Prediction of Bitcoin prices using Sentiment Analysis and Topic Modelling from Social Media Forums

XIONG HUI

SONG BINGHENG

TEO WENLIN

GUI LIN

WANG SIXIANG

A REPORT SUBMITTED

FOR THE PRACTICAL LANGUAGE PROCESSING

PRACTICE MODULE

INSTITUE OF SYSTEMS SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2021

Content

1 EXECUTIVE SUMMARY

Bitcoin is a decentralized digital currency implemented by open-source software on a peer-to-peer network, which first came into use in 2009. Bitcoins are created through the process of mining and are used for various (mostly anonymous) transactions and as a form of investment. There are currently more than 100 million bitcoin wallets with value on the 2 biggest bitcoin wallet providers, Coinbase and Blockchain.com.

Previous research has shown that the fluctuation in bitcoin prices have some correlation with various factors, one of which is the value placed on it by end users. Researchers have used several methods to determine whether and the extent to which user sentiment can explain the variation in bitcoin prices.

In this project, we built a text-processing fine-tuning BERT model using comments on 2 bitcoin forums to predict bitcoin prices for the next day, over a period of 4 years from 2017 to 2020. We also extracted the main topics discussed in these 2 bitcoin forums using topic modelling methods. The results of our models are presented on an interactive website.

2 PROBLEM DESCRIPTION

2.1 Current Situation

Bitcoin may be considered to be an uncertain or risky asset because its value tends to fluctuate more than traditional currencies. This fluctuation is generally agreed to be due to price speculation, as other factors such as interest rates and government regulation are absent. Broken down further, the perception of the intrinsic value of bitcoin change (i) with global geopolitical events and government's attitudes towards the currency, (ii) with the expansion of its use cases, including in various types of transactions or as stored value investments, and (iii) with the perception of the security of such currencies.

Accordingly, researchers have attempted to find relationships between bitcoin price fluctuations and sentiment of bitcoin users, using sentiment analysis models. Kaminski and Gloor (2014) analyzed Twitter data over 104 days and compared this to the fluctuation in bitcoin prices, using a small lexicon of 15 words describing the polarity of financial sentiment. They found based on this dataset that the Twitter sentiment mirrors but does not predict bitcoin prices. Karalevicius et al (2017) measured positive and negative sentiments on news articles using a financial sentiment dictionary and found an interaction between media sentiment mined from bitcoin-related news portals and Bitcoin prices. Linardatos and Kotsiantis (2018) used several deep learning models, using Google trends data and bitcoin-related tweets over 2 years as input

to predict bitcoin price for the following day, with error rates between 0.99 and 2.66%.

2.2 Project Objectives

The objective of our project is to build a deep learning model to predict the price of bitcoin using sentiment extracted from 2 popular bitcoin forums.

As the perceptions of bitcoin uses change over time, it is also worthwhile to ascertain the main issues discussed by bitcoin users. To this end, we carried out topic modelling to find the main topics discussed by bitcoin users in the last 4 years, as a whole as well as segmented by year.

3 TOOLS AND TECHNIQUES

3.1 Dataset

3.1.1 Forum Posts

Data quality is the key to the success of NLP work, only typical text data about bitcoin will be helpful to find the insights of relationships between discussion and bitcoin price. In order to achieve this goal, we set up 3 criteria for selecting data source:

i. Established long enough

We needed to make sure we have adequate posts in the past 4 years, which can cover at least a whole cycle of price fluctuation.

ii. Health and stability

This means the target forums should not be governed by commercial parties, to avoid a potential conflicts of interest which may affect the topics people comment about.

iii. A friendly and active atmosphere

People are more willing to talk freely when they feel safe. Posts in a friendly forum will help to get closer to the core secrets of bitcoin in this NLP task.

With these considerations, over 10 Bitcoin forums were carefully examined, every perspective was scored from 1-10 (worst to best) and the Ranks were calculated.

Table 1. Comparison of different bitcoin forums

No.	Source Name	Duration	Commercial	Atmosphere	Score	Rank
1	All Crypto Talk	4	6	5	15	5
2	Bitcoin Forum	5	5	3	13	8

No.	Source Name	Duration	Commercial	Atmosphere	Score	Rank
3	Bitcoin Garden Forum	3	6	5	14	6
4	Bitcoin Stack Exchange	4	6	7	17	3
5	BitcoinTalk	8	7	8	23	2
6	CryptoCompare	4	4	6	14	6
7	Cryptocurrency Talk	3	5	4	12	9
8	CryptoInTalk	5	6	5	16	4
9	Jackobian Forums	3	5	3	11	10
10	Reddit r/BitcoinMarkets	7	9	8	24	1

After ranking, we chose to obtain post data from Reddit as it is more like a whiteboard that people can talk freely on. For comparison testing, we also chose to obtain data from Bitcoin-talk, an old brand bitcoin forum with lots of fans running over 10 years, people on this forum are all true lovers of bitcoin so their voices are valuable for the tasks.

In design of the data down loader, we choose to use selenium and request to visit the pages and xpath / beautifulsoup4 to extract the text of interest (forum headers and comments/posts). Each row (post) in our extracted data consists of 5 columns: date, url, username, header and text.

After web crawling, we obtained 840,432 posts from Reddit ranging from Jun 2016 to Mar 2021 and 927,547 posts from Bitcoin-talk ranging from Apr 2016 to Mar 2021.

All posts were cleaned and tagged with unified labels and stored in csv files for sharing among team members, for ease of loading through pandas.

3.1.2 Bitcoin Price

Bitcoin price data was obtained from coin-desk forum (https://www.coindesk.com/price/bitcoin) from 2014-09-17 to 2021-01-26. The price data consists of date, open, high, low, close, adjusted close and volume columns. All the price units are in US dollars.

Fig 1. Bitcoin price during 2017-2021

3.1.3 Google trends data

Google trends data was obtained for the search term 'bitcoin' over the period of 1 January 2017 to 31 March 2021, worldwide, with daily granularity, as shown in Fig 2 below:

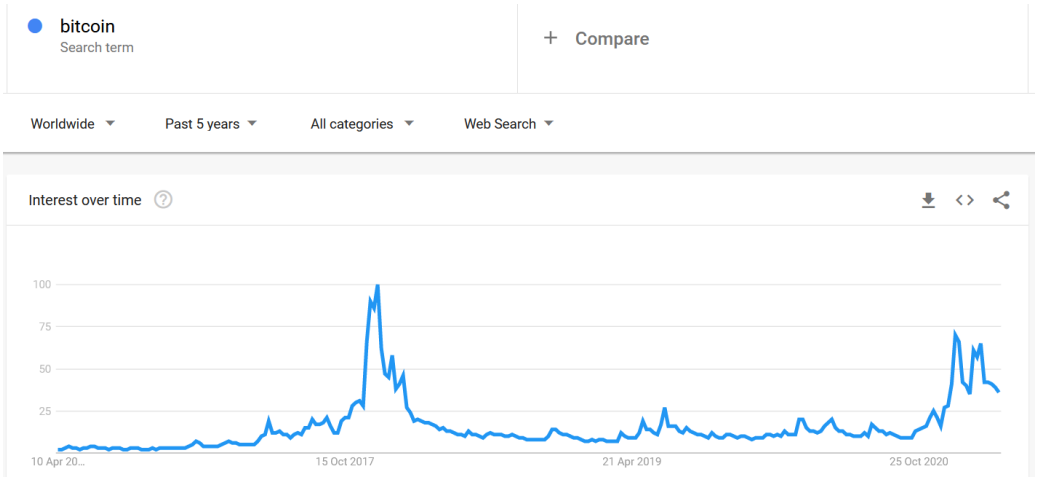

Fig 2. Google trends data on 'bitcoin'

3.2 Fine-tuning BERT Model

Transfer learning is a technique where a deep learning model trained on a large dataset is used to perform similar tasks on another dataset. Such a deep learning model is called a pre-trained model.

There are two approaches to use the pretrained model. One is called feature-based approach which creates task-specific architecture. The feature-based approach uses the pre-trained model as a feature extractor, and the weights in the pre-trained model are not updated during training. The structure is shown in Fig 3:

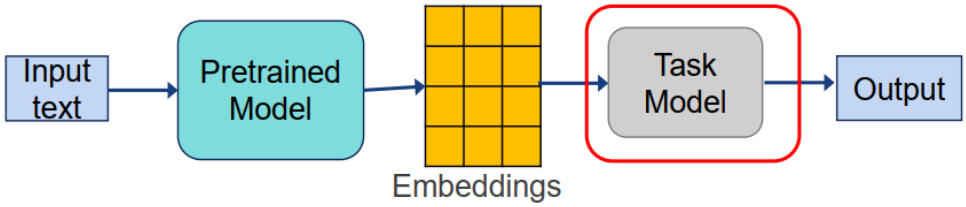

Fig 3. Feature-based approach from pre-trained model

Another is called fine-tuning approach. It introduces minimal task-specific parameters and is trained on the downstream task by fine-tuning all pre-trained parameters. The structure is shown below:

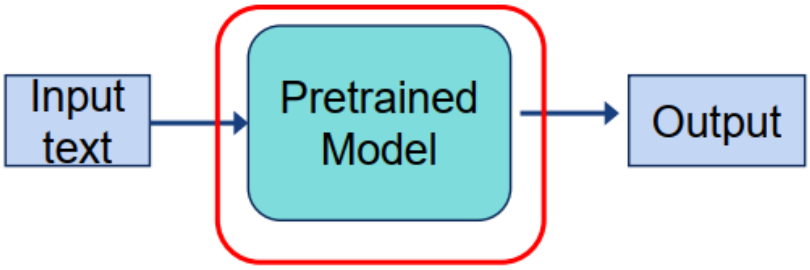

Fig 4. Fine-tuning approach from pre-trained model

The transformer concept was created by Google in the paper "Attention is All You Need" which made a breakthrough in traditional recurrent based NLP models. It does not process an input sequence token by token; rather it takes the entire sequence as input in one go which is a big improvement over RNN based models because now the model can be accelerated by the GPUs. Transformers allow us to provide a huge amount of unlabeled text data to train a transformer-based model to carry out tasks such as text classification, text generation, etc.

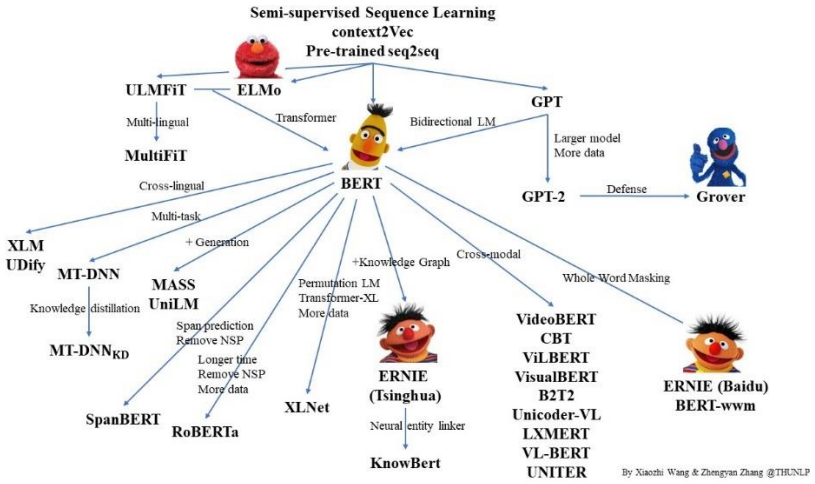

Fig 5. Sentiment analysis using BERT

BERT stands for bidirectional encoder representations from transformers, which takes in a huge number of parameters ranging from 100 million to over 300 million. Using a pre-trained BERT model that was trained on a huge dataset as a starting point enables the extraction of high-quality language features. Fine-tuning the pre-trained BERT model on a specific task with our own data can produce state of the art predictions.

Based on the above idea, we utilized fine-tuning BERT to do sentiment analysis and simulate the last layer operation to generate a sentiment score for each forum post.

We utilized the pre-trained classification model called "BERT-base-uncased" (text without casing) with an added linear layer for comment classification. To train the model for sentiment scoring, we used a financial tweets sentiment dataset obtained from Kaggle as training dataset. We excluded the neutral comments, and relabeled then with sentiment polarity (0 or 1).

The model has 12 encoder layers, 768 hidden units, 12 attention heads, and 110M total parameters. The detailed structure is shown in Fig 6.

```
The BERT model has 201 different named parameters.

==== Embedding Layer ====

bert.embeddings.word_embeddings.weight                  (30522, 768)
bert.embeddings.position_embeddings.weight                (512, 768)
bert.embeddings.token_type_embeddings.weight                (2, 768)
bert.embeddings.LayerNorm.weight                              (768,)
bert.embeddings.LayerNorm.bias                                (768,)

==== First Transformer ====

bert.encoder.layer.0.attention.self.query.weight          (768, 768)
bert.encoder.layer.0.attention.self.query.bias                (768,)
bert.encoder.layer.0.attention.self.key.weight            (768, 768)
bert.encoder.layer.0.attention.self.key.bias                  (768,)
bert.encoder.layer.0.attention.self.value.weight          (768, 768)
bert.encoder.layer.0.attention.self.value.bias                (768,)
bert.encoder.layer.0.attention.output.dense.weight        (768, 768)
bert.encoder.layer.0.attention.output.dense.bias              (768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight        (768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias          (768,)
bert.encoder.layer.0.intermediate.dense.weight           (3072, 768)
bert.encoder.layer.0.intermediate.dense.bias                 (3072,)
bert.encoder.layer.0.output.dense.weight                 (768, 3072)
bert.encoder.layer.0.output.dense.bias                        (768,)
bert.encoder.layer.0.output.LayerNorm.weight                  (768,)
bert.encoder.layer.0.output.LayerNorm.bias                    (768,)

==== Output Layer ====

bert.pooler.dense.weight                                  (768, 768)
bert.pooler.dense.bias                                        (768,)
classifier.weight                                           (2, 768)
classifier.bias                                                 (2,)
```

Fig 6. BERT model structure

To feed our text to BERT, it must be split into tokens, and then these tokens must be mapped to their index in the tokenizer vocabulary. The tokenization must be performed by the tokenizer included with BERT.

To make best use of the corpus, we first made some data preprocessing to get the max sentence length comments. However, the result is 4489 which is not practical and Bert allowed max length is 512. After some consideration and combined the reality, we set the max length to 64.

Besides these, BERT also requires: Mask IDs (also known as "attention mask") to distinguish tokens and paddings in a sequence; Positional Embeddings to get token positions in the sequence (automatically created as absolute positional embeddings in Transformers). Text reformatting is shown in Fig 7.

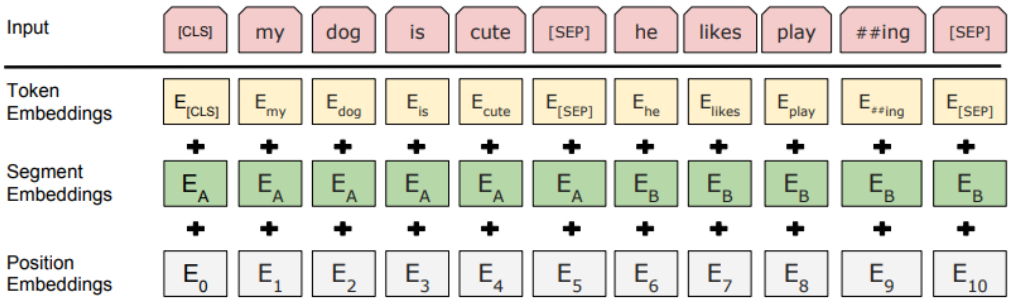

Fig 7. Text reformatting

The overall workflow is shown in Fig 8.

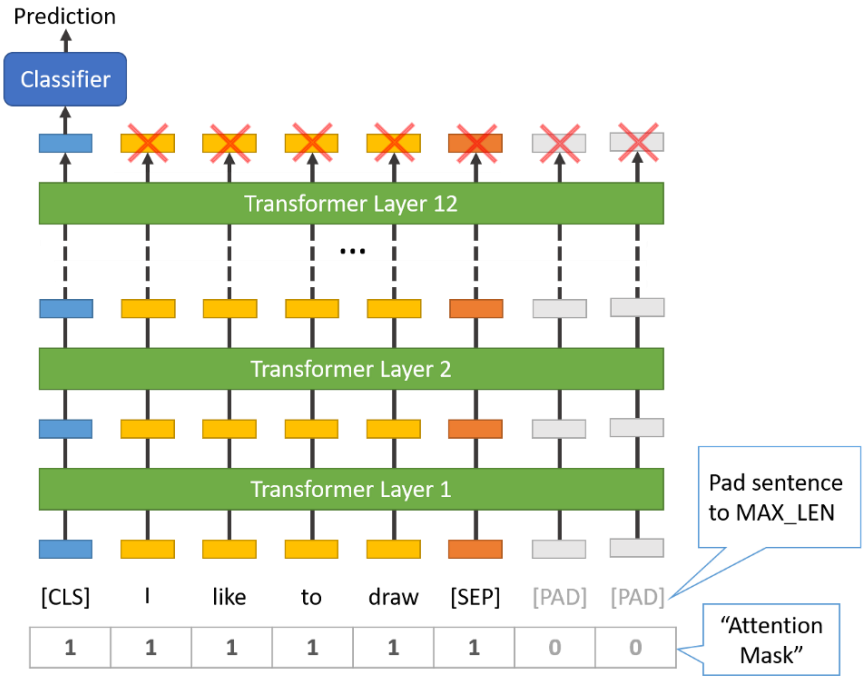

Fig 8. Overall workflow

In the training phase, we only utilized the classification labels, however, what we really want is the sentiment score. Thus, we generated our own score function based on the last classification function, using the code shown in Fig 9.

```
def softmax(x):
    e_x = np.exp(x - np.max(x))
    return e_x / e_x.sum(axis=0) # only difference

i = 0
score = []
for p in predictions:
    for ind in p:
        score.append(softmax(ind))

def score_sentiment(score):
    sentiment_score = []
    for s in score:
        if(s[0]>s[1]):
            sentiment_score.append(-s[0])
        else:
            sentiment_score.append(s[1])
    return sentiment_score
```

Fig 9. Customized score function

Instead of simply making a max comparison, we first passed the logits outputs to a soft-max function to get a pair of scores (positive and negative). Where the negative label score was larger than the positive label score we used the negative number as the score, and vice versa.

3.3 LSTM model for price prediction

The second model in task 1 is to utilize closing price of Bitcoin, using google trends data and custom features based on sentiment analysis of social media comments on Bitcoin. The custom features based on sentiment analysis including the sentiment score generated from fine tuning BERT model and the number of comments which we named "Activeness".

Based on the above-mentioned features, we created a machine learning model (LSTM model based on pytorch) for the prediction of the future price of Bitcoin. The model structure and parameters size are shown in Fig 10.

```
BitcoinPrediction(
  (lstm1): LSTM(4, 32, batch_first=True)
  (lstm2): LSTM(32, 64, batch_first=True)
  (dense): Linear(in_features=64, out_features=512, bias=True)
  (fc): Linear(in_features=512, out_features=1, bias=True)
)
12
torch.Size([128, 4])
torch.Size([128, 32])
torch.Size([128])
torch.Size([128])
torch.Size([256, 32])
torch.Size([256, 64])
torch.Size([256])
torch.Size([256])
torch.Size([512, 64])
torch.Size([512])
torch.Size([1, 512])
torch.Size([1])
```

Fig 10. LSTM Model structure

3.4 Topic Models

Topics hidden in the posts are also crucial to understand the thought process of bitcoin traders as their concerns about bitcoin can affect their trading behaviour and even the price of bitcoin. On the other hand, different topics can be signs of price movement, reveal the factors behind the price fluctuation and offer hints in the observation to the market.

As an art that can extract meaning, from words to sentences to paragraphs to documents, topic modelling is capable of distinguishing and extracting the topics across a collection of documents. The basic hypothesis is that each post has a mixture of different topics and each topic comprises a collection of representative words. LSA, PLSA, LDA & lda2Vec are the common methods for topic modelling tasks. In this project, the above techniques were fully discussed among the team.

As a classic algorithm, Latent Semantic Analysis (LSA) highly relies on the similarity among documents and similarity among words calculated by TF-IDF scores and truncated Singular Value Decomposition, however, the result could be too arbitrary and may lack interpretable embeddings.

Compared to LSA, pLSA (Probabilistic Latent Semantic Analysis) uses joint probabilistic methods instead of SVD to assign topics based on documents and word distributions. This is far better than pure LSA but is prone to overfitting especially in a case such as ours where the data size exceeds a million. Moreover, pLSA is not capable of assigning topics to new documents which are outside the scope of the model.

Latent Dirichlet Allocation (LDA) is the most prevalent model for topic modelling for a long time and is widely used in many topic modelling tasks. It has several advantages including:

ease of interpretability by humans and capable of expansion on new data sets. Furthermore, LDA models can be evaluated objectively through perplexity and coherence scores.

As the latest technique, lda2vec takes the benefits from deep learning, combining LDA and word2vec that jointly learns word, document, and topic vectors, generating "context" vectors for each word in the document. It not only learns word embeddings (and context vector embeddings) for words, it also simultaneously learns topic representations and document representations as well. However, lda2vec is hard to fine-tune and not as interpretable compared to LDA.

In this topic modelling task, we choose LDA as our main technique as shown in the workflow in Fig 11.

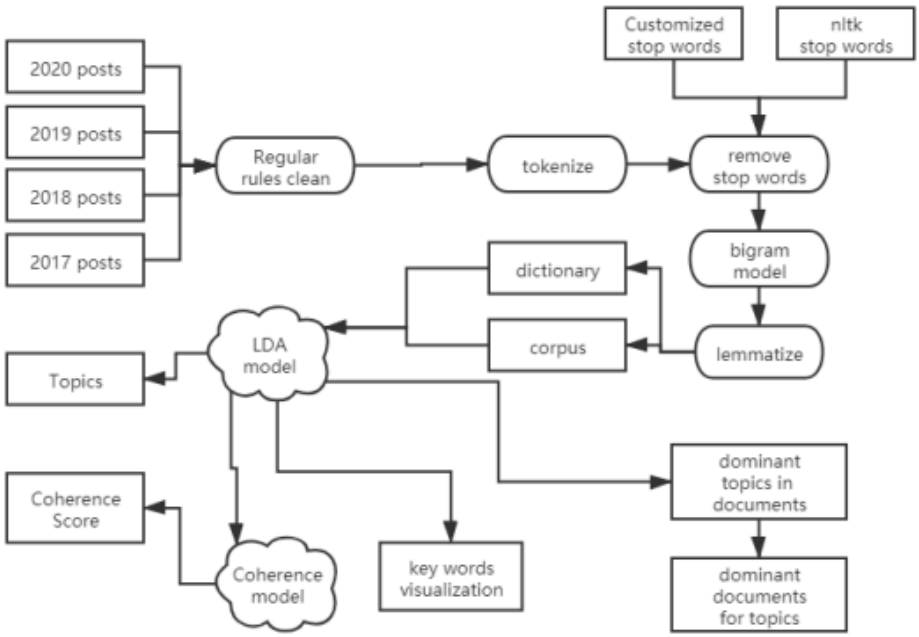

Fig 11. Topic modelling technique workflow

In the LDA analysis process, in order to find the prevalent topics for each of the 4 years, the whole data is divided into 4 parts and each part processed sequentially as follows:

i. Regular expression cleaning

Since all the posts are gathered from web pages, there are many HTML symbols mixed in it that are obstacles for the topic modelling task. To remove these meaningless terms, regular expression rules were designed based on typical posts.

ii. Tokenization

After cleaning, the gensim package was used for tokenization; specifically, the simple_process function from gensim.utils which is suitable for fast tokenization.

iii. Stopwords and punctuation removal

Besides the standard stopwords from nltk library, a customized stopwords list was generated though manual inspection and added during the topic modelling results. Words like “bitcoin”, “style”, ”font”, “valign” are added since they are not helpful in this task. Punctuation were also removed from the tokens.

iv. Bigram model building

Considering the possible of words combination, bigram language model is selected to be used in the modelling process and is built also by the gensim package.

v. Lemmatization

Spacy is most suitable for lemmatization work for this task and was used for word lemmatization.

vi. Dictionary & corpus

After lemmatization process, 2 main inputs, dictionary and corpus were built using gensim. Using Part-of-Speech tagging, we also built an alternate corpus and dictionary consisting only of nouns, based on prior research which shows some success in nouns-only topic modelling. During POS tagging, certain words which have special significance in the context of bitcoin trading were identified specifically as nouns (e.g. ‘short’, ‘long’, ‘ath’, ‘fud’).

vii. Topic model training

In order to find the best hyper-parameter for number of topics, an assessment on a typical data set with different numbers was trained and measured by coherence scores. The best number will be used in the training process of other data sets.

viii. Topic visualization

The pyLDAvis package is the best tool for topic visualization, the graph gives a good measurement for the success of the task, by displaying the distribution of topics in 2 dimensions using dimension reduction, where the distance between topics on the graph corresponds to their dissimilarity.

ix. Topic interpretation

Topic interpretation is a crucial work that can be only done manually. Based on a bunch of representative words for each topic, we have to consider and decide what topic these words are implying. Besides topic reading, it is also necessary to find out the topics hidden in a given document or the most related document to a specific topic.

4 SYSTEM DESIGN AND MODELS

4.1 SYSTEM ARCHITECTURE

Using the tools and techniques described above, every components are combined and all results displayed in an interactive website, with the architecture shown in Fig 12.

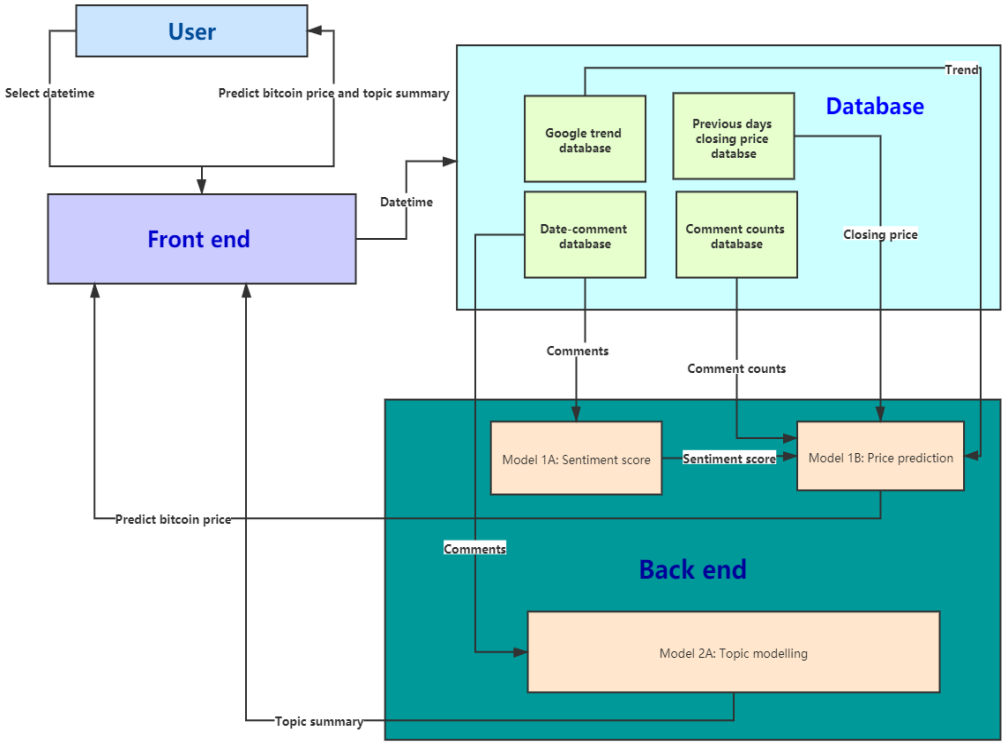

Fig 12. Overall System Architecture

The system is divided into four parts. They are user input, front end interface, database implementation and backend model prediction. Firstly, the user selects a datetime into front end. And then the input datetime transfer to the database to query data. There are four databases: Date comment database, Google trend database, Previous day closing price database and comment counts database. The system first gets the comments data through date comment database. And then transfer the comments data as the input passed into the sentiment score model and get sentiment score. Next, the system will get trends, previous day closing price data and comment counts through other three database and together with sentiment score as the input passed into price prediction model to get the predicted price. Additionally, the system uses the comments as input passed into the topic model to get the topic summary. Finally, the result of predict price and topic summary will transfer to front end as the output displayed to the user. This is our whole system architecture.

4.2 SYSTEM IMPLEMENTATION

The front-end interface of our system uses html, css, javascripts. We make use of bootstrap framework to produce wonderful pages. And the backend framework use python flask. And the database use MySQL. When the user opens our website home page, user can choose a predict date of bitcoin, with the home page shown in Fig 13.

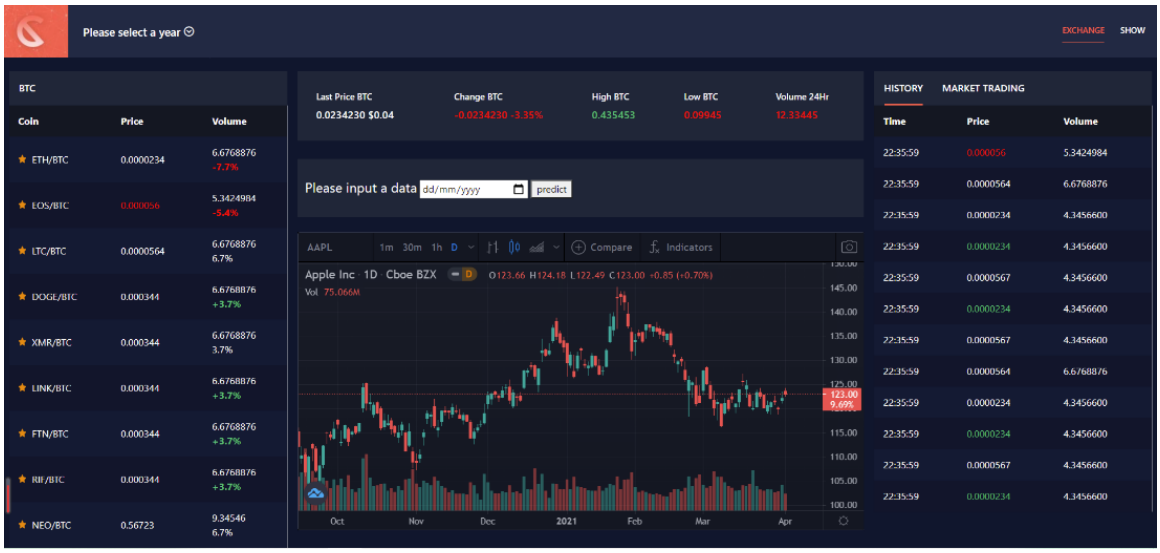

Fig 13. Home page

And then the system will give the result of the predict bitcoin price and bitcoin trend, which is our first model task, with the result interface shown in Fig 14.

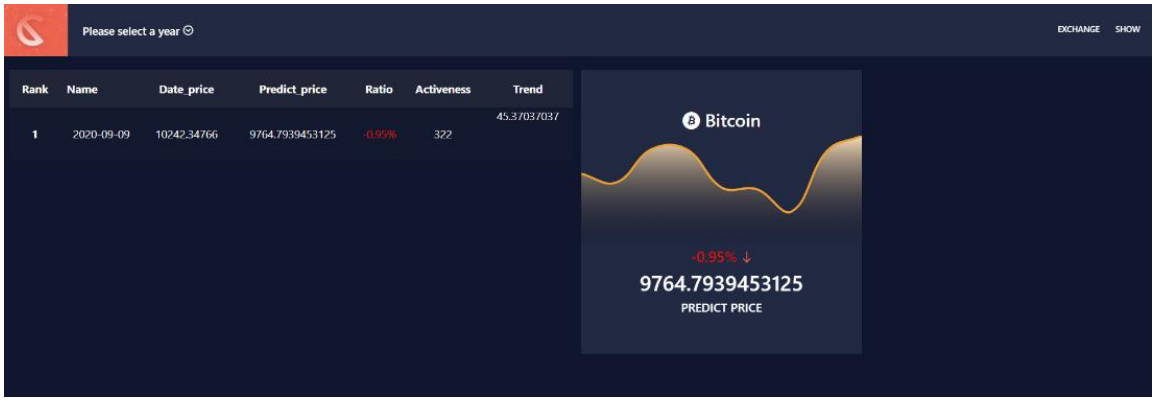

Fig 14. Result interface

Additionally, user can also view the topic keywords of each year (2017-2020), with the topic keywords shown in Fig 15.

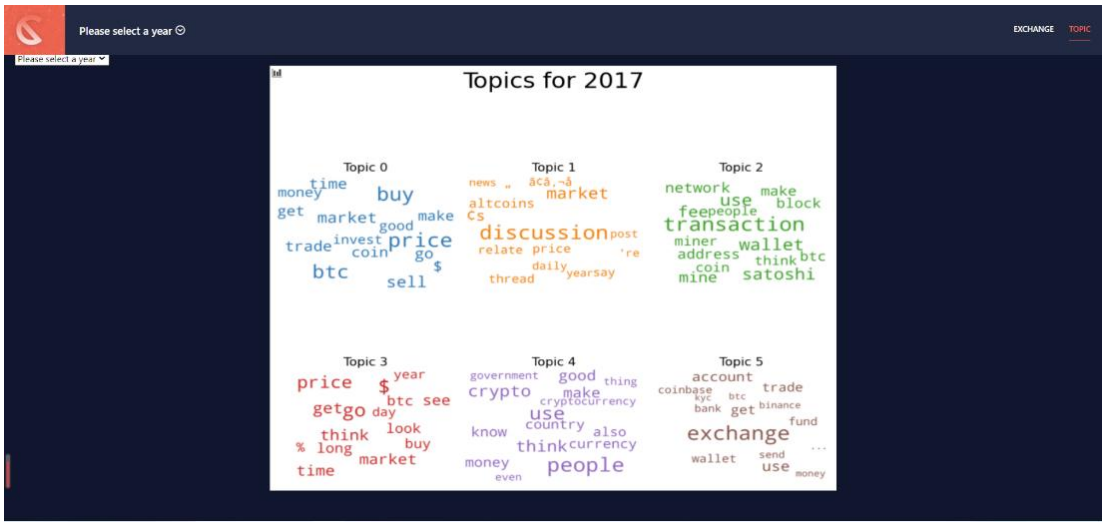

Fig 15. Topic keywords

Users can also browse the distribution of each topic in the year by selecting the year, with the topic distribution shown in Fig 16.

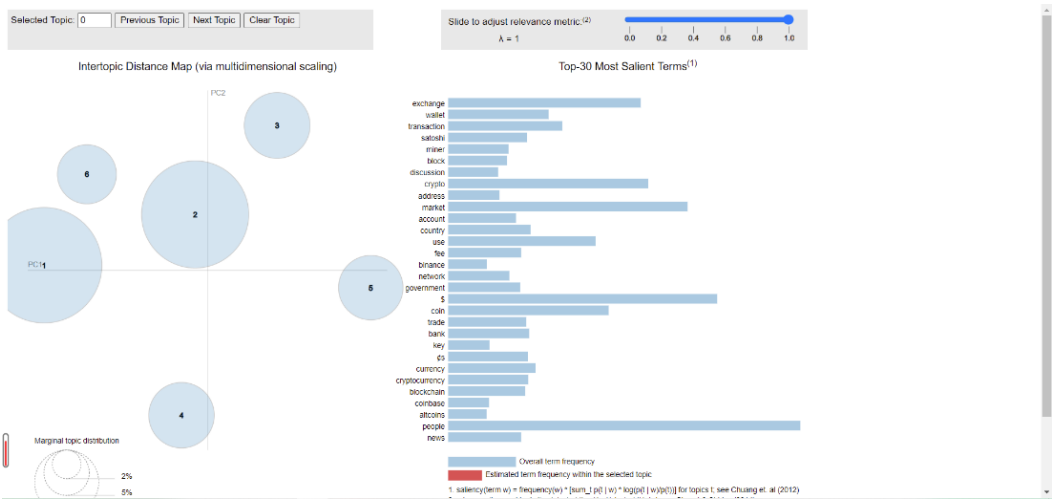

Fig 16. Topic distribution

4.3 SYSTEM PERFORMANCE

4.3.1 Fine-Tuning BERT Model

For the training dataset we accomplished 99% accuracy in two epochs and achieved 81.9% accuracy on the testing dataset, which achieved our target since all the data are random from social media.

The model accuracy on training dataset which consists of 11110 posts is shown in Figure 17.

```
======== Epoch 1 / 2 ========
Training...
  Batch    40  of    313.    Elapsed: 0:00:14.
  Batch    80  of    313.    Elapsed: 0:00:27.
  Batch   120  of    313.    Elapsed: 0:00:40.
  Batch   160  of    313.    Elapsed: 0:00:53.
  Batch   200  of    313.    Elapsed: 0:01:07.
  Batch   240  of    313.    Elapsed: 0:01:20.
  Batch   280  of    313.    Elapsed: 0:01:34.

  Average training loss: 0.20
  Training epcoh took: 0:01:45

Running Validation...
  Accuracy: 0.98
  Validation Loss: 0.06
  Validation took: 0:00:04

======== Epoch 2 / 2 ========
Training...
  Batch    40  of    313.    Elapsed: 0:00:13.
  Batch    80  of    313.    Elapsed: 0:00:27.
  Batch   120  of    313.    Elapsed: 0:00:40.
  Batch   160  of    313.    Elapsed: 0:00:54.
  Batch   200  of    313.    Elapsed: 0:01:07.
  Batch   240  of    313.    Elapsed: 0:01:21.
  Batch   280  of    313.    Elapsed: 0:01:34.

  Average training loss: 0.05
  Training epcoh took: 0:01:45

Running Validation...
  Accuracy: 0.99
  Validation Loss: 0.05
  Validation took: 0:00:04

Training complete!
Total training took 0:03:37 (h:mm:ss)
```

Fig 17. Training accuracy

The testing accuracy on testing dataset which consists of 1967 posts is show in Figure 18.

```
Overall accuracy for test set: 0.8189516129032258
```

Fig 18. Testing accuracy

4.3.2 LSTM model for price prediction

For the bitcoin price prediction model, we use the mean square error as loss function since it can regard as a regression problem and is used for predicting continuous values. We needed to do normalization (factor scaling) at first. Here we utilized MinMaxScaler in training dataset and saved the scalar for later testing dataset and price rescaled.

We achieved convergence at last, and the training loss is very small which was below 0.00037(Figure 19). The testing dataset loss behaved well as it was less than 0.050, which we thought was good.

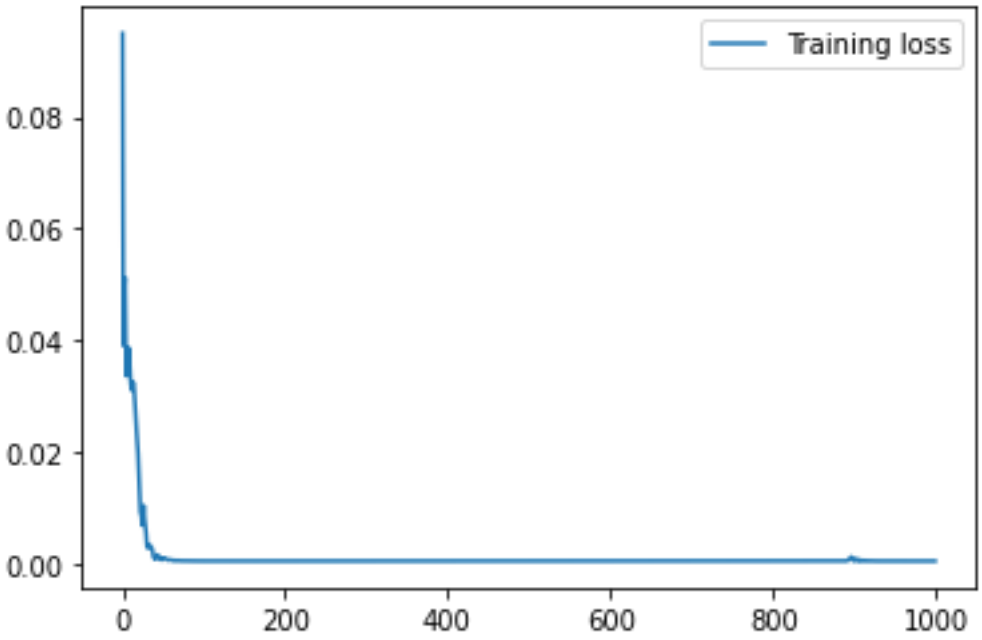

Fig 19. Training loss in LSTM

To do the price prediction and comparison we needed to revert results to raw scale, and we drew the predicted price and real price as in Figure 20.

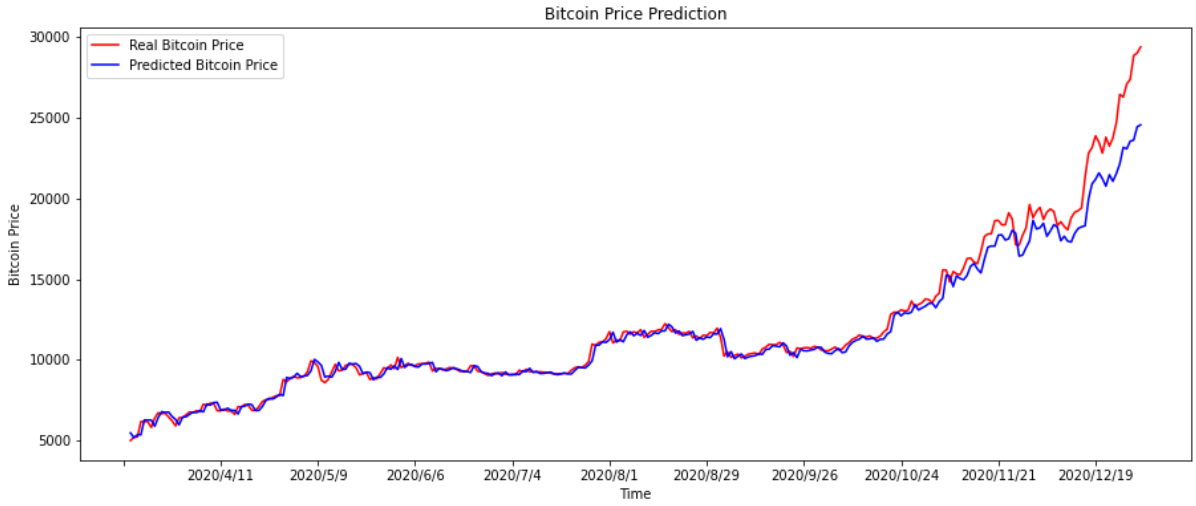

Fig 20. Predicted and real bitcoin price

We can see that the overall trends are synchronous and distance between predicted and real price is small.

4.3.3 Topic models

To find the optimal number for the topics, posts in 2020 are chosen to perform an iteration search from 2 to 18 topics as following.

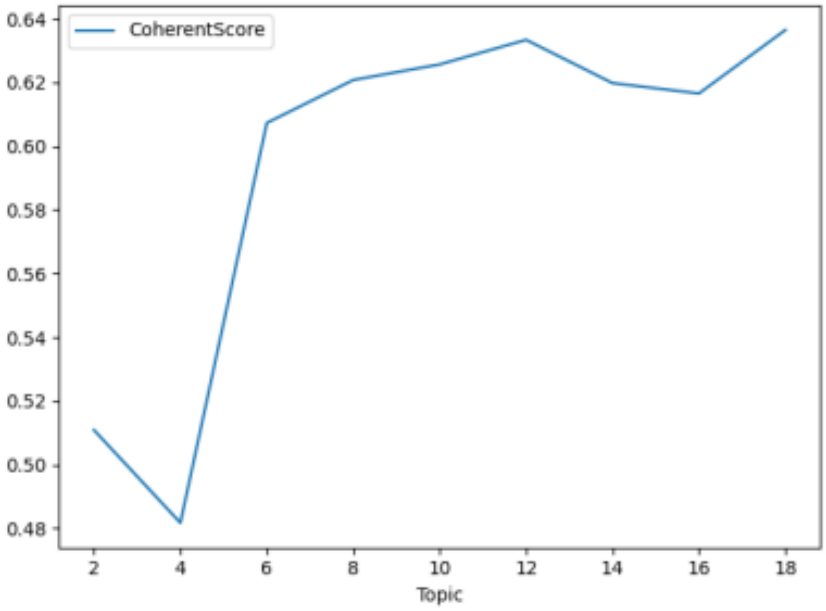

Fig 21. coherence scores for number of topics

The coherence score increases dramatically from 4 topics and stops growing upwards of 8, as too many topics may be distracting and time consuming to interpret. In this task, 6 is the most appropriate number.

To have an overview on all the topics in all data, a basic LDA model is built to get most prevailing topics from 2017 to 2021, most popular words are shown in figure 22.

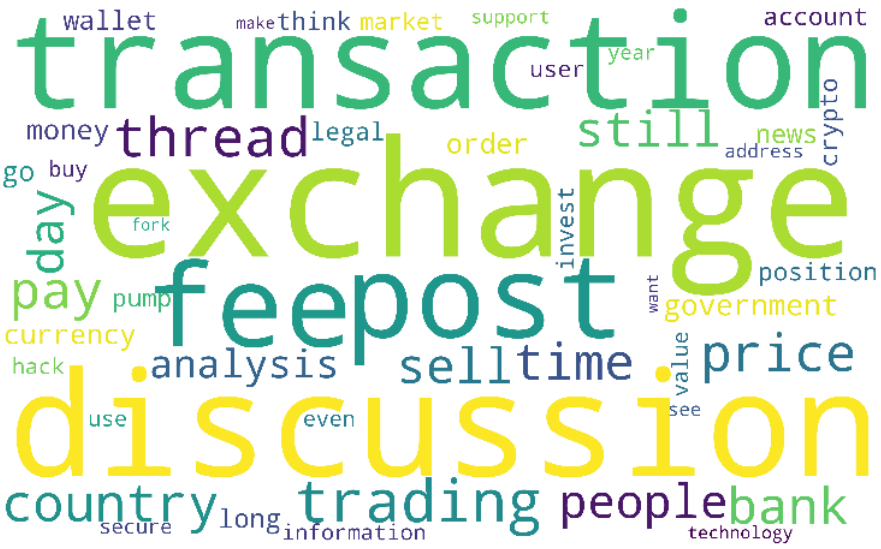

Fig 22. Most popular words in all posts

Specifically, 6 most popular topics are interpreted and summarized from all 20 topics. These topics could be considered as the most concerns for all bitcoin players in these 4 years period.

Table 2. Most popular topics during 2017-2020

ID	Topics/Typical posts	weight
1	**Bitcoin account/personal information security**	0.21
	Will my address get exposed if I'm trading bitcoin? How to secure that my account will not be hacked? Will exchange be closed?	
2	**Bitcoin legality/government policies**	0.16
	Will bitcoin become a legal currency in countries? Hope bitcoin could be accepted in Walmart.	
3	**Optimistic attitude towards Bitcoin**	0.13
	Price of BTC will raise on the long run. BTC will be the most successful investment in this century! There's still huge space to grow...	
4	**Transaction fee**	0.11
	Which exchange has the lowest fee rate？ Transaction fee is too high.	
5	**Encourage people to buy/successful cases**	0.09
	BTC can really make money, for my experience… I earned over 2k in just week…	
6	**Trading chances/skills/suggestions**	0.08
	This is the best change to buy in... Price will not last till the end of March... I bet that the price will go over 24.3k within 3 months…	

For deeper analysis on the topics in each year, other models are build based on the yearly data sets.

Figure 23 shows the visualization of topics over all 4 years using the pyLDAvis library. The diameter of each circle represents the number of documents having that topic. The distance between circles represents the extent to which the topics are differentiated from each other. The bar chart shows the most salient terms in all of the documents.

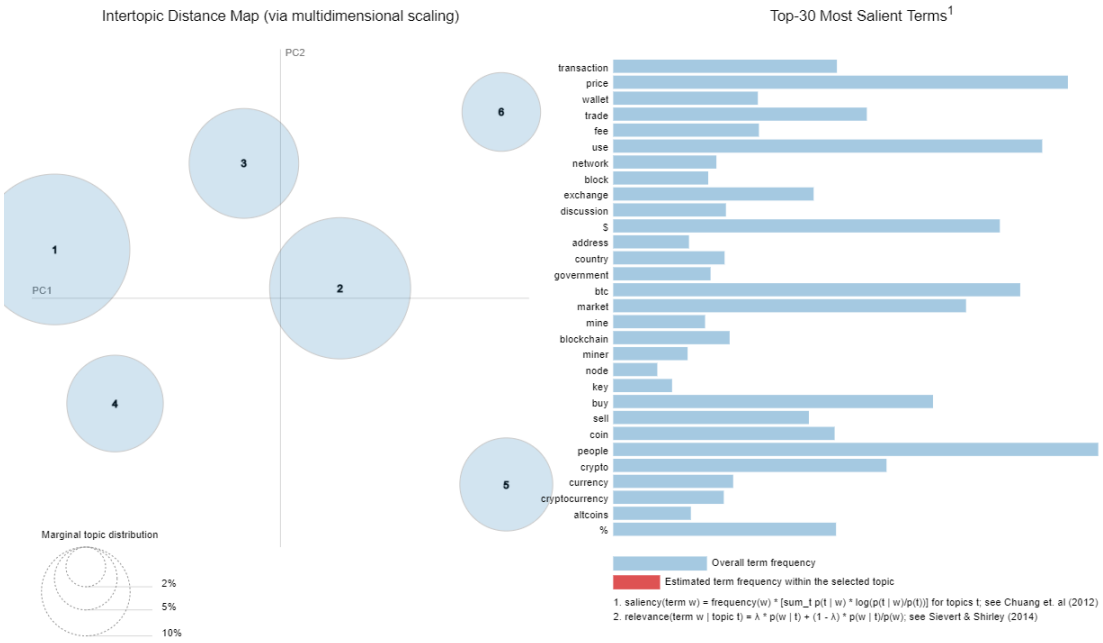

Fig 23. Visualization of topics and topic words

While not as informative as the pyLDAvis figures, word clouds provide an easy visual of the most salient words in each topic. Figures 24a and 24b are word clouds of the 6 topics, using all words and using nouns only, over all years.

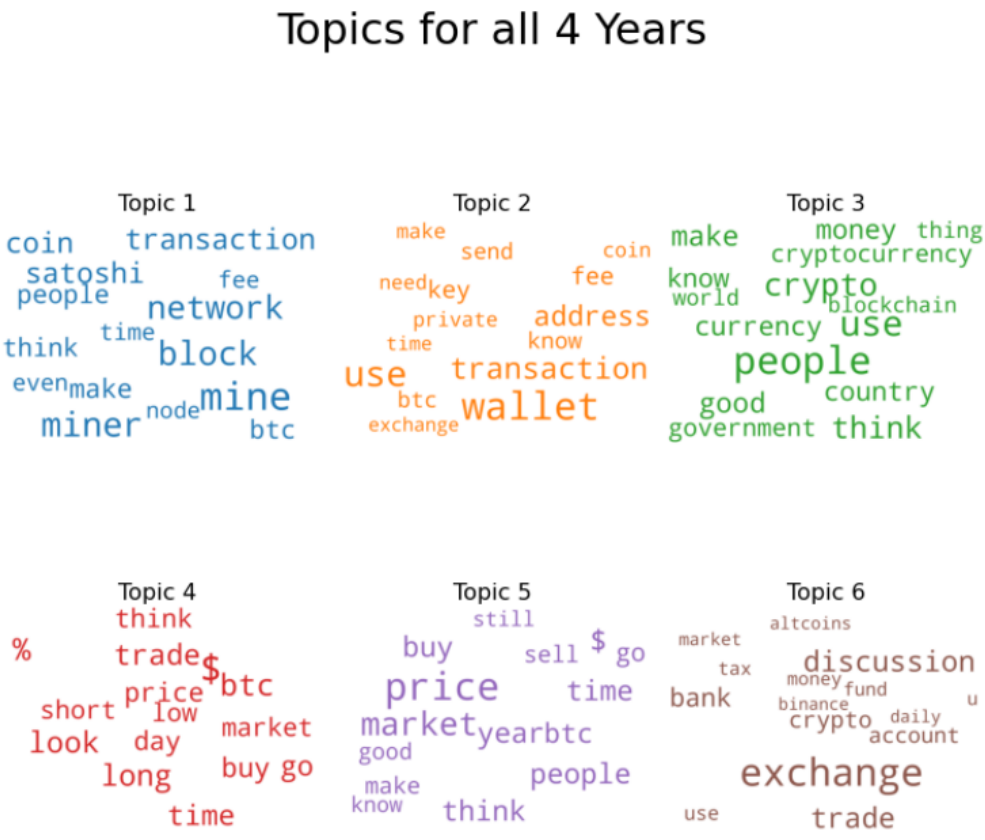

Fig 24a.

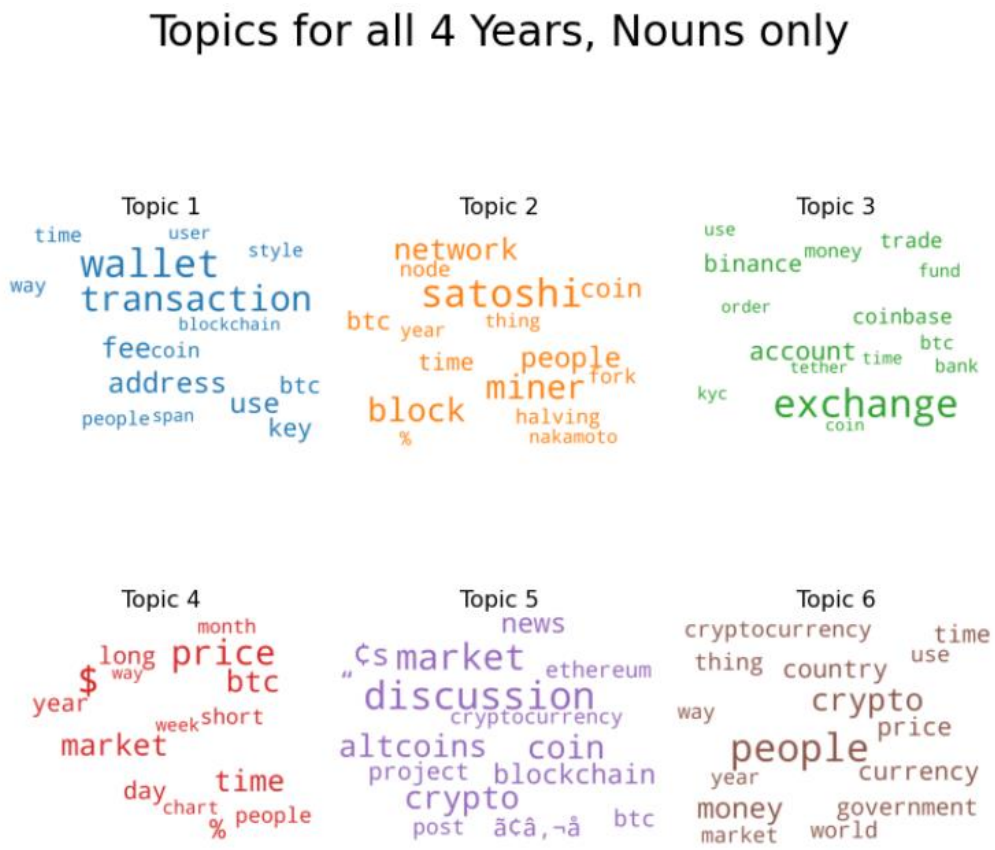

Fig 24b.

From the pyLDAvis charts and the word clouds in Figures 24a and 24b and also as mentioned above, one of the topics of concern is the legalization / regularization of bitcoin as a currency by countries (Topic 3 in 24a and Topic 6 in 24b). Another relates to the technical aspects of blockchain mechanisms on which bitcoin transactions are created and run (Topic 1 in 20a and 2 in 24b). The third is the alternative cypto currencies to bitcoin (Topic 6 in 24a and 5 in 24b), the fourth involves the security concerns related to bitcoin transactions (Topic 2 in 24a and Topic 1 in 24b), while the remainder are nuances on the general topic of trading prices and exchange platforms.

5 LIMITATIONS AND IMPROVEMENTS

5.1 Bitcoin price prediction

In this task, we can only predict the closing price for bitcoin, which is not enough for investors. From the application perspective, we only analyzed the related factors which may influence bitcoin price without revealing the relationship between bitcoin price and factors.

In the future, we can make use of each relevant factor as a feature with which to predict bitcoin price fluctuation like maximum and minimum price on certain days, which can provide more

information for investors.

5.2 Topic modelling

In this task, only LDA models were used in topic modelling process. Newer models such as lda2vec technique could be introduced to test the different results among models. We could have also experimented with other tagging mechanisms and utilized Name Entity Recognition to identify proper nouns, to increase the accuracy of the model.

6 CONCLUSION

Our team had a wonderful but also hard time working on this project, and definitely picked up useful skills along the way.

Model construction was a crucial part of the entire process, thanks to the shoulders of giants we could use transfer learning to make state of the art predictions. Our group now has a better understanding of how to apply word2vector, tokenization, position encoding and various deep learning models in NLP projects.

Building the system itself presented a whole new set of learning points. We got to apply practical knowledge of machine learning, as well as tap on our existing expertise in Python, JavaScript, and front-end frames like Vue.js and Node.js. Since our team is a cross-subject team (three of us from Intelligent systems and two of us from EBAC), working on the exercise together allowed everyone to learn technical skills from one another.

Overall, it was truly a multi-dimensional problem due to the separate models' design and complicated dataset. Deep learning on unstructured datasets (text) can make machines understand human language and help human beings dig deeper information which can make great treasures.

7 BIBLIOGRAPHY

Articles

Davis, Joshua (10 October 2011). *"The Crypto-Currency: Bitcoin and its mysterious inventor"*. The New Yorker.

Mitchell, Eddie (23 November 2020). "*How Many People Use Bitcoin in 2021?"*. Bitcoin Market Journal, https://www.bitcoinmarketjournal.com/how-many-people-use-bitcoin/, last accessed on 3 April 2021

Reiff, Nathan (16 June 2020). "*Why Bitcoin Has a Volatile Value*". Investopedia, https://www.investopedia.com/articles/investing/052014/why-bitcoins-value-so-volatile.asp, last accessed on 3 April 2021

Research Papers

Kaminski, J. and Gloor, P.A. (2014), "Nowcasting the bitcoin market with twitter signals", CoRR abs/1406.7577, available at: http://arxiv.org/abs/1406.7577

Karalevicius, V., Degrande, N. and De Weert, J. "Using sentiment analysis to predict inter day Bitcoin price movements". The Journal of Risk Finance Vol. 19 No. 1, 56-75,Emerald Publishing Limited, 2018

Loughran, T. and McDonald, B.(2011), "When is a liability not a liability? Textual analysis, dictionaries, and 10-ks", The Journal of Finance, Vol. 66 No. 1, pp. 35-65, available at: http://dx.doi.org/10.1111/j.1540-6261.2010.01625.

Linardatos, P. and Kotsiantis, S. "Bitcoin Price Prediction Combining Data and Text Mining" in Advances in Integrations of Intelligent Methods, Chapter 3, Springer, 2018

Vaswani, A., Shazeer, N., et al. (2017), "Attention is All you Need", ArXiv abs/1706.03762, available at https://arxiv.org/abs/1706.03762v5.

Martin, F. and Johnson, M. 2015. "More Efficient Topic Modelling Through a Noun Only Approach", Proceedings of Australasian Language Technology Association Workshop, pages 111-115.

Data Sources

1. Reddit bitcoin forum: https://www.reddit.com/r/Bitcoin
2. Bitcoin talk forum: https://bitcointalk.org/index.php
3. Financial tweets sentiment dataset: https://www.kaggle.com/vivekrathi055/sentiment-analysis-on-financial-tweets

Libraries

1. https://huggingface.co/transformers/model_doc/bert.html#bertforsequenceclassification
2. https://pytorch.org/tutorials/beginner/transformer_tutorial.html