Finding Qs: Profiling QAnon Supporters on Parler

Anonymous Author(s)

Supplementary materials

Stance detection

We used stance detection to compute the average stance of a user towards QAnon. Specifically, we continued pre-training of BERT(Devlin et al. 2019) on a large corpus of approximately 5 million posts from Parler. In addition, we added the most important stance tokens towards/against QAnon using the methodology outlined in (Kawintiranon and Singh 2021) to the original BERT vocabulary. Overall, this should allow our model to capture better Parler-specific language compared to using standard BERT for the subsequent downstream task of stance detection (Kawintiranon and Singh 2021). In the next step, we fine-tuned our new language model on a corpus of 1250 stance labeled posts from Parler and computed the average stance of a user towards QAnon. The hyperparameters used for the pre-training and fine-tuning are listed in Table 1. For the pre-training, we used one Nvidia Titan V and a Nvidia Titan Xp. The fine-tuning was conducted on a Nvidia V100.

	Pre-training	**Fine-tuning**
Hyperparameter	**Value**	
Batch size	8	8
Learning rate	$5^{*}10^{-4}$	$5^{*}10^{-5}$
Number of epochs	2	3

Table 1: Hyperparameters used for the pre-training and fine-tuning of BERT.

Robustness checks

We check the robustness of our machine learning approach by a series of additional analyses:

Alternative classifiers: We repeat our main analysis with alternative classifiers. For this purpose, we fit three additional classifiers: (1) A LASSO model to assess the relative gain of considering non-linearites. (2) A random forest model as a common, out-of-the-box alternative to our XGBoost classifier. (3) A neural network with the same neural architecture search as in earlier works for user profiling (e. g., predicting verified status on Twitter (Paul et al. 2019)).

The training of the above classifiers was conducted as follows. We again trained separate classifiers for each of the three feature groups as well as a classifier for the combined set of all features. Moreover, we again trained each classifier using 10-fold cross-validation in combination with grid search.

Tbl. 2 shows the prediction performance for different feature groups and machine learning classifiers. Overall, we find that XGBoost performs best, as documented by the highest ROC AUC. This justifies why we selected this classifier for our main analysis. However, more importantly, all of the other classifiers reach a similar performance. For example, the random forest classifier using all features achieves a ROC AUC of 0.75, while the corresponding XGBoost model is only marginally better with a ROC AUC of 0.76. Similar patterns are observed for user features (0.73 vs. 0.74), linguistic features (0.67 vs. 0.67), network features (0.63 vs. 0.63) and content features (0.67 vs. 0.69). Here, the ordering of the different feature groups remains the same (i. e., user features have the largest predictive power, followed by content features). The neural network reaches similar performance levels. Of note, all of the ROC AUC are substantially better than a random guess with a ROC AUC of 0.50. The improvement is also statistically significant (with $p < 0.01$ for all feature groups and classifiers). These findings confirm that machine learning can discriminate QAnon from non-QAnon supporters across different classifiers. This contributes to the robustness of our findings for RQ3, thus establishing that the different feature groups lend discriminatory power in order to distinguish QAnon vs. non-QAnon supporters.

As expected, we observe a lower performance for the LASSO due to its linear structure (i. e., a ROC AUC of 0.70 when using all feature groups). This allows us to assess the relative gain due to considering classifiers that capture non-linearities, thus highlighting its overall relevance.

Feature combinations: We now evaluate the predictive performance of our classifier for different feature combinations (e. g., user and content features). Specifically, we test all possible combinations of our feature groups which amounts to a total of 15 different models (i. e., 4 single feature groups, 6 combinations of two feature groups, 4 combinations of three feature groups, 1 combination of all feature groups)

Input	XGBoost	LASSO	RF	NN
User features	**0.74*****	0.61***	0.73***	0.72***
Linguistic features	**0.67*****	0.67***	0.67***	0.62***
Network features	**0.63*****	0.56***	0.63***	0.62***
Content features	**0.69*****	0.68***	0.67***	0.65***
All features	**0.76*****	0.70***	0.75***	0.68***

p-values are obtained using the Mann–Whitney U-test (Mason and Graham 2002): $^{***}p < 0.01$; $^{**}p < 0.05$; $^{*}p < 0.1$
Best value per feature (= row) in bold

Table 2: Performance of discriminating QAnon supporters vs. non-QAnon supporters based on user, linguistic, network, and content features. Here, we report the ROC AUC for different classifiers: XGBoost, LASSO, random forest (RF), and a neural network (NN) to confirm the robustness of our findings.

and compare them to the results of the full model using a DeLong test (DeLong, DeLong, and Clarke-Pearson 1988). Following our main analysis, we train an XGBoost classifier for each feature combination using 10-fold cross-validation in combination with a grid search.

The results for all feature combinations are shown in Tbl. 3. As for our main analysis, we perform Mann–Whitney U-tests (Mason and Graham 2002) to confirm that the performance of each classifier is above that of a random guess. We find that all combinations of feature groups have predictive power based on which QAnon vs. non-QAnon supporters can be discerned. For each combination of feature groups, the respective ROC AUC is above 0.50. Further, the improvement is statistically significant ($p < 0.01$).

Comparing the ROC AUC of all combinations, we find that models including user features achieve higher predictive performance compared to models without user features. This is consistent with our main analysis where we find that user features have the highest predictive power. In addition, we find that models including user features score similar values in ROC AUC (e. g., user + linguistic, user + network + content) compared to the model using all features. The model using user, linguistic, and content features even reaches a marginally higher ROC AUC (0.77). Yet, the actual difference (i. e., before rounding) is less than 0.2 %. We also check if any model achieves higher ROC AUC scores at a statistically signifcantly level using a DeLong test. For all models, we find no statistically significant improvement in ROC AUC compared to the model using all features.

Feature selection: We now check if the exclusion of features with low predictive power can increase the classification performance. To do so, we train a LASSO on our complete feature group using 10-fold cross-validation in combination with a grid search. Subsequently, we select the features with non-zero coefficients in the LASSO. By this, we select 398 features out of all different feature groups (i. e., user, linguistic, network, content features). In the next step, we train a XGBoost classifier on the selected features using the training strategy of our main analysis. The new classifier achieves a predictive performance of 0.76 ROC AUC.

Input	ROC AUC	Sensitivity	Specificity	F1
User features	0.74***	0.77	0.71	0.75
Linguistic features	0.67***	0.62	0.72	0.66
Network features	0.63***	0.62	0.65	0.63
Content features	0.69***	0.67	0.70	0.68
User + linguistic	0.76***	0.78	0.74	0.76
User + network	0.74***	0.77	0.71	0.75
User + content	0.76***	0.77	0.74	0.76
Linguistic + network	0.67***	0.64	0.73	0.67
Linguistic + content	0.70***	0.66	0.73	0.69
Network + content	0.69***	0.67	0.71	0.68
User + linguistic + network	0.74***	0.77	0.71	0.75
User + linguistic + content	0.77***	0.79	0.75	0.77
User + network + content	0.76***	0.77	0.75	0.76
Linguistic + network + content	0.70***	0.67	0.74	0.69
All features	0.76***	0.77	0.75	0.77

p-values are obtained using the Mann–Whitney U-test (Mason and Graham 2002): $^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 3: Performance of classifying users into QAnon (=1) vs. non-QAnon (=0) supporters based on different feature groups.

This is similar to our classifier based on the complete feature group (ROC AUC = 0.76). We also test whether feature selection can statistically significantly improve the predictive performance of our classifier using a DeLong test (DeLong, DeLong, and Clarke-Pearson 1988). We find no statistically significant improvement ($p = 0.68$). Hence, the exclusion of features with low predictive power does not increase the classification performance.

Hyperparameter tuning

For the training of the classifiers reported in the main paper we use 10-fold cross-validation in combination with a grid search. Overall, we use the following 5 classifiers:

1. **XGBoost**: Here, we vary the learning rate (*LearningRate*), the maximum tree depth (*MaxTreeDepth*), the subsample ratio of training instances (*Subsample*), the subsample ratio of variables to construct each tree (*ColSample*), and the number of boosting iterations (*Nrounds*). The corresponding tuning range for each parameter is reported in Tbl. 4.
2. **LASSO** was tuned using a custom grid for λ (i. e., 100 equally spaced values from 0.001 to 1000).
3. **Random forest** (RF) use grid varying the number of variables to split at each node (*Mtry*) and the splitting rule (*Splitrule*). Again, the corresponding tuning range for each parameter is shown in Tbl. 4.
4. **NN (literature)**. We used state-of-the-art network consistent with prior literature (Paul et al. 2019). Here, we used an additional validation set, a batch size of 32, and early stopping. Consistent with (Paul et al. 2019), we further used ReLU activation functions and Adam for optimization. We set the number of hidden layers to 3 and the number of neurons to 100, 30, 10, respectively, as in

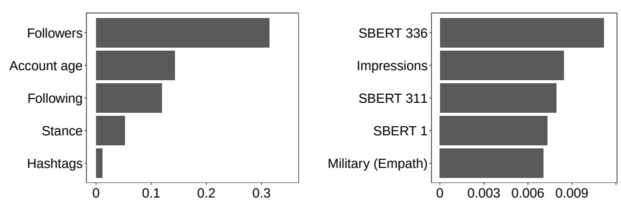
(Paul et al. 2019). This was done to obtain results where the network architecture is comparable to earlier research. The network was implement in Keras.

5. **NN (custom)**. We implemented a customized neural network, where, instead, all parameters (i. e., number of layers, neurons per layer, dropout, batch size, etc.) were manually tuned. The network was again implement in Keras. However, this led to overall similar results to the NN (literature). Hence, we omitted NN (custom) for brevity.

Classifier	Hyperparameter	Tuning range
XGBoost	*LearningRate*	[0.3, 0.4]
	MaxTreeDepth	[1, 2, 3, 4, 5]
	Subsample	[0.5, 0.625, 0.75, 0.875, 1]
	ColSample	[0.6, 0.8]
	Nrouds	[50, 100, 150, 200, 250]
RF	*Mtry*	[2, 46, . . . , 692, 693]
	Splitrule	[“Gini”, “Extratrees”]

Table 4: Tuning range for XGBoost and random forest (RF) hyperparameters.

Supplementary figures



(a) Followers (b) Followees

Figure 1: Feature importance of (a) top 1–5 and (b) top 6–10 features discriminating QAnon and non-QAnon supporters out of all user, linguistic, network, and content features.

Supplementary tables

Variable	#Total	Per user
Followers	279,298,787	437
Followees	194,442,610	304
Posts	157,849,243	247
Comments	42,321,848	66
Impressions (posts)	25,675,925,576	40,190
Upvotes (posts)	256,789,008	402
Upvotes (comments)	102,436,423	160
Downvotes (comments)	3,854,418	6

Table 5: Overview of the variables. Reported are the cumulative counts across all users (#Total) and the mean values per user (per user).

The following list of keywords was used to identify QAnon-supporters (adapted from (Sharma, Ferrara, and Liu 2022)):

anons	anon
qanons	qanon
q	thegreatawakening
greatawakening	wwg1wga
wgaworldwide	qarmy
obamagate	pizzagate
savethechildren	saveourchildren
taketheoath	deep state
deepstate	deepstatecoup
deepstatecabal	deepstateexposed
plandemic	scamdemic
sheepnomore	adrenochrome
thestorm	followthewhiterabbit
downtherabbithole	thesepeoplearesick
wearethenewsnow	pizzagateisreal
thestormisuponus	newworldorder
darktolight	clintonbodycount

Table 6: List of keywords used to identify QAnon supporters (adapted from (Sharma, Ferrara, and Liu 2022)).

References

DeLong, E. R.; DeLong, D. M.; and Clarke-Pearson, D. L. 1988. Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. *Biometrics* 44(3): 837. ISSN 0006341X.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Kawintiranon, K.; and Singh, L. 2021. Knowledge enhanced masked language model for stance detection. In *NAACL*.

Mason, S. J.; and Graham, N. E. 2002. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society* 128(584): 2145–2166.

Paul, I.; Khattar, A.; Chopra, S.; Kumaraguru, P.; and Gupta, M. 2019. What sets verified users apart? Insights, analysis and prediction of verified users on Twitter. In *WebSci*.

Sharma, K.; Ferrara, E.; and Liu, Y. 2022. Characterizing online engagement with disinformation and conspiracies in the 2020 U.S. Presidential Election. In *ICWSM*.