POLITECNICO DI TORINO

Master degree in
Data Science and Engineering

Master Degree Thesis

Financial Time Series Summarization

Supervisors:
Prof. Luca Cagliero
Ing. Jacopo Fior

Candidate
Tommaso Calo'

October 2021

ABSTRACT

The use of Machine Learning techniques for financial data analysis has become increasingly popular. The problem of explaining financial data with the help of Machine Learning and Natural Language Processing techniques has become crucial for companies that need to support business decisions and to understand machine learning results. This thesis proposes a method for summarizing financial time series based on textual protoforms, which consist of a set of informative summaries used to explain the performance of a certain stock over time. The method aims at synthesizing the key information conveyed by both historical prices and the economic stock indicators. The analyzed financial time series are represented into a unified vector space by using a popular embedding algorithm to leverage time series similarities in the generation of the output summaries. The resulting summaries can explain the performance of different stocks in different time periods, compare multiple stocks with respect to a performance measure or compare the performances of different sectors. In addition, several indicators are generated about the quality and the informativeness of the summaries. The produced indicators allow the comparison between different summaries and gives the final user information about the confidence of the provided statements.

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

One of the biggest obstacles in the adoption of artificial intelligent tools by the business is that the majority of the most performant machine learning models works as a "black box", where even its designers cannot explain why an AI arrived at a specific decision[22]. If humans are to accept algorithmic prescriptions, they need to trust them, and trust can only be achieved through explainability and interpretability. [15] Our goal is to improve the explainability of temporal financial data through the usage of a set of informative summary templates, or "protoforms" [18]. This approach allows traders to gain more insights into the stock behaviour and help them to adopt the best investment strategy. Within the field of summarization, there are three main approaches when it comes to data-to-text summary generation: probabilistic/statistical, neural, and rule-based methods [12]. Statistical methods uses predictive algorithms to convert data into text, neural methods exploits the usage of deep learning methods. Rule-based approaches are typically the most robust ones and produce higher quality summaries. The text quality of trainable systems — e.g. statistical and neural models — is generally lower and their development slower [35, 28]. However, trainable systems use data-driven algorithms and do not rely on manually written resources for text generation, while most template systems require manually written templates and rules for text generation. This makes trainable systems potentially more adaptable and maintainable. In our approach we propose a rule-based method that works as follows: starting from the temporal series of a certain stock we discretize it based on the relevant events occurring on each date. The events are encoded in an alphabet and can be classified as: events directly related to the behaviour of the temporal series — e.g. "Simple moving average of 5 periods crossed above simple moving average of 20 periods" —, and events not directly related to the behaviour of the time series, — e.g sentiment obtained from a database of news related to the stock —. The discretized time series is modelled as a *document* (series of events), composed by *words* (events). We generate a document for different periods of time (week, months, trimester and years) in order to compare the behaviour of the temporal series over different time periods. We then apply an established document embedding algorithm [29] to obtain vector representation of such documents, thus we obtain a vector representation of the time series over different periods of time. We then infer from the aforesaid vector representation the behaviour of certain stock with respect to either the other stocks or some reference stocks, which are selected using external indexes of performance derived from fundamental analyses (R&D investments, EBITDA etc.). With this method we produce five different types of summaries that extend the expressiveness of previous methods, e.g., [19] To demonstrate the usefulness and applicability of the proposed approach we apply the same evaluation metrics presented as in [19] to deliver with every summary a quantitative evaluation of its quality and informativeness. They provide end-users with quantitative evaluators of the generated protoforms, which can be used to drive the summary exploration and knowledge discovery processes.

Chapter 2

Related Works

2.1 Data-To-Text Generation

Data-to-text generation problem is generally approached as a machine translation problem, a very large field of study which includes all the language translation algorithms, the main method developed to tackle such problem are statistical machine translation [27], neural machine translation [26] and rule-based linguistic summarization [9]. Statistical machine translation uses predictive algorithms to translate text. These models are created, or learned, from parallel bilingual text corpora and used to create the most probable output, based on different bilingual examples. Using this already translated text, a statistical model guesses or predicts how to translate foreign language text. SMT has different subgroups, including word-based, phrase-based, syntax-based and hierarchical phrase-based. The benefit of SMT is its automation. One drawback is that this system needs bilingual material to work from, and it can be hard to find content for obscure languages. SMT is a "rule-based" MT method, using the basis of corpora translations to create its own text segments. Neural machine translation (NMT) is an approach to machine translation that uses an artificial neural network to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model. They require only a fraction of the memory needed by traditional statistical machine translation (SMT) models. Furthermore, unlike conventional translation systems, all parts of the neural translation model are trained jointly (end-to-end) to maximize the translation performance. [25] [38] [11] Neural and statistical methods relies on measures like the BLEU score [34] to evaluate these summaries. BLEU (bilingual evaluation understudy) is an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU. BLEU was one of the first metrics to claim a high correlation with human judgements of quality, and remains one of the most popular automated and inexpensive metrics. [10]
Rule-based methods typically obviates the need for measures like the BLEU score using semantically meaningful templates or protoforms to generate their output which can rely on human evaluation to judge their utility, but they also can use objective measures such as significance, frequency, and other metrics [9] to judge the quality of the summary output. A comparison made by van der Lee et al. [39] shows the difference in performance and

text quality between rule-based, neural, and statistical methods. They concluded that rule-based methods generally produce higher text quality, but the the creation of the protoforms is time intensive since every protoform must be created manually. They also observe that rule-based methods are generally restricted to more simple situations and may be less useful in more complex cases. Statistical methods avoid the manual creation of protoforms, but are generally poorer in text quality. Furthermore statistical text generation methods typically need a large set of training pairs containing the input data and the target natural language summary, which is often not available. In our project we will follow the rule-based linguistic paradigm to maximize the text quality and to create more complex summaries.

2.2 Time series Data Mining

Following a review by Batyrshin and Sheremetov [8], we depicted a broad range of works on time series data mining: the construction of rules based on patterns found in the data [14], transformation of time series into state intervals to create association rules [21], generating reports about stocks [36], and so on. Our work utilizes an approach symilar to the Symbolic Aggregate approXimation (SAX) [30] to discretize time series data into a symbolic sequence from which we can mine patterns and trends. In facts instead of discretizing the series based on its absolute value such as [30], we discretize it exploiting relevant trigger events in the context of stock fluctuation, such as moving average convergence/divergence, on balance volume and others, at every of this event corrisponds a symbol, the discretization of the time series corrisponds to a sequence of this symbols.

2.3 Time Series Summarization

Our algorithm is inspired by summarization methods [40] [23] [41] [19] that rely on the concept of protoforms and fuzzy logic [9] to summarize data. Linguistic summarization methods have also been applied to time series data in various domains, such as elderly care [42], physical activity tracking [37], driving simulation environments [16], deforestation analysis [13], human gait study [7], periodicity detection [32], time series forecasting [24], generation of longer temporal summaries from neonatal intensive care [17], generation of summaries about nutrition and health behaviours [19], within the financial domain [33] uses a neural network to generate natural language summaries. As we already depicted, neural network based methods suffer from lack of high quality text, dependence on large database of data and difficulty to explain the time series patterns directly from the raw temporal data [39]. In contrast our work uses a neural networks to generete an embedding of the time series of the stock [29], but the final summary is generated with a rule-based method using protoforms, with this method we exploit both the feature extraction power of neural networks and the advantage of using a rule based approach. The closest work to ours is [19], where they use data-mining techniques to extract relevant information from the time series, and rule generation to generate the summaries from existing protoforms. Their work such as our is completely unsupervised, they use a protoform composed of an attribute, a time window, a quantifier and a summarizer. An example summary of an event from their work is: "On most of the days in the past week, your calorie intake was high", The quantifier "On most" is calculated resulting from fuzzy-membership functions that pairs the attribute and the summarizer over the time window of interest, the quantifier though represents how often the finding is found to be true in the data. the attribute "calorie intake" represents the variable of interest, and the summarizer "high" represents the conclusion from the data. In our work, we use such technique to generate the summaries, the main difference with the previous work [19] are:

- Time series discretization since we use a general document embedding algorithm [29] which encodes more information about the time series.
- Comparison Summaries generated comparing the time series embedding with a reference profile, in order to produce useful information about the performance of a certain stock over a time window.
- We leverage time series embedding models to compare the price series of different stocks and quantify their correlation level.

Chapter 3

Methods

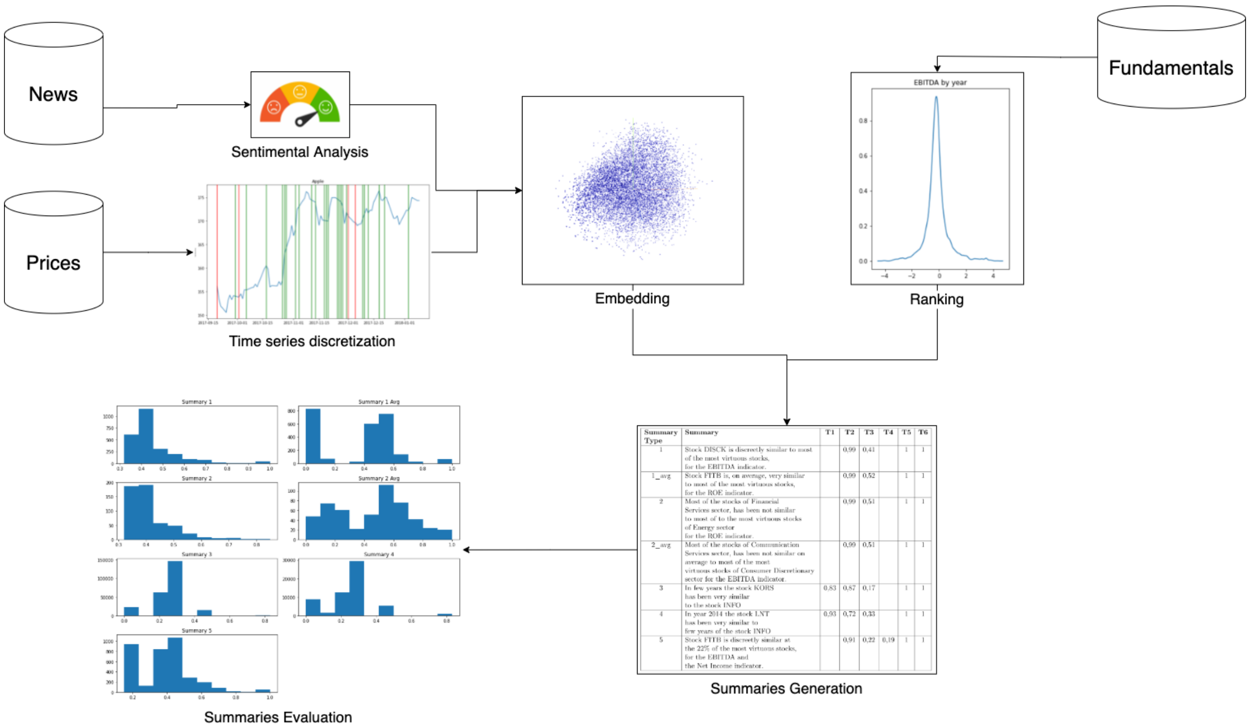

Figure 3.1: Overview of the method used to generate summaries

Figure 3.1 presents our method to generate the summaries which work as follows: We discretize the price temporal series of a certain stock based on the relevant events occurring on each date. The events are directly related to the behaviour of the temporal series — e.g. "Simple moving average of 5 periods crossed above simple moving average of 20 periods" —, in addition to these type of events, we added a sentiment obtained from a database of news related to the stock in a certain day. The discretized time series is modelled as a *document* (series of events), composed by *words* (events). We apply an established document embedding algorithm [29] to obtain vector representation of such documents, thus we obtain a vector representation of the time series. We then infer from the aforesaid vector representation the behaviour of certain stock with respect to either the other stocks or some reference stocks, which are selected using external indexes of performance derived from fundamental analyses (R&D investments, EBITDA etc.). With this method we produce five different types of summaries. To demonstrate the usefulness and applicability of the proposed approach we apply the same evaluation metrics presented as in [19] to deliver with every summary a quantitative evaluation of its quality and informativeness. They provide end-users with quantitative evaluators of the generated protoforms, which can be used to drive the summary exploration and knowledge discovery processes.

3.1 Time Series Discretization

To generate our document describing the behaviour of a certain stock over time we exploit two characteristics: The events are encoded in an alphabet and can be classified as: events directly related to the behaviour of the temporal series — e.g. "Simple moving average of 5 periods crossed above simple moving average of 20 periods" —, and events not directly related to the behaviour of the time series, — e.g sentiment obtained from a database of news related to the stock —. The discretized time series is modelled as a *document* (series of events), composed by *words* (events). We generate a document for different periods of time (week, months, trimester and years) in order to compare the behaviour of the temporal series over different time periods.

- Events directly related to the behaviour of the temporal series: moving average convergence divergence, relative strength index, aroon oscillator, percentage volume oscillator, accumulation distribution indicator.
- Events not directly related to the behaviour of the time series: sentiment obtained from a database of news related to the stock.

Following a detailed explainations of the indicators and how they are implemented:

3.1.1 Moving average convergence divergence

Moving average convergence divergence (MACD) is an indicator that shows the relationship between two moving averages of a the price time series. The MACD is calculated by subtracting the a-period exponential moving average (EMA) from the b-period EMA where b>a. In our case, we used 5,20; 20,50; 50,200 as periods, furthermore, in addition to exponential moving average we calculated MACD with simple moving average. When happens to be a crossing between these two moving average, an event is triggered and added to the document describing the time series, this indicator in then discribed by 12 words, 3 for each a-b pair, times 2 for each kind of moving average (EMA and SMA) times 2 for each kind of event (crossing above or crossing below).

3.1.2 Relative strength index

The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. The formula to calculate it is the following:

$$RSI = 100 - \left[\frac{100}{1 + \frac{\text{Average gain}}{\text{Average loss}}} \right] \tag{3.1}$$

We trigger two distinct events: when RSI goes above 70 and below 30 percent.

3.1.3 Aroon oscillator

The Aroon oscillator indicates the beginning of a new trend, its strength and can help anticipate changes from trading ranges to trends. The formula to calculate it is the following:

$$\begin{aligned} \text{Aroon Oscillator } &= \text{ Aroon Up - Aroon Down} \\ \text{Aroon Up } &= 100 * \frac{(25 - \text{ Periods Since 25-Period High })}{25} \\ \text{Aroon Down } &= 100 * \frac{(25 - \text{ Periods Since 25-Period Low })}{25} \end{aligned} \tag{3.2}$$

the Aroon Oscillator can generate trade signals or provide insight into the current trend direction of an asset. We trigger two distinct events: when RSI goes above 70 and below 30 percent.

3.1.4 Percentage volume oscillator

The percent volume oscillator (PVO) instead of analyzing the performances of the prices, analyzes the volumes, in particular it measures the change in volume through two moving averages. For example a positive value shows that the recent trend in volume's rate of change is positive, this may imply that the price could start moving in the direction of the prevailing trend, and vice versa PVO is calculated by exploiting moving averages such as in MACD with b = 26 and a = 12, the moving average is of the type of an exponential moving average (EMA). Two distinct events are triggered either when volumes exponential moving average of 12 periods crossed above volumes exponential moving average of 26 periods or vice versa.

3.1.5 Accumulation/Distribution Indicator

The accumulation/distribution indicator is a cumulative indicator that uses volume and price to assess whether a stock is being accumulated or distributed. The A/D measure identifies divergences between the stock price and the volume flow. This provides insight into how strong a trend is. in figure 3.2 is shown how the accumulation/distribution indicator is implemented. If the price is rising but the indicator is falling, then it suggests that buying or accumulation volume may not be enough to support the price rise and a price decline could be forthcoming.

A/D = Previous A/D + Current Period's Money Flow Volume

Money Flow Volume = Money Flow Multiplier x Volume for the Period

$$\text{Money Flow Multiplier} = \frac{[(\text{Close} - \text{Low}) - (\text{High} - \text{Close})]}{(\text{High} - \text{Low})}$$

Close = Closing Price
Low = Low Price
High = High Price

Figure 3.2: Accumulation/Distribution Indicator formula.

3.1.6 Sentiment analysis

In addition to the indicator of performances directly related to the behaviour of the stock, we added the sentiment analysis of a collection of news related about the stock, the sentiment analys has been performed exploiting Valence Aware Dictionary for Sentiment Reasoning (VADER) technique. VADER [20] sentiment analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text. Two distinct events are then generated, one for positive and one for a negative news.

3.2 Embedding Algorithm

The discretized times series is in the form:
"2,2,2,2,2,2,2,Z2,U2,2,2,2,2,2,2,2,2,2,2,2,2,BP2..." where every word is denoted by a symbol that encode the triggered events on the original stock time series on a given day. There is one such documents for each year of each stock, but we can generate a documents for different time periods such as trimester or months just by simply modify the aggregation method, however to generate the embeddings we will use the year granularity. The embedding are generated by means of the doc2vec [29] embedding algorithm: Doc2vec algorithm is based on word2vec embedding algorithm [31]. As well as word2vec, doc2vec uses an unsupervised learning approach to learn the document embedding representation. The input of words per document can be various while the output is a fixed-length vector. Paragraph vector and word vectors are initialized. Paragraph vector is unique among all document while word vectors are shared among all document such that word vector can be learnt from different document.

There are 2 distinct models for learning document embedding vectors.

3.2.1 Distributed Memory Model of Paragraph Vectors (PV-DM)

The document vectors and word vectors are initialized randomly. Every document vector is assigned to single document while word vectors are shared among all documents. The model attempts to guess the next word from its precedents words with the addition of a paragraph ID.

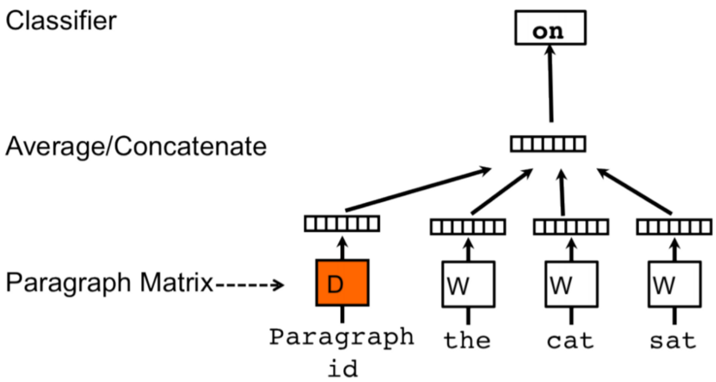

Figure 3.3: Illustration of the PV-DM model.

This approach is similar to continuous bag-of-words (CBOW) approach in word2vec.

3.2.2 Distributed Bag of Words version of Paragraph Vector (PV-DBOW)

Another approach goes a different way. Instead of predicting next word, it use a paragraph vector to classify entire words in the document. During training, sampling a list of word and then form a classifer to classify whether word belongs to the document such that word vectors can be learnt.

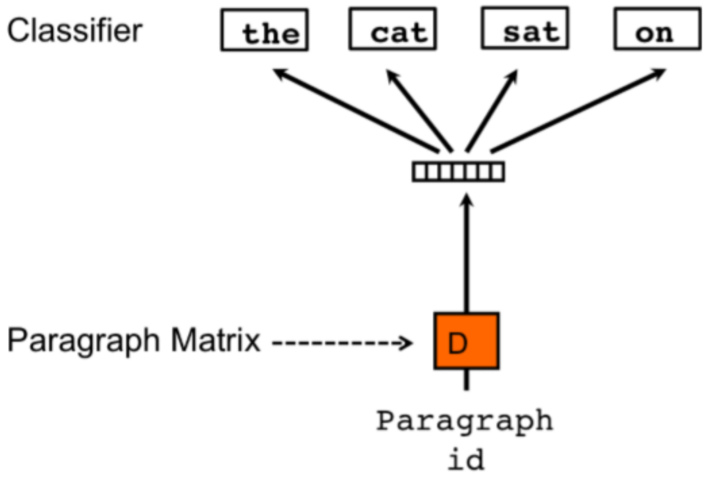

Figure 3.4: Illustration of the PV-DBOW model.

This approach is similar to skip-gram approach in word2vec.

There is only one difference between skip-gram and distributed bag of words (DBOW) is instead of using the target word as the input, Distributed Bag of Words (DBOW) takes the document ID (Paragraph ID) as the input and tries to predict randomly sampled words from the document.

In our case, as already stated, the words are the event triggered by the discretization algorithm of the time series and the news sentiment analysis, and the document is a time frame of a given stock (for example one year).

3.3 Ranking

In parallel with the discretization of the time series a ranking algorithm is applied on the Fundamental database of the stocks which contains balance data of them, in order to obtain reference profiles (very virtuous stocks) for certain indicators. The indicators which we will consider in the Fundamental database are ROE, ROA, EBITDA, R&D investments and Net income. Let's see in an exaple how these most performant stocks are selected: For a certain stock we can calculate the EBITDA for each year under analysis

	GPC
2018	1419172.0
2017	1218445.0
2016	1242911.0
2015	1287018.0
2014	1291140.0

Table 3.1: Performances of a given indicator for GPC stock through years.

Each entry of the database is normalized subtracting the column mean and dividing by the column standard deviation resulting in:

	GPC
2018	1.523912
2017	0.549119
2016	0.667933
2015	0.882131
2014	0.902149

Table 3.2: Normalized performances of a given indicator for GPC stock through years.

From the resulting dataset is derived the difference in performance between two consecutive years, this allow to compare the stocks not in their absolute perfomance but in the relative growth, in doing so, a very performant stock for a certain year is the one which has grown more from the precedent.

	GPC
2018	NaN
2017	-0.974793
2016	0.118815
2015	0.214197
2014	0.020018

Table 3.3: Difference in performances of a given indicator for GPC stock with the previous year.

Given the differences with the precedent year for all the year of each stock, it is evaluated the overall distribution of each entry for each stock for every considered year. the oveall distribution is approximately gaussian as depicted in figure 3.5

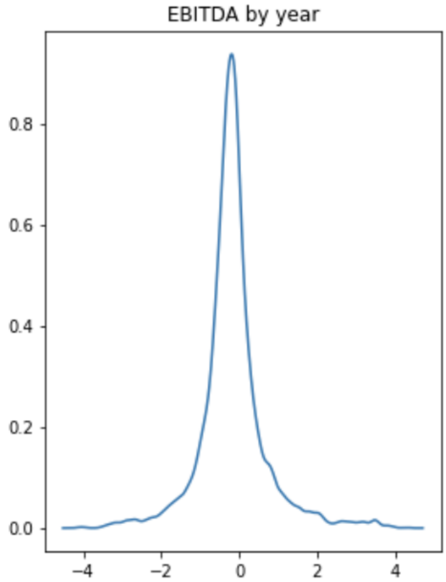

Figure 3.5: Distribution of differences with the precedent year for all the years of all the stocks

We can then classify as virtuous stock for the EBITDA indicator the stocks over the third quartile.

This method is equally applied over all the other indicators: ROE,ROA,R&D and Net income.

3.4 Summaries Generation

3.4.1 Summary 1

Summary 1 is on the form of: "Stock [*stock*] is [*summarizer*] to most of the most virtuous stocks, for [*indicator*]" where summarizer can be one of *very similar, discreetely similar or not similar* and the indicator can be one of *EBITDA, ROE, ROA, R&D investements or Net income.*

Summary 1 for a certain stock is generated as follows: A table is created having at the row the stock for which the summary is being generate and for columns the most virtuous stocks.

	Apple	Facebook	Tesla	Microsoft	Amazon
Google	0.43	0.11	0.37	0.45	0.02

Table 3.4: Table to generate summary 1

Given the overall distribution of similarities we calculate two quantiles named Q_1 and Q_2. We propose two methods to generate the summary:

1. We calculate the fraction of the most virtuous stocks which have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2 with respect to our reference stock. The maximum fraction determines which summarizer to choose, in other words if we have that the major number of the most virtuous stock have a similarity less than Q_1 with respect to our reference stock we choose *not similar*, if we have that the major number of the most virtuous stock have a similarity between Q_1 and Q_2 with respect to our reference stock we choose *dicreetely similar* and if we have that the major number of the most virtuous stock have a similarity higher than Q_2 with respect to our reference stock we choose *very similar*.

2. We calculate the average of the similarities, if the average is less than Q_1 we choose *not similar* as summarizer, if the average is between Q_1 and Q_2 we choose *dicreetely similar* as summarizer and if the average is higher than Q_2 we choose *very similar* as summarizer.

The difference between the two summaries is simple, the second takes more into consideration the overall distribution of similarities with respect to the stock under consideration. Let's focus on an example for which we have $Q_1 = 0.20$ and $Q_2 = 0.30$, if we apply the first method we have $0,43 > Q_2$, $0,11 < Q_1$, $0,37 > Q_2$, $0,45 > Q_2$ and $0,02 < Q_1$, the fraction of similarities less than Q_1 is $\frac{2}{5}$, between Q_1 and Q_2 is 0 and higher than Q_2 is $\frac{3}{5}$ thus we will choose *very similar* as summarizer. If instead we use the second method, the average of similarity equals 0.27 which is between Q_1 and Q_2 and so we would choose *dicreetely similar* as summarizer. The difference lies in the fact that the second method is dependent on the distribution of the all the similarities in the row, while the first is not, but takes into consideration just the most higher counts of similarities between different quantiles, no matter which values the other similarity have.

With these two methods, we generate two types of summaries named **Summary 1** with the first method and **Summary 1_avg** for the second.

3.4.2 Summary 2

Summary 2 is generated based on the methods of summary 1 with the only difference that the first is calculated on sectors instead of on singular stocks. Summary 2 have the form of: "Sector [*sector*] is [*summarizer*] to most of the most virtuous stocks of [*sector*] sector, for [*indicator*]" where summarizer can be one of *very similar, discreetely similar or not similar* and the indicator can be one of *EBITDA, ROE, ROA, R&D investements or Net income.*

Summary 2 for a certain sector is generated as follow: A table is created having at the row the stocks of the sector for which the summary is being generate and for columns the most virtuous stocks of the other sector under consideration.

	Apple	Facebook	Tesla	Microsoft	Amazon
McDonalds	0.43	0.11	0.37	0.45	0.02
Burger King	0.11	0.31	0.53	0.23	0.11
Walmart	0.29	0.03	0.02	0.15	0.92

Given the overall distribution of similarities we calculate two quantiles named Q_1 and Q_2. We propose two methods to generate the summary:

1. We calculate the fraction of the most virtuous stocks which have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2 with respect to the reference stocks. The maximum fraction determines which summarizer to choose, in other words if we have that the major number of the most virtuous stock have a similarity less than Q_1 with respect to our reference stocks we choose *not similar*, if we have that the major number of the most virtuous stock have a similarity between Q_1 and Q_2 with respect to our reference stocks we choose *dicreetely similar* and if we have that the major number of the most virtuous stock have a similarity higher than Q_2 with respect to our reference stocks we choose *very similar*.

2. We calculate the average of the similarities in the , if the average is less than Q_1 we choose *not similar* as summarizer, if the average is between Q_1 and Q_2 we choose *dicreetely similar* as summarizer and if the average is higher than Q_2 we choose *very similar* as summarizer.

With respect to the first summary the method is very similar, the only difference is that we have a table with multiple rows, representing all the stocks of a given column, in the first method, we calculate the proportions all over the dataset instead of a single row, while in the second we perform an average of rows before performing the average on columns.

With these two methods, we generate two types of summaries named **Summary 2** with the first method and **Summary 2_avg** for the second.

3.4.3 Summary 3

With respect to the previous, summary 3 does not take into account reference profiles (most virtuous stocks) but instead compares the behaviour of a first stock with respect the behaviour of a second on different years, since there are no reference profile, there is not an indicator.
Summary 3 have the form of: "In [*quantifier*] years the stock [*Stock*] has been [*summarizer*] to the stock [*Stock*]" where quantifier can be one of *none, few, many, all*, and summarizer can be one of *very similar, discreetely similar or not similar*.
The summary is symmetric, it means that switching the two stocks does not modify the quantifier and the summarizer. In addition, this summary contains a **quantifier**, the quantifier is generated using a triangular membership function on the proportion of data that agrees with the summarizer.

Summary 3 for a certain sector is generated as follow: A table is created having at rows and columns the years of the two stocks under analysis, we take into consideration only the similarities on the diagonal to generate the summary because are the one that compares the stocks behaviour on the same year.

	Apple_2008	Apple_2009	Apple_2010
Google_2008	0.43	0.11	0.54
Google_2009	0.23	0.55	0.45
Google_2010	0.44	0.75	0.21

Given the overall distribution of similarities we calculate two quantiles named Q_1 and Q_2.

We calculate the fraction of the diagonal values which have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2. The maximum fraction is then passed to a triangular membership function to generate a quantifier-summarizer pair.

Let's focus on an example to explain better the latter method.

Given the fractions for which our diagonal values have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2 are: $[0.2, 0.35, 0.45]$, we pass the following to the triangular functions which give as output:

We choose as our quantifier-summarizer pair the one which has the maximum value, in

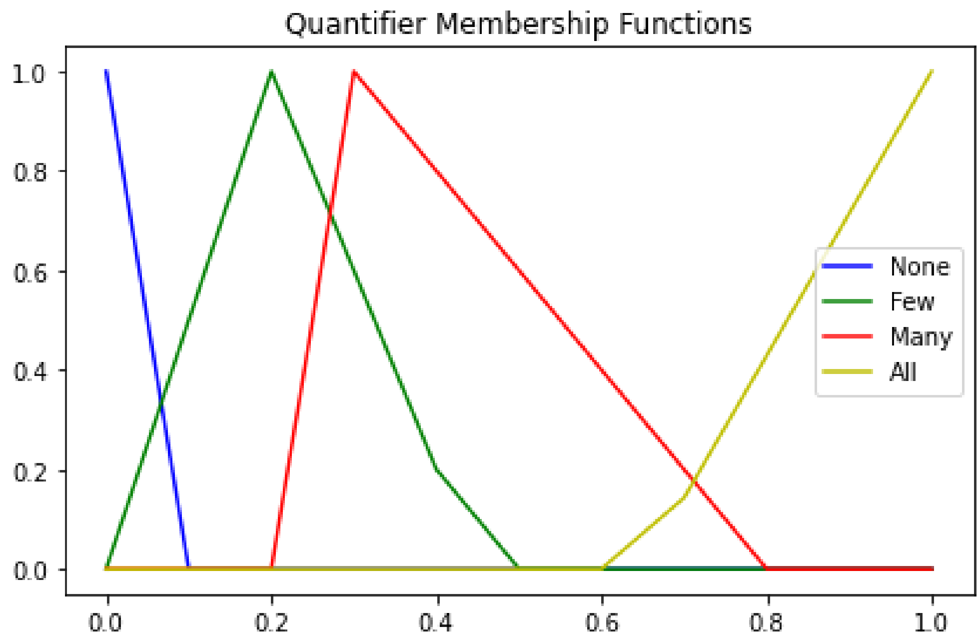

Figure 3.6: Triangular membership functions to generate summarizer-quantifier pairs.

	Not Similar	Discreetely Similar	Very Similar
None	0	0	0
Few	1	0.4	0
Many	0	0.9	0.7
All	0	0	0

this case Few - Not similar. In case of parity, we choose the one in the greater row, and in case of parity in the same row, the value in the greater column, this choice comes under the assumptions that a summary having a quantifier-summarizer pair such as All - Very similar is more informative than one having None - Not similar.

With this method, we generate one type of summaries named **Summary 3**.

3.4.4 Summary 4

Summary 4 is similar to summary 3 but given two stocks, instead of comparing the same years of both, compares one year of the first with respect all the years of the second. Summary 4 have the form of: "In year [*Year*] the stock [*Stock*] has been [*summarizer*] to [*quantifier*] of the stock [*Stock*]" where quantifier can be one of *none, few, many, all*, and summarizer can be one of *very similar, discreetely similar or not similar.*

Summary 4 for a certain sector is generated as follow: A table is created having at rows one year of the stock under analysis and as columns the years of the second stock under analysis.

	Apple_2008	Apple_2009	Apple_2010
Google_2008	0.43	0.11	0.54

Given the overall distribution of similarities we calculate two quantiles named Q_1 and Q_2.

We calculate the fraction of the values which have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2. The maximum fraction is then passed to a triangular membership function to generate a quantifier-summarizer pair.

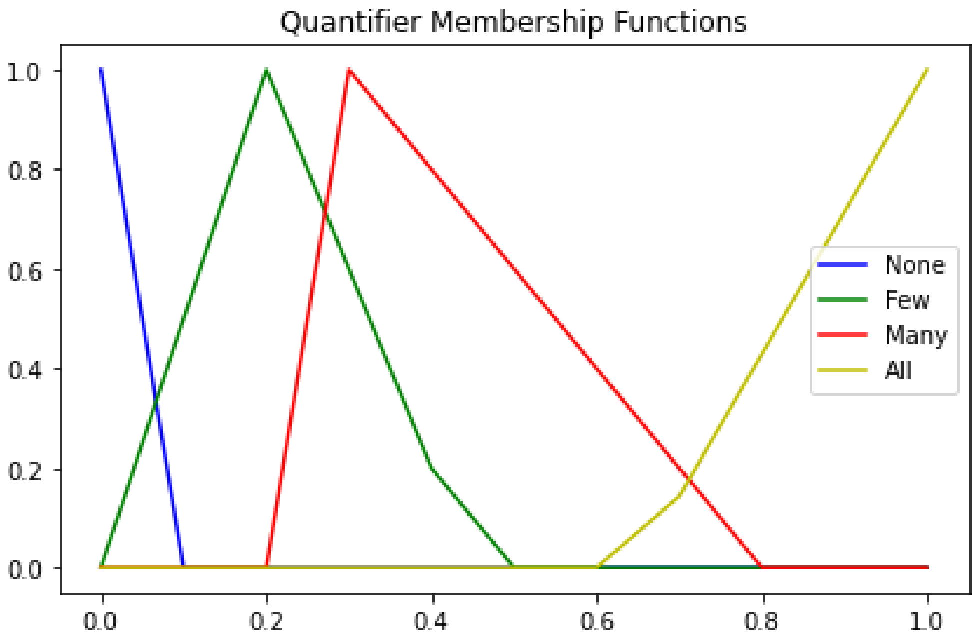

Figure 3.7: Triangular membership functions to generate summarizer-quantifier pairs.

Let's focus on an example to explain better the latter method.

Given the fractions for which our diagonal values have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2 are: $[0.2, 0.35, 0.45]$, we pass the following to the triangular functions which give as output:

	Not Similar	Discreetely Similar	Very Similar
None	0	0	0
Few	1	0.4	0
Many	0	0.9	0.7
All	0	0	0

We choose as our quantifier-summarizer pair the one which has the maximum value, in this case Few - Not similar. In case of parity, we choose the one in the greater row, and in case of parity in the same row, the value in the greater column, this choice comes under the assumptions that a summary having a quantifier-summarizer pair such as All - Very similar is more informative than one having None - Not similar.

With this method, we generate one type of summaries named **Summary 4**.

3.4.5 Summary 5

Summary 5 is on the form of: "Stock [*stock*] is [*summarizer*] at the [*fraction*] of the most virtuous stocks, for [*indicator1*]...[*indicatorN*]" where summarizer can be one of *very similar, discreetely similar or not similar* and the indicator can be one of *EBITDA, ROE, ROA, R&D investements or Net income*. Summary 5 is similar to summary 1, the main differences are that summary 5 reports explicitely the fraction of data that agrees with the summarizer, also, summary 5 can have multiple indicators.

Summary 5 for a certain stock is generated as follow:

A table is created having at the row the stock for which the summary is being generate and for columns the most virtuous stocks. Suppose that we have multiple rows, each for one indicator. Since the most virtuous stocks differs between indicators, some value are null.

	Apple	Facebook	Tesla	Microsoft	Amazon
Google_ebitda	0.25	0.11	0.37	0.45	null
Google_roe	0.22	0.11	null	0.11	0.05

It must be noticed that, there is no connection between the reference stock (Google) and EBITDA and ROE indicator, the indicator is reported in the row stock name just for visualization simplicity: the indicators are connected only to the set of chosen virtuous summaries (in column), a more appropriate choice would have been to output two different tables with at columns the most virtuous stocks for EBITDA and one for ROE indicator.

Given the overall distribution of similarities we calculate two quantiles named Q_1 and Q_2.

The summary is generated as follow: Given the union of the virtuous stock sets for both indicator; in this case we have the virtuous set for EBITDA which contains Apple, Facebook, Tesla and Microsoft; and the virtuous set for ROE which contains Apple, Facebook, Microsoft and Amazon; the union would be Apple, Facebook, Tesla, Microsoft and Amazon, we calculate the fraction of the most virtuous stocks which have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2 for **both** the indicators. In the example, given $Q_1 = 0.20$ and $Q_2 = 0.30$ the fraction of virtuous stocks which have a similarity less than Q_1 is $\frac{1}{5}$ (only Facebook), while the the fraction of virtuous stocks which have a similarity between Q_1 and Q_2 is $\frac{1}{5}$ (only Apple). Microsoft has one value greater than Q_2 and one less than Q_1 so it is not taken into analysis, as well as the other stocks, which contains a null value for one of the indicators. In this case of parity, we choose as the summarizer for our summary the second: *discreetely similar*, under the assumption that a summary containing *not similar* is less informative than a summary containing *very similar* as summarizer.

With this method, we generate a type of summary named **Summary 5**.

3.5 Evaluation Metrics

To evaluate the summaries we generate we exploit the usage of five metrics: the degrees of truth, imprecision, covering, appropriateness, and coverage, along with the length quality. This metrics has already been applied to summary evaluation in [19] to approximate the the maxims of quality, quantity, relevance, and manner. All of the metrics take a range from 0 to 1, where 1 is the ideal case.

3.5.1 Degree of Truth (T1)

The Degree of Truth quantifies how often a quantifier-summarizer pair is found true in the data, it represents a summary's truth value. We use Zadeh degree to measure to what extent our summaries follow the maxim of quality [43]. In our case the only summaries which contains a quantifier-summarizer pair are Summary 3 and 4, the degree of truth is the output of the triangular function presented in figure 3.6 and 3.7, a more detailed explaination on how these value are calculated can be found on the method to generate summary 3 and 4.

3.5.2 Degree of Imprecision (T2)

The degree of Imprecision measures how useful a summary is. Let's call r_{S_j} the fraction of the dataset that agrees with the summarizer j. The method to generate the r_{S_j} is explained in the first point of method to generate summary 1, this value is the fraction of data that agrees with a certain summarizer. In order to calculate the degree of imprecision, we use the following equation:

$$T_2 = 1 - \sqrt[m]{\prod_{j=1...m} r_{S_j}} \tag{3.3}$$

where m is the number of possible summarizers for the protoform type (in the case of summary 1 are 3: very similar, discreetely similar and not similar). We compute then the geometric mean of the fractions over the possible summarizers. The metric is at it's minimum when all the summarizers agrees on the same fraction of data meaning that the summary is not telling a useful information or is telling an obvious one. In the optimal case, we have one summarizer representing the greater fraction of data and the other two (in case of three summarizers) having a fraction close to zero. For example, let's suppose we are generating a summary for stock Google with the similarities with most performant stocks in columns as depicted in 3.6

	Apple	Facebook	Tesla	Microsoft	Amazon
Google	0.43	0.11	0.37	0.45	0.02

Table 3.5: Example of Table to generate summary 1 for Google stock

In this case the summarizer would be 'Very similar' because the similarity of the Google stock with the majority of the reference stock is bigger of the given treshold of 0.22; r_{S_j}, where j = 'Very similar', would be the proportion of the reference stocks that agrees with the summarizer 'Very similar' which is $\frac{3}{5}$. r_{S_j}, where j = 'Discreetely similar', would be the proportion of the reference stocks that agrees with the summarizer 'Discreetely similar' which is $\frac{1}{5}$. r_{S_j}, where j = 'Not similar', would be the proportion of the reference stocks that agrees with the summarizer 'Not similar' which is $\frac{1}{5}$. In the following case $T_2 = 1 - \sqrt[3]{\frac{1}{5}\frac{1}{5}\frac{3}{5}}$. We can see clearly how this metric reach its maximum when the data agrees in a distributed way among all three the summaries i.e when $r_{S_j} = \frac{1}{N}\ \forall j$ where N equals the cardinality of the set of summarizers.

3.5.3 Degree of Covering (T3)

Degree of Covering represent in fraction how often the summarizer is true in the data we are considering. Degree of covering is exactly equal to the r_{S_j} used to calculate T2, those are, one for each summarizer, the fraction of data that agrees with that summarizer. For a given summary, Degree of Covering T3 equals the r_{S_j} value for the chosen summarizer j. For example, let's suppose we are generating a summary for stock Google with the similarities with most performant stocks in columns as depicted in 3.6

	Apple	Facebook	Tesla	Microsoft	Amazon
Google	0.43	0.11	0.37	0.45	0.02

Table 3.6: Example of Table to generate summary 1 for Google stock

In this case the summarizer would be 'Very similar' because the similarity of the Google stock with the majority of the reference stock is bigger of the given treshold of 0.22; r_{S_j} then, where j = 'Very similar', would be the proportion of the reference stocks that agrees with the summarizer which is $\frac{3}{5}$.

3.5.4 Degree of Appropriateness (T4)

Degree of Appropriateness helps avoid trivial multivariate summaries. The degree's value represents how interesting and unexpected a finding in the summary may be. It can be applied only to multivariate summaries, summaries which have more than one attribute (in our case summary 5 which has more than one indicator). Degree of Appropriateness (T4) is calculated as following:

$$T_4 = |r^* - T_3| \tag{3.4}$$

where

$$r^* = \prod_{k=1}^{K} r_k \tag{3.5}$$

Where K is the set of attributes (in case of summary 5: the indicators). To have an intuitive understanding of the measure let's focus on how it is applied on Summary 5, recalling how summary 5 is created, focusing on two indicators for simplicity. Summary 5 is generated giving the union of the virtuous stock sets for both indicator; ifor example the virtuous set for EBITDA which contains Apple, Facebook, Tesla and Microsoft; and the virtuous set for ROE which contains Apple, Facebook, Microsoft and Amazon; we calculate the fraction of the most virtuous stocks which have a similarity less than Q_1, between Q_1 and Q_2 and higher than Q_2 for **both** the indicators. We choose the summarizer which gives the greater fraction to generate our summary and it will become our **T3**, given the summarizer we choose, our r_k are not the fraction of data that agrees with that summarizer for **both** the indicators, but instead for the **single** indicators, thus we would expect that in case of statistical independence between indicators the product of the fractions r_k of data that agrees with the summarizer for the single indicator k would be equal to the fraction of data that agrees with the summarizer for the conjunction of the two indicators.
This is why a finding greater than zero in the absolute value of the difference between r^* and T_3 is an unexpected finding.

3.5.5 Degree of Coverage (T5)

Degree of Coverage determines whether the conclusion made by the summary is supported by enough data. It is simply a function that takes that degree of covering and normalize it between 0 and 1 given two values, r_1 and r_2 which can be interpreted as a prior expected range for the degree of covering. The function is computed as follows:

$$T_5 = \begin{cases} 0, & r_c \leq r_1 \\ 2\left(\frac{r_c - r_1}{r_2 - r_1}\right)^2, & r_1 < r_c < \frac{r_1 + r_2}{2} \\ 1 - 2\left(\frac{r_2 - r_c}{r_2 - r_1}\right)^2, & \frac{r_1 + r_2}{2} \leq r_c < r_2 \\ 1, & r_c \geq r_2 \end{cases} \tag{3.6}$$

3.5.6 Length Quality (T6)

Length Quality states how clear a summary is in function of the number of summarizers included in the summary. Length quality is computed as follows

$$T_6 = 2\left(0.5^{\text{cardS}}\right) \tag{3.7}$$

In the summary we generated for example, there is only one summarizer: Either 'Very Similar', 'Discreetely Similar' or 'Not Similar'. Then the lenght quality will always be equal to 1. But it can happen, maybe in extension of our generated summaries that a summary could include many summarizers, if for example the summary contains two summarizers: 'Very Bright' and 'Very Green', the lenght quality would be equal to $2(0.5^2) = 0.5$. We can see clearly that at the growing of the number of summarizers the length quality tends to diminish.

Chapter 4

Experimental results

4.1 Dataset

We ran experiments on the SP 500 dataset, consisting of ten years of the price time series of the stocks, spanning from 2007 to 2018, and their balance documents. The balance documents from 2007 to 2018 were available only for 468 out of the 500 firms, since some of those where out of the spanning period or corrupted. This problem did not prevent us to carry the analysis, because it just may reduce the number of reference profiles, although it is a point of improvement for future developments of the algorithm.

The sentiment analysis has been conducted with a collection of news about the stocks gathered by Reuters [5].

The collection of news includes 493 stocks with an average of 5253 news per stock, every news has a title and a body, the distribution of the amount of news per stocks depends upon the media hype of the latter, with most popular stocks as Apple or Google having the higher amount of news collected.

4.2 Libraries

The main libraries used to perform the analysis are:

- Pandas: an open source, easy-to-use data structures and data analysis tools.[4]
- NumPy: an open source package for scientific computing.[3]
- VADER: a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains. [6]
- Gensim: a Python library for topic modelling, document indexing and similarity retrieval with large corpora. [1]
- Matplotlib: a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.[2]

4.3 Hardware and Computation

Hardware: Intel®Xeon®X5650, 32 GB of RAM and running Ubuntu18.04.1 LTS.

Task	**Execution time**	**Standard deviation**
Time series Discretization	240s	3s
Sentimental Analysis	324s	5s
Embeddings Generation	594s	15s
Summary 1 Generation	245s	5s
Summary 2 Generation	220s	3s
Summary 3 Generation	2793s	23s
Summary 4 Generation	more than 5000s	
Summary 5 Generation	260s	4s

Table 4.1: Computational Time of each macro-operation.

Table 1.1 shows the computation time for each of the macro processes in the algorithm. The Summary generation time depends strictly on the number of summary generated and on their computational complexity, the summaries more easy to generate are the first, the second and the fifth because are in the order of thousand, the third summary is in the order of hundred thousand since it is generated a summary for every pair of stocks and for every indicator. The fourth is in the order of million since it is generated a summary for every pair of stock and for every year, this latter summary set has been not generate completely for practical reasons, the statistics shown in the next pages are referred to a generated sample of the order of ten thousand.

4.4 Configuration

The main parameters on which the system depends are:

- **Discretization Parameters:** Discretization parameters are the parameters adopted to discretize the stock time series, in particular the periods used to calculate the moving average convergence divergence (5,20;20,50;50,200), percentages for which are triggered events in RSI (30% and 70%) and so on, all choice are found to be explained in chapter 3.1.
- **Embedding Parameters:** The embedding algorithm is trained with PV-DBOW model [29] for a vector size of 100 and 10 epochs and a window of 5 words.
- **Ranking Parameters:** The ranking algorithm takes as parameter the quantile for which a stock is classified as viruous which is in our case the third quartile.
- **Summary Parameters:** The summarization algorithm takes as parameter the treshold for which a stock is classified as very similar, discreetely similar or not similar to

another which is 0,20 and 0,30 based on the analysis of quantiles; another parameter the summarization algorithm takes into is the parametrization of the triangular function to generate summary 3 and 4, which can be found in section 3.4.3 and 3.4.4.

4.5 Experimental results

The analysis of results will go trough the distribution of measures with respect the overall distribution of the summary, then we focus on particular stocks to explain how the analysis of summaries can be made in support of decision making and also we will see how the summaries effectively captures the behaviour of the stock time series.

4.5.1 Distribution of Measures

Summary Type	**Summary**	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
1	Stock DISCK is discreetly similar to most of the most virtuous stocks, for the EBITDA indicator.		0,99	0,41		1	1
1_avg	Stock FITB is, on average, very similar to most of the most virtuous stocks, for the ROE indicator.		0,99	0,52		1	1
2	Most of the stocks of Financial Services sector, has been not similar to most of to the most virtuous stocks of Energy sector for the ROE indicator.		0,99	0,51		1	1
2_avg	Most of the stocks of Communication Services sector, has been not similar on average to most of the most virtuous stocks of Consumer Discretionary sector for the EBITDA indicator.		0,99	0,51		1	1
3	In few years the stock KORS has been very similar to the stock INFO	0,83	0,87	0,17		1	1
4	In year 2014 the stock LNT has been very similar to few years of the stock INFO	0,93	0,72	0,33		1	1
5	Stock FITB is discreetly similar at the 22% of the most virtuous stocks, for the EBITDA and the Net Income indicator.		0,91	0,22	0,19	1	1

Table 4.2: Example of generated summaries.

Table 1.2 shows examples of generated summaries, it can be noticed that there are missing values in the evaluation metrics. The only summaries for which the Degree of truth (T1) holds a result are summary 3 and 4 since the other summaries does not comprehend a quantifier. Also, the only summary which have a degree of appropriateness (T4) is the summary 5 since is the only multivariate summary.

To judge qualitatively our results, we can compare the distribution of a singular summary class to the overall distributions of all the summaries classes, for different metrics. In this way we can evaluate how a summary class or a summary is disposed with respect to the overall distribution of metrics.

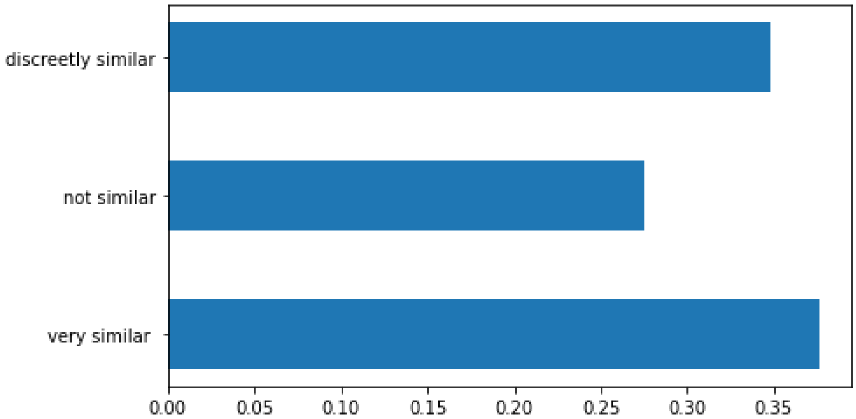

Figure 4.1: Distribution of Summarizers over all the generated summaries

Figure 1.1 shows the overall distribution of the Summarizers, all the three Summarizers are in average, for all the summaries, around the 33% percent which is the expected value by the method we used to choose the threshold of similarities as described in the previous chapters.

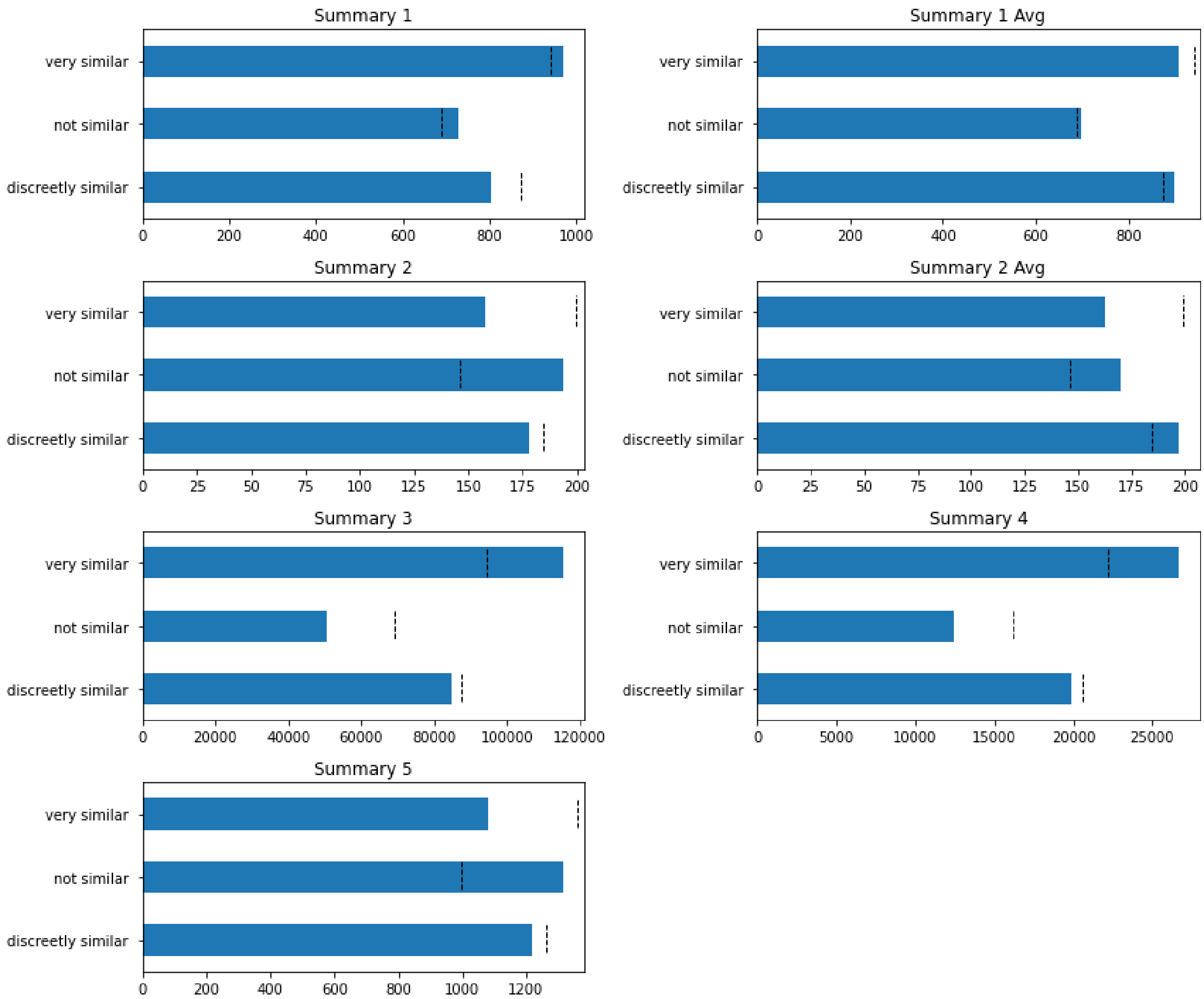

Figure 4.2: Distribution of Summarizers for each of the generated summaries, the overall mean is indicated by dashed lines

In figure 1.2 we can see a comparison of the distribution of the summarizers in the specific summaries classes with respect to the overall distribution of summarizers, the overall distribution, indicated by the dashed line shows some expected behaviors related to the methods of generating the summaries, for example in the case of summary 3, in the choice of which summarizer we would choose for the summary, we choose the one that maximizes the T1 metric, in case of equality we choose the one having the most higher similarity with the reference profile (under the assumption that higher the similarity expressed in the summary, the greater is its informativeness), for example the summary "stock1 is very similar to many years of stock2' is expected to be less informative on the behavior of the stock with respect to "stock1 is not similar to few years of stock2'. By the distribution of the summarizers in summary 3 is clearly reflected this choice because we can see that the 'Very similar' is way above the mean of the classes.

Let's now look in details how the different evaluation metrics distributes over the generated summaries.

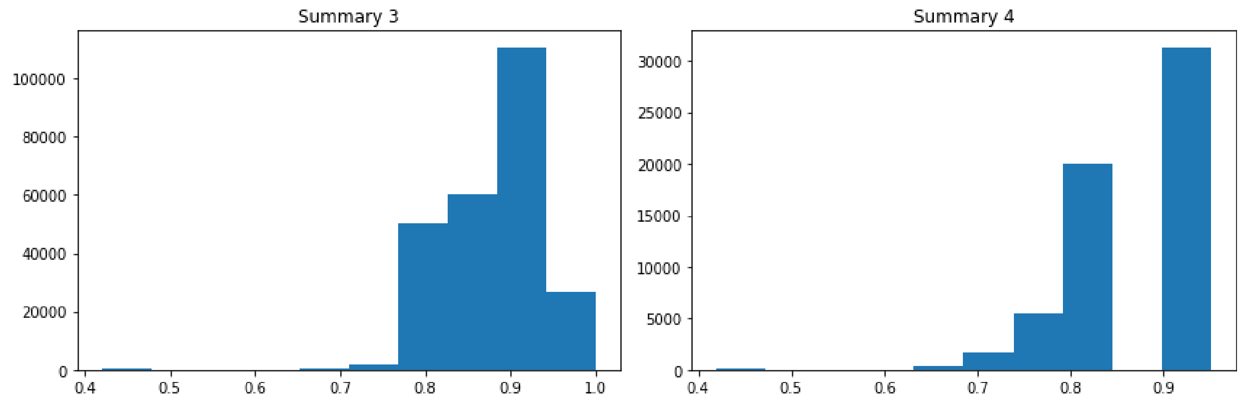

Figure 4.3: Distribution of degree of Truth (T1) for each of the generated summaries.

Degree of truth (T1) quantifies the truth of the quantifier-summarizer pair expressed by the summary, since the quantifier is only present in summaries 3 and 4, the other degree of truth cannot be calculated, as we can see in figure 1.3, the degree of truth (T1) distributes similarly between summary 3 and 4 with a peaked distribution around 0.9.

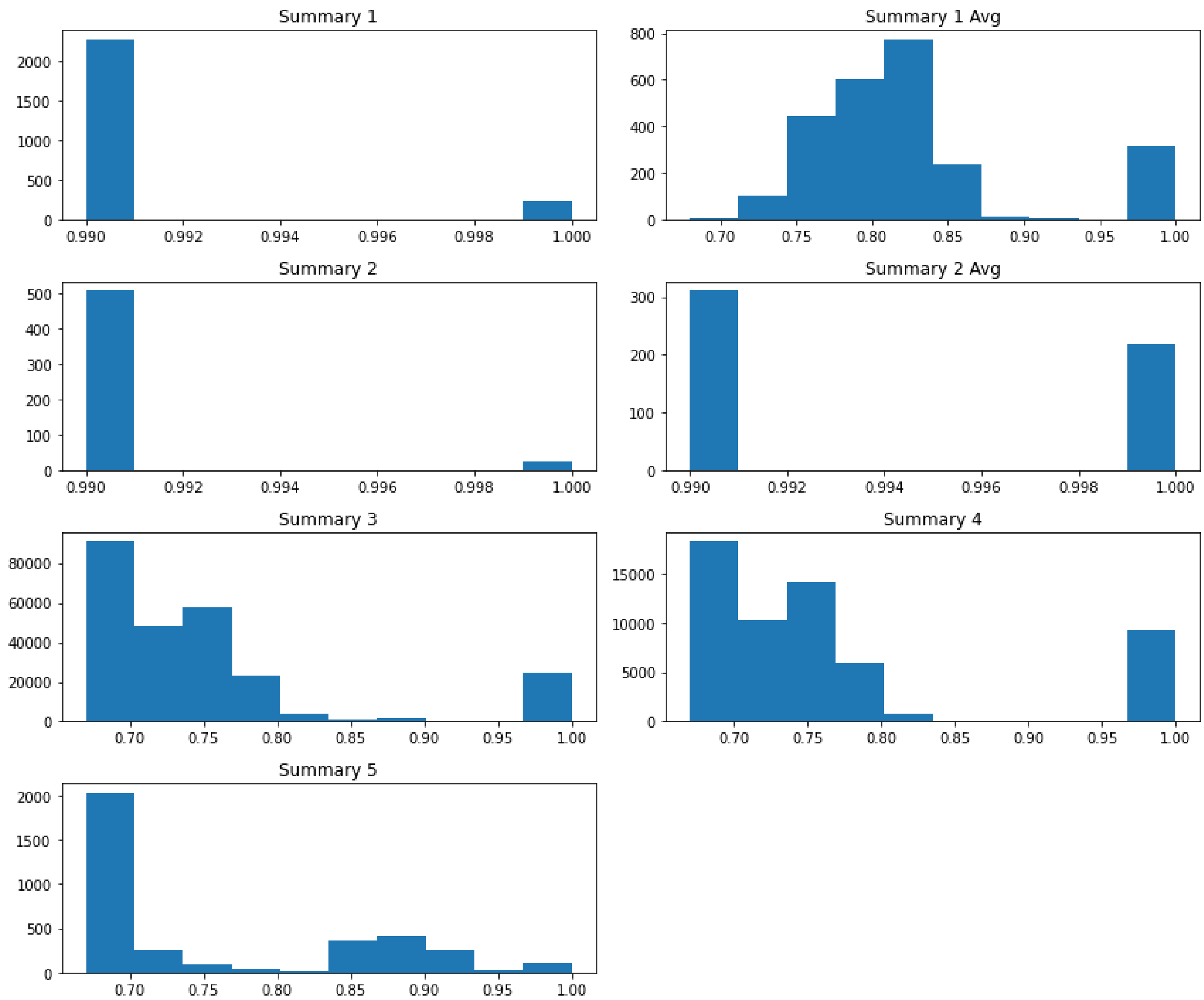

Figure 4.4: Distribution of degree of imprecision (T2) for each of the generated summaries.

Degree of Imprecision (T2) measures how useful a summary is, in function of the covering of each summarizer in the summarizers set. From Figure 1.4 we can see two different distributions: for summary 1 and summary 2 the degree of imprecision is very close to 1, this is due to the fact that these two summaries are generated chosing the summary with the highest degree of covering, an high degree of covering produces an high degree of imprecision. In the case of the other summaries the generation is independent of the degree of covering thus the distributions of the degree of imprecision look similar.

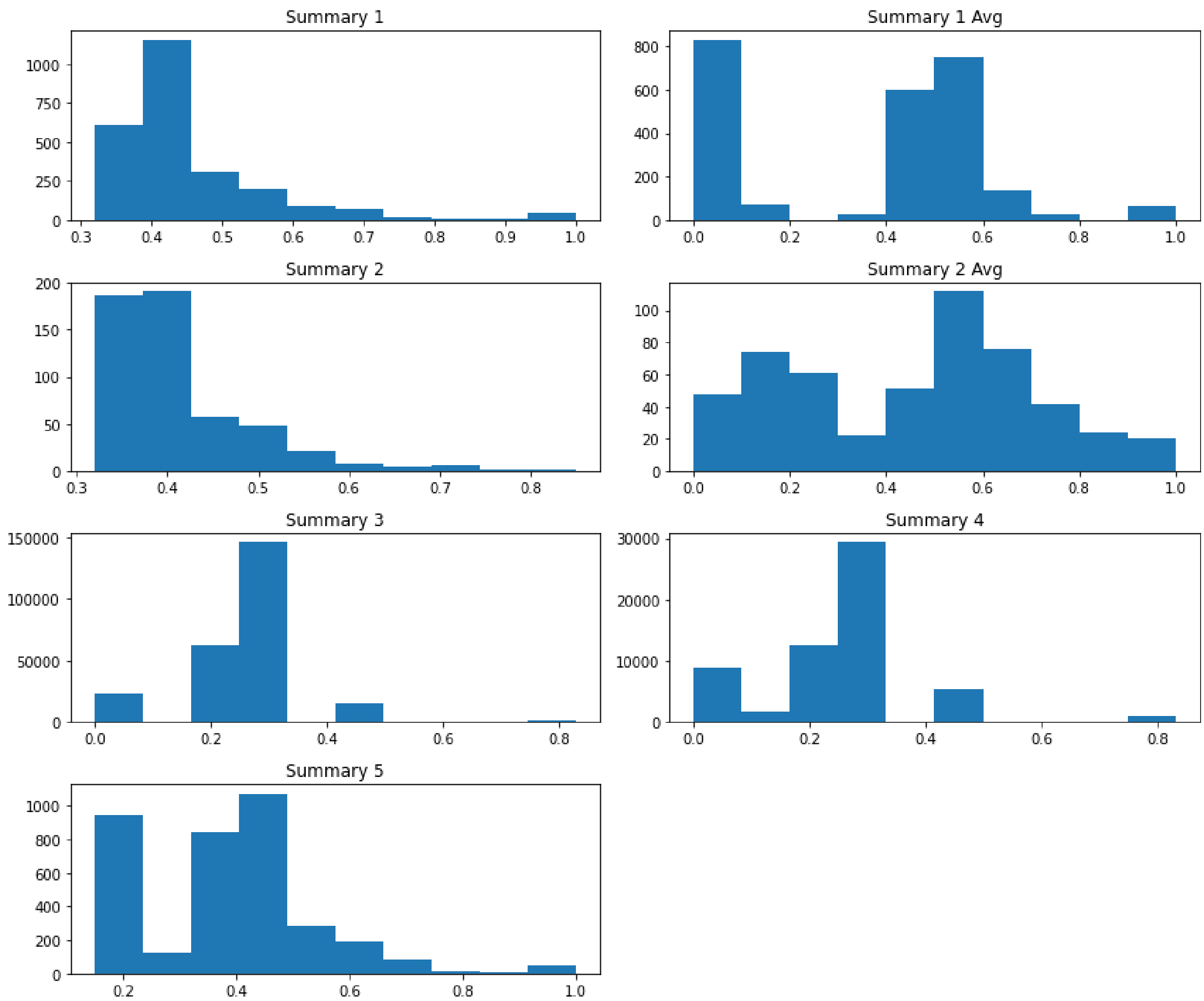

Figure 4.5: Distribution of degree of covering (T3) for each of the generated summaries.

A very important quality indicator in the degree of covering (T3) which measures in percentage how many instances in the database under consideration are covered by the summary statement. We can in figure 1.5 how the degree of covering distribute differently between the summary 1 and the summary 2 and all the others, this is due to the fact that the method of generation of summary 1 and 2 is dependent on the degree of covering, and only summaries with a degree of covering greater or equal of a third are taken into account. This clearly biases the overall distribution. This reflects what is stated above about T2, the distinction between those two distribution is due to the same cause of the distinction between the distributions of the degree of imprecision.

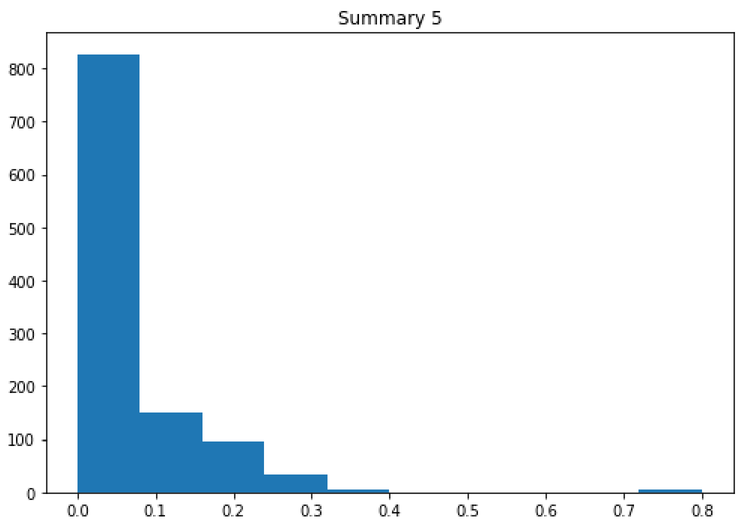

Figure 4.6: Distribution of degree of Appropriateness (T4) for summary 5.

Degree of Appropriateness (T4) helps avoid trivial multivariate summaries. The degree's value represents how interesting and unexpected a finding in the summary may be. This evaluation metrics applies only in the case of multivariate summaries (summaries with more than one attribute) in our generated set thus applies only for summary 5. As it can be noticed in figure 1.6 the majority of the values is around zero, this is expected because there is a mathematical reason under it, the degree of appropriateness is the difference between the conjunction probability of the summary to be true in the data for both attributes and the product of the probability for the summary to be true in data for each attribute, since we hypothesize the independence of the attributes, this two values should be equal and their difference expected value is equal to zero.

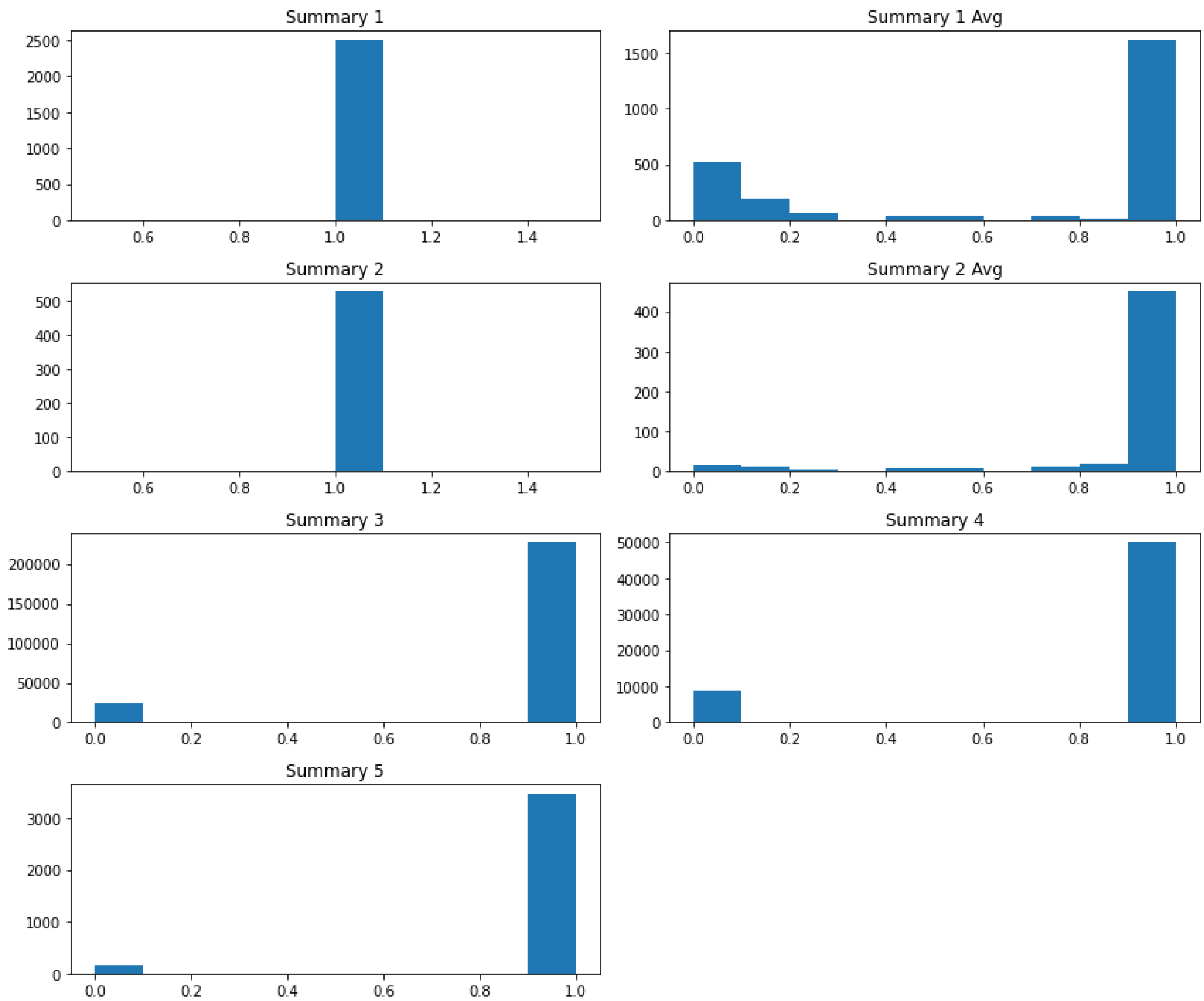

Figure 4.7: Distribution of degree of Coverage (T5) for each of the generated summaries.

Degree of Coverage (T5) determines whether the conclusion made by the summary is supported by enough data. It is simply a normalization of the degree of covering in the range of 0,1. The normalization is such that values of the degree of covering greater than a given number takes value of one and less than a given number take value of zero, the values of the degree of covering between these given one takes value of degree of coverage between 0 and 1. We can see in figure 1.7 that as in the case of degree if imprecision (T2) and degree of covering (T3) that there are two different distribution, the one of summary 1 and 2 and the one of the others, this is due to the reason that in these two summaries the degree of covering is restricted to be greater of 0.33 as stated above.

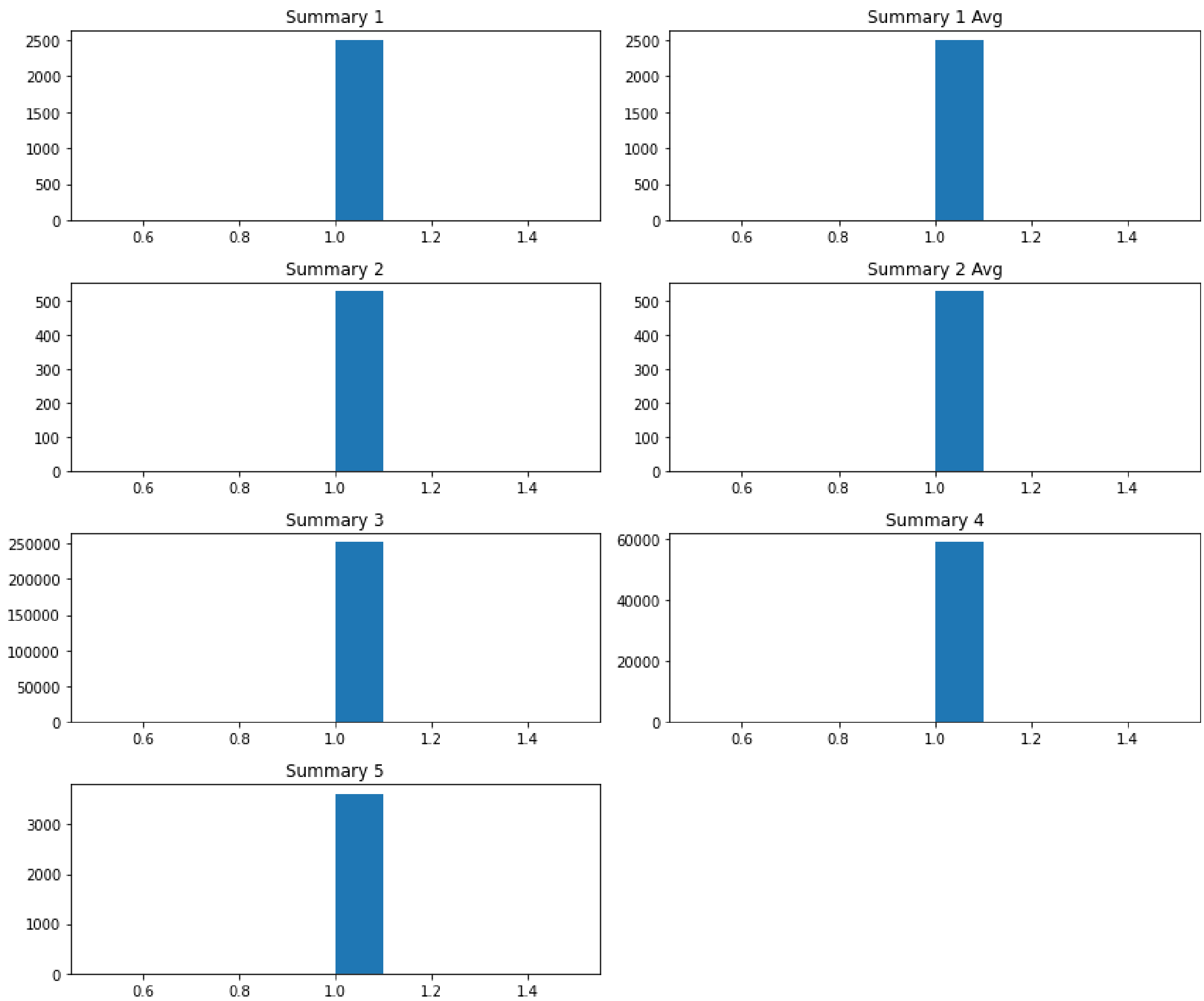

Figure 4.8: Distribution of Length Quality (T6) for each of the generated summaries.

Length Quality (T6) States how clear a summary is in function of the number of summarizers included in the summary, in our case, the generated summary contains one only summarizer, so the length quality takes value of 1 over all the generated summaries.

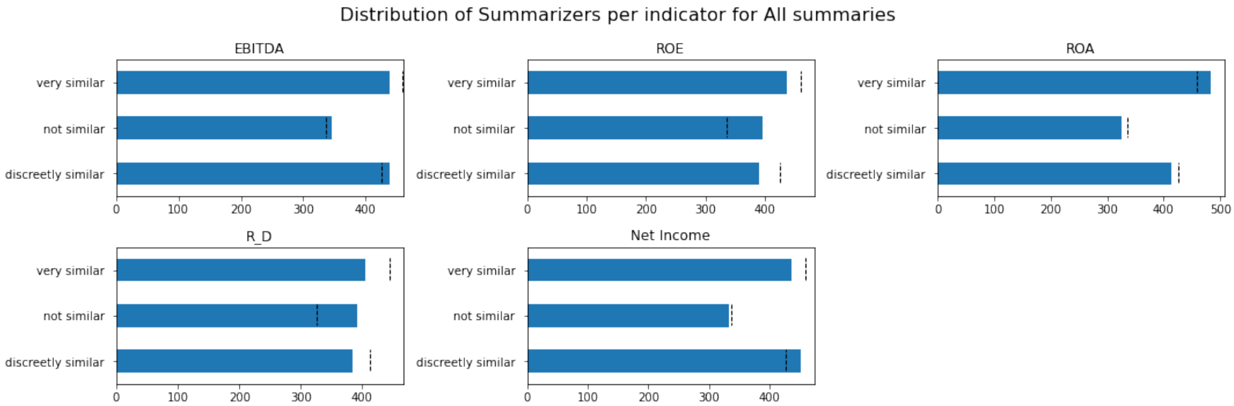

Figure 4.9: Distribution of Summarizer for each of the analyzed reference indicator.

We can also analyze the distribution of the summarizers for each different indicator, in figure 4.9 we can evaluate the distribution of summarizer with respect to the mean distribution indicated in in dashed line, we can conclude that with respect to the ROA reference profiles the majority of the stock has a very similar behaviour, while with respect to the reference profiles for R&D indicator the distribution of summarizers is balanced.

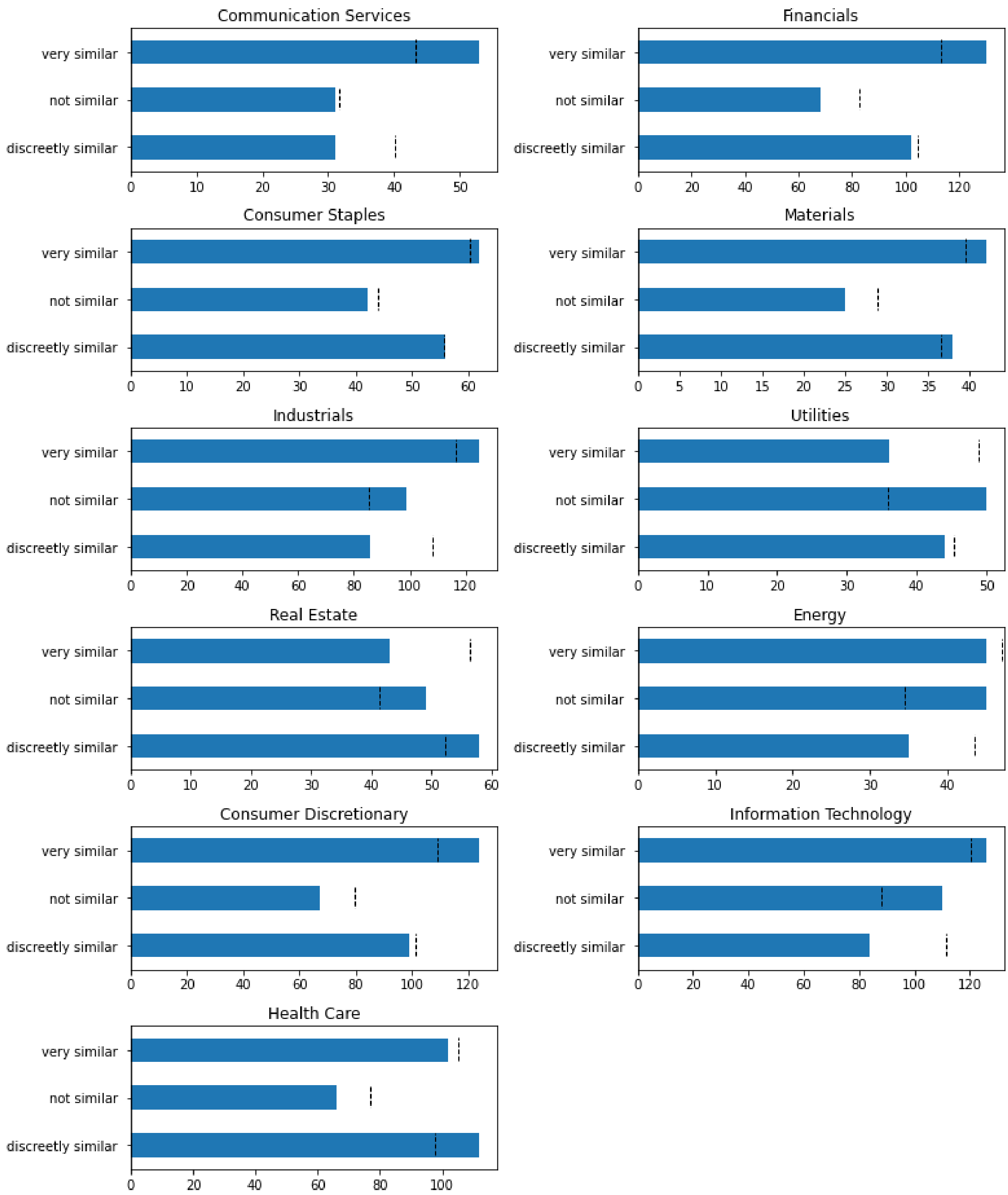

Figure 4.10: Distribution of summarizers per sectors in summary 2 with respect to most performant stocks.

Not only the summary itself but also the aggregation of the summaries can bring value to the final user, for example, as demonstrated in figure 1.9 the summarizers of the summary 2 can be aggregated per sector, thus showing which sector is more similar to the most performant stocks. This analysis can be carried out to see which are the most performant sectors and to compare them with each other.

4.5.2 Analysis of Results

Recalling that the summaries are generated to support the decision of the target user, in this section we will show how effectively the overall behaviour of the stock is captured by the summary and how the summaries can be used in practice to take decisions.

Summary	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Stock AAPL is very similar to most of the most virtuous stocks, for the ROA indicator.		0,99	0,41		1	1
Stock AAPL is not similar to most of the most virtuous stocks, for the EBITDA indicator.		0,99	0,4		1	1
Stock AAPL is very similar to most of the most virtuous stocks, for the ROE indicator.		0,99	0,37		1	1
Stock AAPL is very similar to most of the most virtuous stocks, for the Net Income indicator.		0,99	0,36		1	1
Stock AAPL is very similar to most of the most virtuous stocks, for the R&D indicator.		0,99	0,32		1	1

Table 4.3: Example of generated summaries 1 for stock Apple ordered by T3 measure.

in Table 4.3 has been generated the summary 1 relative to apple stock ordered by T3 measure which can be interpreted as confidence about the generated summary (percentage of data that agrees with the summarizer), a user can take the conclusion that the stock Apple is well performing from the point of view of ROA while it is not from the point of view of EBITDA or at least not as well as the most virtuous stock, let's check how this captures the behaviour of the time series:

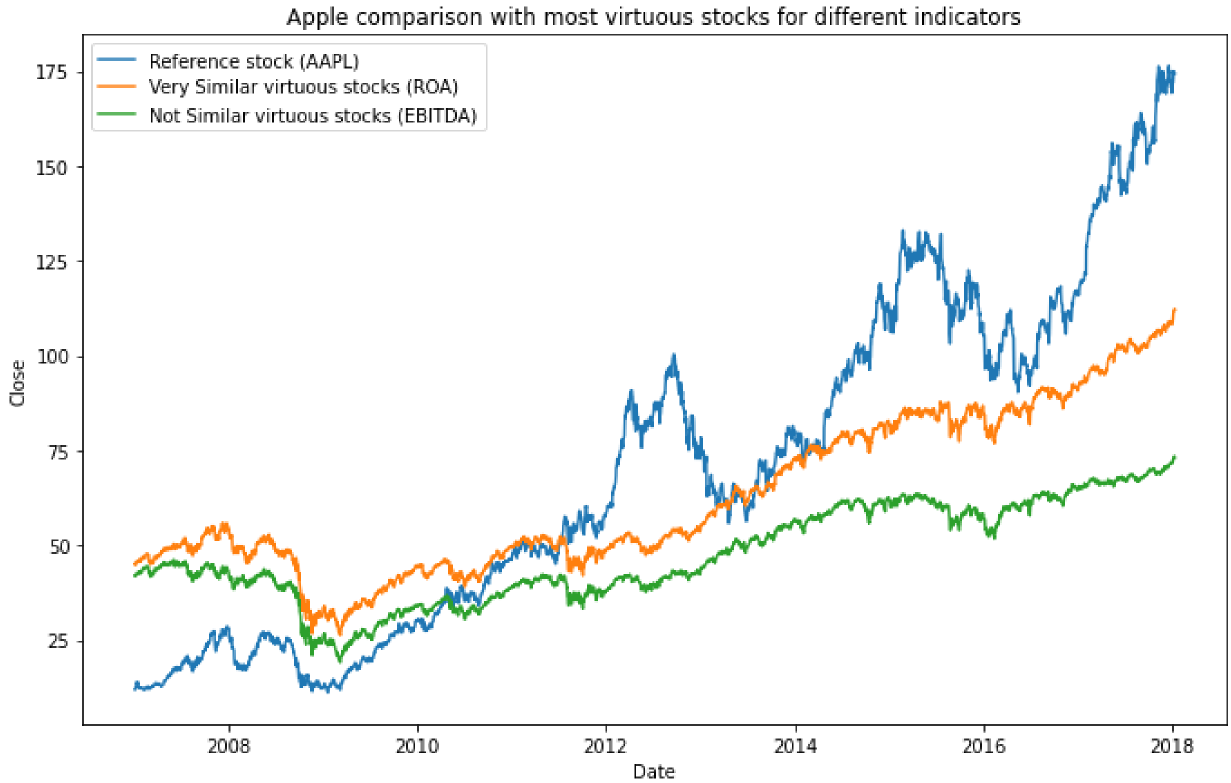

Figure 4.11: Comparison of Apple stock with respect to the most virtuous stocks for ROA and EBITDA indicators

in Figure 4.11 We compared the behaviour of Apple stock with respect to the mean behaviour of the most virtuous stocks for ROA and EBITDA, we can see clearly that the summary captures the higher similarity of the apple stock with the ROA most virtuous stocks with respect to the EBITDA one. It must be noticed that although the very similar and not similar virtuous stock shows similar behaviour in the chart, this is only due to the mean reduction, in facts the summary captures more wide properties of the singular series which cannot be visualized in the chart.

Summary	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Most of the stocks of Energy sector, has been very similar to most of to the most virtuous stocks of Industrial sector for the R&D indicator.		0,99	0,65		1	1
Most of the stocks of Energy sector, has been very similar to most of to the most virtuous stocks of Consumer Discretionary sector for the R&D indicator.		0,99	0,55		1	1
Most of the stocks of Energy sector, has been very similar to most of to the most virtuous stocks of Utilities sector for the R&D indicator.		0,99	0,54		1	1
Most of the stocks of Energy sector, has been very similar to most of to the most virtuous stocks of Materials sector for the R&D indicator.		0,99	0,52		1	1
Most of the stocks of Energy sector, has been not similar to most of to the most virtuous stocks of Utilities sector for the ROA indicator.		0,99	0,47		1	1

Table 4.4: Top-5 generated summaries 2 for Sector Energy by T3 value.

in Table 4.4 has been generated the summary 2 relative to the Energy sector stocks ordered by T3 measure which can be interpreted as confidence about the generated summary (percentage of data that agrees with the summarizer), a user can take the conclusion that the stock of the Energy sector are performing very similar to the one of Industrial, Consumer Discretionary, Utilities and Material with various confidence levels, while is not performing similarly to the Utilities Sector let's check how this captures the behaviour of the time series:

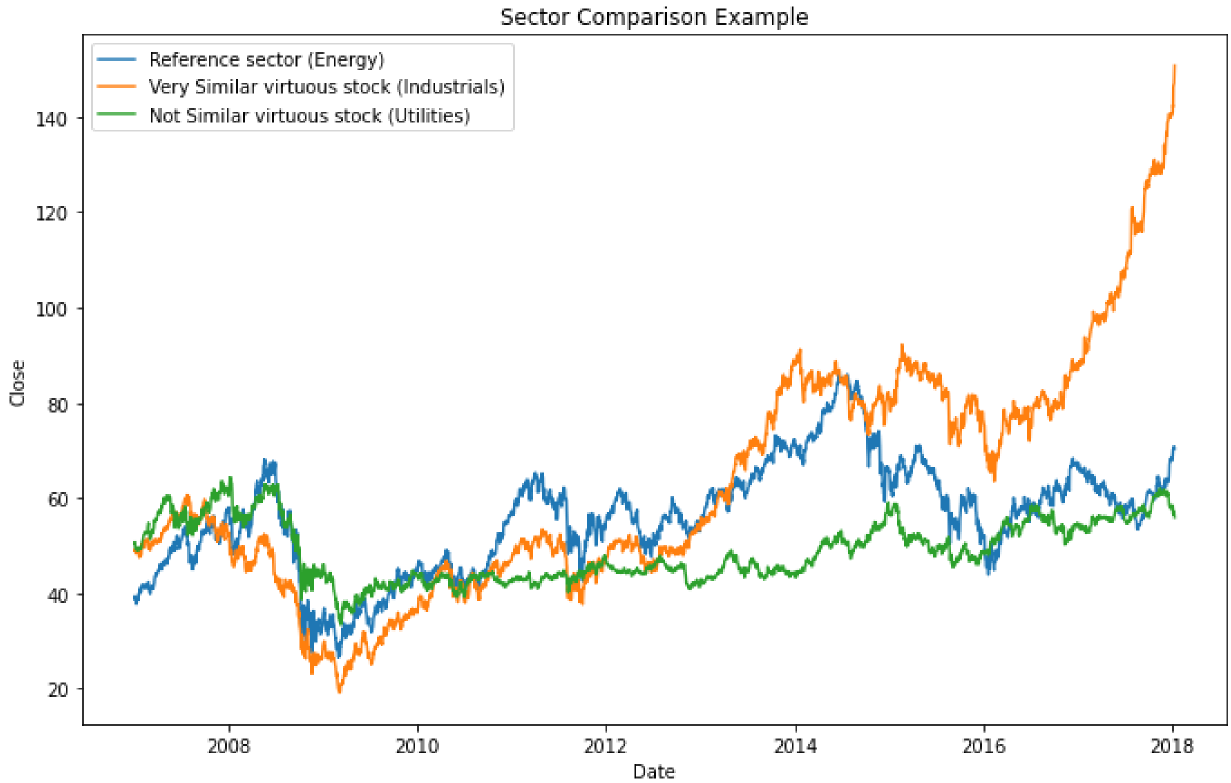

Figure 4.12: Comparison of Energy Sector with respect to the most virtuous stocks for Industrial and Utilities Sectors.

In Figure 4.12 We compared the behaviour of Energy sector with respect to the mean behaviour of the most virtuous stocks for Industrial and Utilities sectors, we can see clearly that the summary captures the higher similarity of the Energy sector with the Industrial most virtuous stocks with respect to the Utilities one. With the stable behaviour of the latter with respect to the more bullish behaviour of the first.

Summary	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
In most years the stock AAPL has been very similar to the stock ANSS	0,77	0,74	0,42		1	1
In most years the stock AAPL has been not similar to the stock VRTX	0,77	0,74	0,42		1	1
In most years the stock AAPL has been discreetely similar to the stock TRIP	0,77	0,76	0,42		1	1
In most years the stock AAPL has been discreetely similar to the stock ABBV	0,77	0,86	0,42		1	1
In most years the stock AAPL has been not similar to the stock CME	0,77	0,74	0,42		1	1

Table 4.5: Top-5 of generated summaries 4 for stock Apple by T3 measure.

In Table 4.5 has been generated the summary 3 relative to the Apple stock ordered by T3 measure which can be interpreted as confidence about the generated summary (percentage of data that agrees with the summarizer), a user can take the conclusion that the Apple stock is performing very similarly to the ANSS stock and not similarly to the stock CME, a user can also specify which stocks to compare, here we reported the top-5 stocks for the T3 measure for simplicity, let's check how this conclusions are reflected on the behaviour of the time series:

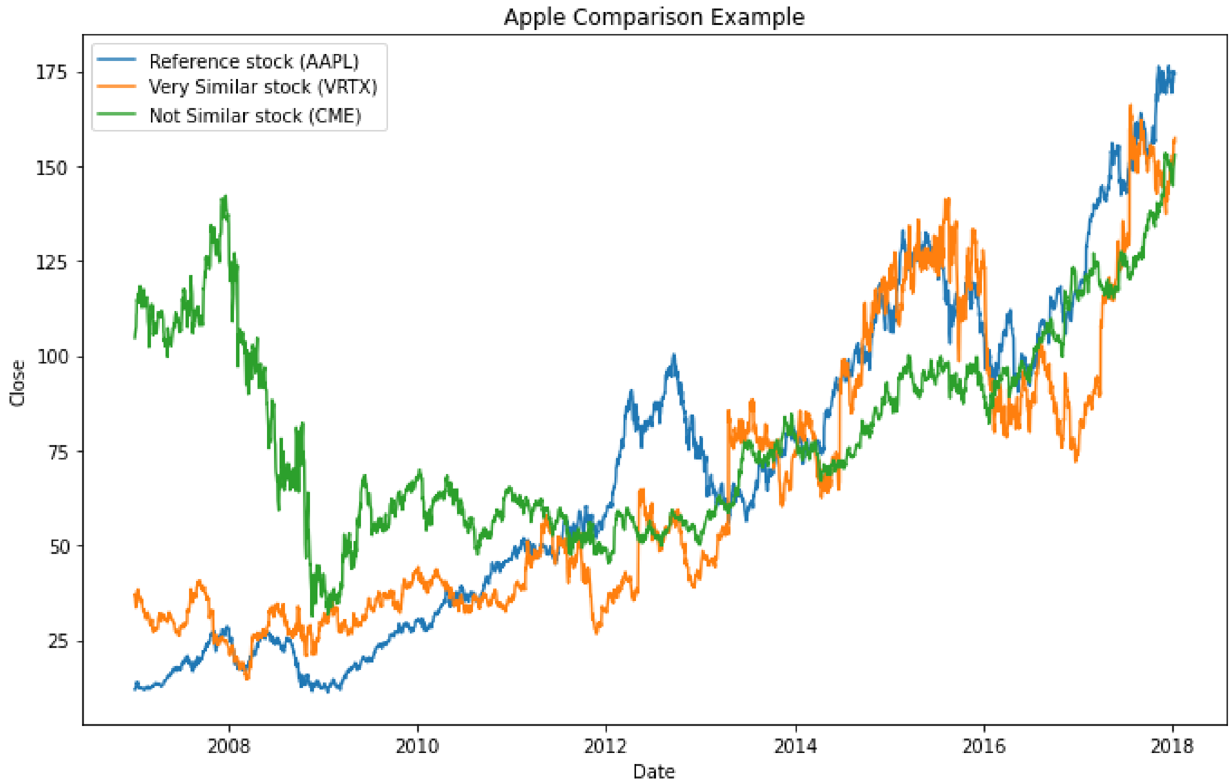

Figure 4.13: Comparison of Apple stock with respect to VRTX and CME stocks.

in Figure 4.13 We compared the behaviour of Apple stock with respect to the behaviour of VRTX and CME stocks, we can see clearly that the summary captures the higher similarity of the Apple stock with the VRTX stock with respect to the CME stock.

Summary	T1	T2	T3	T4	T5	T6
In year 2016 the stock DUK has been not similar to all years of the stock HPE	0,95	1	0,83		1	1
In year 2009 the stock JKHY has been not similar to all years of the stock HPE	0,95	0,82	0,83		1	1
In year 2014 the stock KO has been not similar to all years of the stock HPE	0,95	1	0,83		1	1
In year 2014 the stock DG has been very similar to all years of the stock HPE	0,95	1	0,83		1	1
In year 2014 the stock IP has been discreetely similar to all years of the stock HPE	0,95	1	0,83		1	1

Table 4.6: Example of generated summaries 4 of different stocks in different years with respect to HPE behaviour on all the years.

in Table 4.6 has been generated the summary 4 relative to the HPE stock ordered by T3 measure which can be interpreted as confidence about the generated summary (percentage of data that agrees with the summarizer), a user can take the conclusion for example that the DG stock is performing very similarly to the HPE stock in year 2014 or that the DUK stock has behaved not similarly to the stock HPE in year 2016, a user can also specify which stocks to compare, here we reported the top-5 stocks for the T3 measure for simplicity, let's check how this conclusions are reflected on the behaviour of the time series:

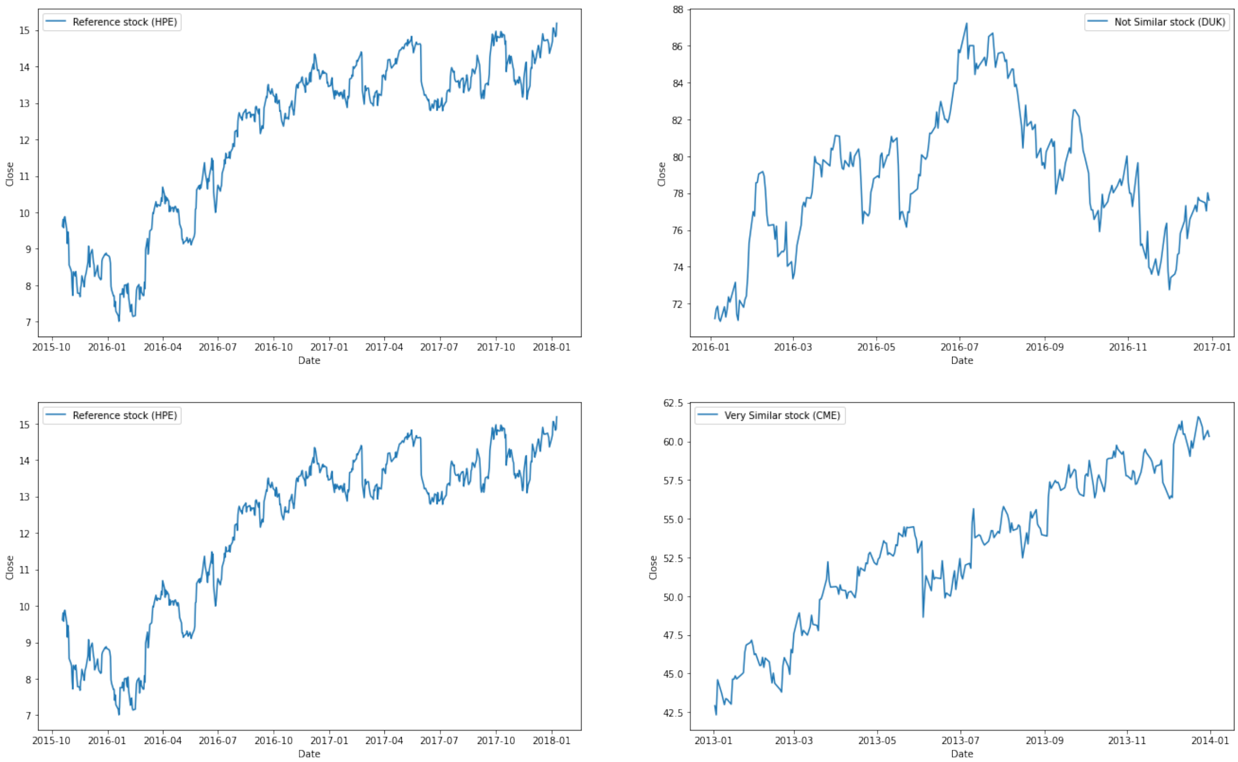

Figure 4.14: Comparison between DG and DUK stock in 2014 and 2016 respectively with respect to HPE stock

in Figure 4.14 We compared the behaviour of DG stock in 2014 with respect to the behaviour of HPE stock and the behaviour of DUK stock in 2016 with respect to the behaviour of HPE, we can see clearly that the summary captures the higher similarity of the HPE stock with respect to the DG stock in 2014 and the high dissimilarity of the HPE stock with respect to the DUK stock in 2016 .

Similarly on how we evaluated the summary 4 for HPE stock with respect to the degree of covering (T3) indicator, we can do the same with other indicators.

Summary	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
In year 2018 the stock PSA has been very similar to none years of the stock HPE	0,95	1	0		0	1
In year 2010 the stock COP has been very similar to none years of the stock HPE	0,95	1	0		0	1
In year 2008 the stock CHRW has been very similar to none years of the stock HPE	0,95	1	0		0	1
In year 2009 the stock FDX has been not similar to all years of the stock HPE	0,95	0,82	0,83		1	1
In year 2010 the stock CHRW has been very similar to none years of the stock HPE	0,95	1	0		0	1

Table 4.7: Example of generated summaries 4 of different stocks in different years with respect to HPE behaviour on all the years sorted by Degree of Truth (T1).

In table 4.7 is shown the summary 4 for stock HBE sort by the Degree of Truth T1, the degree of truth quantifies the truthfullness of the quantifier-summarizer pair, in this case we can see we have the greater degrees of truth for a degree of covering near zero, means that none of the yearly behaviour of stock HPE resemble the listed stock in the given year. The only exception we have is in the fourth listed summary, in which a very high degree of covering leads to an high degree of truth for the summarizer 'all'.

Summary	T1	T2	T3	T4	T5	T6
In year 2018 the stock PSA has been very similar to none years of the stock HPE	0,95	1	0		0	1
In year 2009 the stock PRU has been not similar to none years of the stock HPE	0,95	1	0		1	1
In year 2016 the stock ANTM has been very similar to none years of the stock HPE	0,95	1	0		0	1
In year 2007 the stock ANTM has been not similar to none years of the stock HPE	0,95	1	0		1	1
In year 2011 the stock ANTM has been discreetely similar to none years of the stock HPE	0,95	1	0		0	1

Table 4.8: Example of generated summaries 4 of different stocks in different years with respect to HPE behaviour on all the years sorted by degree of imprecision (T2).

In table 4.8 is shown the summary 4 for stock HBE sort by the Degree of Imprecision T2, the degree of truth measures how useful a summary is in function of the covering of each summarizer in the summarizers set. in this case we can see we have the greater degrees of truth for a degree of covering near zero, this is because the degree of imprecision penalizes obvious summaries (summaries for which the degree of covering is high over all the dataset) and promotes non trivial summaries (summaries for which the degree of covering approaches zero). The only exception we have is in the third listed summary, in which a very high degree of covering leads to an high degree of truth for the summarizer 'all'.

Summary	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
In year 2014 the stock FLR has been very similar to few years of the stock HPE	0,8	0,69	0,25		1	1
In year 2009 the stock BIIB has been very similar to all years of the stock HPE	0,93	0,77	0,33		1	1
In year 2017 the stock ZTS has been very similar to most years of the stock HPE	0,77	0,76	0,42		1	1
In year 2015 the stock ZTS has been not similar to few years of the stock HPE	0,83	0,74	0,17		1	1
In year 2016 the stock ZTS has been discreetely similar to few years of the stock HPE	0,83	0,78	0,17		1	1

Table 4.9: Example of generated summaries 4 of different stocks in different years with respect to HPE behaviour on all the years sorted by degree of coverage (T5).

In table 4.9 is shown the summary 4 for stock HBE sort by Degree of Coverage (T5). Degree of Coverage (T5) determines whether the conclusion made by the summary is supported by enough data. It simply normalizes the Degree of Covering (T3) onto an interval between zero and one given two values selected a priori, in our case 0,02 and 0,15. In this case we can see that ordering the resulted summaries by the Degree of Coverage (T5) results simply into showing the summaries which have a degree of covering greater than 0,15.

Summary	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**
Stock D is not similar at the 15% of the most virtuous stocks, for the EBITDA and the ROA indicator.		1	0,15	0,8	0	1
Stock O is not similar at the 15% of the most virtuous stocks, for the EBITDA and the ROA indicator.		1	0,15	0,78	0	1
Stock V is not similar at the 15% of the most virtuous stocks, for the EBITDA and the ROA indicator.		1	0,15	0,77	0	1
Stock VZ is not similar at the 15% of the most virtuous stocks, for the EBITDA and the ROA indicator.		1	0,15	0,72	0	1
Stock CAH is discreetely similar at the 15% of the most virtuous stocks, for the ROE and the Net Income indicator.		0,94	0,24	0,36	1	1

Table 4.10: Example of generated summaries 5 sorted by degree of appropriatness (T4).

In table 4.10 is shown the summary 4 for stock HBE sort by Degree of Appropriateness (T4). Degree of Appropriateness (T4): helps avoid trivial multivariate summaries. The degree's value represents how interesting and unexpected a finding in the summary may be. It can only be applied to multivariate summaries (summaries with more than one attribute) such as summary 5. We can stock D is the one that more differs from the expected finding on its behaviour with most virtuous stock for EBITDA and ROA indicators.

Chapter 5

Discussion and Conclusion

This thesis work proposes a new method of time series summarization exploiting the power of Natural Language Processing Systems, with respect to the related works, our method can rely on high automation, complex summaries generation and as depicted an accurate feature representation of the time series, although the method to generate new summaries, telling other information encoded by the the time series is yet prohibitive and requires the design of an ad hoc protoform. Also the method we proposed relies strictly on hyper-parameters, which can cause the model to produce different results for different hyper-parameters. Further works can be done on expanding the summaries type set with new summaries and to rely less on internal parameters to generate the summaries. Also a framework to compare different summaries with the used metrics can be proposed, allowing the comparison of summaries of different types over different data sets.

Bibliography

[1] Gensim. `https://radimrehurek.com/gensim/`.

[2] Matplotlib. `https://matplotlib.org/`.

[3] Numpy. `https://numpy.org/`.

[4] Pandas. `https://pandas.pydata.org/`.

[5] Reuters. `https://www.reuters.com/`.

[6] Vader. `https://github.com/cjhutto/vaderSentiment`.

[7] A. Alvarez-Alvarez and G. Trivino. Linguistic description of the human gait quality. *Engineering Applications of Artificial Intelligence*, 26(1):13–23, 2013.

[8] I. Z. Batyrshin and L. B. Sheremetov. Perception-based approach to time series data mining. *Appl. Soft Comput.*, 8(3):1211–1221, June 2008.

[9] F. E. Boran, D. Akay, and R. R. Yager. An overview of methods for linguistic summarization with fuzzy sets. *Expert Systems with Applications*, 61:356–377, 2016.

[10] C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluating the role of bleu in machine translation research. In *EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy*, pages 249–256. Association for Computational Linguistics, 2006. 11th Conference of the European Chapter of the Association for Compuational Linguistics ; Conference date: 03-04-2006 Through 07-04-2006.

[11] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

[12] E. K. Chris van der Lee and S. Wubben. Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods. 2018.

[13] P. Conde-Clemente, J. M. Alonso, Éldman O. Nunes, A. Sanchez, and G. Trivino. New types of computational perceptions: Linguistic descriptions in deforestation analysis. *Expert Systems with Applications*, 85:46–60, 2017.

[14] G. Das, K. ip Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. pages 16–22. AAAI Press, 1998.

[15] H. N. Dosilovic Filip, Brcic Mario. Explainable artificial intelligence: A survey. 2018.

[16] L. Eciolaza, M. Pereira-FariñA, and G. Trivino. Automatic linguistic reporting in driving simulation environments. *Appl. Soft Comput.*, 13(9):3956–3967, Sept. 2013.

[17] A. Gatt, F. Portet, E. Reiter, J. Hunter, S. Mahamood, W. Moncur, and S. Sripada. From data to text in the neonatal intensive care unit: Using NLG technology for decision support and information management. *AI Communications*, 22(3):153–186, 2009. Impact-F=0.449 estim. in 2012.

[18] H. M. G. R. Gautam Das, King-Ip Lin and P. Smyth. Rule discovery from time series. 1998.

[19] J. J. Harris, C. Chen, and M. J. Zaki. A framework for generating explanations from temporal personal health data. *CoRR*, abs/2003.09530, 2020.

[20] C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225, May 2014.

[21] F. Höppner. Learning temporal rules from state sequences. 01 2001.

[22] S. Ian. Computer says no: why making ais fair, accountable and transparent is crucial, 2017.

[23] J. Kacprzyk, A. Wilbik, and S. Zadrożny. Linguistic summarization of time series using quantifier driven aggregation. *Fuzzy Sets and Systems*, 159:1485–1499, 06 2008.

[24] K. Kaczmarek-Majer and O. Hryniewicz. Application of linguistic summarization methods in time series forecasting. *Information Sciences*, 478:580–594, 2019.

[25] N. Kalchbrenner and P. Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA, Oct. 2013. Association for Computational Linguistics.

[26] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. Opennmt: Open-source toolkit for neural machine translation. *CoRR*, abs/1701.02810, 2017.

[27] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[28] I. T. Lauren Sanby and M. C. Keet. Comparing the template-based approach to gf: the case of afrikaans. 2016.

[29] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents, 2014.

[30] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: A novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15:107–144, 08 2007.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space, 2013.

[32] G. Moyse and M.-J. Lesot. Linguistic summaries of locally periodic time series. *Fuzzy Sets and Systems*, 285:94–117, Feb. 2016.

[33] S. Murakami, A. Watanabe, A. Miyazawa, K. Goshima, T. Yanase, H. Takamura, and Y. Miyao. Learning to generate market comments from stock prices. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1374–1384, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[34] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. Bleu: a method for automatic evaluation of machine translation. 10 2002.

[35] E. Reiter. Nlg vs. templates. 1995.

[36] E. Reiter and R. Dale. *Frontmatter*, pages i–vi. Studies in Natural Language Processing. Cambridge University Press, 2000.

[37] D. Sanchez-Valdes, A. Alvarez-Alvarez, and G. Trivino. Dynamic linguistic descriptions of time series applied to self-track the physical activity. *Fuzzy Sets Syst.*, 285(C):162–181, Feb. 2016.

[38] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[39] C. van der Lee, E. Krahmer, and S. Wubben. Automated learning of templates for data-to-text generation: comparing rule-based, statistical and neural methods. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 35–45, Tilburg University, The Netherlands, Nov. 2018. Association for Computational Linguistics.

[40] A. Wilbik and J. Kacprzyk. On the evaluation of the linguistic summarization of temporally focused time series using a measure of informativeness. volume 5, pages 155–162, 10 2010.

[41] A. Wilbik and U. Kaymak. Linguistic summarization of processes – a research agenda. 01 2015.

[42] A. Wilbik, J. Keller, and G. Alexander. Linguistic summarization of sensor data for eldercare. pages 2595–2599, 01 2011.

[43] L. A. Zadeh. A computational approach to fuzzy quantifiers in natural languages. *Computers Mathematics with Applications*, 9(1):149–184, 1983.