The problem:

Extracting crypto related words from conversational text messages.

The solution:

Part 1:

The dataset provided with contains a corpus of crypto related words is limited. Increasing the number of words would be an extremely cumbersome task because it would require a manual updation of the list every time.

I’ve proposed a solution to significantly increase the ease of finding new crypto related words using an unsupervised approach.

The approach:

**Latent Dirichlet Allocation (LDA)** is a generative probabilistic model that assumes each topic is a mixture over an underlying set of words, and each document is a mixture of over a set of topic probabilities.

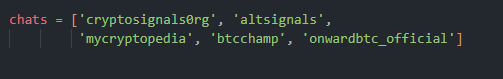
Here the topic is only one (crypto) and for documents, I’ve taken each document to be a separate crypto group on telegram.

Step 1:

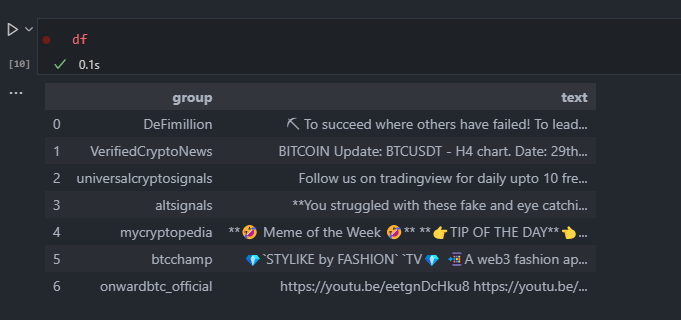
I’ve created a script scrape\_telegram.py to scrape for obtaining the messages

from targeted groups on telegram using an open source library called telethon.

Few of the groups I scrapped from are below.

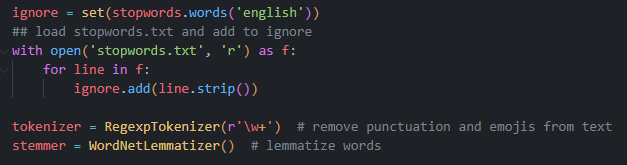


I have all the messages on each group in a separate row of the dataframe data\_scrapped.csv



Step 2:

For preprocessing, I’ve used the stopwords provided by nltk library as well as my custom list that is saved in stopwords.txt. I’ve gone ahead and additionally removed wallet ID and numbers from the text.



Step 3:

For the inputs to the LDA algorithm, I’ve used a count vectorizer that stores the count of each word across documents. Additionally I’ve also used bigrams as well. Bigrams are two consecutive words that are considered as part of the corpus.

.

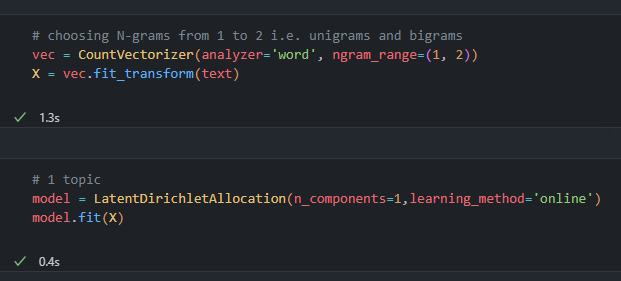
For example: In the sentence, “Espresso is awesome, and user-friendly” the bigrams are :

“Espresso is”

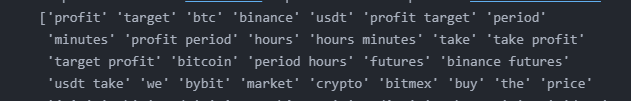
“is awesome”

“and user”

“user friendly”



Running the LDA algorithm on the preprocessed group text gives me a number of words with their importance towards a particular topic.



There are a few words that come up to the top that are not crypto related which can be added to the stopwords list(like “minutes”, “hours”, etc.)

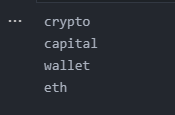
Part 2:

Now that we have a list of relevant crypto words, we can determine keywords used in a text conversation against this list using the same preprocessing methods.

For example in the sentence:

**“Yea this is definitely the case. They only started their crypto division in 2017. So plenty of capital to bail out their subsidiary. Now all eyes on that wallet to see where the eth moves.”**

The extracted keywords are:



The model would perform a lot better if the training size is increased (more documents are added) and extensive experimentation with its hyperparameters.

Advantages:

The LDA approach is an easier way of finding new crypto words.

The unsupervised approach guarantees recognition of newer crypto words (eg., new coins) as more data becomes available every day.

Future scope:

Using simple comparisons to get keywords from text conversation and keyword lists can be a very time consuming task as the size of the list increases. One solution for that is using a pre-trained bart model from hugging face and finetuning it with our keyword list to output these keywords directly.