CMP7202 – Web Social Media Analytics and Visualisation

Crypto analysis

Technical report

Zan Zver
18133498

BIRMINGHAM CITY
University

Faculty of Computing, Engineering and the Built Environment
Birmingham City University

Contents

List of Tables

List of Figures

Glossary

Crypto - Crypto currency
Crosspost - post shared from one subreddit to another
Subreddit - user runned form that focuses on one topic, referred as r/subreddit name

1 Data description

1.1 Reddit

Reddit API (Reddit n.d.) is free to use (in the time of writing) and it only requires user to create an account. There are different datasets available as well, for this report subredit datasets have been used in order to get specific set of data.

Topics (subredits) that were checked are crypto related (r/eth, r/btc, r/crypto). Data is gathered in two ways for each of the subreddits, by new and top in order to see the difference between them.

The Reddit data is used in two sections (A1, B1). Section A1 focuses in data analysis, therefore data with numeric properties was prioritised. B1 focuses on text analytic, therefore only text data is used. Figures bellow showcase what columns are used in each section

Author String	Author is blocked Bool	Awarders List	Banned at utc Date time	Banned by String
Can gild Bool	Category String	Created utc Date time	Downs Int	Edited Bool
Is crosspostable Bool	Is video Bool	Locked Bool	Mod note String	Mod reason by String
Mod reason title String	Mod reports String	Number of comments Int	Number of crossposts Int	Number of duplicates Int
Number of reports Int	Over 18 Bool	Removal reason String	Removed by String	Removed by category String
Score Int	Selftext String	Spoiler Bool	Title String	Top awarded type String
Total awards received Int	Ups Int	Upvote ratio Float	URL String	User reports List

Table 1: Reddit attributes used for data analysis

Author String	Author is blocked Bool	Awarders List	Banned at utc Date time	Banned by String
Can gild Bool	Category String	Created utc Date time	Downs Int	Edited Bool
Is crosspostable Bool	Is video Bool	Locked Bool	Mod note String	Mod reason by String
Mod reason title String	Mod reports String	Number of comments Int	Number of crossposts Int	Number of duplicates Int
Number of reports Int	Over 18 Bool	Removal reason String	Removed by String	Removed by category String
Score Int	Selftext String	Spoiler Bool	Title String	Top awarded type String
Total awards received Int	Ups Int	Upvote ratio Float	URL String	User reports List

Table 2: Reddit attributes used for text analysis

1.2 Crypto

In order to get crypto currency data, Etherscan API (Etherscan n.d.) was used. The usage is free for 5 calls per second and maximum of 100000 calls per day (in total).

Data is delivered in hexadecimal (eg 0x0FF) format, therefore everything is saved as a string by default. In the program, this is changed to the correct data type. After transformation, some of the data (eg From, To) would still not be as clear due to data being cryptic.

baseFeePerGas String	difficulty String	extraData String	gasLimit String	gasUsed String
hash String	logsBloom String	miner String	mixHash String	nonce String
number String	parentHash String	receiptsRoot String	sha3Uncles String	size String
stateRoot String	timestamp String	totalDifficulty String	transactions String	blockHash String
blockNumber String	from String	gas String	gasPrice String	maxFeePerGas String
maxPriorityFeePerGas String	hash String	input String	nonce String	to String
transactionIndex String	value String	type String	accessList String	chainId String
v String	r String	s String	jsonrpc String	id String

Table 3: Crypto attributes used

1.3 News

To get news from different sources, Newsapi (Newsapi n.d.) is used. Table 4 showcases what items are used from Newsapi.

Data is in String format which is appropriate for text analysis. Title and content sections are used for text analysis in this case.

source String	id String	name String	author String	title String
description String	url String	urlToImage String	publishedAt String	content String

Table 4: Attributes used from NewsAPI

2 Statistical analysis

2.1 A1

Crypto currency had its "boom" moments in the pasts and this can be seen from the posts over time. We can see that each quarter there wasn't a lot of posts until first crypto boom in Feb 2018. This happend again in October of 2021.

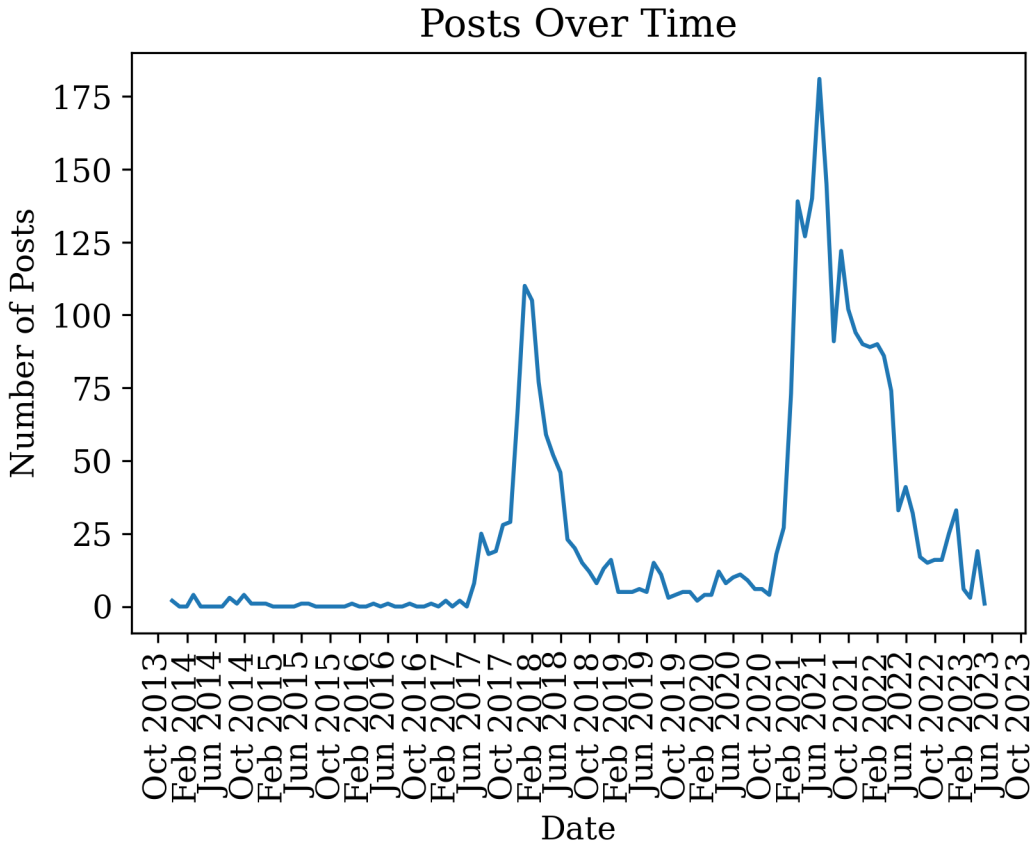


Figure 1: Number of posts over time

We can compare Google Trends (Google n.d.) searches to posts over time. By doing that, we see that post spikes matches with searches.

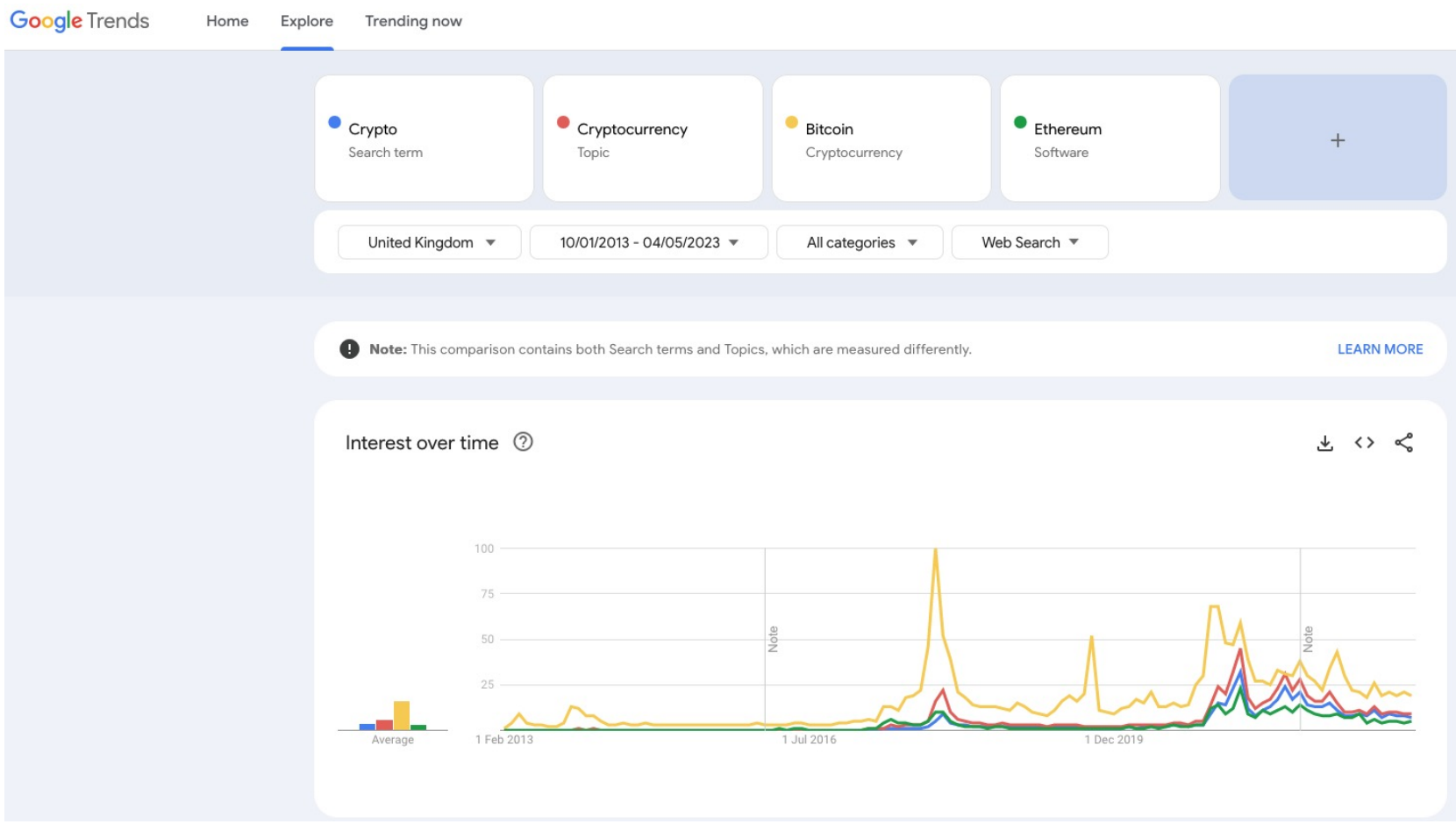


Figure 2: Google Trends for crypto search

If information is interesting, it drives a lot of attention. By this measure, does it mean that people will share (crosspost) this information more often due to its popularity?

The answer seems to be yes, since we can see that post with id 1066 has the highest number of crossposts and highest number of duplicates as well. After that there seems to be a decline for number of duplicates is somewhat correlated to number of crossposts. But there are some outliers (eg post 2147) that showcases number of duplicates being lower. One of the reasons this can be is post moderation (even auto moderation) where duplicates are removed automatically.

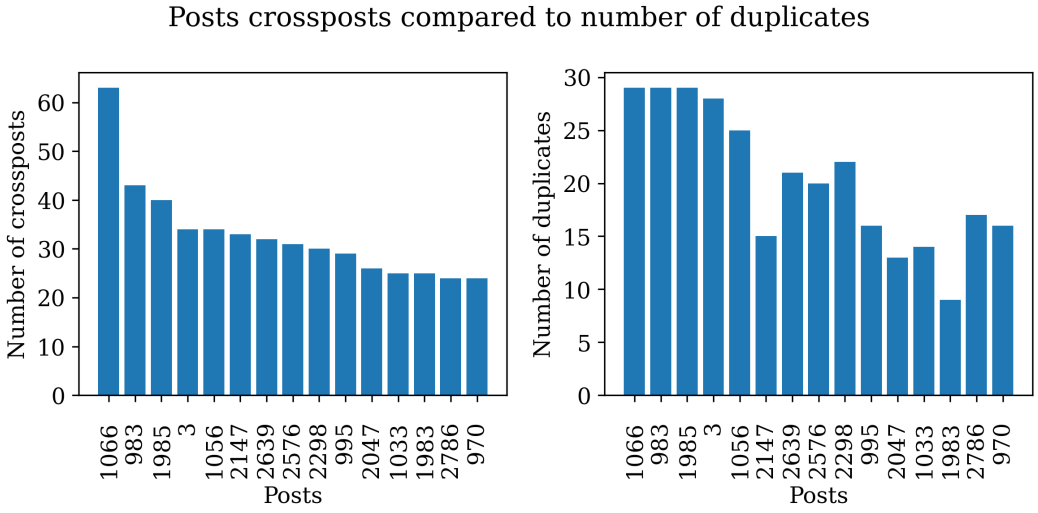


Figure 3: Crossposts and duplicates compared

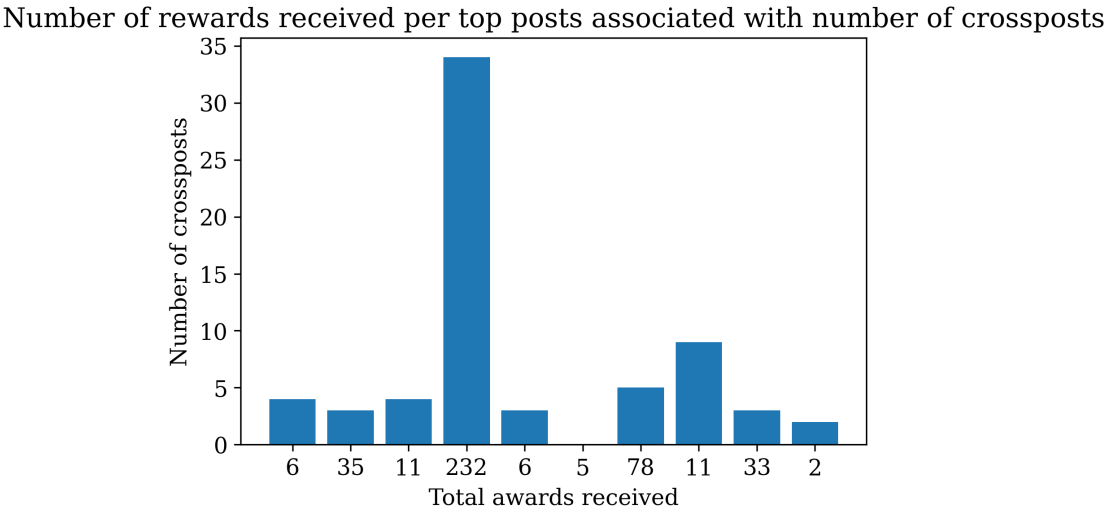


Figure 4: Crossposts associated with rewards

Assumption could be made that more comments post has more upvotes are there and more is it shared.

It turns out that subreddits seem to isolate topics, which means that it doesn't matter how many upvotes or comments you will get, number of crossposts are not related to that. We can see the most upvoted post with 7741 upvotes is just a "regular post", while a 3rd largest post with 7444 upvotes has a lot of comments and not a lot crosspots. 4th largest post (7225) is opposite of 3rd one, it has a lot of crosspots but not a high amount of comments.

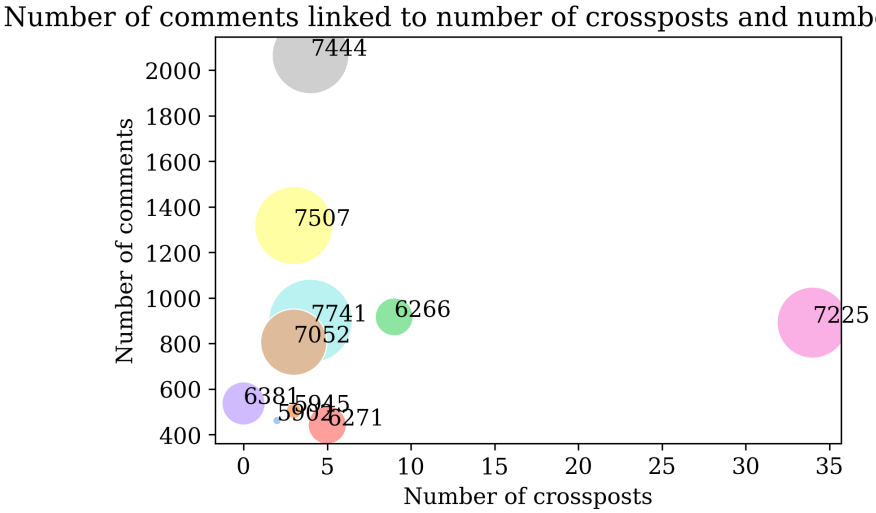


Figure 5: Number of comments correlated with with crosspots

Being active poster in the community will bring users some recognition. We can see that users with more posts, have higher number of upvotes.

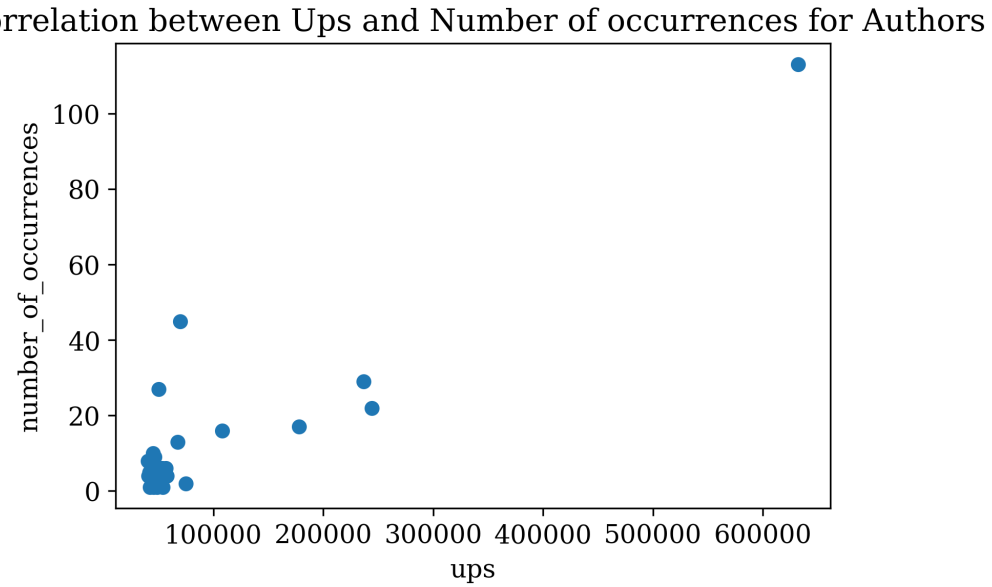


Figure 6: Number of upvotes per user

Post upvotes and awards are not correlated. If a post is more popular, it will not get more awards.

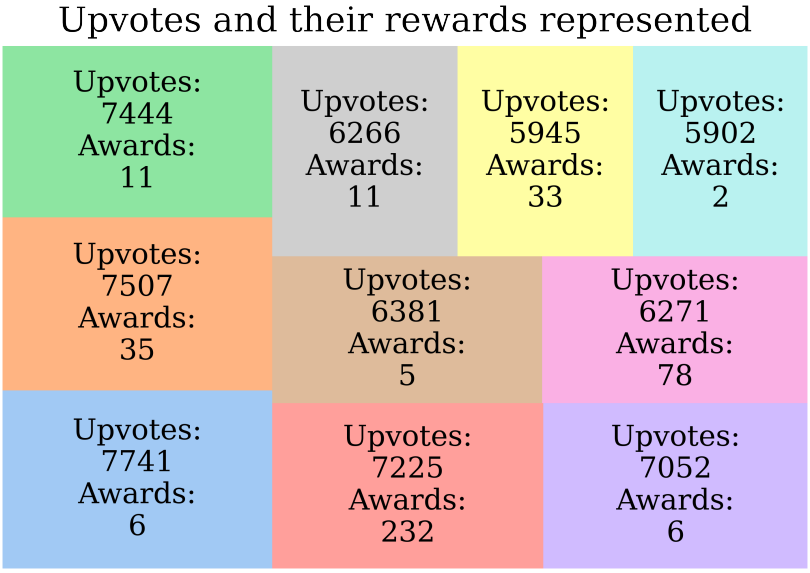


Figure 7: Top posts and their awards

2.2 A2

After inserting crypto transactions data to Gephi and configuring as needed, we can generate a map of connections. There are loads of nodes in our graph, using

Yifan Hu layout algorithm, and setting node size based on number of degree, we can showcase crypto map. Based in that, we can see green node has the most transactions and pink nodes seems to have the most members in its community.

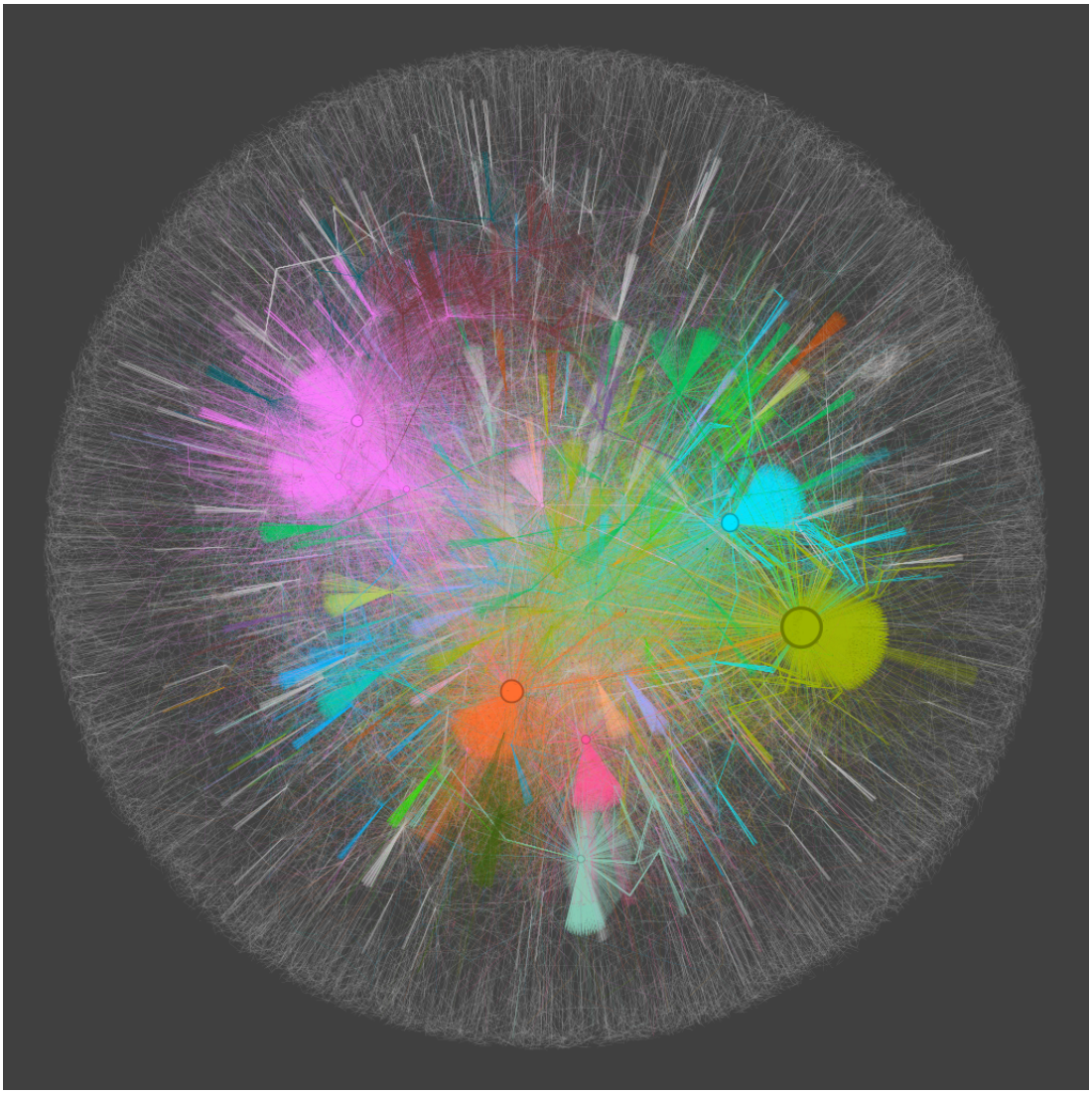


Figure 8: Transfers of crypto currency

In Gephi we can see all connections of the node. If we click on lime green, we can see how it is connected to the most of the nodes.

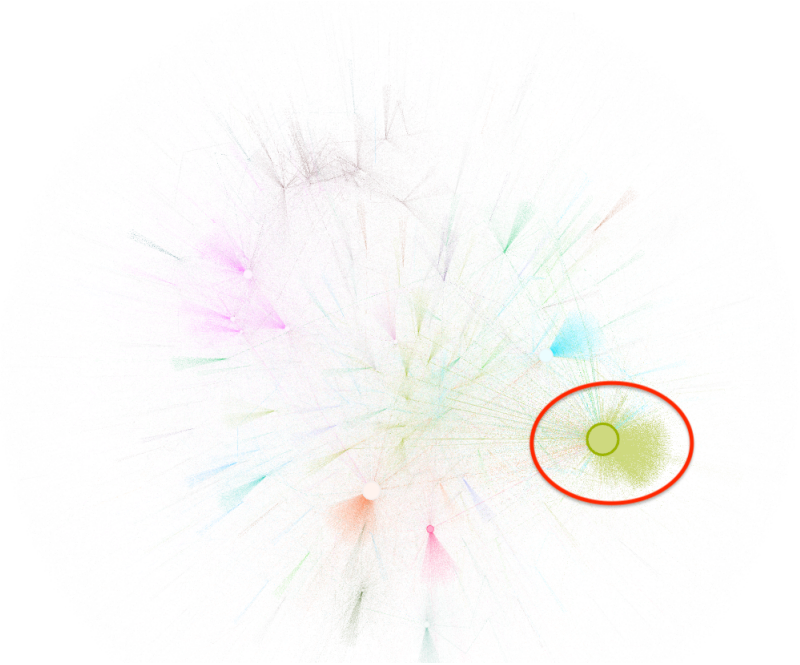


Figure 9: Showcase of biggest node

Selecting red (pink) node we can see that number of connections looks similar but less than main node.

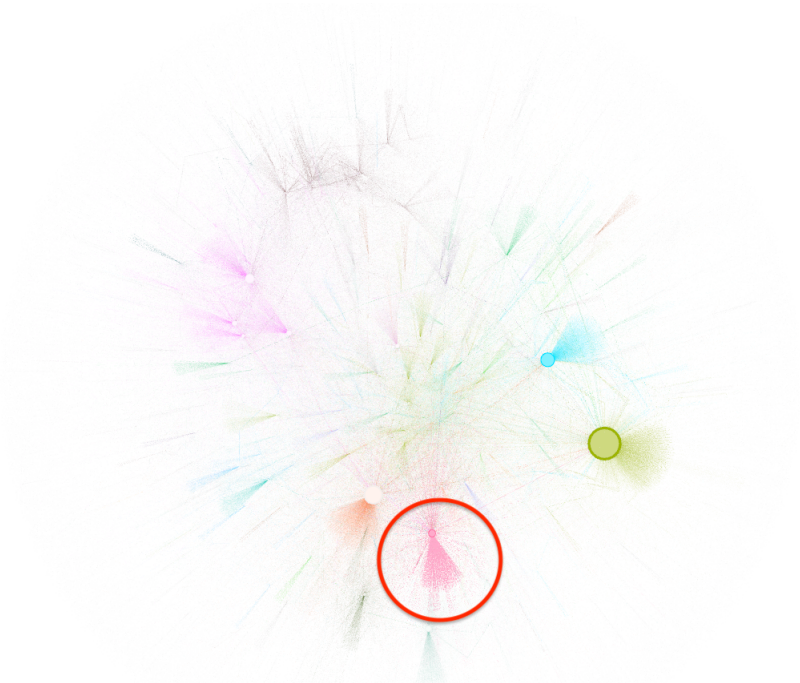


Figure 10: Showcase of smaller node

Gephi also generates network (graph) information for us. For example we can

see that average degree is 1.177 and modality is 0.82.

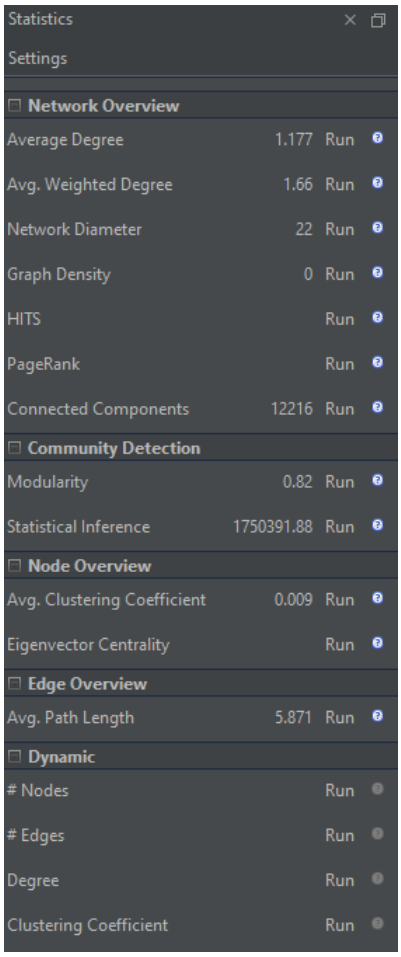


Figure 11: Gephi statistics

In total, this network has 152165 nodes and 179024 edges.

Figure 12: List of nodes and edges

3 Text mining

3.1 B1

In order to analyse the data, 3 subreddits were used with 2 filters applied. Table bellow showcases them

Subreddit	Top	New
Ethereum	✓	✓
Bitcoin	✓	✓
CryptoCurrency	✓	✓

Table 5: Subreddit data filter

Investigating top words, we can see that Ethereum is top mentioned word, but bitcoin is there as well. This is not surprising since bitcoin is the most popular (and well known) crypto currency. Besides that, there are other related words such as NFT, blockchain, fee, etc... that are used in the context of crypto currency topics.

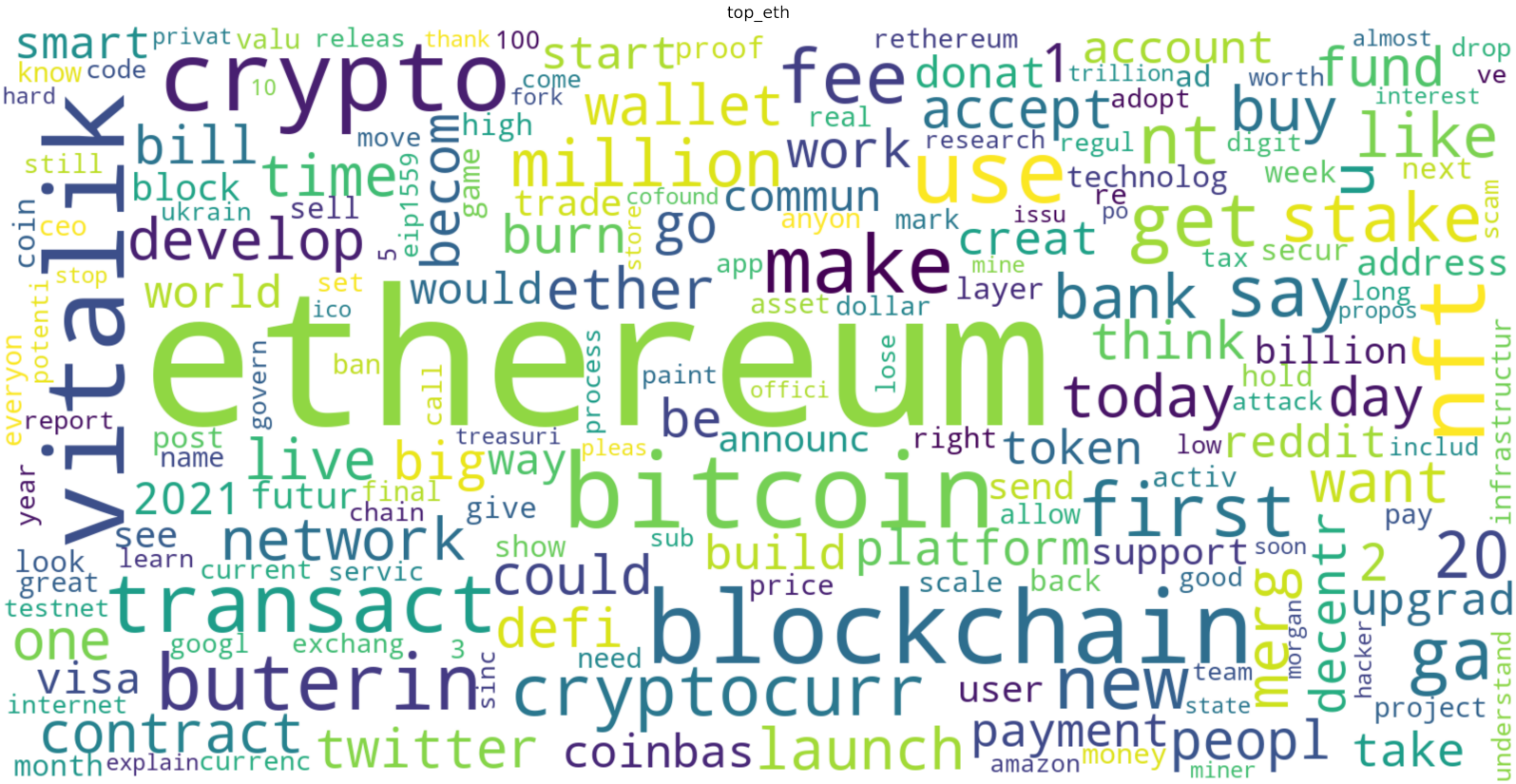


Figure 13: Top r/Ethereum wordcloud

Reflecting to new data, Ethereum is still the top mentioned world (alongside blockchain and NFT). There are some new words that are introduced such as web3 and stake due to change of direction Ethereum is heading.

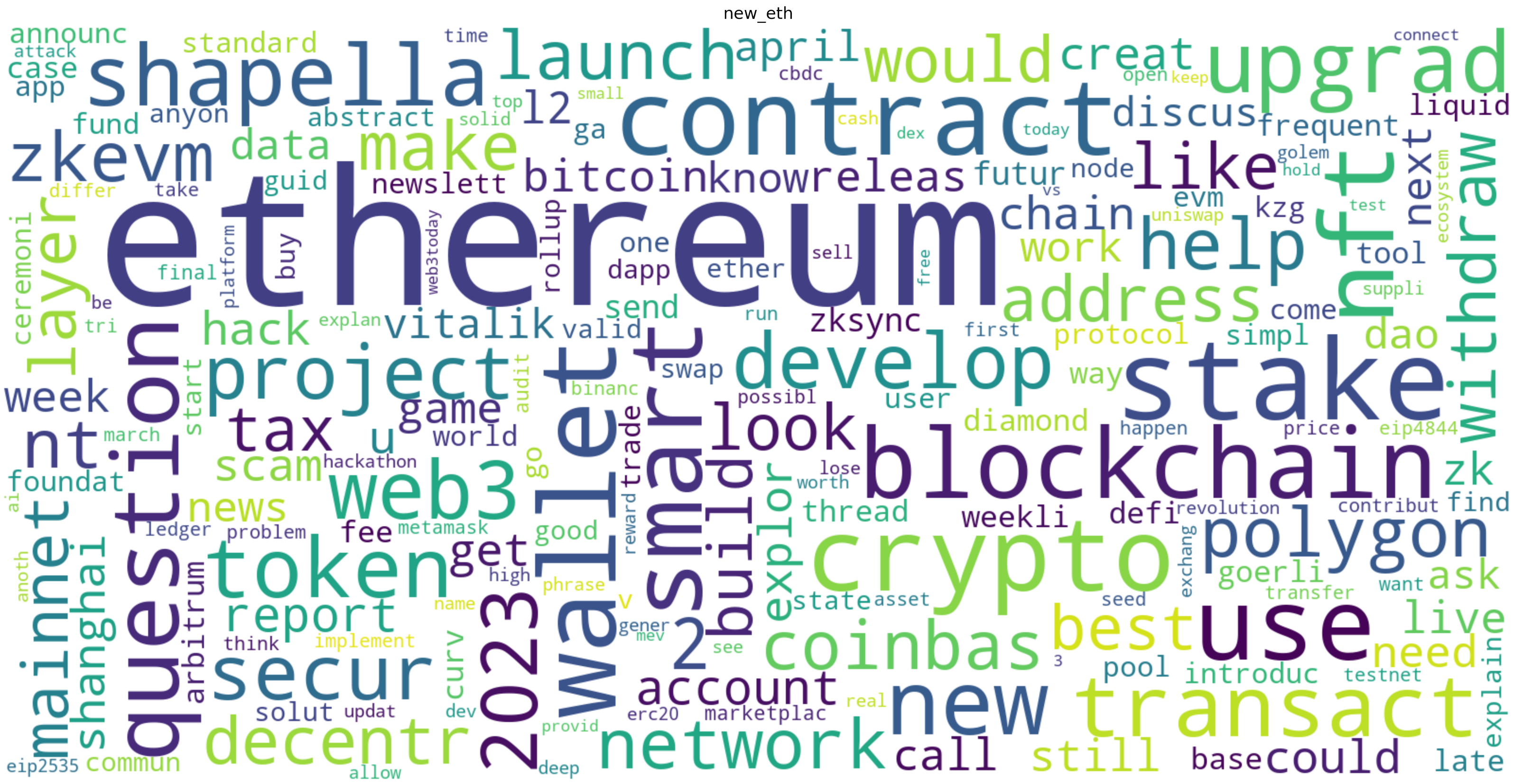


Figure 14: New r/Ethereum wordcloud

Top word for bitcoin is bitcoin and there aren't any other crypto currencies mentioned. There is mention of author (Satoshi) and positive influence of buy, get, go, which could imply users saying "buy bitcoin now". Sell is also mentioned but not as much as buy.

Figure 15: Top r/Bitcoin wordcloud

Trends continues with new words, it is in similar domain. Bitcoin is top mentioned, buy is there as well, and price is mentioned. This could imply the same message as the top words form above.

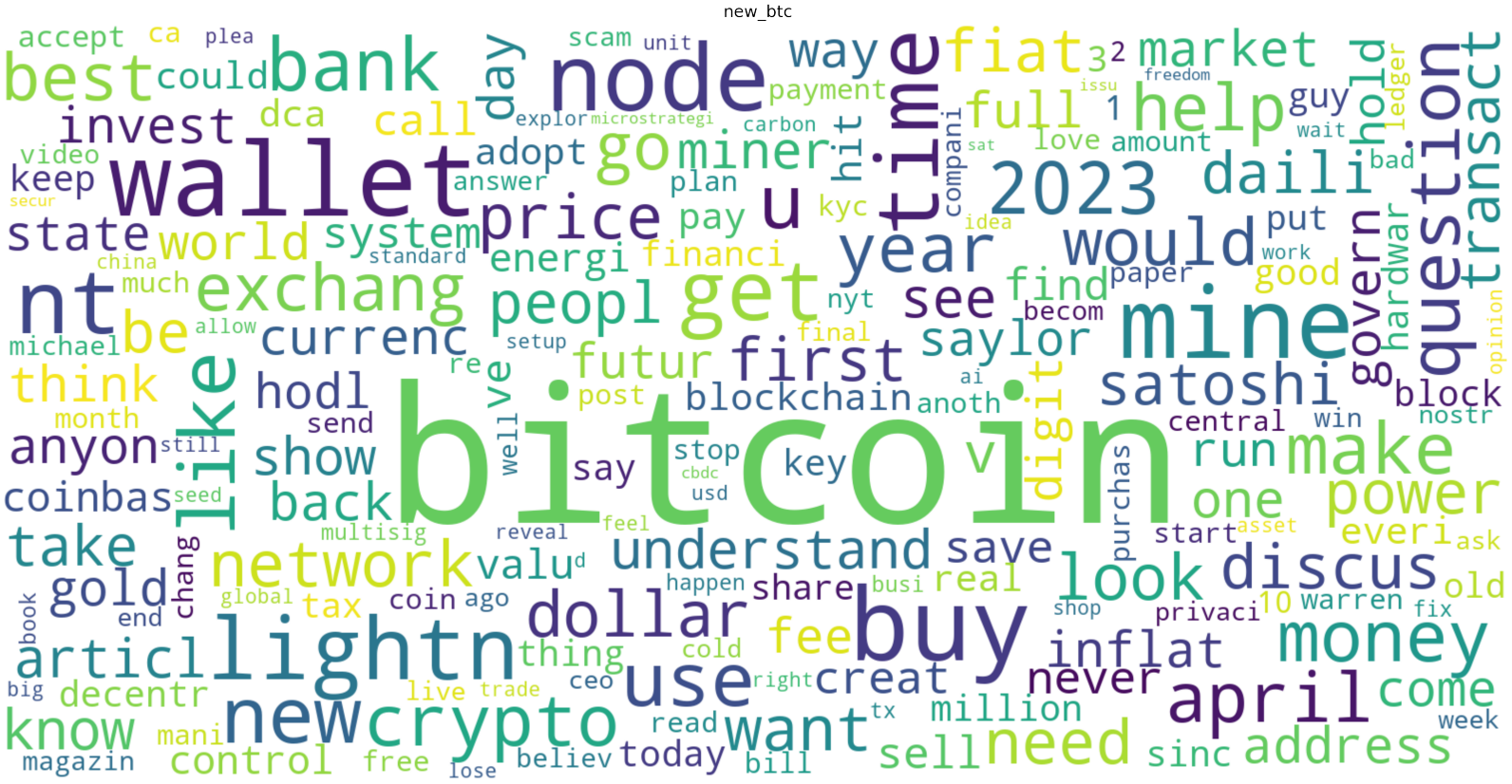


Figure 16: New r/Bitcoin wordcloud

Top crypto topics seems to be around crypto currencies and bitcoin sits on top. Ethereum is there as well but not as popular. Based on the words (market, price, monkey, worth) it seems this is more of a discussion regarding crypto value and not advice regarding selling or buying (but there is mention of sell).

Figure 17: Top r/CryptoCurrency wordcloud

New worldcloud has the similar structure as top. Main topics are still crypto, bitcoin and Ethereum but there are new topic on horisen such as sec and Gensler (head of sec) due to new rules of crypto markets.

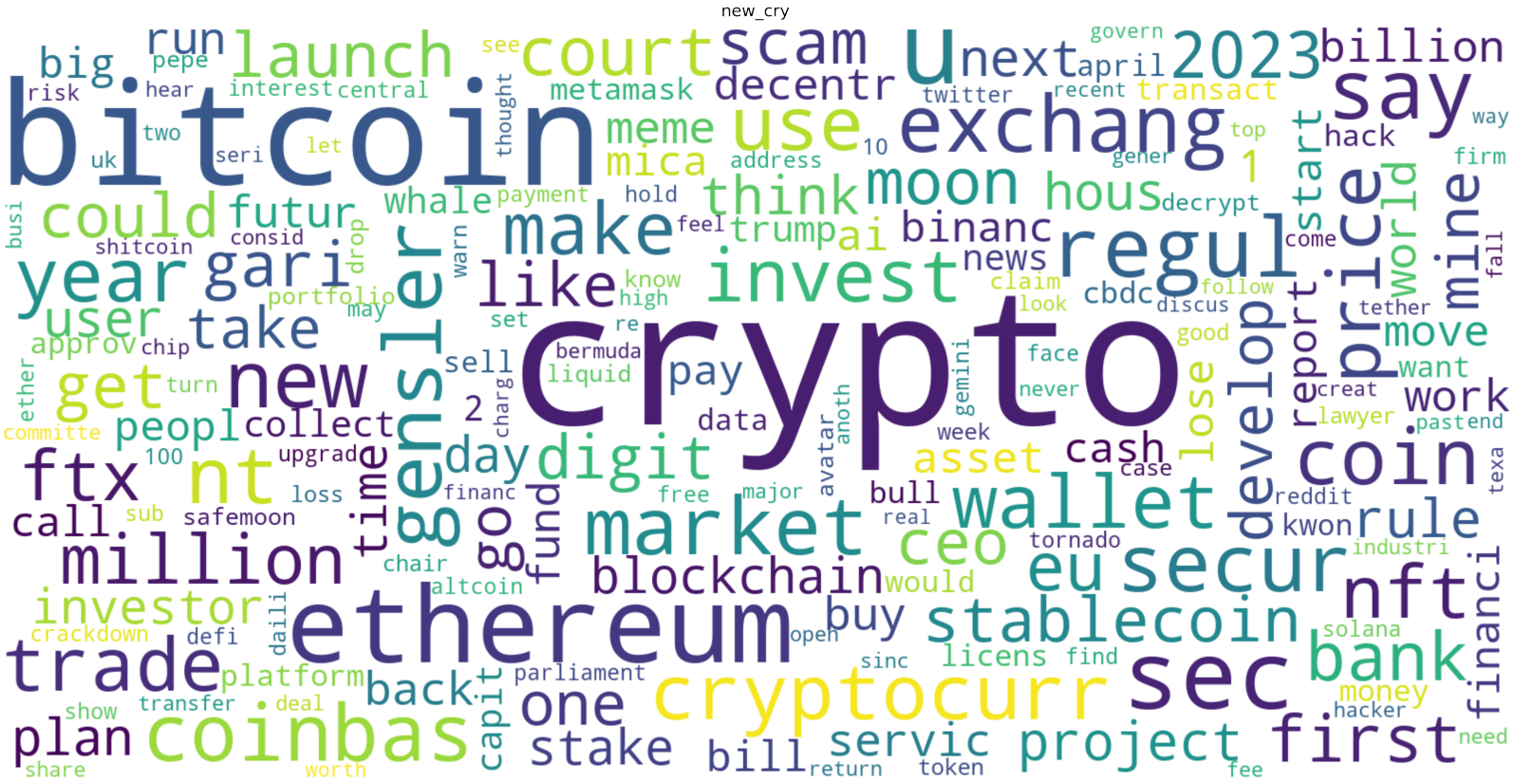
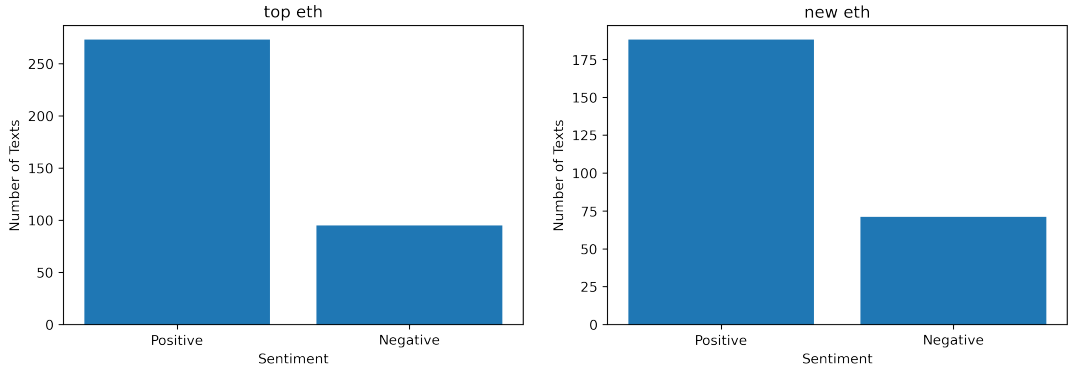


Figure 18: New r/CryptoCurrency wordcloud

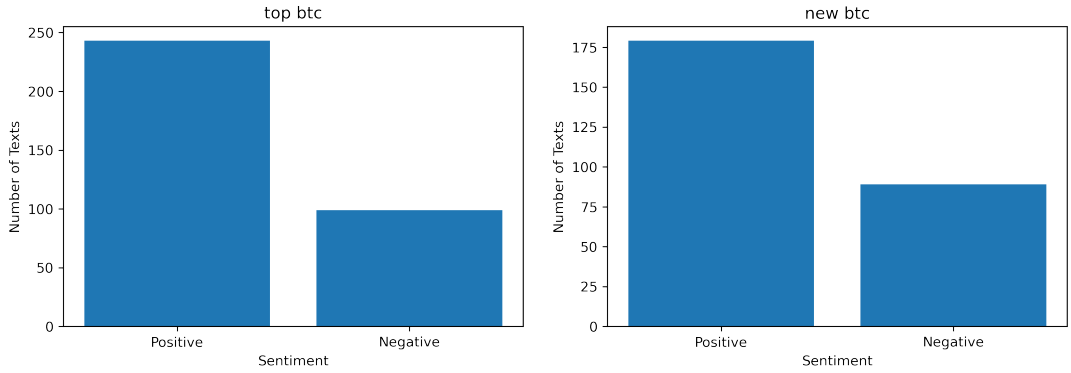
Creating sentiment analysis on Ethereum subreddit shows overall positive approach of messages posted.



(a) Top r/Ethereum sentiment (b) New r/Ethereum sentiment

Figure 19: Comparison of r/Ethereum sentiment

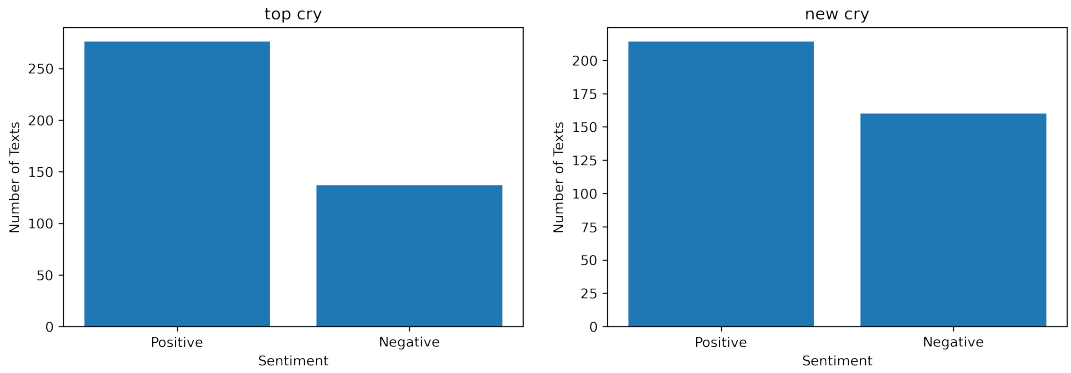
Looking over to bitcoin subreddit, it is similar in therms of top posts but new posts seems to be more on negative side.



(a) Top r/Bitcoin sentiment (b) New r/Bitcoin sentiment

Figure 20: Comparison of r/Bitcoin sentiment

Crypto currency subreddit seems to have the most negative observations. This is in terms of top posts and new.



(a) Top r/CryptoCurrency sentiment (b) New r/CryptoCurrency sentiment

Figure 21: Comparison of r/CryptoCurrency sentiment

3.2 B2

In order to have just alpha numeric characters in our dataset, custom regex function was created for title, description and content columns. This would exclude special characters (such as emojis) from our data.

```
def apply_regex(data_frame):
    pattern = r'[^a-zA-Z0-9 ]'

    data_frame['title'] = data_frame['title']
                          .apply(lambda x: re.sub(pattern, '', x))
    data_frame['description'] = data_frame['description']
                          .apply(lambda x: re.sub(pattern, '', x))
    data_frame['content'] = data_frame['content']
                          .apply(lambda x: re.sub(pattern, '', x))
    return data_frame
```

Listing 1: Regex function

In text processing, we can remove stop words (Vijayarani, Ilamathi, Nithya, et al. 2015) in order to get rid of unnecessary text. With the help of NLTK (Nltk n.d.) library, function bellow does this for us.

```
def remove_stopwords(df):
    stop_words = set(stopwords.words('english'))

    df['title'] = df['title'].apply(lambda x: ' '
    .join([word for word in x.split() if word.lower() not in stop_words]))
    df['description'] = df['description'].apply(lambda x: ' '
    .join([word for word in x.split() if word.lower() not in stop_words]))
    df['content'] = df['content'].apply(lambda x: ' '
    .join([word for word in x.split() if word.lower() not in stop_words]))

    return df
```

Listing 2: Remove stopwords function

If we want to reduce words to its root, we can apply stemming function (Jivani et al. 2011). This will be done to title, description and content columns of our dataframe.

```
def apply_stemming(df):
    stemmer = PorterStemmer()

    df['title'] = df['title'].apply(lambda x: ' '
    .join([stemmer.stem(word) for word in word_tokenize(x)]))
    df['description'] = df['description'].apply(lambda x: ' '
    .join([stemmer.stem(word) for word in word_tokenize(x)]))
    df['content'] = df['content'].apply(lambda x: ' '
    .join([stemmer.stem(word) for word in word_tokenize(x)]))

    return df
```

Listing 3: Stemming function

On the other hand, if we want to apply lemmatization (Plisson, Lavrac, Mladenic, et al. 2004) in order to reduce words to their dictionary form, we can call the function bellow. It will act in a same way as stemming function.

```
def apply_lemmatization(df):
    lemmatizer = WordNetLemmatizer()

    df['title'] = df['title'].apply(lambda x: ' '
    .join([lemmatizer.lemmatize(word) for word in word_tokenize(x)]))
    df['description'] = df['description'].apply(lambda x: ' '
    .join([lemmatizer.lemmatize(word) for word in word_tokenize(x)]))
    df['content'] = df['content'].apply(lambda x: ' '
    .join([lemmatizer.lemmatize(word) for word in word_tokenize(x)]))

    return df
```

Listing 4: Lematization function

If we want to have separate steamming and lemmatization function (that does both), we can call apply_stemming_and_lemmatization. It does the same as two functions above but it is joined.

```
def apply_stemming_and_lemmatization(df):
    stemmer = PorterStemmer()
    lemmatizer = WordNetLemmatizer()

    df['title'] = df['title'].apply(lambda x: ' '
    .join([lemmatizer.lemmatize(stemmer.stem(word))
    for word in word_tokenize(x)]))
    df['description'] = df['description'].apply(lambda x: ' '
    .join([lemmatizer.lemmatize(stemmer.stem(word))
    for word in word_tokenize(x)]))
    df['content'] = df['content'].apply(lambda x: ' '
    .join([lemmatizer.lemmatize(stemmer.stem(word))
    for word in word_tokenize(x)]))

    return df
```

Listing 5: Stemming and lemma function

With topic discovery (Pons-Porrata, Berlanga-Llavori, and Ruiz-Shulcloper 2007), we send in number of topics and words we want to get (defaults are set). Afterwards, data (title, description and content) is joined to list and tokenised. LdaModel (from gensim library) is used in order to generate topic models (this can be optimised with ldamulticore). At the end, dataframe of topics is returned.

```
def discover_topics(df, num_topics=5, num_words=10):
    data = df[['title', 'description', 'content']].values.tolist()
    data = [' '.join(simple_preprocess(str(d))) for d in data]
    tokens = [d.split() for d in data]

    dictionary = Dictionary(tokens)
    corpus = [dictionary.doc2bow(doc) for doc in tokens]

    lda_model = LdaModel(corpus=corpus,
                         id2word=dictionary,
                         num_topics=num_topics,
                         random_state=100,
                         chunksize=100,
                         passes=10,
                         alpha='auto',
                         per_word_topics=True)

    topics = lda_model.show_topics
    (num_topics=num_topics, num_words=num_words, formatted=False)

    topics_df = pd.DataFrame(columns=['topic_words'])

    for topic in topics:
        topic_id = topic[0]
        topic_words = [word[0] for word in topic[1]]
        topic_words = ', '.join(topic_words)
        topics_df = topics_df.append
        ({'topic_words': topic_words}, ignore_index=True)

    return topics_df
```

Listing 6: Topic discovery function

In order to summarise articles, dataframe is passed in the function. In it, external library (gensim.summarization) (Gensim n.d.) is used in order to create summarization for each article which is returned at the end.

```
def summarize_description(data_frame):
    summaries = []
    for _, row in data_frame.iterrows():
        description = row['description']
        if len(description.split('.')) == 1:
            summaries.append(description)
        else:
            summary = summarize(description, word_count=30)
            summaries.append(summary)

    return pd.DataFrame({'summary': summaries})
```

Listing 7: Topic discovery function

In the main function we can see execution of all the functions mentioned above. Data is read from CSV due to consistency. If we want to get new batch of news, get news function can be called in order to generate new data.

```
#news = get_news(news_api_key,"bitcoin")
news_bitcoin = pd.read_csv("Data/News/bitcoin.csv", index_col=0)
news_etherium = pd.read_csv("Data/News/etherium.csv", index_col=0)
news_crypto = pd.read_csv("Data/News/crypto.csv", index_col=0)
news_cryptocurrency = pd.read_csv("Data/News/cryptocurrency.csv", index_col=0)

frames = [news_bitcoin, news_etherium, news_crypto, news_cryptocurrency]
news_all = pd.concat(frames)

# Apply regex
news_all = apply_regex(news_all)

news_stopwords = remove_stopwords(news_all)

apply_stemming(news_stopwords)[["title","description","content"]]
apply_lemmatization(news_stopwords)[["title","description","content"]]
news_stem_lemma = apply_stemming_and_lemmatization
(news_stopwords)[["title","description","content"]]

topics = discover_topics(news_stopwords, num_topics=6, num_words=10)
generate_wordclouds(news_stopwords)


plot_source_pie(news_all, n=6)
plot_publication_dates(news_all)
plot_word_count(news_stem_lemma)
```

Listing 8: Main function

In the news (content section) bitcoin was the most mentioned word. This matches crypto currency subreddit. Only anomaly is duplication of some words (eg crypto).

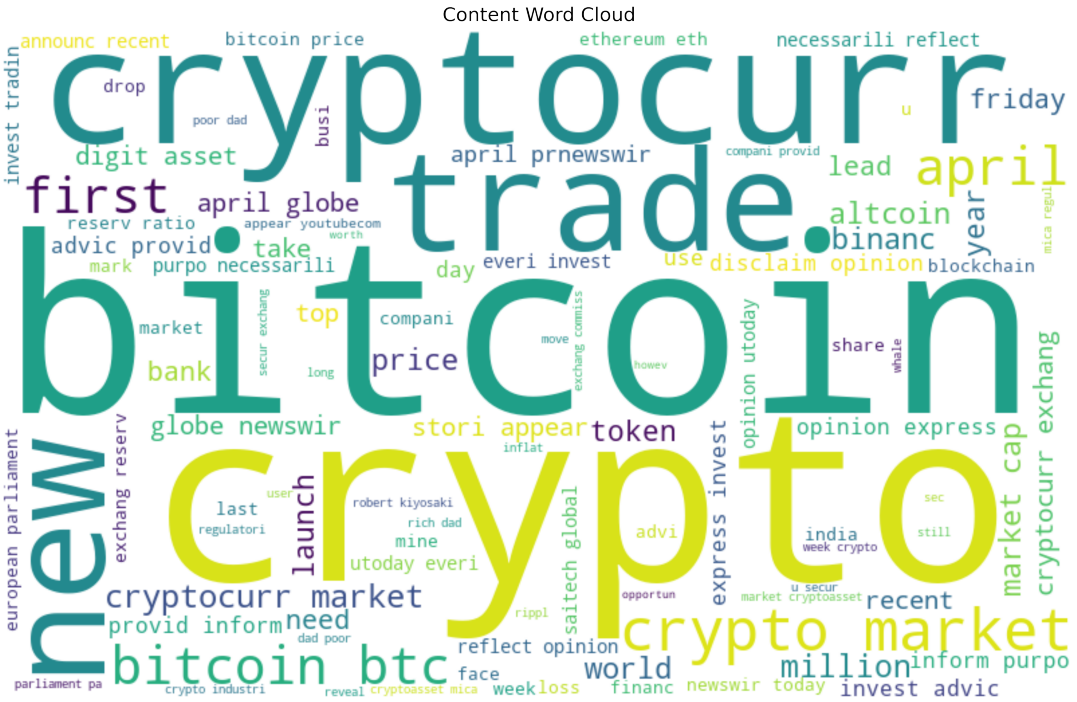


Figure 22: News content wordcloud

Description section has similar structure, in comparison to content.This makes sense due to authors describing articles in similar styles compared to actual story.

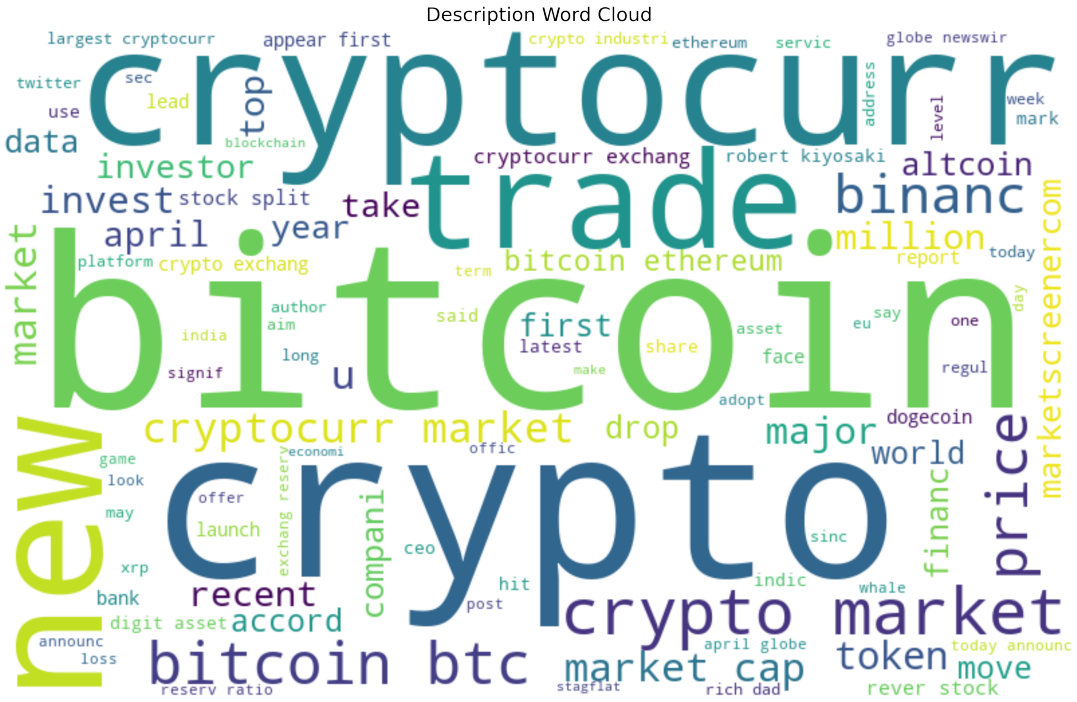


Figure 23: News description wordcloud

Titles are much different compared to content and description worldclouds. Bitcoin and crypto are the top mentioned, but new words make an appearance (such as market). This is probably a tactical move in order to lure in new readers with the phrases known to them.

Figure 24: News title wordcloud

Publication hours for most of the articles are around noon. This could simply be writers attempt to get articles out before lunch time since users tend to read after that time (and not in the evening).

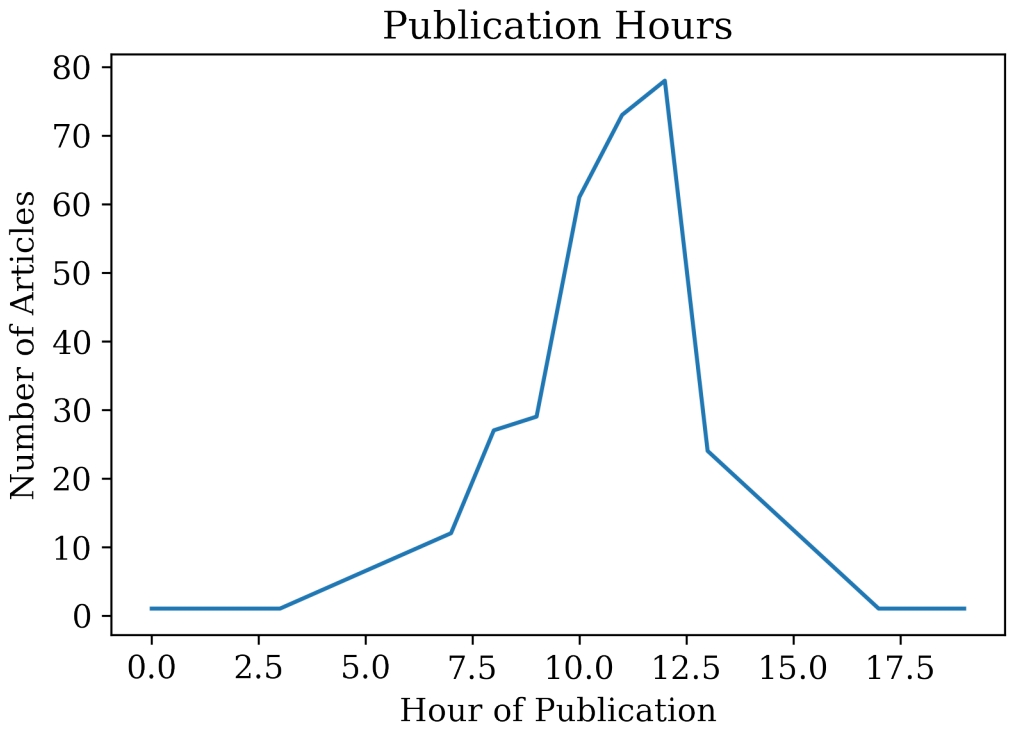


Figure 25: Publication hours of articles

Majority of articles are from Biztoc, a financial outlet. After further investigation, Biztoc is just a collection of articles from other sources (eg: Wallstreet Jurnal). After that, Marketscrenner and Coin desk seem to be specialising into financial news that are hosted on their site. Due to that we can say Biztoc is the biggest source of news but Marketscrenner and Coin desk are the biggest sources of original articles.

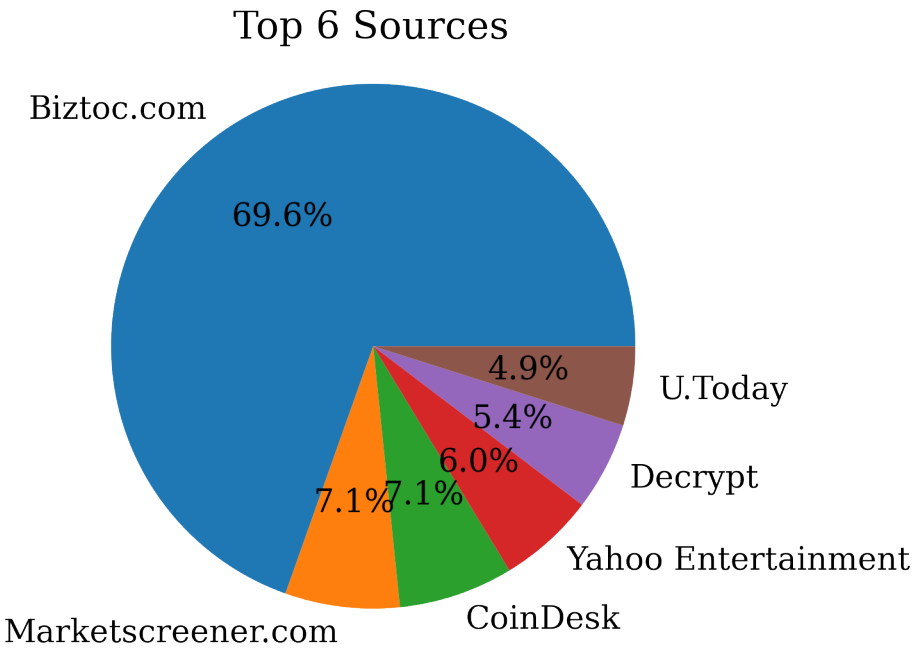


Figure 26: Top 6 sources pie chart

Creating a pie chart for most common words, we can see that crypto and bitcoin are the most popular under all of the words.

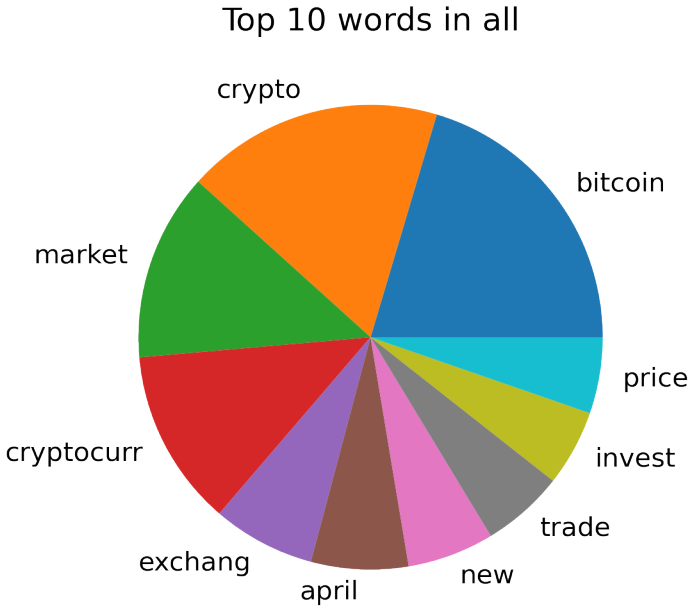


Figure 27: Top words for all

Content section looks similar to all, but market has a smaller portion and there

is a new candidate ”21”.

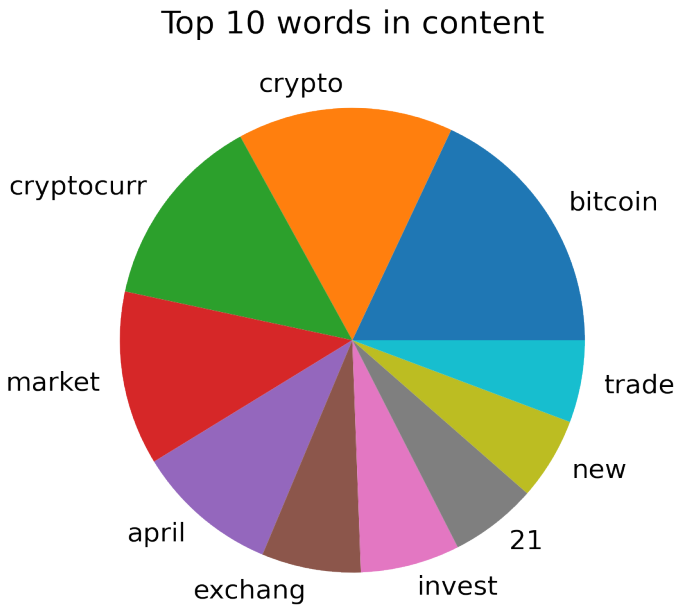


Figure 28: Top words for content

Description looks similar as well but it has 2023 in instead of 21.

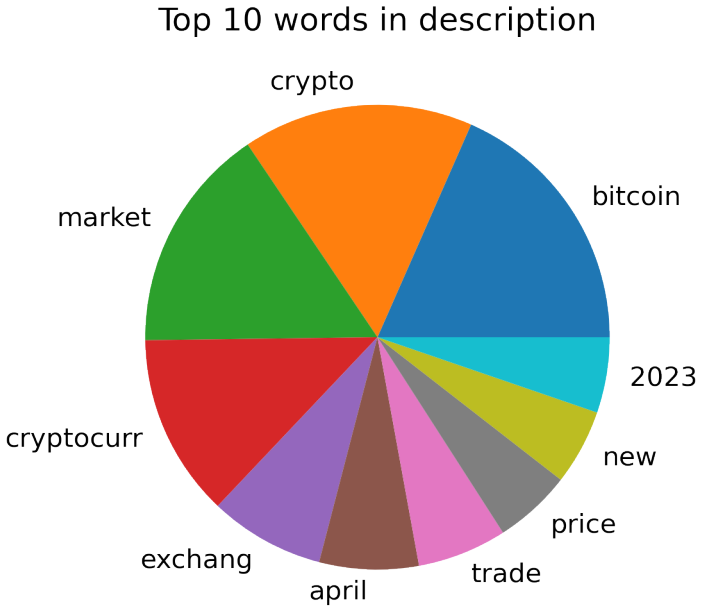


Figure 29: Top words for description

Title pie chart looks the most different. Although Bitcoin and crypto are on top, they share 50% of the pie. Other well known segments from before (example: market) are much smaller.

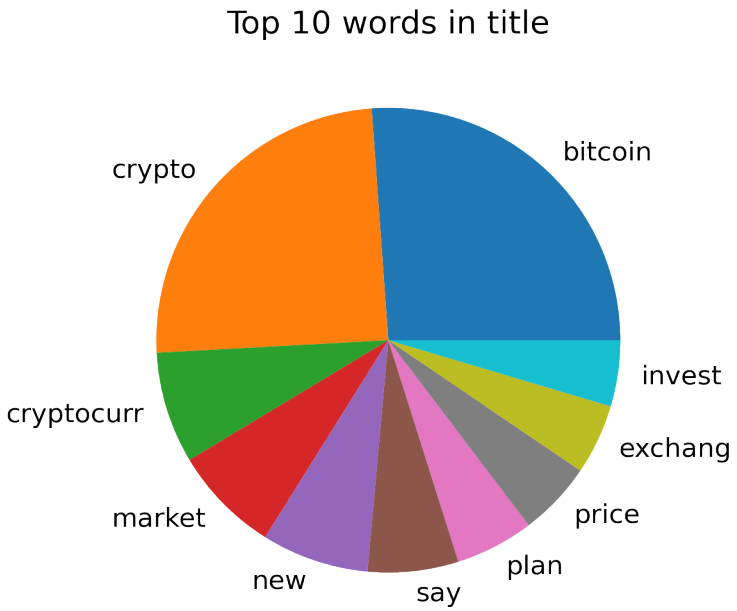


Figure 30: Top words for title

Looking at provided topics, we can see that our function did extract most popular topics. They seem to match search terms and its subtopics (example: energy).

topics

✓ 0.0s

	topic_words
0	bitcoin, cryptocurr, ethereum, drop, market, c...
1	bitcoin, energi, consumpt, char, mine, network...
2	crypto, cryptocurr, payment, rippl, recent, ch...
3	plan, announc, india, ceo, april, char, new, c...
4	gemini, hub, cryptocurr, consensu, exchang, ch...
5	singapor, crypto, list, mooi, bitcoin, sec, ne...

Figure 31: Article topics

Summaries are not as impressive as thought. Since they have been text processed, some of the context has been removed. With that it is harder to get deeper understanding of the article. Example is article number 3, where we have upgrad downgrad, this two words don't have any special meaning.

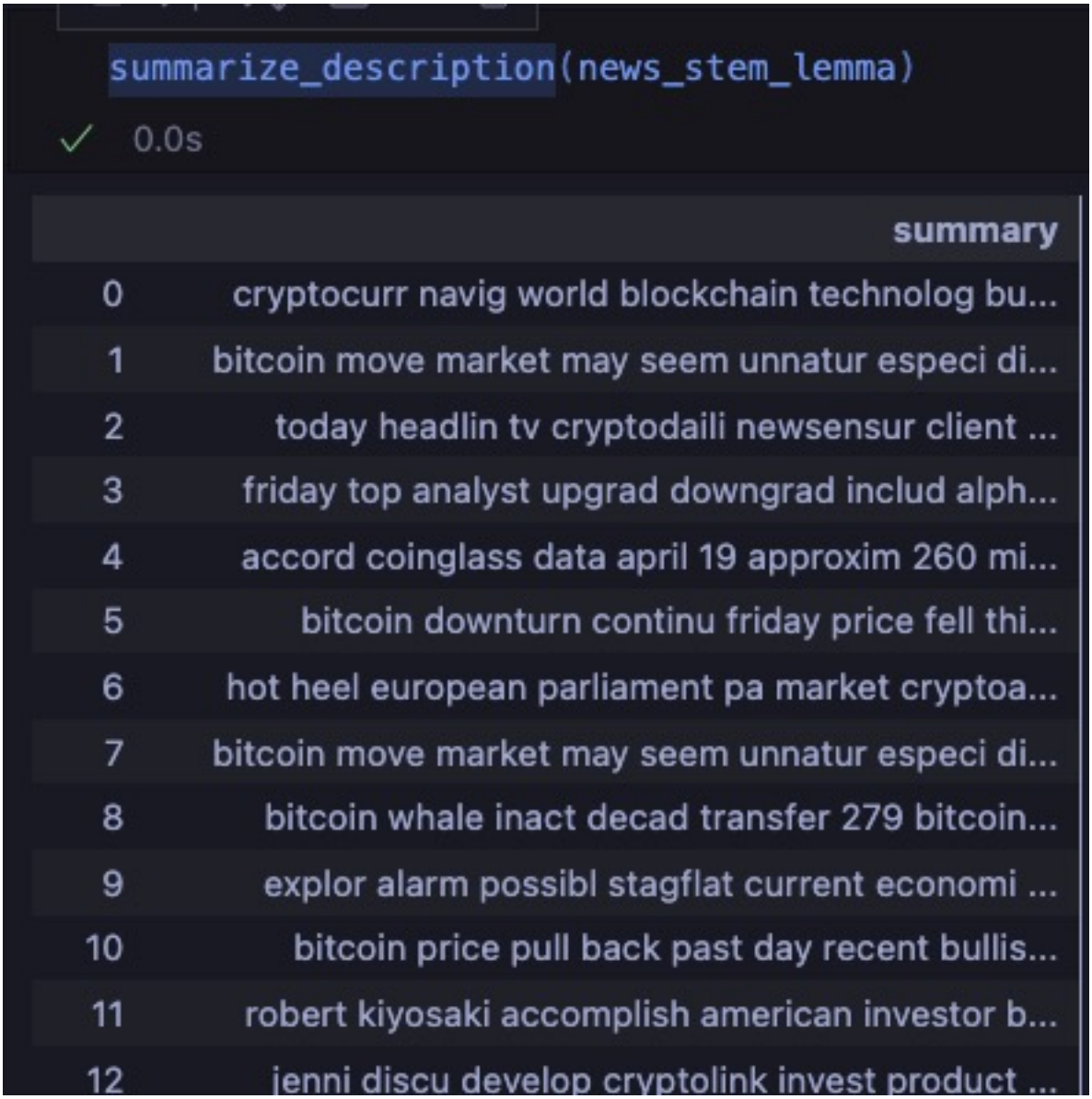


Figure 32: Article summary

4 Conclusions

Text mining is a powerful technique in order to analyse not just web data but user data as well. We can see how real world trends effect user behaviour.

With the rise of new technologies (such as ChatGPT and Google Bard), we need to be careful as users and as researchers. Internet is source of information for those models and there is a lot of misinformation on it. If input data is false, we could be reading false information generated from these models.

With simple Python tools and few APIs, we can get much more data as demonstrated. This could be extended by linking to other social media websites and create "whats hot" model that could get us key topics in crypto or in general.

Code for this report is available on GitHub (Zver n.d.).

5 References

References

Etherscan (n.d.). *Etherscan*. Last accessed 4 May 2023. URL: https://etherscan.io/apis.

Gensim (n.d.). *Gensim*. Last accessed 4 May 2023. URL: https://radimrehurek.com/gensim_3.8.3/summarization/summariser.html.

Google (n.d.). *GoogleTrends*. Last accessed 4 May 2023. URL: https://trends.google.com/trends/explore?date=2013-01-10%5C%202023-05-04&geo=GB&q=Crypto,%5C%2Fm%5C%2F0vpj4_b,%5C%2Fm%5C%2F05p0rrx,%5C%2Fm%5C%2F0108bn2x&hl=en-GB.

Jivani, Anjali Ganesh et al. (2011). "A comparative study of stemming algorithms". In: *Int. J. Comp. Tech. Appl* 2.6, pp. 1930–1938.

Newsapi (n.d.). *Newsapi*. Last accessed 4 May 2023. URL: https://newsapi.org/.

Nltk (n.d.). *Nltk*. Last accessed 4 May 2023. URL: https://www.nltk.org/.

Plisson, Joël, Nada Lavrac, Dunja Mladenic, et al. (2004). "A rule based approach to word lemmatization". In: *Proceedings of IS*. Vol. 3, pp. 83–86.

Pons-Porrata, Aurora, Rafael Berlanga-Llavori, and José Ruiz-Shulcloper (2007). "Topic discovery based on text mining techniques". In: *Information processing & management* 43.3, pp. 752–768.

Reddit (n.d.). *Reddit*. Last accessed 4 May 2023. URL: https://www.reddit.com/dev/api/.

Vijayarani, S, Ms J Ilamathi, Ms Nithya, et al. (2015). "Preprocessing techniques for text mining-an overview". In: *International Journal of Computer Science & Communication Networks* 5.1, pp. 7–16.

Zver, Zan (n.d.). *Code on GitHub*. Last accessed 4 May 2023. URL: https://github.com/ZanZver/SocialMediaAnalyticsCoursework.