The Cryptocurrency Network

Cheung, Hau Yee (2013004)

Carlo De Dominicis (2026816)

Roberto Russo (2006665)

Ivan Kulazhenkov(2004275)

Abstract

A cryptocurrency is a form of virtual or digital money, a peer-to-peer, electronic cash system, which is an innovative technology that offers several benefits, such as fast transaction speeds, low costs, and the elimination of the need for a third-party intermediary to process transactions. They can be transferred through a computer or smartphone without an intermediate financial institution. Bitcoin is one of the most famous cryptocurrency, which operates free of any central control or the oversight of banks or governments. Twitter is one of the most popular information source available on the Web. In this project, we firstly scrape the data for building the network from Twitter. Then developed our deep learning network to predict semantic similarity. Finally, present here the hashtags relationship related to bitcoin by the NetworkX algorithm.

Before describing our work, here we describe how we divided the work among us. Even if for pragmatic reasons everyone of us has done only a part of the work, all of us were aware and conscious of everything that has been done. Everyone of us agreed on every part of the work, and the final work is the result of the efforts of everyone of us. Below the subdivision of work following the point of this report, where is specified what everyone of us has implemented:

1. Ivan has worked on data preprocessing, data cleaning, creating of naive ml method and lstm network for sentiment classification as well as hashtag preprocessing for network creation.

2. Roberto has implemented every part of section 4

3. Hau Yee has worked on data scraping, data preprocessing, graph visualization and part of the graph for network analysis

4. Carlo made some scraping and data cleaning, but mostly worked on Sentiment Analysis with VADER and BERT models, and manipulation of average hashtags' emotion to apply sentiment analysis to the network.

1. Introduction

This course project aims to build a network to show the relationship between hashtags added in the tweets related to Bitcoin. Such a network could then be applied for many relevant uses such as but not limited to providing references for finding related information to predict future movements.

Before actually building the network, we used to perform semantic analysis based on BERT Transformer, Long Short Term Memory(LSTM) Neural Net and Naive Bayes. The texting style will change, could be from one person to another, or even in different situations for the same person due to many factors. Due to variations in speaking styles, performing the semantic analysis can be a challenging task. This makes it an interesting project topic with much room for experimentation.

Then, we built a network to present the relationship between hashtags related to bitcoin according to the semantic analysis result.

All code is written in python in a google colab notebook. For the network implementation, we used the NetworkX library. The dataset we used was scraped from Twitter API.

In the following sections, we will subsequently present the dataset used for this project, the method for performing semantic analysis, the architecture for building the network, and the experiments performed as well as the obtained results. Finally, we briefly summarise which methods work best for us and discuss the problems encountered during our work.

2. Dataset

2.1. Data scraping

Online social networks are becoming more and more popular in the Internet era. Their prevalence in day to day life has resulted in large groups and communities of like-minded people being formed within the users. Not only that but Twitter offers an environment encouraging freedom of speech where people from all around the world can share information. After many years this has allowed the social media platform to become a real-time repository of knowledge in the form of both facts and opinions. We decided to leverage this platform as the main data source of our experiment. We scraped the data from Twitter through the use of a Python library and combined with some existing datasets to create a large data repository for our experiments. The narrowing down of user groups and communities provides an individual an easily accessible destination specifically tailored to them and their interests. This is something that we will leverage in our experiment through the use of sentiment analysis and topic specific analysis in the form of hashtags.

Data gathering was performed through twitter scraping. The focus of the scraper was on searching for the keyword 'Bitcoin' and pulling tweets from Twitter relating to that keyword. We selected only verified profiles' tweets to be included in the dataset. We combined the scraped data with data obtained through other sources such as readily available crypto oriented datasets. In total, we gathered over 2 million unprocessed tweets. Some important features of the data we gathered included but was not limited to tweet, language, hashtags, usernames, location, number of retweets, number of likes, number of replies, etc.

2.2. Tweet Data Cleaning

A tweet contains a lot of opinions about the data which are expressed in different ways by different users. The raw data having polarity is highly susceptible to inconsistency and redundancy. Any algorithm development hinges on a high quality data set and in natural language processing tasks, including sentiment analysis, it is even more crucial. As stated by Kotsiantis et al a lack of pre-processing especially the removal of noise and data outlier's can negatively impact algorithm performance [5]. Moreover, when dealing with a large sample size as was our case it is critical that we perform a representative sample of meaningful data before we begin our experiment. Data pre-processing is a technique that transforms raw data into an understandable format. The most critical part of our project comprised of data preparation and cleaning. The transformations and data selections that we did are as follows: clean text of emojis,hashtags url links, special characters, etc. , remove stop words from tweet text, remove elongated words from text that are misspelled to make them longer and take lastly take care of negation in the English language example being replacing words like doesn't to does not.

2.3. Tweet data pre-processing

The text data that we obtained from our tweets can be considered to be in sequence of character, of words or sequence of larger text options (such as sentences). Many language processing tasks, as well as the task our team will be working on, consider the textual data to be a sequence of words. In order to be able to construct a robust sentiment analysis tasks we implemented several pre-processing steps that are considered necessary by state of the art.

1. Tokenization - For machine learning and deep learning model to be able to understand text we need to represent each word in numerical terms. Through this we create a lexicon of all words present in the dataset with each being represented by a number.

2. Sequencing - After generating the tokenized word representation we represented each sentence of words as a sequence of numbers.

3. Padding - The final step of text pre-processing we needed was to account for tweet sentences of different lengths. We used a histogram distribution plot to pick the max length of the sentences that we would use. This is due to the fact that all neural networks require inputs with the same size. Once we have selected a max padding length we cut off tweets longer than this max limit and pad the values if the sentence is shorter. Most commonly the padding is done with the '0' token.

The above steps are important as their implementation has consistently shown significant increase in performance for neural networks [2]. More importantly we also see benefits in using a consistent pre-processing approach when running an experiment. This is why we adopted the same text processing techniques for both the tweet text and hashtag data.

3. Semantic analysis model

3.1. Sentiment analysis through transfer learning

After the sentiment analysis, the features included in the cleaned dataframe: User name, tweet, compound, sentiment and hashtags

3.2. Sentiment generation

After performing text processing the next step was generating a sentiment evaluation of the text in each tweet. This was sub divided into two distinct approaches.

- Sentiment Intensity Analyzer: We used VADER (Valence Aware Dictionary for Sentiment Reasoning) to create
- LSTM(Long Short Term Memory) network inference generation: Train LSTM using the VADER dictionary described above and use it to generate sentiment predictions for new tweet text.
- BERT(Bidirectional Encoder Representations from Transformers) BERT can be used for a wide variety of language tasks and has become state of the art in many of them. One of the main strengths of BERT is how large its parameter size is. Unlike the LSTM it does not suffer as much from over fitting as traditional neural networks. Performance of the model only increases with more data and more training steps. This approach was the final one we applied for inference of text sentiment.

Looking at the above in more detail; VADER is a model used for text sentiment analysis that can be used for both polarity (positive/negative emotions) and the underlying strength (intensity) of the emotion's polarity. This model proved to be beneficial for the primary reason of working well with unlabeled text data.

The VADER sentimental analysis model uses a dictionary mapping lexical features to to generate emotional intensity through sentiment scores. The sentiment score for text coming from a tweet can be calculated by the summation of the intensity belonging to each constituent word in the tweet.

For example the sentence "I love bitcoin a lot" would convey a strong positive sentiment primarily through the strength of love and lot words in that sentence.

3.3. Naive Sentiment inference

Before going ahead with the creation of our neural network. We wanted to obtain a baseline performance measure. This was done for two reasons. First of all, we did not know how feasible it was to perform sentiment inference on the scraped data that we had. Secondly, we were optimistic that a simple Naive Bayes(NB) algorithm could give us respectable performance at a fraction of the computational cost.

The equation below (3.3) shows the generic Gaussian NB solution. We tested both the Gaussian and the Multinomial versions of the algorithm in order to pick the best baseline for sentiment Inference. The final selected model ended up being the Multinomial Naive Bayes as it had a slightly better performance overall than the Gaussian model.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

3.4. Neural Network Sentiment inference

We previously discussed the approach we took to perform inference on sentiment in text. We decided to use a neural network commonly applied to text classification problems[4]; the LSTM. The pipeline that we used to get from tweet to prediction can be seen in figure (1). In more detail the steps we took were as follows:

1. (Tweet Text Processing): This initial step combines all the pre processing that we did for the text contained within our tweets. In simple terms we performed text cleaning and word separation here.
2. (Tweet Word Tokenization): The second text processing stage involved word tokenization and padding of tweets. We created a lexicon of words for vectorization and applied them to the entire dataset. After analysing the distribution of tweet lengths we decided to use a padding up to a maximum of 20 words.
3. (Create Embedding Layer): The creation of the word embedding layer provides context in terms of euclidean distance between words. Words that are more similar than others will as a consequence have a smaller euclidean distance between them. We initialised this layer using the full size of our lexical vocabulary.
4. (Create LSTM Layer(s)): In the network we used two LSTM layers. One with 64 and one with 32 neurons. Given the twitter topic and relatively short average tweet length this gave us very good predictive power.
5. (Add Dropout Layer): To limit any over fitting from training we also included a dropout layer. With a probability of 0.2 that the neuron would be zeroed.
6. (Add Embedded Linear Layer for Prediction): Our final step/layer was the output linear layer. Since our classification problem was binary our sigmoid output had a final value range between 0 and 1.

3.5. BERT Model for Sentiment Discrimination

Bidirectional Encoder Representations from Transformers (BERT) is the last approach we used to perform sentiment analysis on the scraped tweets. Thanks to its structure, BERT potentials brought to the NLP field great changes, also, it is used quite a lot in social and economic research fields, for this reasons we decided to apply this model this project.

Without go too much in details, BERT is a deep bidirectional network based on Transformer architecture, so it is

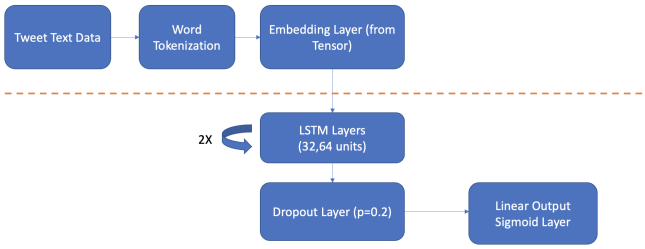

Figure 1. Data Processing for the Neural Network

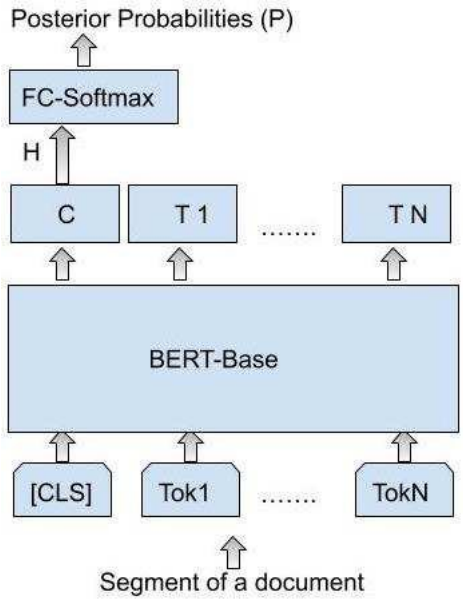

Figure 2. BERT model structure

able to learn information by navigating the tokens and extrapolate the context by looking to the left and right tokens next to the currently observed one.

We used the model as presented in the state of the art (from Google research in [3]) composed of 12 transformer layers, the model has almost 110 millions parameters (a simple example of model structure is showed in Figure 2). The training has been performed on a sample of 3400 complaining/non-complaining tweets prepared and tokenized as described in the previous sections. Then we used our tweets dataset as test set on which make predictions. BERT's output is represented by the likelihood of the tweet to contain positive sentiments, so we normalized the results between [-1, 1] to get a more intuitive point of view, assuming that under 50% (under 0), a tweet is supposed to express negative emotions.

Finally, for the purpose of the network, the sentiment score has been attached on the singles hashtags present in each tweet, so we grouped the result by hashtag obtaining the average sentiment related to the usage of the individual ones.

4. Network

4.1. Network Preparation

Before constructing any network graph we created a cleaned, sampled data set of users and the hashtags associated with their tweets. We cleaned the hashtags using the same methodology that we used for the tweet text itself. Working with extracted hashtags is proven to be significantly better when using a state of the art cleaning approach[6]. An additional step that we included was the removal of hashtags related to Bitcoin itself, as that was the subject of the experiment as well as the removal of as many similar hashtags as possible from tweets. An example of the latter is "cryptocurrency, crypto and cryptocurrencies" for example.

4.2. Network creation

The rationale behind the creation of the network is the following: two hashtags are linked if they appear in the same tweet and the weight of a link is equal to the number of tweets where they appear together. To build our network we chose the list of the first 1000 most occurring hashtags. For each pair of hashtags present in the list, we count the number of tweets where they appear together. The average number of tweets where two hashtags appear together is 65.52. We decide to discard all the links that have a weight lower of 66, because those links are not very informative, since those links reveal a weak connection between two hashtags and they add just noise to our network. Doing so we discarded 86.4% of the links, but now we keep only the links that reveal a strong (and informative) connection between two hashtags. With this procedure, our network is composed of 948 nodes (the reduction w.r.t 1000 is due to the fact that some hashtags do not appear more than 65 times together with other hashtags that are present in our network) and 11154 indirect edges.

4.3. Analysis of the network

4.3.1 Regime of the network

The first thing we did was to plot the log-log distribution of the degrees of the nodes, and then compute the γ parameter of the Power Law, to assess some properties of the network. Particularly, we fit the distribution of the nodes that have a degree bigger than 10, and doing so we that $\gamma = 2.036$ confirming that we are in a small world regime. We can see that also because the most common hashtags are linked with almost every other hashtag in the network.

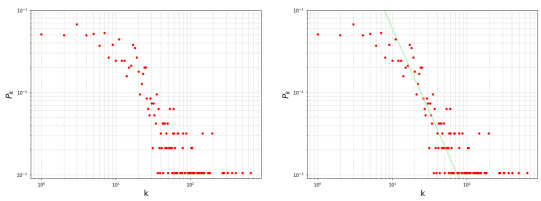

Figure 3. Degree distribution

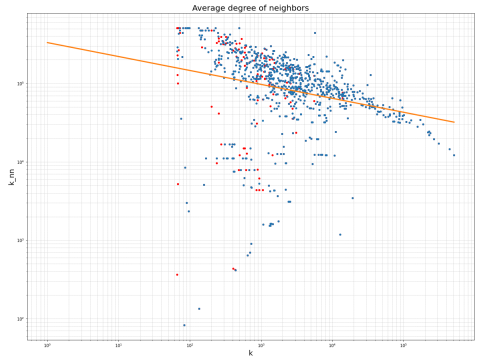

Figure 4. Assortativity plot

4.3.2 Assortativity

Then we proceed to calculate the assortativity of the network measured as the Pearson correlation between the degree of every pair of nodes that are linked. The network is slightly disassortative with a Pearson correlation of -0.24251. The disassortativity is due to the huge number of links, since the majority of the nodes have a lower degree compared to the hubs, we have that all those links influence the Pearson correlation. we computed also the $\mu = -0.17759$ that is the slope of the linear fit of the $\log - \log$ distribution of the average neighborhood degree over the degree of a node, which confirms the disassortativity we found.

4.3.3 Centrality

The third step was to calculate the centrality of the nodes, even if we already expect that the most central will coincide with the most occurring hashtags. The results for the first 30 nodes according to the PageRank score are in the (1). We compute the PageRank score, the HITS authority and HITS hub scores, and the eigenvalue centrality. In our case, the most informative are the HITS score because they emphasize how the authorities and the hubs often coincide. The PageRank and the Eigenvalue centrality are computed taking into account the weights of the links. The distribution of PageRank score vs Authority score, PageRank score vs Hub score and PageRank score vs Authority score vs Hub score were plotted in Figure 5, Figure 6 and Figure 7 respectively.

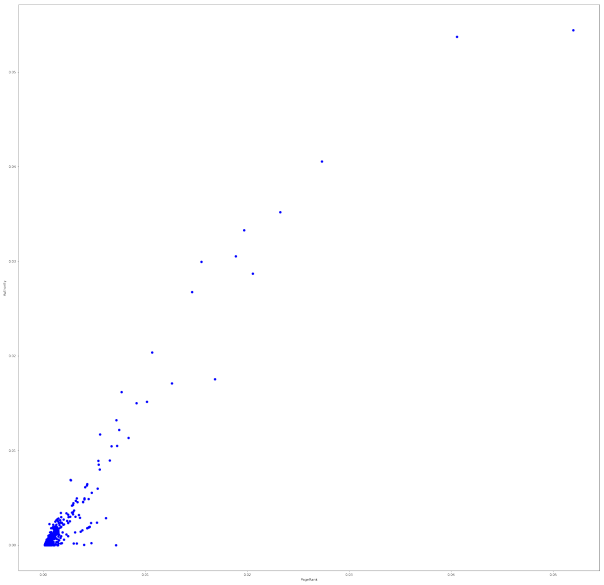

Figure 5. PageRank vs Authority

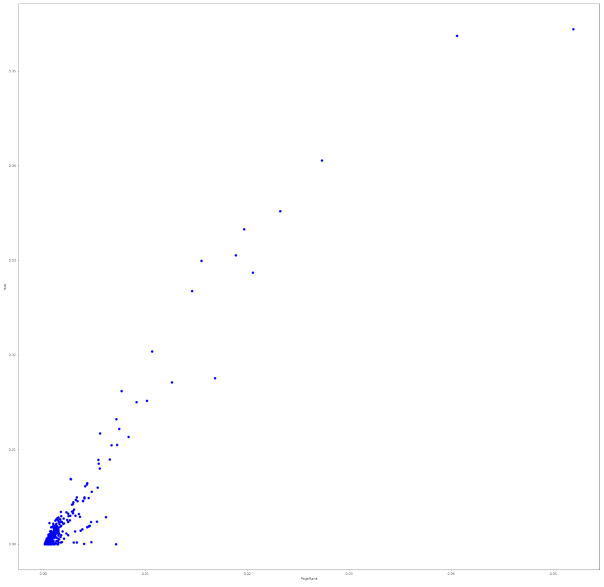

Figure 6. PageRank vs Hub

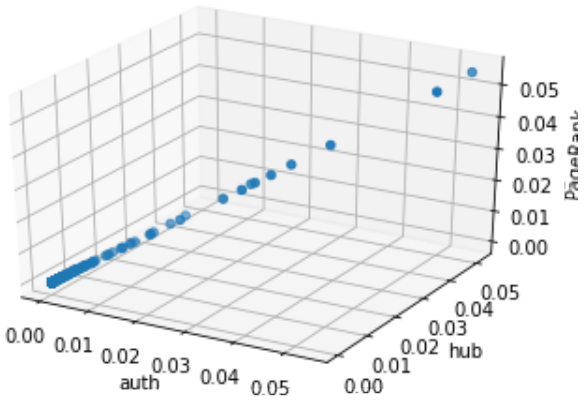

Figure 7. PageRank vs Authority vs Hub

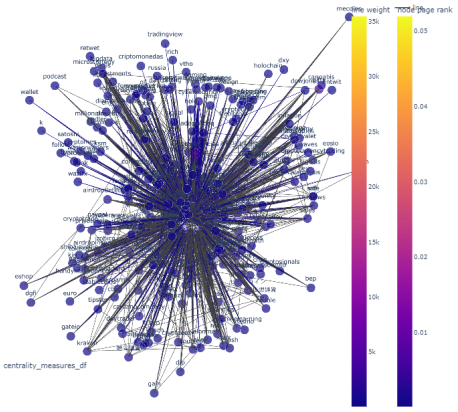

Figure 9. 3D Network

For the network in Figure 8, the colors of the nodes were changed according to the PageRank result shown in Table 1.

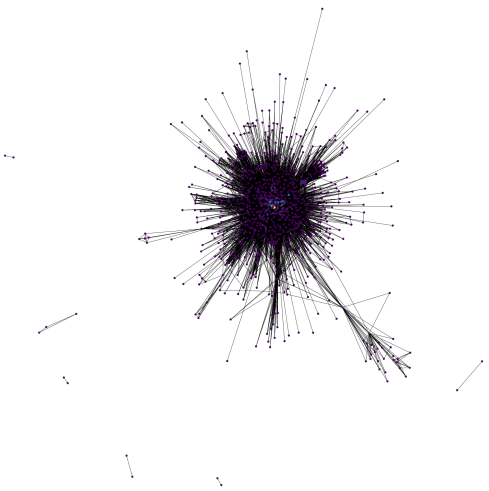

Figure 8. Network

In order to easier read the distribution of the network, a 3D version of the same network of Figure8 was plotted below in Figure 19. The colors of the nodes were changed according to the PageRank and the colors of the edges were changing according to the line weight.

4.3.4 Communities detection

The most, at first glance, obscure aspect of our network that needed to be inspected was the presence of communities. Having 948 hashtags we expected a lot of them to be apparently meaningless, or anyway not so related to our topic. This is due to meanly to phenomena:

1. The first is the presence of hashtags that are linked with the particular topic of the tweet where Bitcoin or more in general cryptocurrencies play a central role. Hence those hashtags reflect the presence of different topics somehow related with our main topic;

2. The second is just the fact that people use popular hashtags to reach more users even if those hashtags do not relate with the content of the tweet, and we can say that #bitcoin and #btc are popular too, so we can expect that a portion of our tweet is composed of tweets that are about bitcoin and have also hashtags that do not relate with it, and tweets that are not about cryptocurrencies but have bitcoin, or similar in them.

The first case is preferable and actually interesting to observe, while the second case is just noise from our point of view, but reflects a specific behavior of the users. To reduce the noise we should remove from our list the last elements, but in doing so we possibly could throw away informative links, so we decide not to do that. Knowing that our network is in a small world regime, and the network is densely connected, using community detectors based on the betweenness of the edges did not seem a good choice, because it would have struggled to try to separate the hubs. Not knowing what a community in our network could look like, given also that most of the hubs are connected with almost every node (the node with the highest degree is crypto with 902 connections, and we remark here that the network is composed of 948 nodes), the best thing we could do was to use a greedy modularity algorithm described in [1], that tries to maximize the quality of the partition, given that high modularity means good partition, it returns the partition with the higher modularity.

$$Q = \frac{1}{2m} \sum_{vm} \left[A_{vm} - \gamma \frac{k_v k_m}{2m} \right] \delta(c_v, c_m)$$

Where $A - vm$ is the vm component of the adjency matrix, $\frac{k_v k_m}{2m}$ is the probability that the vm was made randomly, and $\delta(c_v, c_m))$ is the function that returns 1 if v and m belongs to the same community. The algorithm starts by assuming that there are as many communities as nodes in the graph, then repeatedly merges two communities at time until the modularity stops increasing. This algorithm was developed to deal with graphs composed of hundreds of thousands of nodes, but as is shown later, it gave useful results also in our case. Modularity is governed by a hyperparameter γ called resolution. When $\gamma < 1$ partitions with a low number of communities get bigger modularity than partitions with a high number of communities, and the modularity is close to 1. When $\gamma > 1$ instead, partitions with a higher number of the community get a bigger modularity score than partitions with a low number of communities but, as in our case, the maximum modularity tends to decrease. Hence, also the community detector is governed by the resolution and will return the partition with bigger or lower numerosity accordingly. We tried different resolution values, as shown in (3), then we evaluated the partition using three metrics:

1. The modularity, that we already explained;
2. The coverage

$$C = \frac{1}{2m} \sum_{vm} A_{vm} \delta(c_v, c_m)$$

   that measures the number of connections intra-community over the total number of connections, and is useful to know how the links are embedded within the communities;

3. The performance

$$\frac{|\{(i,j) \in E, c_i = c_j\}| + |\{(i,j) \notin E, c_i \neq c_j\}|}{m(m-1)/2}$$

   that measures the total number of connections intra - community plus the total number of non-connections inter-community over the total number of potential links; in our case, it grows with the number of communities because the number of inter-community non-edges increases with the number of communities.

The results are shown in the (3) Usually, good partitions have modularity between 0.3 and 0.7, taking also into account the coverage, and the performance we choose partition number 5 because has the best balance between modularity, coverage, and performance. In order of numerosity we have:

1. the first, which contains most of the hashtags in the network, contains many words related to the cryptocurrency's world as #crypto, or #eth, but also has many words that taken alone cannot be reconducted to our topic. This is due to a mixture of the two effects described above, but mainly to the first because they appear together with hashtags #bitcoin and #btc and also with other "cryptocurrency's" words, so many times to belong to the same community of the latter.

Figure 10. First community

Figure 11. Second community

Figure 12. Third community

4.3.5 Centrality

2. In the second community, we have a lot of very common and generic hashtags that are not really related to the cryptocurrency's world, this is due mainly to the second effect: people use those hashtags to reach more people, so those hashtags appear together many times so that they form a community.

3. the third community contains very technical words, related to finance and technology like :#dowjones, #nasdaq, #python, #machinelearning, #ai and so on.

4. the fourth contains words related to covid. At first glance, it can seem normal that in the period we analyzed those hashtags appear, but again they both the

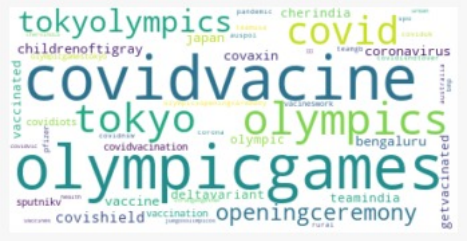

Figure 13. Fourth community

Figure 14. Fifth community

Figure 15. Sixth community

two effects described above play a role: with the pandemic all the hashtags related to covid became popular and many times they are used regardless of the topic of the tweet; but also the pandemic has influenced the crypto's world, so people discuss it.

5. the fifth is related to pollution, waste, and plastic, here we think that the first effect occurred most and they refer to a specific topic related to bitcoin;

6. the sixth is a community specialised in NFT give-away, we specified that because #NFT and other NFT related hashtags are present in the first community;

7. the seventh is about sport gambling;

8. finally the eighth is about real estate.

The other communities are about a really specific topic that for simplicity we omit because are also very small, but still, they are consistent. To notice that if we would have chosen the partition number 4, we would not have the communities about the environment, the real estate, and the third one. Even if we still would have the less relevant ones. On the other hand, if we would have picked the last partition

Figure 16. Seventh community

Figure 17. Eighth community

we would just have the first community divided into more communities that do not make much sense, and other communities like the third or the fifth would have words that do not actually relate to the topic. This means that our criterion of balancing the three measures works well. The last mention is that if we take the last partition, the first community contains only hashtags that are names of a lot of cryptocurrencies, except for that the partitioning is not satisfactory at all, and we are happy to choose the partition number 5. The different colors in Figure 18 were based on the group in partition number 5.

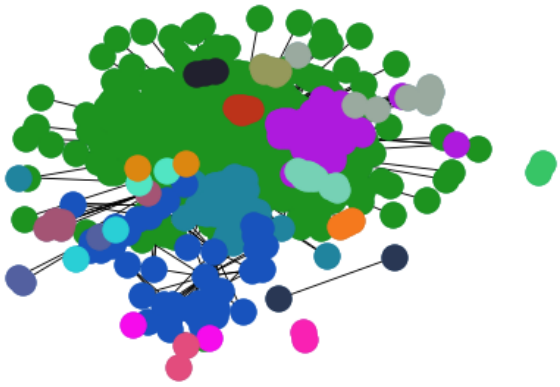

Figure 18. Partion 5 network

In order to easier to read the distribution of the network, a 3D version of the same network of Figure18 was plotted below in Figure 19. The colors of the nodes were changed according to the PageRank and the colors of the edges were changing according to the line weight.

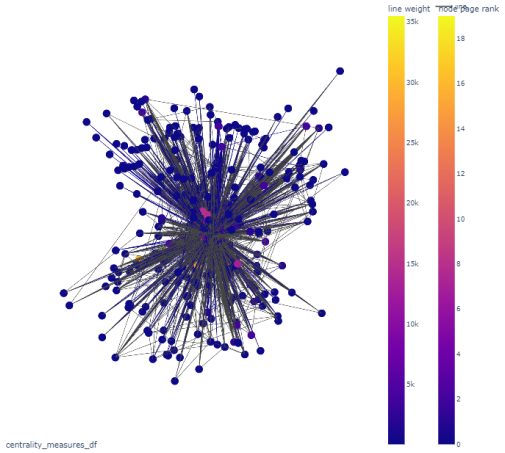

Figure 19. 3D Partion 5 network

4.4. Sentiment Analysis applied to the network

The final step of this work was to see how the sentiment analysis that has been done relates to the hashtags. The positivity score goes from -1 to 1, and we calculated it for each tweet in our data set. Then the positivity score has been calculated in this way

$$Ps_i = \frac{\sum_t Ps_{t:i \in t}}{n}$$
$$n = |\{t : i \in t\}|$$

where i is the hashtag and t is the tweet. After, we compute the numerical homophily of the hashtags with respect to the sentiment that corresponds to the Pearson correlation of the positivity score for each pair of nodes that are linked that is equal to 0.485659. So we can observe that between linked nodes there is a discrete amount of homophily tendency. The final step was to compute the score at the community level that is, for each community, simply the average of the scores of the hashtags that belong to it. The result are shown in (4). We can see how the communities that we described and interpreted have a positive score. A score has to be considered really positive if it goes beyond 0.7, otherwise, the positivity is not so intense. We can see that for example, the second community has a really high score, due to the fact all those popular hashtags are used somehow in a happy context. Yet, for the sixth community, the high score is motivated by the fact that tweets with hashtags belonging to that community, talk about giving away NFT for free, and usually, this is done for advertising a collection of NFTs or anyway a website, so they are happy tweets. The eighth is close to being neutral, and the reason may be due to the fact that hashtags related to bitcoin are there just to reach more people. For what concerns the first and the third communities, we can see that they have a high score, because those hashtags, especially in the third community are used by tech enthusiasts that believe in cryptos. Despite that, there are tweets that contain hashtags belonging to those communities, are neutral or negative. That explains why the score is not so high. There is also a small community of hashtags that has a negative score and contains the hashtags #fibonacci and #breakout, which are two words related (in our case) to the way to analyze the series of prices of stocks, or also cryptos by trying to interpret candle graphs, using some techniques that make use of the Fibonacci number. Those techniques are famous among the community of traders that try to predict the peaks of price historical series drawing lines that pass through some of the previous peaks of that series.

5. Conclusion

The analysis of social networks has recently emerged as an incredibly powerful and popular method for understanding the many complex relationships that exist in online communities. Through the gathering of a large dataset and performing a large thorough cleaning of it we were able to create a representative sample of the bitcoin crypto-currency community over many years. Through the use of a variety of neural network techniques we were able to quite accurately determine and classify the sentiment content of tweet text. The testing of a variety of sentiment analysis models gives us the assurance needed that we selected the state of the art for the dataset and given problem at hand. After generating the sentiment content within tweets we then were able to create a network graph of our data points. We realised by refining the network nodes and edges we could avoid the more common problems of long tail degree distributions for nodes. Thus by creating this refined network we were able to obtain statistical measures for the networks structure. We were able to split the hashtag network in many different and distinct communities. These include traders/vendors of bitcoin, investors through stock exchange and fintech oriented blockchain communtities, gambling through bitcoin(something that has become popular recently) and also interestingly a community which is talking about the potential environmental and polluting effect that mining bitcoin has. The sentiment analysis that we computed was then applied to the network in order to better understand what is the over bearing sentiment of each node community. Over the vast majority of bitcoin communities the sentiment is very positive. This is inline with what we can expect to see from overall bitcoin trend during the time period.

6. Future Work

There are a number of potential areas for future work that can be enacted on with this dataset and network. Given more time these are some of the things that we considered for potentially improving our work further.

1. More inter-time period comparability. It would be interesting to investigate how much the network graphs and the community compositions have changed over a certain smaller time section. We may be able to more closely follow peaks and troughs in trader sentiment during these smaller time periods.

2. Investigate the difference in sentiment between larger and smaller communities as well as investigating inter community sentiment to observe ratio of positivity and negativity within.

3. We also considered the possibility of segregating the dataset by sentiment prior to drawing out the network. Would be interesting to compare the strongest network communities in the sentiment specific datasets whilst comparing the hashtags that have been posted about.

References

[1] Cristopher Moore Aaron Clauset, M. E. J. Newman. Finding community structure in very large networks. *Phys. Rev. E 70, 066111*, 2004.

[2] Jose Camacho-Collados and Mohammad Taher Pilehvar. On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[4] Minlie Huang, Yujie Cao, and Chao Dong. Modeling rich contexts for sentiment classification with LSTM. *CoRR*, abs/1605.01478, 2016.

[5] Sotiris Kotsiantis, Dimitris Kanellopoulos, and P. Pintelas. Data preprocessing for supervised learning. *International Journal of Computer Science*, 1:111–117, 01 2006.

[6] Abhay Kumar, Nishant Jain, Suraj Tripathi, and Chirag Singh. From fully supervised to zero shot settings for twitter hashtag recommendation. 2019.

7. Appendix

Table 1. Page Rank Ordered Centrality Measures (descending)

Hashtag	Page Rank	HITS Authority	HITS Hub	Eigenvalue Centroid
crypto	0.051984	0.054434	0.054434	0.416759
cryptocurency	0.04057	0.053739	0.053739	0.411434
ethereum	0.027336	0.040573	0.040573	0.310639
eth	0.023245	0.035201	0.035201	0.269509
blockchain	0.020552	0.028696	0.028696	0.219703
bnb	0.019687	0.033284	0.033284	0.254824
binance	0.018864	0.030536	0.030536	0.233788
dogecoin	0.01683	0.017532	0.017532	0.134273
bsc	0.015508	0.029934	0.029934	0.229174
airdrop	0.014578	0.026742	0.026742	0.204737
doge	0.0126	0.017092	0.017092	0.130871
defi	0.010661	0.020374	0.020374	0.155982
xrp	0.010149	0.015155	0.015155	0.116028
cryptocurencies	0.009125	0.015008	0.015008	0.114902
nft	0.008354	0.011327	0.011327	0.086723
binancesmachain	0.007679	0.016194	0.016194	0.123985
altcoin	0.007426	0.012194	0.012194	0.09336
trading	0.007233	0.010493	0.010493	0.080338
cryptonews	0.00717	0.013191	0.013191	0.100996
covidvacine	0.007114	0.000006	0.000006	0.000045
airdrops	0.006676	0.010437	0.010437	0.079909
money	0.006531	0.008934	0.008934	0.068401
afiliatemarketing	0.006129	0.002847	0.002847	0.021839
pancakeswap	0.005556	0.011695	0.011695	0.089538
ada	0.005523	0.008004	0.008004	0.061283
altcoins	0.005412	0.0085	0.0085	0.065078
cryptotrading	0.005388	0.008914	0.008914	0.068246
forex	0.005327	0.005981	0.005981	0.04579
shopping	0.005244	0.002358	0.002358	0.018096
ltc	0.004746	0.00553	0.00553	0.042339

Table 2. HITS Ordered Centrality Measures (descending)

Hashtag	Page Rank	HITS Authority	HITS Hub	Eigenvalue Centroid
crypto	0.051984	0.054434	0.054434	0.416759
cryptocurency	0.04057	0.053739	0.053739	0.411434
ethereum	0.027336	0.040573	0.040573	0.310639
eth	0.023245	0.035201	0.035201	0.269509
bnb	0.019687	0.033284	0.033284	0.254824
binance	0.018864	0.030536	0.030536	0.233788
bsc	0.015508	0.029934	0.029934	0.229174
blockchain	0.020552	0.028696	0.028696	0.219703
airdrop	0.014578	0.026742	0.026742	0.204737
defi	0.010661	0.020374	0.020374	0.155982
dogecoin	0.01683	0.017532	0.017532	0.134273
doge	0.0126	0.017092	0.017092	0.130871
binancesmachain	0.007679	0.016194	0.016194	0.123985
xrp	0.010149	0.015155	0.015155	0.116028
cryptocurencies	0.009125	0.015008	0.015008	0.114902
cryptonews	0.00717	0.013191	0.013191	0.100996
altcoin	0.007426	0.012194	0.012194	0.09336
pancakeswap	0.005556	0.011695	0.011695	0.089538
nft	0.008354	0.011327	0.011327	0.086723
trading	0.007233	0.010493	0.010493	0.080338
airdrops	0.006676	0.010437	0.010437	0.079909
money	0.006531	0.008934	0.008934	0.068401
cryptotrading	0.005388	0.008914	0.008914	0.068246
altcoins	0.005412	0.0085	0.0085	0.065078
ada	0.005523	0.008004	0.008004	0.061283
yieldfarming	0.002668	0.006897	0.006897	0.052806
bounty	0.002701	0.006839	0.006839	0.052361
safemon	0.004316	0.006452	0.006452	0.049399
ripple	0.004283	0.006298	0.006298	0.04822
cardano	0.004113	0.006116	0.006116	0.046823

Table 3. Measures of the quality of the partitions

Partition Number	Modularity	Coverage	Performance	Number of Communities	Gamma
0	0.950117	0.998297	0.108575	8	0.05
1	0.900657	0.998297	0.108575	8	0.1
2	0.858627	0.970594	0.265464	11	0.15
3	0.822274	0.970504	0.267275	11	0.2
4	0.786537	0.969249	0.288936	11	0.25
5	0.621336	0.920836	0.459867	20	0.5
6	0.492045	0.829926	0.597735	26	0.75
7	0.40296	0.606599	0.800968	28	1
8	0.368054	0.576295	0.851579	23	1.25
9	0.345201	0.527255	0.892432	25	1.5
10	0.315592	0.509055	0.908695	25	1.75

Table 4. Network Communities Positivity Scores

Community Number	Positivity Score
1	0.669048321128231
2	0.832040147648917
3	0.674539813919672
4	0.573164706428845
5	0.506817817687988
6	0.848560783598158
7	0.750524946621486
8	0.328166484832763
9	0.499690604209899
10	0.733269850413004
11	0.051632920900981
12	0.590744654337565
13	-0.259529501199722
14	0.284090995788574
15	0.636531054973602
16	0.728054702281951
17	0.763987600803375
18	0.480783879756927
19	0.392869055271148
20	0.758999288082122