A classification system to detect questionable information

Pedro Mota
Dept. of Computer Science of Faculty of Science
University of Porto
Porto, Portugal
up201805248@up.pt

Tatiana Araújo
Dept. of Computer Science of Faculty of Science
University of Porto
Porto, Portugal
up201805169@up.pt

***Abstract*—Propaganda has been around for centuries and often misleads the public to make wrong decisions. It also poses serious threats to social order. In the digital age, social networks became the latest mean of communication to be abused to spread misinformation. Specially in politics, misinformation is intended to polarize and change the public opinion, making it a major enemy of a democratic society. In this study, we did a exploratory analysis of a set of tweets retrieved before the 2020 U.S. Presidentials and built a data mining pipeline to detect if a tweet contained questionable information or not.**

***Index Terms*—data mining, machine learning, twitter, misinformation, R**

I. Introduction

From a dataset comprised of 18k tweets about the last U.S. Presidential election, which were retrieved before the election was held, labeled as questionable or not, the goal of this project was to create a classification system, using Machine Learning algorithms, in order to identify where a new (unseen) tweet is disseminating questionable information, or not. Despite the simple problem statement, it is a hard task because natural language is full of ambiguities and imprecise characteristics, therefore making it impossible for a human to devise, manually, a set of rules to determine if a tweet contains questionable information or not. With this in mind, we tried to tackle this problem by exploiting, for the questionable and non-questionable tweets, the different vocabulary used (which can be captured by a term frequency analysis), the different sentiment and emotion distribution and, also, the conflicting valence found in the tweets, usually called *emotional entropy*, which can be thought of as a measure of unpredictability and surprise based on the consistency or inconsistency of the emotional language in a given message. For all the analysis and model training, we used the language R and some of its packages from the CRAN.

II. Exploratory Data Analysis

A. Features

Our initial data set consisted of 11 variables: *id, questionable_domain, user_friends_count, user_followers_count, user_favourites_count, user_verified, user_description, description, title, favorite_count, retweet_count* and *contains_profanity*, whose meaning can be found on the official documentation of the Twitter API. We, immediately, removed the attribute *title*, since it was redundant.

Table I: Correlation between user-verification and the tweet questionability

	Questionable tweet	**Non-questionable**
Verified user	0.8627451	6.444759
Non-verified user	99.1372549	93.555241

B. Missing values, Duplicates and Outliers

Regarding the existence of missing values and duplicates, with the help of the package *dlookr*, we found that we had none of them. With respect to outliers, we concluded the same, mainly, because the data collection was autonomously done and it doesn't suffer from reading problems, such as sensor data. In any case, we inspected the range of the values for the attributes and they all seemed plausible.

C. The numerical attributes

We started by analysing how the user activity (represented by the attributes *user_verified*, *user_friends_count*, *user_followers_count* and *user_favourites_count*) is related to the tweet's questionability.

Firstly, regarding the *user_verified* attribute, we expected that verified users don't post questionable tweets, since verified users usually represent notable people or organizations, such as government officials. With table I, we confirmed our guesses, since the vast majority of verified users do not post questionable tweets, therefore making *user_verified* a relevant attribute.

Now, regarding the *user_followers_count*, for the same reason as before, we expect to see users with an high number of followers, to not post questionable tweets. With respect to *user_friends_count*, we were more unsure about our intuitions, because an high number of friends can mean a more informed user, since it follows a lot of other users, but it can also mean, for instance, that we are in a presence of a Twitter bot, which follows almost everyone automatically, in order to be noticed and spread whatever is their message, which may be of a questionable nature. However, we found that the more friends a user has, they post less questionable tweets. Lastly,

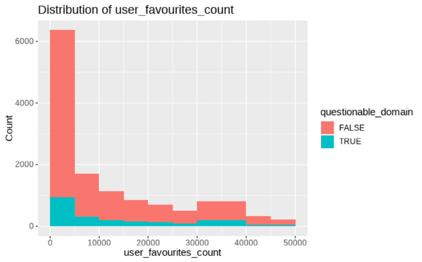


Figure 1: Distribution for the user activity regarding the tweet's questionability

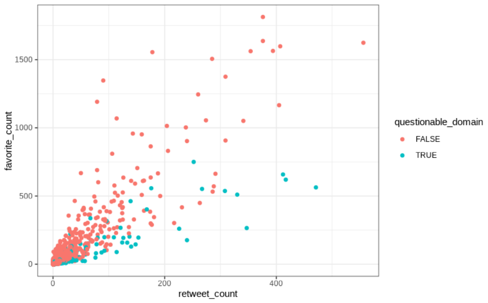


Figure 2: Correlation between the tweet's activity and the target variable

with respect to *user_favourites_count*, we don't expect to see a direct correlation with the target variable, in part, for the same reason as the *user_friends_count* attribute, but, again, we found that the more favourites a user has, the less questionable tweets they post. Fig. 1 shows their distribution regarding the tweet's questionability.

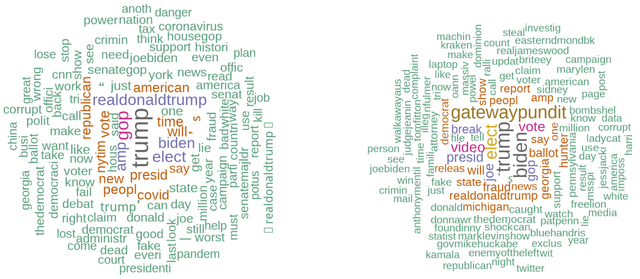
Now, let's see how the tweet activity (represented by the attributes *favorites_count* and *retweets_count*) is related to the questionability of the tweet. Fig. 2 shows their relation. As expected, the higher the engagement of the tweet, the less likely it is for the tweet to contain questionable information.

We also explored the correlation between within variables. As we can see from table II - a correlation in the interval [0, 0.3[is represented by an empty space, [0.3, 0.6[by a dot, [0.6, 0.8[by a comma and [0.8, 1[by a star - we have a lot of correlation between our variables, namely between the ones regarding the user activity and between the ones regarding the tweet activity.

Table II: Correlation Matrix

	(a)	(b)	(c)	(d)	(e)	(f)	(g)
user_friends_count (a)	1						
user_followers_count (b)	+	1					
user_favourites_count (c)	.	.	1				
user_verified (d)		.		1			
favorite_count (e)	.	.		.	1		
retweet_count (f)	.	.		.	,	1	
contains_profanity (g)							1

Figure 3: Word clouds



((a)) Non-questionable tweets

((b)) Questionable tweets

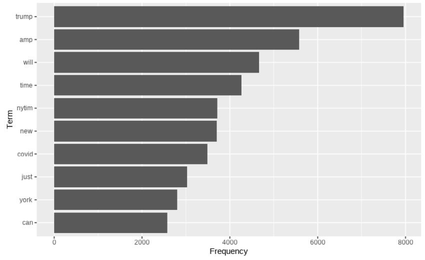


Figure 4: Term frequency for non-questionable tweets.

D. The description attribute

Finally, we will look into the content of the descriptions of the tweets, that is, the field *description*.

We created two corpus, one for the non-questionable and one for the questionable tweets. After this, we cleaned up each corpus, by removing numbers, punctuation, white spaces, hyperlinks and stop words. Fig. 3 shows the word clouds for non-questionable and questionable tweets, respectively.

As we can see, besides the obvious appearances of terms like *trump*, *realdonaldtrump* (which was the Twitter username for Donald Trump's official account), *biden*, there are other differences on the terms being used in the questionable and non-questionable tweets. Thus, this may help us classifying a tweet as questionable or not. Figs. 4 and 5 show, more precisely, the frequency of ten most frequent terms, for the non-questionable and questionable tweets.

We started the analysis of the most frequent terms by seeing how the main candidates, Biden and Trump, were portrayed in the non-questionable and questionable tweets. What we

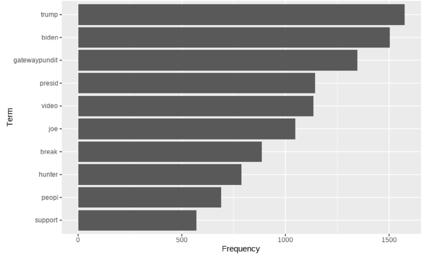


Figure 5: Term frequency for questionable tweets.

saw, is that in the questionable tweets, the terms related to Trump were more neutral, generally, associated to the right-wing american world, but, with respect to Biden, they were more controversial, but, as far as we known, they are targeted to his son, Hunter Biden, which is a story that we will explore later. In the non-questionable tweets, it's actually curious to see Biden being associated to terms such as *enemi*, *china* and *socialist*, this really shows the dichotomy of Republicans vs Democrats, in the U.S. elections.

Other important term which appeared in the most frequent terms for the questionable tweets is the term *gatewaypundit*. The Gateway Pundit is an American far-right fake news website. With further analysis, we noticed that 36.5% of the questionable tweets contain the term *gatewaypundit*, but the same term only occurs roughly 0.03% in the non-questionable tweets. This gives the term *gatewaypundit* a high discriminant power to distinguish between the two sets of tweets.

Also, in the questionable tweets, it appears frequently the word *hunter*, which, probably, refers to the son of Joe Biden, Hunter Biden. Our first idea was that the content of the tweets were probably related to his, supposedly, controversial life style and business activities, which was a topic, again, recovered with the war in Ukraine. We validated our idea by looking into the terms associated to his name. The ones with higher correlation were: *laptop*, *biden*, *email*, *text*, *girl*, *photo*, *porn* and *vacuum*. Excluding the term *biden*, which is his last name, the rest of the words are related to the history of his laptop, which allegedly belonged to him and was later recovered, and, with this, a trove of personal emails and photographs found on it, which lead to the various stories stated above. All of this was published less than three weeks before the presidential election, by the New York Post. Since he was the son of one of the candidates to the 2020 presidentials, he was a recurrent topic brought by the main opposition party. Since any of this is, at least, officially, confirmed, it was all classified as questionable information.

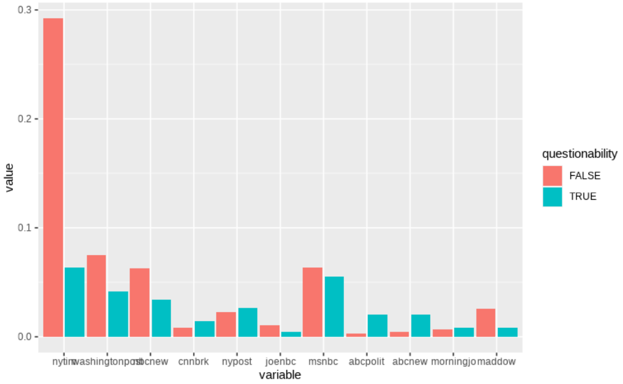


Figure 6: Distribution of the terms regarding newspapers and telivision hosts between the two sets of tweets

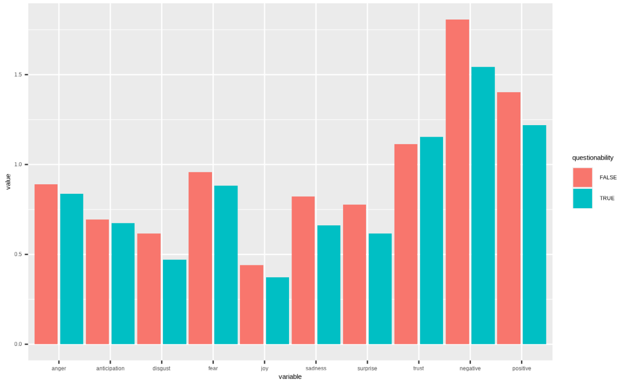


Figure 7: Emotion and Sentiment analysis

Regarding the non-questionable tweets, one term that appears frequent is the term *nytim*, which refers to the American daily newspaper *The New York Times*. We looked into the terms associated with *nytim* and found that the ones with higher correlation were all other newspapers or television news program hosts, such as *joenbc*, which refers to Joe Scarborough, a television host, political commentator, and former politician who is the co-host of Morning Jo and *maddow*, which refers to Rachel Anne Maddow, an American television news program host and liberal political commentator. What we thought that could be interesting was to analyse the frequency of these terms regarding the two sets of tweets, since, specially during the final stage of the presidentials campaigns, a lot is said about the impartiality of these news programs. Fig. 6 shows the distribution of these terms between the two sets of tweets. Sadly, only the *nytim* term seems to have a significant difference regarding the type of the tweet. We inspected the tweets that contained the term and it seems to be mostly citations of the news. Anyway, we would need major care to conclude something if a different distribution regarding the tweets appeared, since the questionability of the tweets mentioning any of these newspaper can be due to wrong citations and/or false accusations, which, with a analysis only from the distribution regarding both set of tweets, would influence the agency negatively, but, in reality, it's nothing related to the agency itself.

Next, we tried a sentiment and a emotional analysis, but in both sets of tweets, they have the same distribution, as we can see from fig. 7. This is a quite intuitive, because it's easy to imagine that a user can have both positive and negative emotions writing a tweet that is questionable or not. With further analysis, to our surprise, what we noticed is that in the questionability tweets, the emotions tend to reach higher absolute values than in the non-questionable tweets.

Finally, we also looked into the emotional ambiguity, i.e., the conflicting valence found in the tweet. We expect this to be important, since questionable tweets tend to be unpredictable and sometimes even contradictory. Surprisingly, we saw a evenly distributed metric entropy between the two sets of tweets, but, with experimental evidence, we saw that this variable was somewhat important.

III. Baseline model

In this section, we will present our baseline model. Firstly, we started by splitting our data set into two independent data

Table III: Baseline Model Results

Model	Accuracy	Precision	Recall	F1
kNN	0.83	0.86	0.96	0.91
Naive Bayes	0.30	0.86	0.19	0.31
XGBoost	0.85	0.85	0.99	0.91
Random Forests	0.85	0.85	0.99	0.92
Neural Networks	0.83	0.83	1.00	0.91

Table IV: Principal Component Analysis

	PC1	PC2	PC3	PC4	PC5	PC6
Standard Deviation	1.38	1.0799	1.04	0.99	0.93	0.91
Proportion of Variance	0.27	0.17	0.16	0.14	0.12	0.12
Cumulative Proportion	0.27	0.44	0.60	0.74	0.86	0.98

Table V: Improved Results

Model	Accuracy	Precision	Recall	F1
kNN	0.86	0.88	0.96	0.92
Naive Bayes	0.80	0.86	0.90	0.88
XGBoost	0.90	0.90	0.99	0.94
Random Forests	0.8919019	0.89	0.99	0.94
Neural Networks	0.82	0.89	0.90	0.89

Table VI: Confusion Matrix for the XGBoost model

		Prediction		
		False	Positive	Total
Reference	False	2237	263	2500
	Positive	5	187	192
	Total	2242	450	2692

sets, one in which we will perform our analysis and train our models and the other one that will be used to do the predictions. We used 15% of the data set for testing. We made sure that the distribution of the data between the training set and testing set were similar.

For the baseline model, we will use the entire data set as it is, excluding the *description* attribute. We decided to train five models (with variations), using the package *caret*:

- k-Nearest Neighbors (with k equals to 5, 8, 11 and 14)
- Naive Bayes (with Laplace correction of 1)
- XGBoost (using 50, 100, 150 and 200 iterations)
- Random Forests (with 300 trees)
- Neural Networks (we varied the number of neurons for only hidden layer from 1 to the number of attributes)

The results that we obtained are shown in table III. To sum up, the Random Forest model was the best model, overall. Note that the XGBoost and Neural Networks models are taking advantage of the fact that the target variable is unbalanced - they are simply classifying the tweet as non-questionable.

IV. Improvements

As we saw in section II-C, the numerical attributes were highly correlated. This can present a problem to our models. So, in order to remove this correlation, we could simply remove one of the correlated variables, but we decided to perform a principal component analysis (PCA) in order to remove the correlation between our variables and also reduce the dimensionality of our data set. Table IV shows the results that we got from the PCA. One can see that the first six components already explain 98% of the variability of the data. The last principal component, which is not represented in the figure, has a standard deviation of 0.41 and only explains 2% of the data. This means if we use the first six principal components, we will have one less variable than the seven original ones and we still maintain roughly 98% of the variability in the data. With this in mind, in our data, we substituted the old numeric attributes by the new principal components.

From section II-D, we saw that the term frequency may be useful to distinguish between both set of tweets. So, with this in mind, in order to gain advantage of this, we added 13 new features, which were named after terms that they represent. The value for every entry is their tf-idf score within the whole corpus. The chosen terms were: *trump*, *amp*, *time*, *nytim*, *presid*, *covid*, *gatewaypundit*, *biden*, *video*, *hunter*, *will*. Also, as stated, we saw that the in questionable tweets, the emotions tend to reach higher absolute values than in the non-questionable tweets, so, with this in mind, we added a new feature, named *extreme_emotion*, which is true if any of the emotions we had selected gets higher than 5, or false, otherwise.

V. Results

Fig. V shows the results that we got after we processed the data set. As we can see, every metric was improved, for every model.

We also tried to balance our data set, by doing both under-sampling and over-sampling and, also, only doing under-sampling but we got worse results in every metric we measured compared to when we simply ignored it.

Overall, our best model was a XGBoost model. Table VI shows the obtained confusion matrix.

VI. Conclusion

The biggest challenge was in the data pre-processing and feature engineering part, specially, dealing with the tweet's content - the attribute *description*. The reality is that it is not trivial to classify some information as questionable or not. For a particular and reduced data set, a simple syntactic and/or semantic analysis may provide good results, but, overall, it would requires us to process the statement being said and factually check it, against a knowledge base. For this particular problem, future work could pass from gathering new features, incorporating the time stamps of the tweets (in order to a more complete analysis as time evolved) and gather even more data, in order to improve our classifications models. With a good model, Twitter could use this in order to stop, or, at least, reduce, the spread of misinformation. In fact, it would be interesting to test this model in the 2024 upcoming elections, since it's expected to be, again, a race between Biden and Trump.