CRYPTO ASSET TAXONOMY CLASSIFICATION AND CRYPTO NEWS SENTIMENT ANALYSIS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OZAN KÖSE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2020

Approval of the thesis:

CRYPTO ASSET TAXONOMY CLASSIFICATION AND CRYPTO NEWS SENTIMENT ANALYSIS

submitted by **OZAN KÖSE** in partial fulfillment of the requirements for the degree of **Master of Science** i**n Computer Engineering, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of the Department, **Computer Engineering** _____

Prof. Dr. Pınar Karagöz
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Assist. Prof. Dr. Pelin ANGIN
Computer Engineering, METU _____

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Assist. Prof. Dr. Adnan ÖZSOY
Computer Engineering, Hacettepe University _____

Date: 03.09.2020

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Ozan Köse

Signature :

ABSTRACT

CRYPTO ASSET TAXONOMY CLASSIFICATION AND CRYPTO NEWS SENTIMENT ANALYSIS

Köse, Ozan
Master of Science, Computer Engineering
Supervisor: Prof. Dr. Pınar Karagöz

September 2020, 147 pages

There are over 1900 cryptocurrencies trading in cryptocurrency exchanges as of June 2020 and the number is rapidly growing. In the current crypto scene, cryptocurrencies are seen as investment vehicles by many, yet every crypto asset is designed to operate in a specific sector within a pre-defined business model. There are many characteristics that differentiate one crypto asset from another and there are numerous internal and external factors that affect each crypto asset differently depending on these characteristics. Crypto investors can leverage these factors and characteristics and use these indicators to create different trading strategies. In this thesis, to guide the investors, we classify the crypto assets under various characteristics and provide sentiment analysis on crypto related news. As our first major task, we focus on automated annotation of the cryptocurrencies in terms of sector, transaction anonymity and asset type through the public information. To this aim, we generated an annotated dataset by collecting information from various sources. The collected dataset includes cryptocurrency descriptions annotated with sector, asset type and transaction anonymity labels. For this task, we utilized a divide-and-conquer supervised learning approach and compared its performance against several supervised learning algorithms for the three aspects. As our second

major task, we focus on automated sentiment analysis of crypto asset related news on news title, summary and content. For this, we have generated an annotated news dataset that is collected from various public news sources. We use this dataset to fine-tune a successful neural network model that is trained on financial sentiment analysis task and compared its performance to various dictionary-based methods.

Keywords: Cryptocurrency Document Classification, Crypto News Sentiment Analysis, Sector Classification, Asset Type Classification, Transaction Anonymity Classification

ÖZ

KRİPTO VARLIK TAKSONOMİ SINIFLANDIRMASI VE KRİPTO HABER DUYGU ANALİZİ

Köse, Ozan
Yüksek Lisans, Bilgisayar Mühendisliği
Tez Yöneticisi: Prof. Dr. Pınar Karagöz

Eylül 2020, 147 sayfa

Haziran 2020 itibariyle 1900'den fazla kripto para birimi ticareti yapılmakta ve bu sayı hızlıca artmaktadır. Mevcut kripto ticareti senaryosunda, kripto para birimleri birçokları tarafından yatırım araçları olarak görülse de, her kripto varlığı önceden belirlenmiş bir iş modeli içinde belirli bir sektörde çalışmak üzere tasarlanmış durumdadır. Bir kripto varlığını diğerinden ayıran birçok özellik vardır ve bu özelliklere bağlı olarak her bir kripto varlığını farklı şekilde etkileyen çok sayıda iç ve dış faktör bulumaktadır. Kripto para yatırımcıları bu faktörlerden ve özelliklerden yararlanabilir ve bu göstergeleri farklı ticaret stratejileri oluşturmak için kullanabilirler. Biz bu tezde, yatırımcılara rehberlik etmek için kripto varlıklarını çeşitli özellikler altında sınıflandırdık ve kripto paralarla ilgili haberlerde duygu analizi yaptık. İlk büyük görevimiz olarak, açık kaynaklardan topladığımız veriler yoluyla kripto para birimlerinin sektör, işlem anonimliği ve varlık türü açısından otomatik olarak sınıflandırılmasına odaklandık. Bu amaçla, çeşitli kaynaklardan bilgi toplayarak işaretlenmiş bir veri kümesi oluşturduk. Toplanan bu veri kümesi, sektör, varlık türü ve işlem gizliliği etiketleriyle, kripto para birimi hakkındaki yazılı açıklamaları, tanımları içermektedir. Sınıflandırma problemini çözmek için, bölme ve fethetme odaklı bir denetimli öğrenme yaklaşımını kullandık ve modelin

performansını bu üç özellik için birkaç denetimli öğrenme algoritmasıyla karşılaştırdık. İkinci büyük görevimiz olarak, kripto varlıklarla ile ilgili haberlerin haber başlığı, özeti ve içeriği üzerine otomatik duygu analizine odaklandık. Bunun için, öncelikle çeşitli açık haber kaynaklarından toplanan işaretlenmiş bir haber veri seti oluşturduk. Bu veri setini, finansal duygu analizi görevinde başarılı olmuş bir sinir ağı modeline ince ayar yapmak ve performansını çeşitli sözlük tabanlı yöntemlerle karşılaştırmak için kullandık.

Anahtar Kelimeler: Kripto Para Belge Sınıflandırması, Kripto Haber Duygu Analizi, Sektör Sınıflandırması, Varlık Türü Sınıflandırması, İşlem Gizliliği Sınıflandırması

To my beautiful Mom…

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Dr. Pınar Karagöz for her guidance, support and positive attitude towards me for leading and guiding my MSc thesis work in a constructive manner.

I would like to thank to my employers Dr. Gökçe Phillips and Richard Phillips for giving me the opportunity, guidance and support for this thesis study.

I would like to thank my mother Hatice Ünal and my father-in-law Fatih Ünal for all their support and struggle for bringing me up to the level where I am today. I would also like to give my specials thanks to my beloved Bengüsu who has been listening all my complaints throughout my work and leading me to the light whenever I got lost at some stages. Finally, I would like to thank my friends for supporting me and not complaining about the events that I have missed.

This work has been supported by CryptoIndexSeries and TUBITAK with grant no 3191547.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

Cryptocurrencies have been popular with then infamous Bitcoin and experienced considerable peaks when Bitcoin traded around $19.000 in December 2017. Since then, the number of cryptocurrencies in market grew exponentially and many whitepapers have been published to propose yet another crypto asset serving a particular purpose (e.g. as a means of payment, to pay for a service or a good, as a security that gives right to a digital/non-digital asset) [1] [2].

Figure 1.1. Crypto Asset Taxonomy (Flag is used for indicating binary attributes. Crypto assets either categorized as having such attribute or not.)

Whilst the crypto asset projects are still under development, traders allowed to buy/sell crypto assets in Crypto Exchanges. Today, cryptocurrencies are becoming more popular day by day among investors as an investment tool due to possible high returns in short term. As the attention grows, investors are looking for new investing strategies to reduce their risks and increase their incomes [2]. In analysing crypto assets, it is important to be able to comprehend the positioning of the asset from many different perspectives: Technical, regulatory, Token economics, Business and so on. To this end, crypto asset taxonomy works emerged.

In the crypto asset literature, there are several studies on crypto asset taxonomy [3][4][5][6]. Advantages of establishing a taxonomy with the aim of understanding technical, regulatory, financial and business characteristics of crypto assets are also emphasised in such studies, yet none of these works provide an automated way of classifying cryptocurrencies or constructing these taxonomies. We believe an automated detection and integration of crypto asset taxonomy to the digital platforms could lead to interesting and advantageous investment opportunities for trading parties. The work explained in this thesis may lead researches for automatically detecting, classifying and integrating such crypto asset taxonomies to digital platforms.

In this work, we consider the crypto asset taxonomy skeleton given in Figure 1.1 as the reference. This is a proprietary crypto asset taxonomy work in-progress proposed by CryptoIndexSeries after analysing crypto asset literature and various crypto assets. The taxonomy classifies crypto assets from four different perspectives: Technical, Financial / Regulatory, Business and Token Economics. Methodology used for establishing this taxonomy is out of this thesis's scope.

For this thesis, we focus on one characteristic from each perspective given in Figure 1, except *Token Economy* perspective as it was not finalised during the course of this work. For this thesis, we choose; *Sector* information from *Business* perspective, *Asset Type* information from *Financial/Regulatory* perspective and *Transaction Anonymity* information from *Technical* perspective for automated classification of

crypto assets. In the following three paragraphs, we have explained our motivation for selecting these characteristics from each perspective.

In traditional financial markets, sector index is a well-known benchmark. Companies are grouped under sectors and sectoral indices are used as benchmarking tools to see how well a particular asset is performing with respect to others in the same sector [7]. As of June 2020, there are over 1900+ cryptocurrencies trading across 300+ exchanges globally. Some of these crypto assets will start operating in their defined sectors as soon as their product development stage is over and the actual product is live where as some has already started operating. The sectoral classification of crypto assets, together with sectoral indices help investors establish new investment strategies: examine crypto assets on a per-sector basis, take impact investment decisions by focusing on specific sectors only or diversify their portfolio by distributing the risk.

Knowing the token type of cryptocurrency is crucial for investors. Based on the guidance reports by Financial Supervisory Authority (FINMA) [8] for Switzerland and European Securities and Markets Authority (ESMA) [9] for EU, there are mainly three types of crypto assets: payment type, utility type and security type (also named as asset type). Utility tokens are defined for purchase/sales of goods and services and utilised mainly in predefined platforms. Payment tokens are utilised/created as a means of payment and not limited to a certain platform where as security tokens give investors the right for a profit share or a percentage of a real/digital asset, etc. Current guidelines by Security Exchange Commission (SEC) in the USA, FINMA in Switzerland and ESMA in the EU suggest that the security token holders will be liable for tax when the regulations are established in countries. Therefore, investors should also be aware of the token types at the time of the purchase to hedge themselves against such risky situations6 . In addition, utility tokens are designed to be used only in specific platforms like games, websites etc. Understanding this could lead investors to conduct research on these platforms for better investment options.

Another important feature for cryptocurrencies is the transaction anonymity of the cryptocurrency. Many people believe that transactions made through Bitcoin Blockchain are completely hidden [5]. However, the truth is that although Bitcoin Blockchain does not show the sender or the recipient information, it is possible to see the public wallet IDs of the parties. In other words, Bitcoin is a pseudonymous cryptocurrency [10]. There are other cryptocurrencies in the market that provides complete anonymity such as ZCash [11] and Monero [12]. This feature is also important for people who want to buy and use cryptocurrencies that are not traceable and completely anonymous.

Even though traditional markets are not volatile as cryptocurrency market, they are open to market manipulation with news [13]. There are many researches in literature for developing new investment strategies for traditional financial markets using news sentiment analysis. This can also be applied to cryptocurrency market if we were able to analyze cryptocurrency news.

1.2 Proposed Methods and Models

In this rapidly growing market, tracing each new cryptocurrency and its mentioned characteristics is hard and time consuming. In this work, we have approached this problem as a text classification problem and aimed solving this by using supervised learning algorithms. As the basic requirement of this method, we collected a dataset of documents and descriptions from online public resources. The collection is annotated by domain experts for sector, asset type and transaction anonymity labels. Additionally, we have also proposed an annotated news dataset and crypto news sentiment analysis model.

1.3 Contributions and Novelties

Our contributions to this field of study are as follows:

- First structured dataset annotated by sector, asset type and transaction anonymity labels,
- First research on literature on cryptocurrency field for sector, asset type and transaction anonymity classification using supervised classification algorithms,
- First structured and annotated dataset for cryptocurrency news sentiment analysis.
- Examining different variations of accumulating sentence level sentiment analysis to long text's sentiment analysis.

1.4 Outline of the Thesis

The rest of this thesis organized as below:

- In Chapter 2, we will discuss previous works in literature related to works in our thesis,
- In Chapter 3, we will provide some background information on cryptocurrency sector, asset type and transaction anonymity types with neural networks used in experiments of this thesis study,
- In Chapter 4, we will demonstrate our dataset and model of crypto currency document classification,
- In Chapter 5, we will demonstrate our dataset and model of crypto news sentiment analysis,
- In Chapter 6, we will present and discuss our experiments related to document classification and news sentiment analysis,
- Finally, in chapter 7, we will conclude our work and discuss possible future studies.

CHAPTER 2

RELATED WORK

2.1 Classification Towards Taxonomy

In this section, we have analyzed some studies regarding crypto asset taxonomy and document classification literature. At the time of the writing, we were able to find neither a publicly available dataset nor a research related to automated classification of cryptocurrencies with respect to sector, asset type and transaction anonymity. Most of the research in the literature regarding cryptocurrencies either focuses on forecasting prices [14][15][16][17] or automated trading strategies [18][19][20]. However, there are some taxonomy related works in the literature regarding cryptocurrencies.

Lausen proposed a taxonomy of crypto assets for deciding which ICOs (Initial Coin Offerings) would be subject to regulations. The proposed taxonomy investigates crypto assets in fourteen (14) different dimensions of which each having two (2) to four (4) different characteristics. Lausen suggests that these characteristics could be used in three (3) ways. As the first way, the regulators can decide which crypto assets will be regulated on which characteristics. Secondly, the crypto asset companies could use this taxonomy to use features that would result in best profit. Lastly, the audience can gain insights and extract common design patterns that most crypto assets employ [3].

Ankenbrand et al. proposed a taxonomy on crypto assets with fourteen (14) different attributes supported by crypto asset literature. These attributes include claim structure, technology, underlying, consensus-validation mechanism, legal status, governance, information complexity, legal structure, information interface, total supply, issuance, redemption, transferability and fungibility [4].

Oliveira et al. collected hundred and thirteen (113) crypto asset projects from various online resources and interviewed sixteen (16) representatives to propose taxonomy to be used for token classification and decision making. Their taxonomy includes class, function, role, representation, supply, incentive system, transactions, ownership, burnability, expirability, fungibility, layer, and chain dimensions which each contains multiple characteristics [5].

Tasca et al. introduces a multi-layer taxonomy for the categorizing crypto assets with different logical and functional components. They indicate that crypto assets need a way of standardizing for gaining global adaptation and compatibility, creating cost-effective and cross-industry solutions [6].

Papers and research projects explained above focus on establishing a crypto asset taxonomy. On the contrary, the research conducted and explained in this thesis employs machine learning techniques for assessing characteristics to crypto assets in an automated manner. In the rest of this section, we have analyzed recent studies regarding document classification.

Kim et al. used convolutional neural networks for sentence classification task, their model improved most of the recent (at the time it published) classification models accuracies by %5-6 on six (6) public datasets. They concluded a simple CNN with one layer of convolution performed well even with little parameter tuning [21]. Zhang et al. used convolutional neural networks with characters as input features. They concluded their work by indicating character level CNNs are powerful yet their usage highly dependent on dataset size, curated text and choice of alphabet [22].

Ying et al. proposed hierarchical attention networks (HANs) for document classification. Their approach improved state-of-art models' performance on known datasets by %8-10 on accuracy. Their model produces document vectors by aggregating vectors for important words within sentence and then aggregating vectors for important sentenced within documents [23]. Johnson et al. used word-level pyramid convolutional neural networks for training deep neural networks on

large dataset setting. Their model achieved better accuracies than state-of-art methods on six (6) benchmark datasets [24].

Howard et al. introduced ULMFit for efficiently using transfer learning approach on any NLP tasks. They have also introduced couple of methods for to prevent catastrophic forgetting [25]. Devlin et al. introduced BERT, a deep transformer network for language modelling and general-purpose classification task. BERT achieved many state-of-art results on many benchmark datasets [26]. Yang et al. proposed XLNet, a generalized autoregressive pretraining method that uses permutation language modelling and beat BERT on various text classification tasks. XLNet achieved better accuracies than BERT in some text classification and sentiment analysis benchmark datasets [27].

As mentioned, recent state-of-art scores and studies are based on neural network architectures. As indicated in many papers, neural networks (especially deep ones) need large number of training instances and resources for effectively tuning the model parameters. Since we lack on both aspects, we have focused on less complex models in this thesis for document classification. The neural network models we focus on this thesis are also novel approaches and have achieved better performance than state-of-are methods on benchmark datasets at the time they are proposed. Those neural network models and their architectures are explained in detail at Section 3.2.

2.2 News Sentiment Analysis

In this section, we have analyzed some studies regarding sentiment analysis in crypto asset literature and financial literature. Cerda et al. used sentiment analysis of some crypto influencers tweets to see if it effects the Bitcoin price. They have analyzed nine thousand and one hundred thirty-six (9136) Tweets from one hundred and thirty-five (135) crypto influencers using SentiWordNet's SentiWord tool to show that influencers does affect the price of Bitcoin [28].

Colianni et al. used sentiment analysis scores of tweets as features for predicting if Bitcoin price will increase or decrease. They randomly collected tweets that includes terms regarding to Bitcoin and after some pre-processing, using NLTK's sentiment analyzer they extracted sentiment scores per word in tweets and used for training Naïve Bayes, Logistic Regression model and Support Vector Machines to predict Bitcoin price direction. They achieved an accuracy exceeding %90 on day-to-day and hour-to-hour market movement predictions [29].

Nasekin et al. collected approximately 1 million messages from StockTwits of which 38% are labeled as *Bullish* (indicates that market is positive), %7 are labeled as *Bearish* (indicates that market is negative) and rest are unlabeled. They excluded unlabeled messages from their dataset and applied RNNs to predict if given message is *Bearish* or *Bullish*. They achieved 90% F1 score on predicting *Bullish* messages but failed to achieve same performance on predicting *Bearish* messages concluding data imbalance as the main problem [30].

Rouhani et al. analyzed general sentiment of Bitcoin, Ripple, Cardano, Ethereum and Litecoin related tweets. They collected a dataset from Twitter that mentions above coins and annotated the tweets using WordNet on positive, negative and neutral labels. Then by using this dataset they trained and compare five (5) different supervised learning methods on tweet sentiment analysis. In their work, SVM achieved best results for all analyzed coins [31].

Sentiment analysis works explained above focus on tweets and short-length messages. This thesis on the contrary, in addition to sentences, focuses on published news articles and their title, summary and full content's sentiment analysis. This thesis also experiments various accumulation procedures for assessing sentiment labels to long texts from the analysis of their sentences.

In the remainder of this section we have analyzed some state-of-art models in literature. As a famous lexicon based method in financial domain, Loughran and McDonald generated a dictionary by analyzing U.S. Security and Commissions

portal from 1994 to 2008 that consists of financial lexicons that are categorized under six (6) different labels including positive and negative [32].

More recent state-of-art research on financial domain focuses on Neural Networks. Maia et al. achieved state-of-art results on Financial PhraseBank [33] by using LSTM network with text simplification [34].

Sousa et al. collected five hundred and sixty-two (562) financial news articles from various resources and four (4) researchers from the team annotated articles on positive, negative and neutral labels. On experiments performed with 10-Fold Cross Validation upon Naïve Bayes, SVM, TextCNN and BERT-based classifier; BERT-based classifier as we used for news sentiment analysis in this thesis showed best performance by large margin [35].

CHAPTER 3

BACKGROUND

3.1 Crypto Asset Taxonomy

In this section, types of cryptocurrency characteristics we have focused on this thesis are briefly explained.

3.1.1 Sector

Sector of a crypto asset indicates the industrial segment in which that crypto asset will operate. Usually, this information is indicated in the whitepaper of crypto assets. We briefly introduce the seventeen (17) sectors that are referred in whitepapers.

Finance. The crypto assets in this sector mainly focuses on providing secure financial data management, healthier information on credit scoring, minimalizing fees on money transferring and micropayments [36].

Technology. The primary focal point of the assets in this sector is providing more secure, unmodifiable storage infrastructures, more secure IOT communication between devices and the cloud, unmodifiable and unbreakable digital contracts that signed between parties (Smart Contracts) [37].

Entertainment. The main aim of crypto assets in this sector include, but not limited to, increasing the content creators earnings in various entertainment industries, diminishing the fees paid to third parties, securing intellectual property rights of movie and game producers, and allowing in-game content purchases without any commission to game stores [38].

Content and Advertisement The crypto assets in this sector aim to provide smarter advertisement to right focus groups, faster payments to content creators, prevent content theft and awarding contents with the votes of real audience [39].

Education. The main focus of the use of crypto assets in this sector is to eliminate third parties by allowing instructors to be paid directly by students, to improve digital library infrastructure in schools and to verify credentials that are claimed by job applicants in resume [40].

Healthcare and Pharma. The crypto assets in this sector mainly focus on, more secure way of health data sharing by anonymizing users and supporting health data interoperability works [41].

Energy. In this sector, crypto assets are utilized to prevent energy frauds by enabling immutable energy data management and energy trading between individuals [42].

Real Estate. Cryptocurrencies in this sector aim to provide easy and fast property selling, renting and buying, holding shares on properties and property investment management [43].

Charity and Aid. There have been controversies about charity organizations regarding the lack of trust due to donation abuse. With blockchain technology, cryptocurrencies in this sector aim to provide more secure and transparent donations [44].

Travel. This industry suffers from a lot of third parties forcing high commission on fees and causing high travel costs. Payments with cryptocurrencies would eliminate the middleman thus reduce the fees and immutable booking records also would also prevent corrupted booking data thus financial losses [45].

Transportation. Cryptocurrencies in this sector aim to provide more secure and reliable way of tracking transportation assets. Reliable data on transportation assets helps owners better plan and organize assets. In addition, tokenization of assets would open opportunities to partial ownership/rental of vehicles [46]**.**

Food Production. Reliability and transparency are two of the main problems in Food Production industry. Consumers would like to know where their food comes from and how healthy it is. The crypto assets in this sector, using blockchain technology, aim to solve these issues by providing more reliable food supply chain and improved quality with incentive based approach [47].

Telecommunication. The cryptocurrencies focusing on this industry aim to provide solutions to Telecom companies for identity theft/fraud, fees paid to third parties and invalid/misplaced communication fees [48].

Cannabis. This industry suffers from lack of coordination between growers, processors, retailers, and consumers and compliance to allowed dosages per person. Cryptocurrencies in this sector focus to solve such problems using blockchain industry [49].

Retail. This sector has been through a rapid change where day by day consumers would like to have access to products rapidly and easily online. Yet, this raises concerns for fraud, identity theft and sub-optimal quality products. By integrating blockchain technology on every step from the start of supply chain to delivery of the products to end users, crypto assets in this sector aim to be utilized in minimizing the aforementioned risks and concerns [50].

Insurance. The cryptocurrencies in this sector aim to be used in solving increased concerns regarding insurance fraud, identity theft, risk management and easy delivery [51].

Art. Tokenization of assets would allow individuals to claim part/full ownership of art pieces. This would help contributing to artists in sector by easily delivering payments and prevent art fraud [52].

3.1.2 Asset Type

Asset type, also referred as token type, is mainly grouped under three (3) categories as indicated in ESMA Report[9] and FINMA report[8] and represents the main functionality of the token. There are hybrid tokens, but for the sake of simplicity, they are not considered within the scope of this thesis [53].

Payment Token. These tokens are used as an alternative to real money in digital asset world. Payment tokens can be used to buy goods/services.

Security Token. These tokens represent ownership rights on real or digital assets like real estates, real/digital art pieces, company shares, etc. Security tokens are like equities and bonds in traditional financial markets.

Utility Token. These tokens represent access rights to goods and services provided by companies but not ownership rights.

3.1.3 Transaction Anonymity

Transaction anonymity of a cryptocurrency indicates what level of security measure is used on its blockchain. There are three (3) types of transaction anonymity classes as briefly explained below [54].

Pseudonymous. Most of the cryptocurrencies fall under this category. For this type of cryptocurrencies, even though transactions can-not be linked to real people or organizations directly, all transactions (having public wallet IDs) are transparent and accessible to everyone.

Anonymous. Compared to pseudonymous, the transactions of cryptocurrencies under this category are completely anonymous. IP obfuscation and transaction encryption are popular techniques for achieving complete anonymity.

Selective Transparency. Cryptocurrencies in this category leaves the decision of transaction anonymity to its users. Users can select which of their transactions should be anonymous or pseudonymous.

3.2 Neural Networks

In this section, neural network architectures referred in this thesis are briefly explained.

3.2.1 fastText

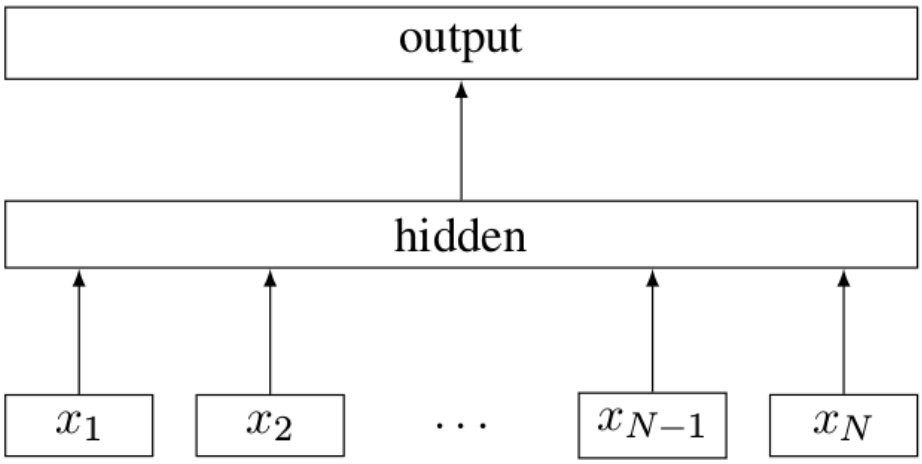


Figure 3.1. Architecture of fastText for a sentence with N n-gram features [55]

fastText model is designed to be fast and effective baseline method for sentence and document classification by Facebook AI Research team [55]. Architecture of fastText is given in Figure 3.1, each word in input text is converted to n-gram features and then n-gram features embedded and averaged to generate the hidden unit. This hidden unit is connected to a linear unit like SoftMax or hierarchical SoftMax to perform classification tasks.

3.2.2 TextCNN

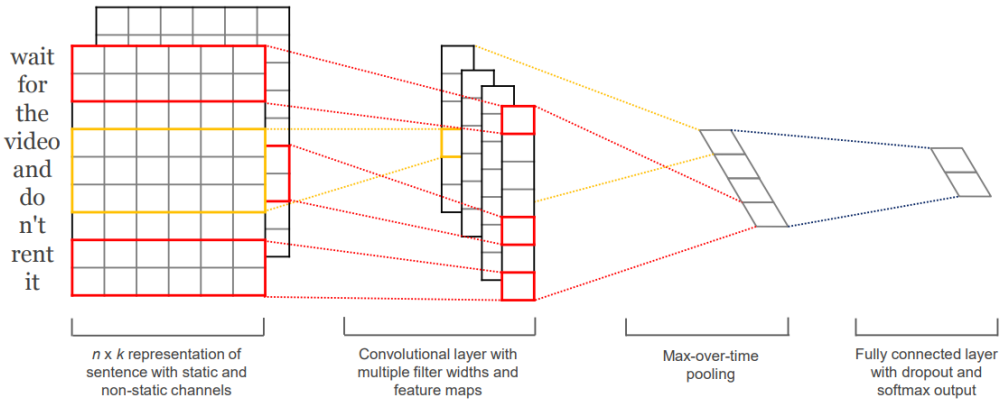


Figure 3.2. Architecture of TextCNN model [21]

TextCNN model employs Convolutional Neural Networks for text and sentence classification. In convolution phase, a 1D convolutional filter applied to window of selected filter size words to generate convolutional embeddings. The aim of this convolution is to extract context information that depends on window size that filter is applied on (E.g. For filter size 3, feature generated for filter includes embedding information of 3 words). In paper, model uses multiple size filters (100 filters per 3,4,5 sizes in paper) and generates multiple convolutional embeddings. Then max over time pooling which is a max pooling operation on 1D space is applied to feature maps to generate single representative feature vectors for each filter. These pooled features from different filters are concatenated and connected to full connected layer and SoftMax for classification. In addition, authors also applied dropout to pooled feature vector by randomly zeroing out some parts of it with respect to the selected dropout portion.

3.2.3 RCNN

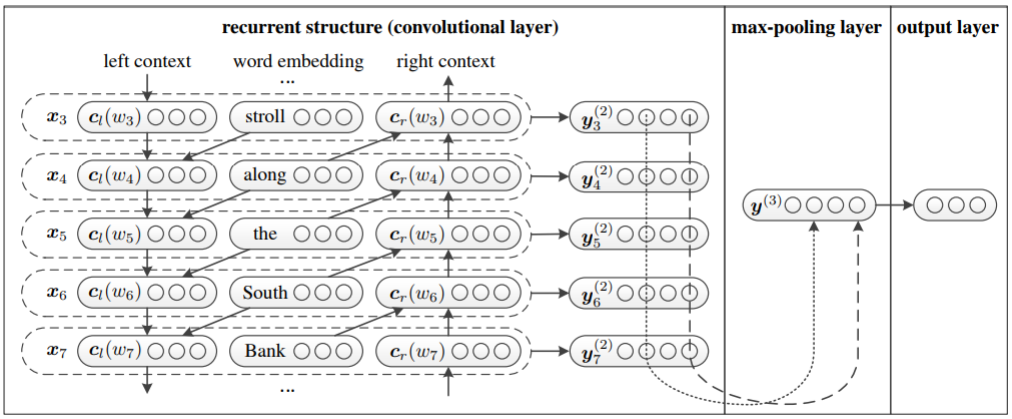


Figure 3.3. Architecture of RCNN model [56]

Recurrent Neural Networks are successful models to capture contextual information since they are storing information regarding previously seen text in fixed sized hidden layers [57]. Yet, as the later occurring words are more dominant than the words seen earlier, the model is biased. As explained in Section 3.2.2, Convolutional Neural Networks are prune this bias problem, yet they are dependent on hyperparameter filter sizes and they may fail to detect context relations between two words if the space between words are larger assigned filter size. Lai et al. in their paper combined RNN and CNN architectures to introduce Recurrent Convolutional Neural Networks (RCNN) to eliminate mentioned weaknesses of both models [56].

As seen from Figure 3.3, model uses a recurrent structure as a convolutional layer. It first captures right and left context information per word by using a bi-directional recurrent structure. Then for each word, right context, left context and embedding of word is concatenated to represent word-based context feature vector. After applying a linear layer with tanh activation function, features are passed through a max pooling layer to generate fixed length vector for input text. The last part of architecture is a linear layer with a SoftMax activation function to decide on predicted label [56].

3.2.4 RNN-Attention

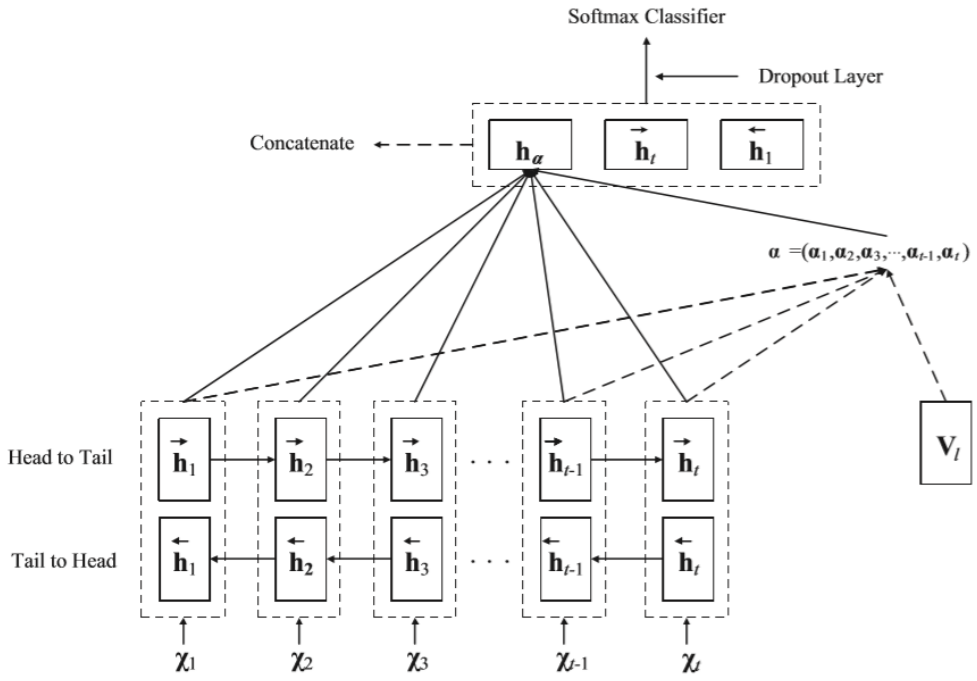


Figure 3.4. RNN-Attention Architecture [58]

Attention mechanism on Recurrent Neural Networks is first introduced in paper of Bahdanaou et al. [59] for aligning input words to output words in sequence to sequence neural networks for translations. Model proposed is a sequence to sequence RNN architecture which given an input text, outputs another text. Sequence to sequence models are commonly used for question answering and translation tasks. The attention mechanism proposed for overcoming the shortcomings of traditional RNNs with longer sequences and aligning input words with output words for translation tasks. This is achieved by learning an attention similarity score for each word in the input vector indicating how much important that word is to the meaning of its sentence. This attention weights are expected to be high for important words in text.

The architecture given in Figure 3.4 is adaptation of RNN with attention mechanism for text classification tasks proposed by Du et al. [58]. In proposed model, they have used bidirectional LSTM to learn hidden representations of each word from both directions. These hidden representations of words are used to learn attention weights. Then hidden representations of words are concatenated with weighted sum of all hidden representations with respect to attention weights to create context vector for input text. This context vector is forwarded to SoftMax classifier to perform text classification.

3.2.5 Transformer

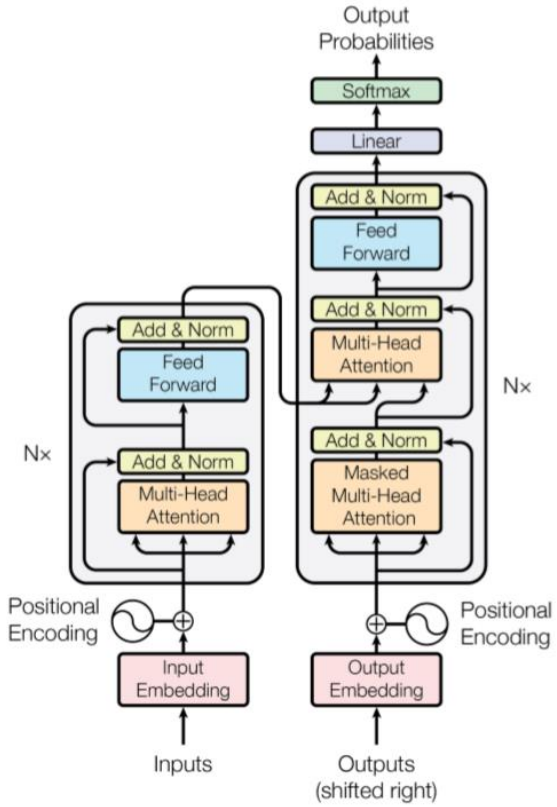


Figure 3.5. Transformer Architecture [60]

Transformer architecture is similar to RNN-Attention model and proposed as a sequence to sequence architecture thus includes encoder and decoder parts. For the scope of classification tasks, we will only focus to the encoder part. Different from traditional RNN encoder-decoder architecture, Transformer model replaces RNNs with multi-headed self-attention layer and feed forward mechanism. Proposed architecture includes 6 identical Transformer layers. Within self-attention layers three matrices; Key, Query and Value matrices are learned. For generating representations, for each token's Key vector is multiplied by all other token's query vectors and a similarity score between tokens is obtained. Then these similarity scores are used to weight Value vectors for respective tokens. When there are multiple matrices initialized per encoder, this is called Multi-Headed Attention

mechanism. These representations obtained from multiple self-attention matrices are concatenated and forwarded to feed-forward neural network [60].

As the RNNs in nature are sequential networks and since they are eliminated from Transformer architecture, model becomes much easier to parallelize on GPUs. For text classification task, output of encoders connected to dense layer followed by a SoftMax classifier.

3.2.6 BERT

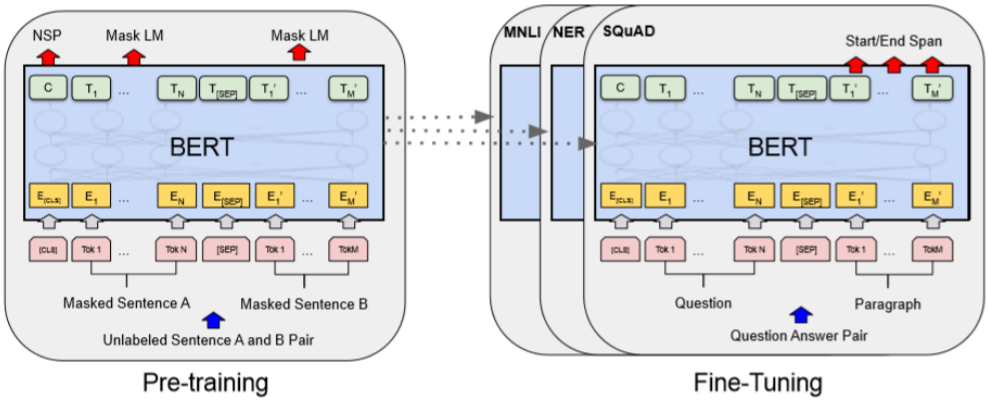


Figure 3.6. BERT Architecture [26]

BERT is a language representation model consist of multiple Transformer (Section 3.2.5) encoders stacked on top of each other. BERT model is pretrained with two (2) unsupervised tasks. First by masking %15 of tokens, model is trained to predict masked tokens and then model is further trained to predict on next sentence prediction task [26]. For classification tasks, a dense layer is added hidden state of last [CLS] token as applied by FinBERT [61].

CHAPTER 4

CLASSIFICATION TOWARDS TAXONOMY

4.1.1 Data Collection

Our dataset is generated by collecting information from several sources on internet including websites of cryptocurrencies. There are two main sources of information; first one is descriptions of cryptocurrencies scraped from public online sources and the other one is technical documents of cryptocurrencies called whitepapers.

Web scraper periodically checks each website for new coins and it extracts various metadata information for each cryptocurrency including its symbol, name, website URL, logo, description, whitepaper URL, sector, asset type and transaction anonymity labels. If metadata scraped from websites includes a valid whitepaper URL, it is downloaded and textual content is stored with cryptocurrency record.

Table 4-1 Data Annotation

Coin ID	Symbol	Sector	Transaction Anonymity	Asset Type	Text
1	BTC	Finance	Pseudonymous	Payment	…
1	BTC	Finance	Pseudonymous	Payment	…
1	BTC	Finance	Pseudonymous	Payment	…
1	BTC	Finance	Pseudonymous	Payment	…
2	BLAZR	Cannabis	-	-	…
2	BLAZR	Cannabis	-	-	…

Metadata information scraped from various providers are mapped to a single cryptocurrency. This mapping is done by checking cryptocurrency symbol/name equality and supported by manual checks.

Each instance in our dataset includes a textual information that is uniquely mapped with a cryptocurrency identifier. Instances are annotated in cryptocurrency level by domain experts and labels are assigned to instances with respect to their cryptocurrency ID. An example dataset annotation is given at Table 4-1.

4.1.2 Pre-processing

Pre-processing textual data obtained from different resources includes several common steps. Additionally, we applied several specific cleaning steps for whitepapers.

Since most whitepapers contain a large amount of content that is not relevant to the cryptocurrency, such as investor distribution, disclaimer, etc., we have cropped whitepaper contents to capture valuable information about cryptocurrency characteristics. For cropping, we skipped the first couple of pages that has fewer words than half of the average number of words per page and took %20 of the rest. With this method, we aimed to skip pages that contains few to no words and extract the content of introductory pages.

Since whitepapers are written by companies that are from various nationalities, some whitepapers included non-English names and words (e.g. Cyrillic). Therefore, whitepaper text cleaning includes non-English word elimination. In addition, some whitepapers are made of only images, but they also contain some text like email addresses, names…etc. hidden in its papers. Therefore, whitepapers that have less than one hundred (100) meaningful tokens have eliminated from the dataset.

As an additional pre-processing step, on the basis of our observation that some of the textual content are computer generated texts, we have filtered and eliminated those instances by applying text pattern matching.

After this cleaning phase, all textual content is tokenized. Punctuations, digits and stop words are removed from tokens. Stop words list is extended for each coin with its name and symbol, since their description is likely to include them and does not

give any valuable information. Lastly, tokens that are shorter than three (3) characters are eliminated and all tokens are converted to lowercase.

In English, words that have the same meaning can take different forms with respect to its tense and parts-of-speech. In order to overcome this problem, a text normalization procedure is required. For normalization of textual data, we have decided to use three (3) different approaches and generated three (3) versions of dataset.

- **Lemmatization** is a normalization technique that converts words to their base dictionary format by using the parts-of-speech of the words [62]. For lemmatization, we used Natural Language Toolkit's [63] WordNet lemmatizer which under the hood uses Princeton's WordNet lexical database [64].
- **Stemming** is a normalization technique that changes words' forms by removing some parts from start and end of the word with respect to some algorithm. Stemming is much more aggressive and unsupervised normalization technique compared to lemmatization [62]. Since cryptocurrencies are a recent subject and current WordNet dictionary may not include its linguistics, we have also generated a version of dataset that is normalized with stemming. For stemming, we have used Snowball Stemmer [65] from Natural Language Toolkit [63].
- **Lemmatization and Stemming** takes advantage of both normalization technique. We have applied lemmatization followed by stemming and created a third dataset. Since stemming crops the words with respect to some algorithm some words that has same meaning but different form may be cropped to different tokens. In order to prevent that we first applied lemmatization to convert that words to same format and then stemming to map words that may be missed by lemmatization process.

4.1.3 Vectorization

We have used TF-IDF weighting scheme to vectorize document tokens. TF-IDF is a weighting scheme that assigns a weight to a word in document that reflects how many times a word appeared in that document and how rare is that word across whole documents. TF-IDF scheme is useful for assigning higher values to distinctive words that are important for the document [66].

After TF-IDF vectorization, document vectors are normalized to have unit length. For vectorization and normalization, we have used scikit-learn's TF-IDF vectorizer which handles both vectorization and normalization process [67].

4.1.4 Hierarchical Classification

Here we present our supervised model which adapts a divide and conquer approach. We have grouped our classes together into meta-classes with respect to some algorithm to create a class hierarchy. Then we have applied a supervised classification algorithm on each node until we reach to our actual classes at leaf nodes.

In the remaining of this section, we explain our schemes to extract various class hierarchy options. Since *Transaction Anonymity* and *Asset Type* classification tasks has three (3) classes each, we have decided to use all three (3) variations for those tasks. Here we propose five (5) different class hierarchy extraction schemes for *Sector* classification task.

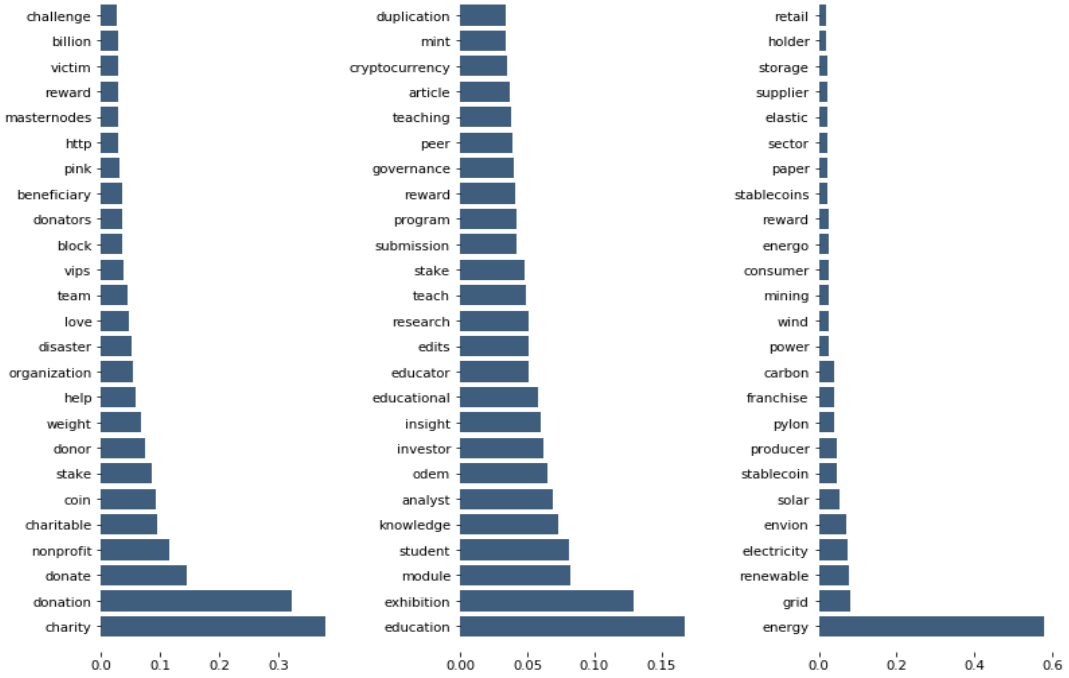
Manually Constructed Hierarchy. As a base class hierarchy option, we generated a class hierarchy manually with intuition.

Class Size Based Hierarchy. This scheme is based on size of classes in our dataset. Extraction of child nodes in class hierarchy works as follows. Firstly, proportion of each class of node is calculated and sorted by their proportion. Then, node classes are divided to two by putting classes that has the larger proportion to *left* until sum

of *left* proportion is less than 50% and the remaining is put to *right*. This process continues until all leaf nodes contains single classes.

Confusion Matrix Based Hierarchy. This scheme is based on confusion matrix generated from best performing supervised learning model's prediction. The aim of this scheme is to group classes under same tree branch that are likely to be mixed by supervised classifiers. In input confusion matrix, rows represent the number of true class instances and columns represent the number of predicted class instances. Firstly, columns in confusion matrix are multiplied by the inverse of the class proportion to prevent class imbalances to dominate to results. With this approach, we want to give higher weights to misclassified instances in classes that has smaller number of instances, the logic applied here is similar to TF-IDF vectorization. For example, assume we have three (3) *Art* instances and ten (10) *Finance* instances in our dataset and if one (1) *Art* and one (1) *Finance* instance is misclassified as *Technology*; since *Art* instances are less than *Finance* instances, *Technology* column in *Art* row will have larger weight then *Finance* thus will indicate more similarity. After vector construction, each parent node in tree is divided to two branches using Agglomerative Clustering with two (2) clusters until all leaf nodes contain single classes.

Class Similarity Based Hierarchy (All Features). This scheme is based on textual similarities between classes. TF-IDF vectors generated for instances are used to generate one vector for each class by taking the mean class vectors. Since distances are not meaningful for high dimensional vectors due to curse of dimensionality [68], before calculating the mean for each class we have applied LSA to reduce dimension space [69]. After vector construction, at each parent node children nodes are selected by using Agglomerative Clustering with two (2) clusters until all leaf nodes contain single classes.



a) Charity and Aid b) Education c) Energy

Figure 4.1. 25 High Weighted Features for *a) Charity and Aid, b) Education* and c) *Energy*

Class Similarity Based Hierarchy (25 High Weighted Features). Procedure applied in this scheme is the same as the class similarity-based hierarchy defined above; only difference is that we have selected twenty-five (25) tokens from each class that has the highest TF-IDF values. Firstly, instance vectors are constructed with these selected tokens and a representative vector for each class is generated by taking mean of the class instance vectors. We have applied LSA also to this scheme and used Agglomerative Clustering with two (2) clusters at each parent node until all leaf nodes contain single classes. An example words for classes can be seen at Figure 4.1.

4.1.5 Cryptocurrency Based Label Detection

In classification phase, models are trained to make predictions in document level. However, since the target problem is cryptocurrency classification, we also required to have a mapping scheme for converting document level predictions to single coin predictions. Here we have proposed three (3) different mapping schemes:

Count Based. This scheme decides coin prediction by counting the number of predictions for each class. The class voted mostly with respect to document label predictions are assigned to coin. In case of equality, label with higher precedence is selected. For example, let us assume our model predicts following labels for each provider respectively *Technology*, *Technology*, *Finance*, *Finance, Finance*, *Finance* and *Technology*. Since there are four (4) *Finance* predictions compared to three (3) *Technology* predictions, *Finance* label is assigned to coin label.

Class Weight Based. This scheme decides coin prediction by weighting each prediction with the inverse label frequency of predicted class as in IDF scheme. A score is assigned to each document prediction with the formula given in Equation 1. Class prediction with maximum score is assigned as coin prediction.

$$PScore_{class} = \#P_{class} * \frac{\#Documents}{\#Class} \quad (1)$$

For example, let's assume we have class frequency distribution with 10 *Art* classes and 90 *Finance* class instances and our model predicts following labels for each provider respectively *Art, Finance* and *Finance.* While there are two (2) *Finance* predictions compared to one (1) *Art* prediction; since the prediction score of *Art* (10) is higher than prediction score of *Finance* (2.22), *Art* is assigned as the coin label.

Confidence Based. The scheme decides coin prediction by using documents' predicted class confidences. A score is assigned to each document prediction with

respect to its predicted class probability. Then for each class, a score is calculated by taking the mean of these document prediction probabilities and the class with the higher score is assigned as coin prediction. For example, let us assume our model predicts following labels and with prediction confidence given in braces respectively *Finance (0.7)*, *Cannabis (0.9)*, *Cannabis (0.6)* and *Cannabis (0.3)*. While there are three (3) *Cannabis* predictions compared to one (1) *Finance* prediction; since the prediction score of *Finance* (0.7) is higher than prediction score of *Cannabis* (0.6), *Finance* is assigned as the coin label.

CHAPTER 5

NEWS SENTIMENT ANALYSIS

5.1.1 Data Collection

The dataset that we used for news sentiment analysis is collected by us from various online resources that publish news based on crypto assets. The collection mechanism periodically checks RSS feeds of the online sources and extracts news title, summary and full content. Some news providers publish full news content in their RSS feeds, but some do not. For sources that do not publish full content we have also implemented scrappers designed per source for scrapping the full content from source itself.

Title, summary and full content of the news are stored separately for each news and subject to sentiment analysis.

5.1.2 Annotation

Dataset that we have collected is manually annotated by experts. Annotation process includes title, summary, content and sentence annotation. For annotation, we have selected one hundred (100) news randomly which were recent at the time of selection. Then we ask three (3) experts to annotate news' title, summary and content separately. In addition, we have also asked them to select positive and negative sentences from the content. The sentences which are not selected by annotators are labeled as neutral.

Title, summary and content annotations are used for validation of sentiment analyzers performance and part of sentence annotations are used for training our model.

5.1.3 Pre-processing

Some summary and full content we have fetched from RSS feeds contains common but irrelevant phrases like "…Continue reading from" or "Liked This Article" so we have implemented filters per source to get rid of such irrelevant texts. In addition, some content we fetched with scrappers were also includes irrelevant phrases like "Rate this news" and some additional links to similar news which includes also the name of the referenced news (which may change the sentiment of analyzed news). To get rid of that, we have also focused on each scrapper separately and filter such irrelevant text from news.

All the news are divided to their sentences before forwarding them to models by using Natural Language Toolkit's sentence tokenizer [63]. We did not apply any additional vectorization process since BERT itself is a language embedding model and designed to learn representations from given words.

5.1.4 Fine-Tuning FinBERT

FinBERT adapts BERT-base architecture to financial domain by fine-tuning the base model on a Financial corpus using Financial PhraseBank [33]. FinBERT achieves state-of-art results on financial sentiment analysis. We have adapted FinBERT's transfer learning approach for cryptocurrency news sentiment analysis by further training the FinBERT on our annotated news dataset [61].

Since cryptocurrency domain can be considered as a subject of Financial domain and we don't have enough data to train a BERT model from scratch, by fine-tuning, we aim to leverage and transfer an already successful model's performance to cryptocurrency sentiment analysis.

For fine-tuning, we have also adapted the procedures applied by FinBERT to prevent catastrophic forgetting. Catastrophic forgetting is a phenomena which indicates losing the already learned language characteristics during fine-tuning [25]. In order

to avoid that, we have adapted the discriminative fine-tuning and gradual unfreezing methods introduced by Howard and Ruder [25] and used by FinBERT model. In discriminative fine-tuning, lower learning rates are used in lower layers of the model to not lose base language features. In gradual unfreezing, training starts with freezing all layers except the classification layer and as training proceeds starting from higher layers, other layers are un-frozen. In this method, the aim is to affect lower layers as little as possible to keep deep-level language features intact [61].

Given a sentence the model outputs four (4) scores which indicates probabilities of given sentence belongs to positive, negative or neutral classes. Model also outputs a compound score which is calculated by extracting negative probability from positive probability. For our evaluations, we have used compound score for analyzing news.

5.1.5 Sentence Score Accumulation

FinBERT model works on sentences and returns a score how much positive or negative some news are. Yet as explained in Section 5.1.1, we need to predict scores for title, summary and content which is not required to be a single sentence. Therefore, we need a mechanism to calculate a compact score for a text includes multiple sentences. Here we present four (4) different methods for accumulating sentence scores to scores of a text which possible include multiple sentences. For future reference, positive sentence indicates sentences having compound score larger than 0.1 and negative sentence indicates sentences having compound score smaller than -0.1.

Mean. In this method full content score is calculated by taking the mean of positive and negative sentences. Neutral sentences are not included in calculation since they may reduce the sentiment intensity score by pulling the score towards zero (0). This method can be used for predicting positive, negative and neutral labels and sentiment intensity score.

Median. Similar to mean approach, but in this method instead of an artificial score a score closest to mean score of the full content selected as the sentiment score. As same as mean method, sentences that fell into neutral zone are excluded. This method also can be used for predicting positive, negative and neutral labels and sentiment intensity score.

Count. This method cannot be used to predict a sentiment intensity score since it depends on number of negative, positive and neutral sentences for predicting given text. This method decides on final prediction by counting number of sentences labeled as positive, negative, neutral and selects the label with higher number of occurrences. In case of equality different approaches are used; if a label indicates polarity (positive or negative) has equal number of sentences with neutral; polarity label is selected, if positive and negative labels are equal, neutral label is selected.

Min-Max. As mean and median methods this method can also be used for predicting label and sentiment intensity. In this method, sentence's score with the highest absolute value is assigned as the content's sentiment.

CHAPTER 6

EXPERIMENTS

6.1 Classification Towards Taxonomy

In this section, we have explained the steps that we followed for evaluating proposed model on our dataset for sector, transaction anonymity and asset type labels. Dataset used for experiments for each prediction task is explained below.

Table 6.1 Transaction Anonymity Label Distribution

Transaction Anonymity	***Document***	***Coin***
Anonymous	223	41
Pseudonymous	7518	1557
Selective Transparency	38	7
Total	**7779**	**1605**

Sector. Dataset that we have used for sector experiments has 5575 annotated documents which are in total mapped to 1162 unique coins. Three major classes in dataset are Finance, Technology and Entertainment classes with 1826 (388 Coins), 1663 (326 Coins) and 951 (199 Coins) instances respectively which make up approximately %80 of the dataset. In addition, three minor classes in dataset are Art, Retail and Food Production classes with 13 (4 Coins), 20 (4 Coins) and 28 (8 Coins) respectively (Table 6.2).

Asset Type. Dataset we have used for asset type experiments has 4577 annotated documents which are in total mapped to 842 unique coins. %93 of this dataset is made up by Utility Token instances (Table 6.3).

Table 6.2 Sector Label Distribution

Sector	***Document***	***Coin***
Art	13	4
Cannabis	43	10
Charity and Aid	42	12
Content & Advertisement	318	68
Education	42	10
Energy	123	28
Entertainment	951	199
Finance	1826	388
Food Production	28	8
Healthcare and Pharma	144	28
Insurance	64	13
Real Estate	71	16
Retail	20	4
Technology	1663	326
Telecommunication	61	13
Transportation	84	18
Travel	82	17
Total	**5575**	**1162**

Transaction Anonymity. Dataset we have used for transaction anonymity experiments has 3956 instances which are mapped to 993 unique coins. %97 of this dataset is made up by Pseudonymous instances (Table 6.1)

Table 6.3 Asset Type Label Distribution

Asset Type	***Document***	***Coin***
Security Token	71	32
Payment Token	240	35
Utility Token	4266	775
Total	**4577**	**842**

In the following subsections, firstly we have explained our experiments setting, then we have presented our experiment results and lastly, we have discussed the experiment results.

6.1.1 Experiment Setting

Before starting experiments, we have divided our dataset into train and test sets with 9:1 ratio. To not introduce any bias, we have divided our dataset with respect to coin instances, so train and test sets do not have document instances that belongs to same coin.

For experiments we consider following supervised models, from Python's Sci-kit learn library; LinearSVC, SVC, NaiveBayes, DecisionTree, RandomForest models [67], a gradient boosting tree model implementation from LightGBM's BoostingTree [70], as neural network models we have implemented fastText [55], TextCNN [21], RCNN [56], TextRNN [71], RNN-Attention [58], Transformer [60] models using PyTorch and lastly our hierarchical classification model. In addition, for transaction anonymity and asset type prediction tasks we have presented six (6) hierarchical model variations each and for sector prediction task we have presented ten (10) different variations of hierarchical classification model. Below we have discussed training and optimization phases.

Class Imbalance. All three prediction tasks have class imbalance problem, in order to solve this, in training phase, we have used all our models with weighted mode which increases the impact of class instances inversely by class proportion. In addition, we have also presented experiments by selectively under sampling some major class instances from dataset. Under sampling is done randomly with respect to coins in order not to introduce a bias due to intersecting coin documents between test and train datasets. For sector prediction task, we have under sampled three (3) major sector classes (Finance, Technology, Entertainment) to eighty (80) instances, for transaction anonymity prediction task, we have under sampled Pseudonymous and Anonymous classes to twenty (20) instances and for asset type prediction task, we have under sampled Utility Token class to fifty (50) instances. The results for these variations are presented in Evaluation section.

Bayesian Optimization. We have used Bayesian optimization for hyperparameter tuning on each model. Given hyperparameter spaces and number of iterations, algorithm tries to find best possible hyperparameter setting by searching through given space that minimizes some objective function [72]. Since we have class imbalance problem, we have decided to use Cohen-Kappa metric subtracted from one (1).

Coin Based 10-Fold Cross Validation. Since textual content from different providers that belongs to same coin may introduce bias in our validation. Therefore, while dividing our data to 10-Folds for validation we have used coin IDs.

Model Selection. To select the best performing model, we have generated one thousand (1000) different hyperparameter configurations using Bayesian Optimization and we have used coin based 10-Fold cross validation. The best performing model is selected upon variations that gives the best Cohen-Kappa score. The special case for hierarchical models is given below. For LinearSVC, SVC, NaiveBayes, DecisionTree, RandomForest, Boosting Tree models hyperparameter optimization also includes min_df and max_df parameters of TF-IDF vectorizer [67].

Hierarchical Classification. We have proposed two (2) different variations for training hierarchical classification models. On our first case, on each branch we have performed experiments LinearSVC, SVC, NaiveBayes, DecisionTree, RandomForest, Boosting Tree models by optimizing each for two hundred (200) steps and we have selected the best performing model for the branch with respect to Cohen Kappa score. On our second case, we have optimized only LinearSVC model on each branch for one thousand (1000) steps and selected best performing model with respect to Cohen Kappa score. In each branch, for hyperparameter space generation of TF-IDF vectorizer's min_df parameter, we have also considered the class distributions of selected branch to prevent losing important features and including redundant features. To exemplify, while optimizing for Art and Finance classes having min_df more than eight (8) may exclude selective features for Art instances from our feature set. We have used one fifth of the minor class size in each branch as lower bound and one fifth of major class size in each branch as upper bound for hyperparameter space.

6.1.2 Experiments

In this section, we have presented our experiment results. This section is divided to subsections with respect to label prediction tasks. There are six (6) experiments in total for each prediction task. In each experiment, we have presented validation, document-based test results and cryptocurrency-based test results on selected best three (3) models for that experiment.

6.1.2.1 Sector

In this section, we have presented our experiment results of Sector prediction task. There is an experiment for each dataset generated by different text normalization techniques (Experiment 1, 3, 5) and their semi-balanced variations generated by under sampling three major sector (Finance, Technology and Entertainment) class instances randomly to have eighty (80) unique cryptocurrency instances (Experiment 2, 4, 6).

6.1.2.1.1 Experiment 1: Experiments on Dataset Generated by Stemming

For this experiment, we have used the dataset version which we have generated by using stemming as text normalization.

Table 6.4 Experiment 1 Sector Document Based Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.74	**0.78**	0.74	**0.76**	**0.6**
SVC	**0.75**	0.77	**0.75**	0.75	0.59
BoostingTree	0.7	0.75	0.7	0.72	0.54
DecisionTree	0.51	0.63	0.51	0.53	0.28
NaiveBayes	0.71	0.75	0.71	0.72	0.54
RandomForest	0.67	0.7	0.67	0.68	0.48
RCNN	0.02	0.4	0.02	0.01	0.01
RNN-Attention	0.03	0.46	0.03	0.05	0
TextCNN	0.02	0	0.02	0	0.01
TextRNN	0.02	0.44	0.02	0.03	0
Transformer	0.1	0.47	0.1	0.14	0.03
fastText	0	0	0	0	0
Hierarchical	0.72	0.76	0.72	0.73	0.56
Hierarchical-CM	0.71	0.77	0.71	0.73	0.55
Hierarchical-SIM	0.71	0.75	0.71	0.72	0.54
Hierarchical-SIM25	0.7	0.75	0.7	0.72	0.53
Hierarchical-Size	0.7	0.77	0.7	0.73	0.55
HierarchicalHard	0.72	0.76	0.72	0.74	0.57
HierarchicalHard-CM	0.71	0.76	0.71	0.73	0.55
HierarchicalHard-SIM	0.7	0.75	0.7	0.72	0.54
HierarchicalHard-SIM25	0.71	0.76	0.71	0.73	0.55
HierarchicalHard-Size	0.69	0.76	0.69	0.72	0.53

As seen from Figure A.1, LinearSVC model is the best model in terms of validation scores for each metric. Among Hierarchical model variations, although it is a close competition, HierarchicalHard-CM model performed better than others by a small margin. Neural network models have failed to compete with the other models for this experiment.

As seen from Table 6.4, LinearSVC and SVC models have best document-based test scores for this experiment. All hierarchical models have close scores, but they are not able to compete with LinearSVC and SVC models.

Table 6.5 Experiment 1 Sector Cryptocurrency Based Label Prediction Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.82	**0.84**	0.82	0.82	0.71
	Count Based	0.82	**0.84**	0.82	0.82	0.71
	Class Weight Based	0.8	0.83	0.8	0.81	0.68
SVC	Confidence Based	**0.83**	0.82	**0.83**	**0.83**	0.72
	Count Based	0.8	0.79	0.8	0.79	0.67
	Class Weight Based	0.81	0.81	0.81	0.8	0.68
HierarchicalHard	Confidence Based	0.79	0.82	0.79	0.8	0.67
	Count Based	0.77	**0.84**	0.77	0.8	**0.74**
	Class Weight Based	0.79	0.82	0.79	0.8	0.67

As seen from Table 6.5, in terms of cryptocurrency-based prediction; HierarchicalHard and SVC model have better scores than LinearSVC model. Confidence Based scheme of SVC model is more preferable since the tradeoff between scoring metrics is smaller than the HierarchicalHard model.

6.1.2.1.2 Experiment 2: Experiments on Under Sampled Dataset Generated by Stemming

For this experiment, we have used the semi-balanced dataset version which we have generated by using stemming as text normalization.

Table 6.6 Experiment 2 Sector Document Based Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.68	**0.78**	0.68	0.72	0.55
SVC	0.71	0.76	0.71	0.73	0.58
BoostingTree	0.65	0.74	0.65	0.69	0.51
DecisionTree	0.47	0.59	0.47	0.49	0.26
NaiveBayes	**0.72**	0.75	**0.72**	**0.73**	**0.59**
RandomForest	0.65	0.72	0.65	0.67	0.49
RCNN	0.05	0.34	0.05	0.06	0
RNN-Attention	0.03	0.33	0.03	0.06	0
TextCNN	0.04	0.05	0.04	0.04	0.02
TextRNN	0.11	0.38	0.11	0.16	0.01
Transformer	0.09	0.38	0.09	0.13	0.01
fastText	0	0	0	0	0
Hierarchical	0.7	0.73	0.7	0.71	0.55
Hierarchical-CM	0.58	0.74	0.58	0.62	0.43
Hierarchical-SIM	0.65	0.72	0.65	0.68	0.5
Hierarchical-SIM25	0.62	0.71	0.62	0.65	0.46
Hierarchical-Size	0.67	0.75	0.67	0.7	0.53
HierarchicalHard	0.66	0.73	0.66	0.69	0.52
HierarchicalHard-CM	0.64	0.74	0.64	0.68	0.5
HierarchicalHard-SIM	0.64	0.72	0.64	0.67	0.49
HierarchicalHard-SIM25	0.64	0.72	0.64	0.67	0.49
HierarchicalHard-Size	0.67	0.74	0.67	0.7	0.52

As seen from Figure A.2, LinearSVC model is the best model in terms of validation scores for each metric. Among Hierarchical model variations, although it is a close competition, HierarchicalHard-CM and HierarchicalHard-SIM25 models performed better than others by a small margin. Neural network models have failed to compete with the other models for this experiment.

As seen from Table 6.6, NaiveBayes model have best document-based test scores for this experiment. SVC model is also a close competitor for this experiment in terms of document-based scores.

Table 6.7 Experiment 2 Sector Cryptocurrency Based Label Prediction Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
SVC	Confidence Based	0.81	0.83	0.81	0.82	0.72
	Count Based	0.78	0.8	0.78	0.79	0.68
	Class Weight Based	0.77	0.81	0.77	0.78	0.66
NaiveBayes	Confidence Based	**0.82**	**0.84**	**0.82**	**0.83**	**0.73**
	Count Based	0.8	0.83	0.8	0.81	0.7
	Class Weight Based	0.8	0.83	0.8	0.81	0.7
Hierarchical	Confidence Based	0.78	0.79	0.78	0.78	0.66
	Count Based	0.74	0.83	0.74	0.78	0.71
	Class Weight Based	0.77	0.8	0.77	0.78	0.66

As seen from Table 6.7, in terms of cryptocurrency-based prediction; NaiveBayes model have better scores than all other models. Confidence Based scheme of NaiveBayes clearly the best performing model in this experiment.

6.1.2.1.3 Experiment 3: Experiments on Dataset Generated by Lemmatization

For this experiment, we have used the dataset version which we have generated by using lemmatization as text normalization.

Table 6.8 Experiment 3 Sector Document Based Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.75	0.78	0.75	0.76	0.61
SVC	0.73	**0.79**	0.73	0.75	0.58
BoostingTree	0.72	0.76	0.72	0.73	0.57
DecisionTree	0.52	0.62	0.52	0.52	0.27
NaiveBayes	0.74	0.76	0.74	0.74	0.58
RandomForest	0.69	0.72	0.69	0.7	0.51
RCNN	0.03	0.44	0.03	0.05	0.01
RNN-Attention	0.07	0.38	0.07	0.1	0
TextCNN	0.02	0.15	0.02	0.01	0.02
TextRNN	0.04	0.42	0.04	0.07	0
Transformer	0.06	0.42	0.06	0.09	0.01
fastText	0.01	0	0.01	0	0
Hierarchical	**0.77**	**0.78**	**0.77**	**0.77**	**0.62**
Hierarchical-CM	0.7	0.78	0.7	0.73	0.55
Hierarchical-SIM	0.7	0.74	0.7	0.71	0.53
Hierarchical-SIM25	0.73	0.77	0.73	0.74	0.57
Hierarchical-Size	0.72	0.75	0.72	0.73	0.56
HierarchicalHard	0.72	0.74	0.72	0.73	0.55
HierarchicalHard-CM	0.73	0.77	0.73	0.74	0.57
HierarchicalHard-SIM	0.69	0.74	0.69	0.7	0.52
HierarchicalHard-SIM25	0.73	0.76	0.73	0.74	0.57
HierarchicalHard-Size	0.72	0.75	0.72	0.74	0.56

As seen from Figure A.3, LinearSVC model is the best model in terms of validation scores for each metric. Among Hierarchical model variations, although it is a close competition, HierarchicalHard-CM model performed better than others by a small margin. Neural network models have failed to compete with the other models for sector prediction task.

As seen from Table 6.8, LinearSVC, SVC and Naïve Bayes model have promising test scores even though they were not able to compete with Hierarchical model in testing phase.

Table 6.9 Experiment 3 Sector Cryptocurrency Based Label Prediction Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.81	0.83	0.81	0.82	0.7
	Count Based	0.81	0.83	0.81	0.82	0.7
	Class Weight Based	0.8	0.84	0.8	0.81	0.69
NaiveBayes	Confidence Based	0.82	0.82	0.82	0.82	0.71
	Count Based	0.8	0.81	0.8	0.8	0.68
	Class Weight Based	0.81	0.82	0.81	0.81	0.69
Hierarchical	Confidence Based	**0.84**	0.85	**0.84**	**0.84**	0.74
	Count Based	0.8	**0.86**	0.8	0.83	**0.76**
	Class Weight Based	0.82	0.84	0.82	0.83	0.71

As seen from Table 6.9, in terms of cryptocurrency-based prediction; Hierarchical model have better scores than LinearSVC and NaiveBayes models. Confidence Based scheme of Hierarchical model is more preferable since the tradeoff between scoring metrics is smaller than the Count Based scheme.

6.1.2.1.4 Experiment 4: Experiments on Under Sampled Dataset Generated by Lemmatization

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization as text normalization.

Table 6.10 Experiment 4 Document Based Sector Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.7	**0.79**	0.7	0.74	0.57
SVC	0.72	0.77	0.72	0.74	0.59
BoostingTree	0.66	0.74	0.66	0.69	0.51
DecisionTree	0.48	0.61	0.48	0.49	0.26
NaiveBayes	**0.74**	0.77	**0.74**	**0.75**	**0.62**
RandomForest	0.65	0.69	0.65	0.67	0.49
RCNN	0.1	0.37	0.1	0.15	0.01
RNN-Attention	0.09	0.36	0.09	0.13	0.01
TextCNN	0.04	0.3	0.04	0.06	0
TextRNN	0.03	0.36	0.03	0.05	0
Transformer	0.05	0.35	0.05	0.08	0
fastText	0	0.32	0	0	0
Hierarchical	0.71	0.75	0.71	0.73	0.57
Hierarchical-CM	0.65	0.75	0.65	0.69	0.52
Hierarchical-SIM	0.61	0.72	0.61	0.66	0.46
Hierarchical-SIM25	0.62	0.73	0.62	0.67	0.48
Hierarchical-Size	0.69	0.77	0.69	0.72	0.56
HierarchicalHard	0.62	0.67	0.62	0.64	0.46
HierarchicalHard-CM	0.66	0.75	0.66	0.7	0.53
HierarchicalHard-SIM	0.61	0.7	0.61	0.64	0.45
HierarchicalHard-SIM25	0.62	0.71	0.62	0.66	0.47
HierarchicalHard-Size	0.63	0.72	0.63	0.66	0.47

As seen from Figure A.4, for each of the evaluation metrics LinearSVC is the best among all models by having approximately %9 better performance on each metric. Although our Hierarchical models are not able to compete with LinearSVC, they can still compete with other traditional supervised models. In addition, even though it is small, we have achieved some performance improvement by switching to HierarchicalHard models. In this experiment, neural network models are the worst models by large margin and are not able to compete with other models.

Table 6.11 Experiment 4 Sector Cryptocurrency Based Label Prediction Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.78	**0.85**	0.78	0.8	0.68
	Count Based	0.78	**0.85**	0.78	0.8	0.68
	Class Weight Based	0.75	**0.85**	0.75	0.79	0.64
SVC	Confidence Based	0.8	0.84	0.8	**0.82**	0.71
	Count Based	0.8	0.83	0.8	0.81	0.7
	Class Weight Based	0.79	0.83	0.79	0.8	0.69
NaiveBayes	Confidence Based	**0.82**	0.84	**0.82**	**0.82**	**0.72**
	Count Based	0.8	0.83	0.8	0.81	0.7
	Class Weight Based	0.8	0.83	0.8	0.81	0.7

As seen from Table 6.10, in document prediction testing phase NaiveBayes is performed best by having highest scores on four (4) metrics. LinearSVC is the second interesting model to further investigate since it has the highest score in precision. Lastly, in terms of Cohen-Kappa, SVC model is the third best model selected for cryptocurrency-based result investigation.

In Table 6.11, we have presented cryptocurrency-based prediction scores for three (3) models selected from document prediction phase. As seen, even though LinearSVC model has the highest precision it fails to compete in other metrics.

Confidence Based scheme of SVC and NaiveBayes in close competition for to be the best in this experiment. Yet, NaiveBayes has higher scores on three (3) metrics than SVC model therefore it is the best performing model for this experiment.

6.1.2.1.5 Experiment 5: Experiments on Dataset Generated by Lemmatization Followed by Stemming

For this experiment, we have used the dataset version which we have generated by using lemmatization followed by stemming as text normalization.

Table 6.12 Experiment 5 Sector Document Based Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.72	**0.79**	0.72	0.75	0.58
SVC	**0.77**	**0.79**	**0.77**	**0.77**	**0.63**
BoostingTree	0.72	0.76	0.72	0.73	0.56
DecisionTree	0.55	0.6	0.55	0.55	0.28
NaiveBayes	0.73	0.76	0.73	0.74	0.57
RandomForest	0.68	0.71	0.68	0.69	0.49
RCNN	0.01	0.65	0.01	0.01	0
RNN-Attention	0.02	0.49	0.02	0.02	0
TextCNN	0.17	0.22	0.17	0.18	0.07
TextRNN	0.03	0.33	0.03	0.05	0
Transformer	0.07	0.4	0.07	0.11	0
fastText	0.24	0.13	0.24	0.17	-0.04
Hierarchical	0.73	0.76	0.73	0.74	0.58
Hierarchical-CM	0.74	0.76	0.74	0.75	0.58
Hierarchical-SIM	0.7	0.74	0.7	0.71	0.53
Hierarchical-SIM25	0.7	0.74	0.7	0.71	0.53
Hierarchical-Size	0.72	0.75	0.72	0.73	0.56
HierarchicalHard	0.71	0.74	0.71	0.72	0.55
HierarchicalHard-CM	0.75	0.77	0.75	0.76	0.59
HierarchicalHard-SIM	0.7	0.74	0.7	0.71	0.53
HierarchicalHard-SIM25	0.7	0.75	0.7	0.71	0.54
HierarchicalHard-Size	0.72	0.75	0.72	0.73	0.55

As seen from Figure A.5, LinearSVC model is the best model in terms of validation scores for each metric. Among Hierarchical model variations, although it is a close competition, HierarchicalHard-CM model performed better than others by a small margin. Neural network models have failed to compete with the other models for this experiment.

As seen from Table 6.12, LinearSVC has the best document-based test scores for this experiment. Even though SVC model has the highest precision score, Hierarchical-CM and HierarchicalHard-CM models higher scores on multiple metrics than SVC model.

Table 6.13 Experiment 5 Sector Cryptocurrency Based Label Prediction Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
SVC	Confidence Based	**0.85**	**0.85**	**0.85**	**0.85**	**0.76**
	Count Based	0.84	0.84	0.84	0.84	0.74
	Class Weight Based	0.82	0.84	0.82	0.82	0.7
Hierarchical-CM	Confidence Based	0.79	0.8	0.79	0.8	0.65
	Count Based	0.79	0.82	0.79	0.8	0.65
	Class Weight Based	0.77	0.82	0.77	0.79	0.63
HierarchicalHard-CM	Confidence Based	0.8	0.83	0.8	0.81	0.68
	Count Based	0.78	0.84	0.78	0.81	0.73
	Class Weight Based	0.78	0.82	0.78	0.79	0.65

As seen from Table 6.13, in terms of cryptocurrency-based prediction; SVC model have better scores than Hierarchical-CM and HierarchicalHard-CM models. Confidence Based scheme of SVC model is clearly the best model for this experiment.

6.1.2.1.6 Experiment 6: Experiments on Under Sampled Dataset Generated by Lemmatization Followed by Stemming

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization followed by stemming as text normalization.

Table 6.14 Experiment 6 Sector Document Based Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.67	**0.8**	0.67	0.72	0.54
SVC	0.71	0.76	0.71	0.73	0.58
BoostingTree	0.65	0.75	0.65	0.69	0.51
DecisionTree	0.45	0.62	0.45	0.47	0.25
NaiveBayes	**0.73**	0.76	**0.73**	**0.74**	**0.6**
RandomForest	0.65	0.71	0.65	0.68	0.5
RCNN	0.16	0.3	0.16	0.18	0
RNN-Attention	0.1	0.34	0.1	0.14	0
TextCNN	0.01	0.35	0.01	0.02	0
TextRNN	0.03	0.32	0.03	0.04	0
Transformer	0.05	0.34	0.05	0.09	0
fastText	0.16	0.03	0.16	0.05	-0.01
Hierarchical	0.71	0.75	0.71	0.73	0.58
Hierarchical-CM	0.62	0.73	0.62	0.66	0.47
Hierarchical-SIM	0.62	0.73	0.62	0.67	0.48
Hierarchical-SIM25	0.62	0.73	0.62	0.67	0.48
Hierarchical-Size	0.69	0.77	0.69	0.72	0.56
HierarchicalHard	0.61	0.67	0.61	0.63	0.44
HierarchicalHard-CM	0.64	0.75	0.64	0.68	0.5
HierarchicalHard-SIM	0.65	0.73	0.65	0.68	0.51
HierarchicalHard-SIM25	0.63	0.71	0.63	0.66	0.48
HierarchicalHard-Size	0.66	0.73	0.66	0.69	0.51

As seen from Figure A.6, LinearSVC model is the best model in terms of validation scores for each metric. Among Hierarchical model variations, although it is a close competition, HierarchicalHard-SIM25 model performed better than others by a small margin. Neural network models have failed to compete with the other models for this experiment.

As seen from Table 6.14, NaiveBayes model have best document-based test scores for this experiment. Even though LinearSVC has the highest precision, SVC and Hierarchical models have better scores than LinearSVC on multiple metrics.

Table 6.15 Experiment 6 Sector Cryptocurrency Based Label Prediction Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
SVC	Confidence Based	0.8	**0.83**	0.8	0.81	0.7
	Count Based	0.78	0.81	0.78	0.79	0.67
	Class Weight Based	0.77	0.82	0.77	0.79	0.66
NaiveBayes	Confidence Based	**0.81**	**0.83**	**0.81**	**0.82**	**0.72**
	Count Based	0.79	0.82	0.79	0.8	0.69
	Class Weight Based	0.8	**0.83**	0.8	0.81	0.7
Hierarchical	Confidence Based	0.79	0.82	0.79	0.8	0.69
	Count Based	0.75	0.82	0.75	0.78	**0.72**
	Class Weight Based	0.78	0.81	0.78	0.79	0.67

As seen from Table 6.15, in terms of cryptocurrency-based prediction; NaiveBayes model have better scores than all other models. Confidence Based scheme of NaiveBayes clearly the best performing model for this experiment.

6.1.2.2 Asset Type

In this section, we have presented our experiment results of Asset Type prediction task. There is an experiment for each dataset generated by different text normalization techniques (Experiment 1, 3, 5) and their semi-balanced variations. For balancing the data, we have under sampled Utility class instances randomly to have fifty (50) unique cryptocurrency instances (Experiment 2, 4, 6).

6.1.2.2.1 Experiment 1: Experiments on Dataset Generated by Stemming

For this experiment, we have used the dataset version that we have generated by using stemming as text normalization.

Table 6.16 Experiment 1 Balanced Document Based Asset Type Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.56	0.74	0.56	0.53	0.34
SVC	0.55	**0.76**	0.55	0.53	0.33
BoostingTree	0.64	0.71	0.64	0.63	0.46
DecisionTree	0.55	0.62	0.55	0.54	0.33
NaiveBayes	0.38	0.42	0.38	0.26	0.07
RandomForest	0.65	0.71	0.65	0.64	0.47
RCNN	0.34	0.42	0.34	0.34	0.01
RNN-Attention	0.35	0.36	0.35	0.35	0.03
TextCNN	0.33	0.26	0.33	0.2	-0.01
TextRNN	0.33	0.11	0.33	0.17	-0.01
Transformer	0.29	0.28	0.29	0.28	-0.07
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.5	0.36	0.5	0.41	0.25
Hierarchical-1	**0.67**	0.72	**0.67**	**0.66**	**0.5**
Hierarchical-2	0.58	0.73	0.58	0.57	0.36
HierarchicalHard	0.59	0.59	0.59	0.55	0.38
HierarchicalHard-1	0.59	0.71	0.59	0.56	0.39
HierarchicalHard-2	0.57	0.72	0.57	0.54	0.36

As seen from Figure B.1, Hierarchical-1, Hierarchical-2 and their variations are close winners of the Experiment 1's training and validation phase. SVC and LinearSVC models also show competitive scores.

As expected, Hierarchical-1 model is the close winner of the testing phase as seen from Table 6.16. BoostingTree and RandomForest models also have competitive scores for document-based classification task.

Table 6.17 Experiment 1 Asset Type Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
BoostingTree	Confidence Based	0.63	0.74	0.63	0.6	0.45
	Count Based	0.71	0.77	0.71	0.7	0.56
	Class Weight Based	0.68	0.73	0.68	0.68	0.52
RandomForest	Confidence Based	**0.77**	**0.82**	**0.77**	**0.75**	**0.65**
	Count Based	0.75	0.79	0.75	0.73	0.62
	Class Weight Based	0.71	0.74	0.71	0.69	0.56
Hierarchical-1	Confidence Based	0.75	0.79	0.75	0.74	0.63
	Count Based	0.64	0.81	0.64	0.64	0.6
	Class Weight Based	0.7	0.72	0.7	0.69	0.55

As seen from Table 6.17, even though Hierarchical-1 has performed better on document classification, RandomForest have performed better in terms of cryptocurrency-based prediction. Confidence Based scheme of RandomForest is the clear winner of this experiment.

6.1.2.2.2 Experiment 2: Experiments on Under Sampled Dataset Generated by Stemming

For this experiment, we have used the semi-balanced dataset version which we have generated by using stemming as text normalization.

Table 6.18 Experiment 2 Balanced Document Based Asset Type Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.65	0.69	0.65	0.63	0.48
SVC	0.67	0.71	0.67	0.66	0.51
BoostingTree	0.68	0.7	0.68	0.68	0.52
DecisionTree	0.47	0.47	0.47	0.47	0.2
NaiveBayes	0.68	0.75	0.68	0.67	0.52
RandomForest	0.65	0.66	0.65	0.63	0.47
RCNN	0.36	0.4	0.36	0.32	0.03
RNN-Attention	0.24	0.23	0.24	0.23	-0.13
TextCNN	0.5	0.55	0.5	0.48	0.24
TextRNN	0.29	0.25	0.29	0.22	-0.07
Transformer	0.48	0.49	0.48	0.48	0.22
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.6	0.64	0.6	0.55	0.39
Hierarchical-1	0.72	0.76	0.72	0.71	0.58
Hierarchical-2	**0.79**	**0.78**	**0.79**	**0.78**	**0.68**
HierarchicalHard	0.56	0.55	0.56	0.52	0.33
HierarchicalHard-1	0.62	0.67	0.62	0.61	0.43
HierarchicalHard-2	0.66	0.7	0.66	0.64	0.48

As seen from Figure B.2, SVC, Hierarchical-1 and Hierarchical-2 models are the best performing models of the Experiment 2's training and validation phase. RandomForest model also is a close competitor of this phase.

As seen from Table 6.18, Hierarchical-2 model is the best performing model of the testing phase by having better scores on all metrics. Hierarchical-1 and NaiveBayes models are the second and third best performing models.

Table 6.19 Experiment 2 Asset Type Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
NaiveBayes	Confidence Based	0.67	0.76	0.67	0.63	0.5
	Count Based	0.66	0.75	0.66	0.62	0.48
	Class Weight Based	0.65	0.74	0.65	0.62	0.48
Hierarchical-1	Confidence Based	0.7	0.75	0.7	0.69	0.55
	Count Based	0.69	0.77	0.69	0.69	0.58
	Class Weight Based	0.68	0.71	0.68	0.66	0.52
Hierarchical-2	Confidence Based	0.8	0.81	0.8	0.8	0.71
	Count Based	**0.81**	**0.83**	**0.81**	**0.82**	**0.75**
	Class Weight Based	0.8	0.8	0.8	0.8	0.71

As seen from Table 6.19, as expected Hierarchical-2 is clearly the best performing model in cryptocurrency-based prediction task. Upon prediction mapping schemes Count Based scheme has the best scores.

6.1.2.2.3 Experiment 3: Experiments on Dataset Generated by Lemmatization

For this experiment, we have used the dataset version which we have generated by using lemmatization as text normalization.

Table 6.20 Experiment 3 Balanced Document Based Asset Type Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.55	0.75	0.55	0.53	0.32
SVC	0.57	**0.76**	0.57	0.55	0.36
BoostingTree	0.52	0.65	0.52	0.5	0.28
DecisionTree	0.55	0.61	0.55	0.56	0.33
NaiveBayes	0.45	0.43	0.45	0.37	0.18
RandomForest	0.58	0.69	0.58	**0.58**	0.37
RCNN	0.44	0.62	0.44	0.37	0.16
RNN-Attention	0.41	0.38	0.41	0.37	0.12
TextCNN	0.29	0.38	0.29	0.26	-0.06
TextRNN	0.37	0.43	0.37	0.35	0.06
Transformer	0.4	0.45	0.4	0.4	0.1
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.55	0.6	0.55	0.52	0.32
Hierarchical-1	0.59	0.72	0.59	0.56	0.39
Hierarchical-2	0.58	0.72	0.58	0.56	0.37
HierarchicalHard	0.5	0.39	0.5	0.42	0.24
HierarchicalHard-1	**0.61**	0.72	**0.61**	**0.58**	**0.41**
HierarchicalHard-2	0.59	0.69	0.59	0.56	0.39

As seen from Figure B.3, Hierarchical-1, Hierarchical-2 and their variations are close winners of the Experiment 3's training and validation phase. SVC and Random Forest also have competitive but lower scores.

As seen from Table 6.20, HierarchicalHard-1 model has the highest scores in four (4) metrics among other models. Next two models are SVC model which has highest precision and Hierarchical-1 having the second highest Cohen-Kappa score.

Table 6.21 Experiment 3 Asset Type Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Hierarchical-1	Confidence Based	**0.72**	0.81	**0.72**	0.69	0.58
	Count Based	**0.72**	0.82	**0.72**	**0.7**	**0.6**
	Class Weight Based	0.69	0.75	0.69	0.66	0.53
HierarchicalHard-1	Confidence Based	0.7	0.78	0.7	0.67	0.55
	Count Based	0.6	0.46	0.6	0.52	0.48
	Class Weight Based	0.67	0.72	0.67	0.64	0.5
SVC	Confidence Based	0.66	0.82	0.66	0.63	0.49
	Count Based	0.66	0.81	0.66	0.63	0.49
	Class Weight Based	**0.72**	**0.83**	**0.72**	0.68	0.58

As seen from Table 6.21, Hierarchial-1's Count Based scheme and SVC's Class Weight Based scheme are the best performing models in cryptocurrency-based classification. Yet, Hierarchical-1 is more preferable since model has highest scores in four (4) metrics.

6.1.2.2.4 Experiment 4: Experiments on Under Sampled Dataset Generated by Lemmatization

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization as text normalization. As seen from Figure B.4, Hierarchical-1, Hierarchical-2 and SVC models have the best scores among other models in training and validation phase of Experiment 4.

Table 6.22 Experiment 4 Balanced Document Based Asset Type Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.65	0.69	0.65	0.64	0.48
SVC	0.6	0.7	0.6	0.57	0.39
BoostingTree	0.59	0.6	0.59	0.58	0.38
DecisionTree	0.41	0.39	0.41	0.38	0.11
NaiveBayes	**0.69**	**0.77**	**0.69**	**0.68**	**0.53**
RandomForest	0.64	0.66	0.64	0.64	0.46
RCNN	0.49	0.56	0.49	0.43	0.23
RNN-Attention	0.41	0.42	0.41	0.38	0.12
TextCNN	0.37	0.35	0.37	0.35	0.06
TextRNN	0.31	0.29	0.31	0.28	-0.03
Transformer	0.41	0.41	0.41	0.41	0.11
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.49	0.34	0.49	0.4	0.23
Hierarchical-1	0.57	0.61	0.57	0.57	0.36
Hierarchical-2	0.63	0.66	0.63	0.62	0.45
HierarchicalHard	0.56	0.57	0.56	0.55	0.34
HierarchicalHard-1	0.65	0.69	0.65	0.63	0.47
HierarchicalHard-2	0.61	0.64	0.61	0.58	0.41

In Table 6.22, we have provided document-based test results on balanced test set. As seen, NaiveBayes is the best performing model by having highest scores for each metric in document classification task of this experiment. Next two best models in terms of Cohen-Kappa are LinearSVC and HierarchicalHard-1 model.

Table 6.23 Experiment 4 Asset Type Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.64	0.7	0.64	0.61	0.47
	Count Based	0.64	0.7	0.64	0.61	0.47
	Class Weight Based	0.64	0.68	0.64	0.6	0.46
NaiveBayes	Confidence Based	0.67	0.76	0.67	**0.64**	0.51
	Count Based	**0.68**	**0.77**	**0.68**	**0.64**	**0.52**
	Class Weight Based	0.67	0.75	0.67	**0.64**	0.51
HierarchicalHard-1	Confidence Based	0.58	0.63	0.58	0.54	0.37
	Count Based	0.58	0.68	0.58	0.55	0.41
	Class Weight Based	0.58	0.62	0.58	0.54	0.37

As seen from Table 6.23, Count Based scheme of NaiveBayes model is the best performing model in cryptocurrency-based prediction task of this experiment by having highest scores on all metrics.

6.1.2.2.5 Experiment 5: Experiments on Dataset Generated by Lemmatization Followed by Stemming

For this experiment, we have used the dataset version which we have generated by using lemmatization followed by stemming as text normalization.

As seen from Figure B.5, Hierarchical-1 and Hierarchical-2 models are close winners of the Experiment 5's training and validation phase. SVC, LinearSVC and RandomForest models also have competitive but lower scores.

Table 6.24 Experiment 5 Balanced Document Based Asset Type Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.55	0.75	0.55	0.53	0.33
SVC	0.56	**0.76**	0.56	0.54	0.35
BoostingTree	**0.61**	0.68	**0.61**	**0.6**	**0.41**
DecisionTree	0.53	0.58	0.53	0.53	0.29
NaiveBayes	0.4	0.41	0.4	0.29	0.1
RandomForest	0.53	0.67	0.53	0.51	0.3
RCNN	0.32	0.24	0.32	0.26	-0.02
RNN-Attention	0.34	0.31	0.34	0.29	0.02
TextCNN	0.55	0.57	0.55	0.52	0.32
TextRNN	0.41	0.4	0.41	0.38	0.11
Transformer	0.4	0.43	0.4	0.4	0.1
fastText	0.33	0.23	0.33	0.24	0
Hierarchical	0.52	0.62	0.52	0.5	0.29
Hierarchical-1	0.6	0.72	0.6	0.57	0.4
Hierarchical-2	0.59	0.72	0.59	0.56	0.38
HierarchicalHard	0.5	0.39	0.5	0.43	0.25
HierarchicalHard-1	**0.61**	0.72	**0.61**	0.57	**0.41**
HierarchicalHard-2	0.6	0.71	0.6	0.56	0.39

As seen from Table 6.24, HierarchicalHard-1 and BoostingTree models are the best performing models in testing phase. Even though SVC model has highest precision score, its other metrics are low, Hiearchical-1 model on the other hand has the third highest Cohen-Kappa score among other models.

Table 6.25 Experiment 5 Asset Type Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
BoostingTree	Confidence Based	0.73	0.8	0.73	0.69	0.59
	Count Based	**0.8**	**0.84**	**0.8**	**0.79**	**0.7**
	Class Weight Based	0.76	0.78	0.76	0.75	0.64
Hierarchical-1	Confidence Based	0.67	0.78	0.67	0.63	0.51
	Count Based	0.67	0.8	0.67	0.64	0.54
	Class Weight Based	0.63	0.68	0.63	0.59	0.44
HierarchicalHard-1	Confidence Based	0.68	0.77	0.68	0.64	0.52
	Count Based	0.6	0.46	0.6	0.52	0.48
	Class Weight Based	0.65	0.72	0.65	0.61	0.48

As seen from Table 6.25, BoostingTree model is clearly the best performing model for this experiment in cryptocurrency-based prediction. Even though, HierarchicalHard-1 and Hierarchical-1 models had competitive results in document prediction task, they have failed in cryptocurrency prediction task. Count Based scheme has the best scores on all metrics and clearly the winner of this experiment.

6.1.2.2.6 Experiment 6: Experiments on Under Sampled Dataset Generated by Lemmatization Followed by Stemming

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization followed by stemming as text normalization.

As seen from Figure B.6, Hierarchical-1 and Hierarchical-2 models and their variations are close winners of training and validation phase. LinearSVC and RandomForest models also have promising but lower scores.

Table 6.26 Experiment 6 Balanced Document Based Asset Type Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.67	0.71	0.67	0.66	0.5
SVC	0.62	0.69	0.62	0.56	0.42
BoostingTree	0.58	0.59	0.58	0.57	0.37
DecisionTree	0.42	0.44	0.42	0.43	0.13
NaiveBayes	0.69	**0.75**	0.69	0.67	0.53
RandomForest	**0.7**	0.73	**0.7**	**0.7**	**0.55**
RCNN	0.51	0.48	0.51	0.43	0.26
RNN-Attention	0.33	0.24	0.33	0.28	0
TextCNN	0.52	0.54	0.52	0.51	0.28
TextRNN	0.36	0.33	0.36	0.33	0.03
Transformer	0.4	0.41	0.4	0.4	0.1
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.52	0.36	0.52	0.42	0.28
Hierarchical-1	0.58	0.63	0.58	0.58	0.37
Hierarchical-2	0.64	0.67	0.64	0.63	0.46
HierarchicalHard	0.56	0.57	0.56	0.55	0.34
HierarchicalHard-1	0.65	0.69	0.65	0.63	0.47
HierarchicalHard-2	0.61	0.64	0.61	0.59	0.42

As seen from Table 6.26, even though Hierarchical-1 and Hierarchical-2 models have performed better on training and validation phase, RandomForest model is the best performing model in testing phase followed by NaiveBayes and LinearSVC models.

Table 6.27 Experiment 6 Asset Type Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.62	0.66	0.62	0.57	0.43
	Count Based	0.62	0.66	0.62	0.57	0.43
	Class Weight Based	0.61	0.63	0.61	0.56	0.41
NaiveBayes	Confidence Based	**0.7**	**0.77**	**0.7**	0.66	**0.55**
	Count Based	0.69	**0.77**	0.69	0.65	0.54
	Class Weight Based	0.69	**0.77**	0.69	0.65	0.54
RandomForest	Confidence Based	**0.7**	0.76	**0.7**	**0.69**	**0.55**
	Count Based	0.67	0.73	0.67	0.66	0.51
	Class Weight Based	0.66	0.7	0.66	0.65	0.49

As seen from Table 6.27, RandomForest's Confidence Based scheme and NaiveBayes's Confidence Based scheme are best performing models. Yet, RandomForest's Confidence Based scheme is more preferable for this experiment by having higher F1 score.

6.1.2.3 Transaction Anonymity

In this section, we have presented our experiment results of Transaction Anonymity prediction task. There is an experiment for each dataset generated by different text normalization techniques (Experiment 1, 3, 5) and their semi-balanced variations generated by under sampling Pseudonymous and Anonymous class instances randomly to have twenty (20) unique cryptocurrency instances (Experiment 2, 4, 6).

6.1.2.3.1 Experiment 1: Experiments on Dataset Generated by Stemming

For this experiment, we have used the dataset version which we have generated by using stemming as text normalization. As seen from Figure C.1, there is a competition between couple of models. In terms of accuracy, Hierarchical-1 model is the best model. For other metrics, NaiveBayes, LinearSVC, HierarchicalHard and HierarchicalHard-1 models are close winners of the performance scoring in training and validation phase.

Table 6.28 Experiment 1 Balanced Document Based Transaction Anonymity Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.54	0.37	0.54	0.43	0.31
SVC	0.54	0.36	0.54	0.43	0.31
BoostingTree	0.48	0.35	0.48	0.4	0.23
DecisionTree	0.46	0.31	0.46	0.37	0.19
NaiveBayes	0.66	0.75	0.66	0.62	0.49
RandomForest	0.53	0.36	0.53	0.43	0.3
RCNN	0.4	0.48	0.4	0.37	0.09
RNN-Attention	0.32	0.32	0.32	0.31	-0.02
TextCNN	0.43	0.29	0.43	0.34	0.14
TextRNN	0.27	0.22	0.27	0.24	-0.09
Transformer	0.27	0.23	0.27	0.24	-0.1
fastText	0.28	0.11	0.28	0.15	-0.07
Hierarchical	0.5	0.34	0.5	0.4	0.25
Hierarchical-1	0.76	0.76	0.76	0.75	0.64
Hierarchical-2	**0.86**	**0.86**	**0.86**	**0.86**	**0.79**
HierarchicalHard	0.55	0.38	0.55	0.45	0.33
HierarchicalHard-1	0.53	0.36	0.53	0.43	0.29
HierarchicalHard-2	0.6	0.61	0.6	0.57	0.4

Even though NaiveBayes, LinearSVC and HierarchicalHard models have competitive scores in validation phase, but as seen from Table 6.28, they have failed to generalize their performance on test set. However, Hierarchical-2 is able to generalize its performance on test set and it is the best model by large margin in testing phase. NaiveBayes and Hiearchical-1 are the next two models selected for further investigation in terms of Cohen-Kappa scores.

Table 6.29 Experiment 1 Transaction Anonymity Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
NaiveBayes	Confidence Based	0.58	0.42	0.58	0.48	0.36
	Count Based	0.57	0.42	0.57	0.47	0.36
	Class Weight Based	0.57	0.41	0.57	0.47	0.35
Hierarchical-1	Confidence Based	0.8	0.8	0.8	0.79	0.7
	Count Based	0.71	0.8	0.71	0.73	0.69
	Class Weight Based	0.75	0.76	0.75	0.74	0.63
Hierarchical-2	Confidence Based	**0.88**	0.88	**0.88**	**0.88**	0.82
	Count Based	0.81	**0.95**	0.81	0.87	**0.94**
	Class Weight Based	0.87	0.89	0.87	0.87	0.81

As seen from Table 6.29, as expected, variations of Hierarchical-2 model have the highest scores by large margin in this experiment. In terms of cryptocurrency prediction schemes, Confidence Based scheme and Count Based scheme are in close competition. Yet, Count Based scheme has much higher Cohen-Kappa and Precision scores therefore it is more preferable for this experiment.

6.1.2.3.2 Experiment 2: Experiments on Under Sampled Dataset Generated by Stemming

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization as text normalization. As seen from Figure C.2, HierarchicalHard model has the highest scores on all metrics by a small margin. RandomForest and SVC models also have closely higher scores compared to other models in training and validation phase.

Table 6.30 Experiment 2 Balanced Document Based Transaction Anonymity Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.52	0.38	0.52	0.43	0.28
SVC	0.51	0.35	0.51	0.41	0.27
BoostingTree	0.44	0.32	0.44	0.37	0.16
DecisionTree	**0.57**	**0.57**	**0.57**	**0.57**	**0.36**
NaiveBayes	0.53	0.39	0.53	0.44	0.3
RandomForest	0.45	0.34	0.45	0.38	0.17
RCNN	0.38	0.44	0.38	0.36	0.07
RNN-Attention	0.33	0.24	0.33	0.27	-0.01
TextCNN	0.27	0.22	0.27	0.23	-0.1
TextRNN	0.28	0.32	0.28	0.27	-0.08
Transformer	0.23	0.2	0.23	0.21	-0.15
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.48	0.39	0.48	0.42	0.22
Hierarchical-1	0.44	0.37	0.44	0.39	0.16
Hierarchical-2	0.43	0.35	0.43	0.37	0.15
HierarchicalHard	0.5	0.38	0.5	0.42	0.25
HierarchicalHard-1	0.49	0.4	0.49	0.42	0.23
HierarchicalHard-2	0.46	0.36	0.46	0.39	0.19

As seen from Table 6.30, even though HierarchicalHard has the best scores in validation phase, it fails to show the same performance on testing dataset. In testing phase, DecisionTree model outperforms others by high margin. LinearSVC and NaiveBayes models are the other models selected for further investigation considering Cohen-Kappa scores of the models.

Table 6.31 Experiment 2 Transaction Anonymity Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.58	0.44	0.58	0.48	0.37
	Count Based	0.58	0.44	0.58	0.48	0.37
	Class Weight Based	0.56	0.41	0.56	0.47	0.35
DecisionTree	Confidence Based	0.47	0.37	0.47	0.41	0.21
	Count Based	**0.78**	**0.79**	**0.78**	**0.77**	**0.67**
	Class Weight Based	0.72	0.76	0.72	0.7	0.58
NaiveBayes	Confidence Based	0.57	0.43	0.57	0.48	0.36
	Count Based	0.57	0.42	0.57	0.47	0.36
	Class Weight Based	0.56	0.41	0.56	0.47	0.35

As seen from Table 6.31, as expected DecisionTree model is the best among others in cryptocurrency-based label prediction task. In addition, best model for this prediction task is Count Based scheme of DecisionTree model.

6.1.2.3.3 Experiment 3: Experiments on Dataset Generated by Lemmatization

For this experiment, we have used the dataset version which we have generated by using lemmatization as text normalization. As seen from Figure C.3, there is a competition between couple of models. In terms of accuracy, DecisionTree and Hierarchical-2 are the best performing models. For other metrics, NaiveBayes, Hierarchical-2 and HierarchicalHard models are close winners of the performance scoring. Hierarchical-2 is more preferable due to consistent scores on all metrics.

Table 6.32 Experiment 3 Balanced Document Based Transaction Anonymity Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.55	0.39	0.55	0.45	0.33
SVC	0.56	0.37	0.56	0.45	0.34
BoostingTree	0.5	0.35	0.5	0.41	0.26
DecisionTree	0.37	0.28	0.37	0.31	0.06
NaiveBayes	0.58	0.39	0.58	0.47	0.38
RandomForest	0.53	0.36	0.53	0.43	0.3
RCNN	0.43	0.3	0.43	0.33	0.14
RNN-Attention	0.19	0.29	0.19	0.15	-0.21
TextCNN	0.46	0.46	0.46	0.46	0.19
TextRNN	0.34	0.25	0.34	0.29	0.01
Transformer	0.29	0.23	0.29	0.25	-0.06
fastText	0.33	0.11	0.33	0.17	0.0
Hierarchical	0.52	0.37	0.52	0.43	0.28
Hierarchical-1	0.54	0.45	0.54	0.47	0.31
Hierarchical-2	**0.83**	**0.83**	**0.83**	**0.83**	**0.75**
HierarchicalHard	0.49	0.37	0.49	0.4	0.24
HierarchicalHard-1	0.56	0.39	0.56	0.45	0.33
HierarchicalHard-2	0.62	0.67	0.62	0.59	0.43

Even though NaiveBayes, DecisionTree and HierarchicalHard models have competitive scores in validation phase, as seen from Table 6.32, they have failed to generalize their performance on test set. However, Hierarchical-2 is able to generalize its performance on test set and is the best model by large margin of testing phase. NaiveBayes and HiearchicalHard-2 are the next two selected models for further investigation in terms of Cohen-Kappa.

Table 6.33 Experiment 3 Transaction Anonymity Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
NaiveBayes	Confidence Based	0.57	0.45	0.57	0.48	0.36
	Count Based	0.57	0.41	0.57	0.47	0.36
	Class Weight Based	0.56	0.41	0.56	0.47	0.35
Hierarchical-2	Confidence Based	**0.88**	**0.88**	**0.88**	**0.88**	**0.82**
	Count Based	0.74	0.85	0.74	0.78	0.8
	Class Weight Based	0.8	0.8	0.8	0.79	0.7
HierarchicalHard-2	Confidence Based	0.53	0.4	0.53	0.44	0.3
	Count Based	0.53	0.4	0.53	0.45	0.31
	Class Weight Based	0.51	0.39	0.51	0.43	0.26

As seen from Table 6.33, as expected variations of Hierarchical-2 model have the highest scores by large margin in this experiment. In terms of cryptocurrency prediction schemes, Confidence Based scheme is the best performing scheme for this experiment.

6.1.2.3.4 Experiment 4: Experiments on Under Sampled Dataset Generated by Lemmatization

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization as text normalization.

Table 6.34 Experiment 4 Balanced Document Based Transaction Anonymity Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.52	0.38	0.52	0.43	0.28
SVC	0.51	0.39	0.51	0.43	0.27
BoostingTree	0.43	0.31	0.43	0.36	0.15
DecisionTree	0.42	0.31	0.42	0.35	0.13
NaiveBayes	0.53	0.36	0.53	0.43	0.29
RandomForest	0.49	0.36	0.49	0.41	0.24
RCNN	0.31	0.25	0.31	0.26	-0.04
RNN-Attention	0.45	0.5	0.45	0.37	0.18
TextCNN	0.35	0.27	0.35	0.3	0.03
TextRNN	0.27	0.2	0.27	0.23	-0.09
Transformer	0.47	0.46	0.47	0.46	0.21
fastText	0.32	0.11	0.32	0.17	-0.02
Hierarchical	0.47	0.36	0.47	0.4	0.2
Hierarchical-1	0.48	0.38	0.48	0.4	0.21
Hierarchical-2	**0.72**	**0.72**	**0.72**	**0.71**	**0.59**
HierarchicalHard	0.55	0.57	0.55	0.53	0.32
HierarchicalHard-1	0.47	0.34	0.47	0.39	0.21
HierarchicalHard-2	0.47	0.33	0.47	0.39	0.21

As seen from Figure C.4, in terms of validation accuracy, DecisionTree, BoostingTree and HierarchicalHard-2 are the best three models. In terms of F1

validation score, HierarchicalHard-2 and BoostingTree are the best models. In terms of Cohen-Kappa validation score, LinearSVC, NaiveBayes and HierarchicalHard-2 are the best models. In terms of validation scores, HierarchicalHard-2 and BoostingTree are feasible options for transaction anonymity document-based prediction task.

As seen from Table 6.34, even though BoostingTree and HierarchicalHard-2 have the best scores in validation phase, they fail to show the same performance at testing phase. In testing phase, Hierarchical-2 model performs best among others by high margin. NaiveBayes and HierarchicalHard models are the other models selected for further investigation considering Cohen-Kappa scores of the models.

Table 6.35 Experiment 4 Transaction Anonymity Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
NaiveBayes	Confidence Based	0.57	0.42	0.57	0.47	0.35
	Count Based	0.56	0.42	0.56	0.47	0.34
	Class Weight Based	0.56	0.41	0.56	0.46	0.34
Hierarchical-2	Confidence Based	0.69	0.7	0.69	0.66	0.53
	Count Based	**0.71**	**0.76**	**0.71**	**0.72**	**0.65**
	Class Weight Based	**0.71**	0.72	**0.71**	0.69	0.57
HierarchicalHard	Confidence Based	0.55	0.41	0.55	0.46	0.32
	Count Based	0.5	0.4	0.5	0.43	0.31
	Class Weight Based	0.49	0.37	0.49	0.42	0.23

As seen from Table 6.35, as expected Hierarchial-2 model is the best among others in every label selection scheme. In addition, best scheme for cryptocurrency label prediction for Hierarchical-2 model is Count Based scheme.

6.1.2.3.5 Experiment 5: Experiments on Dataset Generated by Lemmatization Followed by Stemming

For this experiment, we have used the dataset version which we have generated by using lemmatization followed by stemming as text normalization. As seen from Figure C.5, there is a competition between couple of models, in terms of accuracy, DecisionTree and HierarchicalHard are the best models. For other metrics, NaiveBayes model has the highest scores in training and validation phase.

Table 6.36 Experiment 5 Balanced Document Based Transaction Anonymity Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.54	0.36	0.54	0.43	0.31
SVC	0.55	0.37	0.55	0.44	0.33
BoostingTree	0.48	0.34	0.48	0.4	0.22
DecisionTree	0.34	0.28	0.34	0.3	0.01
NaiveBayes	0.57	0.39	0.57	0.46	0.36
RandomForest	0.54	0.38	0.54	0.44	0.31
RCNN	0.45	0.32	0.45	0.36	0.18
RNN-Attention	0.33	0.22	0.33	0.26	-0.01
TextCNN	0.33	0.29	0.33	0.3	-0.01
TextRNN	0.44	0.48	0.44	0.44	0.16
Transformer	0.25	0.25	0.25	0.25	-0.12
fastText	0.39	0.27	0.39	0.31	0.08
Hierarchical	0.51	0.41	0.51	0.44	0.26
Hierarchical-1	0.52	0.44	0.52	0.45	0.28
Hierarchical-2	**0.82**	**0.83**	**0.82**	**0.82**	**0.74**
HierarchicalHard	0.5	0.35	0.5	0.41	0.25
HierarchicalHard-1	0.56	0.39	0.56	0.46	0.34
HierarchicalHard-2	0.62	0.66	0.62	0.59	0.43

Even though NaiveBayes and HierarchicalHard models have competitive scores in validation phase, as seen from Table 6.36, they have failed to generalize their performance on test set. However, Hierarchical-2 is able to outperform other models in testing phase and is the best model by large margin. NaiveBayes and HiearchicalHard-2 are the next two selected models for further investigation in terms of Cohen-Kappa.

Table 6.37 Experiment 5 Transaction Anonymity Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
NaiveBayes	Confidence Based	0.58	0.42	0.58	0.48	0.37
	Count Based	0.57	0.42	0.57	0.48	0.36
	Class Weight Based	0.57	0.41	0.57	0.47	0.35
Hierarchical-2	Confidence Based	**0.89**	**0.89**	**0.89**	**0.89**	**0.84**
	Count Based	0.75	0.88	0.75	0.79	0.82
	Class Weight Based	0.79	0.8	0.79	0.79	0.69
HierarchicalHard-2	Confidence Based	0.53	0.4	0.53	0.44	0.29
	Count Based	0.53	0.4	0.53	0.44	0.3
	Class Weight Based	0.5	0.39	0.5	0.42	0.25

As seen from Table 6.37, as expected variations of Hierarchical-2 model have the highest scores by large margin in this experiment. In terms of cryptocurrency prediction schemes, Confidence Based scheme is best performing scheme for this experiment.

6.1.2.3.6 Experiment 6: Experiments on Under Sampled Dataset Generated by Lemmatization Followed by Stemming

For this experiment, we have used the semi-balanced dataset version which we have generated by using lemmatization followed by stemming as text normalization. As seen from Figure C.6, in terms of validation accuracy, DecisionTree is the best model. In other scoring metrics, HierarchicalHard-2 has the distinctive scores in training and validation phase.

Table 6.38 Experiment 6 Balanced Document Based Transaction Anonymity Test Results

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
LinearSVC	0.53	0.36	0.53	0.43	0.29
SVC	0.5	0.36	0.5	0.41	0.25
BoostingTree	0.48	0.35	0.48	0.4	0.23
DecisionTree	0.37	0.3	0.37	0.32	0.05
NaiveBayes	0.53	0.36	0.53	0.43	0.29
RandomForest	0.51	0.35	0.51	0.41	0.27
RCNN	0.31	0.24	0.31	0.27	-0.04
RNN-Attention	0.38	0.37	0.38	0.36	0.07
TextCNN	0.41	0.41	0.41	0.4	0.12
TextRNN	0.41	0.42	0.41	0.41	0.11
Transformer	0.5	0.5	0.5	0.48	0.25
fastText	0.33	0.11	0.33	0.17	0
Hierarchical	0.48	0.34	0.48	0.4	0.22
Hierarchical-1	0.47	0.38	0.47	0.39	0.2
Hierarchical-2	**0.73**	**0.72**	**0.73**	**0.71**	**0.59**
HierarchicalHard	0.55	0.58	0.55	0.53	0.32
HierarchicalHard-1	0.47	0.34	0.47	0.39	0.21
HierarchicalHard-2	0.48	0.34	0.48	0.4	0.22

As seen from Table 6.38, even though DecisionTree and HierarchicalHard-2 have the best scores in validation phase, they fail to show the same performance at testing phase. In testing phase, Hierarchical-2 model outperforms others by high margin. LinearSVC and HierarchicalHard are the other models selected for further investigation considering Cohen-Kappa scores of the models.

Table 6.39 Experiment 6 Transaction Anonymity Cryptocurrency Based Label Prediction Balanced Test Results of 3 Best Models

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
LinearSVC	Confidence Based	0.55	0.4	0.55	0.46	0.32
	Count Based	0.55	0.4	0.55	0.46	0.32
	Class Weight Based	0.55	0.4	0.55	0.46	0.33
Hierarchical-2	Confidence Based	0.69	0.7	0.69	0.67	0.54
	Count Based	**0.72**	**0.78**	**0.72**	**0.74**	**0.68**
	Class Weight Based	**0.72**	0.72	**0.72**	0.7	0.58
HierarchicalHard	Confidence Based	0.55	0.41	0.55	0.46	0.32
	Count Based	0.49	0.4	0.49	0.43	0.31
	Class Weight Based	0.49	0.37	0.49	0.42	0.23

As seen from Table 6.39, as expected Hierarchial-2 model is best among others in every label selection scheme. In addition, best scheme for cryptocurrency label prediction is Count Based scheme of Hierarchical-2.

6.1.3 Discussion

In this section, we will discuss our cryptocurrency-based test results given in Section 6 for each prediction task.

Neural Network models were unable to achieve competitive results for any of the tasks. Neural Networks are complex models which require large number of training samples. Due to high class imbalance and lack of having enough training examples, we have yet to achieve competitive scores on neural network models. In addition, due to manual tuning process, we may also fall short to find the global optimum setting.

In terms of label mapping schemes, Class Weight Based scheme was unable to achieve best performance in any of the experiments. This scheme is prune to false positives when the competition is between a high frequency label and low frequency label. Even though, there are multiple votes for frequent label, having one false prediction on less frequent label could lead to incorrect results. Therefore, since all experiments are conducted on imbalanced datasets, we believe this scheme is failed to achieve competitive results

Table 6.40 Cryptocurrency Based Test Results of Sector

Model	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
(Exp1) SVC	Confidence Based	0.83	0.82	0.83	0.83	0.72
(Exp2) NaiveBayes	Confidence Based	0.82	0.84	0.82	0.83	0.73
(Exp3) Hierarchical	Confidence Based	0.84	**0.85**	0.84	0.84	0.74
(Exp4) NaiveBayes	Confidence Based	0.82	0.84	0.82	0.82	0.72
(Exp5) SVC	Confidence Based	**0.85**	**0.85**	**0.85**	**0.85**	**0.76**
(Exp6) NaiveBayes	Confidence Based	0.81	0.83	0.81	0.82	0.72

We believe that traditional language characteristics were unable to cover language characteristics introduced by cryptocurrency literature. Therefore, WordNet alone,

was not able to map characteristic words correctly to their roots and all prediction tasks performed better on datasets that includes the touch of Stemming.

Sector. As seen from Table 3.40, Confidence Based scheme of SVC applied on lemmatization followed by stemming dataset performed best among experiments. For all experiments, Confidence based mapping scheme achieved the better scores on all metrics. We believe that, due to large number of classes, our Count Based scheme is failed because of its tiebreaker logic, using the Confidence Based scheme as a tie breaker could improve Count based results.

In addition, except Experiment 3 (Section 6.1.2.1.3), our hierarchical models failed to compete with other models for sector prediction tasks. We believe the reason for that since it has large number of classes and brute force testing all combinations of possible hierarchies is not feasible, we may have failed to find optimal hierarchy. In addition, as number of leaves in hierarchical tree increased, the number of models to train increase and thus our possibility of making errors. Applying a different methodology which also considers multiway hierarchies (as our intuitive Hierarchical model) would improve hierarchical model results.

Table 6.41 Cryptocurrency Based Balanced Test Results of Asset Type

Model	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
(Exp1) RandomForest	Confidence Based	0.77	0.82	0.77	0.75	0.65
(Exp2) Hierarchical-2	Count Based	**0.81**	0.83	**0.81**	**0.82**	**0.75**
(Exp3) Hierarchical-1	Count Based	0.72	0.82	0.72	0.7	0.6
(Exp4) NaiveBayes	Count Based	0.68	0.77	0.68	0.64	0.52
(Exp5) BoostingTree	Count Based	0.8	**0.84**	0.8	0.79	0.7
(Exp6) RandomForest	Confidence Based	0.7	0.76	0.7	0.69	0.55

Asset Type. As seen from Table 6.41, Count Based prediction scheme of Hierarchial-2 model achieved the best performance by having higher scores in four (4) metrics. For this task, as a semi-balanced dataset performed best, we believe that

eliminating excessive data which may distort solution space increased the performance of the winning model.

As explained in Section 3.1.2, both Utility Token and Security Token labeled assets represent rights on goods and services which in a way contains some similar characteristics. Therefore, we believe such hierarchy that groups these types as given in Figure F.3 achieved better performance than other options.

Table 6.42 Cryptocurrency Based Balanced Test Results of Transaction Anonymity

Model	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
(Exp1) Hierarchical-2	Count Based	0.81	**0.95**	0.81	0.87	**0.94**
(Exp2) DecisionTree	Count Based	0.78	0.79	0.78	0.77	0.67
(Exp3) Hierarchical-2	Confidence Based	0.88	0.88	0.88	0.88	0.82
(Exp4) Hierarchical-2	Count Based	0.71	0.76	0.71	0.72	0.65
(Exp5) Hierarchical-2	Confidence Based	**0.89**	0.89	**0.89**	**0.89**	0.84
(Exp6) Hierarchical-2	Count Based	0.72	0.78	0.72	0.74	0.68

Transaction Anonymity. As seen from Table 3.42, best models from Experiment 1 (Section 6.1.2.3.1) and Experiment 5 (Section 6.1.2.3.5) are in close competition for this prediction task. For imbalanced datasets, accuracy may be a misleading metric but since the results are given in balanced test sets, we also considered it as important as others. Model used in Experiment 5 (Section 6.1.2.3.5) has higher scores on three (3) metrics including accuracy and therefore it is selected as best for this task. In addition, in all experiments except Experiment 2 (Section 6.1.2.3.2), Hierarchical-2 and its mapping variations achieved best scores. As explained in Section 3.1.3, Pseudonymous and Anonymous tokens have different characteristics while Selective Transparent tokens includes characteristics from both classes thus fell in a gray area. Therefore, we believe that a hierarchy which first separates Selective Transparency from others as given in F.3 achieved better performance than other variations.

6.2 News Sentiment Analysis

In this section, we have explained and demonstrate how we evaluate the proposed model on news sentiment analysis. There are different datasets used for evaluating performance of respective models on different components of a news. These datasets are explained below.

Table 6.43 Sentiment Label Distribution of News Title, Summary and Content

Sentiment Label	***Title***	***Summary***	***Content***
Positive	26	26	16
Negative	17	15	23
Neutral	30	26	11
Total	**73**	**64**	**50**

As explained in Section 5.1.2, one hundred (100) randomly selected news are annotated by three (3) experts. We have only used annotations which the majority experts agreed on. Table 6.43 shows the number of title, summary and content annotations which experts agreed on. This dataset given in Table 6.43 is only used for comparing performance of the selected models.

Table 6.44 Sentence Sentiment Label Distribution

Sentiment Label	***# Sentences***
Positive	647
Negative	478
Neutral	2111
Total	**3236**

In Table 6.44, we have provided the sentence dataset which we have used for training, validating and testing our model. As seen, there is an imbalance between sentences indicates polarity (negative and positive) and sentences that does not

indicate any polarity (neutral sentences). Approximately %65 sentences in dataset are neutral sentences.

6.2.1 Experiment Setting

Before starting our experiments, we have divided our initial sentence data train and test sets with 9:1 ratio. Remaining training data is split into training and validation sets with also 9:1 ratio. Content dataset also filtered with respect to selected test sentences since training sentences are selected from them and may cause bias in testing phase.

In our experiments, to compare our proposed model, we have adapted two dictionary based methods; Vader which achieves competitive scores on analyzing sentiment of short text/tweets [73] and Loughran-McDonald (ML) dictionary which is based on financial domain [32]. In order to use the dictionary provided by Lougran and McDonald, we have adapted Vader algorithm to work with Lougran-McDonald dictionary. We have also accepted vanilla FinBERT model as one of baseline models.

Model Selection. In each epoch, model is evaluated with Cross Entropy Loss as recommended by FinBERT paper. Model achieves the minimum validation loss among training epochs is selected for testing evaluation. We have also feed our loss function with class weights to eliminate the effects of class imbalance in validation set.

6.2.2 Experiments

In this section we have demonstrated the experiments we performed for evaluating the proposed model performance. There are four (4) experiments in total and in each experiment, we have presented the performance of the model on training, validation, test sets and best model's performance on title, summary and content sets. Evaluation of models on title, summary and content datasets are given at Section 6.2.3 for best models from each experiment and baseline models. Performance of baseline methods are given in Section 6.2.2.5.

6.2.2.1 Experiment 1: Fine-Tuning FinBERT on Initial Sentence Dataset

In this experiment, we have trained FinBERT model for four (4) epochs with batch size 32 (264 steps per epoch). Different versions of the model are generated by freezing some layers of the model. In this experiment, "Layer 12" means all the layers are included in training and "Layer 6" means first six (6) layers (low level features) are frozen and left untouched during training.

Table 6.45 Sentence Sentiment Analysis Test Results of Experiment 1

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
FinBERT Layer 0	0.31	**0.74**	0.31	0.29	**0.12**
FinBERT Layer 2	0.31	**0.74**	0.31	0.29	**0.12**
FinBERT Layer 4	0.29	0.71	0.29	0.27	0.11
FinBERT Layer 6	0.31	0.72	0.31	0.29	**0.12**
FinBERT Layer 8	0.28	0.7	0.28	0.25	0.1
FinBERT Layer 10	**0.33**	0.68	**0.33**	**0.32**	**0.12**
FinBERT Layer 12	0.27	0.63	0.27	0.22	0.09

As seen from Figure G.1, in training and validation phase, FinBERT Layer 12 model performed best with respect to accuracy and precision. For other metrics, baseline models performed better than trained FinBERT models. For this experiment, including lower level features in training increases validation scores and reduces overfitting.

As seen form Table 6.45, in testing phase training FinBERT by freezing last two (2) layers has the highest scores in accuracy, recall and F1. Even though training only linear layer has higher precision, FinBERT Layer 10 model is the best among others.

Table 6.46 FinBERT Layer 10's Performance on Title, Summary and Content Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Title	Mean	**0.53**	**0.55**	**0.53**	**0.47**	**0.32**
	Median	**0.53**	**0.55**	**0.53**	**0.47**	**0.32**
	Min-Max	**0.53**	**0.55**	**0.53**	**0.47**	**0.32**
	Mod	**0.53**	**0.55**	**0.53**	**0.47**	**0.32**
Summary	Mean	**0.55**	**0.55**	**0.55**	**0.48**	**0.33**
	Median	**0.55**	**0.55**	**0.55**	**0.48**	**0.33**
	Min-Max	0.52	0.35	0.52	0.41	0.28
	Mod	0.53	0.35	0.53	0.42	0.3
Content	Mean	**0.54**	0.57	**0.54**	**0.51**	**0.34**
	Median	0.5	**0.73**	0.5	0.41	0.31
	Min-Max	0.44	0.25	0.44	0.32	0.23
	Mod	0.46	0.25	0.46	0.33	0.26

As seen from Table 6.46, for title sentiment analysis, all schemes have the same results, since titles of our dataset are made of single sentence. In summary sentiment analysis, both Mean and Median schemes have the same scores thus both are preferable. In content sentiment analysis, even though Median scheme has the highest score in precision, mean Mean is more preferable by having higher scores on four (4) metrics.

6.2.2.2 Experiment 2: Effects of Further Training Fine-Tuned FinBERT Model on Initial Dataset

In this experiment, we have further trained the models which we had in Experiment 1 (Section 6.2.2.1) for four (4) more epochs to see how it effects the results. Layer freezing logic is same with the Experiment 1 (Section 6.2.2.1).

As seen from Figure G.2, the scenario is almost the same as Experiment 1 (Section 6.2.2.1) except Layer-8 and Layer-10 models have better validation scores on Cohen-Kappa than base models. All models' validation scores seem improved with further training with respect to Experiment 1 (Section 6.2.2.1) metric scores.

Table 6.47 Sentence Sentiment Analysis Test Results of Experiment 2

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
FinBERT Layer 0	0.3	0.71	0.3	0.28	0.11
FinBERT Layer 2	0.3	0.71	0.3	0.28	0.11
FinBERT Layer 4	0.29	**0.73**	0.29	0.26	0.12
FinBERT Layer 6	**0.33**	0.71	**0.33**	**0.32**	**0.13**
FinBERT Layer 8	0.32	0.69	0.32	0.31	0.11
FinBERT Layer 10	0.32	0.67	0.32	0.31	0.11
FinBERT Layer 12	0.29	0.7	0.29	0.26	0.11

As seen from Table 6.47, best layer configuration for this experiment is FinBERT model with its last six (6) layers are frozen during training. Compared to Experiment 1 (Section 6.2.2.1), further training increased scores for models that include low level layers (FinBERT Layer 6, 8, 10, 12). On the contrary, we believe due to overfitting, further training decreased scores for models that does not include lower layers (FinBERT Layer 0, 2, 4). Despite FinBERT Layer 10 model has the highest scores on Experiment 1 (Section 6.2.2.1), FinBERT Layer 6 seems to achieve better generalization on testing set.

Table 6.48 FinBERT Layer 6's Performance on Title, Summary and Content Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Title	Mean	**0.49**	**0.38**	**0.49**	**0.4**	**0.26**
	Median	**0.49**	**0.38**	**0.49**	**0.4**	**0.26**
	Min-Max	**0.49**	**0.38**	**0.49**	**0.4**	**0.26**
	Mod	**0.49**	**0.38**	**0.49**	**0.4**	**0.26**
Summary	Mean	0.58	0.6	0.58	0.52	0.37
	Median	**0.59**	**0.74**	**0.59**	**0.51**	**0.4**
	Min-Max	0.58	**0.74**	0.58	0.48	0.38
	Mod	0.58	**0.74**	0.58	0.48	0.37
Content	Mean	**0.64**	**0.74**	**0.64**	**0.62**	**0.48**
	Median	0.5	0.52	0.5	0.39	0.31
	Min-Max	0.42	0.25	0.42	0.3	0.21
	Mod	0.5	0.28	0.5	0.36	0.32

As seen from Table 6.48, as same title sentiment metrics have same scores for all schemes as expected. In addition, for summary dataset Median scheme has the highest scores on all metrics and for content dataset Mean scheme has the highest scores on all metrics for FinBERT Layer 6 model.

6.2.2.3 Experiment 3: Effects of Increasing Training Data on Fine-Tuning FinBERT

In this experiment, we have trained FinBERT model by increasing the training data with the additional sentences. We have evaluated models in this experiment with same test dataset we have used for evaluating Experiment 1 (Section 6.2.2.1) and Experiment 2 (Section 6.2.2.2). The aim of this experiment is to observe the effects of increased training data. As we did for Experiment 1 (Section 6.2.2.1), we have trained FinBERT model for four (4) epochs with batch size 32 (340 steps per epoch). Layer freezing logic is same with the other experiments.

As seen from Figure G.3, training and validation phase is similar to what we presented in Experiment 1 (Section 6.2.2.1). For accuracy and precision validation scores, trained FinBERT models performed better and for recall and F1 validation scores, baseline models performed better. But, models in this experiment have slightly better Cohen-Kappa validation scores compared to models in Experiment 1 (Section 6.2.2.1).

Table 6.49 Sentence Sentiment Analysis Test Results of Experiment 3

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
FinBERT Layer 0	0.32	0.71	0.32	0.3	0.12
FinBERT Layer 2	0.32	0.71	0.32	0.3	0.12
FinBERT Layer 4	0.31	0.7	0.31	0.3	0.12
FinBERT Layer 6	0.3	0.68	0.3	0.28	0.11
FinBERT Layer 8	0.32	0.66	0.32	0.3	0.11
FinBERT Layer 10	0.34	0.69	0.34	0.33	0.13
FinBERT Layer 12	**0.37**	**0.76**	**0.37**	**0.36**	**0.17**

As seen from Table 6.49, for this experiment, model trained by including all layers in training performed best by having higher scores on all metrics. In addition, as it

seems introducing more training data improved models performances on same test dataset.

Table 6.50 FinBERT Layer 12's Performance on Title, Summary and Content Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Title	Mean	**0.56**	**0.56**	**0.56**	**0.51**	**0.35**
	Median	**0.56**	**0.56**	**0.56**	**0.51**	**0.35**
	Min-Max	**0.56**	**0.56**	**0.56**	**0.51**	**0.35**
	Mod	**0.56**	**0.56**	**0.56**	**0.51**	**0.35**
Summary	Mean	**0.61**	0.65	**0.61**	**0.57**	0.41
	Median	**0.61**	0.74	**0.61**	0.55	**0.42**
	Min-Max	**0.61**	**0.75**	**0.61**	0.54	**0.42**
	Mod	0.59	0.74	0.59	0.53	0.39
Content	Mean	**0.62**	**0.66**	**0.62**	**0.6**	**0.44**
	Median	0.56	0.66	0.56	0.49	0.38
	Min-Max	0.46	0.25	0.46	0.32	0.26
	Mod	0.48	0.26	0.48	0.34	0.28

As seen from the Table 6.50, same scenario applies for title and content sentiment scores as in Experiment 1 (Section 6.2.2.1) and Experiment 2 (Section 6.2.2.2). Title scores are same for all schemes. Mean scheme has the highest scores for content test data. Yet for summary sentiment analysis, this time Min-Max accumulation scheme performed better than other schemes for this experiment.

6.2.2.4 Experiment 4: Effects of Further Training of Fine-Tuned FinBERT Model on Increased Training Data

In this experiment, as we did in Experiment 2 (Section 6.2.2.2), we have further trained the models in Experiment 3 (Section 6.2.2.3) for four (4) more epochs.

As seen from Figure G.4, results are similar with Experiment 3 (Section 6.2.2.3). Trained FinBERT models have better scores than baseline models for accuracy and precision validation scores. FinBERT Layer 10 and FinBERT Layer 12 models have better scores than baseline models for Cohen-Kappa validation score.

Table 6.51 Sentence Sentiment Analysis Test Results of Experiment 4

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
FinBERT Layer 0	0.31	0.7	0.31	0.3	0.11
FinBERT Layer 2	0.31	**0.74**	0.31	0.3	0.12
FinBERT Layer 4	0.3	0.69	0.3	0.28	0.11
FinBERT Layer 6	0.31	0.67	0.31	0.29	0.11
FinBERT Layer 8	0.3	0.66	0.3	0.28	0.11
FinBERT Layer 10	0.35	0.66	0.35	0.35	0.13
FinBERT Layer 12	**0.38**	0.69	**0.38**	**0.38**	**0.15**

As seen from Table 6.51, as in Experiment 3 (Section 6.2.2.3), FinBERT Layer 12 performed best for this experiment. As it seems, exposing high level features to further training degrades model performances. On the contrary, exposing lower layers with higher layers for further training improves model performances. This effect can be observed by comparing FinBERT Layer 0, FinBERT Layer 2, FinBERT Layer 10, FinBERT Layer 12 models' performances with their respective versions in Experiment 3 (Section 6.2.2.3). This pattern may indicate overfitting as we prohibit low level features to adapt data and focus on train high level features for longer time.

Table 6.52 FinBERT Layer 12's Performance on Title, Summary and Content Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Title	Mean	**0.63**	**0.63**	**0.63**	**0.62**	**0.45**
	Median	**0.63**	**0.63**	**0.63**	**0.62**	**0.45**
	Min-Max	**0.63**	**0.63**	**0.63**	**0.62**	**0.45**
	Mod	**0.63**	**0.63**	**0.63**	**0.62**	**0.45**
Summary	Mean	0.58	0.62	0.58	0.55	0.37
	Median	0.59	0.69	0.59	0.56	0.4
	Min-Max	**0.61**	**0.76**	**0.61**	**0.57**	**0.42**
	Mod	0.59	0.74	0.59	0.56	0.4
Content	Mean	**0.58**	**0.65**	**0.58**	**0.55**	**0.4**
	Median	0.46	0.26	0.46	0.33	0.26
	Min-Max	0.42	0.24	0.42	0.3	0.21
	Mod	0.48	0.26	0.48	0.34	0.29

As seen from Table 6.52, best performing schemes are the same with the Experiment 3 (Section 6.2.2.3) for all sentiment analysis tasks. Yet, while we slightly better scores for title and summary sentiment analysis tasks, we have slightly worse scores for content sentiment analysis task compared to Experiment 3 (Section 6.2.2.3) scores.

6.2.2.5 Base Model Performances

Table 6.53 Performance of Baseline Methods on Sentence Test Set

Models	***Acc.***	***Pre.***	***Rec.***	***F1***	***Cohen-Kappa***
Vanilla FinBERT	0.41	**0.67**	0.41	0.45	**0.13**
ML	**0.51**	0.6	**0.51**	**0.54**	0.06
Vader	0.41	0.6	0.41	0.45	0.05

As seen from Table 6.53, ML model has higher accuracy, recall and F1 scores compared to other baseline models, but those metrics are prune to data imbalance problems. Since our data imbalanced, a metric like Cohen-Kappa (which is less prune to data imbalance) would be more preferable for model selection. Vanilla FinBERT has better Cohen-Kappa score than other models therefore, it is more suitable for sentence prediction task.

Table 6.54 Base Model Performances on Title Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Vanilla FinBERT	Mean	**0.62**	**0.63**	**0.62**	**0.61**	**0.43**
	Median	**0.62**	**0.63**	**0.62**	**0.61**	**0.43**
	Min-Max	**0.62**	**0.63**	**0.62**	**0.61**	**0.43**
	Mod	**0.62**	**0.63**	**0.62**	**0.61**	**0.43**
ML	Mean	0.36	0.4	0.36	0.31	-0.01
	Median	0.36	0.4	0.36	0.31	-0.01
	Min-Max	0.36	0.4	0.36	0.31	-0.01
	Mod	0.36	0.4	0.36	0.31	-0.01
Vader	Mean	0.48	0.53	0.48	0.43	0.16
	Median	0.48	0.53	0.48	0.43	0.16
	Min-Max	0.48	0.53	0.48	0.43	0.16
	Mod	0.48	0.53	0.48	0.43	0.16

In Table 6.54, we have provided baseline model performances on title sentiment analysis task. Since titles are contained of single sentences, there are no differences between different sentence score accumulation schemes. As seen from this table, Vanilla FinBERT performs better that other models by having higher scores on all metrics.

Table 6.55 Base Model Performances on Summary Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Vanilla FinBERT	Mean	0.58	0.59	0.58	0.55	0.37
	Median	0.58	0.59	0.58	0.55	0.37
	Min-Max	**0.59**	0.6	**0.59**	**0.56**	**0.39**
	Mod	0.56	0.57	0.56	0.53	0.34
ML	Mean	0.42	0.49	0.42	0.39	0.12
	Median	0.42	0.49	0.42	0.39	0.12
	Min-Max	0.45	0.52	0.45	0.43	0.17
	Mod	0.45	0.52	0.45	0.43	0.17
Vader	Mean	0.52	0.58	0.52	0.5	0.27
	Median	0.55	0.6	0.55	0.53	0.32
	Min-Max	0.56	**0.62**	0.56	0.55	0.34
	Mod	0.55	0.59	0.55	0.54	0.32

As seen from Table 6.55, the results for summary sentiment analysis are a little bit different than title sentiment analysis results. For this scenario, there is a competition between Min-Max scheme of Vader and Vanilla FinBERT model. Yet, Vanilla FinBERT is more preferable here by having better scores on four (4) metrics.

Table 6.56 Base Model Performances on Content Sentiment Analysis

Models	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
Vanilla FinBERT	Mean	0.52	**0.65**	0.52	0.46	0.33
	Median	0.46	0.5	0.46	0.37	0.26
	Min-Max	0.38	0.25	0.38	0.28	0.17
	Mod	0.44	0.25	0.44	0.32	0.23
ML	Mean	0.26	0.23	0.26	0.19	0
	Median	0.3	0.27	0.3	0.23	0.06
	Min-Max	0.38	0.26	0.38	0.29	0.17
	Mod	0.34	0.27	0.34	0.26	0.13
Vader	Mean	**0.6**	0.59	**0.6**	**0.59**	**0.35**
	Median	0.36	0.35	0.36	0.3	0.08
	Min-Max	0.36	0.2	0.36	0.26	0.12
	Mod	0.36	0.2	0.36	0.25	0.1

As seen from Table 6.56, the case is different than summary sentiment analysis for content sentiment analysis. For this task, Vader's Mean accumulation scheme performed better than other models by having highest scores on four (4) metrics.

6.2.3 Discussion

In this section, we will discuss the title, summary and content sentiment analysis scores that we have presented in Section 6.2.2.

For title sentiment analysis task, scheme discussion is out of context, since in our title dataset, all titles are made of single sentences. As seen form Table 6.57, model trained for eight (8) epochs on additional training data without any layer freezing mechanism used, performed best among other models. Even though the difference is small, our trained model performed better than best baseline model for title sentiment analysis task.

Table 6.57 Model Performances on Title Sentiment Analysis

Model	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
(Base) Vanilla FinBERT	0.62	**0.63**	0.62	0.61	0.43
(Exp1) FinBERT Layer 6	0.53	0.55	0.53	0.47	0.32
(Exp2) FinBERT Layer 10	0.49	0.38	0.49	0.4	0.26
(Exp3) FinBERT Layer 12	0.56	0.56	0.56	0.51	0.35
(Exp4) FinBERT Layer 12	**0.63**	**0.63**	**0.63**	**0.62**	**0.45**

As seen from Table 6.58, for summary sentiment analysis task as same as title sentiment analysis task, model trained for eight (8) epochs with additional data without any freezing mechanism performed better than others. Different from title sentiment analysis task, base model improvement is more obvious. In addition, best sentence accumulation scheme for this task is Min-Max. We believe the reason for that is news summaries may contain multiple sentences but one of the sentences expected to be punchline, most important aspect of the news. So, if the news is negative, punchline is expected to be most negative sentence and, vice versa. Min-Max scheme is suited for that scenario as it accepts the most positive or negative sentiment score as the given text's sentiment score.

Table 6.58 Model Performances on Summary Sentiment Analysis

Model	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
(Base) Vanilla FinBERT	Min-Max	0.59	0.6	0.59	0.56	0.39
(Exp1) FinBERT Layer 6	Mean/Median	0.55	0.55	0.55	0.48	0.33
(Exp2) FinBERT Layer 10	Median	0.59	0.74	0.59	0.51	0.4
(Exp3) FinBERT Layer 12	Min-Max	**0.61**	0.75	**0.61**	0.54	**0.42**
(Exp4) FinBERT Layer 12	Min-Max	**0.61**	**0.76**	**0.61**	**0.57**	**0.42**

As seen from Table 6.59, model trained for eight (8) epochs without additional training data with its last two (2) layers are frozen performed better than best baseline model and other experiment model. In addition, Mean scheme is the best scheme for all models in this task. It is expected that, if the content is either positive or negative in general, it should be dominated by either positive or negative sentence sentiment scores.

Table 6.59 Model Performances on Content Sentiment Analysis

Model	***Scheme***	***Acc.***	***Pre.***	***Rec.***	***F1***	***Ch.-Kp.***
(Base) Vader	Mean	0.6	0.59	0.6	0.59	0.35
(Exp1) FinBERT Layer 6	Mean	0.54	0.57	0.54	0.51	0.34
(Exp2) FinBERT Layer 10	Mean	**0.64**	**0.74**	**0.64**	**0.62**	**0.48**
(Exp3) FinBERT Layer 12	Mean	0.62	0.66	0.62	0.6	0.44
(Exp4) FinBERT Layer 12	Mean	0.58	0.65	0.58	0.55	0.4

For all sentiment analysis tasks, even though it is small for title sentiment analysis task, we have achieved some improvements on baseline model. Except content sentiment analysis task, adding more data seems to improve our model results. Since neural networks are complex models (base BERT model especially has 110 Million

parameters), we were already expecting to improve model performances by adding more data. Yet, it is nice to see an improvement with such a small dataset as we know we are on right track.

Table 6.60 Sentiment Analysis Examples

Sentences	***Vader***	***ML***	***(Exp. 4) FinBERT***
1 - Ripple Price (XRP) Approaching Key Support: BTC & ETH Declining	0.11	-0.79	0.4
2 - Bitcoin Tumbles to $9.8K, Investors Continue Plowing Crypto Into DeFi	0.0	0.0	-0.65
3 - US Attorney's Office Indicts Two Suspects in EtherDelta Hack	-0.34	-0.85	-0.61
4 - Ethereum plummets to $190 as Crypto Markets faltered	0.0	0.0	-0.905
5 - Last week cryptocurrency prices bounced around after a majority of coins dropped in value on August 21.	0.39	-0.7	-0.9
6 - Today on August 26, digital currency markets have gained around 1.52%, gathering $4 billion since the initial slump.	0.38	0.7	0.92
7 - Despite the volatility, cryptocurrencies have consolidated and a few speculators believe a breakout is on the cards that could send prices sky-high or plunge below the current support	0.4	0.0	0.39

There is an obvious space for fine-tuning and improvement as recent sentiment analysis [74][75] works with neural networks including FinBERT [61] achieves accuracies in %80-90 band on known sentiment datasets. Yet as indicated, we have manually tuned the models, therefore we may fall short to achieve optimum solution.

Due to lack of resources, we couldn't apply an automated tuning logic like we had in Classification Towards Taxonomy (Section 6.1.1), If we have enough resources in future, we may be able to improve our results even more by experimenting on large number of hyperparameter configurations. In addition, even though we have used a loss function that takes data imbalance in consideration, we may also achieve some improvement by using balanced batches during training.

Sentiment analysis task is a hard task due its subjective nature. There are some sentences that are perceived by most as positive and negatives, and there are also sentences which sits in a gray area perceived differently by the readers. We have commented on our model's performance by sharing some interesting sentences from our news dataset (Table 6.60).

Sentence 1 and 2 (Table 6.60) are the examples of one of problems that we faced mostly and prioritized to solve it in future. In sentence 1, there is a neutral/positive polarity around XRP, since its suggesting XRP reaching a support line and likely to stop its decline, yet there is an obvious negative sentiment for ETH and BTC crypto assets. In sentence 2, the distinction is more obvious where the title indicates a negative sentiment for BTC crypto asset and positive sentiment for DeFi cryptos. For such cases, our sentiment analysis model needs to perform distinctive analysis with respect to different aspects of given sentence. We are planning to implement this aspect-based sentiment analysis in near future.

Sentence 3 is one of the sentences which sits in gray area. Our model outputs negative scores due to the mention of some "hack". Yet this sentence may be perceived positively by some people, due to the positive developments on this "hack" case. In addition, sentence 7 is also a pretty confusing sentence in terms of market direction prediction. Author says the prices will go up or down thus indicating a neutrality, yet Vader and FinBERT assess false positive scores to this sentence.

Trained FinBERT model has distinctive score for Sentence 4 compared to baseline models. Baseline models' dictionaries do not include the highly polar verb "plummet" thus failing to catch the polarity of sentence.

Sentence 5 and 6 are written back to back in a news content. Firstly, the author mentions a past event where crypto markets fall, then negates that with the sentence 6 and indicates the markets are now up. FinBERT and ML models assess scores to both sentences correctly, yet Vader assess a false positive score to sentence 5. This is another problem we faced mostly when accumulating sentence scores to full content scores. For mean sentiment score of these two sentences, ML and FinBERT output neutral scores, only Vader outputs a positive score due to its initial false positive assessment. The problem here is, from investors perspective, past events are irrelevant for market direction predictions. Therefore, we are planning to implement a filtering mechanism that excludes scores of sentences mentions from the past.

CHAPTER 7

CONCLUSIONS

In this thesis, firstly we worked on establishing a crypto asset taxonomy, and as a start, we focused on categorizing crypto assets in terms of sector, asset type and transaction anonymity aspects. As a solution, we propose a two-phase approach such that the first phase predicts document labels for each of these three (3) aspects, and the second phase aggregates the document-level prediction results to generate labels for cryptocurrencies. As another major task of the classifying crypto assets towards taxonomy work, we collected and manually annotated a data set including whitepapers and descriptions related to these three (3) aspects of the cryptocurrencies.

As for crypto asset label prediction, while we achieved better results on transaction anonymity and asset type with proposed Hierarchical model. In sector prediction tasks, base supervised models achieved better results. As a future work, we plan to develop a method for generating non-binary hierarchies for sector prediction task. In theory, this would reduce the error space and may achieve better results on Hierarchical models.

For this version of crypto taxonomy dataset, neural network models did not seem suitable but better results could be achieved by using neural networks that do not rely on large training examples. In addition, over-sampling methods which generates artificial instances like SMOTE could also be used but those methods are still questionable on high dimensions [76]. Also, we believe that applying feature selection techniques can also improve current results.

Lastly, during the construction of the crypto taxonomy dataset, what we struggled most was the extracting information from whitepapers due to their unstructured and

unstandardized nature. Establishing a common template/structure for whitepapers would help categorizing crypto assets.

As the second part of this thesis, we focused on analyzing sentiment analysis of cryptocurrency related news. For this task again, we proposed a two-phase model, first we analyzed news in sentence level and then applied an accumulation technique for mapping these sentence level predictions to news' title, summary and content level sentiment analysis. For this task, we also collected and manually annotated a news sentiment dataset on title, summary and content.

For sentiment analysis task, we were able to achieve promising results by fine-tuning the FinBERT model on our sentiment dataset. As we observed the positive effects of increasing training data; as a future work, we plan to increase our training data by annotating more news. In addition, we also plan to implement an aspect-based sentiment analysis model which would extract sentiment scores with respect to mentioned crypto assets. Lastly, we are also considering implementing a filtering mechanism that excludes sentences that mention past events.

REFERENCES

[1] K. Micallef, “Cryptocurrency and why it’s become so popular,” 2019. https://www.maltablockchainsummit.com/news/cryptocurrency-and-why-its-become-so-popular/.

[2] J. Redman, “What Are Altcoins and Why Are There Over 5,000 of Them?” https://news.bitcoin.com/altcoins-why-over-5000/.

[3] J. Lausen, “Regulating initial coin offerings? A taxonomy of crypto-assets,” in *27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019*, 2020.

[4] T. Ankenbrand, D. Bieri, R. Cortivo, J. Hoehener, and T. Hardjono, “Proposal for a Comprehensive (Crypto) Asset Taxonomy,” in *Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2020, doi: 10.1109/cvcbt50464.2020.00006.

[5] L. Oliveira, I. Bauer, L. Zavolokina, and G. Schwabe, “To token or not to token: Tools for understanding blockchain tokens,” in *International Conference on Information Systems 2018, ICIS 2018*, 2018.

[6] P. Tasca and C. J. Tessone, “A Taxonomy of Blockchain Technologies: Principles of Identification and Classification,” *Ledger*, 2019, doi: 10.5195/ledger.2019.140.

[7] “Sector Indices.” https://capital.com/sector-indices-definition.

[8] “ICO Guidelines,” 2018. [Online]. Available: https://www.finma.ch/en/~/media/finma/dokumente/dokumentencenter/myfinma/1bewilligung/fintech/wegleitung-ico.pdf?la=en.

[9] “Crypto Advice,” 2019. [Online]. Available: https://www.esma.europa.eu/sites/default/files/library/esma50-1571391_crypto_advice.pdf .

[10] “Bitcoin Is Not Anonymous.” https://www.elliptic.co/our-thinking/bitcoin-transactions-money-laundering.

[11] E. Ben-Sasson *et al.*, “Zerocash: Decentralized anonymous payments from bitcoin,” in *Proceedings - IEEE Symposium on Security and Privacy*, 2014, doi: 10.1109/SP.2014.36.

[12] N. Van Saberhagen, “CryptoNote v 2.0,” *Self-published*, 2013.

[13] K. Rappoza, “Can ‘Fake News’ Impact The Stock Market?” https://www.forbes.com/sites/kenrapoza/2017/02/26/can-fake-news-impact-the-stock-market/#4d9d8dc82fac.

[14] L. Alessandretti, A. Elbahrawy, L. M. Aiello, and A. Baronchelli, “Anticipating Cryptocurrency Prices Using Machine Learning,” *Complexity*, 2018, doi: 10.1155/2018/8983590.

[15] S. Tandon, S. Tripathi, P. Saraswat, and C. Dabas, “Bitcoin Price Forecasting using LSTM and 10-Fold Cross validation,” in *2019 International Conference on Signal Processing and Communication, ICSC 2019*, 2019, doi: 10.1109/ICSC45622.2019.8938251.

[16] B. Sakiz and E. Kutlugun, “Bitcoin price forecast via blockchain technology and artificial intelligence algorithms,” in *26th IEEE Signal Processing and Communications Applications Conference, SIU 2018*, 2018, doi: 10.1109/SIU.2018.8404719.

[17] L. Catania, S. Grassi, and F. Ravazzolo, “Forecasting cryptocurrencies under model and parameter instability,” *Int. J. Forecast.*, 2019, doi: 10.1016/j.ijforecast.2018.09.005.

[18] I. Madan, S. Saluja, and A. Zhao, “Automated Bitcoin Trading via Machine Learning Algorithms,” *URL http//cs229. stanford. edu/proj2014/Isaac\% 20Madan*, 2015.

[19] A. G. Shilling, “Market Timing: Better Than a Buy-and-Hold Strategy,”

Financ. Anal. J., 1992, doi: 10.2469/faj.v48.n2.46.

[20] Z. Jiang and J. Liang, "Cryptocurrency portfolio management with deep reinforcement learning," in *2017 Intelligent Systems Conference, IntelliSys 2017*, 2018, doi: 10.1109/IntelliSys.2017.8324237.

[21] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, doi: 10.3115/v1/d14-1181.

[22] X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015.

[23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, 2016, doi: 10.18653/v1/n16-1174.

[24] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, doi: 10.18653/v1/P17-1052.

[25] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2018, doi: 10.18653/v1/p18-1031.

[26] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies -*

Proceedings of the Conference, 2019.

[27] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, 2019, pp. 5753–5763.

[28] G. C. Cerda and J. R. De La Maza, "Bitcoin price prediction through opinion mining," in *The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019*, 2019, doi: 10.1145/3308560.3316454.

[29] S. Colianni, S. Rosales, and M. Signorotti, "Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis," *CS229 Proj.*, 2015.

[30] S. Nasekin and C. Y. Chen, "Deep Learning-Based Cryptocurrency Sentiment Construction," *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3310784.

[31] S. Rouhani and E. Abedin, "Crypto-currencies narrated on tweets: a sentiment analysis approach," *Int. J. Ethics Syst.*, 2019, doi: 10.1108/IJOES-12-2018-0185.

[32] T. Loughran and B. Mcdonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," *J. Finance*, 2011, doi: 10.1111/j.1540-6261.2010.01625.x.

[33] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, "Good debt or bad debt: Detecting semantic orientations in economic texts," *J. Assoc. Inf. Sci. Technol.*, 2014, doi: 10.1002/asi.23062.

[34] M. Maia, A. Freitas, and S. Handschuh, "FinSSLx: A Sentiment Analysis Model for the Financial Domain Using Text Simplification," in *Proceedings - 12th IEEE International Conference on Semantic Computing, ICSC 2018*, 2018, doi: 10.1109/ICSC.2018.00065.

[35] M. G. Sousa, K. Sakiyama, L. d. S. Rodrigues, P. H. Moraes, E. R. Fernandes, and E. T. Matsubara, "BERT for Stock Market Sentiment

Analysis," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, Nov. 2019, pp. 1597–1601, doi: 10.1109/ICTAI.2019.00231.

[36] G. Phillips and İ. Kiçeci, "Blockchain in Finance Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report-Finance.pdf.

[37] G. Phillips and İ. Kiçeci, "Blockchain in Technology Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Technology.pdf.

[38] G. Phillips and İ. Kiçeci, "Blockchain in Entertainment Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Entertainment.pdf.

[39] G. Phillips and İ. Kiçeci, "Blockchain in Content & Advertiesment Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Content and Advertisement.pdf.

[40] G. Phillips and İ. Kiçeci, "Blockchain in Education Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report-Education.pdf.

[41] G. Phillips and İ. Kiçeci, "Blockchain in Healthcare and Pharma Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Health Care and Pharma.pdf.

[42] G. Phillips and İ. Kiçeci, "Blockchain in Energy Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Energy.pdf.

[43] G. Phillips and İ. Kiçeci, "Blockchain in Real Estate Sector." [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Real Estate.pdf.

[44] G. Phillips and İ. Kiçeci, “Blockchain in Charity and Aid Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Charity and Aid.pdf.

[45] G. Phillips and İ. Kiçeci, “Blockchain in Travel Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report-Travel.pdf.

[46] G. Phillips and İ. Kiçeci, “Blockchain in Transportation Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Transportation.pdf.

[47] G. Phillips and İ. Kiçeci, “Blockchain in Food Production Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Food Production.pdf.

[48] G. Phillips and İ. Kiçeci, “Blockchain in Telecommunication Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report-Telecommunication.pdf.

[49] G. Phillips and İ. Kiçeci, “Blockchain in Cannabis Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Cannabis.pdf.

[50] G. Phillips and İ. Kiçeci, “Blockchain in Retail Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report-Retail.pdf.

[51] G. Phillips and İ. Kiçeci, “Blockchain in Insurance Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Insurance.pdf.

[52] G. Phillips and İ. Kiçeci, “Blockchain in Art Sector.” [Online]. Available: https://files.cryptoindexseries.com/cis-files/sector_info/CIS Report- Art.pdf.

[53] “Types of tokens. The four mistakes beginner crypto-investors make.”

https://medium.com/swlh/types-of-tokens-the-four-mistakes-beginner-crypto-investors-make-a76b53be5406.

[54] M. O'Dair, *Distributed creativity: How blockchain technology will transform the creative economy*. Palgrave Macmillan, 2018.

[55] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference*, 2017, doi: 10.18653/v1/e17-2068.

[56] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the National Conference on Artificial Intelligence*, 2015.

[57] J. L. Elman, "Finding structure in time," *Cogn. Sci.*, 1990, doi: 10.1016/0364-0213(90)90002-E.

[58] C. Du and L. Huang, "Text classification research with attention-based recurrent neural networks," *Int. J. Comput. Commun. Control*, 2018, doi: 10.15837/ijccc.2018.1.3142.

[59] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[60] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[61] D. Araci, "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models." 2019.

[62] "Stemming and Lemmatization." https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html.

[63] B. Loper, S. Loper, E. Loper, and E. Klein, *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[64] G. A. Miller, "WordNet: A Lexical Database for English," *Commun. ACM*, 1995, doi: 10.1145/219717.219748.

[65] M. F. Porter, "An algorithm for suffix stripping," *Program*. pp. 130–137, 1980, doi: 10.1108/eb046814.

[66] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," *Proc. first Instr. Conf. Mach. Learn.*, 2003.

[67] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, 2011.

[68] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data," *ACM Trans. Knowl. Discov. Data*, 2009, doi: 10.1145/1497577.1497578.

[69] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Process.*, 1998, doi: 10.1080/01638539809545028.

[70] G. Ke *et al.*, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017.

[71] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, doi: 10.3115/v1/d14-1179.

[72] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *30th International Conference on Machine Learning, ICML 2013*, 2013.

[73] C. J. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for

sentiment analysis of social media text," in *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, 2014.

[74] S. Wen *et al.*, "Memristive LSTM Network for Sentiment Analysis," *IEEE Trans. Syst. Man, Cybern. Syst.*, 2019, doi: 10.1109/tsmc.2019.2906098.

[75] C. Sun, L. Huang, and X. Qiu, "Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019.

[76] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics*, 2013, doi: 10.1186/1471-2105-14-106.

APPENDICES

A. Sector Experiments Training and Validation Scores

Figure A.1 Experiment 1 Training and Validation Results

Figure A.2 Experiment 2 Training and Validation Results

Figure A.3 Experiment 3 Training and Validation Results

Figure A.4 Experiment 4 Training and Validation Results

Figure A.5 Experiment 5 Training and Validation Results

Figure A.6 Experiment 6 Training and Validation Results

B. Asset Type Experiments Training and Validation Scores

Figure B.1 Experiment 1 Training and Validation Results

Figure B.2 Experiment 2 Training and Validation Results

Figure B.3 Experiment 3 Training and Validation Results

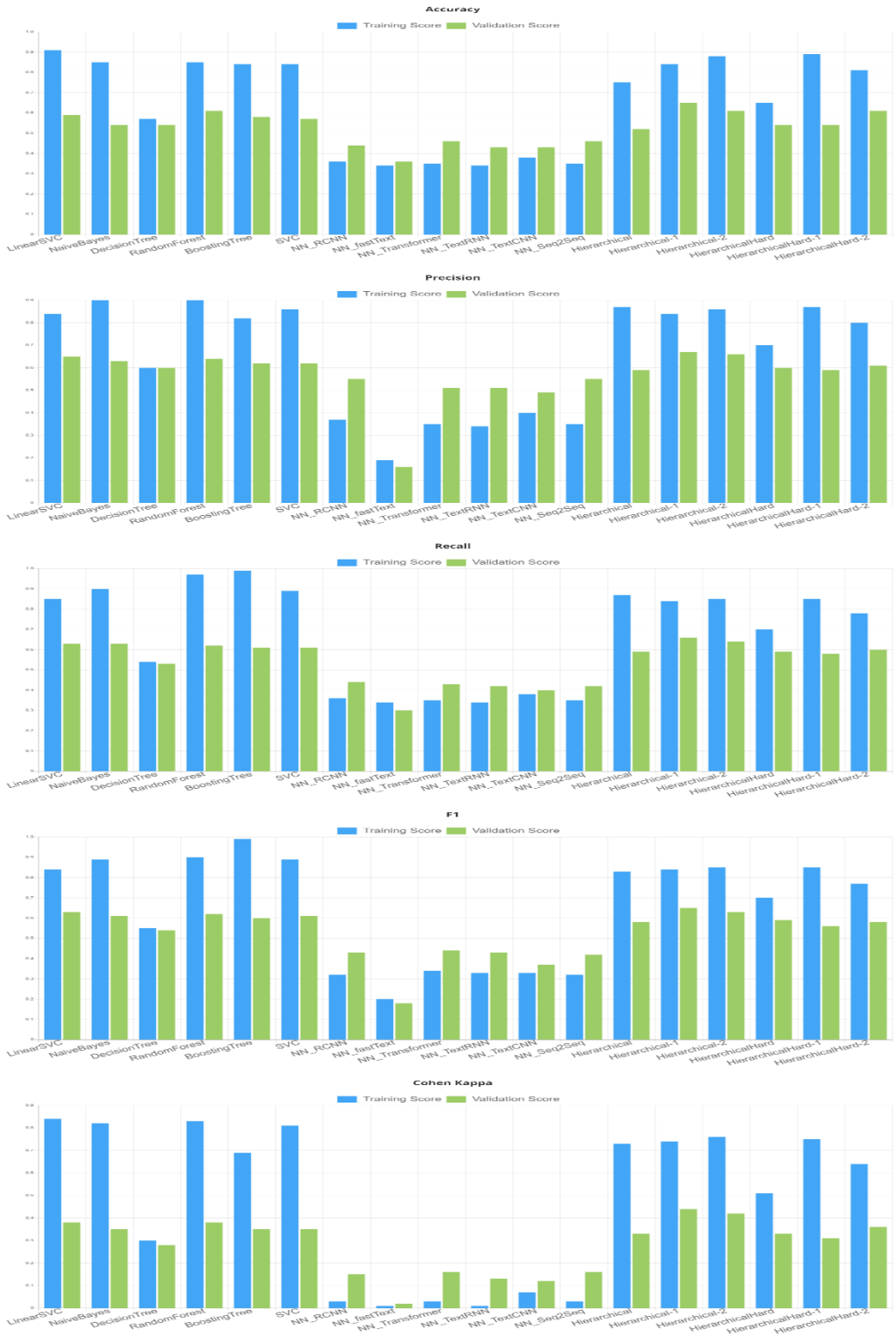


Figure B.4 Experiment 4 Training and Validation Results

Figure B.5 Experiment 5 Training and Validation Results

Figure B.6 Experiment 6 Training and Validation Results

C. Transaction Anonymity Training and Validation Results

Figure C.1 Experiment 1 Training and Validation Results

Figure C.2 Experiment 2 Training and Validation Results

Figure C.3 Experiment 3 Training and Validation Results

Figure C.4 Experiment 4 Training and Validation Results

Figure C.5 Experiment 5 Training and Validation Results

Figure C.6 Experiment 6 Training and Validation Results

D. Sector Class Hierarchies

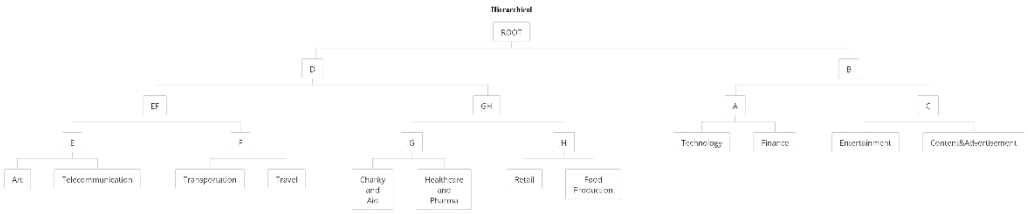


Figure D.1 All Experiments Intuition Based

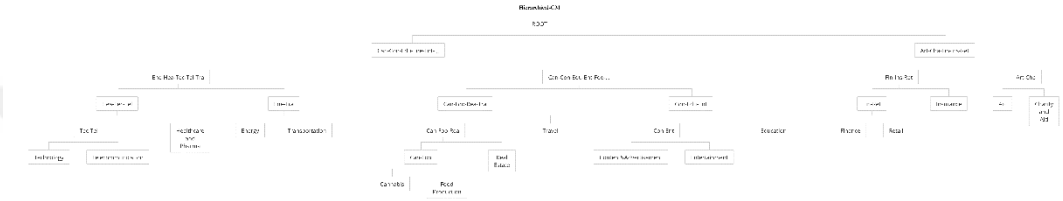


Figure D.2 Experiment 1 Confusion Matrix based Hierarchy

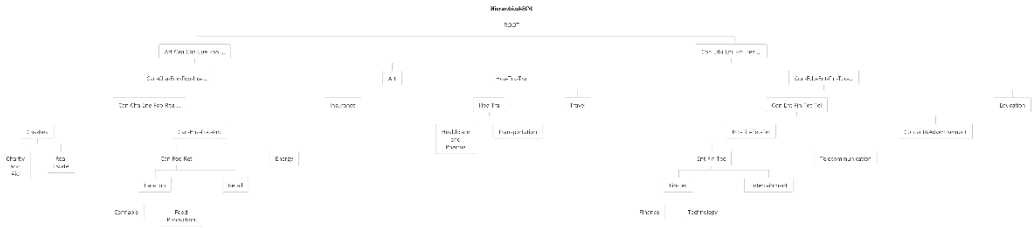


Figure D.3 Experiment 1 Similarity based Hierarchy

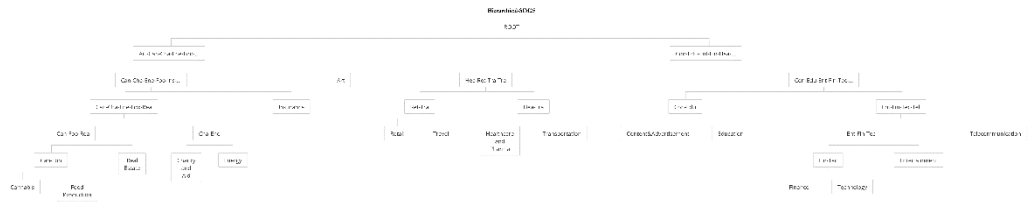


Figure D.4 Experiment 1 Similarity with 25 High TF-IDF Scored Features based Hierarchy

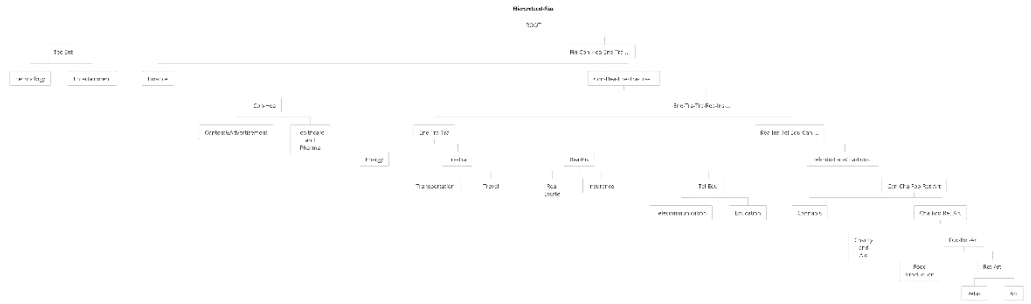


Figure D.5 Experiment 1 Class Size Distribution Based Hierarchy

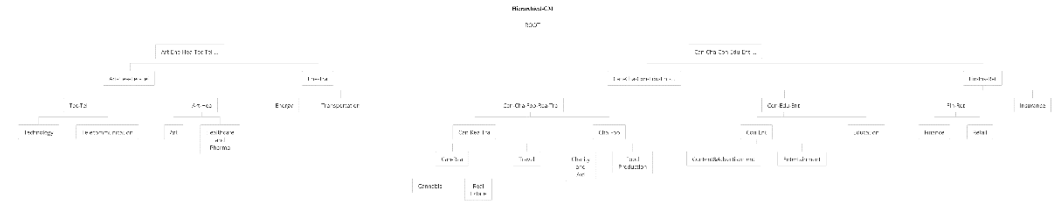


Figure D.6 Experiment 2 Confusion Matrix based Hierarchy

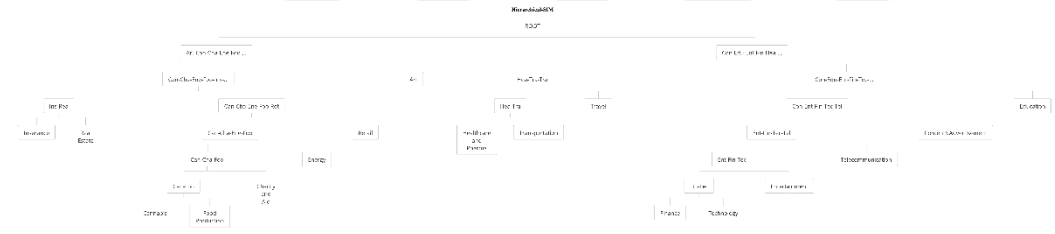


Figure D.7 Experiment 2 Similarity based Hierarchy

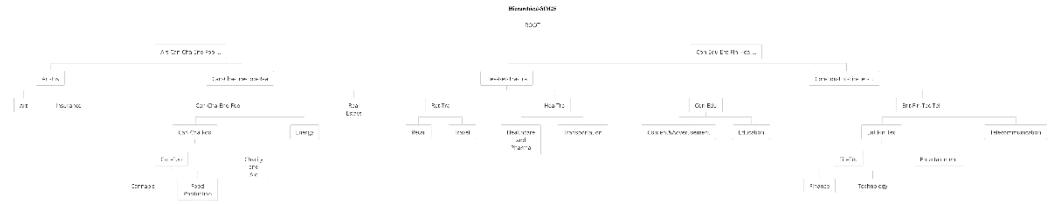


Figure D.8 Experiment 2 Similarity with 25 High TF-IDF Scored Features based Hierarchy

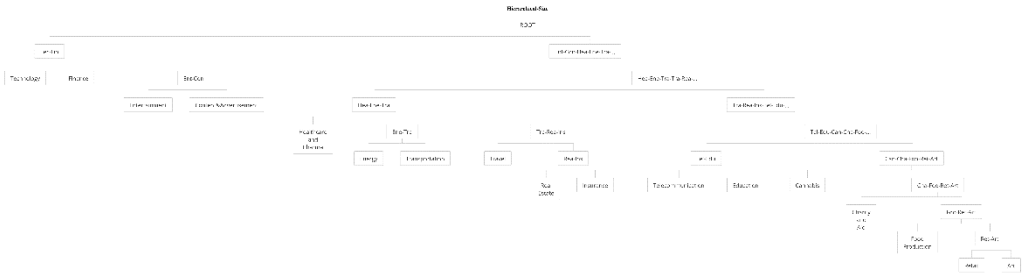


Figure D.9 Experiment 2 Class Size Distribution Based Hierarchy

Figure D.10 Experiment 3 Confusion Matrix based Hierarchy

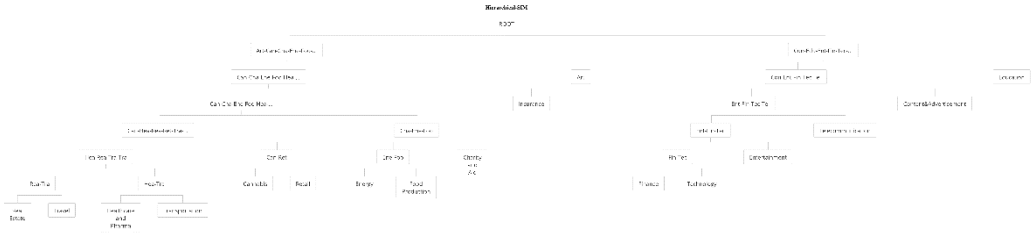


Figure D.11 Experiment 3 Similarity based Hierarchy

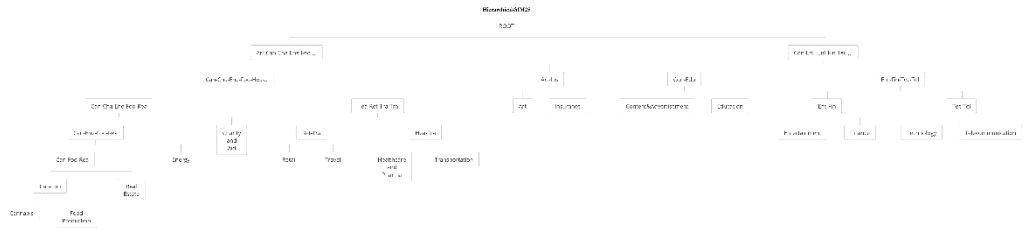


Figure D.12 Experiment 3 Similarity with 25 High TF-IDF Scored Features based Hierarchy

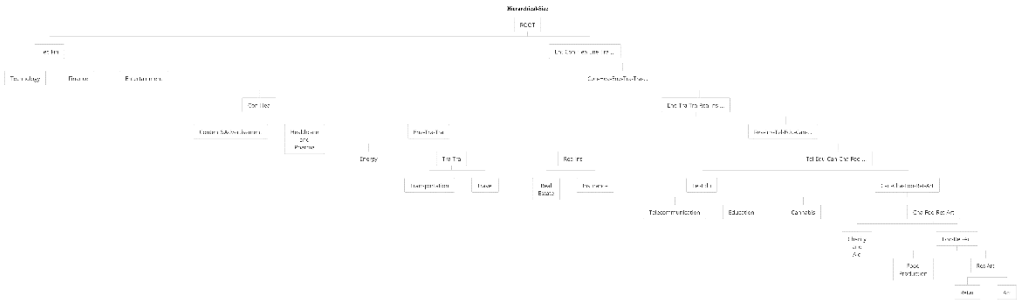


Figure D.13 Experiment 3 Class Size Distribution Based Hierarchy

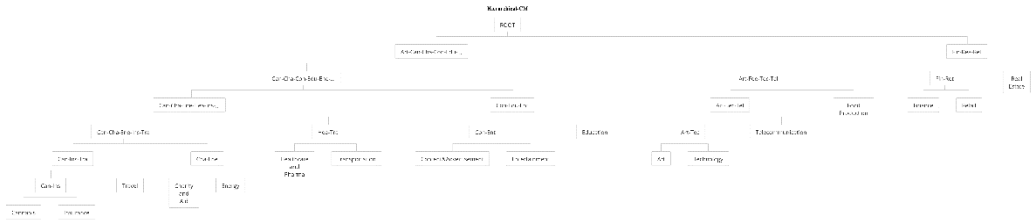


Figure D.14 Experiment 4 Confusion Matrix based Hierarchy

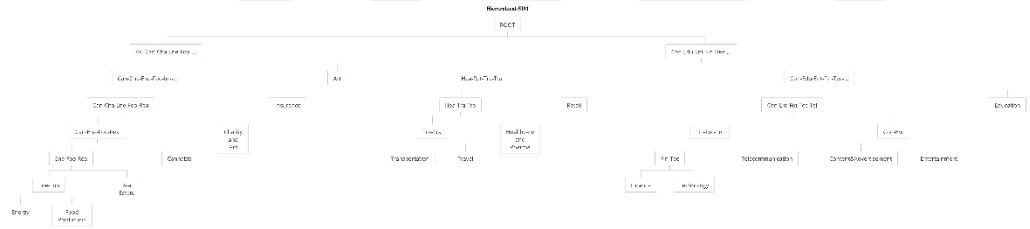


Figure D.15 Experiment 4 Similarity based Hierarchy

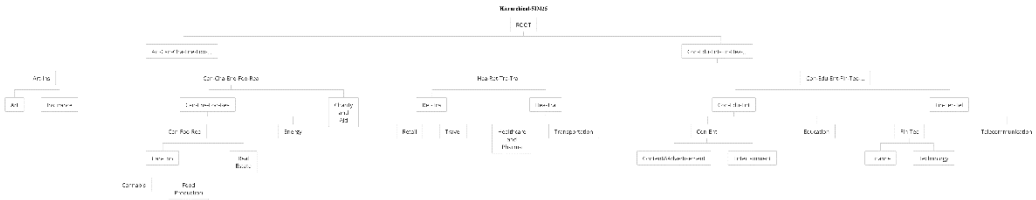


Figure D.16 Experiment 4 Similarity with 25 High TF-IDF Scored Features based Hierarchy

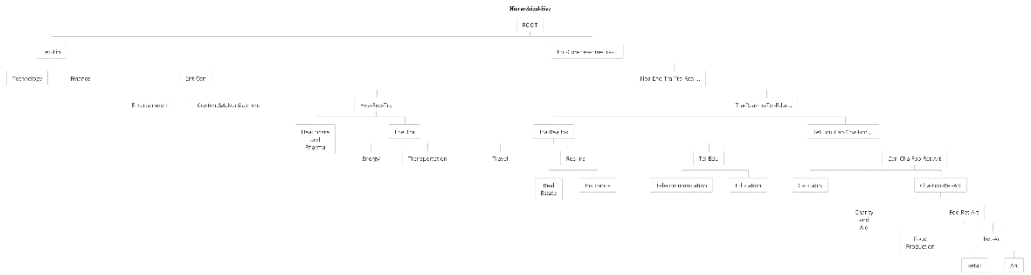


Figure D.17 Experiment 4 Class Size Distribution Based Hierarchy

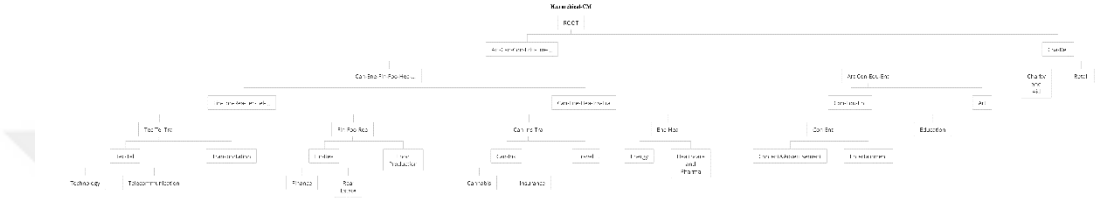


Figure D.18 Experiment 5 Confusion Matrix based Hierarchy

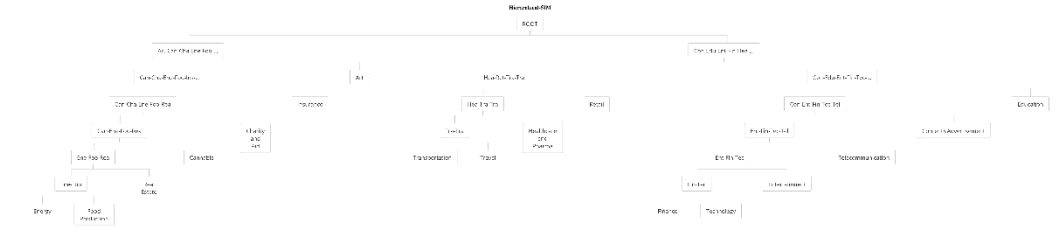


Figure D.19 Experiment 5 Similarity based Hierarchy

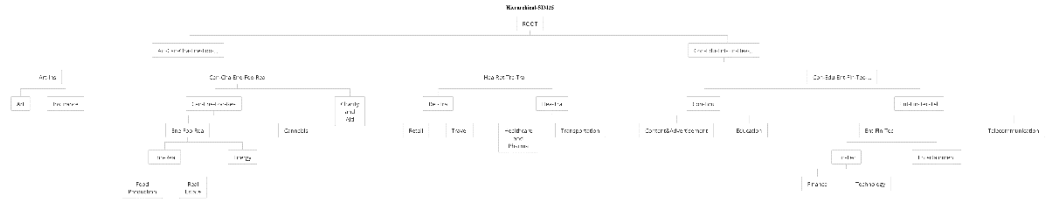


Figure D.20 Experiment 5 Similarity with 25 High TF-IDF Scored Features based Hierarchy

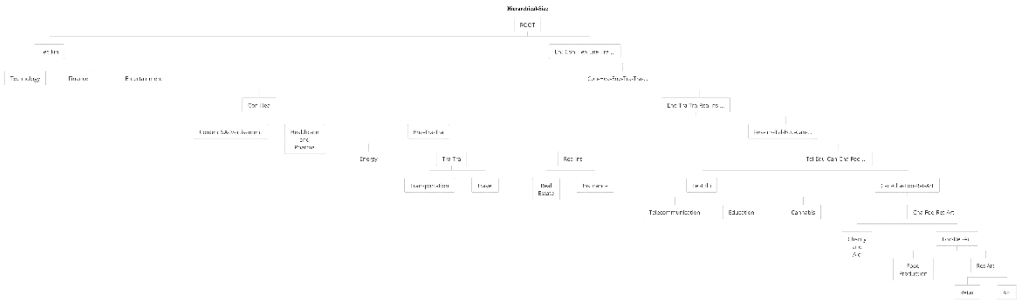


Figure D.21 Experiment 5 Class Size Distribution Based Hierarchy

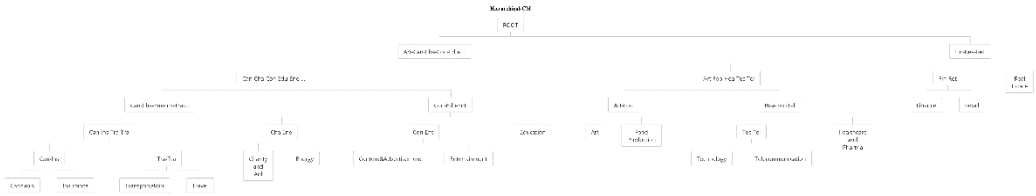


Figure D.22 Experiment 6 Confusion Matrix based Hierarchy

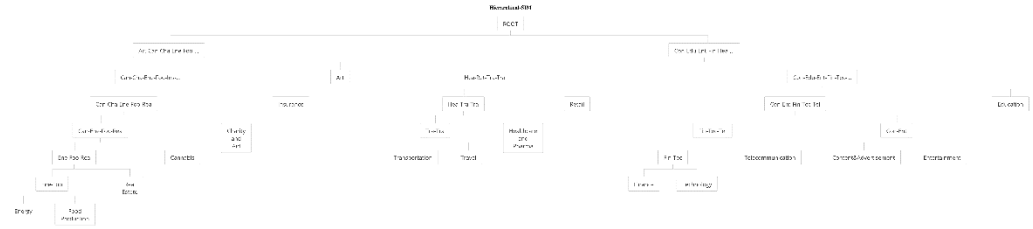


Figure D.23 Experiment 6 Similarity based Hierarchy

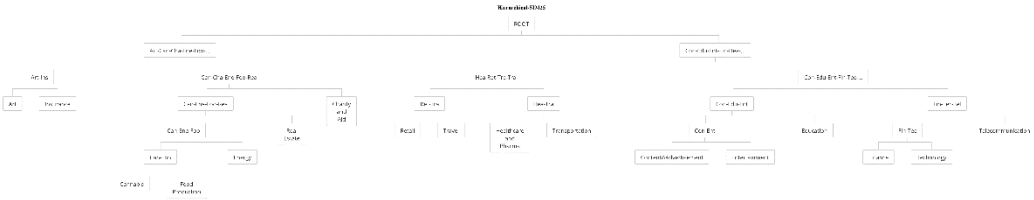


Figure D.24 Experiment 6 Similarity with 25 High TF-IDF Scored Features based Hierarchy

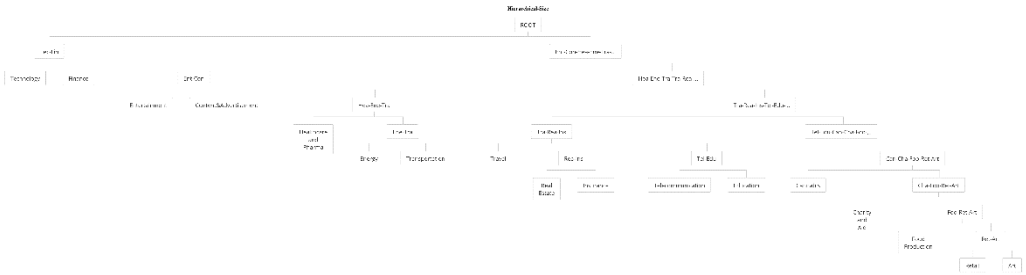


Figure D.25 Experiment 6 Class Size Distribution Based Hierarchy

E. Asset Type Class Hierarchies

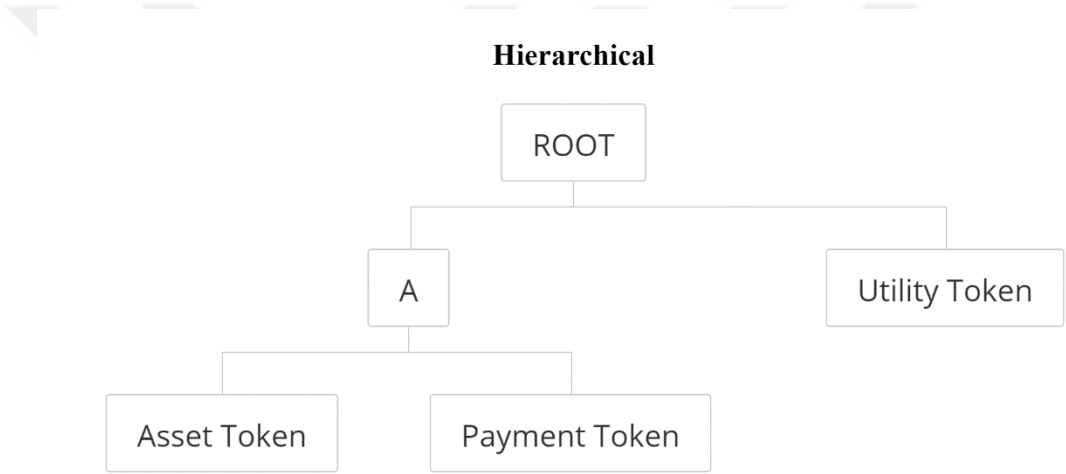


Figure E.1 All Experiments Hierarchy

Hierarchical-1

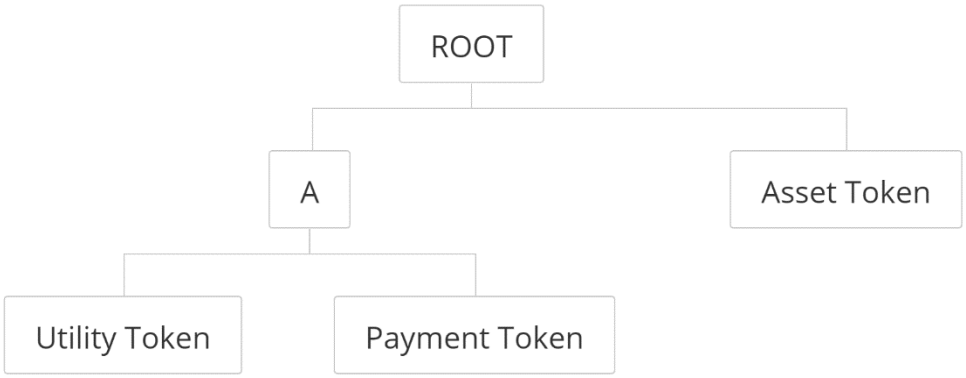


Figure E.2 All Experiments Hierarchical-1

Hierarchical-2

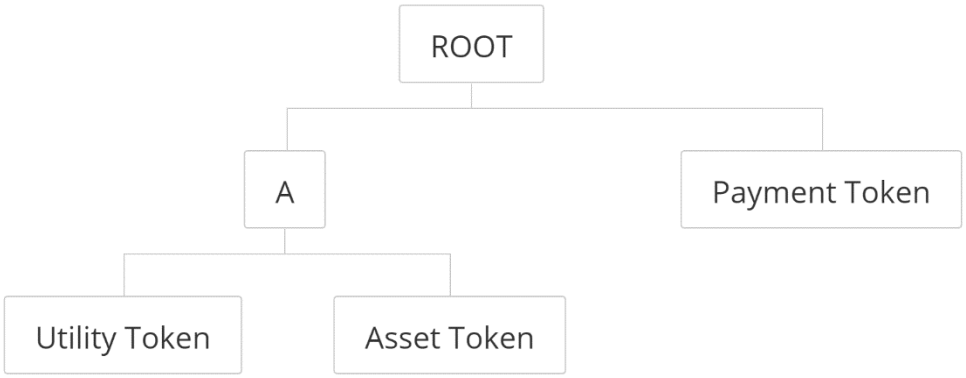


Figure E.3 All Experiments Hierarchical-2

F. Transaction Anonymity Class Hierarchies

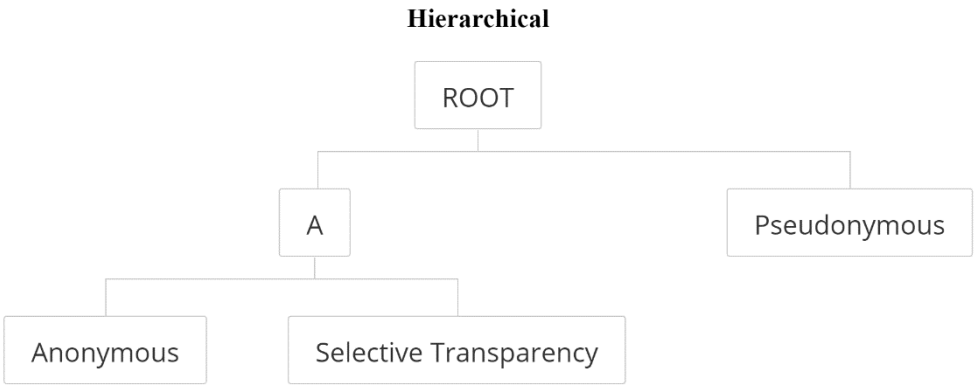


Figure F.1 All Experiments Hierarchical

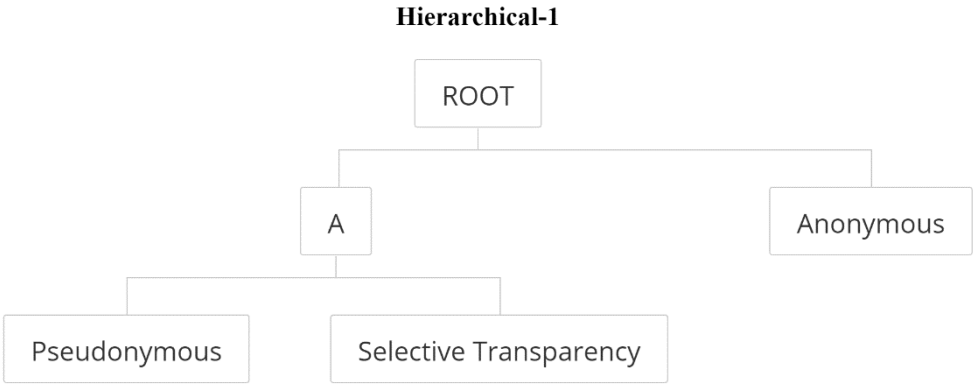


Figure F.2 All Experiments Hierarchical-1

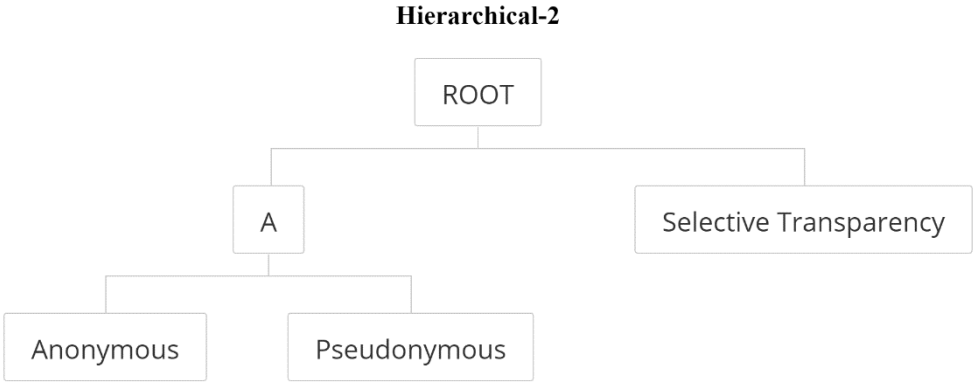


Figure F.3 All Experiments Hierarchical-2

G. News Sentiment Training and Validation

Figure G.1 Training and Validation Scores of Experiment 1

Figure G.2 Training and Validation Scores of Experiment 2

Figure G.3 Training and Validation Scores of Experiment 3

Figure G.4 Training and Validation Scores of Experiment 4