Iwamoto (May 2020). “How Universal are Universal Dependencies? Exploiting Syntax for Multilingual Clause-level Sentiment Detection”. eng. In: *Proceedings of the Twelfth Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4063–4073. ISBN: 979-10-95546-34-4.
- Kanayama, Hiroshi et al. (Aug. 2024). “Incorporating Syntax and Lexical Knowledge to Multilingual Sentiment Classification on Large Language Models”. In: *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, pp. 4810–4817. DOI: [10.18653/v1/2024.findings-acl.286](https://doi.org/10.18653/v1/2024.findings-acl.286).
- Karhu, Mari, Mari Suoheimo, and Jonna Häkkinä (2021). “People’s perspectives on social media use during COVID-19 pandemic”. In: *Proceedings of the 20th International Conference on Mobile and Ubiquitous Multimedia*, pp. 123–130.
- Ke, Guolin et al. (2017). “Lightgbm: A highly efficient gradient boosting decision tree”. In: *Advances in neural information processing systems* 30.
- Keeling, Matt J et al. (2022). “Comparison of the 2021 COVID-19 roadmap projections against public health data in England”. In: *Nature communications* 13.1, p. 4924.
- Khare, Shruti et al. (2021). “GISAID’s Role in Pandemic Response”. In: *China CDC Weekly* 3.49, pp. 1049–1051. DOI: [10.46234/ccdcw2021.255](https://doi.org/10.46234/ccdcw2021.255).
- Kim, Jongseon et al. (2025). “A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges”. In: *Artificial Intelligence Review* 58.7, pp. 1–95.
- Kim, Taesung et al. (2021). “Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift”. In: *International Conference on Learning Representations*.
- Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. URL: <https://arxiv.org/abs/1412.6980>.
- Koenker, Roger and Gilbert Bassett Jr (1978). “Regression quantiles”. In: *Econometrica: journal of the Econometric Society*, pp. 33–50.

- Kostandova, Natalya et al. (2024). “A systematic review of using population-level human mobility data to understand SARS-CoV-2 transmission”. In: *Nature communications* 15.1, p. 10504.
- Kretzschmar, Mirjam E et al. (2022). “Challenges for modelling interventions for future pandemics”. In: *Epidemics* 38, p. 100546.
- Kreuter, Frauke et al. (2020). “Partnering with a global platform to inform research and public policy making: What needs to be in place to make a global COVID-19 survey work?” In: *Survey Research Methods*. Vol. 14. 2, pp. 159–163.
- Kwak, Haewoon et al. (2010). “What is Twitter, a social network or a news media?” In: *Proceedings of the 19th International Conference on World Wide Web*. WWW '10. Raleigh, North Carolina, USA: Association for Computing Machinery, pp. 591–600. ISBN: 9781605587998. DOI: [10.1145/1772690.1772751](https://doi.org/10.1145/1772690.1772751).
- Lamsal, Rabindra (2021). “Design and analysis of a large-scale COVID-19 tweets dataset”. In: *Applied Intelligence* 51.5, pp. 2790–2804.
- Lamsal, Rabindra, Aaron Harwood, and Maria Rodriguez Read (Nov. 2022). “Twitter conversations predict the daily confirmed COVID-19 cases”. In: *Applied Soft Computing* 129. ISSN: 15684946. DOI: [10.1016/j.asoc.2022.109603](https://doi.org/10.1016/j.asoc.2022.109603).
- Lazarus, Jeffrey V. et al. (Feb. 2023). “A survey of COVID-19 vaccine acceptance across 23 countries in 2022”. In: *Nature Medicine* 29 (2), pp. 366–375. ISSN: 1546170X. DOI: [10.1038/s41591-022-02185-4](https://doi.org/10.1038/s41591-022-02185-4).
- Lazer, David et al. (2014). “The parable of Google Flu: traps in big data analysis”. In: *science* 343.6176, pp. 1203–1205.
- Lecun, Y. et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep learning”. In: *nature* 521.7553, pp. 436–444.
- Lee, Anton et al. (Apr. 2025). “Kolmogorov-Arnold Networks Still Catastrophically Forget but Differently from MLP”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 39.17, pp. 18053–18061. DOI: [10.1609/aaai.v39i17.33986](https://doi.org/10.1609/aaai.v39i17.33986).

- Lerner, Jennifer S and Dacher Keltner (2000). “Beyond valence: Toward a model of emotion-specific influences on judgement and choice”. In: *Cognition & emotion* 14.4, pp. 473–493.
- Leung, Xing Yu et al. (2023). “A systematic review of dengue outbreak prediction models: Current scenario and future directions”. In: *PLoS neglected tropical diseases* 17.2, e0010631.
- Lewis, Mike et al. (July 2020). “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, pp. 7871–7880. DOI: [10.18653/v1/2020.acl-main.703](https://doi.org/10.18653/v1/2020.acl-main.703).
- Li, Chunyuan et al. (2018). *Measuring the Intrinsic Dimension of Objective Landscapes*. URL: <https://arxiv.org/abs/1804.08838>.
- Li, Michael et al. (2023). “Predicting the daily counts of COVID-19 infection using temporal convolutional networks”. In: *Journal of Global Health* 13, p. 03029.
- Li, Wenxiang and K. L. Eddie Law (2024). “Deep Learning Models for Time Series Forecasting: A Review”. In: *IEEE Access* 12, pp. 92306–92327. DOI: [10.1109/ACCESS.2024.3422528](https://doi.org/10.1109/ACCESS.2024.3422528).
- Liaw, Richard et al. (2018). “Tune: A Research Platform for Distributed Model Selection and Training”. In: *arXiv preprint arXiv:1807.05118*.
- Lim, Ah-Young et al. (2023). “A systematic review of the data, methods and environmental covariates used to map Aedes-borne arbovirus transmission risk”. In: *BMC infectious diseases* 23.1, p. 708.
- Lim, Bryan et al. (2021). “Temporal Fusion Transformers for interpretable multi-horizon time series forecasting”. In: *International Journal of Forecasting* 37.4, pp. 1748–1764. ISSN: 0169-2070. DOI: <https://doi.org/10.1016/j.ijforecast.2021.03.012>.
- Lison, Adrian et al. (2023). “Effectiveness assessment of non-pharmaceutical interventions: lessons learned from the COVID-19 pandemic”. In: *The Lancet Public Health* 8.4, e311–e317.

- Liu, Carol Y et al. (2021). “Rapid review of social contact patterns during the COVID-19 pandemic”. In: *Epidemiology* 32.6, pp. 781–791.
- Liu, Jia, Yasir Shahab, and Hafiz Hoque (2022). “Government response measures and public trust during the COVID-19 pandemic: evidence from around the world”. In: *British Journal of Management* 33.2, pp. 571–602.
- Liu, Yinhan et al. (2019). “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692*.
- Liu, Zewen et al. (2024a). “A Review of Graph Neural Networks in Epidemic Modeling”. In: *CoRR* abs/2403.19852.
- Liu, Zewen et al. (2024b). “EpiLearn: A Python Library for Machine Learning in Epidemic Modeling”. In: *epiDAMIK 2024: The 7th International Workshop on Epidemiology meets Data Mining and Knowledge Discovery at KDD 2024*.
- Liu, Ziming et al. (2025). *KAN: Kolmogorov-Arnold Networks*. URL: <https://arxiv.org/abs/2404.19756>.
- Loureiro, Daniel et al. (May 2022). “TimeLMs: Diachronic Language Models from Twitter”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Dublin, Ireland: Association for Computational Linguistics, pp. 251–260. DOI: [10.18653/v1/2022.acl-demo.25](https://doi.org/10.18653/v1/2022.acl-demo.25).
- Lundberg, Scott M and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30.
- Lwin, May Oo et al. (Apr. 2020). “Global sentiments surrounding the COVID-19 pandemic on Twitter: Analysis of Twitter trends”. In: *JMIR Public Health and Surveillance* 6 (2). ISSN: 23692960. DOI: [10.2196/19447](https://doi.org/10.2196/19447).
- Lyu, Joanne Chen, Eileen Le Han, and Garving K Luli (June 2021). “COVID-19 Vaccine-Related Discussion on Twitter: Topic Modeling and Sentiment Analysis”. In: *J Med Internet Res* 23.6, e24435. ISSN: 1438-8871. DOI: [10.2196/24435](https://doi.org/10.2196/24435).
- Machado Júnior, Celso et al. (2023). “Volatility of the COVID-19 vaccine hesitancy: sentiment analysis conducted in Brazil”. In: *Frontiers in Public Health* 11, p. 1192155.

- Mahase, Elisabeth (June 2021). “Covid-19: UK has highest vaccine confidence and Japan and South Korea the lowest, survey finds”. In: *BMJ (Clinical research ed.)* 373, n1439. ISSN: 17561833. DOI: [10.1136/bmj.n1439](https://doi.org/10.1136/bmj.n1439).
- Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos (2018). “Statistical and Machine Learning forecasting methods: Concerns and ways forward”. In: *PloS one* 13.3, e0194889.
- (2020). “The M4 Competition: 100,000 time series and 61 forecasting methods”. In: *International Journal of Forecasting* 36.1. M4 Competition, pp. 54–74. ISSN: 0169-2070. DOI: <https://doi.org/10.1016/j.ijforecast.2019.04.014>.
- (2022). “M5 accuracy competition: Results, findings, and conclusions”. In: *International Journal of Forecasting* 38.4. Special Issue: M5 competition, pp. 1346–1364. ISSN: 0169-2070. DOI: <https://doi.org/10.1016/j.ijforecast.2021.11.013>.
- Margraf, Jürgen, Julia Brailovskaia, and Silvia Schneider (2021). “Adherence to behavioral Covid-19 mitigation measures strongly predicts mortality”. In: *PLoS One* 16.3, e0249392.
- Mathieu, Edouard et al. (2020a). “Coronavirus (COVID-19) Vaccinations”. In: *Our World in Data*.
- Mathieu, Edouard et al. (2020b). “Coronavirus Pandemic (COVID-19)”. In: *Our World in Data*.
- Mathieu, Edouard et al. (2020c). “COVID-19 Pandemic”. In: *Our World in Data*. <https://ourworldindata.org/coronavirus>.
- Mathieu, Edouard et al. (2021). “A global database of COVID-19 vaccinations”. In: *Nature human behaviour* 5.7, pp. 947–953.
- McAllester, David A (1998). “Some pac-bayesian theorems”. In: *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 230–234.
- McCombs, Maxwell E and Donald L Shaw (1972). “The agenda-setting function of mass media”. In: *Public opinion quarterly* 36.2, pp. 176–187.
- McHugh, Mary L (2012). “Interrater reliability: the kappa statistic”. In: *Biochemia medica* 22.3, pp. 276–282.

- Meta (Facebook) Data for Good (n.d.). *Movement Range Maps (COVID-19 Mobility Data)*. Accessed: 2025-09-03. URL: <https://dataforgood.facebook.com/dfg/tools/movement-range-maps/#accessdata>.
- Mikolov, Tomas et al. (2013). *Efficient Estimation of Word Representations in Vector Space*. URL: <https://arxiv.org/abs/1301.3781>.
- Mossong, Joël et al. (2008). “Social contacts and mixing patterns relevant to the spread of infectious diseases”. In: *PLoS medicine* 5.3, e74.
- Msemburi, William et al. (2023). “The WHO estimates of excess mortality associated with the COVID-19 pandemic”. In: *Nature* 613.7942, pp. 130–137.
- Mula, Silvana et al. (Sept. 2021). “The Mediation Role of Desire for Cultural Tightness on Concern With COVID-19 and Perceived Self-Control”. In: *Frontiers in Psychology* 12. ISSN: 16641078. DOI: [10.3389/fpsyg.2021.713952](https://doi.org/10.3389/fpsyg.2021.713952).
- Müller, K-R et al. (1997). “Predicting time series with support vector machines”. In: *International conference on artificial neural networks*. Springer, pp. 999–1004.
- Murray, Lawrence M and Thomas B Schön (2018). “Automated learning with a probabilistic programming language: Birch”. In: *Annual Reviews in Control* 46, pp. 29–43.
- Nash, Rebecca K, Pierre Nouvellet, and Anne Cori (2022). “Real-time estimation of the epidemic reproduction number: Scoping review of the applications and challenges”. In: *PLOS Digital Health* 1.6, e0000052.
- Nazir, Ambreen et al. (2020). “Issues and challenges of aspect-based sentiment analysis: A comprehensive survey”. In: *IEEE Transactions on Affective Computing* 13.2, pp. 845–863.
- Ng, Jing Han and Eng King Tan (Feb. 2023). *COVID-19 vaccination and cultural tightness*. DOI: [10.1017/S0033291721001823](https://doi.org/10.1017/S0033291721001823).
- Nguyen, Dat Quoc, Thanh Vu, and Anh Tuan Nguyen (Oct. 2020). “BERTweet: A pre-trained language model for English Tweets”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, pp. 9–14.

- Nie, Yuqi et al. (2023). *A Time Series is Worth 64 Words: Long-term Forecasting with Transformers*. URL: <https://arxiv.org/abs/2211.14730>.
- Nivette, Amy et al. (2021). “Non-compliance with COVID-19-related public health measures among young adults in Switzerland: Insights from a longitudinal cohort study”. In: *Social science & medicine* 268, p. 113370.
- Nixtla (2025). *mlforecast: Scalable machine learning for time series forecasting*. Accessed: 2025-09-17. URL: <https://github.com/Nixtla/mlforecast>.
- Nouvellet, Pierre et al. (2021). “Reduction in mobility and COVID-19 transmission”. In: *Nature communications* 12.1, p. 1090.
- Öhman, Emily (Dec. 2021). “The Validity of Lexicon-based Sentiment Analysis in Interdisciplinary Research”. In: *Proceedings of the Workshop on Natural Language Processing for Digital Humanities*. Ed. by Mika Härmäläinen et al. NIT Silchar, India: NLP Association of India (NLP AI), pp. 7–12.
- Olivares, Kin G. et al. (2022). *NeuralForecast: User friendly state-of-the-art neural forecasting models*. PyCon Salt Lake City, Utah, US 2022. URL: <https://github.com/Nixtla/neuralforecast>.
- Olivares, Kin G. et al. (Apr. 2023). “Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx”. In: *International Journal of Forecasting* 39.2, pp. 884–900. ISSN: 0169-2070. DOI: [10.1016/j.ijforecast.2022.03.001](https://doi.org/10.1016/j.ijforecast.2022.03.001).
- Oord, Aaron van den et al. (2016). *WaveNet: A Generative Model for Raw Audio*. URL: <https://arxiv.org/abs/1609.03499>.
- OpenAI (2022). *GPT-3.5, a Language Model by OpenAI*.
- Orabi, Mariam et al. (2020). “Detection of Bots in Social Media: A Systematic Review”. In: *Information Processing & Management* 57.4, p. 102250. ISSN: 0306-4573. DOI: <https://doi.org/10.1016/j.ipm.2020.102250>.
- Oreshkin, Boris N. et al. (2020). *N-BEATS: Neural basis expansion analysis for interpretable time series forecasting*. URL: <https://arxiv.org/abs/1905.10437>.

- Organization, World Health (2025). *WHO Coronavirus (COVID-19) Dashboard Data*. Accessed: 14 September 2025. URL: <https://data.who.int/dashboards/covid19/data>.
- Ortiz-Ospina, Esteban (2019). “The rise of social media”. In: *Our World in Data*. <https://ourworldindata.org/rise-of-social-media>.
- Ouyang, Long et al. (2022). *Training language models to follow instructions with human feedback*. URL: <https://arxiv.org/abs/2203.02155>.
- Owuor, Innocensia and Hartwig H. Hochmair (Sept. 2023). “Temporal Relationship between Daily Reports of COVID-19 Infections and Related GDELT and Tweet Mentions”. In: *Geographies* 3 (3), pp. 584–609. ISSN: 26737086. DOI: [10.3390/geographies3030031](https://doi.org/10.3390/geographies3030031).
- Pang, Bo, Lillian Lee, et al. (2008). “Opinion mining and sentiment analysis”. In: *Foundations and Trends® in information retrieval* 2.1–2, pp. 1–135.
- Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.
- Pennington, Jeffrey, Richard Socher, and Christopher Manning (Oct. 2014). “GloVe: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 1532–1543. DOI: [10.3115/v1/D14-1162](https://doi.org/10.3115/v1/D14-1162).
- Perez-Guzman, Pablo N et al. (2023). “Epidemiological drivers of transmissibility and severity of SARS-CoV-2 in England”. In: *Nature Communications* 14.1, p. 4279.
- Petherick, Anna et al. (2021). “A worldwide assessment of changes in adherence to COVID-19 protective behaviours and hypothesized pandemic fatigue”. In: *Nature Human Behaviour* 5.9, pp. 1145–1160.
- Pfizer Inc. (Nov. 9, 2020). *Pfizer and BioNTech Announce Vaccine Candidate Against COVID-19 Achieved Success in First Interim Analysis from Phase 3 Study*. Pfizer Inc. URL: <https://www.pfizer.com/news/press-release/press-release-detail/pfizer-and-biontech-announce-vaccine-candidate-against> (visited on 11/09/2020).

- Phan, Du, Neeraj Pradhan, and Martin Jankowiak (2019). *Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro*. URL: <https://arxiv.org/abs/1912.11554>.
- Pilipiec, Patrick, Isak Samsten, and András Bota (Feb. 2023). “Surveillance of communicable diseases using social media: A systematic review”. In: *PLoS ONE* 18 (2 February). ISSN: 19326203. DOI: [10.1371/journal.pone.0282101](https://doi.org/10.1371/journal.pone.0282101).
- Poudel, Amrit and Tim Weninger (2024). “Navigating the post-api dilemma”. In: *Proceedings of the ACM Web Conference 2024*, pp. 2476–2484.
- Praveen, R. Gnana, Patrick Cardinal, and Eric Granger (2023). “Audio–Visual Fusion for Emotion Recognition in the Valence–Arousal Space Using Joint Cross-Attention”. In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 5.3, pp. 360–373. DOI: [10.1109/TBIOM.2022.3233083](https://doi.org/10.1109/TBIOM.2022.3233083).
- Prokhorenkova, Anna et al. (2018). *CatBoost: unbiased boosting with categorical features*. Accessed: 2025-09-17. URL: <https://github.com/catboost/catboost>.
- Prokhorenkova, Liudmila et al. (2019). *CatBoost: unbiased boosting with categorical features*. URL: <https://arxiv.org/abs/1706.09516>.
- Qazi, Umair, Muhammad Imran, and Ferda Ofli (July 2020). “GeoCoV19: a dataset of hundreds of millions of multilingual COVID-19 tweets with location information”. In: *SIGSPATIAL Special* 12.1, pp. 6–15. DOI: [10.1145/3404820.3404823](https://doi.org/10.1145/3404820.3404823).
- Quelle, Dorian and Alexandre Bovet (2025). “Bluesky: Network topology, polarization, and algorithmic curation”. In: *PloS one* 20.2, e0318034.
- Rader, Benjamin et al. (2024). “Adherence to non-pharmaceutical interventions following COVID-19 vaccination: a federated cohort study”. In: *NPJ Digital Medicine* 7.1, p. 241.
- Radford, Alec and Karthik Narasimhan (2018). “Improving Language Understanding by Generative Pre-Training”. In.
- Radford, Alec et al. (2019). “Language Models are Unsupervised Multitask Learners”. In.

- Raffel, Colin et al. (Jan. 2020). “Exploring the limits of transfer learning with a unified text-to-text transformer”. In: *J. Mach. Learn. Res.* 21.1. ISSN: 1532-4435.
- Raschka, Sebastian (2020). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*. URL: <https://arxiv.org/abs/1811.12808>.
- Reitenbach, Andreas et al. (2024). “Coupled infectious disease and behavior dynamics. A review of model assumptions”. In: *Reports on Progress in Physics*.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). “” Why should i trust you?” Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.
- Ribeiro, Matheus Henrique Dal Molin et al. (2020). “Short-term forecasting COVID-19 cumulative confirmed cases: Perspectives for Brazil”. In: *Chaos, Solitons & Fractals* 135, p. 109853. ISSN: 0960-0779. DOI: <https://doi.org/10.1016/j.chaos.2020.109853>.
- Rodríguez-Ibáñez, Margarita et al. (2023). “A review on sentiment analysis from social media platforms”. In: *Expert Systems with Applications* 223, p. 119862.
- Rosenblatt, Frank (1958). “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6, p. 386.
- Rosenstock, Irwin M (1974). “Historical origins of the health belief model”. In: *Health education monographs* 2.4, pp. 328–335.
- Ross Arguedas, Amy et al. (2022). “Echo chambers, filter bubbles, and polarisation: A literature review”. In.
- Rostami, Peyman et al. (June 2025). *POLITISKY24: U.S. Political Bluesky Dataset with User Stance Labels*. DOI: [10.5281/zenodo.15616911](https://doi.org/10.5281/zenodo.15616911). URL: <https://doi.org/10.5281/zenodo.15616911>.
- Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). “Learning representations by back-propagating errors”. In: *nature* 323.6088, pp. 533–536.
- Russell, James A (1980). “A circumplex model of affect.” In: *Journal of personality and social psychology* 39.6, p. 1161.

- Russell, Timothy W et al. (2020). “Reconstructing the early global dynamics of under-ascertained COVID-19 cases and infections”. In: *BMC medicine* 18.1, p. 332.
- Saad-Roy, Chadi M. and Arne Traulsen (2023). “Dynamics in a behavioral–epidemiological model for individual adherence to a nonpharmaceutical intervention”. In: *Proceedings of the National Academy of Sciences* 120.44, e2311584120. DOI: [10.1073/pnas.2311584120](https://doi.org/10.1073/pnas.2311584120).
- Salomon, Joshua A et al. (2021). “The US COVID-19 Trends and Impact Survey: Continuous real-time measurement of COVID-19 symptoms, risks, protective behaviors, testing, and vaccination”. In: *Proceedings of the National Academy of Sciences* 118.51, e2111454118.
- Salton, G., A. Wong, and C. S. Yang (Nov. 1975). “A vector space model for automatic indexing”. In: *Commun. ACM* 18.11, pp. 613–620. ISSN: 0001-0782. DOI: [10.1145/361219.361220](https://doi.org/10.1145/361219.361220).
- Salton, Gerard and Christopher Buckley (1988). “Term-weighting approaches in automatic text retrieval”. In: *Information Processing Management* 24.5, pp. 513–523. ISSN: 0306-4573. DOI: [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0).
- Sampath, Shrikanth et al. (2021). “Pandemics throughout the history”. In: *Cureus* 13.9.
- Schapire, Robert E (1990). “The strength of weak learnability”. In: *Machine learning* 5.2, pp. 197–227.
- Schuller, Björn W (2018). “Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends”. In: *Communications of the ACM* 61.5, pp. 90–99.
- Shi, Yuan et al. (2016). “Three-month real-time dengue forecast models: an early warning system for outbreak alerts and policy decision support in Singapore”. In: *Environmental health perspectives* 124.9, pp. 1369–1375.
- Shojaie, Ali and Emily B Fox (2022). “Granger causality: A review and recent advances”. In: *Annual Review of Statistics and Its Application* 9.1, pp. 289–319.
- Siegrist, Michael and Alexandra Zingg (2014). “The role of public trust during pandemics”. In: *European psychologist*.

- Singh, Kamakhya Narain et al. (2021). “Analysis and Validation of Risk Prediction by Stochastic Gradient Boosting Along with Recursive Feature Elimination for COVID-19”. In: *Applications of Artificial Intelligence in COVID-19*. Springer, pp. 307–323.
- Sloan, Luke and Jeffrey Morgan (Nov. 2015). “Who Tweets with Their Location? Understanding the Relationship between Demographic Characteristics and the Use of Geoservices and Geotagging on Twitter”. In: *PLOS ONE* 10.11, pp. 1–15. DOI: [10.1371/journal.pone.0142209](https://doi.org/10.1371/journal.pone.0142209).
- Smith, Louise E. et al. (Dec. 2022). “Engagement with protective behaviours in the UK during the COVID-19 pandemic: a series of cross-sectional surveys (the COVID-19 rapid survey of adherence to interventions and responses [CORSAIR] study)”. In: *BMC Public Health* 22 (1). ISSN: 14712458. DOI: [10.1186/s12889-022-12777-x](https://doi.org/10.1186/s12889-022-12777-x).
- Sneath, Peter HA and Robert R Sokal (1973). “Unweighted pair group method with arithmetic mean”. In: *Numerical Taxonomy*, pp. 230–234.
- Snøeijer, Berber T et al. (2021). “Measuring the effect of Non-Pharmaceutical Interventions (NPIs) on mobility during the COVID-19 pandemic using global mobility data”. In: *NPJ digital medicine* 4.1, p. 81.
- So, Jiyeon, Kai Kuang, and Hyunyi Cho (2016). “Reexamining fear appeal models from cognitive appraisal theory and functional emotion theory perspectives”. In: *Communication Monographs* 83.1, pp. 120–144.
- Sohail, Shahab Saquib et al. (2025). “Geospatial and Linguistic Analysis of Twitter Behavioral Trends: Examining the Impact of Socioeconomic Development on Social Media Use”. In: *Human Behavior and Emerging Technologies* 2025.1, p. 1376983.
- SPI-M-O (July 2021). *SPI-M-O: Summary of further modelling of easing restrictions 2014 Roadmap Step 4 on 19 July 2021, 7 July 2021*. Last updated 13 May 2022; accessed 15 August 2025. URL: <https://www.gov.uk/government/publications/spi-m-o-summary-of-further-modelling-of-easing-restrictions-roadmap-step-4-on-19-july-2021-7-july-2021>.

- Statista (2025). *Number of social media users worldwide from 2017 to 2028 (in billions)*. Accessed: 2025-08-09. URL: <https://www.statista.com/topics/1164/social-networks/>.
- Stiennon, Nisan et al. (2022). *Learning to summarize from human feedback*. URL: <https://arxiv.org/abs/2009.01325>.
- Summers, Ed et al. (Apr. 2023). *DocNow/twarc: v2.14.0*. Version v2.14.0. DOI: [10.5281/zenodo.7799050](https://doi.org/10.5281/zenodo.7799050).
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). “Sequence to Sequence Learning with Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc.
- Szeto, Stephanie, Algae Kit Yee Au, and Sophie Kai Lam Cheng (2024). “Support from Social Media during the COVID-19 Pandemic: A Systematic Review”. In: *Behavioral Sciences* 14.9, p. 759.
- Taboada, Maite et al. (June 2011). “Lexicon-Based Methods for Sentiment Analysis”. In: *Computational Linguistics* 37.2, pp. 267–307. ISSN: 0891-2017. DOI: [10.1162/COLI_a_00049](https://doi.org/10.1162/COLI_a_00049).
- Tang, Lu et al. (2018). “Social media and outbreaks of emerging infectious diseases: A systematic review of literature”. In: *American journal of infection control* 46.9, pp. 962–972.
- Tanner, Alastair R et al. (Feb. 2023). “Epicosm—a framework for linking online social media in epidemiological cohorts”. In: *International Journal of Epidemiology* 52.3, pp. 952–957. ISSN: 0300-5771. DOI: [10.1093/ije/dyad020](https://doi.org/10.1093/ije/dyad020).
- Terry, Kirsty et al. (2023). “The role of social media in public health crises caused by infectious disease: a scoping review”. In: *BMJ Global Health* 8.12.
- Tibshirani, Robert (1996). “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1, pp. 267–288.
- Topal, Mustafa Burak, Aysun Bozanta, and Ayse Basar (2024). “Sentiment Analysis with LLMs: Evaluating QLoRA Fine-Tuning, Instruction Strategies, and Prompt

- Sensitivity”. In: *2024 34th International Conference on Collaborative Advances in Software and COmputiNg (CASCON)*, pp. 1–10. DOI: [10.1109/CASCON62161.2024.10838185](https://doi.org/10.1109/CASCON62161.2024.10838185).
- Touvron, Hugo et al. (2023a). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. URL: <https://arxiv.org/abs/2307.09288>.
- Touvron, Hugo et al. (2023b). *LLaMA: Open and Efficient Foundation Language Models*. URL: <https://arxiv.org/abs/2302.13971>.
- Tsao, Shu-Feng et al. (2021). “What social media told us in the time of COVID-19: a scoping review”. In: *The Lancet Digital Health* 3.3, e175–e194.
- Tulchinsky, Alexander Y et al. (June 2025). “Machine Learning–Driven COVID-19 Hospitalization Forecasting: From Theory to Practice in a Major Northeastern Academic Medical Center”. In: *Open Forum Infectious Diseases* 12.6, ofaf307. ISSN: 2328-8957. DOI: [10.1093/ofid/ofaf307](https://doi.org/10.1093/ofid/ofaf307).
- Turiel, Jeremy, Delmiro Fernandez-Reyes, and Tomaso Aste (Dec. 2021). “Wisdom of crowds detects COVID-19 severity ahead of officially available data”. In: *Scientific Reports* 11 (1). ISSN: 20452322. DOI: [10.1038/s41598-021-93042-w](https://doi.org/10.1038/s41598-021-93042-w).
- Twitter (2023). *Advanced filtering with geo data*. Accessed: 2023-11-07. URL: <https://developer.twitter.com/en/docs/tutorials/advanced-filtering-for-geo-data>.
- United Nations Department of Economic and Social Affairs, Population Division (2024). *World Population Prospects 2024 Revision: Indicator 49 (Total Population for World, 1950–2100)*. UN Population Division Data Portal. Accessed September 2, 2025. URL: <https://population.un.org/dataportal/data/indicators/49/locations/900/start/1950/end/2100/line/linetimeplotsingle>.
- Ushio, Asahi and Jose Camacho-Collados (Aug. 2024). “A Systematic Analysis on the Temporal Generalization of Language Models in Social Media”. In: *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*. Ed. by Orphée De Clercq et al. Bangkok, Thailand: Association for Computational Linguistics, pp. 52–62. DOI: [10.18653/v1/2024.wassa-1.5](https://doi.org/10.18653/v1/2024.wassa-1.5).

- Vajjala, Sowmya and Shweta Shimangaud (2025). *Text Classification in the LLM Era – Where do we stand?* URL: <https://arxiv.org/abs/2502.11830>.
- Vapnik, Vladimir (2013). *The nature of statistical learning theory*. Springer science & business media.
- Vapnik, Vladimir, Steven Golowich, and Alex Smola (1996). “Support vector method for function approximation, regression estimation and signal processing”. In: *Advances in neural information processing systems* 9.
- Varma, Sudhir and Richard Simon (2006). “Bias in error estimation when using cross-validation for model selection”. In: *BMC bioinformatics* 7.1, p. 91.
- Vaswani, Ashish et al. (2017). “Attention is all you need”. In: *Advances in neural information processing systems* 30.
- Verelst, Frederik, Lander Willem, and Philippe Beutels (2016). “Behavioural change models for infectious disease transmission: a systematic review (2010–2015)”. In: *Journal of The Royal Society Interface* 13.125, p. 20160820.
- Verity, Robert et al. (2020). “Estimates of the severity of coronavirus disease 2019: a model-based analysis”. In: *The Lancet infectious diseases* 20.6, pp. 669–677.
- Viboud, Cécile et al. (2018). “The RAPIDD ebola forecasting challenge: Synthesis and lessons learnt”. In: *Epidemics* 22. The RAPIDD Ebola Forecasting Challenge, pp. 13–21. ISSN: 1755-4365. DOI: <https://doi.org/10.1016/j.epidem.2017.08.002>.
- Waibel, Alexander et al. (2013). “Phoneme recognition using time-delay neural networks”. In: *Backpropagation*. Psychology Press, pp. 35–61.
- Walkowiak, Marcin Piotr and Dariusz Walkowiak (2022). “Underestimation in Reporting Excess COVID-19 Death Data in Poland during the First Three Pandemic Waves”. In: *International Journal of Environmental Research and Public Health* 19.6. ISSN: 1660-4601. DOI: [10.3390/ijerph19063692](https://doi.org/10.3390/ijerph19063692).
- Wang, Jianghao et al. (Mar. 2022a). “Global evidence of expressed sentiment alterations during the COVID-19 pandemic”. In: *Nature Human Behaviour* 6 (3), pp. 349–358. ISSN: 23973374. DOI: [10.1038/s41562-022-01312-y](https://doi.org/10.1038/s41562-022-01312-y).

- Wang, Tao et al. (2022b). “COVID-19 hospitalizations forecasts using internet search data”. In: *Scientific Reports* 12.1, p. 9661.
- Wang, Yuan and Yonghao Chen (2023). “Characterizing discourses about COVID-19 vaccines on Twitter: a topic modeling and sentiment analysis approach”. In: *Journal of Communication in Healthcare* 16 (1), pp. 103–112. ISSN: 17538076. DOI: [10.1080/17538068.2022.2054196](https://doi.org/10.1080/17538068.2022.2054196).
- Wang, Zengzhi et al. (2024). *Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study*. URL: <https://arxiv.org/abs/2304.04339>.
- Wankhade, Mayur, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni (2022). “A survey on sentiment analysis methods, applications, and challenges”. In: *Artificial Intelligence Review* 55.7, pp. 5731–5780.
- Ward, Thomas and Alexander Johnsen (2021). “Understanding an evolving pandemic: An analysis of the clinical time delay distributions of COVID-19 in the United Kingdom”. In: *Plos one* 16.10, e0257978.
- Watson, Oliver J et al. (2021). “Leveraging community mortality indicators to infer COVID-19 mortality and transmission dynamics in Damascus, Syria”. In: *Nature communications* 12.1, p. 2394.
- Werbos, P. J. (1981). “Applications of Advances in Nonlinear Sensitivity Analysis”. In: *Proceedings of the 10th IFIP Conference, 31.8 - 4.9, NYC*, pp. 762–770.
- Wesolowski, Amy et al. (Nov. 2016). “Connecting Mobility to Infectious Diseases: The Promise and Limits of Mobile Phone Data”. In: *The Journal of Infectious Diseases* 214.suppl_4, S414–S420. ISSN: 0022-1899. DOI: [10.1093/infdis/jiw273](https://doi.org/10.1093/infdis/jiw273).
- Weston, Dale, Katharina Hauck, and Richard Amlôt (2018). “Infection prevention behaviour and infectious disease modelling: a review of the literature and recommendations for the future”. In: *BMC public health* 18.1, p. 336.
- Whitaker, Matthew et al. (2022). “Variant-specific symptoms of COVID-19 in a study of 1,542,510 adults in England”. In: *Nature Communications* 13.1, p. 6856.
- WHO (2022). *Infodemic*. Accessed: 2023-11-07. URL: https://www.who.int/health-topics/infodemic#tab=tab_1.

- Williams, David M, Ryan E Rhodes, and Mark T Conner (2019). *Conceptualizing and intervening on affective determinants of health behaviour*.
- Williams, Ronald J and David Zipser (1989). “A learning algorithm for continually running fully recurrent neural networks”. In: *Neural computation* 1.2, pp. 270–280.
- Witte, Kim (1992). “Putting the fear back into fear appeals: The extended parallel process model”. In: *Communications Monographs* 59.4, pp. 329–349.
- Wold, Herman O. A. (1938). *A Study in the Analysis of Stationary Time Series*. UppsalaA Time Series is Worth 64 Words: Long-term Forecasting with Transformers: Almqvist & Wiksell.
- Wolf, Thomas et al. (Oct. 2020). “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Ed. by Qun Liu and David Schlangen. Online: Association for Computational Linguistics, pp. 38–45. DOI: [10.18653/v1/2020.emnlp-demos.6](https://doi.org/10.18653/v1/2020.emnlp-demos.6).
- World Health Organization Regional Office for Europe (2025). *Coronavirus disease (COVID-19) pandemic*. Accessed: 2025-08-11. URL: <https://www.who.int/europe/emergencies/situations/covid-19>.
- Wright, Liam et al. (2022). “Do people reduce compliance with COVID-19 guidelines following vaccination? A longitudinal analysis of matched UK adults”. In: *J Epidemiol Community Health* 76.2, pp. 109–115.
- Wu, Haixu et al. (2021a). “Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 22419–22430.
- Wu, Haixu et al. (2023). *TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis*. URL: <https://arxiv.org/abs/2210.02186>.
- Wu, Jeff et al. (2021b). *Recursively Summarizing Books with Human Feedback*. URL: <https://arxiv.org/abs/2109.10862>.

- Wu, Shijie and Mark Dredze (Nov. 2019). “Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: [10.18653/v1/D19-1077](https://doi.org/10.18653/v1/D19-1077).
- Xie, Tianyi et al. (Mar. 2023). “Public Awareness and Sentiment Analysis of COVID-Related Discussions Using BERT-Based Infoveillance”. In: *AI (Switzerland)* 4 (1), pp. 333–347. ISSN: 26732688. DOI: [10.3390/ai4010016](https://doi.org/10.3390/ai4010016).
- Yang, Shihao, Mauricio Santillana, and S. C. Kou (2015). “Accurate estimation of influenza epidemics using Google search data via ARGO”. In: *Proceedings of the National Academy of Sciences* 112.47, pp. 14473–14478. DOI: [10.1073/pnas.1515373112](https://doi.org/10.1073/pnas.1515373112).
- Yin, Jason Dean Chen and Juliana Nga Man Lui (Dec. 2024). *Factors influencing risk perception during Public Health Emergencies of International Concern (PHEIC): a scoping review*. DOI: [10.1186/s12889-024-18832-z](https://doi.org/10.1186/s12889-024-18832-z).
- Yule, George Udny (n.d.). “On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer’s Sunspot Numbers”. In: *Philosophical Transactions of the Royal Society A* 226 (), pp. 267–298.
- Zbezhkhovska, Uliana and Dmytro Chumachenko (2025). “Smoothing Techniques for Improving COVID-19 Time Series Forecasting Across Countries”. In: *Computation* 13.6. ISSN: 2079-3197. DOI: [10.3390/computation13060136](https://doi.org/10.3390/computation13060136).
- Zeff, Maxwell (2024). *OpenAI unveils GPT-4o Mini, a smaller and cheaper AI model*. URL: <https://techcrunch.com/2024/07/18/openai-unveils-gpt-4o-mini-a-small-ai-model-powering-chatgpt/>.
- Zeng, Ailing et al. (2022). *Are Transformers Effective for Time Series Forecasting?* URL: <https://arxiv.org/abs/2205.13504>.
- Zhang, Cheng et al. (2018). “Advances in variational inference”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.8, pp. 2008–2026.

- Zhang, Wenxuan et al. (June 2024). “Sentiment Analysis in the Era of Large Language Models: A Reality Check”. In: *Findings of the Association for Computational Linguistics: NAACL 2024*. Ed. by Kevin Duh, Helena Gomez, and Steven Bethard. Mexico City, Mexico: Association for Computational Linguistics, pp. 3881–3906. DOI: [10.18653/v1/2024.findings-naacl.246](https://doi.org/10.18653/v1/2024.findings-naacl.246).
- Zhang, Xinyang et al. (2023). “Twhin-bert: A socially-enriched pre-trained language model for multilingual tweet representations at twitter”. In: *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 5597–5607.
- Zhong, Qihuang et al. (2023). *Can ChatGPT Understand Too? A Comparative Study on ChatGPT and Fine-tuned BERT*. URL: <https://arxiv.org/abs/2302.10198>.
- Zhou, Haoyi et al. (2021). *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. URL: <https://arxiv.org/abs/2012.07436>.
- Zhu, Xiliang et al. (Aug. 2024). “The Model Arena for Cross-lingual Sentiment Analysis: A Comparative Study in the Era of Large Language Models”. In: *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*. Ed. by Orphée De Clercq et al. Bangkok, Thailand: Association for Computational Linguistics, pp. 141–152. DOI: [10.18653/v1/2024.wassa-1.12](https://doi.org/10.18653/v1/2024.wassa-1.12).
- Ziegler, Daniel M. et al. (2020). *Fine-Tuning Language Models from Human Preferences*. URL: <https://arxiv.org/abs/1909.08593>.

A Appendix: Time series methods

A.1 Least Absolute Shrinkage and Selection Operator

The bias-variance trade-off

The theoretical justification for the lasso depends on the bias-variance trade-off (Hastie, Tibshirani, and Friedman, 2001). Before presenting this justification, I provide a high-level overview of the concept, both to support my explanation of the lasso and to help understand subsequent sections. Assuming a squared error loss (for example, MSE), the expected prediction loss on out-of-sample data can be decomposed into three distinct components: bias, variance and irreducible error. These components are given below.

- Bias measures how far the average prediction deviates from the true function. High bias typically indicates that the model is too simple and lacks sufficient flexibility to capture the underlying function.
- Variance quantifies the model's sensitivity to the specific training data used for fitting. High variance means the model has fit the training data too closely, potentially capturing noise rather than the underlying signal.
- Irreducible error represents a lower bound on the prediction error. It reflects the inherent noise in the data that no model can explain. This may arise from measurement errors or other random variability in the observations.

Within the framework of the bias-variance trade-off, two key concepts arise, namely underfitting and overfitting. Both terms describe reasons why a model may fail to generalise. Overfitting occurs when a model fits the data too closely, including the noise; high variance is an example of this. Underfitting occurs when a model is too simple and fails to capture enough of the underlying signal; high bias is an example. The goal in learning from data is to strike a balance between bias and variance to minimise the expected prediction loss.

To achieve this balance, it is necessary to choose a suitable model class or specification within a class. Typically, there is a trade-off between bias and variance when selecting

a model, known as the bias-variance trade-off. A model with high variance often has lower bias, and vice versa.¹

It should be noted that the bias-variance decomposition applies quantitatively under the MSE framework but is difficult to apply with MAE, as the latter loss function is non-differentiable at zero. Nevertheless, the trade-off between bias and variance remains relevant when fitting models using MAE.

The theoretical justification of the lasso

With an understanding of bias-variance trade-off, I now return to the theoretical justification of the lasso. Assuming MSE and the standard linear regression assumptions hold (linearity, unbiasedness, homoscedasticity, uncorrelated errors and no exact multicollinearity), the Gauss–Markov theorem states that the ordinary least squares (OLS) estimator of the parameter vector β is the Best Linear Unbiased Estimator (BLUE). Here, “best” means having the lowest variance among all linear unbiased estimators. However, the lasso introduces a shrinkage term that induces bias but can substantially reduce variance by simplifying the model. As a result, in some cases, the lasso achieves a lower overall MSE than the unbiased ordinary least squares (OLS) estimator by trading off some bias for a larger reduction in variance.

The lasso uses an ℓ_1 norm as its penalty, corresponding to $\sum_{j=1}^{l+p} |\beta_j|$. This yields the penalised formulation below:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ \mathcal{L}(\mathbf{y}_T, \hat{\mathbf{y}}_T) + \lambda \sum_{j=1}^{p+l} |\beta_j| \right\} \quad (\text{A.1})$$

$$= \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{t=1}^T \left(y_t - \sum_{k=1}^l z_{t-t_k}^y \beta_k - \sum_{j=1}^p z_{t-t_j}^x \beta_j \right)^2 + \lambda \sum_{j=1}^{p+l} |\beta_j| \right\} \quad (\text{A.2})$$

$$(\text{A.3})$$

The ℓ_1 constraint corresponds to a diamond-shaped (or cross-polytope in higher dimensions) constraint region. This geometry allows some parameter estimates to be

¹While this usually applies, for problems with simpler functional forms this may be less of an issue, for example linear problems.

exactly zero, enabling variable selection. The value of λ controls the extent of shrinkage. As λ increases, the feasible parameter space shrinks and coefficients become smaller because there is a higher penalty on each coefficient. In the limit, as $\lambda \rightarrow \infty$, all coefficients shrink to zero, and the model reduces to a constant equal to the intercept. Intuitively, λ controls model complexity, where a lower value leads to a more complex model and a higher value results in a simpler model. This process of controlling complexity is known as regularisation. Because of this penalised formulation, a closed-form solution for β no longer exists, so an iterative optimisation procedure, as described earlier, is used.

A.2 Regression trees

To determine c_m for each region using MAE as the loss function, the aim is to find the value of c that minimises the loss function given in Equation A.4. While a proof is not provided, it can be mathematically shown that the optimal value of c under this loss is the median of the response values within the region (Breiman et al., 1984).

$$c_m = \arg \min_c \sum_{\mathbf{z}_t \in R_m} |y_{t+1} - c| \quad (\text{A.4})$$

Given the above, growing the tree reduces to partitioning the predictor space so that the resulting medians minimise the MAE. Since exhaustively evaluating all possible splits is computationally infeasible, a greedy, recursive strategy is adopted. At each node, the best split is chosen locally rather than globally for the entire tree. This involves selecting both the optimal splitting variable j and the corresponding split point s that minimises the loss at that node. For MAE, this is given more formally in Equation A.5.

$$\min_{j,s} \left\{ \sum_{\mathbf{z}_t \in R_1(j,s)} |y_{t+1} - c_1| + \sum_{\mathbf{z}_t \in R_2(j,s)} |y_{t+1} - c_2| \right\} \quad (\text{A.5})$$

where R_1 and R_2 define the two half-planes given by: $R_1(j, s) = Z_j < s$, $R_2(j, s) = Z_j \geq s$. Note that the subscript j selects both a time lag and whether the variable is

AR or exogenous.

Although a tree can be grown recursively as described, a key challenge is deciding how large the tree should be, as noted in Section 3.7.2. This issue is addressed through cost-complexity pruning. According to this approach, Denoting the full tree as T_0 , the task is to identify a subtree $T \subseteq T_0$ where the number of leaves T depends on the value of $\alpha = 0$. Formally, this is achieved by adding a penalty term $\alpha|T|$ to the optimisation problem defined in Equation A.4, and finding the subtree which minimises the regularised loss.

A.3 Support Vector Regression

In this Appendix, I outline the primal and dual formulations of the SVR optimisation problem. Based on ε -insensitive loss, the primal optimisation problem can be written as follows:

$$\min_{\beta, \beta_0, \{\xi_i, \xi_i^*\}} \frac{1}{2} \|\beta\|^2 + C \sum_{t=1}^T (\xi_t + \xi_t^*) \quad (\text{A.6})$$

$$\text{subject to } \begin{cases} y_{t+1} - \beta^\top \phi(\mathbf{z}_t) - \beta_0 \leq \varepsilon + \xi_t, \\ \beta^\top \phi(\mathbf{z}_t) + \beta_0 - y_{t+1} \leq \varepsilon + \xi_t^*, \\ \xi_t, \xi_t^* \geq 0, \quad t = 1, \dots, T \end{cases} \quad (\text{A.7})$$

where:

- $\beta \in R^{l+p}$
- $\mathbf{z}_t \in R^{l+p}$, as previously defined
- $C \sum_{i=1}^n (\xi_i + \xi_i^*)$, is penalty term for the slack variables
- C is the regularisation coefficient, similar to λ in the previous models, where higher values C result in stricter penalty and a simpler model.
- The first two enforce that ξ, ξ^* capture the difference above and below the ε tube for an offending sample.

The above is a constrained optimisation problem, to solve this, it is reframed using Lagrange multipliers in the dual form given below:

$$\begin{aligned} \max_{\alpha_{t_i}, \alpha_{t_i}^*} \quad & -\frac{1}{2} \sum_{t_i=1}^T \sum_{t_j=1}^T (\alpha_{t_i} - \alpha_{t_i}^*)(\alpha_{t_j} - \alpha_{t_j}^*) K(\mathbf{z}_{t_i}, \mathbf{z}_{t_j}) \\ & - \varepsilon \sum_{t_i=1}^T (\alpha_{t_i} + \alpha_{t_i}^*) + \sum_{t_i=1}^T y_{t_i+1} (\alpha_{t_i} - \alpha_{t_i}^*) \end{aligned} \quad (\text{A.8})$$

Subject to:

$$\sum_{t=1}^T (\alpha_{t_i} - \alpha_{t_i}^*) = 0 \quad (\text{A.9})$$

$$0 \leq \alpha_{t_i}, \alpha_{t_i}^* \leq C, \quad \text{for } t_i = 1, \dots, T \quad (\text{A.10})$$

where:

- α_{t_i} and $\alpha_{t_i}^*$ are the dual variables
- $K(\mathbf{z}_{t_i}, \mathbf{z}_{t_j})$ is the kernel function, representing the inner product in the transformed feature space

This reformulation has two important implications. First, the solution to SVR depends only on points for which α_{t_i} and $\alpha_{t_i}^*$ are non-zero. These correspond to data points that lie either on the boundary of the ε -insensitive tube (called margin support vectors) or outside it (called slack support vectors). The dual variables highlights why only the support vectors contribute to the final prediction function, as noted earlier.

Second, the solution depends only on the kernel function applied to the inputs. In this case, the kernel function is the inner product in the transformed feature space: $K(\mathbf{z}_{t_i}, \mathbf{z}_{t_j}') = \langle \phi(\mathbf{z}_{t_i}), \phi(\mathbf{z}_{t_j}') \rangle$. This is valuable because it means the input data does not need to be explicitly mapped into the high-dimensional feature space, which may be computationally expensive or at times infeasible. Instead, only evaluations of the kernel function are required, including at prediction time.

A.4 Vanilla Recurrent Neural Network

The hidden state output at time t in an RNN (see Figure 3.6 for an illustration of the unrolled architecture) can be expressed formally as:

$$\mathbf{h}_t = f(W_z \mathbf{z}_t + V_h \mathbf{h}_{t-1} + b_h) \quad (\text{A.11})$$

where:

- $\mathbf{h}_t \in \mathbb{R}^d$ is the hidden state at time t , where d is the dimension of a single encoder layer.
- $\mathbf{z}_t \in \mathbb{R}^p$ is the input at time t , where it is assumed there are p predictors, either AR or exogenous².
- $\mathbf{h}_t \in \mathbb{R}^h$ is the hidden state at time t .
- $W_z \in \mathbb{R}^{d \times p}$ is the input-to-hidden weight matrix.
- $V_h \in \mathbb{R}^{h \times h}$ is the matrix of hidden-to-hidden weights.
- $b_h \in \mathbb{R}^h$ is the bias vector for the hidden layer.
- $f(\cdot)$ is the activation function.

A.5 Gated cells mechanisms

Two RNN gated cells mechanisms are illustrated in Figure A.1. Each mechanism is described in greater detail below.

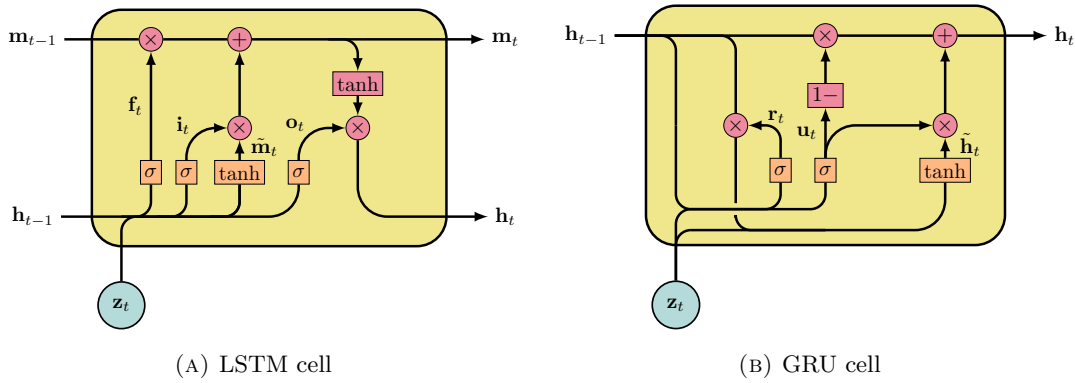


FIGURE A.1: Diagram of (A) an LSTM cell and (B) a GRU cell. Elements highlighted in red indicate the components contributing to the final cell state.

The gating mechanism of the LSTM, shown in Figure A.1a, is more complex than that of the GRU. The LSTM includes three gates: an input gate ($\mathbf{i}_t$), a forget gate

²This notation differs from the previous $\mathbf{z}_t$ notation used for machine learning models.

($\mathbf{f}_t$), and an output gate ($\mathbf{o}_t$). In addition, it introduces a dedicated memory state, $\mathbf{m}_t$, alongside the hidden state $\mathbf{h}_t$. The forget gate determines what proportion of information to discard from the previous memory cell ($\mathbf{m}_{t-1}$). A new candidate value, based on the previous hidden state and current input, is generated, and the input gate determines how much of this candidate should be added to the current memory state. The memory state is then updated by forgetting irrelevant parts of the previous memory and adding relevant information from the candidate. Finally, the output gate determines which part of the current memory state should be output as the hidden state.

I formalise the descriptions in the preceding paragraph below. All components are as defined in the RNN formulation above. In this case, specific activation functions are used: σ denotes a sigmoid activation function, and $\odot$ denotes the Hadamard (element-wise) product. Together, this implies an element-wise multiplication of a gating proportion and the corresponding memory state.

$$\mathbf{f}_t = \sigma(W_f \mathbf{z}_t + V_f \mathbf{h}_{t-1} + \mathbf{b}_f) \quad (\text{Forget gate}) \quad (\text{A.12})$$

$$\mathbf{i}_t = \sigma(W_i \mathbf{z}_t + V_i \mathbf{h}_{t-1} + \mathbf{b}_i) \quad (\text{Input gate}) \quad (\text{A.13})$$

$$\mathbf{o}_t = \sigma(W_o \mathbf{z}_t + V_o \mathbf{h}_{t-1} + \mathbf{b}_o) \quad (\text{Output gate}) \quad (\text{A.14})$$

$$\tilde{\mathbf{m}}_t = \tanh(W_m \mathbf{z}_t + V_m \mathbf{h}_{t-1} + \mathbf{b}_m) \quad (\text{Memory state candidate}) \quad (\text{A.15})$$

$$\mathbf{m}_t = \mathbf{f}_t \odot \mathbf{m}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{m}}_t \quad (\text{Memory state update}) \quad (\text{A.16})$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{m}_t) \quad (\text{Hidden state update}) \quad (\text{A.17})$$

In contrast, a GRU, shown in Figure A.1b, is simpler and includes only two gates: an update gate ($\mathbf{u}_t$) and a reset gate ($\mathbf{r}_t$). There is no explicit memory state; all information is stored in the hidden state. A GRU generates a new candidate hidden state ($\tilde{\mathbf{h}}_t$) for the current time step. It uses the reset gate to determine how much information from the previous state should be used when generating this candidate. The update gate then combines the candidate and previous hidden states to produce the final hidden state for the current time step. Rather than having explicit forget and input gates, the update gate uses complementary proportions. Due to their simplicity, GRUs are more computationally efficient. The decision to use a GRU, an

LSTM, or neither depends on the sequence length, the data available, and ultimately the complexity of the problem being solved.

I formalise the descriptions in the preceding paragraph below. All components are as defined in the respective RNN and LSTM formulations above.

$$\mathbf{u}_t = \sigma(W_u \mathbf{z}_t + V_u \mathbf{h}_{t-1}) \quad (\text{Update gate}) \quad (\text{A.18})$$

$$\mathbf{r}_t = \sigma(W_r \mathbf{z}_t + V_r \mathbf{h}_{t-1}) \quad (\text{Reset gate}) \quad (\text{A.19})$$

$$\tilde{\mathbf{h}}_t = \tanh(W_c \mathbf{z}_t + V_c(\mathbf{r}_t \odot \mathbf{h}_{t-1})) \quad (\text{Candidate hidden state}) \quad (\text{A.20})$$

$$\mathbf{h}_t = (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \tilde{\mathbf{h}}_t \quad (\text{Final hidden state}) \quad (\text{A.21})$$

B Appendix: curating a dataset of tweets

B.1 Pydantic model

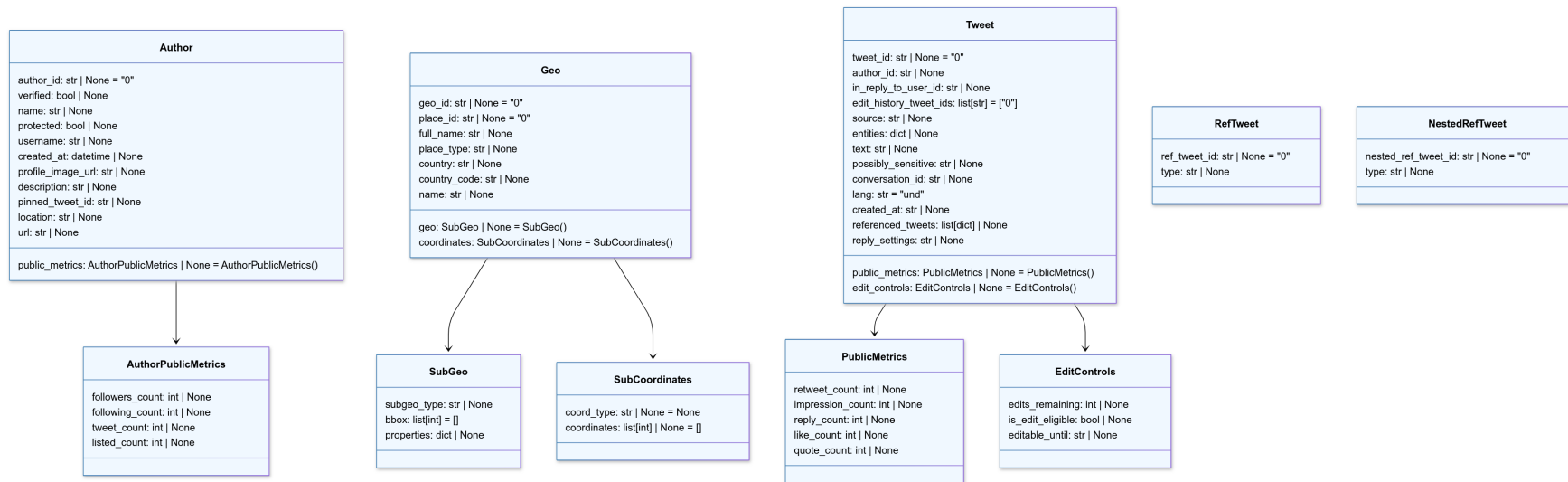


FIGURE B.1: Representation of the Pydantic model used to parse JSON responses from the Twitter API v2.

B.2 Creating mapping files

```

COPY (select tweet_id,
            sqm.search_query_id,
            search_query,
            is_ref_tweet,
            split_part(search_query, 'place_country:', 2) AS country
from 'sq_mapping/*.parquet' as sqm
LEFT JOIN 'key_tables/search_query.parquet' on
    → sqm.search_query_id=search_query.search_query_id
WHERE country == '<COUNTRY-CODE>') TO
    → 'sq_mapping_<COUNTRY-CODE>.parquet' (FORMAT PARQUET);

```

where <COUNTRY-CODE> was replaced with one of: DE, ES, FR, IT, NL, PL, PT, or UK

B.2.1 Bot filtering

To better explain the process of bot filtering, below is an illustrative example.

Consider an author who had posted 60 tweets in a single day. I applied the process described in Algorithm 1 and created clusters using a cut-off height of 0.25. Figure B.2 shows the results of this and indicates that there were 2 clusters using this cut-off height. The exact clusters and their values are given in Table B.1. This Table shows that this author made four distinct tweets, three of which were very similar and only differ by the number of users they tagged. To calculate the proportion of tweets that were similar, only cluster two which was below the chosen cut height was included. This resulted in a total proportion of similar tweets of 0.98 (59/60). Given that no similar tweets are allowed within the set for any author who posted 50 or more tweets in a single day, the author in this example was excluded and assumed to be a bot.

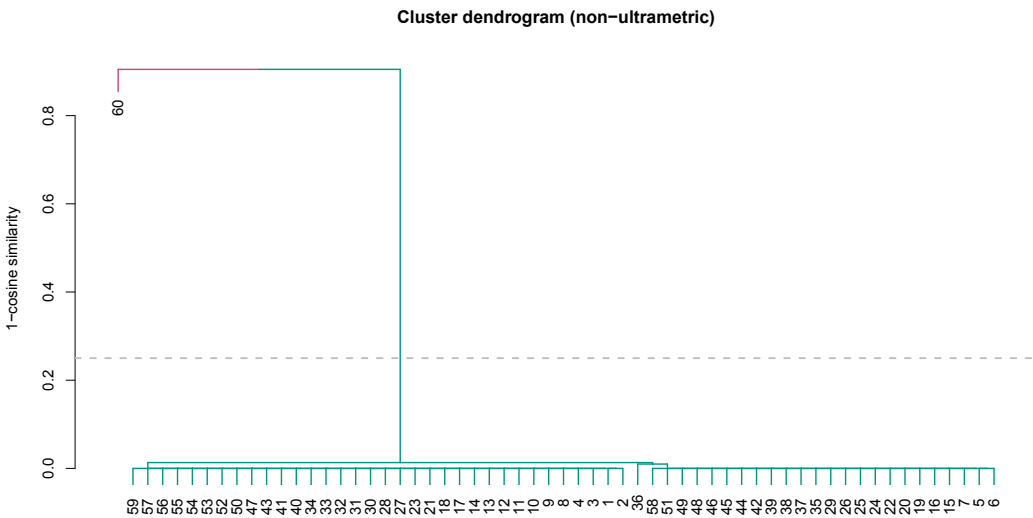


FIGURE B.2: Dendrogram of a single set of 60 tweets created using the bot detection process described in Section 4.5.2.

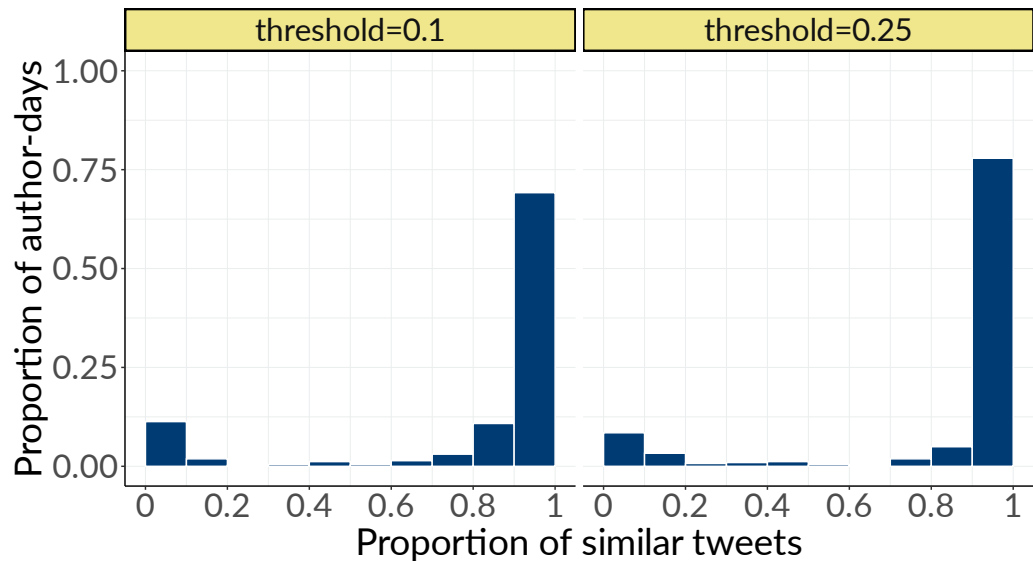


FIGURE B.3: Histogram showing the relative frequency of authors in the ≥ 50 tweets group under two distance thresholds (0.1 and 0.25).

Cluster	Size	Match	Tweet number	distinct tweets
1	1	1	60	@user @user ...OUR 11 YEAR OLD SON CONTRACTED COVID AT SECONDARY SCHOOL...HE HAS BEEN VERY UNWELL FOR 3 WEEKS AND IS STILL HAVING TROUBLE BREATHING..ESPECIALLY AT NIGHT, HE IS ABSOLUTELY EXHAUSTED...CHILDREN DO CATCH THE VIRUS AND THEY DO SUFFER...
		34	59	@user THE NASTY PARTY HAVE BEEN CAUGHT OUT,CELEBRATING THE HUGE PROFIT, THEY HAVE AMASSED BY DEFUNDING OUR NHS & PENALISING THOSE WHO ARE MOST VULNERABLE. THEY BLATANTLY INSULT & DISRESPECT US & THE LAW & THE MEDIA LOOK THE OTHER WAY. THIS IS OUR FUTURE THEY ARE DESTROYING. #ENOUGH
2	59	1	32	@user @user @user THE NASTY PARTY HAVE BEEN CAUGHT OUT,CELEBRATING THE HUGE PROFIT, THEY HAVE AMASSED BY DEFUNDING OUR NHS & PENALISING THOSE WHO ARE MOST VULNERABLE. THEY BLATANTLY INSULT & DISRESPECT US & THE LAW & THE MEDIA LOOK THE OTHER WAY. THIS IS OUR FUTURE THEY ARE DESTROYING. #ENOUGH
		24	60	@user @user THE NASTY PARTY HAVE BEEN CAUGHT OUT,CELEBRATING THE HUGE PROFIT, THEY HAVE AMASSED BY DEFUNDING OUR NHS & PENALISING THOSE WHO ARE MOST VULNERABLE. THEY BLATANTLY INSULT & DISRESPECT US & THE LAW & THE MEDIA LOOK THE OTHER WAY. THIS IS OUR FUTURE THEY ARE DESTROYING. #ENOUGH

TABLE B.1: Composition of the clusters in the bot detection example. Column description: Size - number of tweets in the cluster, Match - number of tweets which match the tweet shown in the distinct tweets column, Tweet number: the number of the tweet shown (corresponds to Figure B.2, and Distinct tweets - the tweet text.

B.2.2 Language composition

Country	SQs	Authors	Tweets	Langs	Official	English
United Kingdom	113	314 793	2 580 152	27	0.96	-
Spain	132	141 919	1 182 082	21	0.86	0.03
France	109	75 148	564 045	16	0.89	0.05
Italy	115	48 013	455 526	17	0.82	0.09
Germany	120	37 649	320 232	21	0.76	0.12
Netherlands	101	28 201	269 242	18	0.77	0.13
Portugal	89	14 394	88 351	8	0.85	0.07
Poland	76	6 910	59 708	5	0.82	0.08
Total	855	655 815	5 519 338	36	0.91	0.04

TABLE B.2: Summary of distinct tweets dataset with reference tweets removed, stratified by country. Note that the total row displays the distinct counts and proportions across the entire dataset and, consequently, may not equal the sum or mean of individual country values. Column description: SQs - total number of search terms, Langs - total number of languages, Official - the proportion of tweets in the official language, and English - the proportion of English tweets.

B.3 Tweet composition

Country	1	2	3<terms≤7
Germany	91.4	7.7	0.9
France	93.8	5.7	0.5
Italy	93.5	6.0	0.5
Netherlands	93.9	5.6	0.5
Poland	88.0	11.2	0.8
Portugal	97.7	2.2	0.1
Spain	92.2	7.0	0.8
United Kingdom	94.2	5.3	0.5
Total	93.5	5.9	0.6

TABLE B.3: The proportion of tweets in the analysis dataset with a specific number of search terms (as specified in column headings).

B.4 Search terms

The tables below present the search terms included for each country (in their official language) in the analysis dataset (Table 4.3). I use bold to indicate the terms included in the annotated dataset; by implication, the bold terms also represent those with higher tweet volumes compared to the unbolded terms. Note, however, that the annotated dataset was sampled before plural forms were mapped and may therefore contain plural variants of the highlighted terms.

B.4.1 France

Policy	Disease	Vaccine
aplatir la courbe attestation de déplacement ment attestation dérogatoire de déplacement auto-isolement cas contact cas positif cas positifs charge virale cluster clusters confinement confinement partiel contact rapproché couvre-feu crise sanitaire déconfinement dépistage distanciation physique distanciation sociale école à distance état d'urgence fermeture fermetures gel hydroalcoolique gestes barrières immunité collective immunité de groupe	corona coronavirus covid covid-19 difficulté à respirer essoufflement fièvre pas d'odorat pas de goût perte d'odorat perte de goût rona sars-cov-2 température élevée toux persistante variant variant alpha variant beta variant delta variant delta plus variant omicron variants préoccupants ventilateur	astrazeneca dose de rappel passes sanitaires troisième dose vaccin vaccin biontech vaccin moderna vaccin pfizer
<i>Continued on next page</i>		

Policy	Disease	Vaccine
isolement masque mesure restrictive mesures restrictives nombre de reproduction pass de contrôle pass sanitaire pass vaccinal pass vaccinaux pcr protocole de dépistage protocole sanitaire quarantaine r0 reconfinement règle des 100km salle de classe salles de classe stratégie de dépistage stratégies de dépistage taux d'incidence taux de reproduction télétravail télétravailler test antigénique test d'anticorps test diagnostique test nasopharyngé test rapide test sérologique tousanticovid traçage des contacts vaccination de masse vaccination obliga- toire vaccinodrome vaccinodromes		

TABLE B.4: Search terms used for tweets geotagged to France in the analysis dataset

B.4.2 Germany

Policy	Disease	Vaccine
abflachen der kurve abriegelung abstand abstrichstest alltagsmasken antigen-schnelltest antigen-selbsttest antigen-test ausgangssperre christian drosten corona-warn-app digitale impfnachweis digitalen impfnachweise	atemnot beta variante corona coronavirus covid covid-19 delta variante fieber husten kein geruchssinn kein geschmackssinn omikron variante sars-cov-2	astrazeneca auffrischimpfung auffrischungsimpfung biontech comirnaty von biontech covpass app impfstoff impfung impfzwang moderna pfizer spikevax vaxzevria von as- trazeneca
einschränkungen für das öffentliche leben enger kontakt epidemische lage von na- tionaler tragweite fallzahlen fernlernen früherkennung gesichtsmaske gesundheitskrise gesundheitsmaßnahmen häusliche isolierung heimarbeit heimbüro herdenimmunität herdenschutz hospitalisierungsrate hygiene impfstrategie infektionsrate isolation isolierung kapazität des gesund- heitssystems kapazitäten des gesund- heitssystems konjunkturpaket	störungen des geruchssinns variante virus variante	
Continued on next page		

Policy	Disease	Vaccine
kontaktbeschränkung kontaktpersonen- nachverfolgung kontakttagebuch lockdown lothar vieler maske maskenpflicht massenquarantäne mindestabstandsregelung mutation negativer schnelltest pandemiemüdigkeit pcr priorisierung quarantäne r0 reisebeschränkung reproduktionsfaktor reproduktionszahl risikobewertung risikokommunikation schnelltest schnutenpulli schritt eins schutzausrüstung screening selbstisolation selbstisolierung serologische tests soziale distanzierung teil-lockdown teststrategie unterstützung für un- ternehmen wissenschaftliche be- ratung		

TABLE B.5: Search terms used for tweets geotagged to Germany in the analysis dataset

B.4.3 Italy

Policy	Disease	Vaccine
annullamento antigenico app immuni auto-isolamento bonus carica virale contact tracing contatto stretto controllo coprifuoco didattica a distanza dispositivo di protezione individuale distanziamento distanziamento so- ziale dpcm dpi fase uno immunità di gregge incidenza isolamento lavaggio delle mani lavoro da casa lockdown maschera medicina generale passaporto sanitario passaggeri periodi di quarantena periodo di isolamento piano vaccinale priorità quarantena r0 recovery fund rinvio ristori roberto speranza sanificazione screening	corona coronavirus covid covid-19 difficoltà di respirazione difficoltà respiratorie febbre alta fiato corto non sento gli odori not used perdita del gusto perdita del senso del gusto perdita del senso dell'olfatto perdita dell'olfatto sars-cov-2 tosse continua tosse persistente variante variante alpha variante beta variante delta variante delta plus variante omicron ventilatore ventilatore polmonare	astrazeneca booster booster vaccinale certificato verde certificazione vaccinale dose di richiamo dose vaccinale green pass pass covid richiamo seconda dose vaccino vaccino biontech vaccino moderna vaccino oxford vaccino pfizer
Continued on next page		

Policy	Disease	Vaccine
seriologico smart working stato di allerta stato di emergenza tamponi tamponi di massa tasso di contagio tasso di riproduzione terapie intensive test and trace test antigenico test diagnostico test rapido test serilogico tracciamento dei contatti vaccinazione di massa voli zona rossa, arancione, gi- alla zone arancione zone gialla zone rossa		

TABLE B.6: Search terms used for tweets geotagged to Italy in the analysis dataset

B.4.4 Netherlands

Policy	Disease	Vaccine
afschaling antilichaamtest avondklok basispakket bron en contactonderzoek contactberoep contactonderzoek coronamaatregel diagnostische tests fieldlab experiment groepsimmunititeit het afvlakken van de curve horeca	aanhoudende hoest beademingsapparaat corona coronavirus covid covid-19 delta plus variant delta variant geen geur geen smaak hoge temperatuur kortademigheid moeite met ademen	astrazeneca biontech-vaccin boostervaccinatie coronabewijs coronacheck coronacheck-app herhaalprik moderna-vaccin oxford-vaccin pfizer-vaccin vaccin

Continued on next page

Policy	Disease	Vaccine
infectiegraad intelligente lockdown isolatie kudde-immuniteit lf test lft luchtkwaliteit mondkapje mondkapjesdracht mondkapjesplicht nabij contact noodwet pcr persconferentie persoonlijke bescher- mingsmiddelen quarantaine quarantaineplicht r0 reproductiegetal screening serologische test snelttest sociale afstand steunpakket testbewijs testen testen en traceren thuiswerken uitstrijkje vaccinatiebewijs vaccinatiestrategie veiligheidsregio versoepelingen waarschuingsapp zelfisolatie zorgbonus	omicron variant rona sars-cov-2 variant verlies van geur verlies van smaak	

TABLE B.7: Search terms used for tweets geotagged to the Netherlands in the analysis dataset

B.4.5 Poland

Policy	Disease	Vaccine
bliski kontakt edukacja na odległość epidemia hospitalizacja izolacja kwarantanna lockdown lockdowny maski nauka online oddział intensywnej terapii oddziały intensywnej terapii pandemia pcr polityka fiskalna polityka monetarna pomoc dla biznesu praca zdalna samoizolacja śledzenie kontaktów środki ochrony osobistej sytuacja zagrożenia zdrowia publicznego szczepienie szybki test szybki test antygenowy telemedycyna test antygenowy test diagnostyczny test na przeciwciała test serologiczny test wymazowy testy testy na przeciwciała testy wymazowe zasilek dla bezrobotnego	bez smaku brak zapachu ciągły kaszel covid covid-19 duszność korona korona wirus koronawirus respiratory rona sars-cov-2 trudności w oddychaniu utrata smaku utrata zapachu wariant beta wariant brytyjski wariant delta wariant omicron warianty wysoka temperatura	astrazeneca booster dawka przypominająca paszport szczepionkowy szczepionka szczepionka biontech szczepionka moderna szczepionka pfizera

B.4.6 Portugal

Policy	Disease	Vaccines
achatar a curva antigénio capacidade de resposta carga viral confinamento contacto próximo contato próximo contenção de carro distanciamento social drive-through emergência de saúde pública ensino à distância equipamento de proteção fadiga pandémica imunidade de grupo isolamento isolamento voluntário máscara pcr protecção individual quarentena r0 rastreamento recolher obrigatório resposta à pandemia restrição de viagem taxa de infeção taxa de mortalidade taxa de reprodução teletrabalho testar e rastrear teste antigénio teste de diagnóstico teste pcr teste rápido teste serológico trabalhador essencial trabalho a partir de casa transmissão comunitária triagem	corona coronavírus covid covid-19 falta de ar falta de olfato falta de sabor febre perda de olfato perda de sabor sars-cov-2 sem cheiro sem gosto temperatura alta variante variante delta+ variante omicron ventilador	astra astrazeneca da universidade de oxford dose de reforço passaporte da vacina reforço da vacina segunda dose da vacina terceira dose vacina vacina da moderna vacina da pfizer vacina de oxford vacinação
Continued on next page		

Policy	Disease	Vaccine
--------	---------	---------

TABLE B.9: Search terms used for tweets geotagged to Portugal in the analysis dataset

B.4.7 Spain

Policy	Disease	Vaccine
aforo aforos aislamiento aplanar la curva autoaislamiento ayuda económica carga viral certificado covid cierre de la actividad no esencial cierre de la hostelería cierre perimetral confinamiento contacto contacto cercano contagio corredor turístico crisis del coronavirus crisis sanitaria cuarentena cuaretenas cuarta ola distancia social el palito enfermedad covid-19 equipo de protección per- sonal estado de alarma exudado nasofaríngeo fondo de recuperación guante inmunidad de rebaño intervenciones no farma- cológicas mascarilla medida de prevención	corona coronavirus covid covid19 dificultad para respirar dolor de pecho escalofrío falta de aire fatiga fiebre pérdida del gusto pérdida del olfato sars-cov-2 sin gusto sin olfato tos variante variante alfa variante beta variante de preocupación variante delta variante delta plus variante omicron ventilador	aztrazeneca centro de vacunación dosis dosis de refuerzo pasaporte covid pasaportes covid tasa de vacunación vacuna vacuna de moderna vacuna de oxford vacuna de pfizer vacuna de refuerzo vacuna moderna vacuna oxford vacuna pfizer vacunación

Continued on next page

Policy	Disease	Vaccine
número de reproducción pcr plan de vacunación prueba pcr pruebas r0 rastreo rastreo de contactos rescate financiero restricción riesgo de contagio semáforo covid tasa de contagios tasa de incidencia tasa de reproducción teletrabajar teletrabajo tercera ola test de anticuerpos test de antígenos test diagnóstico test pcr test rápido test serológico tests toque de queda transición epidemiológica transmisión uso de mascarillas vacunación masiva		

TABLE B.10: Search terms used for tweets geotagged to Spain in the analysis dataset

B.4.8 United Kingdom

Policy	Disease	Vaccine
antibody test chris whitty circuit breaker close contact contact tracing covid app covid marshals	alpha variant beta variant continuous cough corona coronavirus covid covid-19	astrazeneca biontech booster shot moderna pfizer pfizer biontech pfizer vaccine
<i>Continued on next page</i>		

Policy	Disease	Vaccine
covid-19 secure guidelines	delta	vaccine
diagnostic test	delta plus	vaccine booster
drive-thru testing	delta plus variant	vaccine oxford
essential workers	delta variant	vaccine passport
flattening the curve	difficulty breathing	
furlough	high temperature	
herd immunity	kent variant	
isolation	loss of smell	
jonathan van-tam	loss of taste	
jvt	no smell	
lateral flow	no taste	
lf test	omicron	
lf tests	omicron variant	
lft	rona	
lockdown	sars-cov-2	
mask	shortness of breath	
mass testing	variant of concern	
nhs	ventilator	
nhs app		
nightingale hospitals		
non-pharmaceutical inter-		
vention		
npis		
patrick vallance		
pcr		
personal protective equip-		
ment		
plan a		
plan b		
ppe		
quarantine		
r rate		
r0		
rapid test		
roadmap		
rule of six		
rule of two		
screening		
self-isolation		
serology test		
shielding		
social distancing		
Continued on next page		

Policy	Disease	Vaccine
state of emergency stay alert stay at home order step 1 step 2 step 3 support bubbles swab test test to release tier 1 tier 2 tier 3 tier 4 tier system track and trace travel corridors vaccine rollout viral load work from home		

TABLE B.11: Search terms used for tweets geotagged to the United Kingdom in the analysis dataset

B.5 Plural mapping

The tables below present the plural-to-singular search term mapping used for each country (in their official language).

B.5.1 France

Plural	Singular
cas contacts	cas contact
masques	masque
pass sanitaires	pass sanitaire
passe sanitaire	passes sanitaires
protocoles sanitaires	protocole sanitaire
tests antigéniques	test antigénique
tests diagnostiques	test diagnostique
tests nasopharyngés	test nasopharyngé
tests rapides	test rapide
tests sérologiques	test sérologique
vaccinations de masse	vaccination de masse
vaccinations obligatoires	vaccination obligatoire
<i>Continued on next page</i>	

Plural	Singular
vaccins	vaccin
variants	variant
ventilateurs	ventilateur
équipements de protection person- nelles	équipement de protection person- nelle

TABLE B.12: Plural-to-singular mapping for tweets geotagged to France

B.5.2 Germany

Plural	Singular
abstrichtests	abstrichtest
antigen-schnelltests	antigen-schnelltest
antigen-selbsttests	antigen-selbsttest
antigen-tests	antigen-test
auffrischimpfungen	auffrischimpfung
diagnostische tests	diagnostischer test
digitale impfnachweise	digitaler impfnachweis
gesichtsmasken	gesichtsmaske
gesundheitsmaßnahmen	gesundheitsmaßnahme
impfstoffe	impfstoff
impfstrategien	impfstrategie
kontaktbeschränkungen	kontaktbeschränkung
masken	maske
maskenpflichten	maskenpflicht
mutationen	mutation
reisebeschränkungen	reisebeschränkung
risikobewertungen	risikobewertung
risikokommunikationen	risikokommunikation
schnelltests	schnelltest
schnutenpullis	schnutenpulli
seitliche flusstests	seitlicher flusstest
serologische tests	serologischer test
teststrategien	teststrategie
varianten	variante
virale belastungen	virale belastung

TABLE B.13: Plural-to-singular mapping for tweets geotagged to Germany

B.5.3 Italy

Plural	Singular
annullamenti	annullamento
distanziamenti sociali	distanziamento sociale
mascherine	mascherina
passaporti sanitari	passaporto sanitario
periodi di isolamento	periodo di isolamento
piani vaccinali	piano vaccinale
rinvii	rinvio
sanificazioni	sanificazione
tampone	tamponi
tamponi di massa	tamponaggio di massa
test rapidi	test rapido
test sierologici	test sierologico
vaccinazioni di massa	vaccinazione di massa
zona arancione	zone arancioni
zona gialla	zone gialle
zona rossa	zone rossa
zona rossa, arancione, gialla	zone rossa, arancione, gialla

TABLE B.14: Plural-to-singular mapping for tweets geotagged to Italy

B.5.4 Netherlands

Plural	Singular
antilichaamtesten	antilichaamtest
basispakketten	basispakket
beademingsapparaten	beademingsapparaat
boostervaccinaties	boostervaccinatie
contactberoepen	contactberoep
coronamaatregelen	coronamaatregel
diagnostische tests	diagnostische test
fieldlab experimenten	fieldlab experiment
If tests	If test
mondkapjes	mondkapje
noodwetten	noodwet
persconferenties	persconferentie
persoonlijke beschermingsmidde-	persoonlijk beschermingsmiddel
len	
serologische tests	serologische test
sneltesten	sneltest
steunpakketten	steunpakket
testbewijzen	testbewijs
uitstrijkjes	uitstrijkje
vaccinatiebewijzen	vaccinatiebewijs
<i>Continued on next page</i>	

Plural	Singular
vaccinatiestrategieën	vaccinatiestrategie
vaccins	vaccin
varianten	variant
varianten van zorg	variant van zorg
veiligheidsregio's	veiligheidsregio
zorgbonussen	zorgbonus

TABLE B.15: Plural-to-singular mapping for tweets geotagged to the Netherlands

B.5.5 Poland

Plural	Singular
hospitalizacje	hospitalizacja
maski	maska
paszporty szczepień covid-19	paszport szczepień covid-19
paszporty szczepionkowe	paszport szczepionkowy
polityki fiskalne	polityka fiskalna
polityki monetarne	polityka monetarna
respirator	respiratory
szczepienia	szczepienie
szczepionki	szczepionka
szybkie testy	szybki test
szybkie testy antygenowe	szybki test antygenowy
testy antygenowe	test antygenowy
testy diagnostyczne	test diagnostyczny
testy na przeciwciała	test na przeciwciała
testy przeciwciałowe	test przeciwciałowy
testy serologiczne	test serologiczny
warianty zainteresowania	wariant zainteresowania
wirusowe obciążenia	wirusowe obciążenie
zasiłki dla bezrobotnych	zasiłek dla bezrobotnego

TABLE B.16: Plural-to-singular mapping for tweets geotagged to Poland

B.5.6 Portugal

Plural	Singular
antigénios	antigénio
eventos superpropagadores	evento superpropagador
febres	febre
intervenções não farmacêuticas	intervenção não farmacêutica

Continued on next page

Plural	Singular
máscaras	máscara
passaportes da vacina	passaporte da vacina
restrições de viagem	restrição de viagem
temperaturas altas	temperatura alta
testes antigénio	teste antigénio
testes de anticorpos	teste de anticorpos
testes de diagnóstico	teste de diagnóstico
testes de esfregaço	teste de esfregaço
testes drive-thru	teste drive-thru
testes pcr	teste pcr
testes rápidos	teste rápido
testes sorológicos	teste serológico
trabalhadores essenciais	trabalhador essencial
vacinas	vacina
vacinações	vacinação
variantes	variante
ventiladores	ventilador

TABLE B.17: Plural-to-singular mapping for tweets geotagged to Portugal

B.5.7 Spain

Plural	Singular
aislamientos	aislamiento
centros de vacunación	centro de vacunación
certificados covid	certificado covid
cierres de la actividad no esencial	cierre de la actividad no esencial
cierres de la hostelería	cierre de la hostelería
cierres perimetrales	cierre perimetral
confinamientos	confinamiento
contactos	contacto
contactos cercanos	contacto cercano
contagios	contagio
corredores turísticos	corredor turístico
cuarentenas	cuarentena
escalofríos	escalofrío
fiebres	fiebre
fondos de recuperación	fondo de recuperación
guantes	guante
maskarillas	maskarilla
medidas de prevención	medida de prevención
planes de vacunación	plan de vacunación
restricciones	restricción

Continued on next page

Plural	Singular
tests de anticuerpos	test de anticuerpos
tests de antígenos	test de antígenos
tests diagnósticos	test diagnóstico
tests rápidos	test rápido
tests serológicos	test serológico
transiciones epidemiológicas	transición epidemiológica
vacunaciones masivas	vacunación masiva
vacunas	vacuna
vacunas de refuerzo	vacuna de refuerzo
variantes	variante
ventiladores	ventilador

TABLE B.18: Plural-to-singular mapping for tweets geotagged to Spain

B.5.8 United Kingdom

Plural	Singular
antibody tests	antibody test
booster shots	booster shot
diagnostic tests	diagnostic test
masks	mask
non-pharmaceutical interventions	non-pharmaceutical intervention
rapid tests	rapid test
serology tests	serology test
swab tests	swab test
vaccines	vaccine
variants of concern	variant of concern
ventilators	ventilator

TABLE B.19: Plural-to-singular mapping for tweets geotagged to the United Kingdom

B.6 Catch-all terms

France, Germany, Italy, Netherland, Portugal, UK	Spain	Poland
corona corona virus covid covid-19 sars-cov-2 rona	corona corona virus covid covid19 sars-cov-2 enfermedad covid-19	korona korona wirus covid covid-19 sars-cov-2 rona koronawirus

TABLE B.20: Catch-all terms used during categorisation. Six countries had the exact same catch-all terms as indicated by first column name.

C Appendix: identifying the best approach to multilingual sentiment classification and practical recommendations

C.1 Annotation

C.1.1 Annotator details

There were 10 annotators in total. The demographics of the annotators were as follows: 90% female, 90% white and 10% were mixed race. For English annotators, the annotators had diverse backgrounds - 1 England National Health Service worker, 1 student, and 1 department of public health postgraduate. The remaining annotators were all either department of public health postgraduates or staff members and were considered to have expert knowledge of the countries whose tweets they annotated. All annotators were native speakers of the language they annotated.

C.1.2 Guidelines

In the guidelines below, `{{COUNTRY}}` was replaced with one of: France, Germany, Italy, The Netherlands, Poland, Portugal, Spain, or The United Kingdom

Sentiment

You are given `{{N}}` tweets, posted by users who have geotagged their location as `{{COUNTRY}}`. You are asked to annotate each tweet based on its sentiment and emotion - you can find more details for each of the tasks below.

Warning: The sample of tweets to annotate may contain sensitive content such as offensive words or swear words.

You are asked to annotate each one based on its sentiment, with three potential category-label pairs:

For each tweet, you should assign the single label that best categorises the text; for example, In cases where a tweet conveys both positive and negative sentiments, the

Category	Label
Negative	-1
Neutral	0
Positive	1

annotation should be done based on the predominant sentiment. You should try to annotate all tweets in the sample, even in the cases with limited context or hard to decide.

Detailed description for each class

- Negative (-1): tweets which attack a person, group, product or policy, express disgust or unhappiness towards something, or criticize something.
- Positive (1): tweets which express happiness, praise a person, group, country or product, or applaud something.
- Neutral (0): tweets that state facts, give news or are advertisements. In general, those which don't fall into the above 2 categories.

Extra information

There are three additional pieces of information that should be captured:

1. Is the tweet easy to annotate? A single column is provided for both sentiment and emotion. Difficulty in annotation includes both cases where there is not enough context to provide an annotation or where it is difficult to determine the appropriate category.
2. Does a tweet relate to the `{{COUNTRY}}` or not.
3. Does a tweet relate to the COVID-19 pandemic or not.

The labels and column names related to each are provided below.

1. Column: Hard-to-annotate

Category	Label
Hard to annotate	1
Easy to annotate	Leave blank

2. Column: `{{COUNTRY}}`-related

Category	Label
Not {{COUNTRY}} related	-1
{{COUNTRY}} related	1
Not sure	Leave blank

3. Column: COVID-related

Category	Label
Not COVID related	-1
COVID related	1
Not sure	0

Notes:

- The labelling should be done only by the information available in the text, i.e. do not follow links present in a tweet.
- In cases where a tweet only includes news titles, try to assess the sentiment of the shared news.
- If a tweet does not relate to COVID, do not label sentiment, but still do label whether it is related to {{COUNTRY}} or not.

C.2 Prompts

This Appendix includes prompts in English only; however, the prompts for other countries are identical in content, differing only in language and the substitution of COUNTRY with the relevant country in the annotation prompt; that is, one of: France, Germany, Italy, The Netherlands, Poland, Portugal, Spain, or The United Kingdom. The full set of prompts in all languages can be found in this (GitHub) file: [prompts.yml](#).

For prompts with examples, the tweet example used in the static configuration is indicated by (static). In the tweet example in the dynamic configuration is indicated by (dynamic). In this configuration, {{NEG_TWEET}}, {{NEU_TWEET}}, and {{POS_TWEET}} are replaced by country-specific tweets.

C.2.1 Zero-shot

You're given a tweet about COVID-19. Classify it to one of the following categories: NEGATIVE, NEUTRAL, or POSITIVE.

Provide only a single word, the relevant category, as your response.

C.2.2 Few-shot

You're given a tweet about COVID-19. Classify it to one of the following categories: NEGATIVE, NEUTRAL, or POSITIVE.

Provide only a single word, the relevant category, as your response.

###

Example 1:

(static)

Tweet: Well what a very shitty day and a waste of a bank holiday...

I would honestly rather have gone to work!! Damn you rain and
lockdown

(dynamic)

Tweet: {{NEG_TWEET}}

Label: NEGATIVE

###

Example 2:

(static)

Tweet: @crawleytown is a lateral flow test needed for tonight's game?

(dynamic)

Tweet: {{NEU_TWEET}}

Label: NEUTRAL

###

Example 3:

(static)

Tweet: So happy to see the sun shining again today!!!

Social distancing street party gig goes ahead in St Helens this

afternoon!!!

Happy Saturday peeps

(dynamic)

Tweet: {{POS_TWEET}}

Label: POSITIVE

C.2.3 Annotation

You are given a tweet, posted by users who have geo-tagged their location as the {{COUNTRY}}.

You are asked to annotate the tweet based on its sentiment into one of three categories: NEGATIVE, NEUTRAL, or POSITIVE.

{{Line below is removed in response length experiment}}

Provide only a single word, the relevant category, as your response.

Detailed descriptions are given for each class below:

NEGATIVE: Tweets which attack a person, group, product or policy, express disgust or unhappiness towards something, or criticize something.

POSITIVE: Tweets which express happiness, praise a person, group, country or product, or applaud something.

NEUTRAL : Tweets that state facts, give news or are advertisements. In general, those which don't fall into the above 2 categories.

For each tweet, you should assign the single label that best categorises the text; for example, In cases where a tweet conveys both positive and negative sentiments, the annotation should be done based on the predominant sentiment. You should try to annotate all tweets in the sample, even in the cases with limited context or hard to decide.

Here are some additional notes for when you label the tweet:

- The labeling should be done only by the information available in the text, i.e. do not follow links present in a tweet.
- In cases where a tweet only includes news titles, try to assess the

sentiment of the shared news title

- All urls have been removed and have been replaced by http
- Only the handles of verified users are shown, all other users are anonymised and their handle is replaced with @user

###

Example 1:

(static)

Tweet: Well what a very shitty day and a waste of a bank holiday...

I would honestly rather have gone to work!! Damn you rain and lockdown

(dynamic)

Tweet: {{NEG_TWEET}}

Label: NEGATIVE

###

Example 2:

(static)

Tweet: @crawleytown is a lateral flow test needed for tonight's game?

(dynamic)

Tweet: {{NEU_TWEET}}

Label: NEUTRAL

###

Example 3:

(static)

Tweet: So happy to see the sun shining again today!!!

Social distancing street party gig goes ahead in St Helens this afternoon!!!

Happy Saturday peeps

(dynamic)

Tweet: {{POS_TWEET}}

Label: POSITIVE

C.2.4 Prompts per fold

The prompts that replace {{NEG_TWEET}}, {{NEU_TWEET}}, and {{POS_TWEET}} in the United Kingdom dynamic prompts are given in the table below. The full set of tweets used in all languages can be found in this (GitHub) file: [tweets_per_fold.json](#).

Fold	Negative	Neutral	Positive
1	From Monday you can fly to twelve countries with no quarantine requirements on return but you can't sit inside a pub socially distanced in Glasgow. Nice one.	Station 19 star Danielle Savre donates eggs to help gay best friend have baby via surrogate: U.K. · Latest headlines Lilibet Diana Mountbatten-Windsor World News You Mag Event Books Mail Shop Property Horoscopes Blogs Motoring Covid-19. http	Anyone struggling with lockdown so far....honestly go out and get a walk in....does the world of good. I've been out everyday after I finish my work, and it helps massively.
2	I felt as if I live in a parallel universe this evening, watching the UK daily briefing. I knew about loss of sense of taste and smell months ago. #Coronavirus	"Face mask in "Seeing In The Dark". Reusable, washable, pleated,..." in #Rugby #UnitedKingdom http	2017! Forgot about this #BlackInTheUnionJack great to have seen the documentary again recently pre lockdown http

Continued on next page

Fold	Negative	Neutral	Positive
3	Why? When you have an underlying health problem, why would the relief of getting the vaccine not make you feel better? http	Anyone here is planning to jump to the digital nomadism lifestyle when all this covid situation is over? #digitalnomad	http Ed Sheeran has reportedly donated over £1 million to charities in his local area - #COVID19Pandemic #ThankYouNHS #coronavirus #covid19 #EdSheeranMusic @ed-sheeran #EdSheeran http http
4	mask Hard of hearing Tory Tim Montgomerie's not so casual, on message tweet regarding Sadiq Khan's choice to wear a mask has bred some strange responses, including the use of the impersonal pronoun for objects and very small animals. Disclaimer: Other mask wearers are available mask http	A strong article from @SundayTimesZA terrible affect on death numbers http #delayinlockdownbyUkGovt @BorisJohnson @UK-Labour @user	It's the little things that matter... Just handed a seafarer a tube of toothpaste. He had tears in his eyes as he thanked me and said that he had been praying for someone to come and help him today. pray #seafarers #EssentialWorkers #FrontLine-Heroes
5	Brilliant this No one spoke up when the dirty Socialists up north were put in lockdown but now its london suddenly it's important For the record, lockdowns were still the right thing to do tho http	A different perspective.. #coronavirus http	Lockdown's not so bad shrug @ Bicester, Oxfordshire http

TABLE C.1: Set of UK tweets used in each fold for dynamic prompts.

Prompt	Runs	Folds	Dataset size per run
Zero-shot	5	-	2430
Few-shot (static)	5	-	2430
Few-shot (dynamic)	1	5	486*
Annotation (static)	5	-	2430
Annotation (dynamic)	1	5	486*

TABLE C.2: Run configuration across the different prompting configurations.

*indicates that a different set of tweets is used per fold

C.3 Run details

C.4 Hyperparameter tuning

C.4.1 Quantisation configuration

The following `bitsandbytes` quantisation configuration was used:

- `load_in_4bit=True`
- `bnb_4bit_quant_type="nf4"`
- `bnb_4bit_compute_dtype=torch.bfloat16`
- `bnb_4bit_use_double_quant=True`

C.4.2 Run time

Model	Wall time	N trials
XLM-T _{SENT}	4h 58m 11s	75
Bernice	6h 44m 10s	75
XLM-T _{LARGE}	21h 16m 58s	75
Llama 3 8B	43h 21m 38s	25
Mistral v0.3 7B	49h 28m 28s	25

TABLE C.3: The total wall time in hours, minutes, and seconds to fine-tune each model. N trials refers to the number of hyperparameter trials run.

C.4.3 Encoder results

Model	Fold	LR	Batch	Epochs
Bernice	1	5.1e-5	16	11
	2	6.1e-6	8	6
	3	1.4e-4	64	3
	4	6.1e-6	8	6
	5	5.1e-5	16	11
XLM-T _{SENT}	1	1.8e-5	2	4
	2	6.1e-6	8	6
	3	1.4e-4	64	3
	4	5.1e-5	16	11
	5	9.3e-5	64	7
XLM-T _{LARGE}	1	1.9e-6	8	7
	2	1.8e-5	8	10
	3	1.9e-6	8	7
	4	1.8e-5	8	10
	5	1.8e-5	8	10

TABLE C.4: The best hyperparameter tuning configuration per encoder LM. LR refers to the learning rate.

C.4.4 LLM results

Model	Fold	LR (e-6)	Batch	Epochs	Temp	Drop	α	r
Llama 3 8B	1	9.5	2	8	0.53	0	256	32
	2	7.1	4	8	0.61	0	32	256
	3	2.2	4	6	0.61	0.1	256	128
	4	5.0	4	9	0.41	0	256	128
	5	3.3	2	9	0.97	0.1	16	64
Mistral v0.3 7B	1	5.2	2	10	0.47	0	8	16
	2	4.0	2	9	0.78	0.05	128	64
	3	5.6	2	12	0.16	0.05	256	128
	4	7.1	2	8	0.65	0.1	64	64
	5	3.8	8	6	0.16	0.1	256	8

TABLE C.5: The best hyperparameter tuning configuration for each LLM per fold. Drop which refers to dropout, α , and r are all LORA hyperparameters. LR refers to the learning rate and Temp refers to the LLM temperature.

C.5 Cost estimate

C.5.1 Encoder LMs and LLMs

Assuming compute is unavailable, I provide an estimate of running my experiments using cloud computing services. I emphasise that this is a crude estimate.

To obtain a cost estimate of running each model used in my experiment, I started by using this [hugging face model-memory-usage space](#) to obtain the memory requirement for each model. I inflated this by 20% to allow for additional overheads during inference, taking the ceiling to the nearest first decimal as the result. For Mistral v0.3 7B I obtained the GPU requirement from their [official guide](#) rather than using the hugging face tool.

Based on the memory requirement I found an appropriate GPU instance on Amazon Web Services. I identified the following instances and their [associated costs](#). I used on-demand costing which I acknowledge may not be the most efficient. Moreover, I acknowledge that there may be cheaper providers, especially in cases where you can customise the virtual machine you use.

- **g6e.xlarge**: 4 vCPUs (AMD EPYC 7R13), 32 GB (DDR4) 1 GPU with 48GB vRAM (NVIDIA L40S Tensor Core GPUs). The cost per hour is 1.861 USD. I use this instance to cost Llama 3 and 3.1.
- **g4dn.xlarge**: 4 vCPUs (Intel Cascade Lake P-8259C), 16 GB (DDR4) 1 GPU with 16GB vRAM (NVIDIA T4 Tensor Core). The cost per hour is 0.526 USD. I use this instance to cost the remaining models.

Using the above costs, I multiplied them by the wall times in my experiments. I excluded fine-tuned LLMs as these are more complicated to cost as the memory requirements depend on the quantisation strategy. The results of applying this process are given in Table C.6.

Model name	vRAM	I cost	T cost	FT cost
Llama 3.1	34	0.97	8.55	-
Llama 3	34	1.07	8.25	80.69
Mistral v0.3	16	0.36	2.86	92.07
XLM-T _{LARGE}	2.5	0.05	0.05	11.19
Bernice	1.3	0.002	0.002	3.54
XLM-T _{SENT}	1.3	0.002	0.002	2.61

TABLE C.6: The cost of running each model using AWS compute. I cost is the inference cost, for LLMs this is the cost for the single best performing tweet. T cost is the total cumulative cost across all tweets and runs and FT cost is the cost to fine-tune the model. The cost is given in USD for all columns. The vRAM requirements assume a batch size of 1.

Model name	Prompt	F1	Acc	Prec	Rec	C-o
Mistral v0.3 ^{FT} 7B	Annotation (dynamic)	0.67	0.69	0.73	0.69	0.02
ChatGPT 4o-mini	Few-shot (dynamic)[eng]	0.67	0.71	0.70	0.67	0.02
Llama 3.1 8B	Annotation (dynamic)[eng]	0.67	0.71	0.70	0.68	0.05
ChatGPT 4o-mini	Annotation (static)	0.66	0.71	0.74	0.66	0.02
Llama 3 ^{FT} 8B	Annotation (dynamic)[eng]	0.66	0.68	0.67	0.69	0.06
ChatGPT 4o-mini	Few-shot (static)	0.66	0.70	0.72	0.65	0.03
ChatGPT 4o-mini	Few-shot (dynamic)	0.66	0.70	0.70	0.66	0.03
Llama 3.1 8B	Annotation (static)[eng]	0.66	0.69	0.69	0.65	0.04
Mistral v0.3 ^{FT} 7B	Annotation (static)[eng]	0.65	0.67	0.69	0.65	0.02
Llama 3 8B	Annotation (dynamic)	0.65	0.68	0.69	0.68	0.05
ChatGPT 4o-mini	Few-shot (static)[eng]	0.65	0.69	0.71	0.64	0.02
ChatGPT 4o-mini	Annotation (static)[eng]	0.64	0.69	0.74	0.65	0.02
Llama 3 8B	Annotation (dynamic)[eng]	0.64	0.69	0.68	0.65	0.03
Mistral v0.3 ^{FT} 7B	Few-shot (static)[eng]	0.64	0.65	0.67	0.63	0.03
ChatGPT 4o-mini	Zero-shot	0.64	0.68	0.67	0.63	0.04
Llama 3 ^{FT} 8B	Annotation (dynamic)	0.64	0.68	0.69	0.67	0.08
Llama 3 8B	Annotation (static)	0.64	0.68	0.69	0.64	0.03
Mistral v0.3 7B	Annotation (dynamic)[eng]	0.63	0.66	0.65	0.65	0.04
Llama 3 ^{FT} 8B	Few-shot (dynamic)[eng]	0.63	0.66	0.64	0.64	0.04
Mistral v0.3 ^{FT} 7B	Few-shot (dynamic)	0.63	0.65	0.66	0.66	0.05
Mistral v0.3 ^{FT} 7B	Zero-shot[eng]	0.63	0.64	0.64	0.64	0.04
Llama 3.1 8B	Few-shot (dynamic)[eng]	0.63	0.67	0.64	0.64	0.05
Mistral v0.3 ^{FT} 7B	Few-shot (static)	0.62	0.65	0.65	0.63	0.03
Mistral v0.3 ^{FT} 7B	Zero-shot	0.62	0.65	0.64	0.64	0.05
ChatGPT 4o-mini	Zero-shot[eng]	0.62	0.68	0.68	0.61	0.04
Llama 3.1 8B	Annotation (dynamic)	0.62	0.68	0.68	0.63	0.03
Mistral v0.3 7B	Annotation (static)[eng]	0.62	0.65	0.66	0.62	0.03
Llama 3 8B	Annotation (static)[eng]	0.62	0.66	0.68	0.62	0.03
Mistral v0.3 7B	Few-shot (static)[eng]	0.62	0.65	0.63	0.62	0.05
Mistral v0.3 7B	Few-shot (dynamic)[eng]	0.61	0.65	0.63	0.63	0.05
Mistral v0.3 ^{FT} 7B	Annotation (static)	0.61	0.64	0.70	0.63	0.03
Llama 3 8B	Few-shot (dynamic)[eng]	0.61	0.67	0.64	0.62	0.04
Mistral v0.3 7B	Annotation (dynamic)	0.61	0.65	0.65	0.63	0.04
Mistral v0.3 7B	Annotation (static)	0.61	0.65	0.66	0.62	0.04
Llama 3 ^{FT} 8B	Annotation (static)	0.60	0.65	0.63	0.60	0.05

Continued on next page

Model name	Prompt	F1	Acc	Prec	Rec	C-o
Llama 3 8B	Few-shot (static)	0.60	0.65	0.64	0.62	0.05
Llama 3 ^{FT} 8B	Few-shot (static)[eng]	0.60	0.64	0.62	0.61	0.05
Mistral v0.3 7B	Few-shot (static)	0.60	0.64	0.62	0.61	0.05
Llama 3 ^{FT} 8B	Few-shot (static)	0.60	0.65	0.63	0.64	0.08
Llama 3 8B	Zero-shot[eng]	0.59	0.64	0.62	0.59	0.05
XLM-T _{SENT}	-	0.59	0.63	0.62	0.60	0.07
Mistral v0.3 7B	Zero-shot[eng]	0.59	0.63	0.59	0.61	0.06
Llama 3 8B	Few-shot (static)[eng]	0.59	0.64	0.64	0.59	0.04
Llama 3 ^{FT} 8B	Annotation (static)[eng]	0.58	0.63	0.61	0.59	0.06
Mistral v0.3 7B	Few-shot (dynamic)	0.58	0.63	0.61	0.60	0.05
Llama 3 8B	Few-shot (dynamic)	0.58	0.65	0.63	0.60	0.06
Llama 3.1 8B	Few-shot (static)[eng]	0.58	0.63	0.65	0.59	0.03
Llama 3.1 8B	Annotation (static)	0.58	0.66	0.70	0.59	0.03
Llama 3.1 8B	Zero-shot[eng]	0.58	0.64	0.63	0.58	0.05
Llama 3 ^{FT} 8B	Few-shot (dynamic)	0.57	0.62	0.59	0.60	0.07
Llama 3.1 8B	Few-shot (static)	0.57	0.63	0.66	0.58	0.04
Llama 3 8B	Zero-shot	0.57	0.62	0.60	0.58	0.06
Mistral v0.3 7B	Zero-shot	0.56	0.61	0.59	0.58	0.07
Llama 3 ^{FT} 8B	Zero-shot	0.56	0.60	0.57	0.58	0.06
Llama 3 ^{FT} 8B	Zero-shot[eng]	0.56	0.61	0.59	0.56	0.04
Llama 3.1 8B	Few-shot (dynamic)	0.54	0.61	0.61	0.56	0.04
VADER	-	0.48	0.50	0.51	0.52	0.18
Llama 3.1 8B	Zero-shot	0.46	0.56	0.55	0.50	0.07

TABLE C.8: Macro-averaged classification metrics of applying each model to the annotated dataset, sorted by F1. Columns from Acc (left to right): accuracy, precision, recall, and proportion of cross-over misclassification between negative and positive.

C.6.2 Best model results without Dutch and Polish

To ensure that my results were not biased by languages not officially supported by Llama 3.1, I recalculated performance metrics after excluding these languages. The updated results are presented in Table C.9 below. This exclusion had no effect on the relative F1 scores of the Llama models. However, the F1 score for both Llama 3.0 and 3.1 reduced by 0.01. Notably, the optimal instruction language for the fine-tuned Llama 3 model shifted to multilingual, suggesting that its previous preference for English prompts may have been influenced by poor performance on the excluded languages when using native-language instructions. Across the remaining models, the

impact was similarly limited; however, I highlight the increase in F1 score for XLM-T_{SENT} to 0.61, given its explicit lack of official support for the excluded languages.

Model name	Prompt	F1	Acc	Prec	Rec	C-o
XLM-T _{LARGE} ^{FT}	-	0.72	0.76	0.74	0.72	0.05
Bernice ^{FT}	-	0.71	0.76	0.72	0.71	0.06
Mistral v0.3 ^{FT} 7B	Annotation (dynamic)[eng]	0.69	0.71	0.70	0.70	0.04
ChatGPT 4o-mini	Annotation (dynamic)	0.69	0.73	0.74	0.69	0.02
XLM-T _{SENT} ^{FT}	-	0.68	0.73	0.68	0.68	0.06
Llama 3.1 8B	Annotation (static)[eng]	0.66	0.70	0.69	0.67	0.04
Llama 3 ^{FT} 8B	Annotation (dynamic)	0.66	0.70	0.69	0.69	0.05
Llama 3 8B	Annotation (dynamic)	0.64	0.68	0.68	0.67	0.04
Mistral v0.3 7B	Annotation (dynamic)[eng]	0.63	0.66	0.65	0.65	0.04
XLM-T _{SENT}	-	0.61	0.65	0.63	0.62	0.07
VADER	-	0.48	0.49	0.50	0.52	0.18

TABLE C.9: Macro-averaged classification metrics of applying each model to the annotated dataset excluding Dutch and Polish, sorted by F1. Note that only the results are filtered, the fine-tuned models are the same and have been fine-tuned on all languages. Columns from Acc (left to right): accuracy, precision, recall, and proportion of cross-over misclassification between negative and positive.

C.7 Model details

Model name	Link	Citation
Llama 3.1	meta-llama/Llama-3.1-8B	Grattafiori et al., 2024
Llama 3	meta-llama/Meta-Llama-3-8B	
Mistral v0.3	mistralai/Mistral-7B-Instruct-v0.3	Jiang et al., 2023
XLM-T _{LARGE}	cardiffnlp/twitter-xlm-roberta-large-2022	Barbieri, Espinosa Anke, and Camacho-Collados, 2022;
Bernice	jhu-clsp/bernice	Loureiro et al., 2022
XLM-T _{SENT}	cardiffnlp/twitter-xlm-roberta-base-sentiment	DeLucia et al., 2022
ChatGPT 4o-mini	models/gpt-4o-mini	Barbieri, Espinosa Anke, and Camacho-Collados, 2022
VADER	cjhutto/vaderSentiment	Brown et al., 2020
		Hutto and Gilbert, 2014

TABLE C.10: Links and citations of the models used. All links refer to implementations on HuggingFace, except for ChatGPT-4o-mini, for which I provide a link to the OpenAI model page, and VADER, for which I provide a link to the GitHub repository of the implementation I used.

Appendix: temporal evolution of tweet variables and their correlation with epidemiological and public policy variables

D.1 Most popular terms

	Before (<) 2020-11-09			After ($\geq$) 2020-11-00		
Country	Search term	Ratio	Prop	Search Term	Ratio	Prop
France	confinement (lockdown)	1.09	0.29	vaccin (vaccine)	1.01	0.20
	masque (mask)	1.01	0.20	masque (mask)	1.00	0.12
	déconfinement (lifting lockdown)	1.24	0.04	confinement (lockdown)	1.07	0.10
	crise sanitaire (health crisis)	1.08	0.02	pass sanitaire (health pass)	1.33	0.06
	télétravail (remote working)	1.08	0.02	variant	1.65	0.02
Germany	maske (mask)	1.10	0.12	impfung (vaccination)	1.04	0.12
	abstand (distancing)	1.05	0.06	maske (mask)	1.08	0.10
	lockdown	1.25	0.04	lockdown	1.20	0.09
	quarantäne (quarantine)	1.57	0.04	impfstoff (vaccine)	1.32	0.06
	maskenpflicht (mask mandate)	1.31	0.03	abstand (distancing)	1.11	0.04
	quarantena (quarantine)	1.40	0.11	vaccino (vaccine)	1.04	0.20

Continued on next page

	Before 2020-11-01			After 2020-11-01		
Country	Search term	Ratio	Prop	Search Term	Ratio	Prop
Italy	maschera (mask)	1.24	0.09	green pass	1.38	0.08
	lockdown	1.09	0.06	tamponi (swabs)	2.10	0.07
Italy	tamponi (swabs)	1.99	0.05	lockdown	1.52	0.05
	dpcm (Prime Ministerial Decree)	2.66	0.03	maschera (mask)	2.29	0.04
Netherlands	mondkapje (face mask)	1.09	0.12	vaccin (vaccine)	1.13	0.11
	horeca (hospitality industry)	1.04	0.05	avondklok (curfew)	2.03	0.08
	persconferentie (press conference)	2.24	0.05	mondkapje (face mask)	1.19	0.08
	coronamaatregel (corona measure)	1.32	0.04	horeca (hospitality industry)	1.09	0.06
	testen (to test)	1.14	0.04	testen (to test)	1.43	0.06
Poland	pandemia (pandemic)	1.48	0.09	szczepionka (vaccine)	1.76	0.18
	testy (test)	1.48	0.08	szczepienie (vaccination)	1.40	0.17
	epidemia (epidemic)	2.41	0.06	pandemia (pandemic)	1.67	0.11
	maski (masks)	2.33	0.05	testy (test)	2.02	0.06
	lockdown	2.09	0.02	lockdown	2.63	0.05
	quarentena (quarantine)	1.23	0.32	vacina (vaccine)	1.44	0.22
	máscara (mask)	1.21	0.13	máscara (mask)	1.74	0.12
Continued on next page						

	Before 2020-11-01			After 2020-11-01		
Country	Search term	Ratio	Prop	Search Term	Ratio	Prop
Portugal	confinamento (lockdown)	2.12	0.02	confinamento (lockdown)	1.98	0.10
	de carro (by car)	1.94	0.02	vacinação (vaccination)	2.08	0.06
	isolamento (isolation)	1.61	0.02	quarentena (quarantine)	1.77	0.03
Spain	mascarilla (mask)	1.04	0.13	vacuna (vaccine)	1.00	0.15
	cuarentena (quarantine)	1.11	0.12	mascarilla (mask)	1.00	0.11
	confinamiento (lockdown)	1.04	0.09	dosis (doses)	1.01	0.06
	estado de alarma (state of alarm)	1.17	0.05	pruebas (tests)	1.02	0.05
	contagio (contagion/ infection)	1.08	0.04	vacunación (vaccination)	1.05	0.04
UK	lockdown	1.07	0.25	lockdown	1.00	0.16
	nhs	1.00	0.14	nhs	1.00	0.13
	mask	1.00	0.08	vaccine	1.00	0.13
	social distancing	1.02	0.06	mask	1.00	0.09
	isolation	1.00	0.03	isolation	1.01	0.01

TABLE D.1: Top five most popular search terms, based on tweet volume, per country before and on or after 9 November 2020. Ratio represents the ratio of negative to positive tweet volume and prop is the proportion of tweet volume per search term relative to the total tweet volume per country for the period. Where applicable, the English translation of each term is provided in **typewriter** font below the term. The colour of the text indicates which search term category each search term belongs to according to the following key, **teal**: vaccine, **lavender**: disease, and black: policy.

D.2 Extended correlation results

D.2.1 Epidemiological variables and vaccination uptake

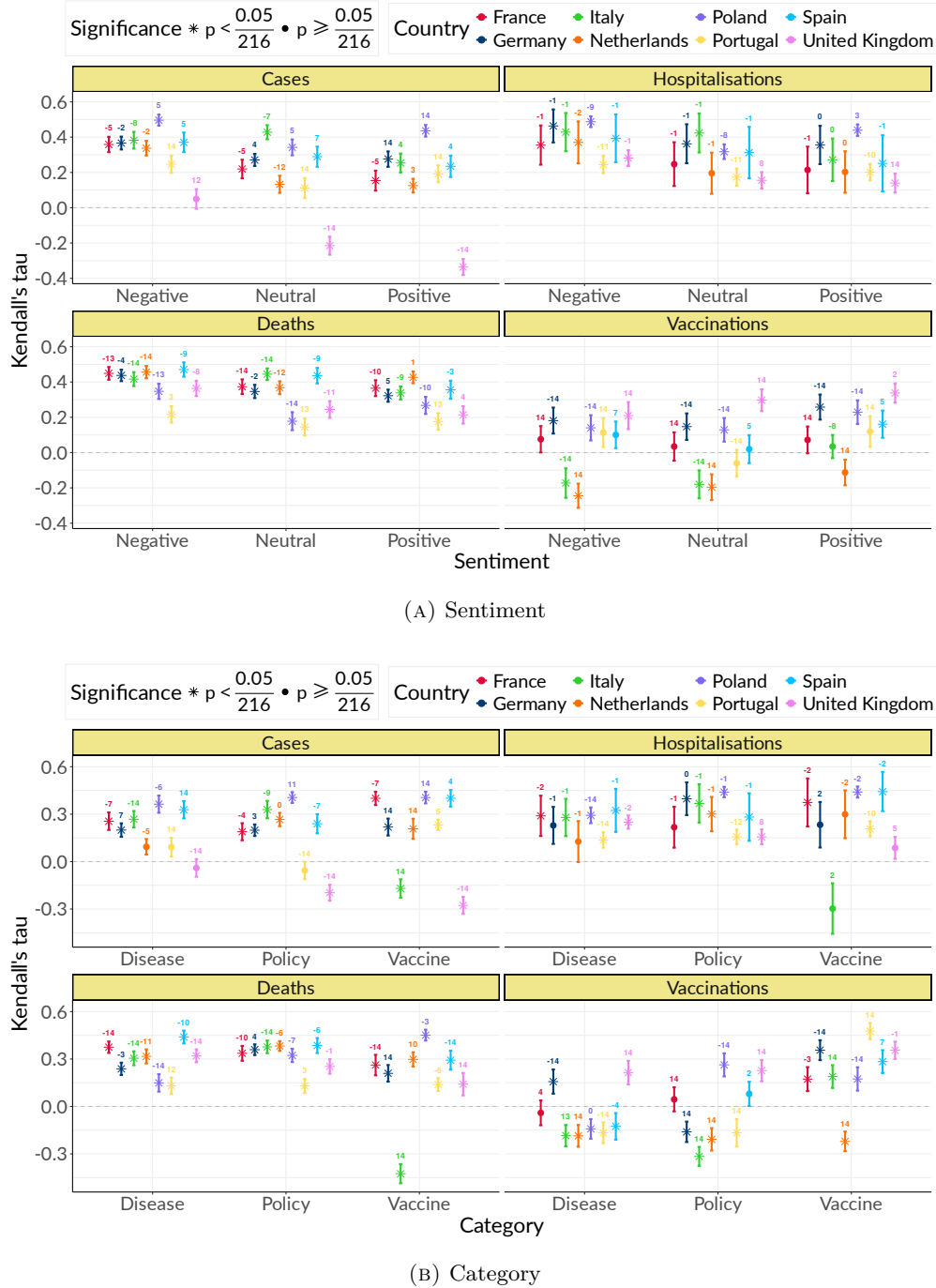


FIGURE D.1: Maximum Kendall's τ between tweet volume and epidemiological variables or vaccination uptake across a 14-day cross-correlation window, with 95% confidence intervals, coloured by country, stratified by (A) sentiment and (B) search term category. The lag at which the maximum is observed is superimposed. Positive values indicate tweet volume leads the outcome, while negative values indicate it lags.

D.2.2 Policy stringency variables

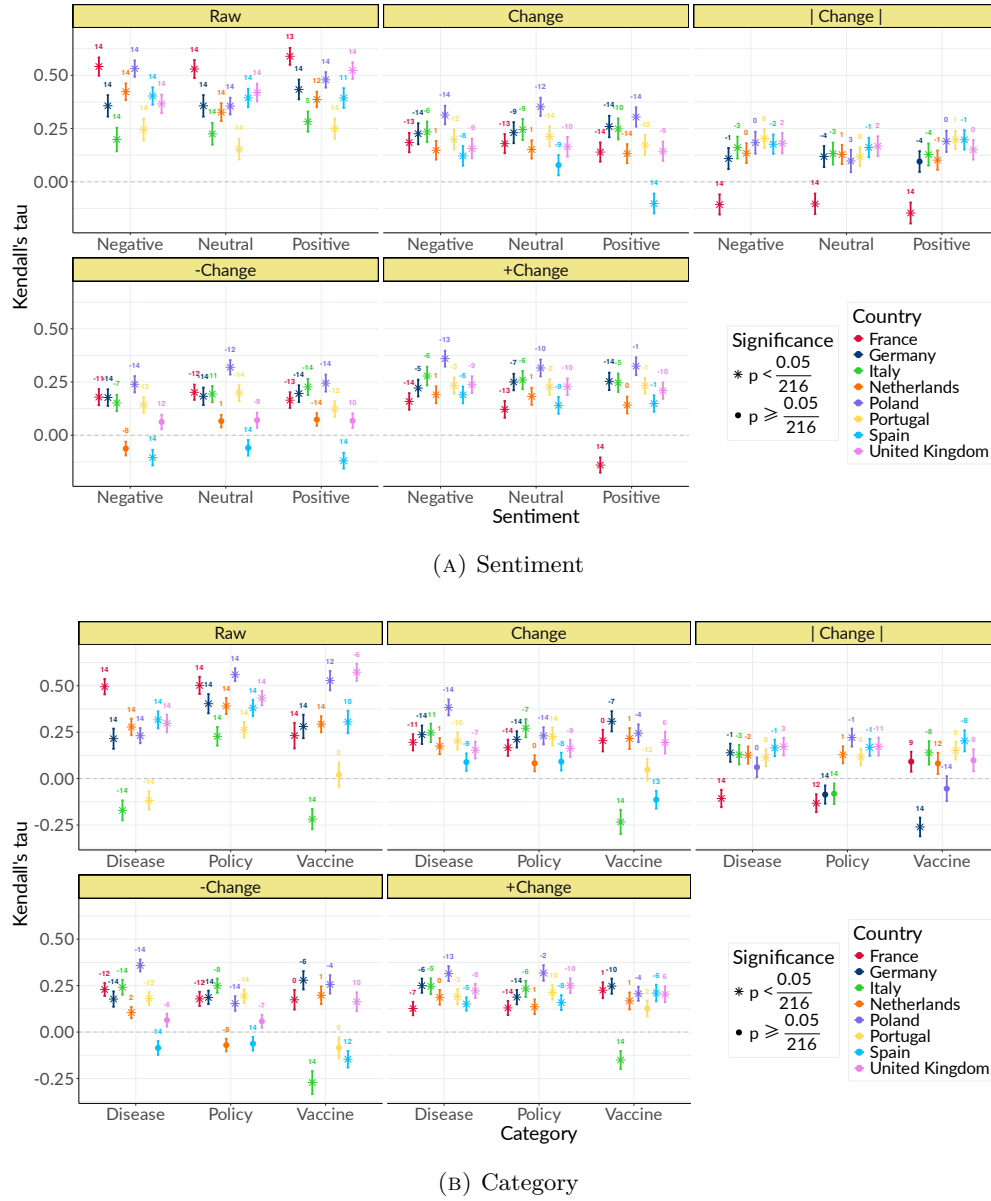


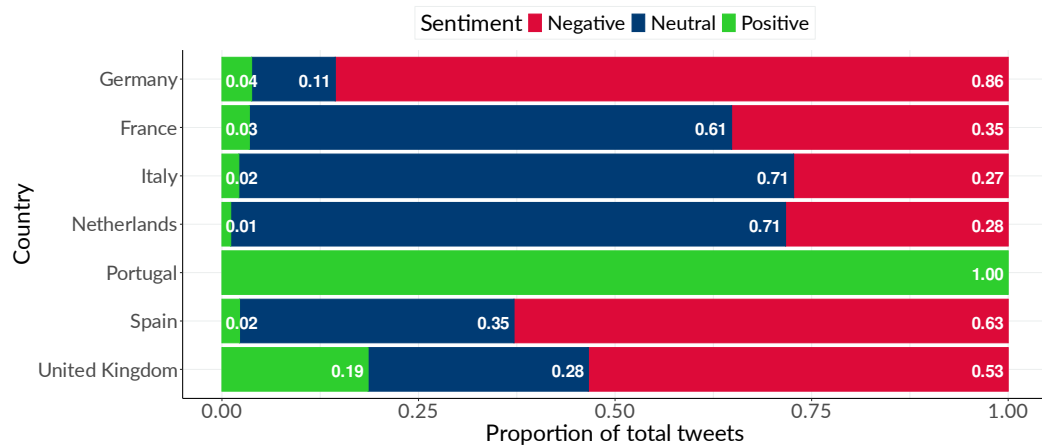
FIGURE D.2: Maximum Kendall's τ between tweet volume and policy stringency variables across a 14-day cross-correlation window with 95% confidence intervals, coloured by country, stratified by (A) sentiment and (B) search term category. The lag at which the maximum is observed is superimposed. Positive values indicate tweet volume leads the outcome, while negative values indicate it lags. **Raw** stringency refers to the seven-day MA-smoothed variable. **Change** denotes the one-day lagged difference of the smoothed profile, while **|Change|** represents its absolute value. **-Change** and **+Change** correspond to the negative and positive components of **Change**, respectively, with values set to zero in the absence of signed changes.

D.3 Sentiment of excluded tweets

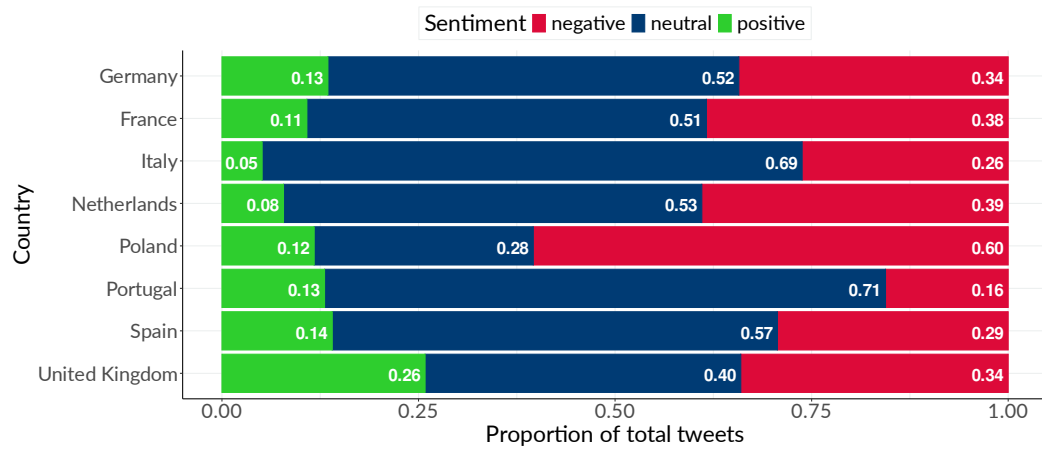
In Section 4.5.2, I outlined the process of removing tweets from bots and deduplicating content from potentially problematic accounts. To assess whether these tweets exhibited distinct sentiment profiles, and thereby support their exclusion, I calculated sentiment distributions for each group. I also examined sentiment for vaccine-related tweets made prior to 2020-11-09, which are also excluded. Table D.2 presents the results. The per-country distributions for each group are provided in Figure D.3, which illustrates that tweets removed as bots exhibited a lower proportion of positive sentiment, with most of the distribution concentrated in neutral sentiment. The deduplicated tweets similarly showed a higher proportion of neutral sentiment, primarily due to a reduction in positive tweets. Finally, vaccine-related tweets made before 9 November 2020 displayed a more negative sentiment profile compared to the full dataset, with a noticeably lower proportion of positive tweets.

Group	Size	Negative	Neutral	Positive
Bots removed	75 717	0.50	0.43	0.07
Tweets deduplicated	79 432	0.33	0.49	0.18
Vaccination tweets	55 981	0.59	0.27	0.14
Analysis dataset	5 098 443	0.52	0.26	0.22

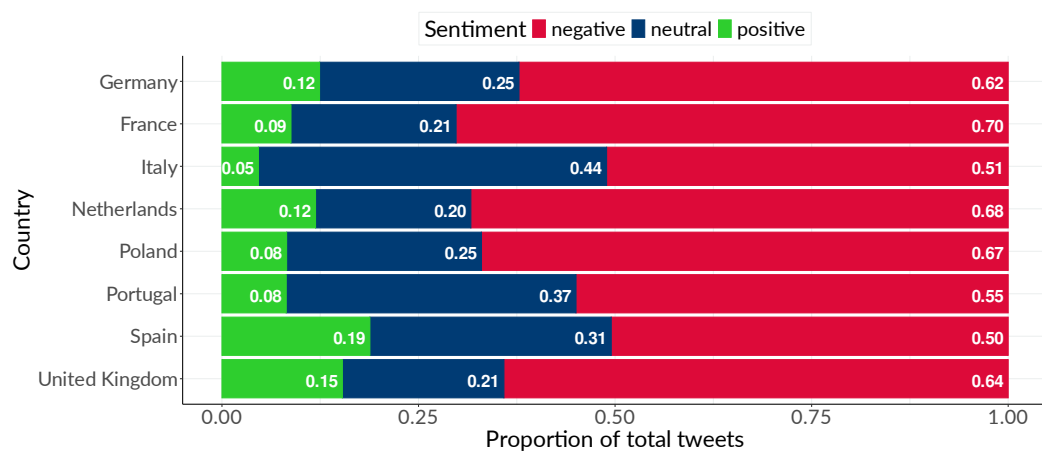
TABLE D.2: Sentiment polarity distribution for each of the three tweet groups excluded from the analysis. The final row shows the sentiment distribution across the analysis dataset for reference.



(A) Bots removed



(B) Tweets deduplicated



(C) Vaccine rows before 9 November 2020

FIGURE D.3: Distribution of sentiment polarity of (A–B) excluded problematic tweets and (C) excluded vaccine rows by country. In Panel A, the proportion of 1 for Portugal reflects a single observation.

D.4 Variant Data

I obtained variant data from the ECDC, including both TESSy submissions and records from the Global Initiative on Sharing All Influenza Data (GISAID) EpiCoV database (Khare et al., 2021). I used GISAID-sourced data, as it was more comprehensive and had an earlier starting date. I mapped WHO variant labels to Pango lineage names using data from Hodcroft (2021).

I defined the dominant variant as the one with the highest proportion on each day and applied a two-week rolling average to smooth transitions between dominant variants. As UK sequencing data were unavailable prior to 4 January 2021, I assigned the period before this to the Wild type. I generated the overall panel by averaging the percentage of each variant across all eight countries. Because the data are reported weekly, the plots display gaps at transition points between variants. The dominant variant by country and the smoothed series are shown in Figure D.4.



FIGURE D.4: Dominant variant by country over time, with shaded areas indicating the dominant period of each variant.

D.5 Vaccination uptake

Figure D.5 shows the weekly ECDC data in the top panel. The middle panel presents the raw daily OWID data, with the problematic period (5 December 2021 to 5 January 2022) marked by dashed red lines. A clear spike is visible during this period in the

raw OWID data, likely caused by a reporting error in which multiple days were aggregated into one. I redistributed these values according to the observed ECDC data, as described in more detail in the main text (Section 6.4.2). The result of this redistribution is shown in the final panel, labelled OWID Interpolated.

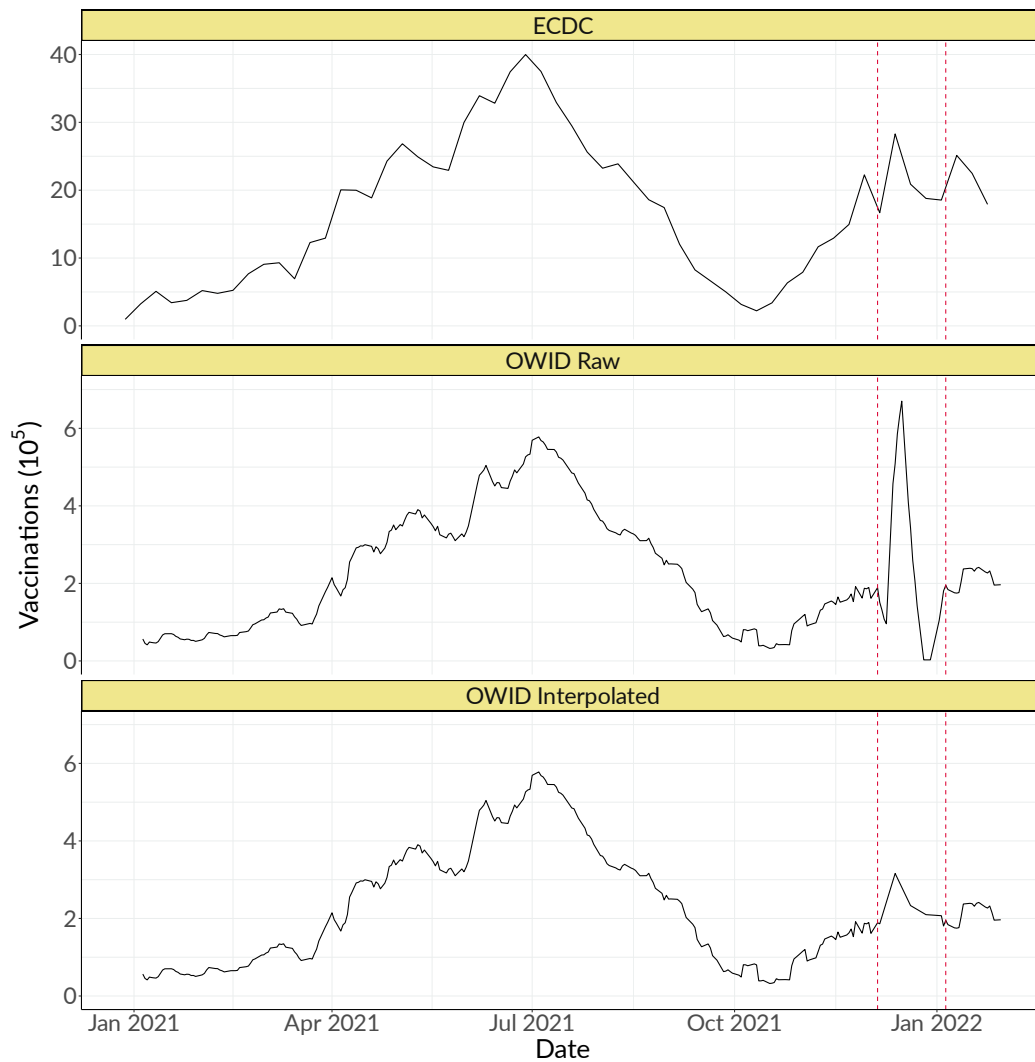


FIGURE D.5: Profile of the number of vaccinations administered in Spain. The first panel presents weekly data from the ECDC, the second panel displays raw daily data from OWID, and the third panel shows interpolated OWID data. The problematic period is highlighted with dashed red lines.

D.6 Policy stringency variables

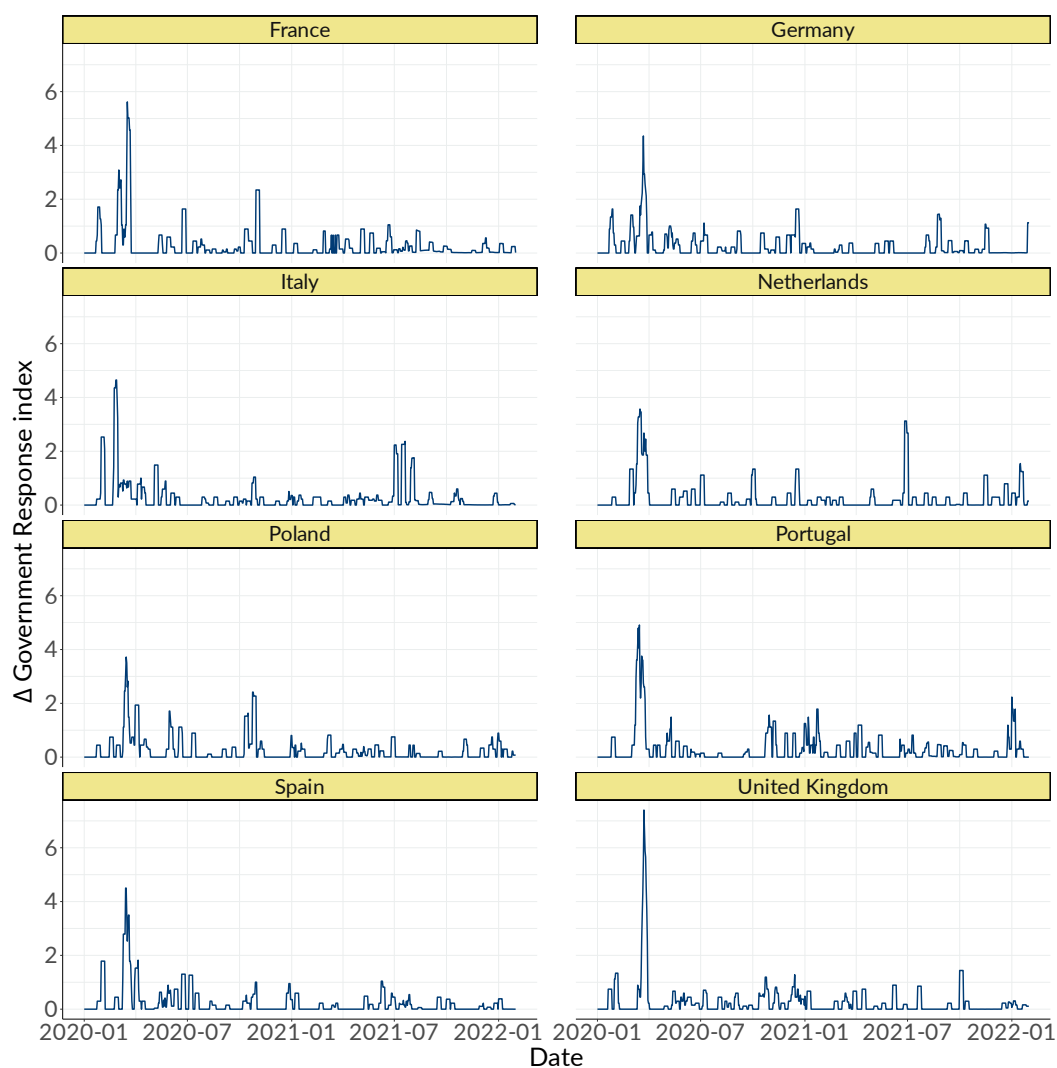


FIGURE D.6: The absolute difference between the seven-day MA of the Government Response index and its one-day lag.

E Appendix: Time series modelling to quantify associations between tweet, epidemiological, and public policy variables

E.1 Hyperparameters spaces

E.1.1 Lasso

Parameter	Values
alpha	$\mathcal{U}(0, 100)$
max_iter	50000
fit_intercept	$\{\text{True}, \text{False}\}$

TABLE E.1: Hyperparameter search space for lasso models

E.1.2 CatBoost

Parameter	Values
n_estimators	[10, 1000]
depth	[1, 16]
learning_rate	$\log \mathcal{U}(10^{-4}, 1)$
subsample	[0.1, 1.0]
colsample_bylevel	[0.1, 1.0]
l2_leaf_reg	$\log \mathcal{U}(10^{-1}, 10^5)$
min_data_in_leaf	[7, 96]
silent	True
random_seed	3141
has_time	True
loss_function	MAE

TABLE E.2: Hyperparameter search space CatBoost models

E.1.3 Random forest

Parameter	Values
n_estimators	$\log \mathcal{U}(10, 1000)$
max_depth	[1, 30]
min_samples_split	[2, 48]
min_samples_leaf	[2, 48]
max_features	[0.1, 1.0]
max_samples	[0.1, 1.0]
ccp_alpha	$\log \mathcal{U}(10^{-1}, 10^5)$
n_jobs	16
random_state	3141
criterion	absolute_error

TABLE E.3: Hyperparameter search space for Random forest models

SVR

Parameter	Values
epsilon	[0, 5]
C	$\log \mathcal{U}(10^{-2}, 10^2)$
degree	[2, 6]
kernel	$\{poly, rbf\}$
tol	10^{-3}
cache_size	2000

TABLE E.4: Hyperparameter search space for SVR.

E.1.4 General DL training

This hyperparameter search space applies to all deep learning models.

Parameter	Values
input_size	$[\mathbf{h}, 8 \cdot \mathbf{h}]$
learning_rate	$\log \mathcal{U}(10^{-7}, 10^{-1})$
max_steps	{100, 500, 1000}
batch_size	{2, 4, 8, 16, 32, 64, 128}
lr_scheduler	LinearLR
scaler_type	revin
loss	MAE
random_seed	3141

TABLE E.5: Hyperparameter search space for general DL training hyperparameter

E.1.5 DilatedRNN

Parameter	Values
cell_type	{GRU, RNN, LSTM}
encoder_hidden_size	{32, 64, 128, 256, 512}
dilations	{{[1, 2], [4, 8]}, {[1, 2, 4, 8]}}
context_size	{7, 14, 28, 56}
decoder_hidden_size	{32, 64, 128, 256, 512}
inference_input_size	None

TABLE E.6: Hyperparameter search space for DilatedRNN models

E.1.6 LSTM

Parameter	Values
encoder_hidden_size	{32, 64, 128, 256, 512}
encoder_n_layers	[1, 6]
encoder_dropout	[0, 0.99]
decoder_hidden_size	{32, 64, 128, 256, 512}
decoder_layers	[1, 6]
inference_input_size	None
h_train	h
recurrent	False

TABLE E.7: Hyperparameter search space for LSTM models

E.1.7 KAN

Parameter	Values
grid_size	{7, 14, 21, 28, 56}
spline_order	{1, 2, 3, 4, 5}
scale_noise	[0, 0.99]
enable_standalone_scale_spline	{True, False}
hidden_size	{32, 64, 128, 256, 512, 1024}
n_hidden_layers	[1, 4]

TABLE E.8: Hyperparameter search space for KAN models.

E.1.8 N-Beatsx

Parameter	Values
n_harmonics	[1, 6]
n_polynomials	[1, 6]
stack_types	{identity, trend} {identity, trend, seasonality} {identity, trend, exogenous} {identity, trend, seasonality, exogenous}
mlp_units	{32, 64, 128, 256, 512}
n_blocks	[1, 4] per stack
dropout_prob_theta	[0, 0.99]

TABLE E.9: Hyperparameter search space for N-BEATSx models. For stack_types, the values in black apply to AR-only models while the values in blue only apply to models which include exogenous variables.

E.1.9 TCN

Parameter	Values
kernel_size	{2, 4, 8, 16, 32}
encoder_hidden_size	{64, 128, 256, 512, 1024}
encoder_activation	{ReLU, Tanh, LeakyReLU}
decoder_hidden_size	{64, 128, 256, 512, 1024}
context_size	{7, 14, 28, 56}
decoder_layers	[1, 6]

TABLE E.10: Hyperparameter search space for TCN models

E.1.10 TFT

Parameter	Values
hidden_size	{32, 64, 128, 256, 512}
n_head	{2, 4, 8, 16}
attn_dropout	[0, 0.99]
grn_activation	{ReLU, Sigmoid, ELU}
n_rnn_layers	[1, 4]
rnn_type	{LSTM, GRU}
dropout	[0, 0.99]

TABLE E.11: Hyperparameter search space for TFT models

E.1.11 TiDE

Parameter	Values
hidden_size	{32, 64, 128, 256, 512, 1024}
decoder_output_dim	{4, 8, 16, 32, 64, 128}
temporal_decoder_dim	{8, 16, 32, 64, 128, 256}
num_encoder_layers	[1, 4]
num_decoder_layers	[1, 4]
temporal_width	{4, 8, 16, 32}
dropout	[0, 0.99]
layernorm	{True, False}

TABLE E.12: Hyperparameter search space for TiDE models

E.2 Hyperparameters results

E.2.1 Lasso

	Cases				Hospitalisations				Deaths			
Key	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
alpha	7.9	1.2	7.5	0.1	0.6	1.2	1.2	0.1	19.1	1.2	0.9	0.1
fit_intercept	True	False	False	False	True	False	False	False	False	False	False	False
Country intercept	1	0	1	0	1	0	0	0	0	0	0	0
AR lag V1	1	1	1	1	1	1	1	1	1	1	1	1
AR lag V2	8	6	5	6	5	6	6	6	5	6	8	6
AR lag V3	10	9	9	10	10	9	9	10	10	9	11	10
AR lag V4	13	12	12	12	13	12	12	12	14	12	12	12
OxGRT lag V1		4		2		4		2		4		2
OxGRT lag V2		8		8		8		8		8		8
OxGRT lag V3		10		11		10		11		10		11
OxGRT lag V4		14		13		14		13		14		13
Tweets lag V1			4	3			4	3			4	3
Tweets lag V2			8	8			8	8			8	8
Tweets lag V3			11	10			10	10			10	10
Tweets lag V4			13	12			14	12			14	12
Validation MAPE (%)	18	9	16	9	7	10	8	8	22	13	17	12

TABLE E.13: Lasso specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses. Note that fit_intercept relates to an overall intercept across all countries, while the country intercept is an additional adjustment. No restriction was imposed to prevent a combination of the two.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
alpha	41.6	93.1	37.2	61.5	37.2	99.5	78.7	90.9
fit_intercept	True	True	True	True	True	True	True	True
Country intercept	1	1	1	1	1	1	1	1
AR lag V1	1	1	1	3	1	2	1	2
AR lag V2	7	6	6	5	6	6	6	6
AR lag V3	10	10	10	11	10	9	11	9
AR lag V4	13	14	14	14	14	13	12	13
Cases lag V1			1	3				
Cases lag V2			6	5				
Cases lag V3			11	9				
Cases lag V4			13	13				
OxGRT lag V1		1		2		2		2
OxGRT lag V2		6		6		7		7
OxGRT lag V3		11		9		10		11
OxGRT lag V4		13		14		12		12
Hospitalisations lag V1					1	3		
Hospitalisations lag V2					6	8		
Hospitalisations lag V3					11	9		
Hospitalisations lag V3					13	13		
Deaths lag V1							1	3
Deaths lag V2							8	6
Deaths lag V3							10	11
Deaths lag V4							13	14
Validation MAPE (%)	52	58	63	66	60	55	50	52

TABLE E.14: Lasso specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response. Note that fit_intercept relates to an overall intercept across all countries, while the country intercept is an additional adjustment. No restriction was imposed to prevent a combination of the two.

E.2.2 CatBoost

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
n_estimators	590	985	849	543	736	975	88	644	783	667	667	586
depth	4	6	5	14	5	9	13	14	5	6	12	15
learning_rate	1	0	0.2	0.1	0.2	0	0.4	0.1	0.2	0.1	0.1	0
subsample	1	0.9	0.9	1	0.6	0.3	0.4	1	0.6	0.7	0.1	1
colsample_bylevel	0.7	1	0.6	1	0.8	0.5	0.8	0.6	0.6	0.8	0.9	0.5
l2_leaf_reg	4.5	0.4	5.5	6787.6	16.9	0.7	0.4	5930.9	10.4	72694.4	2908.4	31.4
min_data_in_leaf	49.2	74	79.9	73.8	57.2	57.4	77.8	71.5	58.9	79.5	48.4	66
Country intercept	0	1	1	1	0	0	0	1	0	0	0	1
AR lag V1	2	1	1	1	2	1	1	1	2	1	3	1
AR lag V2	7	5	8	7	7	8	5	7	7	8	5	7
AR lag V3	11	10	10	10	11	10	10	10	11	9	9	10
AR lag V4	13	13	14	14	13	13	14	14	13	12	14	14
OxGRT lag V1		3		2		3		2		1		1
OxGRT lag V2		7		8		6		8		6		6
OxGRT lag V3		10		10		9		10		10		10
OxGRT lag V4		13		12		12		12		14		12
Tweets lag V1			4	3			4	3			4	3
Tweets lag V2			7	8			6	8			7	7
Tweets lag V3			11	11			11	11			10	9
Tweets lag V4			13	14			14	14			13	14
Validation MAPE (%)	26	24	26	24	21	23	18	27	50	37	60	36

TABLE E.15: CatBoost specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
n_estimators	717	868	317	603	317	758	953	275
depth	9	3	2	11	2	7	5	11
learning_rate	0	0.3	0	0.3	0	0	0.1	0
subsample	0.2	0.7	0.3	0.6	0.3	0.3	0.8	0.6
colsample_bylevel	0.2	0.3	0.5	0.3	0.5	0.5	0.7	0.3
l2_leaf_reg	1310.4	0.2	89.8	0.4	89.8	0.4	19655.5	6
min_data_in_leaf	79.9	84.3	86.7	48.4	86.7	35.4	68.5	45.2
Country intercept	0	1	1	0	1	0	1	0
AR lag V1	1	1	1	1	1	1	1	3
AR lag V2	6	5	6	7	6	7	7	6
AR lag V3	10	10	9	11	9	11	10	11
AR lag V4	13	14	12	14	12	14	14	12
Cases lag V1			2	2				
Cases lag V2			5	8				
Cases lag V3			10	11				
Cases lag V4			14	14				
OxGRT lag V1		4		2		2		2
OxGRT lag V2		7		5		5		5
OxGRT lag V3		11		11		11		11
OxGRT lag V4		14		14		14		14
Hospitalisations lag V1					2	2		
Hospitalisations lag V2					5	8		
Hospitalisations lag V3					10	11		
Hospitalisations lag V4					14	14		
Deaths lag V1							3	1
Deaths lag V2							6	6
Deaths lag V3							10	9
Deaths lag V4							13	12
Validation MAPE (%)	76	85	70	110	66	102	67	161

TABLE E.16: CatBoost specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

E.2.3 Random forest

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
n_estimators	123	230	230	196	356	125	177	862	98	230	177	78
max_depth	26	9	9	23	27	12	11	30	28	9	11	28
min_samples_split	7	3	3	25	31	16	2	15	16	3	2	20
min_samples_leaf	20	4	4	19	28	8	2	21	17	4	2	12
max_features	0.8	0.9	0.9	0.9	0.8	0.8	0.9	1	0.9	0.9	0.9	0.8
max_samples	0.3	0.4	0.4	0.3	0.4	0.9	0.2	0.6	0.2	0.4	0.2	0.3
ccp_alpha	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.3	0.1	0.1	0.1	0.3
Country intercept	0	1	1	0	0	0	1	0	0	1	1	0
AR lag V1	1	2	2	1	1	2	2	1	1	2	2	1
AR lag V2	7	7	7	8	6	7	8	8	5	7	8	8
AR lag V3	11	11	11	11	11	9	11	11	11	11	11	11
AR lag V4	14	12	12	14	14	13	12	14	14	12	12	14
OxGRT lag V1		2		3		4		3		2		3
OxGRT lag V2		8		6		5		8		8		6
OxGRT lag V3		9		9		9		9		9		9
OxGRT lag V4		13		13		12		13		13		13
Tweets lag V1			2	3			2	3			2	3
Tweets lag V2			8	7			8	7			8	7
Tweets lag V3			9	10			9	11			9	10
Tweets lag V4			13	13			13	13			13	13
Validation MAPE (%)	67	36	50	41	44	49	35	64	87	50	79	48

TABLE E.17: Random forest specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
n_estimators	23	763	257	47	113	33	301	33
max_depth	28	14	23	22	27	19	15	19
min_samples_split	4	7	28	26	7	17	6	17
min_samples_leaf	32	23	41	3	26	26	6	26
max_features	0.9	0.4	0.7	0.3	0.7	0.3	1	0.3
max_samples	0.5	0.5	0.6	0.3	0.2	0.4	0.4	0.4
ccp_alpha	0.5	12.4	4.3	6.5	0.2	37.9	0.6	37.9
Country intercept	0	1	1	0	1	0	1	0
AR lag V1	1	1	1	1	2	3	2	3
AR lag V2	5	7	7	7	8	6	6	6
AR lag V3	11	10	9	11	11	11	9	11
AR lag V4	14	13	13	13	12	12	13	12
Cases lag V1			2	1				
Cases lag V2			6	6				
Cases lag V3			9	11				
Cases lag V4			14	12				
OxGRT lag V1		3		2		2		2
OxGRT lag V2		6		7		5		5
OxGRT lag V3		10		11		11		11
OxGRT lag V4		12		14		14		14
Hospitalisations lag V1					2	1		
Hospitalisations lag V2					8	6		
Hospitalisations lag V3					11	9		
Hospitalisations lag V4					13	12		
Deaths lag V1							2	1
Deaths lag V2							8	6
Deaths lag V3							9	9
Deaths lag V4							14	12
Validation MAPE (%)	71	136	89	142	93	157	92	157

TABLE E.18: Random forest model specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
kernel	rbf	rbf	rbf	rbf	rbf	rbf	rbf	rbf
Country intercept	0	1	1	1	1	1	0	1
AR lag V1	1	1	1	1	2	1	1	1
AR lag V2	5	8	5	6	8	6	8	6
AR lag V3	10	11	9	9	11	9	9	11
AR lag V4	14	14	12	14	14	14	12	14
Cases lag V1			4	3				
Cases lag V2			5	8				
Cases lag V3			10	11				
Cases lag V4			12	14				
OxGRT lag V1		1		2		2		2
OxGRT lag V2		7		5		5		5
OxGRT lag V3		10		9		9		9
OxGRT lag V4		12		13		13		14
Hospitalisations lag V1					3	3		
Hospitalisations lag V2					5	8		
Hospitalisations lag V3					11	11		
Hospitalisations lag V4					13	14		
Deaths lag V1							1	3
Deaths lag V2							7	8
Deaths lag V3							10	11
Deaths lag V4							14	13
Validation MAPE (%)	51	52	52	52	77	69	51	62

TABLE E.20: SVR specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response. Degree is not shown since a polynomial kernel was not chosen for any combination.

E.2.5 DilatedRNN

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
cell_type	GRU	LSTM	LSTM	RNN	GRU	LSTM	LSTM	LSTM	LSTM	GRU	GRU	GRU
encoder_hidden_size	256	256	128	64	128	128	512	128	64	512	512	64
dilations	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8
context_size	7	7	56	28	7	7	14	28	28	56	56	56
decoder_hidden_size	256	256	64	256	256	512	128	128	256	256	128	64
input_size	12	19	15	13	25	21	16	17	12	7	11	21
learning_rate	4.0×10^{-4}	4.0×10^{-4}	6.2×10^{-5}	2.6×10^{-4}	4.7×10^{-4}	1.9×10^{-3}	3.3×10^{-3}	2.6×10^{-4}	2.0×10^{-3}	5.2×10^{-5}	3.9×10^{-4}	3.1×10^{-4}
max_steps	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
batch_size	64	128	128	32	64	8	8	2	8	128	64	8
Country intercept	1	0	1	0	1	0	1	1	1	0	1	1
Validation MAPE (%)	3	2	5	2	3	5	4	6	8	8	8	14

TABLE E.21: DilatedRNN specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
cell_type	LSTM	LSTM	LSTM	LSTM	LSTM	LSTM	LSTM	LSTM
encoder_hidden_size	512	512	512	64	512	512	512	512
dilations	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8	1, 2, 4, 8
context_size	7	28	7	28	14	14	14	7
decoder_hidden_size	512	128	64	128	64	64	64	512
input_size	28	22	29	24	22	24	22	26
learning_rate	2.6×10^{-3}	3.4×10^{-3}	1.9×10^{-4}	3.6×10^{-3}	1.8×10^{-4}	1.7×10^{-4}	1.8×10^{-4}	2.6×10^{-3}
max_steps	1000	1000	1000	1000	500	500	500	1000

Continued on next page

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
batch_size	16	16	16	16	16	16	16	2
Country intercept	0	1	1	1	1	1	1	0
Validation MAPE (%)	6	12	11	8	13	11	14	7

TABLE E.22: DilatedRNN specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

E.2.6 KAN

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
grid_size	28	7	7	21	7	7	7	7	7	56	7	14
spline_order	2	1	1	5	1	1	1	5	1	3	1	5
scale_noise	0.8	0.3	0.4	0.1	0.3	0.5	0.1	0.1	0.5	0.4	0.3	0.5
enable_standalone*	False	False	False	False	False	False	True	True	False	True	False	True
hidden_size	64	64	256	64	64	64	32	32	256	256	64	64
n_hidden_layers	1	1	2	2	1	2	3	3	2	2	1	2
input_size	10	11	12	12	11	10	12	10	7	10	11	12
learning_rate	4.0×10^{-3}	6.9×10^{-3}	8.0×10^{-4}	1.7×10^{-3}	6.9×10^{-3}	1.5×10^{-2}	8.1×10^{-3}	3.1×10^{-3}	1.8×10^{-2}	3.7×10^{-5}	6.9×10^{-3}	6.7×10^{-3}
max_steps	800	1000	1500	1100	1000	1100	1100	900	1300	1100	1000	800
batch_size	16	32	8	32	32	32	32	32	16	32	32	32
Country intercept	0	0	1	1	0	0	0	0	1	0	0	1
Validation MAPE (%)	2	3	4	4	3	4	4	4	7	8	9	9

TABLE E.23: KAN specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses. *

Full parameter name is enable_standalone_scale_spline.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
grid_size	56	7	7	56	7	21	14	7
spline_order	2	1	1	5	1	2	1	1
scale_noise	0.8	0	0.3	0	0.3	0.4	0.1	0.1
enable_standalone_scale_spline	False	True	False	True	False	False	False	True
hidden_size	1024	64	64	256	64	256	32	32
n_hidden_layers	2	2	1	2	1	1	2	3
input_size	7	8	11	7	11	7	7	12
learning_rate	5.9×10^{-5}	3.5×10^{-4}	6.9×10^{-3}	1.8×10^{-3}	6.9×10^{-3}	1.3×10^{-3}	1.7×10^{-2}	8.1×10^{-3}
max_steps	800	900	1000	800	1000	1500	1400	1100
batch_size	32	32	32	8	32	2	64	32
Country intercept	0	0	0	0	0	0	1	0
Validation MAPE (%)	6	7	7	7	7	8	8	9

TABLE E.24: KAN specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

E.2.7 LSTM

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
encoder_hidden_size	512	128	512	64	512	128	512	512	128	128	64	512
encoder_n_layers	2	1	1	1	2	3	1	1	2	4	2	2
encoder_dropout	0	0.6	0.4	0.1	0.3	0.1	0.5	1	0.4	0.3	0.4	0
decoder_hidden_size	64	512	32	128	64	128	64	128	64	64	32	256
decoder_layers	5	4	5	1	2	2	5	3	3	6	4	5
input_size	22	14	55	7	22	24	31	14	13	13	13	7
learning_rate	1.0×10^{-3}	3.3×10^{-3}	9.9×10^{-4}	3.0×10^{-2}	3.8×10^{-4}	3.4×10^{-3}	1.2×10^{-3}	2.0×10^{-3}	1.7×10^{-3}	9.2×10^{-4}	5.3×10^{-3}	2.1×10^{-3}
max_steps	1000	1000	500	1000	1000	1000	500	500	1000	1000	500	500

Continued on next page

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
batch_size	128	4	128	64	8	8	32	128	4	32	4	2
Country intercept	1	0	1	1	1	1	1	1	0	1	1	1
Validation MAPE (%)	15	4	10	5	8	8	10	8	8	9	9	10

TABLE E.25: LSTM specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
encoder_hidden_size	512	128	512	64	512	512	512	512
encoder_n_layers	1	3	1	1	2	1	1	2
encoder_dropout	0.4	0.1	0.4	0.6	0.2	0.4	0.4	0.6
decoder_hidden_size	64	64	64	256	32	64	64	128
decoder_layers	5	5	5	3	4	5	5	6
input_size	19	7	16	13	7	19	19	15
learning_rate	7.3×10^{-4}	1.5×10^{-3}	7.8×10^{-4}	2.4×10^{-3}	1.5×10^{-4}	7.3×10^{-4}	7.3×10^{-4}	5.2×10^{-4}
max_steps	500	500	500	500	1000	500	500	500
batch_size	64	64	4	128	128	64	64	64
Country intercept	1	1	1	1	1	1	1	1
Validation MAPE (%)	12	9	13	14	9	15	13	13

TABLE E.26: LSTM specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

E.2.8 N-BEATS_x

	Cases				Hospitalisations				Deaths			
Key	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
stack_types	id, tr	id, tr	id, tr ,	id, tr ,	id, tr	id, tr ,	id, tr ,	id, tr	id, tr	id, tr	id, tr	id, tr ,
	,sn	,ex	sn, ex	sn, ex	,sn	sn, ex	sn, ex	,ex		,ex	,ex	sn, ex
n_blocks	2	1	3	3	2	2	1	3	1	2	2	2
mlp_units_options	256	64	256	256	64	256	64	512	64	256	64	256
n_harmonics	1	5	5	3	2	1	1	4	2	3	5	1
n_polynomials	3	4	1	2	4	1	2	3	4	4	5	5
dropout_prob_theta	0.5	0.5	0	0.6	0.2	0.6	0.2	0.4	0.4	0.8	0.7	0.2
input_size	12	9	15	12	18	12	11	15	15	24	14	11
learning_rate	2.8×10^{-4}	1.4×10^{-3}	3.8×10^{-5}	1.8×10^{-4}	7.0×10^{-4}	5.2×10^{-4}	3.5×10^{-3}	3.0×10^{-4}	3.1×10^{-4}	6.1×10^{-4}	1.0×10^{-3}	5.5×10^{-5}
max_steps	500	500	500	500	1000	500	500	500	500	500	500	500
batch_size	4	2	4	2	16	16	2	2	4	16	4	2
Country intercept	0	0	1	0	1	0	1	0	0	1	0	0
Validation MAPE (%)	3	3	5	5	3	4	3	5	7	9	7	8

TABLE E.27: N-BEATSx specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses. Stack type legend: **id**: identity, **tr**:trend , **sn**: seasonality, and **ex**: exogenous.

[illegible]

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
n_polynomials	3	6	5	3	3	3	5	3
dropout_prob_theta	0	0.2	0	0.4	0.2	0.4	0.2	0.4
input_size	10	7	7	7	7	7	7	7
learning_rate	4.3×10^{-4}	1.4×10^{-3}	3.5×10^{-4}	7.0×10^{-4}	5.0×10^{-3}	7.0×10^{-4}	1.0×10^{-3}	7.0×10^{-4}
max_steps	500	500	500	500	500	500	500	500
batch_size	64	64	16	64	16	64	16	64
Country intercept	1	1	0	1	1	1	1	1
Validation MAPE (%)	6	8	6	8	7	8	6	8

TABLE E.28: N-BEATSx specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response. Stack type legend: **id**: identity, **tr**:trend , **sn**: seasonality, and **ex**: exogenous.

E.2.9 TCN

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
kernel_size	32	32	32	32	4	32	32	32	32	32	2	8
encoder_hidden_size	64	64	64	64	64	64	64	64	256	64	256	128
encoder_activation	ReLU	ReLU	ReLU	ReLU	LeakyReLU	ReLU	Tanh	Tanh	ReLU	LeakyReLU	ReLU	Tanh
decoder_hidden_size	256	256	256	256	512	256	1024	256	256	256	64	1024
context_size	56	56	56	56	56	56	56	28	7	7	7	28
decoder_layers	1	1	2	2	1	1	1	2	2	2	2	6
input_size	7	9	7	7	10	9	10	14	13	9	13	7
learning_rate	2.0×10^{-3}	1.7×10^{-3}	3.9×10^{-4}	3.9×10^{-4}	6.1×10^{-4}	1.7×10^{-3}	3.6×10^{-3}	2.0×10^{-3}	5.5×10^{-4}	2.3×10^{-3}	7.8×10^{-3}	2.8×10^{-5}
max_steps	2000	2000	2000	2000	2000	2000	2000	2000	1000	2000	1000	2000
batch_size	64	64	8	64	128	64	64	64	32	64	128	4
Country intercept	1	1	1	1	1	1	1	1	1	1	1	0

Continued on next page

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
Validation MAPE (%)	2	4	3	4	3	5	4	6	8	8	8	9

TABLE E.29: TCN specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
kernel_size	32	32	32	32	4	32	32	32
encoder_hidden_size	64	64	64	64	64	64	64	64
encoder_activation	ReLU	ReLU	ReLU	ReLU	LeakyReLU	ReLU	ReLU	ReLU
decoder_hidden_size	256	256	256	256	256	256	256	256
context_size	56	56	56	56	56	56	56	56
decoder_layers	1	1	1	1	1	1	1	1
input_size	9	7	9	10	12	7	9	7
learning_rate	1.7×10^{-3}	2.1×10^{-3}	1.7×10^{-3}	3.8×10^{-3}	2.4×10^{-3}	2.0×10^{-3}	3.8×10^{-3}	2.0×10^{-3}
max_steps	2000	2000	2000	2000	2000	2000	2000	2000
batch_size	64	8	64	64	128	64	64	64
Country intercept	1	1	1	1	1	1	1	1
Validation MAPE (%)	6	7	6	7	6	7	6	7

TABLE E.30: TCN specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

E.2.10 TFT

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
hidden_state	64	64	128	32	64	512	64	64	64	64	512	64
n_head	4	4	4	8	4	4	4	4	16	4	8	4
attn_dropout	0.9	0.9	0.5	0.5	0.8	0.4	0.7	0.9	0.4	0.8	0.9	1
grn_activation	ELU	ELU	Sigmoid	ELU	ELU	ELU	Sigmoid	ELU	ReLU	ELU	ELU	ELU
n_rnn_layers	1	1	2	1	1	2	2	1	3	1	2	1
rnn_type	LSTM	LSTM	GRU	LSTM	LSTM	LSTM	LSTM	LSTM	GRU	LSTM	LSTM	LSTM
dropout	0.8	0.8	0.4	0.3	0.9	0.7	0	0.8	0.5	1	0.8	0.9
input_size	10	10	15	11	20	11	14	11	9	9	18	13
learning_rate	4.0×10^{-4}	4.0×10^{-4}	6.2×10^{-5}	2.6×10^{-4}	4.7×10^{-4}	1.9×10^{-3}	3.3×10^{-3}	2.6×10^{-4}	2.0×10^{-3}	5.2×10^{-5}	3.9×10^{-4}	3.1×10^{-4}
max_steps	500	500	1000	500	500	500	500	500	100	500	500	500
batch_size	64	64	32	8	8	64	32	64	2	8	8	64
Country intercept	0	0	1	0	0	1	1	0	0	0	0	0
Validation MAPE (%)	2	3	3	3	3	4	4	4	7	7	8	8

TABLE E.31: TFT specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
hidden_state	32	512	32	64	512	32	64	512
n_head	8	4	8	4	2	8	16	4
attn_dropout	0.2	0.8	1	1	0.4	0.5	0.4	0.9
grn_activation	ReLU	ReLU	ReLU	ELU	ReLU	Sigmoid	ReLU	ELU
n_rnn_layers	1	1	3	1	1	1	1	2
rnn_type	LSTM	LSTM	GRU	LSTM	GRU	GRU	GRU	LSTM
dropout	0.8	0.1	0.5	0.8	0.6	0.4	0.6	1

Continued on next page

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
input_size	48	11	7	7	38	8	40	7
learning_rate	1.1×10^{-3}	3.2×10^{-5}	5.2×10^{-4}	2.8×10^{-4}	2.0×10^{-5}	5.4×10^{-3}	5.1×10^{-4}	1.4×10^{-4}
max_steps	500	500	500	500	1000	1000	500	500
batch_size	32	32	8	8	8	32	8	8
Country intercept	0	1	0	0	0	1	0	0
Validation MAPE (%)	8	7	7	7	8	7	8	7

TABLE E.32: TFT specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.

E.2.11 TiDE

Key	Cases				Hospitalisations				Deaths			
	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT	None	OxGRT	Tweets	Tweets + OxGRT
hidden_size	256	256	256	1024	1024	256	64	1024	1024	64	64	256
decoder_output_dim	4	4	4	128	128	4	4	4	4	4	8	32
temporal_decoder_dim	64	64	64	16	16	64	16	32	32	16	32	32
num_encoder_layers	2	3	3	4	2	3	3	3	3	2	3	3
num_decoder_layers	2	1	1	1	2	2	1	2	3	3	1	1
temporal_width	16	16	16	8	8	16	16	8	32	4	4	16
dropout	0.2	0.6	0.6	0.8	0.7	0.8	0.8	0.6	0.7	0.6	0.4	0.2
layernorm	False	False	False	True	True	False	False	False	False	False	False	False
learning_rate	1.2×10^{-3}	8.6×10^{-4}	8.6×10^{-4}	2.0×10^{-2}	9.5×10^{-3}	4.6×10^{-4}	3.4×10^{-3}	3.6×10^{-4}	2.8×10^{-5}	1.3×10^{-4}	1.1×10^{-2}	5.1×10^{-3}
max_steps	1000	1000	1000	1000	1000	500	500	500	1000	1000	1000	1000
batch_size	16	16	16	4	8	16	16	16	8	8	16	16
input_size	7	12	12	12	17	24	8	16	15	13	12	13
Country intercept	1	1	1	0	1	0	0	0	0	1	1	0
Validation MAPE (%)	2	3	3	4	5	5	5	4	7	8	8	8

TABLE E.33: TiDE specification based on 30 hyperparameter trials for each exogenous combination across epidemiological responses.

Key	None	OxGRT	Cases	Cases + OxGRT	Hospitalisations	Hospitalisations + OxGRT	Deaths	Deaths + OxGRT
hidden_size	1024	64	1024	32	64	32	256	32
decoder_output_dim	8	4	8	128	4	32	32	128
temporal_decoder_dim	32	32	64	32	32	8	16	32
num_encoder_layers	1	1	1	3	1	1	2	3
num_decoder_layers	2	1	3	1	1	2	3	1
temporal_width	8	4	16	4	4	4	4	4
dropout	0	0.6	0.4	0.5	0.6	0.4	0.3	0.5
layernorm	False	True	False	True	True	True	True	True
input_size	11	15	17	15	15	8	15	15
learning_rate	1.2×10^{-3}	3.4×10^{-2}	3.2×10^{-5}	2.7×10^{-2}	3.4×10^{-2}	2.6×10^{-2}	2.3×10^{-2}	2.7×10^{-2}
max_steps	500	1000	1000	1000	1000	1000	1000	1000
batch_size	8	4	4	8	4	8	8	8
Country intercept	False	True	True	True	True	True	True	True
Validation MAPE (%)	5	9	8	9	8	9	9	9

TABLE E.34: TiDE specification based on 30 hyperparameter trials for each exogenous combination with a tweet volume response.