Forecasting COVID-19 Caseloads Using Unsupervised Embedding Clusters of Social Media Posts

Anonymous ACL submission

Abstract

We present a novel approach incorporating transformer-based language models into infectious disease modelling. Text-derived features are quantified by tracking high-density clusters of sentence-level representations of Reddit posts within specific US states' COVID-19 subreddits. We benchmark these clustered embedding features against features extracted from other high-quality datasets. In a threshold-classification task, we show that they outperform all other feature types at predicting upward trend signals, a significant result for infectious disease modelling in areas where epidemiological data is unreliable. Subsequently, in a time-series forecasting task we fully utilise the predictive power of the caseload and compare the relative strengths of using different supplementary datasets as covariate feature sets in a transformer-based time-series model.

1 Introduction

Many papers have shown that web search data can be used to forecast the spread of infectious diseases (Lampos et al., 2017), (Lampos et al., 2021), (McDonald et al., 2021), (Reinhart et al., 2021), (Alruily et al., 2022). Alongside this literature, social media has been exploited for its predictive potential in several other fields such as quantitative finance Bukovina (2016), Archary and Coetzee (2020), logistics forecasting Ni et al. (2017) and election forecasting (Bermingham and Smeaton, 2011), (Huberty, 2015). The conjoining of these two applications has resulted in research showing that social media can help predict rises in disease caseloads. Iso et al. (2016) and Samaras et al. (2020) both used pre-defined keywords in order to predict outbreaks of influenza; words such as "Influenza", "fever", "headache" were selected a-priori. These papers assume that useful feature sets have no geographical variation and use the same features regardless of the regional social dynamics; they also assume that useful features are limited to words that refer to symptoms. To address these limitations, Drinkall and Pierrehumbert (2021) set more general and objective inclusion criteria. For each of four US state COVID-19 subreddits, all words over-represented in that US state's COVID-19 subreddit compared to the rest of Reddit were considered to be potential keywords for forecasting. The most informative keywords proved to be highly dependent on the target state, and included many that did not refer to symptoms. However, the paper still relied on static word counts that miss more complex information as the discussion unfolds over time. The present paper extracts more informative features from social media data and, to our knowledge, is the first work to incorporate modern NLP techniques in this setting.

Figure 1: HDBSCAN clusters of the SBERT-NLI-STSb-base representations of r/CoronavirusWA posts made at 50 dimensions but reduced to 2 for visualisation.

New transformer-based language models (Devlin et al., 2019), (Yang et al., 2019), (Liu et al., 2019) provide the potential for identifying more informative features for infectious disease forecasting, and using them in a more effective manner. This paper uses transfer learning and clustering algorithms to isolate useful features for predicting COVID-19 caseloads. We first pilot-tested a straightforward way to exploit transformer-based language models for the task: the caseload target value was encoded alongside each post in a sequence classification framework. Trained using

historical data, this approach generates a prediction from every post, and the results are aggregated for an overall prediction. This method performed very poorly because of noise introduced by irrelevant posts, and we do not discuss it further here (see Appendix C). To achieve better performance, we developed a novel feature identification technique that filters out unrelated posts and generates informative features using high-density clusters of posts within a subreddit's embedding space.

Our work builds off Sia et al. (2020) and Thompson and Mimno (2020) who demonstrated that clusters of contextualised word embeddings are a good basis for topic modelling. In a similar vein, Aharoni and Goldberg (2020) showed that the domain type of a particular text could be identified using the clustering of sentence-level representations. Finally, Rother et al. (2020) showed that clusters of contextualised embeddings could detect meaning shifts in words. The success of these papers motivates our use of high-density clusters of sentence-level representations.

The present paper shows that our novel feature sets outperform more traditional methods by comparing our results to those in Drinkall and Pierrehumbert (2021) in a threshold-classification task. This task provides an understanding of which feature sets provide the most informative trend signals at different caseload growth rates, enabling us to understand the effectiveness of a particular feature type at identifying a distinct epidemiological event. Strong performance on this task is relevant in managing worst-case scenarios like hospital overflow, where success is binary.

The caseload information is not adequately utilised in the threshold-classification task, which motivates a time-series forecasting task to compare feature sets at predicting a more continuous target. Feature selection is a crucial step in time-series modelling (Wang et al., 2013), (Sun et al., 2015); adding extraneous features to a multivariate prediction can result in performance deterioration as the models get more complex, a fact that inspired L1-regularisation. Only highly relevant features, which represent complementary information, improve performance.

Contributions. We introduce a novel unsupervised method for predicting COVID-19 trend signals and forecasting caseloads. We show that sole use of our feature set achieves very high accuracy in trend signal prediction, a significant result for infectious disease modelling in regions where other reported data is unreliable.

2 Datasets

Comparing our Reddit features' performance against other high-quality geographically-specific data sources allows us to understand their value. The following data sources were used to create the feature sets in this paper:

Pushshift API - The Pushshift API (Baumgartner et al., 2020) is used to compile datasets of target subreddits to create the Reddit features. The Pushshift API provides data on every comment and submission posted on Reddit. This paper uses comments to form the subreddit dataset since there are more comments than submissions, and they constitute more conversational and reactionary discourse. No individual comments or users are reported in this paper to observe the anonymity of the users. Update frequency: real-time.

COVID-19 Tracking Project - The state-level COVID-19 epidemiological data is provided by the COVID-19 Tracking Project[1] to create the prediction target and is also used as a feature set in baseline predictions. Update frequency: 24 hours. Start date: 13/01/2020.

Oxford COVID-19 Government Response Tracker (OxCGRT) - The OxCGRT (Hale et al., 2020) defines the local government response. The data covers policies including health, containment and economic measures, and overall stringency scores. Update frequency: "continuously" but can be variable due to human data collection; daily periodicity. Start date: 01/01/2020.

Google's COVID-19 Community Mobility Reports (GCCMR) [2] - The GCCMR provides local movement data in different area types such as parks, workplaces, etc. and has been used to successfully predict COVID-19 caseloads (Wang et al., 2020), (Ilin et al., 2021). The data is freely available for the duration of the ongoing pandemic. Update frequency: 2-3 days. Start date: 15/02/2020.

3 Feature identification

Social media is a complex and noisy data source that requires significant processing to find meaningful predictive features. The pipeline used in this paper consists of three main steps for feature identification: sentence-level encoding, dimensionality

[1] https://covidtracking.com
[2] https://www.google.com/covid19/mobility/

reduction and clustering. Following these steps the Reddit features are reduced further to 25 using a chi-squared test. This process groups together Reddit comments that are semantically similar. Once high-density clusters are identified, the daily counts of comments within these clusters are used as features in the evaluation frameworks in Sections 4 & 5.

3.1 Sentence-level representation

A common technique for identifying sentence representations is to take the average-pooled BERT hidden-state embedding (Aharoni and Goldberg, 2020); however, papers such as Reimers and Gurevych (2019) have shown that the average-pooled BERT embeddings are a relatively poor way of encoding sentences and advocate for further fine-tuning to produce a more semantically meaningful embedding. In Reimers and Gurevych (2019), the best results are achieved by training the language model on Natural Language Inference (NLI) (Bowman et al., 2015), (Williams et al., 2018) and Sentence Textual Similarity (STS) (Cer et al., 2017) data. The NLI data contains many sentence pairs with their semantic relationship labelled. The STS data provides a semantic relatedness score between 0-5. It is possible to use both datasets to fine-tune the language model using both dataset types by manipulating the objective functions. The NLI data is trained using a classification objective function, and the STS data is trained using a regression objective function. Reimers and Gurevych (2019) shows that averaging the final layer BERT embeddings leads to a Spearman rank correlation ρ between the cosine similarity of the sentence representations and the actual labels of the STS data of around $\rho = 54.81$, whereas SBERT-NLI-STSb-base achieves $\rho = 88.31$.

For this paper, there is no domain-specific training. The SBERT-NLI-STSb-base, SRoBERTa-NLI-STSb-base and SDistilBERT-NLI-STSb-base encode the Reddit posts with no further fine-tuning.

3.2 Dimensionality reduction

The language models specified in Section 3.1 have a dimensionality of 768, which means that their embedding space is very sparse, making it challenging to find dense clusters. Lowering the embedding dimensionality is consistent with the findings in Sia et al. (2020) who show that the dimensionality of the embeddings can be reduced by ~80% and still maintain the topic modelling coherence. Therefore, in line with the findings of Sia et al. (2020), the dimensionality of the embedding space is reduced to 50.

UMAP (Uniform Manifold Approximation and Projection for Dimension Reduction) is used as in Rother et al. (2020) to lower the dimensionality of the embedding space. UMAP is appropriate for this task since it preserves global structure better than other manifold learning dimensionality reduction methods such as t-SNE (McInnes et al., 2018) (McConville et al., 2021). UMAP's preservation of global structure has been shown in Reif et al. (2019) to produce clear clusters related to different word senses. It is tested against a PCA algorithm in Appendix B on the Threshold-Classification task outlined in Section 4. The results justify its use as it outperforms PCA when used in conjunction with the best performing clustering algorithm.

3.3 Clustering

For this paper, the HDBSCAN algorithm (Campello et al., 2013) is used for clustering due to the complex structure of the subreddit embedding space. The benefit of using a density-based clustering algorithm is that sparse areas are not fitted into clusters, removing a significant source of noise from the prediction.

HDBSCAN offers an advantage over other density-based clustering algorithms; the cut-off density that characterises the edge of the clusters is non-constant and defined by a stability metric that rewards large and dense clusters. This stability metric is calculated from the data points' Minimum Spanning Tree (MST). The following equation defines the stability of cluster C_i:

$$S(C_i) = \sum_{x_j \in C_i} (\lambda_{max}(x_j, C_i) - \lambda_{min}(C_i)) \quad (1)$$

Here λ represents the density statistic: $\lambda = 1/\epsilon$ where ϵ is equal to the distance between points on the MST. In this equation, $\lambda_{max}(x_j, C_i)$ is the density at which the point x_j would fall out of the cluster C_i, and $\lambda_{min}(C_i)$ is the minimum density threshold at which the cluster still exists.

Clusters with maximum stability are used as the final clusters, and points that fall out of these clusters are discarded. New data points can subsequently be added to the cluster by identifying where they fall in the MST. A point is treated as noise unless it can be grouped into a cluster larger than

$min\_cluster\_size$, which, for this paper, we have set at 25 so that the clusters are not too small and the resulting features are not too sparse. Removing noisy comments from the clusters is shown in Appendix B to have performance benefits over other clustering algorithms that do not reject comments: we have compared HDBSCAN to a Spherical K-Means (KM) algorithm and a Gaussian Mixture Model (GMM), two popular algorithms within the literature base.

4 Threshold-Classification Framework

The threshold-classification framework (henceforth Threshold task) uses the same evaluation methodology as in Drinkall and Pierrehumbert (2021). The problem is presented as a classification task on balanced classes, with a randomised train/test split and test size of 0.25 on data from 07/03/2020 to 17/01/2021. Balanced classes allow us to report accuracy as the performance statistic for this task. The feature sets, derived from a 7-day moving average of the datasets in Section 2, are concatenated to a target value that encodes whether the caseload increase exceeded the threshold within a given time interval. The threshold is defined by a relative increase, $\delta_r(t)$:

$$\delta_r(t) = \frac{\mu(t+\tau) - \mu(t)}{\mu(t)} \tag{2}$$

Where $\mu(t)$ is the 7-day moving average of the caseload, and τ is the prediction horizon.

The model used for classification is a Random Forest (RF) (Breiman, 2001). The advantage of using an RF model over other tree-based models is that it decorrelates the trees, making it robust to correlated feature sets. Social media data is highly correlated as overall take-up surges and wains; therefore, robustness to correlated features is critical. Of course, many more complex models would likely outperform an RF model; however, given that the goal of this task is to compare feature sets, the increased transparency that an RF model offers over more complex models justify its use.

4.1 Evaluation

Each data type is used in isolation to predict the target labels so that the individual performance of the feature types can be compared. To benchmark the performance of our clustered embedding features - T_{DisB}, T_{BERT}, T_{RoB}, corresponding respectively to the features extracted from the SDistilBERT-NLI-STSb-base, SBERT-NLI-STSb-base and SRoBERTa-NLI-STSb-base language models - the performance is compared to the that of the features described in Drinkall and Pierrehumbert (2021), T_{BoW}. The evaluation is conducted in four states where Reddit uptake is high: Washington, California, Texas and Florida. The states represent culturally different communities, instilling confidence that the behaviour is true in multiple domains. A successful result across all four states shows that the behaviour is not just a symptom of an anomalous community or culture.

Feature set	Average	7D	14D	21D	28D
$T_{RoB}++$	.875	.895	.880	.849	.874
$T_{BoW}++$	.810	.836	.809	.805	.791
T_{RoB}	.803	.845	.792	.789	.787
T_{BERT}	.789	.821	.798	.780	.761
T_{DisB}	.780	.808	.771	.774	.768
T_{BoW}	.768	.816	.755	.749	.753
M	.702	.703	.691	.713	.698
G	.702	.713	.710	.695	.691
P	.545	.516	.549	.557	.557
C	.555	.651	.536	.529	.503

Table 1: The average performance across all relative thresholds and states at different prediction horizons. The features are: $T_{<<language\ model>>}$ → our features; T_{BoW} → Drinkall and Pierrehumbert (2021) features; M → GCCMR data; G → OxCGRT data; P → daily post count; C → current caseload; $T_{RoB}++ \rightarrow T_{RoB}+M+G+P+C$; $T_{BoW}++ \rightarrow T_{BoW}+M+G+P+C$. The light grey indicates the highest performing instance of each model setup. The dark grey indicates the highest performance for each prediction horizon.

The results in Table 1 detail the average performance across the different states and relative thresholds. Our $T_{<<language\ model>>}$ features provide the best single feature sets, and when T_{RoB} is used in combination with the comparison datasets, the performance improves further. It is also evident that as better language models are used, the performance on this task increases. Showcasing the relationship between language model complexity and overall performance supports our a-priori belief that improved semantic information from the text is linked with better epidemiological insights. Due to its success, T_{RoB} alone will henceforth be used in the evaluation as it provides the best performing feature set from our methodology.

4.1.1 Varying Thresholds

Table 2 breaks down the performance of classifying the data across different threshold increases. Intuitively, the more extreme events are easier to pre-

	$\delta_r(t)$					
m	0.2	0.4	0.6	0.8	1	$\mu + \sigma$
$T_{RoB}++$	.803	.828	.867	.962	.970	.880 + .039
$T_{BoW}++$	.704	.795	.821	.862	.910	.809 + .024
T_{RoB}	.753	.752	.787	.876	.876	.792 + .034
T_{BoW}	.683	.699	.761	.828	.865	.755 + .025
M	.649	.683	.663	.712	.727	.691 + .026
G	.678	.607	.735	.761	.789	.710 + .019
P	.548	.539	.537	.530	.577	.549 + .030
C	.437	.466	.527	.631	.640	.536 + .039

Table 2: Performance across a range of thresholds at a prediction horizon, $\tau = 14$ days, averaged across all states. μ & σ represent the mean and standard deviation of each feature set's results. The variables and highlighting criteria are the same as Table 1, but for the dark grey which denotes the highest performance at each threshold.

dict, explaining the behaviour across all feature sets. Indeed, when the threshold is large enough, the $T_{RoB}++$ features achieve an accuracy of .970, significantly higher than the comparison feature sets, showing that social media data is a strong candidate for predicting a sharp rise in caseloads. Again, the performance across all thresholds is highest when using the $T_{RoB}++$ features as opposed to the $T_{BoW}++$, highlighting the performance gain from the increased semantic information of transformer-based language models.

4.1.2 Feature Importance

To understand which features the RF model relies on when given the $T_{RoB}++$ and $T_{BoW}++$ feature set, the feature importances are shown in Table 3. The tabulated data represents the sum of all individual feature importances in that class.

Table 3 shows that despite T_{RoB} performing better than T_{BoW}, the other comparison features, G and M, are more heavily weighted in $T_{RoB}++$ than in $T_{BoW}++$ at some prediction horizons. The $T_{RoB}++$ feature set performs better than the $T_{BoW}++$ features, so it appears that the information provided by the T_{RoB} features is complementary to the other feature types. It is also possible

Feature set	m	Average	7D	14D	21D	28D
$T_{RoB}++$	T_{RoB}	.331	.334	.307	.309	.386
	M	.285	.264	.277	.299	.301
	G	.307	.321	.354	.313	.240
	P	.009	.009	.010	.008	.009
	C	.064	.071	.052	.070	.063
$T_{BoW}++$	T_{BoW}	.535	.584	.502	.509	.543
	M	.244	.222	.265	.254	.235
	G	.171	.131	.199	.192	.162
	P	.014	.012	.009	.013	.021
	C	.036	.050	.025	.031	.038

Table 3: Feature importances across varying prediction horizons, at $\delta_r = 0.6$. The variables and highlighting criteria are the same as Table 1.

that there is some skew in the feature importance owing to the reported over-weighting of more continuous features by a Gini Importance algorithm (Strobl, 2007). Regardless of the slight differences, both text-derived feature sets are the most highly weighted when averaged over all prediction horizons, further showing the value of social media in this context.

5 Time-Series Forecasting Task

This section showcases our feature identification methodology within a time-series forecasting framework (henceforth Time-Series task) since this is a widely used prediction task in disease modelling. The high-density clusters are used as covariates in two multivariate time-series models. This setup better utilises the caseload feature and learns the temporal patterns within its historical movement. One difference with the Threshold task in the feature identification pipeline is the feature pruning step that reduces the number of features to 25. In the Threshold task, the target is a binary classification; therefore, a chi-squared test is appropriate. Given that the target is continuous in this task, f-regression is used. F-regression works by firstly calculating the cross-correlation ρ_i of the i^{th} feature $X[:, i]$ and target y:

$$\rho_i = \frac{(X[:, i] - \bar{X}[:, i]) \cdot (y - \bar{y})}{\sigma_{X[:,i]} \cdot \sigma_y} \quad (3)$$

The F-statistic is then calculated along with the associated p-value. Then the top 25 most significant features are filtered to make up the feature set. For each model, the training features and targets are normalised between 0 and 1, and the test set is scaled using the same transformation. No moving average is used since the time-series models should account for the weekly seasonality. The models predict the caseload every day up to the forecast horizon; however, only the prediction error at the final step of the forecast horizon is used for evaluation. The models are trained over 50 epochs on data from 07/03/2020 to 31/12/2020 and tested on data from 01/01/2021 to 01/03/2021. Whilst it is possible to improve the performance by retraining the model on recently evaluated data and sliding the train-test split across the dataset, our proposed framework highlights how well the models perform on completely out-of-sample data.

5.1 Models

We compare a Transformer and Gaussian Process (GP) model against the Martingale property baseline model which assumes that the caseload will not change, i.e. that we have zero predictive power. At a forecast horizon T days in the future, the last observed caseload, μ_t, is used to forecast the caseload: $\mu_{t+T} = \mu_t$.

Gaussian Process Model - GP models were shown by Roberts et al. (2013) to perform well in contexts where prior knowledge regarding the appropriate model is limited. The difficulty in inferring the appropriate parametric model in infectious disease modeling led Lampos et al. (2017), Lampos et al. (2021) and Zou et al. (2018) to adopt a GP time-series model to predict future infectious disease caseloads. More modern methods have since outperformed GP models in time-series forecasting, so this GP model provides a further benchmark to the Transformer model outlined below. Our work uses a radial basis function (RBF) Kernel to specify the covariance function.

Transformer model - Transformers have predominantly been used with textual (Vaswani et al., 2017) and image-based data (Ye et al., 2019); however, the auto-regressive properties of a masked self-attention layer mean that structurally transformers can obey causality. As a result, many papers have used transformers successfully to model time-series data (Lim et al., 2021), (Zerveas et al., 2021). Both papers reported that transformer models significantly outperformed the statistical, recurrent and convolutional comparison methods. This success has been replicated in disease modelling by Wu et al. (2020). Thus, transformer-based time-series models represent the state-of-the-art in many comparable contexts, motivating its use in this framework. The architecture that is used in this paper mimics that of Vaswani et al. (2017) and Alexandrov et al. (2019).

5.2 Time-Series Evaluation

For the Time-Series task, the prediction error of the forecasts is reported in an ablation study, using the same forecast horizons as the Threshold task. Different feature types make up the covariate set and are compared against the univariate case.

Table 4 shows the main ablation study, which averages the root-mean-square-error (RMSE) across the different forecast horizons and states. The results show the overall behaviour of the different feature types. The first conclusion is that the Transformer model significantly outperforms the benchmark and GP model. Poor GP model results are also seen in Lampos et al. (2021), where their persistence model outperforms the univariate and multivariate GP forecasts in multiple countries. Due to the GP model's weaker performance, further analysis will involve the Transformer model, which better learns the caseload time-series. State-level results are displayed in Figure 2 and show that the Transformer performs well at modelling the time-series data.

Data Source	Martingale	GP	Transformer
univariate	.503	.269	**.160**
+ T_{RoB}	”	.283	**.161**
+ M	”	.287	**.166**
+ G	”	.266	**.162**
+ T_{RoB} + G	”	.269	**.181**
+ T_{RoB} + M	”	.304	**.173**
+ M + G	”	.275	**.171**
+ T_{RoB} + G + M	”	.269	**.180**

Table 4: The RMSE error averaged across the same forecast horizons and states that are used in Section 4.1.

The more meaningful conclusion that can be drawn from Table 4 is that the T_{RoB} features no longer provide the same benefit as in the Threshold task. The results in Section 4.1 are clear: Reddit data provides a strong indication of an imminent rise in COVID-19 caseloads; however, from the results in Table 4 it is apparent that this is not carried over to the Time-Series task. There is some degradation in performance as the feature set size increases, with the performance decreasing as the number of feature types increases. A possibility is that information that the T_{RoB} features provide is outweighed by the performance costs of having a large number of variables. There is some evidence of performance improvement when looking at longer forecast horizons. Each covariate set combination has a different performance profile and

Data Source	Av.	7D	14D	21D	28D
uni	.160	.156	.147	.164	.174
+ T_{RoB}	.161	.156	.166	.158	.165
+ M	.166	.157	.148	.120	.241
+ G	.162	.153	.147	.142	.206
+ T_{RoB} + G	.181	.159	.151	.225	.188
+ T_{RoB} + M	.173	.174	.139	.196	.184
+ M + G	.171	.161	.143	.188	.193
+ T_{RoB} + G + M	.180	.154	.142	.194	.228

Table 5: The RMSE error of a Transformer model averaged across all states at varying forecast horizons, using the same highlighting criteria as Table 1.

Size rank	Topic	ID	Frequency	Top 5 words
1	Masks	95	10699	mask, wear, masks, wearing, gloves
2	Unemployment	138	7591	unemployment, claim, pay, money, rent
3	Appreciation	181	3508	thank, thanks, appreciate, good, sharing
4	Schools	120	2808	school, kids, schools, teachers, students
5	Temporal statistics	152	1290	weeks, phase, ago, months, week
6	Lockdown frustration	75	1217	closed, shut, f**k, close, die
7	Agreement	197	892	yes, agree, yeah, exactly, sure
8	Festivities	96	879	thanksgiving, christmas, family, people, party
9	Vaccines	196	877	vaccine, vaccines, vaccinated, vaccination, people
14	Illness	178	569	cough, fever, symptoms, asthma, throat
17	Gyms	50	487	gym, gyms, fitness, open, exercise
19	Trump	218	378	trump, people, stupid, inslee, president

Table 6: The notable clusters from the r/CoronavirusWA subreddit using a SRoBERTa-NLI-STSb-base language model. The Frequency column represents the number of comments that are included in the cluster.

Table 5 shows that in every combination, besides T_{RoB} in isolation, there is an inflection in RMSE error at $\tau = 28$. The consistent performance across all forecast horizons shows that some valuable information is exploited here.

6 Discussion

Transparency is a vital component to understanding the impact and relevance of these results. Knowing the contents of our Reddit features can help us understand the information they provide to the prediction. We took the top 5 non-stop words from the cluster to characterise each cluster and manually named them for better comprehension. Table 6 shows the largest clusters from the SRoBERTa-NLI-STSb-base representation of the r/CoronavirusWA subreddit. These topics identify precise semantic concepts that intuitively provide relevant information for a caseload prediction.

As mentioned, the advantage of the Threshold task is that it provides greater interpretability than the more black-box time-series models. Therefore, the Threshold task is used to understand which features are important to the prediction. Table 11 shows the weightings of the most important features at $\delta_r = 0.6$ & $\tau = 7$ days. The cultural differences of the states can be seen via these features, most obviously the *Houston* feature in Texas and the *Desantis* feature in Florida. The *Spring Break* cluster is only seen in Florida, a state that is famed for this holiday tradition and was a large contributor to an increase in non-COVID-19 compliant events that resulted in an increase in cases at the beginning of the pandemic. Equally, the *Guns* and *Safety* features in California likely identify the strong negative reaction from the libertarian community within California to what were the most stringent lockdown restrictions from any of the analysed states.

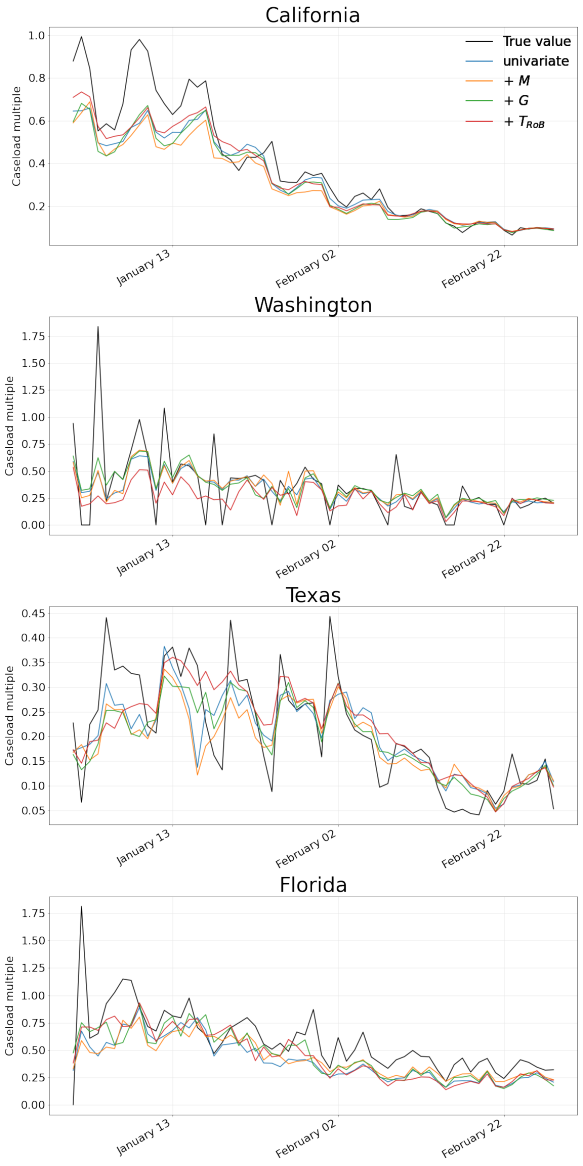

Figure 2: State forecasts at $\tau = 7$. The univariate forecast is compared against three multivariate forecasts where the M, G & T_{RoB} features make up the covariate set.

State	Topic	ID	Importance	Top 5 words
Washington	Working	107	.21	work, office, home, headquarters, let
	Illness	178	.14	cough, fever, symptoms, asthma, throat
	Quarantine	136	.08	quarantine, facility, people, outside, think
	Schools	120	.08	school, kids, schools, teachers, students
	Statistics	150	.07	trendline, graph, using, ggplot2, plotted
California	Illness	163	.12	cough, throat, fever, chest, symptoms
	School closure	122	.09	schools, close, school, closing, closed
	Guns	60	.09	gun, guns, shoot, firearms, buy
	Safety	103	.08	safe, stay, luck, protect, safer
	Flu	166	.06	flu, pneumonia, influenza, season, spanish
Texas	Data	130	.20	source, data, information, info, sources
	Voter Fraud	72	.10	vote, mail, voter, voting, fraud
	Houston	78	.07	houston, harris, county, area, houstonian
	Doctors	138	.06	doctor, doctors, medical, physician, telemedicine
	Illness	155	.04	fever, cough, allergies, asthma, symptoms
Florida	Spring Break	22	.22	spring, break, bike, week, breakers
	Social Media News	111	.19	reddit, facebook, news, echo, chamber
	Statistics	106	.05	numbers, data, believe, trust, graph
	Illness	94	.04	drug, people, fever, virus, sick
	Desantis	67	.04	desantis, care, deathsantis, dbpr, ron

Table 11: Important features from each of the key states at $\delta_r = 0.6$ & $\tau = 7$ days.

The libertarian trait within California is best characterised by the Prop 22 ballot initiative[3] which identifies a political attitude not aligned with strict lockdown measures. Alongside these differences, the *Illness* feature is highly weighted in all states. The use of this feature in all short-term predictions might explain the success of prior work that used static tracking words such as "Influenza", "fever", "headache", etc. (Samaras et al., 2020), (Iso et al., 2016); discussion about symptoms is indicative of a rise in cases in all states. It is clear, however, that exclusive use of symptomatic features is not optimal, since other topics besides symptomatic conversation are useful for the prediction.

7 Conclusion

Reddit data performs well at discerning different trend signals for COVID-19 caseload increases in the Threshold task. Reddit features alone achieved high accuracy at most threshold increases but were especially strong when identifying whether the caseload was likely to double in the next 14 days, achieving an accuracy of .970. However, the value seen in the Threshold task was not as evident in the Time-Series task, with only marginal improvements seen at long forecast horizons. The characteristics of Reddit data make it appealing: it is readily available and updated in real-time, offering the means for monitoring infectious diseases in regions where reported data is unreliable; however global Reddit usage is not constant, and not every area has a subreddit, making our exact methodology hard to scale. As Reddit usage increases and disperses around the world or data from another social media site is adapted to fit within our pipeline, the methods used in this paper will become more scalable. Another notable conclusion is that the predictive information within Reddit data is better extracted by including transformer-based language models in the forecasting pipeline. Language model complexity appears to be linked with performance improvements in the Threshold task. Strong language models allow us to isolate highly specific features predictive of future caseload increases in an unsupervised setting.

8 Future work

More work can be done on feature selection for the Time-Series task. The value of our T_{RoB} features is evident in the Threshold task, but it is possible that the features are not being exploited optimally in the Time-Series task. On top of this, our methodology relies on using textual data that refers to a specific geographic location. Reddit's structure makes this simple; however, more data is needed to replicate our findings in regions where Reddit take-up is low. Geotagged posts and the geolocation of a user's home region are possible avenues for enlarging the social-media dataset. Finally, the unsupervised methodology outlined in this paper can be adapted to other fields in which a social media derived feature set is used, such as quantitative finance, election and logistics forecasting.

[3]URL: https://vig.cdn.sos.ca.gov/2020/general/pdf/topl-prop22.pdf (accessed: 29/12/2021)

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C. Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, Lorenzo Stella, Ali Caner Türkmen, and Yuyang Wang. 2019. Gluonts: Probabilistic time series models in python. *arXiv preprint arXiv:1906.05264*.

Meshrif Alruily, Mohamed Ezz, Ayman Mohamed Mostafa, Nacim Yanes, Mostafa Abbas, and Yasser El-Manzalawy. 2022. Prediction of covid-19 transmission in the united states using google search trends. *Computers, Materials and Continua*, 71(1):1751–1768. Funding Information: Funding Statement: This work is supported in part by the Deanship of Scientific Research at Jouf University under Grant No. (CV-28–41). Publisher Copyright: © 2022 Tech Science Press. All rights reserved.

Diren Archary and Marijke Coetzee. 2020. Predicting stock price movement with social media and deep learning. In *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–5.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):830–839.

Adam Bermingham and Alan Smeaton. 2011. On using Twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 2–10, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Jaroslav Bukovina. 2016. Social media big data and capital markets—an overview. *Journal of Behavioral and Experimental Finance*, 11:18–26.

Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. 2013. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Felix Drinkall and Janet B Pierrehumbert. 2021. Predicting covid-19 cases using reddit posts and other online resources. In *2021 Swiss Text Analytics Conference, SwissText 2021*.

Thomas Hale, Sam Webster, Anna Petheric, Toby Phillips, and Beatriz Kira. 2020. Oxford covid-19 government response tracker. *Blavatnik School of Government.*

Mark Huberty. 2015. Can we vote with our tweet? on the perennial difficulty of election forecasting with social media. *International Journal of Forecasting*, 31(3):992–1007.

Cornelia Ilin, Sébastien Annan-Phan, Xiao Hui Tai, Shikhar Mehra, Solomon Hsiang, and Joshua E Blumenstock. 2021. Public mobility data enables covid-19 forecasting and management at local and global scales. *Scientific reports*, 11(1):1–11.

Hayate Iso, Shoko Wakamiya, and Eiji Aramaki. 2016. Forecasting word model: Twitter-based influenza surveillance and prediction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 76–86, Osaka, Japan. The COLING 2016 Organizing Committee.

Vasileios Lampos, Maimuna S. Majumder, Elad Yom-Tov, Michael Edelstein, Simon Moura, Yohhei Hamada, Molebogeng X. Rangaka, Rachel A. McKendry, and Ingemar J. Cox. 2021. Tracking covid-19 using online search. *npj Digital Medicine*, 4.

Vasileios Lampos, Bin Zou, and Ingemar Johansson Cox. 2017. Enhancing feature selection using word embeddings: The case of flu surveillance. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 695–704, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. 2021. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ryan McConville, Raúl Santos-Rodríguez, Robert J Piechocki, and Ian Craddock. 2021. N2d: (not too) deep clustering via clustering the local manifold of an autoencoded embedding. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5145–5152.

Daniel J. McDonald, Jacob Bien, Alden Green, Addison J. Hu, Nat DeFries, Sangwon Hyun, Natalia L. Oliveira, James Sharpnack, Jingjing Tang, Robert Tibshirani, Valérie Ventura, Larry Wasserman, and Ryan J. Tibshirani. 2021. Can auxiliary indicators improve covid-19 forecasting and hotspot prediction? *Proceedings of the National Academy of Sciences*, 118(51).

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861.

Ming Ni, Qing He, and Jing Gao. 2017. Forecasting the subway passenger flow under event occurrences with social media. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1623–1632.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, Jingjing Tang, Sumit Agrawal, Wael Al Saeed, Taylor Arnold, Amartya Basu, Jacob Bien, Ángel A. Cabrera, Andrew Chin, Eu Jing Chua, Brian Clark, Sarah Colquhoun, Nat DeFries, David C. Farrow, Jodi Forlizzi, Jed Grabman, Samuel Gratzl, Alden Green, George Haff, Robin Han, Kate Harwood, Addison J. Hu, Raphael Hyde, Sangwon Hyun, Ananya Joshi, Jimi Kim, Andrew Kuznetsov, Wichada La Motte-Kerr, Yeon Jin Lee, Kenneth Lee, Zachary C. Lipton, Michael X. Liu, Lester Mackey, Kathryn Mazaitis, Daniel J. McDonald, Phillip McGuinness, Balasubramanian Narasimhan, Michael P. O'Brien, Natalia L. Oliveira, Pratik Patil, Adam Perer, Collin A. Politsch, Samyak Rajanala, Dawn Rucker, Chris Scott, Nigam H. Shah, Vishnu Shankar, James Sharpnack, Dmitry Shemetov, Noah Simon, Benjamin Y. Smith, Vishakha Srivastava, Shuyi Tan, Robert Tibshirani, Elena Tuzhilina, Ana Karina Van Nortwick, Valérie Ventura, Larry Wasserman, Benjamin Weaver, Jeremy C. Weiss, Spencer Whitman, Kristin Williams, Roni Rosenfeld, and Ryan J. Tibshirani. 2021. An open repository of real-time covid-19 indicators. *Proceedings of the National Academy of Sciences*, 118(51).

Stephen Roberts, Michael Osborne, Mark Ebden, Steven Reece, Neale Gibson, and Suzanne Aigrain. 2013. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550.

David Rother, Thomas Haider, and Steffen Eger. 2020. CMCE at SemEval-2020 task 1: Clustering on manifolds of contextualized embeddings to detect historical meaning shifts. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 187–193, Barcelona (online). International Committee for Computational Linguistics.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Loukas Samaras, Elena García-Barriocanal, and Miguel-Angel Sicilia. 2020. Comparing social media and google to detect and predict severe epidemics. *Nature - Sci Rep 10*.

Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. 2020. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1728–1736, Online. Association for Computational Linguistics.

Boulesteix AL. Zeileis A. et al. Strobl, C. 2007. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics*, 8(1):1–21.

Youqiang Sun, Jiuyong Li, Jixue Liu, Christopher Chow, Bingyu Sun, and Rujing Wang. 2015. Using causal discovery for feature selection in multivariate numerical time series. *Machine Learning*, 101(1):377–395.

Laure Thompson and David Mimno. 2020. Topic modeling with contextualized word representation clusters. *arXiv preprint arXiv:2010.12626*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lijing Wang, Xue Ben, Aniruddha Adiga, Adam Sadilek, Ashish Tendulkar, Srinivasan Venkatramanan, Anil Vullikanti, Gaurav Aggarwal, Alok Talekar, Jiangzhuo Chen, Bryan Lewis, Samarth

Swarup, Amol Kapoor, Milind Tambe, and Madhav Marathe. 2020. Using mobility data to understand and forecast covid19 dynamics. *medRxiv*.

Qing-Guo Wang, Xian Li, and Qin Qin. 2013. Feature selection for time series modeling. *Journal of Intelligent Learning Systems and Applications*, 5(03):152–164.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. 2020. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

L. Ye, M. Rochan, Z. Liu, and Y. Wang. 2019. Cross-modal self-attention network for referring image segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10494–10503, Los Alamitos, CA, USA. IEEE Computer Society.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*, KDD '21, page 2114–2124, New York, NY, USA. Association for Computing Machinery.

Bin Zou, Vasileios Lampos, and Ingemar Cox. 2018. Multi-task learning improves disease models from web search. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 87–96, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

A Clustering algorithms hyperparameter tuning

An exhaustive search has been conducted to find the optimal k parameter (number of clusters) for KM and GMM clustering to compare their optimal configurations against HDBSCAN. The standard Silhouette score method was trialled for fine-tuning the k parameter, but the result was $k = 1$, perhaps indicating the unsuitability of KM and GMM for this task. Figure 3 is a plot of the Silhouette score (Rousseeuw, 1987) of a KM clustering algorithm for different values of k on the UMAP reduced SDistilBERT-NLI-STSb-base embeddings space. The maximum Silhouette score should be the most

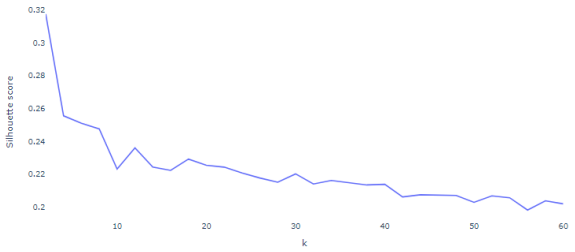

Figure 3: Silhouette score vs. k using a KM clustering algorithm on UMAP-DistilBERT embedding space.

appropriate k value if the data is divided into distinct clusters. The maximum Silhouette score at $k = 1$ indicates that the data is structured into one central cluster with high and low-density areas.

Since the Silhouette score does not provide an obvious k parameter, and yet there needs to be some proof that HDBSCAN is a better algorithm than KM and GMM, an exhaustive search for the optimal k on the development data is conducted to prove that KM and GMM are not suitable for the task. k is tuned using the performance on the r/CoronavirusWA data from 01/03/2021 to 17/01/2021 with UMAP dimensionality reduction.

The same search was conducted to find the optimal k for the PCA space; for KM, the value was 75, and for GMM, the value was 125 on the DistilBERT embedding space. These values of k were used for the testing in Appendix B.

B Dimensionality reduction and clustering algorithms

A test was carried out to see which combination of dimensionality reduction and clustering algorithms resulted in the best overall performance. The different algorithms were tested using SDistilBERT-NLI-STSb-base representations of the comments. The two dimensionality reduction techniques used were PCA and UMAP; the three clustering techniques used were GMM, KM and HDBSCAN. The k values derived in Appendix A were used for the GMM and KM clustering, and the evaluation pipeline used is the Threshold task described in Section 4.

The results from Table 9 show that the combination of UMAP and HDBSCAN is the best combi-

Clustering algorithm	k	Average	7 days	14 days	21 days	28 days
GMM	25	.659	.799	.645	.594	.596
	50	.691	.838	.691	.614	.622
	75	.698	.845	.689	.618	.639
	100	.702	.831	.703	.612	.662
	125	.714	.827	.709	.659	.664
	150	.716	.850	.718	.619	.676
KM	25	.706	.883	.685	.620	.635
	50	.636	.681	.714	.622	.528
	75	.677	.781	.769	.607	.550
	100	.702	.787	.678	.647	.695
	125	.663	.757	.754	.611	.531
	150	.658	.768	.687	.621	.556

Table 8: The average performances of an RF classification model using KM and GMM clustering across all thresholds at different values of k on the r/CoronavirusWA subreddit. The comment-level SDistilBERT-NLI-STSb-base representations' dimensionality was reduced via UMAP. The light grey indicates the highest performing instance of each model setup. The dark grey indicates the highest average performing model configuration.

Language model	Dim. reduction	Clustering	Average	7 days	14 days	21 days	28 days
DistilBERT	PCA	HDBCSAN	.722	.821	.724	.678	.667
		KM	.716	.807	.675	.677	.704
		GMM	.714	.808	.715	.651	.680
	UMAP	HDBCSAN	.807	.905	.827	.759	.737
		KM	.706	.883	.685	.620	.635
		GMM	.716	.850	.718	.619	.676
		Average	.730	.846	.724	.667	.683

Table 9: The average performance, on the r/CoronavirusWA subreddit, of an RF model across all thresholds at different prediction horizons for each of the model pipelines using only T_{RoB} features. The variables and highlights are the same as in Table 8.

nation of algorithms; the UMAP-HDBSCAN combination is the best performing pipeline across all prediction horizons.

C Aggregated Sequence Classification models

As mentioned in Section 1, the most obvious way to incorporate the modern transformer-base language models is to formulate the problem as an Aggregated Sequence Classification (ASC) task. It has been shown that BERT and other similar models are well adapted to performing sequence classification, and this has become a common usage of these language models (Devlin et al., 2019). Therefore, it is important to trial a model that incorporates this more standard methodology before trialling other feature identification methods.

For evaluation, we trialled two language models: BERT-base-uncased, and a domain adapted version of BERT-base-uncased trained on the r/Coronavirus subreddit - CoFReBERT (**Co**VID-19 **F**orecasting from **Re**ddit **BERT**). The language models are then fine-tuned on a Sequence Classification task in which the [CLS] token encodes the "up" or "down" class, indicating a possible increase or decrease in the number of cases. The adapted models are referred to as ASC-BERT and ASC-CoFReBERT. The model is trained on balanced classes with a 4:1 train-test split, where each day is assigned to be a test or a train day, and all comments written on a particular day are categorised together. Once the model labels each comment within the test set as either "up" or "down", the majority class on a given test day is assigned as the prediction for that day.

Models	Av.	7D	14D	21D	28D
ASC-BERT	.631	.769	.655	.561	.537
ASC-CoFReBERT	.701	.846	.690	.634	.634
T_{RoB}	.869	.923	.896	.810	.855
T_{BoW}	.765	.780	.808	.804	.791

Table 10: The average performance, on the r/CoronavirusWA subreddit, across all thresholds at four different prediction horizons. The variables and highlights are the same as in Table 8.

From the results in Table 10, it is clear that the models do not perform as well as hoped in comparison to the traditional static word features and the features outlined in this paper. The main reason for this is likely to be noise from the unsupervised labelling process. Comments that are either unrelated to the prediction or indicate an opposite caseload

trend are included in the prediction. Without manual labelling, it is hard to reduce this noise; however, that would result in investigator bias entering the prediction. Furthermore, it is not completely clear whether a comment is indicative of a rise in cases, shown by the variety of topics considered important to the prediction in Table 11. Therefore, the structure of the ASC models is not well adapted to the task of predicting COVID-19 cases.

D Training and software details

Python Packages The sentence-embedding models from (Reimers and Gurevych, 2019) were used to encode the Reddit post representations using the `sentence-transformers` Python package. The time-series models were both implemented using the `gluonts` Python package (Alexandrov et al., 2019). The ASC models outlined in Appendix C use the BERT-base-uncased model from the `transformers` package and the ASC-CoFReBERT model was trained using the `run_mlm.py` file in the library.

Training Pararmeters Besides the analysis detailed earlier in the Appendix, we do not perform hyperparameter tuning but use common hyperparameter values for all calculations in this paper. For the Random Forest model in the Threshold task, the number of trees is 100, and the maximum tree depth is 20. The Time-Series models were trained over 50 epochs and used default parameter values. The ASC-CoFReBERT model was trained with standard parameter values, using a batch size of 128 and a dropout probability of 0.1.

Computation All experiments in the main body of the paper were run on a personal computer, the ASC model in Appendix C was run on the. The ASC model was run on a Tesla P100 and took between 3 to 6 hours to run, depending on the size of the subreddit.

Licenses There are licenses associated with the use of some of the data and Python packages used in this paper. The OxCGRT dataset and Pushshift API are open access under the Creative Commons Attribution CC BY and 4.0 International standards. The COVID-19 Tracking Project, `gluonts`, `transformers` and `sentence-transformers` Python packages are licensed under the Apache License 2.0.