Sentiment Analysis of Public Crypto Channels on Telegram

Farah Hersi
Birkbeck, University of London

MSc Computer Science
April 2019

This proposal is substantially the result of my own work, expressed in my own words, except where explicitly indicated in the text. I give my permission for it to be submitted to the JISC Plagiarism Detection Service.

Table of Contents

Abstract

I will design, implement, deploy and validate a software solution, based on data extraction and automated sentiment analysis to monitor, possibly in real-time, public channels discussing crypto-currency projects on telegram.

Markets are influenced by greed and fear. In traditional markets, price action and consumer behaviour are both open and analysed. Vice versa the crypto space today is unstructured and chaotic from the outside, crowded with actors, investors, speculators and "sages". Security and anonymity, coupled with hype and snake oil salesmen, obfuscate and muddy the waters. It becomes challenging to infer facts from fiction, action from causality let alone pricing assets and derivatives. In this study, I will focus my research on opinion mining public telegram channels to understand how sentiment drives the market.

Problem Description

Crypto Currency (cc) is considered a digital asset. Bitcoin was the first digital currency which was born in 2009. Since then, over 4000 Alternative currencies (ALTCOINS) have been launched. Cc's are based around the concept of decentralised control where actors and stakeholders "vote" through POS (proof of stake) or POW (proof of work) to verify transactions. As mentioned in [12], the main points with cc are anonymity and decentralisation. POS is a voting mechanism where 'oracles' who are vested in the community vote publicly on transactions and are rewarded with coins for their efforts. POW utilises the

‘Mining’ concept where computational effort (and energy) is expended in solving complex mathematical problems which verify transactions[12]. This is contrary to the current centralised financial system which is based on central banks.

Telegram is an open sourced (recent versions are no longer open source) instant messaging service based on the cloud. It provides end-to-end encryption on voice calls and secret chats between two parties. According to a 2017 statement by its current CEO, telegram has over half a million new registrations daily [1], coupled with over 200 Million active users [2] it provides a wealth of information for analysis. Although telegrams popularity has grown, it has come under sharp criticism by its users, since telegram is generally used for its security features as politically oppressed persons in sensitive states use telegram as a medium to communicate with the outside world. The criticism is mainly based around telegrams storing of users’ messages and multimedia along with their decryption keys on its server [8]. This poses a risk for its users.

My project will examine communication in public crypto channels on telegram. These “crypto channels” are the public-facing channels that these projects use to communicate with their community; these typically consist of technical discussions however groups usually have child groups which focus on niche things. For example;

Project	**Channels**	**Description**	**Size**
Holochain	Holo	official channel	Circa 13,019 Members
	Holotroopers	Unofficial trading channel (conversations tend to be messy)	Circa 3,343 Members
	HoloHodlers	A small group of users who are strongly vested in the project and discuss long-term possibilities.	Circa 1,750 Members

Project Aims

My Intention is to analyse the sentiment of cryptocurrency projects by opinion-mining their public telegram channels. As explained in [4] opinion mining is the process where attributes are extracted from a given set of documents to determine whether the sentiment expressed is positive, negative or neutral. I will seek to find a connection between sentiment expressed in the discussion channels and price fluctuations. Our data analytics might help us answer the question of what came first, hype or price action.

Aims;
Sentiment analysis and opinion mining of crypto-currency projects.

[1] "Durov's Channel". *Telegram*. Retrieved 2017-10-01.

[2] https://telegram.org/blog/200-million

Possible aims;
Measuring sensitivity / correlation of price with the sentiment of channel
Visualisation of this relationship using heat map structure.
Named entity recognition

Natural Language Processing

Natural language processing is a new and current interdisciplinary field which aims to enable computers to identify, understand and express human speech and text. "The goal of this new field is to get computers to perform useful tasks involving human language, tasks like enabling human-machine communication, improving human-human communication, or simply doing useful pro- cessing of text or speech."[4] (Jurafsky, 2013)

Sentiment Analysis

The objective of sentiment analysis is to convert unstructured text data into meaningful data for analysis, to measure customer's opinion, product reviews, feedback, to provide search facility, sentimental analysis and entity modelling to support fact-based decision making. Models uncover the emotional tone behind a series of words, which is then extrapolated to understand the sentiment and opinions expressed. [1].

The scope of this can be either document-level or sentence level. Sentiment analysis can be essentially presented as a classification problem where our class labels are the polarity of a sentence (positive, neutral, negative). Recent trends have found that we process positive and negative opinions in parallel [9], this has resulted in more applications displaying the degree/strength of positivity/negativity a piece of text is, this is illustrated in figure 1.1. There are various approaches to sentiment analysis;

Frequency of Features – *this involves comparing the word count of positive words against negative.*

Machine learning – *this involves the use of statistical algorithms to train a classifier on a known/tagged dataset.*

Lexicon method - *is essentially a list of words which have been tagged according to their semantical orientation (Positive, Negative, Neutral)*

Rule-based approach – *identifies words deemed predictive of opinion, then classifies the text/document according to the count of positive/negative words.*

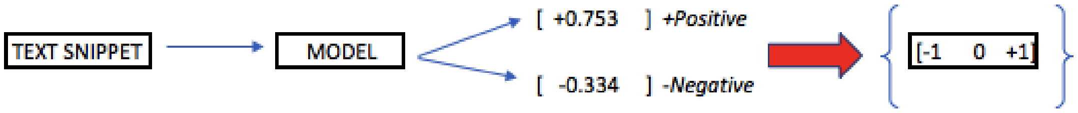


Figure 1.1

NLP Classifiers

TF.IDF

TF-IDF is a useful measure of the importance of a term in a given document collection [3]. It is computed by the dot product of the term frequency and the inverse document frequency. The term frequency is the number of times a term occurs in a given document, although this is useful, if used alone, it gives an unnecessary high weight to common words such as 'the', therefore combining with the inverse document frequency which measures the frequency of words in the entire document set helps to offset this imbalance. Words that often occur throughout the document collection will be weighted less than rarer words. Combining both statistical measures helps reflect the importance of a word in a document collection [3].

Logistic Regression

Logistic regression is a discriminative classifier. This essentially means that through a training set the model learns what distinguishes different classes [1]. the model is given a vector of features in the form of [X_i, X_{ii}, X_{iii}] and through training of labelled data, the model learns which features best differentiate classes. The model will produce weights [**w**] which rank the term in importance in helping differentiate classes. for example, for a sentiment classifier, the term "good" would have a higher weight than the term "horrible" as it helps to differentiate between positive and negative text. There is also a bias [**b**] also known as the intercept which is a constant which is added to the equation.

These inputs are multiplied by their respective weights, and the bias is added to the equation.

$Z = w.x + b$

Z is a real number which can have a value between $-\infty$ to $+\infty$. This is counter-intuitive for determining probability, so it is parsed through the sigmoid function, also known as the logistic function. The sigmoid function essentially takes any real number and maps it to a value between 0 and 1 [1]. This is required for computing probability. The model outputs a decision boundary which is also a value between 0 and 1. This can be considered as a cut-off point where any score outputted by the model which is greater than or equal to the decision boundary is classified as the target variable (positive), and anything below is classified as negative. Logistic regression is occasionally described as the bases of neural networks (NN) since NN can be described as logistic regressions layered on top of each other with an optimisation function at the final layer [1].

Naïve Bayes

Naïve Bayes is a classifier which is part of the machine learning family. It is considered a generative classifier because it aims to generate a representation of the classes and sees which model better fits the testing data [1]. It is deemed a simplistic classifier as it is centred on Bayes theorem that terms are conditionally independent on each other given a class[6].

This assumption has similarities to the ‘bag of words’ model where the arrangement of the terms is not relevant but rather the occurrence (frequency) of these terms [5]. Naïve Bayes is a general term used to describe the conditional independence of features in the model. There are various flavours of naïve Bayes classifiers which are used in text classification. Multinomial naïve Bayes described the distribution of the features. Such that [P(f|c)] follows a multinomial distribution. This can be developed further as intuitively for sentiment analysis the occurrence of the word is more significant than the frequency [1], for example

> D1: The movie was awesome awesome awesome

The fact that the term awesome was used in D1: tells us a lot about the polarity of the text/sentence, however the detail that the frequency is three times does not add much more. Adopting a binarized (Boolean) multinomial naïve Bayes approach reduces the frequency of words to one for each document. This proves to increase the model performance in sentiment analysis as noted in [2]. However, this improvement is only marginal.

As mentioned in [2], logistic regression seems to outperform NB models with large snippets of text. However, as noted in [7] NB models have been proven to perform relatively well with short text snippets as well as using the unigram model (bag of words).

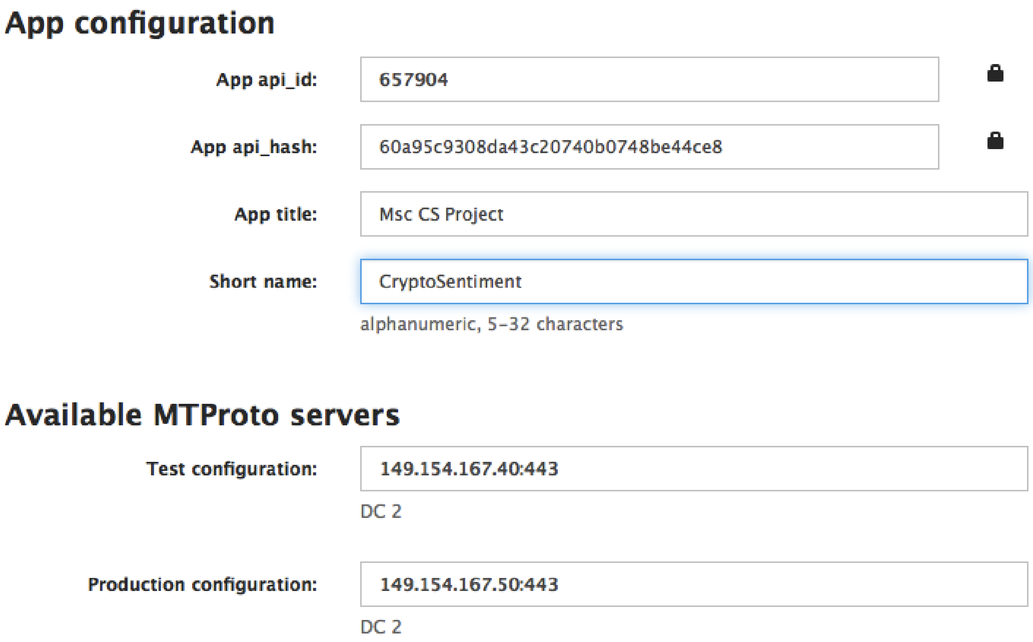
Project Outline

Data Source

Telegram
Telegram has a public API which can be used to access public chats. There are many open source scripts and packages which can be used to retrieve messages from public chats.

> *https://my.telegram.org/auth*

Once filled out you will receive an api_id and api_hash parameters which will be used to access telegram. Figure 2.1, displays the result of this process.



App configuration

App api_id:	657904
App api_hash:	60a95c9308da43c20740b0748be44ce8
App title:	Msc CS Project
Short name:	CryptoSentiment alphanumeric, 5-32 characters

Available MTProto servers

Test configuration:	149.154.167.40:443 DC 2
Production configuration:	149.154.167.50:443 DC 2

Figure 2.1

The account you register for the API must be a member of the group you wish to scrape. Once connected message objects can be extracted from the chat which contains;

Username
Bio
Join Date
Message

This will be parsed through the data pre-processing stage before being tagged and stored in a SQL database. Figure 2.2, displays the structure of this system. In short, the system will access telegram via the public api. The target channel will be identified in the code by the identifier, the Get_Methods function will extract messages from the objects the channel objects. Data wrangling will then begin to normalise the text. Following this, the tokens will be stored in a SQL DB which will feed the tokens to our sentiment model to be classified. The results of this will be fed back to the DB and subsequently parsed to the visualisation tool/ dashboard.

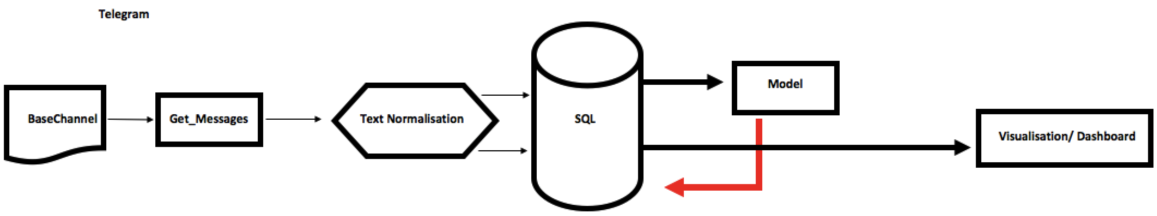


Figure 2.2

Data ETL

I am using the Telegram API to retrieve channel messages. The textual data must first be pre-processed before it can be analysed; this pre-processing is often referred to as data wrangling. This data wrangling is an integral part of natural language processing as the data must first be standardised by lemmatization, separated into single terms by performing tokenisation and cleaned by removing stop words and punctuation. Tokenisation is the process where "Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens"(Manning, Raghavan and Schütze, 2018).

Lemmatization is where words are reduced to their dictionary base terms [10]; this allows extrapolation from a first principle since words with common base terms often mean the same thing. Removal of stop words and punctuation is performed to remove the noise from a document as these words tend not to have any statistical power in the model and just increase dimensionality.

Technology

I will be using python for most of this project as it is a highly efficient scripting language. There are a wealth of packages available to perform the various tasks needed for sentiment analysis. For the data wrangling element, where I would need to perform tokenisation, stemming and tagging, examples of packages are;

NLTK – is an open source platform for natural language processing using python. "It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum." (NLTK.org, 2019)

Vader – short for 'Valence Aware Dictionary and sentiment Reasoner.' Is a python library built for sentiment analysis of social media texts. Vader is specialised for social media text as its lexicon incorporates slang words and colloquial terms. It is fully open source under MIT Licence, the team behind VADER uses amazon's manual classification service (Mechanical Turk) for the classification of their lexicon.

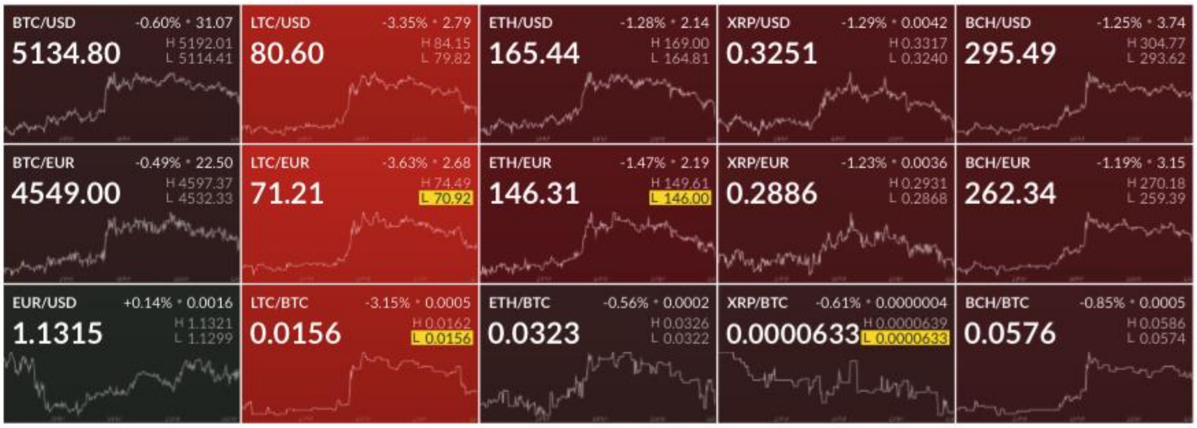
SentiStrength – is a popular peer reviewed sentiment classifier, which expressed the strength of a texts positivity and negativity.

Lexicons – are essentially a list of words which have been tagged according to their sentimental orientation. There are many open source lexicons which have been curated over time. Many have had proven successes; however, sentiments lexicons vary according to the text used. For example, VADER has been especially attuned to social media texts, since it understands colloquial phrases, the use of emoji and slang terms. Therefore, an iterative method would be best in deciding which lexicon to implement in the project. Below are a few examples of open source lexicons which can be used

- The University of Illinois at Chicago – ‘Opinion Lexicon’ which consists of ~6800 English words[3]
- Princeton University – WordNet which consists of ~155 327 words[4]
- SentiWordNet (Included in NLTK)[5]
- Vader[6]
- SentiStrength[7]

Visualisation

Explanatory data analysis will be performed to plot price against sentiment to determine if there is a correlation. This can be plotted as a treemap with projects as the tiles, sentiment shaded from green to red and an arrow detailing price movements. Below is an example from Finviz[8] where crypto assets trading pairs are shaded according to the price which is similar to the desired outcome outlined above.



(Finviz, 2019)

[3] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

[4] https://wordnet.princeton.edu

[5] http://www.nltk.org/howto/sentiwordnet.html
[6] https://github.com/cjhutto/vaderSentiment
[7] http://sentistrength.wlv.ac.uk/
[8] https://finviz.com/crypto.ashx

Proposed Project Schedule

Planning is integral for all projects. It enables the measuring of progress and …. The tasks have been outlined below at a high level with predicted schedules.

Phases	Task	Targets
April – May	Extensive research into the tools used, along with experimentation with the lexicons chosen.	Solid understanding and selection of; - Libraries - Lexicons - Tools
May – June	Data sources – building a data pipeline to access telegram groups and feedback results into a DB/cloud cold storage (s3 bucket)	Building a functioning pipeline where data is flowing from telegram to the storage solution.
June – July	Testing the whole process with an out-the-box solution, to ensure that there is desired functionality end-to-end.	following on the pipeline, by pre-processing the data and prediction sentiment as the data flows through the solution.
July - August	Further development of test solution to refine data pre-processing, prediction, aggregation and optimisation. I expect this process to be interactive with support from Dr Alessandro Provetti	Refining the solution by tuning the algorithms to ensure the best predictive power. During this stage, the report will be written along with development.
August - September	Thorough project evaluation – which will involve measuring the effectiveness of the implementation. Also, if time permits, along with a robust aggregation of sentiment, for this to be extrapolated to understand if there is a correlation with price action.	Performing project evaluation, finalising the report. this will be the final development on the project and if time permits some explanatory data analysis can be performed to plot price against sentiment to determine if there is a correlation.

References:

[1] Jurafsky, D. (2013). Speech and Language Processing. Harlow: Pearson.

[2] Wang, C. Manning. 2012. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. In Proceedings of ACL 2012.

[3] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. Journal of Documentation, 60(5), pp.503-520.

[4] Bing, L.: Sentiment Analysis and Opinion Mining (Synthesis Lectures on Human Language Technologies). Morgan & Claypool Publishers (May 23, 2012) ISBN-13: 978-1608458844

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Sch¨utze. Introduction to Information Retrieval . 2008.

[6] Prabowo, R. and Thelwall, M. (2009). Sentiment analysis: A combined approach. Journal of Informetrics, 3(2), pp.143-157.

[7] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques,” Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86, 2002.

[8] "Why Telegram's security flaws may put Iran's journalists at risk". Committee to Protect Journalists. 31 May 2016. Retrieved 20 July 2016.

[9] Berrios, R., Totterdell, P. and Kellett, S. (2015). Eliciting mixed emotions: a meta-analysis comparing models, types, and measures. Frontiers in Psychology, 6.

[10] Manning, C., Raghavan, P. and Schütze, H. (2018). Introduction to information retrieval. Cambridge: Cambridge University Press.

[11] NLTK.org. (2019). Natural Language Toolkit – NLTK 3.4 documentation. [online] Available at: http://www.nltk.org/ [Accessed 15 Apr. 2019].

[12] Narayanan, A., Bonneau, J., Felten, E., Miller, A. and Goldfeder, S. (2016). Bitcoin and cryptocurrency technologies: A Comprehensive Introduction. Princeton University Press.