universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Exploiting Short-Text Topic Modelling in Application of Cryptocurrency Price Prediction“

verfasst von / submitted by

Markus Tretzmüller BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2019 / Vienna, 2019

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 935
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Medieninformatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr.Wilfried Gansterer, M.Sc.

Contents

Abstract

As the trend in financial industries to include alternative data into investment decision making continues, social sentiment analysis has become an active field of research. The challenge is to quantify public announcements which have significant impact on financial assets in order to predict price changes. In this study a generic technique, that combines sentiment analysis with short-text topic modelling called *Sentiment Biterm Topic Model* (sBTM) is proposed. sBTM expands mood time series with topic dimensions. It performs sentiment analysis on latent topic portions, extracted by the *Biterm Topic Model*, which is specifically designed to classify short-texts such as tweets. A long-ranging collection of tweets is leveraged to investigate if Twitter activities influence cryptocurrency price formation. Multiple baseline forecasts are explored and finally it is tested whether topic-specific analysis can enhance prediction accuracy. It is shown that the topic based approach is more effective than its non-topic counterpart.

Zusammenfassung

Während sich der Trend um alternative Daten in der Finanzindustrie fortsetzt, wurde die Messung öffentlicher Meinungen zu einem populären Forschungsgebiet. Die Herausforderung besteht darin, Meinungen, die signifikanten Einfluss auf die Preisentwicklung von Finanzwerten haben, zu quantifizieren. In dieser Arbeit wird eine generische Technik namens *Sentiment Biterm Topic Model* (sBTM) vorgestellt, die Sentiment-Analyse mit Kurztext-Themen-Modellierung vereint. sBTM erweitert Meinungen um eine Themen-Dimension. Dabei wird eine Sentiment-Analyse an der Wahrscheinlichkeitsverteilung der Themen vorgenommen. Diese Wahrscheinlichkeitsverteilung wird mit dem *Biterm Topic Model*, welches speziell für Kurztexte konzipiert ist, extrahiert. Ein Datensatz über eine Zeitspanne von 1.264 Tagen wird verwendet um herauszufinden ob Twitteraktivitäten Kryptowährungen beinflussen. Es werden mehrere Benchmark-Modelle erstellt um zu testen ob anhand Themen-spezifischer Analyse Kryptowährungen besser vorausgesagt werden können. Es wird bestätigt, dass der Themen-spezifische Ansatz besser funktioniert als das herkömmliche Pendant.

Summary

The value of cryptocurrencies is determined by people's believe in the currency itself, which explains why their price is sensitive to public announcements, news and opinions. In this work Twitter data is used to extract public mood in order to predict future price changes.

The debate whether news or public opinion can be used to predict financial assets has a long history that is summarized in Ch. 2. The influential work of [1] suggested that markets work efficiently. The efficient market theory states, that it is impossible to use news to predict stocks. This dogma is questioned by behavioral economics, where emotional behavior of market participant is supposed. The work of nobel prize winner Daniel Kahneman [2] provides theoretic foundation why investors' emotions play a key role in assets' price formation. [3] suggested a crude measure to quantify investors' emotions, known as lexical sentiment analysis. This method has since been widely adopted, as it is easy to use and easy to reproduce by other researchers. In contrast, supervised machine learning methods [4] involve subjectivity as data is usually labeled manually. [5] shows that Twitter is a useful source to measure public mood and further demonstrates that stocks are influenced by it. Traditionally all documents in a corpus are treated with equal importance, although some topics are intuitively of higher importance than others. [6] combines topic modelling with sentiment analysis to retrieve a finer grained measure of Twitter mood. While he successfully demonstrates that topic-specific sentiment analysis outperforms a non-topic approach, the topic model used is suboptimal in classifying short-texts such as tweets.

The theoretic foundation of probabilistic topic modelling is given in Ch. 3. A topic model especially designed for short-text classification called Biterm Topic Model is introduced. The BTM captures word co-occurrence within a corpus. The strength of BTM is depicted in Tab. 2, where the BTM outperforms the conventional LDA with respect to the coherence score defined by [7]. A novel approach to combine BTM with sentiment analysis called *Sentiment Biterm Topic Model* (sBTM) is formulated in Ch. 4. The sBTM works on timestamped documents and generates multiple time series of topic-specific mood. Each time series corresponds to one topic.

In this study's practical part (Ch. 5) nearly 600,000 tweets over a time period of 1,264 days are gathered. Experiments are conducted on the currencies *Bit-*

coin, *Ethereum*, *Litecoin* and *Ripple*. It is explored whether past prices, Twitter activity (number of tweets per week), topic-specific Twitter activity (number of tweets per topic per week), mood or topic-specific mood helps predicting abnormal weekly returns. Significant autocorrelation is inspected on Ethereum and Litecoin returns, yielding that past price has predictive power. It is shown that Twitter activity effects Litecoin and Ripple. For Bitcoin it is the other way round, price change effects Twitter activity. This proves that media attention and cryptocurrency pricing is linked, but the causal direction varies across currencies. Although overall Twitter activity has no impact on price, activity of certain topics has. This underpins the notion of topic-specific analysis. Further, it is shown that negative, positive, uncertain and modal mood influences Bitcoin price formation. Surprisingly, altcoins are hardly influenced by Twitter mood, neither does topic-specific mood influence altcoins. It is argued that this is due to Bitcoin's dominance in cryptocurrency price formation. Mood regarding particular altcoins is overruled by investors sentiment about Bitcoin. For Bitcoin, the mood of 15 out of 20 topics is influential. Coherently, those 5 irrelevant topics are comprised of typical spam words (like, follow, retweet, ...). This demonstrates that sBTM can be effectively used to separate important from minor important topics.

Finally, a VAR-model is used to measure predictive performance. Up-down accuracy is calculated across currencies and across multiple time lags. While activity (55%), topic-specific activity (56%) and mood (56%) can hardly improve the most basic forecast based on historic price (55%), topic-specific mood does (59%). As stated, mood has most influence on Bitcoin. Weekly price change is predicted with 62% accuracy on average. The topic extended mood works even better with an accuracy of 66% on average, underpinning the effectiveness of sBTM.

1 Introduction

1.1 Problem Statement

The efficient market hypothesis states that financial markets incorporate all existing information at any point of time. Traders act as *homo oeconomicus*, that behaves purely rational and seeks to maximize profits. In contradiction behavioral finance emerged which has underpinned the role of emotions in financial investment. As a consequence, market makers try to incorporate public mood into their investment decision process. The rise of social networks made it possible to measure public mood. Measuring investor and social mood has become a popular field of research. Social media, such as blogs, forums, and social networks have become a primary data source as they are ubiquitous platforms for social networking and content sharing. The microblogging platform Twitter plays a predominant role when it comes to social, political and economic events. Ordinary people but more importantly people of public interest and news wires provide information on this channel, producing massive amounts of data every day.

Meanwhile virtual currencies called cryptocurrencies emerged. Their value is mostly determined by the believe of buyers and sellers into the currency itself. The more people believe in a currency, the more value it has. This explains why cryptocurrencies are extremely sensitive to public mood. The objective of this study is to exploit Twitter data to predict cryptocurrency returns.

1.2 Terminology

Twitter is by definition [8] a social network that provides, *a*) a virtual space in which users can make and present their own profile, *b*) the possibility to create a network with other users and *c*) the possibility to see the network connections of other participants. Twitter users are people of public interest, journalists, news agencies and private persons who post and share content especially about current events. A post is called *tweet*, which is at maximum 280 characters long. Sharing ones content is called *retweeting*. Connections on Twitter are unidirectional and made by *following* a user. Twitter provides tools to annotate content with so called *hashtags* and *cachtags*. While hashtags are designed for overall categorization, cashtags are ment to denote specific assets and financial instruments. For instance a tweet "Bitcoin is a bubble? #crypto $btc $eth" belongs to the topic of cryptocurrencies expressed by the hashtag *#cryto* and corresponds to the assets *Bitcoin* and *Ethereum* expressed by the cashtags *$btc*

$eth.

A *cryptocurrency* is a digital currency that is based on a distributed ledger. It can be traded on various exchanges against legal currencies or other cryptocurrencies. The most popular and most traded[1] cryptocurrency is called *Bitcoin*. Alternatives to Bitcoin are called *altcoins*. Their price is highly correlated with Bitcoin [9], suggesting that Bitcoin is a valid benchmark index for altcoins. Therefore, the *abnormal* or *adjusted return* r_{adj} of an altcoin is defined as follows,

$$r_{adj} = r_{alt} - r_{btc} \tag{1}$$

, where r_{btc} denotes the return of Bitcoin and r_{alt} of an altcoin. For Bitcoin the abnormal return is simply its return, because no valid Bitcoin-benchmark exists at the time of inspection. When further speaking about returns, adjusted returns a meant.

A *corpus* denotes a collections of documents. In this study the expression *document* and *tweet* is used interchangeable. In the domain of natural language processing a word is commonly defined as a delimited string of characters in a document. A "normalized" word, a word which represents all its variations (case, spelling, morphology, singular or plural, etc.) is denoted as *term*. An instance of a term occurring in a document is a *token*. For example, after removing decimals and punctuation, which is a popular preprocessing technique, the document "I bought 2 bitcoins. Bitcoin will rise." contains six words {I, bought, bitcoins, Bitcoin, will, rise}. It contains five terms {I, buy, bitcoin, will, rise} and six tokens {I, buy, bitcoin, bitcoin, will, rise}. In this study *word* and *term* are often used interchangeable. It will be clear from the context, when they are used distinctive.

Sentiment analysis automatically extracts structured, subjective information from textual content. The expression *sentiment* is a wage attitude, thought, or judgment, which can be expressed in arbitrary levels of intensity such as +1 for positive and -1 for negative or as a 1-5 stars system used by popular review sites. In a multidimensional setting various sentiment dimensions are quantified. Opinion mining or aspect based sentiment analysis extends sentiment analysis by extracting a view, judgment, or appraisal formed in the mind about a particular

[1] `https://coinmarketcap.com/de/` - 4.03.2019

matter. More formally an *opinion* is a quintuple,

$$(e, a_e, h, t, s_{a_e ht}), \tag{2}$$

where e is an entity, a_e is an aspect of e, h is the opinion holder, t denotes the time when the opinion is expressed and $s_{a_e ht}$ is the sentiment on aspect a_e of entity e from h at time t. For example Bloomberg tweeted "Bitcoin is worth less than the cost to mine it, JPMorgan says" on Jannuary the 25th, 2019. Then *Bitcoin* is the entity, *Mining* is an aspect of that entity and the sentiment referring to that aspect is negative (mining cost is too high). The opinion holder is JPMorgan and t is the time when JPMorgan announced their opinion (not necessarily the time when the tweet was published).

While *sentiment* and *mood* are usually used as synonyms, in this study they are used distinctive. It is common in sentiment analysis and opinion mining to aggregate the sentiment of multiple documents. Since the Merriam-Webster[2] defines mood as "predominant emotion", it is argued that sentiment refers to a single document's sentiment and mood to the aggregated, overall sentiment of many documents.

1.3 Synopsis

In Ch. 2 an overview of related work is provided, covering news trading, sentiment analysis and topic modelling. The section dealing with sentiment analysis is focused on sentiment analysis in finance. Next, it is summarized how topic models are used in price prediction tasks. Existing approaches that combine sentiment analysis and topic modelling are introduced.

Chapter 3 provides the theoretic background. Gibbs Sampling is explained and further demonstrated on the topic models LDA and BTM. Both models are introduced in detail. At first the generative process is specified, next the sampling procedure is given and lastly the algorithm is explained. Finally, the VAR-framework is introduced. It is shown how VAR is used to model multivariate time series, to perform predictions and to analyze Granger-causality.

In Chapter 4 the sBTM is formalized. The high-level concept is depicted, the sampling procedure and the algorithm is explained. Two variants of the sBTM are introduced, an univariate and a multivariate one.

[2] https://www.merriam-webster.com/dictionary/mood (14.04.2019)

In Ch. 5 it is explained how the data is gathered and filtered to extract high-quality Twitter data over multiple currencies. A comparison of LDA and BTM is conducted on the data. Next, multiple baseline forecasts are explored. Topic-specific mood extraction is accomplished via sBTM and a VAR framework is used to perform price prediction.

In Ch. 6 the insights according to cryptocurrency price formation are recapped and strengths and weaknesses of sBTM are formulated. Some interesting use-cases for sBTM along with promising improvements are outlined.

2 Related Work

2.1 News Trading

News trading denotes a trading strategy where financial news are interpreted as trading signals. The underlying assumption that financial news reveal predictive insight on future stock price movements is controversially debated. With respect to the efficient market hypothesis this assumption does not hold. The strong efficient market hypothesis is based on a "fair game"[1] where *a*) no trading fees exist, *b*) all information is freely available to all participants and *c*) all participants agree on the impact of information on corresponding financial instruments. It states that a financial time series j fully reflect its value at each time t, formally expressed as

$$f(r_{j,t+1}|\Phi_t) = f(r_{j,t+1}) \tag{3}$$

where r denotes the return and Φ all relevant information. This equation asserts that news do not yield any predictive power since the information is already incorporated into the price. The efficient market hypothesis was widely accepted since [10] inspected the adjustment of prices to new information. They assumed that stock splits indicate new public information and modelled price changes with an least square regression model. If a stock split is associated with abnormal behavior, this would be reflected in the estimated regression residuals for the months surrounding the split. However, the residuals increased prior to stock splits indicating that the information at the time of the split was already incorporated into the price. [11] showed that a set of 115 mutual funds in the period from 1955 to 1964 could not acquire excess return. Assuming that fund managers have access to exclusive information he argues that even insider knowledge has no predictive power.

In contradiction to the efficient market hypothesis behavioral finance emerged. Anomalies such as the January effect or the day-of-the-week effect provide evidence that price is rather driven by irrational behavior of share holders. Feedback models [12] describe the importance of information diffusion and share holders emotional bias. In contradiction to the believe of an *homo oeconomicus* as a fundamental condition for an efficient market, [2] showed that people are irrationally biased in decision making. Subjects were asked to guess the occupations of a person whose characteristics were given. They guessed the occupation that was closest to the given characteristics, neglecting the statistical probability for that occupation. [13] examined a extraordinary 330% price jump of an biotechnology company in one day. The jump was caused by an news report of the New York Times about a breakthrough in cancer research associated with that company. In fact the article did not reveal new information. All relevant facts had been published on a scientific journal months before. While the theoretic ground of news trading remains ambivalent there is little doubt that news and financial markets are interlinked.

2.2 Sentiment Analysis

Sentiment analysis is one of the most trending research areas in natural language processing this century. The aim of sentiment analysis is to extract subjective information from texts. Due to the variation of texts in language, length, style, format and context a lot of different methods have arisen. The most prominent ones are lexical and supervised machine learning methods. In what follows, a selective overview of these methods with application in accounting and finance is provided.

In the highly influential paper [3] the Harvard IV-4 Dictionary is used to measure sentiment from *Wall Street Journal's* daily stock news. They measured the overall mood with an accumulation window of 30 days prior, until 3 days after an earning announcement. They found that negative mood is related to both lower stock returns and to higher stock market volatility. The impact of journalistic pessimism is highest when focused on firm fundamentals. Following [3], much investigation in financial sentiment analysis is based on the Harvard dictionary. [14] examines initial public offering[3] prospects on the basis of the Harward IV-4 negative and positive dictionary. They found out that positive sentiment is associated with lower returns after the first trading day and smaller changes in the offer price revision. [15] criticize the use of general purpose

[3] An Initial Public Offering (IPO) is a stock market launch

dictionaries in the financial domain. They inspected that about 74% of the Harvard negative word count typically does not have negative meaning in a financial context. In more recent studies the Loughran and McDonald dictionary (LM) became predominant [16].

As an alternative to lexical approaches supervised machine learning is used to classify the sentiment of documents. The influential work from [4] applied Naive Bayes in financial sentiment analysis. They selected 1000 messages from online message boards to train their algorithm. The found out that those messages have impact on stock return volatility. Higher disagreements among the postings is linked with higher subsequent trading volume. [17] directly labeled messages according to subsequent market fluctuation within an Bayesian framework. Based on this language model forthcoming trends in the stock prices can be predicted. Similarly, using a multivariate regression, [18] formulated an word weighting scheme according to subsequent market reactions. They showed that their method reliably quantifies the sentiment of IPO prospectuses.

Social Media Sentiment: The focus of research consequently shifted from analyzing news documents to analyzing short texts from social media, especially Twitter. [19] found out that Twitter captures most real-world events. 95% of News from 8 popular news sources are covered on Twitter. 52% of events have been reported exclusively on Twitter. But tweets are intrinsically different from traditional news documents. They are shorter and have special notations like emoticons, hashtags and cashtags. Further, the authors are rather private persons than professional journalists. Much work is done to detect spammers [20], to quantify users reputation [21] and influence in the network [22] or their influence on corresponding financial time series [23]. The amount of social media content is enormous and continually increasing. Hence, it is more representative of general mood and public opinion than news articles written by journalists. In a highly influential work [5] used a multidimensional Twitter mood to predict the Dow Jones Industrial Average (DJIA). It is demonstrated that mood dimensions correspond to public events (election, thanksgiving) and that DJIA daily price change is Granger-caused by various mood dimensions (subjectivity, calmness, happiness). Further, a Self-organizing Fuzzy Neural Network predicts DJIA daily up-down movement with an accuracy of 87,6% for a 20 day time window. [24] used a manually labeled corpus in combination with a Naive Bayes for sentiment extraction of 250,000 stock-related tweets to examine their relation with market returns, trading volume, and volatility. They found out that market features have higher impact on tweet features than the other way round.

They further inspected that financial experts gain more retweets and followers. However, the analysis of individual tweets shows that higher quality information is not retweeted more frequently.

2.3 Topic Modelling

Topic models are methods to discover topics in a collection of documents, where a topic is traditionally viewed as as a mixture of words. One initial approach in topic modelling is Latent Semantic Analysis (LSA) where a weighted term-document matrix is reduced using Singular Value Decomposition (SVD). Search engines early adopted LSA to provide thematic relevant documents based on search queries. LSA evolved to a Probabilistic Latent Semantic Analysis (pLSA) [25], which uses Maximum Aposteriori over SVD to extract latent topics. [26] used a modified version of pLSA to extract latent sentiment factors from blog entries. Further an autoregressive model was proposed to leverage the extracted sentiment factors for product sales prediction. [27] introduced a Bayesian version of pLSA called Latent Dirichlet Allocation (LDA), which uses Dirichlet priors for the document-topic and word-topic mixture. With the introduction of Dirichlet priors the notion of topic evolution over time was tackled successfully [28]. In a dynamic topic model, documents are timestamped and divided into discrete time slices, for example by year, month or day. By chaining together multiple LDA models for sequential time frames, where each model uses the posterior distribution of the preceding model as priors, online inference is possible. The critical assumption of [28] is that a fixed number of topics is present along the complete duration of analysis. Hierarchical LDA also called Hierarchical Dirichlet Process (HDP) [29] is an non-parametric generalization of LDA, where the number of topics is not known apriori but instead learned from the data. Using HDP to analyse topic evolution needs some linkage mechanism to detect constant, emerging and ending topics across time slices. [6] suggests such a mechanism and further extracts topic-specific mood for stock market prediction. Since the linkage mechanism provides a variable set of topics for each time frame (day), the five most active topics chains are separately tested in an autoregressive stock prediction model. Finally, the best performing topic-sentiment series gives an up-down accuracy of 61%. [6] is closest related to this study, but here a constant number of topics is assumed. Therefore, no topic-linking has to be performed. This study leverages a Twitter-optimized topic model and circumvents the manual selection of topic-mood time series to be used for prediction. Instead Granger causality is used to select relevant time

series. [30] designed a model that jointly captures topics and sentiment, by incorporating sentiment analysis into the generative process of LDA. LDA works on a document-level while this study is based on a corpus-wide model, better suited for short-text topic modelling.

Short Text Models: A constraint of LSA, pLSA, LDA and HDP is that they all work best when applied on large documents [31], as they reveal topics within a text corpus by implicitly capturing the document-level word co-occurrence patterns. A lot of research is done to overcome the sparsity problem for short-text topic modelling. An intuitive way to cope with the sparsity problem is to assume a document belongs to only one topic, as short text often does. Under this assumption [32] provided a Gibbs sampling procedure that captures latent topics. However, [33] showed that the strong assumption of short-text having only one topic decreases performance. Another way of circumventing the sparsity problem is to extend a short text with background information, where a larger pseudo document is created. [31] grouped tweets containing same words to large pseudo documents and trained LDA on them. [22] treated all tweets from a twitter user as one document and trained an LDA to find out the thematic influence of a twitter user. Similarly, [34] extended the generative process of the conventional LDA with an latent author distribution, as a probability distribution over topics. However, these methods are highly domain and task dependent. First of all, the aggregation of background information is challenging. For example what if a user/author does not provide further documents to incorporate? And most important, the authors are assumed to have a strong topical preference, which is not true for interdisciplinary authors/news wires. [33] proposed the Biterm Topic Model (BTM) which models the generation of word co-occurrence in the corpus and inferences topics using Gibbs sampling. The BTM outperformed the LDA, the user-aggregated LDA version and the mixture of unigrams across various data sets including a tweet corpus. [35], [36] noted that the BTM is prone to background words. Non-informative words, like stop words are treated with equal importance than more informative once. The first tackled this problem by adapting the generative process of the model, the latter applied a rule-based approach to remove background words before topic modelling is applied.

3 Methodological Basis

3.1 Lexical Sentiment Analysis

A document, a corpus or any textual unit in sentiment analysis is typically represented as a vector $d = (f_{w_i}, f_{w_{i+1}}, \ldots, f_{w_V})$, where w_i denotes a word and f_{w_i} specifies how many tokens of w_i occur in d. The critical assumption of independence is made, which means that the order, and thus direct context, of tokens in a text is irrelevant. This is a so-called bag-of-words model. Under a bag-of-words model big collections of data can easily and efficiently be summarized. Sentiment dictionaries can be used to quantify the emotional content of text. [5] demonstrated an straight-forward approach to extract multiple mood dimensions. However, their lexicon data is not publicly available. This study follows their methodology in sentiment analysis but uses the publicly available Loughran and McDonald (LM) dictionary. The LM-dictionary is designed to be used in the financial context and has become the state-of-the-art dictionary in this domain [16]. The LM-dictionary contains six different word lexicons containing either negative, positive, uncertain, litigious, modal or constraining words. In this study the sentiment of a text is simply measured by counting the words occurring in that text. Another popular way of expressing sentiment is called polarity, where the ratio between positive and negative words is calculated. Traditionally the polarity is normalized to be in between -1 and +1, where -1 indicates that a text is negative and +1 that a text is positive. In this study, the polarity of a document is the difference of positive and negative words.These crude quantitative measures do not extract an opinion (as defined in Ch. 1.2 Eq. 2) nor do they capture negotiations or irony and sarcasm, but they are efficient measures that scale to large collections. They can be easily extended to extract sentiment of multiple documents, of a complete corpus or any coherent textual unit. And most importantly, the researcher's subjectivity is avoided and no resources and domain knowledge is needed to label collections. Publicly available dictionaries make research straightforward to reproduce.

3.2 Topic Modelling

A generative topic model is based on a probabilistic sampling scheme that describes how words are generated based on latent variables. Most common, these are bag-of-words models, neglecting the order of tokens. The only relevant information is the number of tokens. Further, the generative process does not restrict words to exclusively belong to one topic. This allows topic models to capture polysemy. Polysemy denotes the fact that words can have different meanings in different context.

3.2.1 Gibbs Sampling

The challenge in generative topic modelling is posterior inference, which is the process of learning the posterior distribution of the latent variables given observed data. In LDA and BTM the posterior latent distributions cannot be computed exactly, which is why approximation techniques are used to infer the parameters. Among them is Gibbs Sampling, which is very popular due to its simplicity. Gibbs sampling is applicable in situations where a multivariate probability distribution needs to be approximated. The basic idea is to make a separate probabilistic choice for each of the dimensions, where each choice depends on the other dimensions. For example, one wants to know a K-dimensional probability distribution $p(z) = p(z_1, ..., z_K)$, where no closed form solution exists but conditional distributions are available. Then $p(z)$ can be approximated by iterative sampling, where each iteration the value of one variables is replaced by a value drawn from the conditional distribution of the remaining variables. In other words, z_k is replaced by a value drawn from the distribution $p(z_k|z_{-k})$, where z_{-k} denotes $z_1, \ldots, z_K$ but with z_k omitted. The procedure looks as follows:

1. Randomly initialize each z_k
2. For $t = 1, \ldots, T$:

   2.1 $z_1^{(t+1)} \sim p(z_1|z_2^{(t)}, z_3^{(t)}, \ldots, z_K^{(t)})$
   2.2 $z_2^{(t+1)} \sim p(z_2|z_1^{(t)}, z_3^{(t)}, \ldots, z_K^{(t)})$
   2.K $z_K^{(t+1)} \sim p(z_K|z_1^{(t)}, z_2^{(t)}, \ldots, z_{K-1}^{(t)})$

This sampling process is repeated T times, where the samples begin to converge to the true distribution. The convergence is theoretically guaranteed with an infinite number of iterations, but there is no way to find out how many iterations are exactly required to reach the true distribution.

3.2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probabilistic model, where documents are represented as random mixtures over latent topics and a topic is defined as a distribution over words.

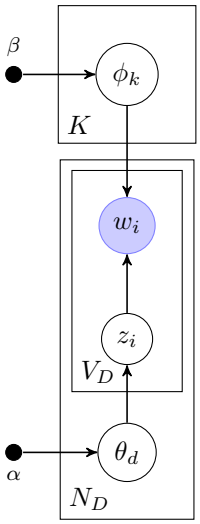

Figure 1: LDA Plate Notation: The parameters α and β are corpus-level priors. ϕ_k are topic-level variables which are sampled once per topic. θ_d are document-level variables, sampled once per document, where N_D is the total number of documents. z_i is a word-level variable which is sampled once for each word in each document, where V_D is the number of words in a document. w_i is filled because it is an observable variable, the other empty variables are not observable, hence are latent variables.

Generative Process: Supposing a corpus with N_D different documents $D = \{d_i\}_{i=1}^{N_D}$, the generative process is as follows:

1. For $k = 1, \ldots, K$:
   (a) $\phi_k \sim Dirichlet(\beta)$
2. For each $d \in D$:
   (a) $\theta_d \sim Dirichlet(\alpha)$
   (b) For each $w_i \in d$:

i. $z_i \sim Multinomial(\theta_d)$

ii. $w_i \sim Multinomial(\phi_{z_i})$

, where K is the number of topics in the collection, ϕ_k is the discrete probability distribution over words of topic k, θ_d is a document-topic distribution, z_i is the topic of word w_i[4], α and β[5] are the priors of the Dirichlet distributions. The process can be summarized as follows: First, the topic-word distribution ϕ_k is conditioned on the prior β, which is the prior believe of word-topic distributions. To generate a document multiple steps are required. First, a document-topic distribution θ_d is drawn, defining how topics are distributed in a document. Again, θ_d is conditioned on its prior α, which represents the initial believe of how document-topic distributions are expected to be. To generate tokens belonging to a document, a topic z_i conditioned on the document-topic distribution is drawn. Then, z_i is used to actually draw a token from the word-topic distribution ϕ_k. In essence, LDA captures topics within a corpus by implicitly modelling the document-level word co-occurrence patterns. Short texts, having a sparse vector representation, do not provide enough words to learn the document-level variables effectively. Hence, the estimation of z_i and θ_d is poor, inhibiting also the learning of the topic-word distributions ϕ_k. This is called the sparsity problem in topic modelling.

Sampling: The latent variables ϕ_k, θ_d and z_i are initially unknown and cannot be computed in a closed form. While conditional distributions can be derived for each of them and therefore each variable can be approximated by separately Gibbs-sampling them, the procedure can be simplified. ϕ_k and θ_d can be calculated from z_i. After integrating out ϕ_k and θ_d[6], sampling only z_i is sufficient. This procedure of *collapsing* variables by integration to apply simplified Gibbs sampling is called *collapsed* Gibbs sampling. In A.1 it is shown that,

$$p(z_i = k|z_{-i}, w) \propto (n_{-i,d|k} + \alpha_k)\frac{n_{-i,w_i|k} + \beta_{w_i}}{n_{-i,.|k} + \beta V} \tag{4}$$

, where z_{-i} denotes $z_1, \ldots, z_K$ with z_i omitted, $n_{-i,d|k}$ is the number of times words of document d excluding w_i have been assigned to topic k, $n_{-i,w_i|k}$ denotes how many times word w_i has been assigned to topic k excluding the current assignment and $n_{-i,.|k}$ denotes how many times words excluding the current

[4] In fact, w_i is a token (instance of a word), however *word* is common in literature.

[5] They are also called hyperparameters as they are parameters of a prior, which is itself used to draw parameters.

[6] *Integrating out* ϕ_k and θ_d means, that all possible values of ϕ_k and θ_d are taken into account, which circumvents to use them as variables explicitly.

one has been assigned to a topic. The first term of the equation describes the probability of a document for a particular topic and the second term describes the probability of a topic for a particular word.

Algorithm 1 LDA

Input: $K, \alpha, \beta, w \in d \in D$

Output: $\phi_{k,w}, \theta_{d,k}$

```
1: Randomly assign topics to words
2: for each iteration do
3:     for each d ∈ D do
4:         for each w_i ∈ d do
5:             Draw z_i ~ p(z_i = k|z_{-i}, w_i)
6:             Update n_{d|k}, n_{w|k} and n_k
7:         end for
8:     end for
9: end for
10: Compute φ_{k,w} (Eq. 5) and θ_{d,k} (Eq. 6)
```

Algorithm: The implementation of an LDA Gibbs sampler is surprisingly concise (see Alg. 1). First, each token w_i gets a topic assigned randomly. Therefore, an iteration over each document and each word of that document is applied to pick a topic assignment from a multinomial distribution with α as prior. According to the topic assignment, the count variables $n_{d|k}$, $n_{w|k}$ and n_k are updated on each iteration, where $n_{d|k}$ are the number of words assigned to topic k in document d, $n_{w|k}$ are the number of times word w is assigned to topic k and n_k are number of words assigned to topic k. This initialization procedure is summarized in Line 1. Next, Gibbs sampling is applied. In each Gibbs sampling iteration all tokens in the corpus are processed. Each token gets an topic assigned, which is sampled from Eq. 4. According to that topic assignment $n_{d|k}$, $n_{w|k}$ and n_k are updated. After all Gibbs iterations are completed $\phi_{k,w}$ and $\theta_{d,k}$ can be estimated as follows,

$$\phi_{k,w} = \frac{n_{w|k} + \beta_w}{\sum_{i=1}^{V} n_{w|k} + \beta_w} \tag{5}$$

$$\theta_{d,k} = \frac{n_{d|k} + \alpha_k}{\sum_{k=1}^{K} n_{d|k} + \alpha_k}. \tag{6}$$

In many applications the estimates of $\phi_{k,w}$ and $\theta_{d,k}$ are required as they are the predictive distributions of sampling a new token from a topic, and sampling a new token in a document from a topic. But most importantly, $\theta_{d,k}$ is often used for document classification.

3.2.3 Biterm Topic Model

The Biterm Topic Model (BTM) was designed with the sparsity problem in mind. It is a generative probabilistic model, which learns topics by directly modelling the generation of *biterms* in the whole corpus, as depicted in Fig. 2. Biterms are co-occurring word-pairs. Considering a document $d = (w_1, w_2, w_3)$ with three distinct words, biterms are generated, such that,

$$d \Rightarrow \{(w_1, w_2), (w_2, w_3), (w_1, w_3)\}. \tag{7}$$

The key idea is that two frequently co-occuring words are more likely to belong to a same topic and the assumption is made that both words of a biterm share the same topic.

Generative Process: A topic is defined as a distribution over words but instead of modelling the document generation process, the biterm generation process is modeled:

1. Draw $\theta \sim Dirichlet(\alpha)$
2. For $k = 1, \ldots, K$:
   (a) $\phi_k \sim Dirichlet(\beta)$
3. For each biterm $b_i \in B$:
   (a) $z_i \sim Multinomial(\theta)$
   (b) $w_{i,1}, w_{i,2} \sim Multinomial(\phi_{z_i})$

, where K is the number of topics in the collection, ϕ_k is the discrete probability distribution over words of topic k, θ is a corpus-level topic distribution, z_i is the topic of biterm $b_i = (w_{i,1}, w_{i,2})$. α and β are single valued priors of a Dirichlet distribution. The process can be summarized as follows: A corpus-level topic distribution θ conditioned on the prior α is drawn. Then, a topic-word distribution ϕ_k conditioned on β is drawn for each topic k. For each biterm in the corpus a topic assignment z_i is drawn conditioned on the topic-word distribution θ. According to the topic distribution of z_i, a biterm is drawn.

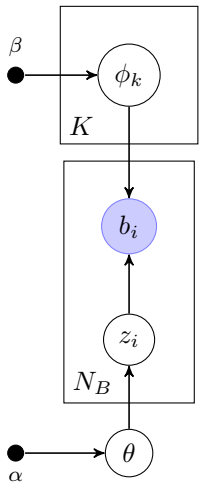

Figure 2: BTM Plate Notation: The parameters α and β are corpus-level priors and ϕ_k are topic-level variables, z_i and b_i are sampled N_B times, once for each biterm occurring in the corpus. Opposed to LDA, θ is a corpus-level topic distribution, which is favourable in classifying short-texts.

Sampling: Similar to LDA, the topic distribution θ, the topic-word distributions ϕ_k and the topic assignments for each biterm z_i need to be inferred. While an Gibbs sampling algorithm can be derived for each of these variables, both θ and ϕ_k can (again) be calculated using just the topic assignemnts z_i. Collapsed Gibbs sampling is used by integrating out the multinomial parameters and sampling only z_i (derivation see A.2).

$$p(z_i = k|z_{-1}, B) \propto (n_{-i,k} + \alpha)\frac{(n_{-i,w_{i,1}|k} + \beta)(n_{-i,w_{i,2}|k} + \beta)}{(n_{-i,.|k} + V\beta + 1)(n_{-i,.|k} + V\beta)}, \tag{8}$$

where z_{-i} are the topic assignments for all biterms except the current biterm b_i, $n_{-i,k}$ is the number of biterms of topic k excluding b_i, $n_{-i,w_{i,1}|k}$ is how often any word excluding $w_{i,1}$ has been assigned to topic k, similarly $n_{-i,w_{i,2}|k}$ is how often any word excluding $w_{i,2}$ has been assigned to topic k, $n_{-i,.|k}$ is how often words have been assigned to topic k and V is the number of distinct words in the corpus. The first factor in the equation corresponds to the probability for a particular topic in the corpus. The second factor corresponds to the probability of a biterm belonging to that topic.

Algorithm 2 BTM

Input: K, α, β, B

Output: $\phi_{k,w}, \theta_k$

1: Randomly assign topics to biterms
2: **for each** iteration **do**
3: **for each** $b_i = (w_{i,1}, w_{i,2}) \in B$ **do**
4: Draw $z_i \sim p(z_i = k|z_{-1}, B)$
5: Update $n_{w_{i,1}|k}$, $n_{w_{i,2}|k}$ and n_k
6: **end for**
7: **end for**
8: Compute $\phi_{k,w}$ (Eq. 10) and θ_k (Eq. 9)

Algorithm: Before actually starting the sampling process, the corpus consisting of documents need to be transformed into a set of biterms B (see Eq. 7), which can be computed in one pass over a document. Next, each biterm b_i gets a topic assigned randomly and the count variables $n_{w_{i,1}|k}$, $n_{w_{i,2}|k}$ and n_k are set up accordingly, where $n_{w_{i,1}|k}$ are the number of times word $w_{i,1}$ has been assigned to topic k, $n_{w_{i,1}|k}$ are the number of times word $w_{i,2}$ has been assigned to topic k and n_k are number of words assigned to topic k (see Algorithm 2, Line 1). In each Gibbs sampling iterations all biterms are passed, where each biterm gets a topic assignment sampled from Eq. 8. Finally $\phi_{k,w}$ and θ_k are calculated as follows,

$$\phi_{k,w} = \frac{n_{w|k} + \beta}{n_{\cdot|k} + V\beta}, \tag{9}$$

$$\theta_k = \frac{n_k + \alpha}{N_B + K\alpha}. \tag{10}$$

As the BTM does not generate a document-topic distribution, $p(z|d)$ has to be calculated in additional steps. A document is defined as $\{b_i^{(d)}\}_{i=1}^{N_d}$, where N_d is the number of biterms contained in document d. The rules of conditional probability state that,

$$p(z|d) = \sum_{i=1}^{N_d} p(z, b_i^{(d)}|d) = \sum_{i=1}^{N_d} p(z|b_i^{(d)}, d)p(b_i^{(d)}|d). \tag{11}$$

It is assumed that a biterm's topic is conditionally independent from d, leading to,

$$p(z|d) = \sum_{i=1}^{N_d} p(z|b_i^{(d)})p(b_i^{(d)}|d). \tag{12}$$

Based on the parameters learned by Gibbs sampling,

$$p(z = k|b_i^{(d)}) = \frac{\theta_k \phi_{k,w_{i,1}^{(d)}} \phi_{k,w_{i,2}^{(d)}}}{\sum_{k'} \theta_{k'} \phi_{k',w_{i,1}^{(d)}} \phi_{k',w_{i,2}^{(d)}}}. \tag{13}$$

While the original paper of the BTM suggests an estimation for $P(b_i^{(d)}|d)$[7], practically it is assumed to be uniform and therefore it is neglected. The final topic distribution $p(z|d) = \sum_{i=1}^{N_d} p(z = k|b_i^{(d)})$.

[7] [33] estimates $P(b_i^{(d)}|d)$ as the frequency of b_i in d divided by the total frequency of b_i

3.3 Vector Autoregression

Autoregressive models deal with time series data, where each observation of a variable, usually denoted as y_t of any point in time $t = 1, \ldots, T$ is given. A *data generating process* or more specifically a *time-series process* is a stochastic process and therefore a statistical, constructed entity. The basic idea of time-series analysis is that the observed time series is a (partial) realization of such a stochastic process. An autoregressive process assumes that a time series is linearly dependent on its past values. Autoregressive models are a popular choice for explorative analysis, structural analysis and for time series prediction. In a one dimensional setting, an autoregressive model (AR) is denoted as follows,

$$y_t = \beta_0 + \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \epsilon_t, \tag{14}$$

where $(y_{t-1}, \ldots, y_{t-p})$ are lagged values of the dependent variable y_t, which are called regressors. $(\beta_0, \ldots, \beta_p)$ are the model parameters, where β_0 is called intercept. p is the maximum of lagged values incorporated into the model, which defines the *order* of an autoregressive model. The error terms ϵ_t are also called residuals or innovations. The errors are the difference between the expected value of the data generating process and its realization. In other words, they express the random portion of the time series process. The multivariate extension, namely the vector autoregressive model (VAR) incorporates $m = 1, \ldots, M$ time series in the form,

$$\begin{aligned} y_t^{(m)} = \beta_1^{(m)} + \beta_{1,1}^{(m)} y_{1,t-1} + \cdots + \beta_{M,1}^{(m)} y_{1,t-1} + \\ + \cdots + \beta_{1,p}^{(m)} y_{1,t-p} + \cdots + \beta_{M,p}^{(m)} y_{M,t-p} + \epsilon_t. \end{aligned} \tag{15}$$

Conceptually, multiple lagged time series are chained to regress each of them. The regressors are shared among them but parameters are different. The time series are assumed to be stable. A stable time series is one whose statistical characteristics such as mean, variance, autocorrelation, are constant over time. More precisely, the errors are expected to have the following properties:

1. $E(\epsilon_t) = 0$, where ϵ is expected to be normally distributed.
2. $E(\epsilon_t \epsilon_s^T) = 0, s \neq t$, which states that there is no correlation over time (autocorrelation) of individual error terms.
3. The contemporaneous covariance matrix $E(\epsilon_t \epsilon_t^T)$ of error terms is positive semidefinite and all elements off the main diagonal are expected to be zero indicating that errors across time series are uncorrelated.

Usually financial data, such as stock prices do not fulfill these properties. One common technique to stabilize a non-stationary time series is to take the first difference which is,

$$\Delta y_t = y_t - y_{t-1}. \tag{16}$$

Intuitively, a VAR-model can be formalized as a multivariate regression model. Therefore, a more compact notation is used:

$y^{(m)} = (y_1^{(m)}, \ldots, y_T^{(m)})'$, $(T \times 1)$

$y_t = (y_t^{(1)}, \ldots, y_t^{(M)})'$, $(M \times 1)$

$x_{t-1} = (1, y_{t-1}, \ldots, y_{t-p})$, $(1 \times (Mp+1))$

$X = (x_0, \ldots, x_T)$, $(T \times (Mp+1))$

$\beta_m = (\beta_0^{(m)}, \beta_{1,1}^{(m)}, \ldots, \beta_{M,1}^{(m)}, \ldots, \beta_{1,p}^{(m)}, \ldots, \beta_{M,p}^{(m)})'$, $((Mp+1) \times 1)$

$\epsilon^{(m)} = (\epsilon_t^{(m)}, \ldots, \epsilon_T^{(m)})'$, $(T \times 1)$

Then, the VAR-model for m different time series can be rewritten as m distinct regression models:

$y^{(1)} = X\beta^{(1)} + \epsilon^{(1)}$,

$\ldots$

$y^{(M)} = X\beta^{(M)} + \epsilon^{(M)}$.

Since there usually is no exact solution to obtain the model parameters β_m, the "best" β_m is obtained by minimizing the error, or more precisely the *Sum of Squared Residuals* (SSR), which is defined as,

$$S(\beta^{(m)}) = ||\epsilon^{(m)}||^2 = ||y - X\beta^{(m)}||^2. \tag{17}$$

Minimizing the SSR is accomplished with an *Ordinary Least Square* estimation, where the best fit $\hat{\beta}^{(m)}$ is determined by differentiating $S(\beta^{(m)})$ with respect to $\beta^{(m)}$ and setting to zero (see A.3),

$$\hat{\beta}^{(m)} = argmin_{\beta^{(m)}} S(\beta^{(m)}) = (X'X)^{-1}X'y^{(m)}. \tag{18}$$

After retrieving $\hat{\beta}_m$ a (one-step) forecast is made as follows,

$$y_{T+1}^{(m)} = X\hat{\beta}^{(m)} \tag{19}$$

Granger Causality: Within a VAR framework the connection between multiple time series can be inspected. Granger causality determines whether one time series has a weak causal dependency on another. A time series X Granger-causes another time series Y if predictions on the value of Y based on its own past values and on the past values of X are better than predictions based merely on its own past values. Considering a bivariate ($M = 2$) VAR-model in the form of Eq. 15, the time series $y_t^{(1)}$ is Granger-causing $y_t^{(2)}$ if at least one of the elements $(\beta_{2,1}, \ldots, \beta_{2,p})$ is significantly larger than zero. The null hypothesis is that no Granger-cause exists and the alternative hypothesis states that it does:

$$\begin{aligned} &H_0 : \beta_{2,1}, \beta_{2,2}, \ldots, \beta_{2,p} = 0 \\ &H_1 : \beta_{2,i} \neq 0, \text{ for at least one of } i = (1, \ldots, p) \end{aligned} \tag{20}$$

A F-test is used to determine whether to keep or reject the null hypothesis, where the test statistic $F = \frac{MS}{MSE}$. The explained variance in the nominator called *Mean Squares* (MS) is $\frac{SSE}{DF}$, where the *Sum of Squared Error* (SSE) is $\sum_{i=t}^{T} \epsilon_t$ and the *Degree of Freedom* (DF) is $Mp - 1$. The unexplained variance in the denominator called *Mean Squared Error* (MSE) is $\frac{SSE}{EDF}$, where the *Error Degree of Freedom* (EDF) is $T - Mp$. The p-value is the probability for the test statistic to occur by chance. In this study a significance level of 0.05 is used. p-values below that level are considered to be statistically significant because their probability to occur by chance is less than 5%.

4 Sentiment Biterm Topic Model

The *Sentiment Biterm Topic Model* (sBTM) combines sentiment analysis and topic modelling, to capture topic-specific mood. With BTM as a basis, it is designed to deal with short texts such as tweets. The sBTM is based on the online version of the BTM and works on timestamped documents. It uses the topic-word distribution to extract topic-specific mood. sBTM is a two pass method [37], where topic modelling is performed before sentiment analysis is applied. As opposed to conventional, document-level approaches it learns topics and extracts sentiment on a corpus-level.

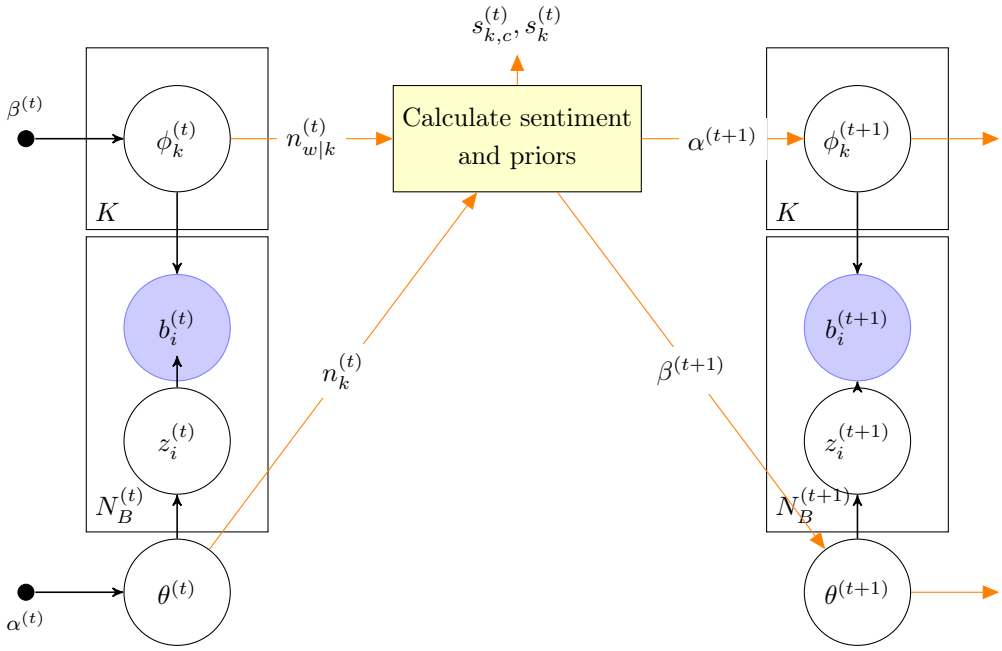

Figure 3: sBTM: For each time slice $t = (1, \ldots, T)$ a BTM is trained, where the topic-word counts $n_{w|k}^{(t)}$ and the topic counts $n_k^{(t)}$ are obtained. The topic-word counts are used to calculate univariate or multivariate topic-mood (Eq. 22, 23). $n_{w|k}^{(t)}$ and $n_k^{(t)}$ are also used to calculate the Dirichlet priors $\alpha^{(t+1)}$ and $\beta^{(t+1)}$ for the subsequent time slice (Eq. 24, 25).

4.1 Sampling

The sBTM is designed to cope with time-stamped documents, where documents are divided into rigid time slices. In essence, the sBTM fits one BTM model for one time slice and uses the posterior Dirichlet parameters as priors for the subsequent time slice. Let t denote the current time slice, then $B^{(t)}$ are all biterms occurring in time slice t. $\alpha^{(t)}$ are K-dimensional Dirichlet priors of time slice t, $\beta^{(t)}$ are $V \times K$-dimensional Dirichlet priors of time slice t, $\phi_k^{(t)}$ is the Dirichlet topic-word distribution of time slice t and $\theta^{(t)}$ is the Dirichlet topic distribution of time slice t. V is the length of the vocabulary and K is the number of topics. For the first time slice uniform priors are set. A biterm's topic is distributed as follows,

$$\begin{aligned} &p(z_i = k | z_{-1}^{(t)}, B^{(t)}, \alpha^{(t)}, \beta^{(t)}) \\ &\propto (n_{-i,k}^{(t)} + \alpha^{(t)}) \frac{(n_{-i,w_{i,1}|k}^{(t)} + \beta^{(t)})(n_{-i,w_{i,2}|k}^{(t)} + \beta^{(t)})}{[\sum_{w=1}^{V}(n_{-i,w|k}^{(t)} + \beta_{k,w}^{(t)}) + 1][\sum_{w=1}^{V}(n_{-i,w|k}^{(t)} + \beta_{k,w}^{(t)})]}. \end{aligned} \tag{21}$$

Again $n_{-i,w|k}$ is the topic-word count with the word w_i omitted. After the sampling procedure of a time slice t is completed, $n_k^{(t)}$, the number of times a topic k has been sampled and $n_{w|k}^{(t)}$, the number of times a word w has been sampled among topic k are obtained.

4.2 Algorithm

Two methods are formulated to calculate topic-mood, a multivariate and an univariate one. The multivariate version takes all sentiment categories $c = (1, \ldots, C)$ of LM-dictionary into account. By parsing these lexicons a multi-dimensional sentiment index $l_{w,c} = \{(u_{w_i|c}, u_{w_{i+1}|c}, \ldots, u_{w_V|c})\}_{c=1}^{C}$ is created, where $u_{w_i|c} = 1$ if the word w_i is of sentiment c, $u_{w_i|c} = 0$ otherwise. Multivariate topic-mood is calculated as follows,

$$s_{k,c}^{(t)} = l_{w,c} n_{w|k}^{(t)}. \tag{22}$$

The univariate version is based on the concept of sentiment polarity that considers the ratio between positive and negative words. The polarity lexicon $p_w = (u_{w_i}, u_{w_{i+1}}, \ldots, u_{w_V})$ is created, where $u_{w_i} = 1$ if word w_i is positive, $u_{w_i} = -1$ if word w_i is negative and $u_{w_i} = 0$ if it is not contained in the dictionary. Univariate topic-mood is calculated as follows,

$$s_k^{(t)} = p_w n_{w|k}^{(t)}. \tag{23}$$

Next, the priors for the subsequent time slice $t+1$ have to be estimated,

$$\alpha_k^{(t+1)} = \alpha_k^{(t)} + \lambda n_k^{(t)}, \tag{24}$$

$$\beta_{k,w}^{(t+1)} = \beta_{k,w}^{(t)} + \lambda n_{w|k}^{(t)}. \tag{25}$$

The parameter $\lambda \in [0, 1]$ is a decay weight that controls how much influence past assignments have on future sampling. When $\lambda = 0$ sampling across time slices is independent, if $\lambda = 1$ past assignments are accumulated. In other words, λ defines the memory of the model. The higher λ, the longer past assignments will be influential. The influence for $0 < \lambda < 1$ decreases exponentially.

Algorithm 3 Sentiment BTM

Input: $K, \alpha, \beta, \lambda, B^1, \ldots, B^T$
Output: $\{s_k^{(t)}\}_{t=1}^T$

1: Set $\alpha^{(1)} = (\alpha, \ldots, \alpha), \beta^{(1)} = \{\beta_k^{(1)} = (\beta, \ldots, \beta)\}_{k=1}^K$
2: **for** $t \leftarrow 1$ to T **do**
3: Randomly assign topics to biterms
4: **for each** iteration **do**
5: **for each** $b_i = (w_{i,1}, w_{i,2}) \in B^{(t)}$ **do**
6: Draw topic $z_i \sim p(z_i = k | z_{-1}^{(t)}, B^{(t)}, \alpha^{(t)}, \beta^{(t)})$
7: Update $n_{w_i,1|k}^{(t)}, n_{w_i,2|k}^{(t)}$ and $n_k^{(t)}$
8: **end for**
9: **end for**
10: Compute $s_k^{(t)}$ (Eq. 23)
11: Update $\alpha_k^{(t+1)}, \beta_{k,w}^{(t+1)}$ (Eq. 24, 25)
12: **end for**

Alg. 3 shows the univariate variant of sBTM. Line 1 shows that sBTM deals with multidimensional priors. They are randomly initialized for the first time slice according to the single-valued hyperparameters α and β. Lines 3-9 are similar to the BTM (see Alg. 2), where the topic-word distribution of a time slice is calculated. In line 10 topic-specific mood is retrieved. In line 11 the priors for the next time slice are calculated.

sBTM draws a topic for each biterm in each iteration over a time slice. It needs $\sum_t^T N_{iter} K N_B^{(t)}$ iterations to complete, where N_{iter} is the number of Gibbs-iterations, K the number of topics and $N_B^{(t)}$ is the number of biterms of time

slice t. Each biterm is passed $N_{iter}K$ times, giving a runtime complexity of,

$$O(N_{iter}KN_B), \tag{26}$$

where N_B is the overall number of biterms. A document containing l distinct words will generate $l(l-1)/2$ biterms. Under the assumption that all documents have the same number of distinct words, the overall number of biterms $N_B = \frac{N_D l(l-1)}{2}$, where N_D is the number of documents in the corpus. Since sBTM is an online algorithm that iterates over each time slice separately, the memory usage is kept low. sBTM has $K + VK + N_B^{(t)}$ variables in memory with V being the vocabuary size.

5 Empirical Study

Although the sBTM is a generic model for topic based mood extraction, its performance is evaluated in application of financial time series prediction. The target assets are cryptocurrencies since there is evidence that these assets are sensitive to public mood [38][39]. Cryptocurrencies are expected to be a homogeneous asset group. The idea is, that sBTM can be applied over multiple currencies to increase the validity of this experiment.

5.1 Data

The cryptocurrencies are chosen according to their market capitalization and release date. Currencies with high market capitalization are expected to be more present on Twitter than low-cap coins. A (relatively) long coin history is essential to get data over a long period of time. The currencies of choice are Bitcoin, Etherum, Litecoin and Ripple, with the target currency being USD. The data is scraped from *Yahoo! Finance*, a financial news site that provides pricing data from a variety of financial assets.

The official, non-premium Twitter API does not provide historical tweets ranging back beyond a 7-day period, forcing one to find roundabout ways to collect the data. A scraper is built to extract tweets accessible through the Twitter Advanced Search[8], a web-interface by Twitter, which allows one to search historic tweets based on certain filter criteria. The scraper retrieves HTML by making REST-Requests. Tweets are parsed from HTML using the Python library *Beautiful Soup*. This way, tweets ranging from August, 8th 2015 to Jannuary, 24th 2019 are retrieved. The filter of the Twitter Advanced Search was chosen to exclusively provide English tweets including the full name of the currencies (case-insensitive). Tweets provided by the Twitter Advanced Search are already filtered to exclude spam-content: "In order to keep your search results relevant, Twitter filters search results for quality Tweets and accounts. Material that jeopardizes search quality or creates a bad search experience for other people may be automatically removed from Twitter search"[9]. In comparison to existing work [5][40], this data set spans over a wider time range and consists of higher quality tweets.

[8] https://twitter.com/search-advanced (25.01.2019)
[9] https://help.twitter.com/en/rules-and-policies/twitter-search-policies - (27.04.2019)

Table 1: Tweets referring to Bitcoin, Ethereum, Litecoin and Ripple are scraped from the Twitter Advanced Search. The data ranges from August, 8th 2015 to Jannuary, 24th 2019. Low quality tweets are removed from the corpus.

Currency	Bitcoin	Ethereum	Litecoin	Ripple	Total
tweets	455,829	100,861	55,702	76,254	599,964
quality tweets	123,133	20,459	12,204	26,749	182,545

5.1.1 Preprocessing

In a first iteration irrelevant and noisy content is removed. As it is common in the Crypto-community to mention multiple coins as hash- and cashtags (e.g. "Crypto is the future #bitcoin #litecoin #ripple"), tweets with more than four hashtags or cashtags are removed. The number of retweets indicates the popularity of a tweet. Under the assumption that popular tweets are more influential in financial markets than unpopular ones, tweets with little retweets are dropped. As depicted in Fig. 4, the popularity of cryptocurrencies increased during the inspection period, which is why the number of retweets has to be set in temporal context. Here, the median of retweets was calculated over a moving window of 30 days and all tweets below are dropped.

sBTM has a relatively high runtime complexity[10] which depends on the number of biterms in the corpus. Therefore, the number of distinct words in the corpus is reduced. Under the bag-of-words assumption irrelevant words can be easily eliminated. The most common 1,000 words of each coin corpus are extracted and supplemented by the 4,135 opinion words defined by Loughran and McDonald[11]. Next, stop words as defined by Loughran and McDonald[11] are removed, since these words (and, the, of, ...) do not provide any information content. The search keywords and single character "words", hashtags, cashtags and URLs are stripped as well. Finally, the Porter stemming algorithm [41] is used to remove unnecessary word variations, leading to a final vocabulary of approximately 2,300 distinct terms. Tweets which contain less than four indexed terms are dropped. In the end 182,545 out of 599,964 tweets are left. With respect to the relative sparse data set, tweets are grouped on a weekly basis. The average number of tweets per week are 684 for Bitcoin, 113 for

[10] BTM has $l(l-1)/2$ times the runtime complexity of LDA, where l is the avg. number of distinct words in a tweet

[11] https://sraf.nd.edu/textual-analysis/resources (23.03.2019)

Ethereum, 234 for Dash, 67 for Litecoin and 147 for Ripple.

5.1.2 Topic Exploration

For LDA and BTM the number of topics needs to be known beforehand. Some experiments have been conducted with multiple numbers of topics. A small number (5,10) lead to unbalanced topic clusters, which is why the number of topics has been set to 20. To check whether LDA or BTM performs best on the given data set, one week of data[12] is picked. Both algorithms are applied with the same hyperparameters, $K = 20$, $\alpha = 1$, $\beta = 0.01$ and 100 Gibbs sampling iterations. The quality of clustering is determined based on the coherence score introduced by [7],

$$C(k, Q^{(k)}) = \sum_{m=2}^{M} \sum_{n=1}^{m-1} \log \frac{D(q_m^{(k)}, q_n^{(k)}) + 1}{D(q_n^{(k)})}, \tag{27}$$

where $Q^{(k)} = (q_m^{(k)}, \ldots, q_M^{(k)})$ are the M most probable words of topic k, $D(q_n^{(k)})$ is the number of times word $q_n^{(k)}$ occurs in a document and $D(q_m^{(k)}, q_n^{(k)})$ is the number of times words $q_m^{(k)}$ and $q_n^{(k)}$ co-occur in a document. The assumption is that a topic is more coherent if its most probable words often co-occur in the corpus. Numbers closer to zero indicate higher coherence of topics. Tab. 2 suggests that the BTM is the right choice for the given data.

[12] The last full week ranging from 14.01.2019 until 20.01.2019

Table 2: LDA and BTM is applied on a subset of the corpus to extract 20 topics. The coherence score of Eq. 27 with $M = 20$ is calculated. BTM does produce more coherent topics (smaller coherence score) across all currencies.

Topic	Bitcoin		Ethereum		Dash		Litecoin		Ripple	
	BTM	LDA	BTM	LDA	BTM	LDA	BTM	LDA	BTM	LDA
0	-113	-106	-91	-75	-76	-77	-52	-40	-83	-69
1	-110	-120	-56	-70	-65	-66	-33	-44	-69	-68
2	-113	-121	-69	-79	-63	-87	-48	-38	-47	-74
3	-106	-114	-77	-90	-76	-81	-53	-51	-49	-70
4	-105	-123	-57	-81	-62	-70	-48	-49	-58	-58
5	-110	-125	-58	-86	-66	-66	-15	-56	-46	-70
6	-105	-113	-72	-93	-61	-81	-31	-39	-43	-65
7	-81	-113	-75	-80	-45	-84	-36	-37	-65	-49
8	-108	-116	-75	-87	-88	-73	-29	-32	-45	-74
9	-112	-118	-67	-90	-67	-73	-11	-61	-38	-64
10	-112	-116	-55	-91	-79	-78	-18	-49	-75	-62
11	-98	-108	-74	-86	-87	-73	-38	-46	-49	-58
12	-114	-117	-64	-79	-69	-72	-41	-43	-69	-62
13	-107	-126	-78	-87	-43	-87	-39	-25	-47	-78
14	-107	-106	-74	-83	-84	-81	-2	-54	-15	-81
15	-99	-121	-78	-95	-63	-80	-35	-50	-82	-66
16	-103	-113	-67	-81	-53	-90	13	-48	-47	-67
17	-108	-123	-66	-74	-68	-79	-21	-47	-47	-75
18	-93	-112	-79	-82	-78	-83	-38	-55	-73	-76
19	-61	-110	-19	-87	-34	-76	0	-42	-26	-72
avg.	-103	-116	-67	-84	-66	-78	-29	-45	-54	-68

To apply sBTM, λ needs to be determined. A rather high λ is chosen. The aim is to have consistent topic over the complete time period. A small λ would lead to a more dynamic topic evolution, which would complicate exploratory analysis. sBTM is applied with $\lambda = 0.95$, $K = 20$, $\alpha = 1$, $\beta = 0.01$ and 100 Gibbs iterations per time slice. The sBTM is implemented in python/cython and open-sourced on Github[13]. Topics are often inspected by looking at the most probable words. Tab. 3 shows the 20 most probable words for each Bitcoin topic. Since preprocessing involves Porter-stemming, the words are shortened.

[13]https://github.com/markoarnauto/biterm - 16.05.2019

Multiple, little informative modal verbs are present (will, can, should) indicating that BTM is prone to background words. While most topics seem fuzzy, which is typical in sub-topic modelling, some patterns can be recognized. For example, topic 6 covers automated price updates, which are usually of the form "The latest Bitcoin Price Index is xxxx USD". Topic 19 covers tweets of the alleged unveiling of the Bitcoin inventor Satoshi Nakamoto, that had high Twitter attraction in early 2016. Topics of other currencies are provided in the Appendix B.

Table 3: The 20 most probable words for each of 20 Bitcoin topics.

Topics	Top 20 words
0	will, use, can, blockchain, your, peopl, world, like, us, crypto, one, what, exchang, money, say, time, new, make, bank, get
1	crypto, blockchain, currenc, new, digit, money, news, invest, via, can, say, launch, other, make, week, valu, fund, asset, read, technolog
2	will, make, like, exchang, could, time, commun, great, support, project, should, even, still, today, open, fork, thank, block, via, real
3	will, can, market, get, use, buy, like, btc, cash, day, today, back, want, trade, look, commun, know, take, coin, week
4	what, bank, currenc, use, market, would, peopl, go, central, see, btc, crypto, big, next, there, govern, time, news, may, im
5	your, here, crypto, buy, get, time, what, go, market, peopl, use, there, btc, year, good, start, us, read, look, still
6	price, usd, latest, index, exchang, across, averag, here, market, btc, news, crypto, last, day, live, watch, sinc, follow, trade, volum
7	like, follow, retweet, money, want, notif, peopl, crypto, there, us, rt, year, see, everyon, one, followback, everyoneturn, day, make, what
8	exchang, can, your, buy, trade, new, will, btc, user, card, payment, sell, want, say, launch, store, cash, transact, coinbas, give
9	market, will, exchang, price, year, trade, time, go, first, new, next, currenc, what, global, becom, week, cap, month, could, one
10	market, one, like, would, futur, new, develop, say, exchang, work, want, great, token, major, user, coin, start, project, contract, compani
11	will, what, can, here, crypto, day, peopl, know, one, use, us, fork, make, come, block, hard, week, follow, happen, futur
12	will, your, follow, first, token, btc, announc, make, wallet, network, trade, cash, use, launch, blockchain, platform, there, payment, time, mine
13	today, trade, market, get, join, day, price, year, us, crypto, valu, use, support, commun, go, block, dont, mine, month, live
14	will, new, your, one, price, there, here, world, market, come, want, today, year, know, see, dont, follow, commun, first, currenc
15	price, new, high, time, your, can, go, what, buy, week, there, one, hit, token, look, hour, month, wallet, updat, everi
16	will, like, market, your, day, price, money, dont, back, time, look, currenc, can, new, get, us, transact, year, work, next
17	will, like, new, day, world, one, price, valu, million, here, first, cash, everyon, make, start, coin, thing, back, adopt, take
18	your, will, payment, first, accept, get, wallet, bank, new, news, use, becom, via, adopt, see, user, transact, retweet, way, good
19	satoshi, he, craig, wright, nakamoto, creator, claim, us, australian, say, man, creat, scheme, founder, report, ponzi, regul, take, fraud, back

Table 4: The autocorrelation coefficients of weekly abnormal return is shown. The bold autocorrelation coefficients are statistically significant. Ethereum's abnormal return is autocorrelated at lag 3 and beyond. Litecoin has autocorrelation on lag 1 and 2.

Lag	Bitcoin	Ethereum	Litecoin	Ripple
1	0.033	0.013	**-0.182**	0.122
2	0.055	0.053	**0.034**	0.022
3	0.095	**0.275**	0.014	-0.028
4	-0.064	**-0.056**	0.009	0.023
5	0.001	**-0.072**	0.062	0.156
6	0.069	**0.089**	-0.005	0.159
7	-0.055	**-0.123**	-0.079	-0.046

5.2 Baselines

The effectiveness of the sBTM in cryptocurrency price prediction is compared against baseline forecasts. The idea is to find some metrics that yield predictive value and test whether sBTM outperforms those. First, it is tested whether historic price yields insight on future price movement by inspecting autocorrelations. Then it is tested whether mere Twitter activity or topic-specific Twitter activity is predictive. Next, one-dimensional and multidimensional mood is analyzed.

5.2.1 Price

A popular and simple baseline in price prediction tasks is to use past prices to predict future prices. If prices yield systematic pattern this is the case. A popular method to check for linear patterns is to inspect the autocorrelation. Since correlation works on stable time series, the weekly abnormal return of altcoins (as defined in Eq. 1) is used instead of the absolute price. Tab. 4 shows that Ethereum and Litecoin are autocorrelated across multiple lags/weeks, with bold-lettered p-values being significant. The significance of autocorrelation is determined with the Ljung Box Test. Usually, financial time series do not yield autocorrelation. A possible explanation for the autocorrelation of Ethereum and Litecoin is the systematic relation to Bitcoin price [9], as the abnormal return is defined as the difference to Bitcoin's return.

5.2.2 Activity

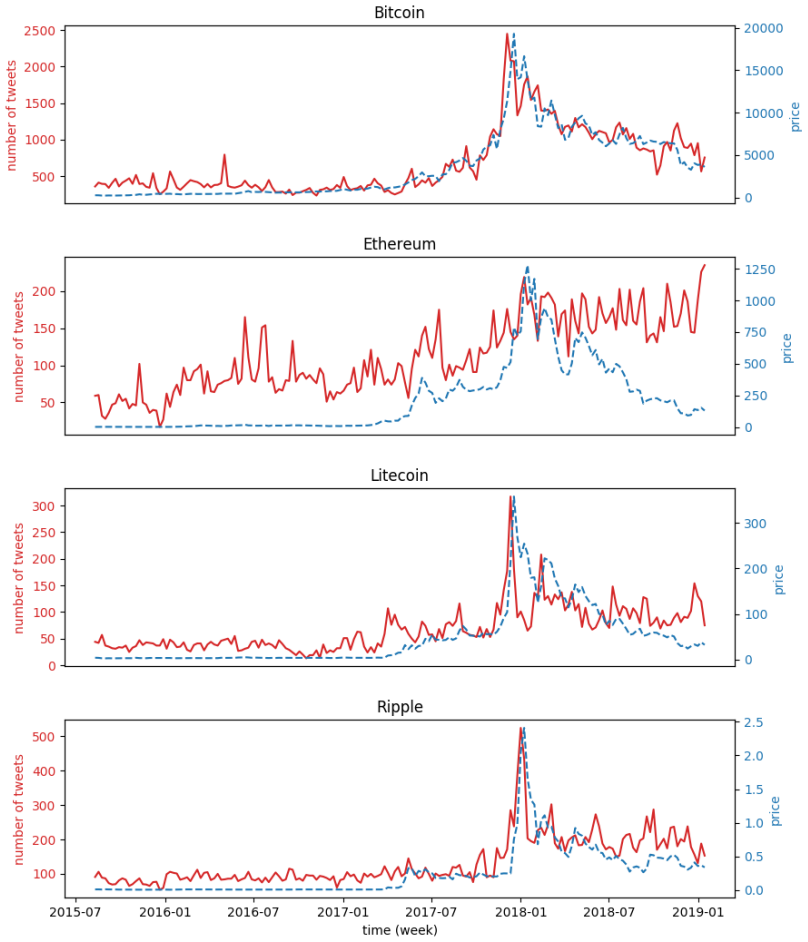

Figure 4: The number of tweets about a currency (red) and the price (blue) is depicted. Twitter activity and asset prices are closely aligned. The price peaked at the end of 2017 before the so-called crypto-bubble burst.

Table 5: Return → Activity: The p-values of Granger causality analysis indicate whether price movement impacts Twitter activity. The lag indicate the order of the VAR-model used for Granger-causality testing. Statistically significant Granger-cause is indicated by bold-lettered p-values. Price movement effects Twitter activity for Bitcoin, Litecoin and Ripple.

Lag	Bitcoin	Ethereum	Litecoin	Ripple
1	**0.001**	0.989	**0.013**	0.270
2	**0.001**	0.656	0.051	0.052
3	**0.000**	0.695	0.145	**0.005**
4	**0.000**	0.617	0.214	**0.001**
5	**0.000**	0.657	**0.020**	**0.011**
6	**0.000**	0.654	0.056	**0.013**
7	**0.000**	0.570	0.110	**0.008**

It is well known that publicity can effect price formation [42][13][38]. Here the number of tweets[14] (Twitter activity) serves as metric for public attention. Fig. 4 suggests that there is correlation between Twitter activity and price formation. It also shows that Twitter activity is not stable over time. With the rise of attention regarding cryptocurrencies the activity increased and peaked at the end of 2017. The first difference of Twitter activity is tested to Granger-cause returns and the other way round. The Granger causality test is conducted on 7 weekly lags. P-values are tested on significance with a significance-level of 0.05 throughout this study. It is observed that Twitter activity regarding Bitcoin does not lead price movement but the other way round, price jumps accelerate Twitter activity. In the case of Litecoin and Ripple, Twitter activity does lead price movements and the other way round. This bidirectional relation between Twitter and financial markets has already been observed by [24]. Here it is argued, that Twitter has an feedback effect on cryptocurrencies. E.g. Litecoin's return increases Twitter activity and Twitter activity again effects price formation.

[14]More precise, it is the number of tweets obtained from the web-scraper. This quantity only approximates the true Twitter activity.

Table 6: Activity → Return: The p-values show that Twitter activity has significant impact on Litecoin and Ripple. Statistically significant p-values are bold-lettered.

Lag	Bitcoin	Ethereum	Litecoin	Ripple
1	0.543	0.631	**0.019**	**0.023**
2	0.139	0.713	**0.000**	**0.045**
3	0.328	0.656	**0.000**	**0.027**
4	0.607	0.797	**0.000**	0.124
5	0.764	0.862	**0.000**	0.057
6	0.169	0.913	**0.000**	**0.010**
7	0.056	0.906	**0.000**	**0.002**

The next baseline is topic specific Twitter activity. A novel approach of extending Twitter activity by topic modelling is tested. Under the assumption that some topics are more influential on price formation than others, it is checked whether topic specific activity has predictive power. The topic word count n_z from the sBTM is used as a measure of topic-specific activity. n_z denotes how many tokens are assigned to a topic. Topic-activity (see appendix Fig. 8) is not stable over time and the first difference is used with a lag of 7 for Granger analysis. Tab. 7 shows which topics are significantly influential. Although, Bitcoin's overall Twitter activity is not influential, topic-activity of topic 2, 3, 11, 12, 14 and 16 are influential. Interestingly, the most probable word of these topics is "will". Obviously, tweets dealing with the future of Bitcoin have impact on future returns. It is argued that topic-specific activity yields finer-grained insight than non-topic activity.

Table 7: Topic activity → return: Topic modelling via BTM is applied yielding 20 distinct topics for each currency. The weekly change in topic-activity is tested to Granger-cause price movements. Certain topic-activities yield predictive insight on Bitcoin, Litecoin and Ripple price movement indicated by bold-lettered p-values. Ethereum's price movement is not influenced by any topic-activity.

Topic	Bitcoin	Ethereum	Litecoin	Ripple
0	0.116	0.999	**0.000**	**0.032**
1	0.316	0.544	**0.021**	0.058
2	**0.036**	0.926	0.838	0.080
3	**0.049**	0.680	**0.007**	**0.026**
4	0.424	0.953	0.056	**0.011**
5	0.120	0.908	**0.000**	**0.008**
6	0.148	0.989	**0.032**	**0.002**
7	0.119	0.675	**0.007**	**0.006**
8	0.104	0.953	**0.004**	**0.026**
9	0.105	0.944	**0.001**	**0.000**
10	0.365	0.977	0.395	**0.028**
11	**0.046**	0.665	**0.001**	0.091
12	**0.045**	0.604	0.053	**0.005**
13	0.287	0.777	0.059	**0.011**
14	**0.037**	0.994	0.053	**0.005**
15	0.105	0.732	**0.002**	**0.007**
16	**0.030**	0.149	**0.005**	**0.001**
17	0.051	0.310	**0.005**	**0.003**
18	0.066	0.539	**0.011**	**0.000**
19	0.400	0.282	0.898	0.080

5.2.3 Mood

In multivariate mood analysis it is tested if mood dimensions Granger-cause price formation. Mood is extracted as described in Ch. 4 and normalized[15]. Each dimension corresponds to a sentiment dimension defined in the LM-dictionary. Fig. 5 depicts that the public discourse about cryptocurrencies has been gradually charged with emotions. It is also shown that constraining mood is absent. Tab. 8 shows that Bitcoin is most sensitive to Twitter mood while altcoins are

[15] Divided by the number of tweets within a time slice

hardly influenced by Twitter mood.

Tab. 8 suggests that positive and negative mood are useful mood dimension. In form of a polarity score, as the difference between positive and negative words, these dimensions are inspected more closely. The average polarity score of all currencies is depicted in the Appendix 9. As indicated by multivariate mood analysis, polarity does have significant impact on Bitcoin (see Tab. 9). Surprisingly, Litecoin is neither caused by negative nor positive mood, but it is caused by polarity. To conclude the findings on multivariate mood and polarity analysis, Bitcoin is driven by Twitter emotions while there is little evidence that alcoins are. It is argued that particular altcoins do not attract enough public attention to be decoupled from overall crypto mood. It is important to note, that this study investigates dependencies regarding a currencies individual mood. Nevertheless, it might be possible that altcoins are driven by Twitter mood referring to Bitcoin or the overall Blockchain-technology which is not inspected here.

Table 8: Mood $\rightarrow$ return: Granger causality is applied with an VAR-model of order 7. Bold-lettered p-values indicate significant Granger-cause. Bitcoin is Granger-caused by negative, positive, uncertain and modal mood and Ripple by positive mood. Ethereum and Litecoin are not driven by Twitter mood.

Sentiment	Bitcoin	Ethereum	Litecoin	Ripple
negative	**0.000**	0.716	0.300	0.783
positive	**0.016**	0.125	0.703	**0.016**
uncertain	**0.040**	0.563	0.751	0.159
litigous	0.062	0.308	0.871	0.893
modal	**0.037**	0.286	0.549	0.474
constraining	0.076	0.405	0.219	0.269

Table 9: Polarity → return: Polarity is the difference of positive and negative mood. It is tested whether polarity Granger-causes returns across 7 lags. Bitcoin is significantly Granger-caused by polarity, indicated by bold-lettered p-values. Litecoin is Granger-caused by polarity on lag 3 and 4. Ethereum and Ripple are not caused by polarity.

Sentiment	Bitcoin	Ethereum	Litecoin	Ripple
1	**0.000**	0.141	0.767	0.629
2	**0.000**	0.307	0.296	0.748
3	**0.000**	0.077	**0.028**	0.660
4	**0.000**	0.161	**0.043**	0.725
5	**0.000**	0.187	0.057	0.757
6	**0.000**	0.179	0.091	0.505
7	**0.000**	0.265	0.111	0.448

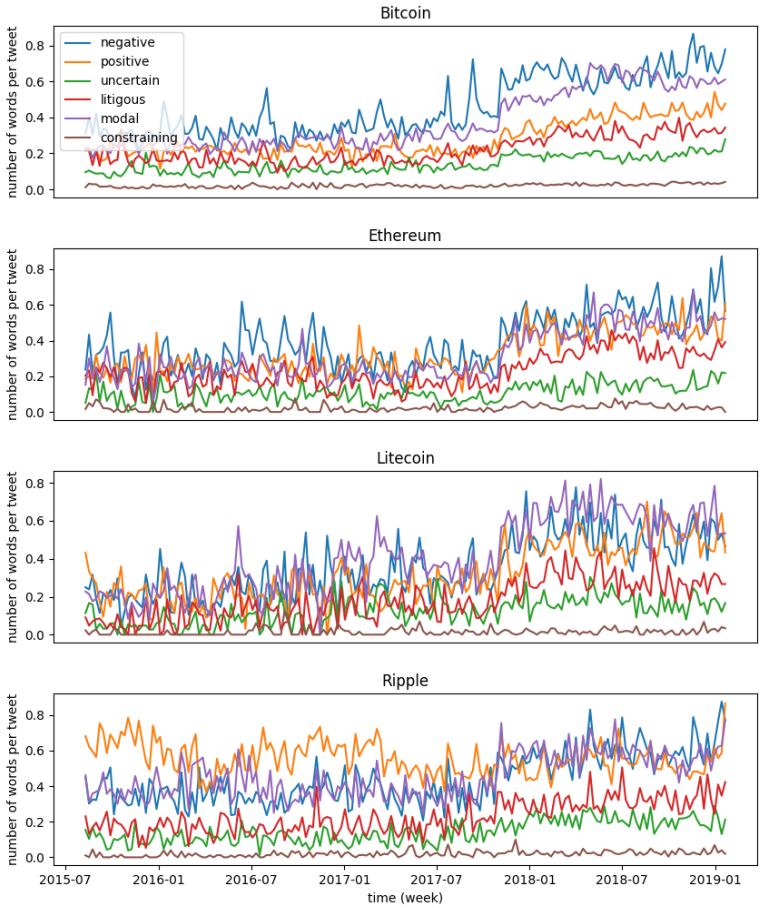

Figure 5: Mood is measured by counting sentimental words as defined in the LM-dictionary. More precisely, mood is the average number of sentimental words occurring in a tweet. The magnitude of mood is dependent on the number of sentimental words available per mood dimension. The LM-dictionary provides 2355 negative, 354 positive, 297 uncertain, 903 litigous, 60 modal and 184 constraining words. Constraining words do hardly appear which is why this mood dimension is of little use.

5.3 Topic Mood

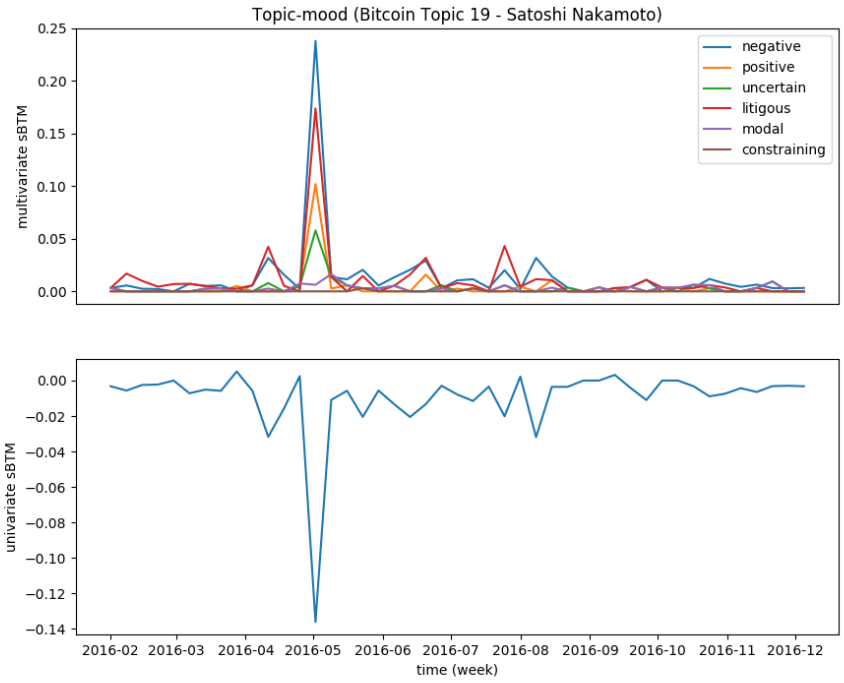

Figure 6: Topic-mood: The topic based mood of Bitcoin topic 19, which refers to the alleged unveiling of the Bitcoin inventor on Mai 2016. Multivariate sBTM is depicted above. The negative and litigous portions are predominant and the positive portions faded almost completely as the unveiling was more and more doubted by the Twitter community. The difference between negative and positive mood is expressed by univariate sBTM beneath.

Applying multivariate sBTM on a corpus results in $K \times C$ time series, where K is the number of topics and C is the number of mood dimensions. Univariate sBTM results in K time series, each expressing the polarity of a topic. Fig. 6 exemplifies the difference between multivariate and univariate sBTM on Bitcoin topic 19, which refers to the alleged unveiling of the Bitcoin inventor on Mai 2016. While the multivariate sBTM extracts various mood dimensions, the univariate sBTM extracts topic polarity.

Tab. 10 depicts the effect of univariate sBTM on the currencies. It shows that there is one influential topic on Ripple and none on Ethereum and Litcoin. This again demonstrates that altcoins are almost independent from Twitter mood. In contrast, Bitcoin is effected by most topics. 15 out of 20 Bitcoin-topics Granger-cause price formation. The remaining topics are considered to be irrelevant or spam. Indeed Topic 7 covers spam content indicated by its most probable words: 'like, follow, retweet, money ...' (see Tab. 3). Topic 19 refers to the unveiling of Satoshi Nakamoto, which is obviously also not influential over the complete testing period.

Table 10: Topic polarity → return: It is tested whether topic-specific mood Granger-causes price formation. The lag of the VAR-model used for Granger analysis is 7. Significant p-values are bold-lettered. Bitcoin's price movement is Granger-caused by each topic's polarity except that of topic 5, 7, 12, 18 and 19. Altcoins are hardly influenced by Twitter mood.

Topic	Bitcoin	Ethereum	Litecoin	Ripple
0	**0.000**	0.711	0.585	0.213
1	**0.000**	0.560	0.502	0.489
2	**0.000**	0.536	0.755	0.061
3	**0.000**	0.641	0.443	0.536
4	**0.000**	0.185	0.278	**0.007**
5	0.134	0.770	0.767	0.428
6	**0.000**	0.090	0.365	0.288
7	0.052	0.749	0.177	0.834
8	**0.000**	0.864	0.339	0.097
9	**0.000**	0.811	0.716	0.327
10	**0.001**	0.106	0.671	0.339
11	**0.000**	0.964	0.477	0.414
12	0.856	0.769	0.664	0.088
13	**0.000**	0.220	0.647	0.451
14	**0.010**	0.942	0.732	0.494
15	**0.000**	0.515	0.185	0.847
16	**0.004**	0.594	0.161	0.315
17	**0.000**	0.594	0.270	0.955
18	0.055	0.748	0.369	0.098
19	0.561	0.190	0.301	0.474

5.4 Prediction

In the prediction task the baselines and the sBTM are compared against each other. Of special interest is the comparison between non-topic and topic-specific approaches. The high dimensional data provided by multivariate sBTM is difficult to incorporate in a prediction task with a limited number of observations. Therefore the univariate (polarity) sBTM is preferred. Univariate sBTM still gives K (number of topics) dimensional data, which is critical to incorporate into a VAR-based prediction. [6] selected relevant topics by examining the most prominent words of a topic. In this study a systematic selection criterion is used. The data set is split by half, yielding 90 weeks of training data and 90 weeks of test data. Granger causality is applied on the training data to tag each dimension with a p-value indicating the probability of a time series to Granger-cause price formation. For this Granger-testing, a VAR-model of order 2 is used two minimize spurious causality yielded by higher order models. Then, 4 dimensions having the smallest p-values are selected as features. This selection procedure does not guarantee to pick all significant dimensions nor does it guarantee that any of the selected dimension is significant but it can be easily applied across multiple currencies. The selected dimensions for Bitcoin and their impulse responses (based on the training data) are depicted in Fig. 7. As expected, an increase in topic-polarity results in increased subsequent return.

There is no easy way to determine the optimal lag for multiple models over multiple currencies. Therefore, prediction is performed over multiple lags. 1 up to 3 lags are used to predict abnormal return. Similar to [6], a moving-window approach is applied, where VAR-models are fit on a window with the size of the training data (90 weeks) and a one-step forecast is performed. Then the window is shifted by one time slice and another model is trained giving the next one-step forecast. This moving-window approach is favourable in economic forecasts. Due to the dynamic nature of financial time series (and public mood) it is more effective in learning short term relationship. This approach also increases the size of the test data. The most basic forecast based on price is performed using an AR-model. The remaining models, namely *Activity*, *Topic Activity*, *Sentiment* and *sBTM* are based on a VAR-model that additionally incorporates past prices. The conventional evaluation metric in economic time series prediction is up-down accuracy. It denotes how often the direction of price movement is predicted in percent. Topic-activity as well as sBTM outperform their non-topic baselines suggesting that topic-specific analysis is favourable. The best prediction accuracy is achieved by Litecoin's topic-activity with 71%. A notable 69% prediction accuracy was achieved by sBTM on Bitcoin outperforming the best

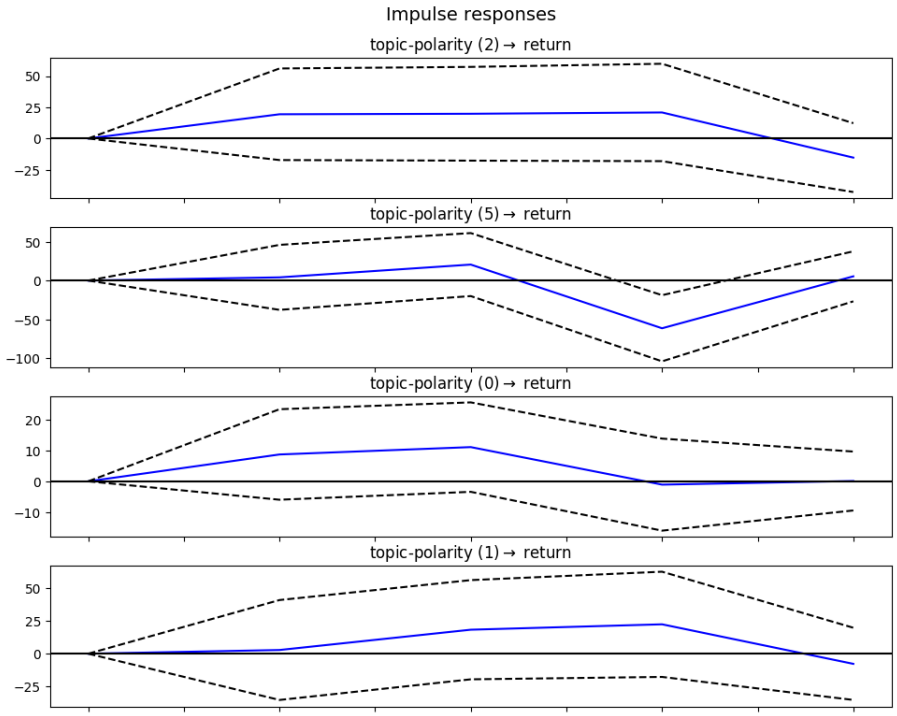

Figure 7: The impulse response of polarity on Bitcoin returns is shown. These 4 topics (2,5,0,1) are most influential in the training set, which is why they are chosen as input features for the prediction task. It is shown that an increase in polarity causes latent increase in price.

conventional method by 3% prediction accuracy.

Table 11: All baselines as well as the sBTM are used in an prediction task. Prediction is accomplished within a VAR-Framework. Each method is tested on multiple currencies over multiple lags. The average accuracy for each method is presented at the bottom.

Currency	Lag	Price	Activity	Topic Activity	Sentiment	sBTM
Bitcoin	1	0.54	0.59	0.57	0.66	0.66
	2	0.53	0.58	0.57	0.63	0.69
	3	0.54	0.53	0.60	0.58	0.64
Ethereum	1	0.50	0.48	0.59	0.54	0.53
	2	0.50	0.53	0.50	0.53	0.60
	3	0.60	0.54	0.49	0.51	0.54
Litecoin	1	0.66	0.63	0.69	0.68	0.64
	2	0.63	0.64	0.69	0.67	0.62
	3	0.59	0.66	0.71	0.61	0.62
Ripple	1	0.51	0.49	0.43	0.43	0.54
	2	0.49	0.50	0.44	0.50	0.51
	3	0.53	0.48	0.43	0.43	0.49
		0.55	**0.55**	**0.56**	**0.56**	**0.59**

6 Discussion

6.1 Conclusion

The experiment has been setup with the assumption that the results among cryptocurrencies are similar. This assumption did not hold. Ethereum and Litecoin price yield systematic pattern, which is why the most basic AR-model achieved extraordinary accuracy. The best accuracy of the AR-model on Ethereum is 60% and on Litecoin 66%. Since abnormal return of altcoins has been defined as the deviation from Bitcoin's return, these patterns might be explained by the correlation of altcoins and Bitcoin [9]. This suggests that statistical arbitrage strategies between Bitcoin and alctoins are profitable. It is shown that Litecoin's and Ripple's return is Granger-caused by Twitter-activity. The fact that mere publicity impacts price formation is coherent with existing literature [42][4]. It was demonstrated that the connection of Twitter activity and price change can also be the other way round. In the case of Bitcoin, Litecoin and Ripple, Twitter activity is accelerated by price movements. This bi-directional causality of asset pricing and publicity is well known [24], which underpins the validity of so-called feedback models. Feedback models suggest that e.g. publicity accelerates price movements, price movements in turn accelerates publicity and so on and so forth.

The novel approach to use topic-activity slightly outperformed it's non-topic counterpart, mostly because of the high performance on Litecoin. Granger analysis (Tab. 5, 7) indicates that activity and topic-activity are also influential on Ripple, nevertheless Ripple's return is predicted badly. Slight discrepancies between Granger analysis and the final prediction is due to the fact that Granger analysis was performed on the complete 180 weeks of data while the prediction model was performed on 90 weeks of data (due to training and test split). The relation between Twitter activity and Ripple price formation must have changed in that time. While topic-specific activity might be a promising subject for further research, there are better models than the BTM. Algorithms that capture topic burstiness, such as [43][44] might be better suited.

Fig. 8 indicates that dynamics drastically changed after the burst of the crypto-bubble in the end of 2017. [5] simply eliminates irregular, abnormal economic episodes. Here the complete time period of the test data is predicted which confirms the robustness of the provided method. The moving VAR-model adapts well to dynamic economics.

Mood of altcoins does not impact price formation. Probably, because altcoins are too closely tied on Bitcoin, which is dominant in cryptocurrency price

formation. The public mood regarding particular altcoins might be overruled by the mood about Bitcoin. But Twitter mood is influential on Bitcoin. Topic-mood of Bitcoin achieved a maximum of 69% accuracy and an average accuracy of 66%. The average non-topic mood prediction was outperformed by 4%. This is because sBTM successfully separates relevant from irrelevant topics. The most characteristic words of irrelevant topics are typical spam-words, such as *like, follow, retweet*. sBTM is proven to be an effective method in extracting topic-specific mood.

6.2 Further Work

The strength of BTM in short-text modelling is undoubted. Nevertheless, in this experiment topics are dense and background words are shared across most of the topics. It is argued that BTM performance is improved if the generation of background-words will be incorporated into the generative process.
It is shown that most topics Granger-cause Bitcoin price. Probably because the topics are extremely similar. sBTM might work best when a thematically diverse data set is inspected. Considering a corpus of global news headlines referring to a broad spectrum of topics, sBTM can be exploited to better forecast assets like the DJIA, which is influenced by various, highly diverse topics.

In principle sBTM is a two-pass method, where topics are extracted before sentiment. But, sentiment analysis is highly dependent on the domain to which it is applied, such that the sentimental value of a word can change across topics. For instance, the word *low* in *low cost* and *low salary* have opposite polarity. It might be beneficial to incorporate sentiment analysis in the probabilistic process of BTM. Similar to [30] and [45], topic and sentiment modelling could be done in parallel.

This study sheds light on Bitcoin and altcoin price formation. While there is significant research on Bitcoin price formation [46][47], there is little investigation done on altcoins. Since altcoins are highly influenced by Bitcoin and Bitcoin is influenced by its Twitter-mood it might be worth testing if Bitcoin mood influences altcoins.

BTM has a relatively long runtime-complexity. As Gibbs sampling is difficult to parallelize, it might be worth investigating into more efficient sampling procedures. A faster version is presented by [48], which could serve as a basis for a faster sBTM.

References

[1] Eugene F. Fama. "Efficient Capital Markets: A Review of Theory and Empirical Work". In: *The Journal of Finance* 25.2 (1970), pp. 383–417. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2325486.

[2] Daniel Kahneman and Amos Tversky. "On the Psychology of Prediction". In: *Psychological Review* 80.4 (1973), pp. 237–251. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.395.3759&rep=rep1&type=pdf.

[3] Paul C. Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. "More Than Words: Quantifying Language to Measure Firms' Fundamentals". In: *The Journal of Finance* 63.3 (2008), pp. 1437–1467. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2008.01362.x.

[4] Werner Antweiler and Murray Z. Frank. "Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards". In: *The Journal of Finance* 59.3 (2004), pp. 1259–1294. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2004.00662.x.

[5] Johan Bollen, Huina Mao, and Xiao-Jun Zeng. "Twitter mood predicts the stock market". In: *CoRR* abs/1010.3003 (2010). URL: http://arxiv.org/abs/1010.3003.

[6] Jianfeng Si et al. "Exploiting Topic based Twitter Sentiment for Stock Prediction". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2013, pp. 24–29. URL: http://aclweb.org/anthology/P13-2005.

[7] David Mimno et al. "Optimizing Semantic Coherence in Topic Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011, pp. 262–272. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145462.

[8] Danah M. Boyd and Nicole B. Ellison. "Social Network Sites: Definition, History, and Scholarship". In: *Journal of Computer-Mediated Communication* 13.1 (Oct. 2007), pp. 210–230. ISSN: 1083-6101. URL: https://doi.org/10.1111/j.1083-6101.2007.00393.x.

[9] Binance Research. *Are Cryptoassets Highly Correlated?* 2019. URL: https://info.binance.com/en/research/marketresearch/crypto-correlations.html (visited on 05/06/2019).

[10] Eugene F. Fama et al. "The Adjustment of Stock Prices to New Information". In: *International Economic Review* 10.1 (1969), pp. 1–21. ISSN: 00206598, 14682354. URL: http://www.jstor.org/stable/2525569.

[11] Michael C. Jensen. "The Performance of Mutual Funds in the Period 1945-1964". In: *The Journal of Finance* 23.2 (1968), pp. 389–416. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2325404.

[12] Robert J. Shiller. "From Efficient Markets Theory to Behavioral Finance". In: *Journal of Economic Perspectives* 17.1 (Mar. 2003), pp. 83–104. URL: http://www.aeaweb.org/articles?id=10.1257/089533003321164967.

[13] Gur Huberman and Tomer Regev. "Contagious Speculation and a Cure for Cancer: A Nonevent That Made Stock Prices Soar". In: *The Journal of Finance* 56.1 (2001), pp. 387–396. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/222474.

[14] Kathleen Weiss Hanley and Gerard Hoberg. "The Information Content of IPO Prospectuses". In: *The Review of Financial Studies* 23.7 (Apr. 2010), pp. 2821–2864. ISSN: 0893-9454. URL: https://dx.doi.org/10.1093/rfs/hhq024.

[15] Tim Loughran and Bill McDonald. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1 (2011), pp. 35–65. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2010.01625.x.

[16] Colm Kearney and Sha Liu. "Textual sentiment in finance: A survey of methods and models". In: *International Review of Financial Analysis* 33 (2014), pp. 171–185. ISSN: 1057-5219. URL: http://www.sciencedirect.com/science/article/pii/S1057521914000295.

[17] Victor Lavrenko et al. "Mining of Concurrent Text and Time Series". In: *In Proceedings of the 6th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2000, pp. 37–44. URL: https://www.cs.cmu.edu/~dunja/KDDpapers/Lavrenko_TM.pdf.

[18] Narasimhan Jegadeesh and Di Wu. "Word power: A new approach for content analysis". In: *Journal of Financial Economics* 110.3 (2013), pp. 712–729. ISSN: 0304-405X. URL: http://www.sciencedirect.com/science/article/pii/S0304405X13002328.

[19] Sasa Petrovic et al. *Can Twitter Replace Newswire for Breaking News?* 2013. URL: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM13/paper/view/6066.

[20] Fabrício Benevenuto et al. "Detecting spammers on twitter". In: *In Collaboration, Electronic messaging, Anti-Abuse and Spam Conference*. 2010. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.5340&rep=rep1&type=pdf.

[21] Majed Alrubaian et al. "Reputation-based credibility analysis of Twitter social network users". In: *Concurrency and Computation: Practice and Experience* 29.7 (2017). URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.3873.

[22] Jianshu Weng et al. "TwitterRank: Finding Topic-sensitive Influential Twitterers". In: *Proceedings of the Third ACM International Conference on Web Search and Data Mining*. WSDM '10. ACM, 2010, pp. 261–270. ISBN: 978-1-60558-889-6. URL: http://doi.acm.org/10.1145/1718487.1718520.

[23] Roy Bar-Haim et al. "Identifying and Following Expert Investors in Stock Microblogs". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Association for Computational Linguistics, 2011, pp. 1310–1319. ISBN: 978-1-937284-11-4. URL: http://dl.acm.org/citation.cfm?id=2145432.2145569.

[24] Timm O. Sprenger et al. "Tweets and Trades: the Information Content of Stock Microblogs". In: *European Financial Management* 20.5 (2014), pp. 926–957. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-036X.2013.12007.x.

[25] Thomas Hofmann. "Probabilistic Latent Semantic Indexing". In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '99. ACM, 1999, pp. 50–57. ISBN: 1-58113-096-1. URL: http://doi.acm.org.uaccess.univie.ac.at/10.1145/312624.312649.

[26] Yang Liu et al. *ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs*. 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.9902&rep=rep1&type=pdf.

[27] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3 (Mar. 2003), pp. 993–1022. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=944919.944937.

[28] David M. Blei and John D. Lafferty. "Dynamic Topic Models". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. ACM, 2006, pp. 113–120. ISBN: 1-59593-383-2. URL: http://doi.acm.org/10.1145/1143844.1143859.

[29] Yee Whye Teh et al. "Hierarchical Dirichlet Processes". In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1566–1581. URL: https://doi.org/10.1198/016214506000000302.

[30] Chenghua Lin and Yulan He. "Joint Sentiment/Topic Model for Sentiment Analysis". In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. CIKM '09. ACM, 2009, pp. 375–384. ISBN: 978-1-60558-512-3. URL: http://doi.acm.org/10.1145/1645953.1646003.

[31] Liangjie Hong and Brian D. Davison. "Empirical Study of Topic Modeling in Twitter". In: *Proceedings of the First Workshop on Social Media Analytics*. SOMA '10. ACM, 2010, pp. 80–88. ISBN: 978-1-4503-0217-3. URL: http://doi.acm.org/10.1145/1964858.1964870.

[32] Jianhua Yin and Jianyong Wang. "A Dirichlet Multinomial Mixture Model-based Approach for Short Text Clustering". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '14. ACM, 2014, pp. 233–242. ISBN: 978-1-4503-2956-9. URL: http://doi.acm.org/10.1145/2623330.2623715.

[33] Xiaohui Yan et al. "A Biterm Topic Model for Short Texts". In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13. Rio de Janeiro, Brazil: ACM, 2013, pp. 1445–1456. ISBN: 978-1-4503-2035-1. URL: http://doi.acm.org/10.1145/2488388.2488514.

[34] Michal Rosen-Zvi et al. "Learning Author-topic Models from Text Corpora". In: *ACM Trans. Inf. Syst.* 28.1 (Jan. 2010), 4:1–4:38. ISSN: 1046-8188. URL: http://doi.acm.org/10.1145/1658377.1658381.

[35] Weizheng Chen et al. "User Based Aggregation for Biterm Topic Model". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Compu-

tational Linguistics, 2015, pp. 489–494. URL: https://www.aclweb.org/anthology/P15-2080.

[36] Yunqing Xia et al. *Discriminative Bi-Term Topic Model for Headline-Based Social News Clustering*. 2015. URL: https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS15/paper/view/10428.

[37] Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan Claypool Publishers, 2012. ISBN: 1608458849, 9781608458844.

[38] Pavel Ciaian, Miroslava Rajcaniova, and d'Artis Kancs. "The economics of BitCoin price formation". In: *Applied Economics* 48.19 (2016), pp. 1799–1815. URL: https://doi.org/10.1080/00036846.2015.1109038.

[39] George Giaglis et al. "Using Time-Series and Sentiment Analysis to Detect the Determinants of Bitcoin Prices". In: Oct. 2015. URL: https://ssrn.com/abstract=2607167.

[40] M. Makrehchi, S. Shah, and W. Liao. "Stock Prediction Using Event-Based Sentiment Analysis". In: *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*. Vol. 1. Nov. 2013, pp. 337–342.

[41] M. F. Porter. "Readings in Information Retrieval". In: ed. by Karen Sparck Jones and Peter Willett. Morgan Kaufmann Publishers Inc., 1997. Chap. An Algorithm for Suffix Stripping, pp. 313–316. ISBN: 1-55860-454-5. URL: http://dl.acm.org/citation.cfm?id=275537.275705.

[42] David H. Solomon, Eugene Soltes, and Denis Sosyura. "Winners in the spotlight: Media coverage of fund holdings as a driver of flows". In: *Journal of Financial Economics* 113.1 (2014), pp. 53–72. ISSN: 0304-405X. URL: http://www.sciencedirect.com/science/article/pii/S0304405X14000403.

[43] Xiaohui Yan et al. "A Probabilistic Model for Bursty Topic Discovery in Microblogs". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, 2015, pp. 353–359. ISBN: 0-262-51129-0. URL: http://dl.acm.org/citation.cfm?id=2887007.2887057.

[44] W. Xie et al. "TopicSketch: Real-Time Bursty Topic Detection from Twitter". In: *IEEE Transactions on Knowledge and Data Engineering* 28.8 (Aug. 2016), pp. 2216–2229. ISSN: 1041-4347.

[45] Thien Hai Nguyen and Kiyoaki Shirai. "Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, July 2015, pp. 1354–1364. URL: https://www.aclweb.org/anthology/P15-1131.

[46] Mehmet Balcilar et al. "Can volume predict Bitcoin returns and volatility? A quantiles-based approach". In: *Economic Modelling* 64 (2017), pp. 74–81. ISSN: 0264-9993. URL: http://www.sciencedirect.com/science/article/pii/S0264999317304558.

[47] Elie Bouri et al. "On the hedge and safe haven properties of Bitcoin: Is it really more than a diversifier?" In: *Finance Research Letters* 20 (2017), pp. 192–198. ISSN: 1544-6123. URL: http://www.sciencedirect.com/science/article/pii/S1544612316301817.

[48] X. He et al. "Optimize collapsed Gibbs sampling for biterm topic model by alias method". In: *2017 International Joint Conference on Neural Networks (IJCNN)*. May 2017, pp. 1155–1162.

A Theory

A.1 LDA sampling-derivation

For collapsed Gibbs sampling the probability of a topic z being assigned to a word w_i, given all other topic assignments, is needed:

$$p(z_i|z_{-i}, \alpha, \beta, w). \tag{28}$$

By the rules of conditional probability, this can be rewritten such that:

$$\begin{aligned} p(z_i|z_{-i}, \alpha, \beta, w) &= \frac{p(z_i, z_{-i}, w|\alpha, \beta)}{p(z_{-i}, w|\alpha, \beta)} \propto p(z_i, z_{-i}, w|\alpha, \beta) \\ &= p(z, w|\alpha, \beta) = \int\int p(z, w, \theta, \phi|\alpha, \beta) d\theta d\phi. \end{aligned} \tag{29}$$

Considering the LDA model definition, the equation can be expanded to get:

$$p(w, z|\alpha, \beta) = \int\int p(\phi|\beta)p(\theta|\alpha)p(z|\theta)p(w|\phi_z)d\theta d\phi \tag{30}$$

Then, terms with dependent variables are seperated:

$$p(w, z|\alpha, \beta) = \int p(z|\theta)p(\theta|\alpha)d\theta \int p(w|\phi_z)p(\phi|\beta)d\phi \tag{31}$$

Both terms are multinomial distributions. The Dirichlet distribution is conjugate to the multinomial distribution. Therefore both terms can be reformulated with the beta function being $B(.) = \frac{\prod_k \Gamma(.)}{\Gamma(\sum_k .)}$. $\Gamma(.)$ is the gamma function. Starting with the first term,

$$\begin{aligned} \int p(z|\theta)p(\theta|\alpha)d\theta &= \int \prod_i \theta_{d,z_i} \frac{1}{B(\alpha)} \prod_k \theta_{d,k}^{\alpha_k} d\theta_d \\ &= \frac{1}{B(\alpha)} \int \prod_k \theta_{d,k}^{n_{d,k}+\alpha_k} d\theta_d \\ &= \frac{B(n_{d,.} + \alpha)}{B(\alpha)}. \end{aligned} \tag{32}$$

Similarly, the second term can be reformulated as follows,

$$\begin{aligned} \int p(w|\phi_z)p(\phi|\beta)d\theta &= \int \prod_d \prod_i \phi_{z_{d,i},w_{d,i}} \prod_k \frac{1}{B(\beta)} \prod_w \phi_{k,w}^{\beta_w} d\phi_k \\ &= \prod_k \frac{1}{B(\beta)} \int \prod_w \phi_{k,w}^{\beta_w + n_{k,w}} d\phi_k \\ &= \prod_k \frac{B(n_{k,.} + \beta)}{B(\beta)}. \end{aligned} \tag{33}$$

Substituting these equations into 31 gives,

$$p(w, z|\alpha, \beta) = \prod_d \frac{B(n_{d,.} + \alpha)}{B(\alpha)} \prod_k \frac{B(n_{k,.} + \beta)}{B(\beta)}. \tag{34}$$

Then the chain rule is applied. The hyperparameters α and β are left out for clarity.

$$\begin{aligned} p(z_i|z_{-i}, w) &= \frac{p(w, z)}{p(w, z_{-i})} = \frac{p(z)}{p(z_{-i})} . \frac{p(w|z)}{p(w_{-i}|z_{-i})p(w_i)} \\ &\propto \prod_d \frac{B(n_{.|d} + \alpha)}{B(n_{-i,.|d} + \alpha)} \prod_k \frac{B(n_{k|.} + \beta)}{B(n_{-i,.|k} + \beta)} \\ &\propto \frac{\Gamma(n_{d|k} + \alpha_k)\Gamma(\sum_{k=1}^K n_{d|k} + \alpha_k)}{\Gamma(n_{-i,d|k} + \alpha_k)\Gamma(\sum_{k=1}^K n_{-i,d|k} + \alpha_k)} . \frac{\Gamma(n_{w|k} + \beta_w)\Gamma(\sum_{w=1}^W n_{-i,w|k} + \beta_w)}{\Gamma(n_{-i,w|k} + \beta_w)\Gamma(\prod_{w=1}^W n_{w|k} + \beta_w)} \\ &\propto (n_{-i,d|k} + \alpha_k) \frac{n_{-i,w|k} + \beta}{\sum_{w'} n_{-i,w|k} + \beta_{w'}} \end{aligned} \tag{35}$$

A.2 BTM sampling-derivation

For Gibbs sampling the probability of a topic z being assigned, given all biterms and all other topic assignments, is needed

$$p(z_i|z_{-1}, B) = \frac{p(z, B)}{p(z_{-i}, B)} \propto \frac{p(B|z)p(z)}{p(B_{-i}|z_{-i})p(z_{-i})}. \tag{36}$$

To use collapsed Gibbs sampling ϕ is integrated out:

$$\begin{aligned} p(B|z) &= \int p(B|z, \phi)p(\phi)d\phi \\ &= \int \left(\prod_{i=1}^{N_B} p(b_i|z_i, \phi_{z_i}) \right) p(\phi)d\phi \\ &= \int \prod_{k=1}^K \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \prod_{w=1}^W \phi_{k,w}^{n_w|k+\beta-1} d\phi_k \right) \\ &= \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W} \right)^K \prod_{k=1}^K \frac{\prod_{w=1}^W \Gamma(n_{w|k} + \beta)}{\Gamma(n_{.|k} + W\beta)}, \end{aligned} \tag{37}$$

where $\Gamma(.)$ is the Gamma function. Next, θ is integrated out:

$$\begin{aligned} p(z) &= \int p(z|\theta)p(\theta)d\theta \\ &= \int \left(\prod_{i=1}^{N_B} p(z_i|\theta)\right) p(\theta)d\theta \\ &= \int \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_k^{n_k+\alpha-1} d\theta \\ &= \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \frac{\prod_k \Gamma(n_k+\alpha)}{\Gamma(N_B+K\alpha)}. \end{aligned} \tag{38}$$

Similarly,

$$\begin{aligned} p(B_{-i}|z_{-i}) &= \left(\frac{\Gamma(W\beta)}{\Gamma(\beta)^W}\right)^K \prod_{k=1}^{K} \frac{\prod_{w=1}^{W} \Gamma(n_{-i,w|k}+\beta)}{\Gamma(n_{-i,.|k}+W\beta)}, \\ p(z) &= \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \frac{\prod_k \Gamma(n_{-i,k}+\alpha)}{\Gamma(N_B-1+K\alpha)} \end{aligned} \tag{39}$$

The Gamma function satisfies $\Gamma(x+1) = x\Gamma(x)$ and $n_{.|k} = n_{-i,.|k}+2$. It follows that,

$$\Gamma(n_{.|k}+W\beta) = (n_i,.|k+W\beta+1)(n_{-i,.|k}+W\beta)\Gamma(n_{-i,.|k}+W\beta). \tag{40}$$

By substituting into Eq. 36 the final conditional distribution is obtained,

$$p(z_i = k|z_{-i}, B) \propto (n_{-i,k}+\alpha)\frac{(n_{-i,w_{i,1}|k}+\beta)(n_{-i,w_{i,2}|k}+\beta)}{(n_{-i,.|k}+W\beta+1)(n_{-i,.|k}+W\beta)}. \tag{41}$$

A.3 OLS derivation

The matrix notation of 3.3 is used. The sum of squared residuals can be expressed as follows,

$$S(b) = (y-Xb)'(y-Xb). \tag{42}$$

The global minimum can be retrieved by differentiating with respect to β and setting equal to zero,

$$0 = \frac{dS}{d\beta}(\beta) = \frac{d}{db}\left(y'y - b'X'y - y'Xbn + b'X'Xb\right) = -2X'y + 2X'X\beta \tag{43}$$

It is assumed that X is of full rank and $X'X$ us invertible. Then the least square estimator is given as,,

$$\beta = (X'X)^{-1}X'y \tag{44}$$

B Experiment

Topics	Top 20 words (Ethereum)
0	will, that, use, your, blockchain, crypto, like, what, network, project, one, transact, contract, other, time, platform, classic, secur, smart, us
1	new, token, wallet, launch, releas, announc, can, vitalik, commun, here, buterin, work, soon, first, app, pleas, talk, see, avail, develop
2	that, network, new, blockchain, day, one, will, time, updat, secur, pleas, go, block, peopl, could, contract, follow, ecosystem, issu, decentr
3	blockchain, will, platform, token, can, launch, project, develop, one, user, servic, build, first, crypto, see, announc, secur, microsoft, base, offer
4	new, post, updat, project, read, blog, check, develop, here, blockchain, support, vitalikbuterin, crypto, first, announc, scale, news, vitalik, follow, futur
5	token, will, blockchain, transact, use, updat, work, can, project, new, commun, go, here, week, get, time, year, make, team, come
6	blockchain, will, your, join, get, use, new, that, us, like, here, network, look, today, world, develop, work, dont, first, launch
7	your, smart, world, contract, what, start, platform, check, develop, get, peopl, new, go, dapp, build, decentr, futur, marketplac, run, learn
8	your, blockchain, use, can, contract, decentr, network, smart, work, classic, app, transact, token, support, user, what, see, want, chain, peopl
9	that, price, will, can, classic, token, like, eth, make, work, fork, develop, hard, crypto, other, here, etc, go, blockchain, trade
10	that, market, classic, get, network, one, develop, what, see, price, take, today, start, us, need, th, want, go, month, futur
11	will, join, what, come, like, see, token, th, here, next, live, open, meetup, peopl, today, vitalikbuterin, decentr, ico, interest, crypto
12	team, day, new, first, ico, thank, commun, well, next, develop, great, sale, futur, rais, million, what, excit, post, investor, th
13	blockchain, join, new, us, platform, come, support, develop, build, market, smart, one, other, year, day, trade, dapp, contract, live, start
14	will, that, token, crypto, support, market, can, get, time, exchang, trade, world, come, eth, next, classic, project, rippl, first, list
15	your, new, get, platform, decentr, first, exchang, support, that, ico, trade, what, wallet, classic, make, open, commun, mine, contract, go
16	that, contract, smart, blockchain, use, develop, platform, network, decentr, make, world, can, build, see, your, code, support, futur, technolog, via
17	that, will, use, can, token, market, smart, work, via, transact, eth, commun, wallet, run, one, project, user, take, secur, erc
18	token, market, day, your, eth, ico, launch, cap, here, start, announc, trade, platform, via, million, one, hour, network, address, follow
19	platform, news, articl, interview, featur, live, ethereumbas, publish, bank, analysi, ceo, take, digit, internet, ico, point, goe, medium, stellar, mobil

Topics	Top 20 Words (Litecoin)
0	will, that, market, crypto, new, payment, coin, your, us, day, rippl, price, make, want, btc, ltcfoundat, give, world, soon, first
1	like, here, can, today, support, come, buy, make, one, day, see, go, year, time, know, there, look, payment, futur, first
2	get, follow, peopl, accept, today, price, want, use, give, retweet, think, im, one, day, look, payment, thank, what, can, cash
3	like, coin, use, satoshilit, other, buy, your, currenc, futur, come, money, way, digit, make, exchang, next, thank, good, trade, gold
4	will, like, satoshilit, want, can, peopl, make, great, look, us, follow, dont, need, day, get, work, what, other, take, accept
5	that, live, will, market, time, futur, use, buy, price, get, btc, day, think, im, show, bet, trade, start, lee, charli
6	satoshilit, will, that, charli, lee, ltc, like, go, support, accept, ltcfoundat, time, he, trade, work, would, adopt, new, thank, follow
7	payment, ltc, your, accept, like, cash, use, crypto, satoshilit, wallet, btc, buy, card, start, also, run, see, bank, soon, support
8	price, analysi, will, technic, support, market, break, see, time, can, buy, bull, new, come, high, cash, look, chart, continu, like
9	new, guid, digit, surviv, west, wild, ebook, thrive, crypto, rt, currenc, understand, ultim, get, beginn, here, trade, want, buy, start
10	your, will, can, here, use, want, that, what, one, coin, need, good, time, help, also, everyon, us, work, adopt, retweet
11	will, ltc, day, go, next, price, first, here, what, year, block, network, segwit, can, mine, start, use, today, think, other
12	that, your, can, ltc, crypto, support, peopl, go, dont, great, like, one, here, what, there, wallet, coin, commun, think, payment
13	will, crypto, dogecoin, mine, get, live, best, cash, rippl, great, exchang, last, transact, wallet, time, via, today, support, money, currenc
14	that, get, your, go, dont, time, other, buy, ltc, year, new, see, also, start, would, trade, us, first, soon, futur
15	free, faucet, everi, best, get, minut, site, moon, daili, new, your, that, bonu, high, list, come, via, decid, payment, claim
16	that, satoshilit, get, your, ltc, crypto, time, buy, good, look, peopl, come, can, im, today, coin, new, next, great, dont
17	will, go, that, see, can, know, one, there, use, make, would, what, still, work, ltc, year, adopt, lightn, let, new
18	support, pleas, polit, canada, exil, will, your, peopl, news, say, sell, project, coin, test, sourc, store, code, number, design, keep
19	ltcfoundat, coinbas, foundat, partnership, trade, receiv, johnkim, via, york, deal, tokenpay, know, ufc, bank, ecurrencyhodl, jonnylitecoin, hope, regul, wait, profit

Topics	Top 20 words (Ripple)
0	rippl, that, effect, can, will, one, like, your, world, day, creat, get, other, dont, time, see, peopl, mani, chang, today
1	rippl, that, will, xrp, go, effect, your, market, get, like, take, us, price, exchang, compani, talk, global, real, still, xrapid
2	rippl, that, effect, world, use, look, market, new, price, what, global, see, chang, one, say, move, around, come, creat, take
3	rippl, hope, time, he, send, lot, tini, kennedi, man, person, effect, ideal, forth, robert, stand, improv, against, world, xrp, can
4	rippl, effect, will, xrp, that, go, new, could, get, day, crypto, like, what, take, say, week, kind, caus, start, other
5	rippl, time, what, send, hope, act, improv, anoth, forth, life, will, societi, appear, someon, day, here, need, like, stand, via
6	rippl, bank, xrp, payment, use, will, blockchain, technolog, new, crossbord, that, asset, ledger, compani, via, xrapid, digit, can, financi, network
7	xrp, go, one, new, make, will, next, get, compani, week, use, follow, here, also, first, xrptrump, crypto, should, run, two
8	rippl, effect, xrp, make, commun, can, go, good, think, invest, what, see, news, posit, trade, first, your, year, famili, there
9	rippl, make, dont, xrp, what, go, think, world, look, peopl, your, time, get, want, payment, know, even, bgarlinghous, way, real
10	rippl, creat, act, kind, thing, there, everi, small, end, rememb, logic, adam, scott, that, effect, can, simpl, one, care, endless
11	rippl, will, use, today, see, come, effect, make, us, broad, dont, thank, bank, work, last, support, great, day, peopl, compani
12	xrp, bank, payment, global, money, year, blockchain, market, like, partner, via, exchang, make, ledger, billion, currenc, could, system, transfer, custom
13	rippl, effect, across, us, go, could, via, new, financi, global, one, work, blockchain, also, here, industri, other, watch, system, see
14	rippl, that, make, one, will, like, money, payment, use, can, peopl, price, crypto, get, work, know, want, effect, thing, team
15	rippl, that, effect, will, can, your, world, what, other, peopl, creat, chang, work, use, live, year, caus, kind, thing, love
16	rippl, xrp, bank, time, one, payment, here, what, can, day, crypto, work, money, global, say, come, great, digit, first, new
17	rippl, that, make, will, your, like, use, would, think, today, effect, year, come, wave, follow, im, never, take, made, caus
18	rippl, will, crypto, use, that, other, global, coin, exchang, money, market, world, partner, digit, currenc, ceo, today, bank, sbi, trade
19	xrptrump, haydentiff, hodor, bankxrp, xrpnew, coinbas, rabbitkickclub, exchang, new, ad, joelkatz, xrphodler, list, add, ckjcryptonew, sourc, ripplexrp, exclus, justmoon, regul

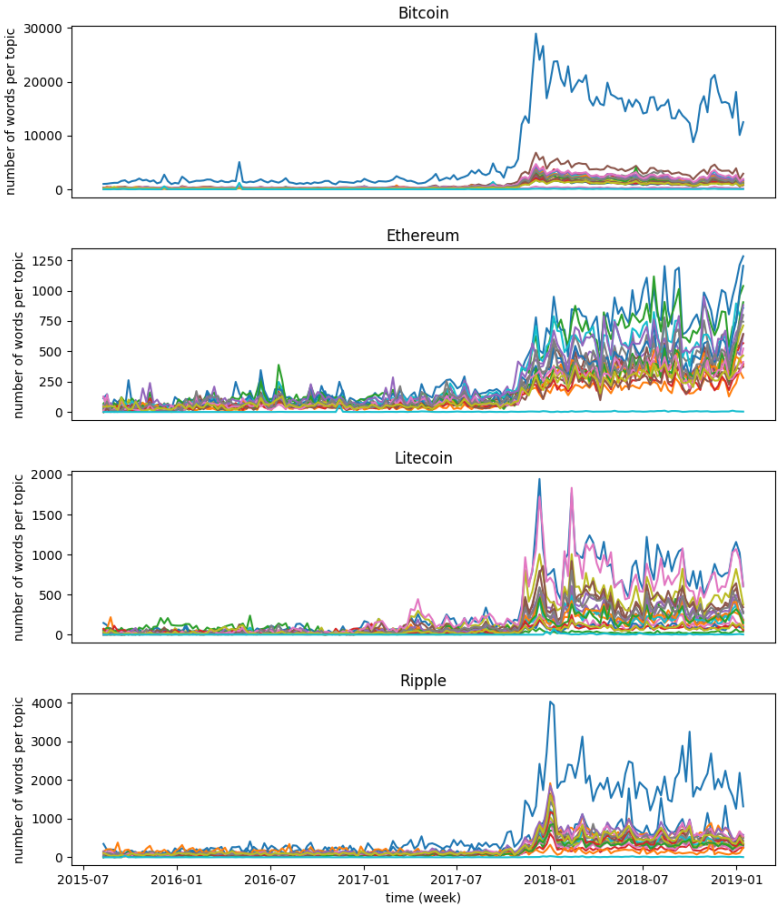

Figure 8: The topic-activity is the number of words sampled per topic. Topic-activity drastically changed with the burst of the crypto-bubble at the end of 2017.

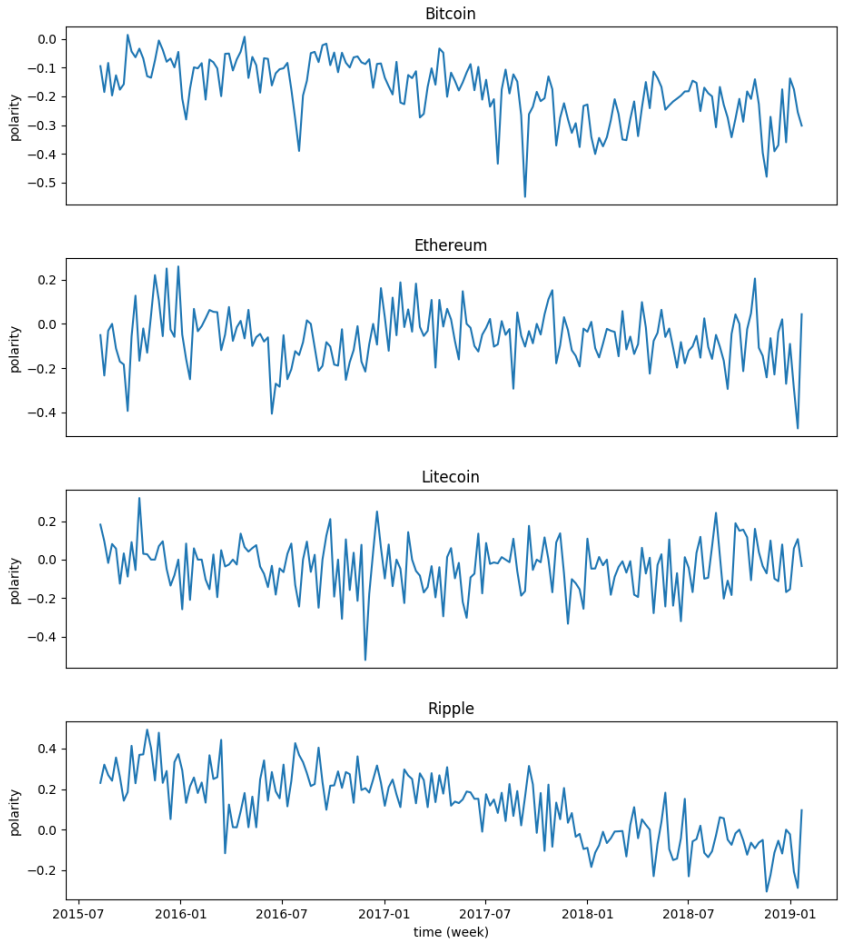

Figure 9: The polarity is the average difference of positive and negative words per tweet. Tweets regarding Bitcoin have a more negative wording than tweets regarding altcoins.