Hierarchical Model for Goal Guided Summarization of Annual Financial Reports

Yash Agrawal
yash.agrawal@research.iiit.ac.in
IIIT
Hyderabad, India

Vivek Anand
vivek.a@research.iiit.ac.in
IIIT
Hyderabad, India

S Arunachalam
s_arunachalam@isb.edu
ISB
Hyderabad, India

Vasudeva Varma
vv@iiit.ac.in
IIIT
Hyderabad, India

ABSTRACT

Every year publicly listed companies file financial reports to give insights about their activities. These reports are meant for shareholders or general public to evaluate the company's health and decide whether to buy or sell stakes in the company. However, these annual financial reports tend to be long, and it is time-consuming to go through the reports for each company. We propose a Goal Guided Summarization technique through which the summary is extracted. The goal, in our case, is the decision to buy or sell company's shares. We use hierarchical neural models for achieving this goal while extracting summaries. By the means of intrinsic and extrinsic evaluation we observe that the summaries extracted by our approach can model the decision of buying and selling shares better compared to summaries extracted by other summarization techniques as well as the complete document itself. We also observe that the summary extractor model can be used to construct stock portfolios which give better returns compared to major stock index.

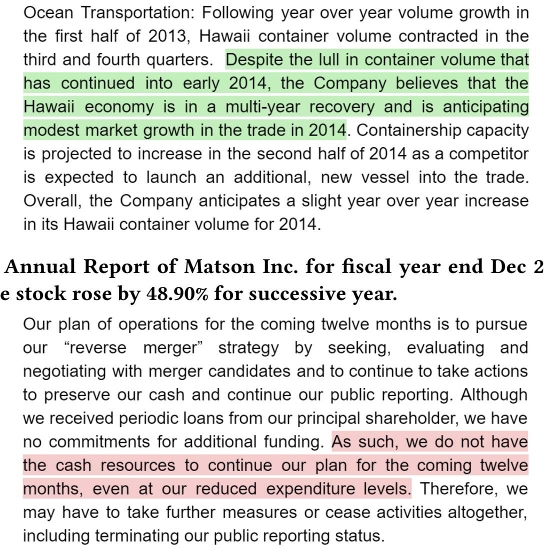
CCS CONCEPTS

- **Computing methodologies → Information extraction**.

KEYWORDS

Guided Summarization, Financial Report Summarization, Hierarchical Neural Model

ACM Reference Format:
Yash Agrawal, Vivek Anand, S Arunachalam, and Vasudeva Varma. 2021. Hierarchical Model for Goal Guided Summarization of Annual Financial Reports. In *Companion Proceedings of the Web Conference 2021 (WWW '21 Companion), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3442442.3451373



Ocean Transportation: Following year over year volume growth in the first half of 2013, Hawaii container volume contracted in the third and fourth quarters. Despite the lull in container volume that has continued into early 2014, the Company believes that the Hawaii economy is in a multi-year recovery and is anticipating modest market growth in the trade in 2014. Containership capacity is projected to increase in the second half of 2014 as a competitor is expected to launch an additional, new vessel into the trade. Overall, the Company anticipates a slight year over year increase in its Hawaii container volume for 2014.

(a) Annual Report of Matson Inc. for fiscal year end Dec 2013. The stock rose by 48.90% for successive year.

Our plan of operations for the coming twelve months is to pursue our "reverse merger" strategy by seeking, evaluating and negotiating with merger candidates and to continue to take actions to preserve our cash and continue our public reporting. Although we received periodic loans from our principal shareholder, we have no commitments for additional funding. As such, we do not have the cash resources to continue our plan for the coming twelve months, even at our reduced expenditure levels. Therefore, we may have to take further measures or cease activities altogether, including terminating our public reporting status.

(b) Annual Report of Magna-Lab Inc. for fiscal year end Feb 2013. The stock fell by 44.00% for successive year.

Figure 1: Illustration of sentences in 10-K Filing that might affect the future stock price movements.

1 INTRODUCTION

There has been advances in stock market prediction based on textual data using NLP techniques. These models use news articles [5–7] and tweets [35]. They show that text-based financial models can be useful for making short term stock movement predictions. However, these models are based on the short-term events that occur within a company and may not give a big picture of the company's health. Thus, data from news and tweets might be useful for short term gains, but in the long term, analyzing the company's overall health is important. Annual financial reports of companies comprehensively present this big picture.

Analysts read the annual reports of publicly listed companies to evaluate them. One of the prime reasons for such an evaluation is to make a call to buy or sell stakes in the company. Annual reports tend to be lengthy and thus time-consuming to read. Further, large number of publicly listed companies make it difficult for an individual to go through every company's annual report to make decisions. Therefore manual analysis of annual reports is costly both in terms of skilled labor and time.

WWW '21 Companion, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8313-4/21/04.
https://doi.org/10.1145/3442442.3451373

Additionally, there is no publicly available dataset for financial report summarization, which makes the problem challenging. Current state-of-the-art summarization models like BertSum [20] require large amounts of annotated data to train which is generally absent in such domain specific areas. We design and experiment with a different approach which extracts summary based on the end goal or use-case of the summary. We propose a Goal Guided Summarization technique for annual financial reports. Our approach is motivated by the fact that the consumers of the information want to make a call on buying or selling stakes in the company. Thus the goal of the summary is to provide information for making buy or sell decisions. This goal is used as a guide for our summarization approach.

Since the documents tend to be long, we use hierarchical neural models as document classifier. In our case the classification task is considered to be stock movement classification task. These hierarchical models give attention weights for each of the sentences while making predictions. The sentence level attention weights are used as ranks to extract summary. Note that task of stock movement prediction which is different from stock price prediction. In the former type, we are interested in predicting a binary label if the stock price will go up or down rather than predicting the actual price of the stock. Hence it is formulated as a two-class classification problem and not a regression problem.

Transformers and BERT based models are the current state-of-the-art for most NLP tasks [4, 33]. We leverage and train a hierarchical BERT based model (which we refer to as H-BERT) for the stock movement prediction task. We also experiment with the Hierarchical Attention Network (HAN) [36] for this task. We train these hierarchical models to predict the stock price movements using the Management's Discussion and Analysis (MD&A) section of 10-K annual filings. We then use the trained models to extract summaries based on the model's inference and attention weights. Figure 1 illustrates a snapshot of MD&A section from 10-K with candidate summary sentence highlighted. The 10-K filings, required by the U.S. Securities and Exchange Commission (SEC), are comprehensive reports filed annually by publicly traded companies about their financial performance. The 10-K report has 15 sections (called Items) like Business, Consolidated Financial Data, Management's Discussion and Analysis, etc. We choose the MD&A section because the company's management discusses the company's operations in detail in this section. It contains forward-looking statements that are useful for predicting giving the company's future aspects. Thus MD&A section becomes relevant for the goal of buying/selling the company's shares.

Evaluating the summary extracted by the model is challenging because there is no gold summary data in the financial domain. To address this, we create a gold standard summary of 50 randomly sampled MD&A documents from evaluation set with the help of experts. This set is used for the intrinsic assessment of the summarization system using ROUGE [19] scores. We also perform an extrinsic evaluation to better support the results [25]. We use the performance on the stock movement prediction task for extrinsically evaluating the summaries. We show that this goal can be achieved with a better accuracy using the summaries generated by our method compared to other summarization techniques. We also evaluate the reliability of the summary extractor model by constructing portfolios of stocks using the model's predictions. We compare this portfolio's returns with major stock indexes and observe that the summary extractor model gives higher returns on average, ensuring that the summaries have practical usage.

The contribution of the paper is two folds. First, we propose a Goal Guided Summarization framework for financial report summarization. Second, we show that the extracted summaries are more effective in modeling the stock movement prediction task compared to other summarization techniques using both intrinsic and extrinsic evaluation.

2 PROBLEM DESCRIPTION

Let M_{CT} be a document containing the Management Discussion and Analysis section of 10-K filing of the company C and for the fiscal year T. It contains sentences $[sent_1, sent_2, ..., sent_N]$, where $sent_i$ is the ith sentence in the document. The task is to assign a label $y_i \in \{1, 0\}$ to each sentence $sent_i$. This label indicates whether the sentence is included in the summary or not. The extracted summary gives important information to make a buy or sell decision for company $C's$ stocks to make profits in year $T + 1$.

3 RELATED WORK

The Efficient Market Hypothesis [12] states that the security prices reflect the available information and everyone has a certain degree of access to the information. There have been several arguments on this while researchers have tried different ways to predict stock prices. Attempts are made by using historical price data and using different indicators [1, 30]. Such methods do not take into account the actual events taking place within a company. To address this, researchers have exploited NLP techniques on textual data like news and events extracted from it [5–7]. Augmenting historical data with tweets is also explored for stock movement prediction [35]. Du and Tanaka-Ishii [8] introduced stock embeddings learned from news and price data to better predict the stock price. Apart from events, sentiment in the language is also explored to make stock predictions [31, 32]. Most of these prediction techniques work on short term events and does not consider the long term future plans of companies.

The use of extra information or intermediate linguistic processing has been a major part of the summarization field under the term Guided Summarization. Ng et al. [24] introduced category-specific importance (CSI) to aid sentence selection for extractive summarization. Xiong and Litman [34] explored the summarization of online reviews by using review helpfulness as a guide. Takase et al. [29] incorporated abstract meaning representation (AMR) results as additional information for the Attention-based Summarization model. Nallapati et al. [23] presented work of adapting neural encoders for extractive summarization and used lexical features like named entity and POS tags for enriching the encoder. Cao et al. [2] used existing summaries as soft templates to guide the Seq2Seq model for summary generation task. Jin et al. [15] used semantics dependency as a guide for neural abstractive summarization. Recently, large pre-trained models like BERT [4] have given better results on many NLP tasks. Liu and Lapata [20] is one of the first work to leverage these pre-trained language models for summarization task.

There has been some research in the field of unsupervised summarization as well. LexRank [11] is an unsupervised graph based approach inspired by PageRank [27] and HITS (hyperlink-induced topic search) algorithms. It uses lexical centrality based approach to find salience among sentences and eigenvector centrality in the graph representation of sentences. TextRank [22] is again inspired by the PageRank algorithm and employs a ranking scheme on the graph where each node represents a sentence in the graph. Once the algorithm converges, we get scores for each sentence in the graph from which the summary can be generated. Ozsoy et al. [26] explored the use of Latent Semantic Analysis in text summarization task. It identifies semantically important sentences in the document through matrix decomposition techniques.

NLP community has been actively working on Annual Financial Reports and other financial disclosures. There have been attempts to predict risk or stock volatility from MD&A section [17] using regression techniques. There are works for detecting omitted risk in annual financial reports [21]. Researchers have also tried forecasting event sequence from multiple 8-K reports of a firm [37]. Such studies suggest that text from financial reports and disclosures are useful to gain insights and make predictions.

Work that closely relates to what we are doing is the FNS shared task [9, 10] in an attempt to summarize annual financial reports filed in the UK. The ground-truth summaries for the task were based on extracting important sections in a financial report like Chairman's statement, CEO review, etc, via a rule based system and not human.

There have been studies that use existing summarization systems on annual financial reports [3] to study the bias present in management provided summaries. Authors conclude that automatically extracted summaries are more neutral compared to positive bias present in management provided summaries, proving the importance of the system for automatic summarization on annual financial reports.

To our knowledge, no existing work has exploited the use of stock price movements as a guide to extract a summary from an annual report. We combine the two task of stock prediction and summarization to get the important information summarized from the annual reports to help make buy/sell decisions for a company's stock.

4 DATA COLLECTION

The U.S. Securities and Exchange Commission(SEC) is an independent federal government regulatory agency that mandates publicly traded companies to report their financial performance in a comprehensive 10-K report. We obtain 10-K report filings from SEC website[1].

We collected a total of 117452 reports over a period of 1994 to 2018 from 11476 different companies. These 10-K fillings have a section called Management Discussion and Analysis which contains important forward looking statements essential in our task of stock movement prediction. These sections are also long, and summarising them can give a better image of companies future in a precise manner [17]. We write a script to filter out the MD&A sections by loosely matching strings with section headers - not all reports pass this filter. There are many reasons for this to happen; one is different reports tend to address the section in different ways. In the present work, we use only the reports which pass the filter [17]. We get 21318 MD&A sections filtered from 3776 different publicly traded companies. We use Yahoo Finance[2] to get the stock price of these companies for respective years.

Table 1: Basic statistics of the dataset

Total Number Of Documents	21318
Total Number Of Companies	3776
Avg. Documents Per Company	5.65
Avg. Sentences Per Document	290.68
Avg. Tokens Per Document	7305.23

Given the 10-K filing of a company for the fiscal year T, we compare the company's stock price for the next fiscal year $T + 1$ to capture the effect of the filing of fiscal year T. For example, we assume the 10-K filing of Apple Inc. for the fiscal year-end *September* 2018 will affect the stock price movements from *October* 2018 to *September* 2019, i.e., the next fiscal year. We compare the closing price of Apple Inc. on the first working day after the end of *September* 2018 and a year after that to get how the price has been affected w.r.t. the filing of fiscal year *September* 2018. We label the 10-K filing for the fiscal year-end T as “Buy" if the stock price is greater in fiscal year-end $T + 1$, and “Sell" otherwise. Since we are labeling the data by comparing the price difference after a year, this captures the long term effect of last year's filling on the company's stock price until the next filing is published. Table 1 gives basic stats about the complete dataset that was gathered.

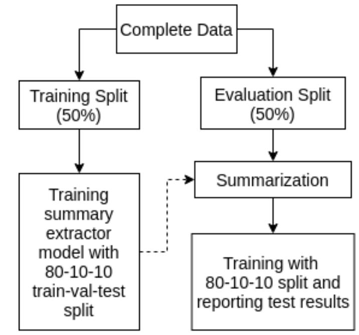

Complete Data
Training Split (50%)
Evaluation Split (50%)
Summarization
Training summary extractor model with 80-10-10 train-val-test split
Training with 80-10-10 split and reporting test results

Figure 2: Representation of data splits used for training and evaluation.

We use this dataset to train the summary extraction model and evaluate the extracted summaries. To make a fair comparison with baselines, we split the complete dataset into half. One half of the dataset is used for training the summary extraction model. The other half is used for evaluating the extracted summary as well as for comparison with different baselines. This splitting of the dataset is shown in Figure 2. This procedure enables the model to extract summaries on unseen data, thus makes a fair comparison with baselines. We take random samples over all the years and not split

[1] https://www.sec.gov/edgar.shtml

[2] https://finance.yahoo.com

the dataset temporally because we want our model to learn and also predict the different types of major events happening across the years. For example, some reports from 2008 market crash data must be present in both the train and test set so that model can learn about the crash, and can be evaluated as well.

5 MODEL

Our approach is motivated by the main goal of the summary as well as the unavailability of the summary dataset for annual financial reports. We propose the use of hierarchical neural models for encoding the MD&A section of 10-K filing to predict the stock price movements. We take the MD&A section because it contains forward looking statements which can be helpful for the task of stock movement prediction.

As the input text from MD&A section tends to be long, we explore the use of hierarchical models. Hierarchical models work at two levels. At the first level, we get the representations for each sentence in the document using an encoding mechanism. At the second level, we get the complete document representation using sentence level representations from the first level. The final document representation is used for the stock movement prediction task.

We use BERT Sentence Transformer [28] to get sentence representation from words or tokens, which is the first level in our hierarchical model. In the second level, we use a Bi-LSTM to encode the sequential information of the sentences. Attention layer is added on top of Bi-LSTM to get a complete document representation.

Consider an input as MD&A section with n number of sentences. Let each sentence be represented by $s_i, i \in [1, n]$. After passing it through Sentence BERT [28]. Each of these sentences contains L_i number of words. Let words be represented by $w_{it}, t \in [1, L_i]$ in ith sentence. Note that we use the pre-trained Sentence BERT and do not fine-tune it while training.

$$s_i = SBERT([w_{it}]), t \in [1, L_i] \tag{1}$$

In the second level, we use a hierarchical model to encode sequential information among the sentences. We use Bi-LSTM and concatenate the hidden states of both the directions to get complete hidden states corresponding to sentences.

$$\overrightarrow{h_i} = \overrightarrow{\text{LSTM}}(s_i), i \in [1, n], \tag{2}$$

$$\overleftarrow{h_i} = \overleftarrow{\text{LSTM}}(s_i), i \in [1, n], \tag{3}$$

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{4}$$

Finally we apply sentence level attention to get a single fixed length document representation.

$$u_i = \tanh(W_s h_i + b_s) \tag{5}$$

$$a_i = \frac{\exp(u_i^T u_s)}{\sum_i \exp(u_i^T u_s)} \tag{6}$$

$$d = \sum_t a_i h_i \tag{7}$$

W_s, b_s and u_s are learnable weights. The document representation d is passed though a linear layer followed by a softmax layer to get probability for stock movement prediction classification.

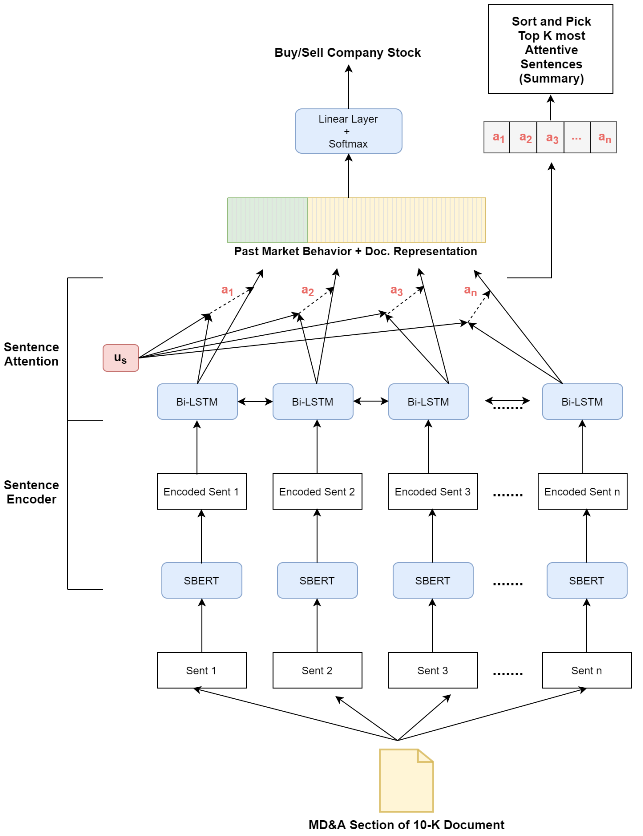


Figure 3: Architecture of H-BERT for Goal Guided Summarization of Annal Financial Reports.

$$p = softmax(W_c d + b_c) \tag{8}$$

In our case, p is a $2D$ vector for Buy and Sell classes. We use the negative log-likelihood of the correct element(P) from the vector as training loss.

$$L = -\sum_D log(P_D) \tag{9}$$

This way, the model is trained to predict stock movements, i.e., whether the stock price will go up or down given an MD&A section of the 10-K filing. This is done on the training split of the dataset discussed in Section 4. The training split is further divided into standard train, validation, and test split of 80%, 10% and 10%, respectively. The validation split is used to find a logical point to stop the training epochs, and model with the least validation loss is considered as the final trained model.

The trained model outputs the classification prediction for stock movements, and we also get the attention weights given to each sentence in MD&A section $a_i, i \in [1, N]$. We sort these attention weights and pick the sentences having the top attention weights.

The intuition is that the most attended sentences are more responsible in predicting the stock movement and thus carry important information that should be present in summary.

The model's training is done only for the stock movement prediction task, which in turn acts as a guide for our summarization task. As the goal of the summary (a concise version of the full document to help make a buy/sell decision) is also the same as the training objective, we call it Goal Guided Summarization.

We also experiment by adding the past market condition to the document representation in Equation 7. We add the returns of the S&P 500[3] Index for the last year for a given annual report by concatenating returns as single 1-D feature to the document representation. The intuition for doing so is to remove any systematic risk factors involved. For example, if there has been a market crash in the last year, there could be many negative sentences in the document. And hence, a sentence with even a slight positive orientation will be important enough to be present in summary. We expect the model to factor in the systematic events by providing the index returns.

For comparison purpose, we also experiment with Hierarchical Attention Network (HAN) [36] as a stock movement predictor. The model is similar, but instead of SBERT in Equation 1, Bi-LSTM layer with attention is used to get sentence representations. Similar to H-BERT, in HAN also, we pick the sentences with highest computed attention weights during test time to extract summaries.

6 EVALUATION

As there is no existing dataset on summarization in financial domain, we create a manually ground-truth dataset of summaries to serve as gold standard. We randomly sample 50 documents from evaluation split discussed in Section 4. Two human experts in the field of finance were given a task to pick important sentences from the documents(MD&A sections of 10-K filings) to help make investment(buy/sell) decision. Both annotators were senior researchers in the field and thus capable of the task. These gold standard summaries were used to report ROUGE [19] score to compare the methods. ROUGE scores were computed by the ROUGE 2.0, a Java package developed by Kavita Ganesan[4] [13].

It has been studied that automatic evaluation may not necessarily correlate with human judgements [18, 25]. Thus we also perform extrinsic evaluation of extracted summaries. We introduce a goal specific evaluation for the summaries. In our paper, we have assumed the goal of the summaries as being able to convey the important information to decide on buying or selling a company's shares.

Under this setting, we use the evaluation split of the dataset discussed in Section 4 to evaluate the summaries. We extract the summaries for the evaluation split using the trained model discussed in Section 5. Note that the evaluation split's data points were not included during the training time and hence are unseen by the trained model. As our model ranks the sentences according to the attention weights, we can decide the extent of summarization. We extract 10% of the top-ranked sentences to get the summary. We also take 10% summary from other ranking based baseline method [11, 22, 26] for fair comparison. We consider the stock movement labels (buy or sell) for these summaries and further split them into the train, validation, and test sets. We train HAN document classifier from the start and report the test results. The same procedure is applied with baseline extracted summaries. The intuition for doing so is to test how useful the summaries are in achieving the goal, i.e. modeling stock price movements.

After convergence, we observe the test scores of these models. We use standard Accuracy and Matthews Correlation Coefficient (MCC) as metrics for comparison. These two metrics are used by the previous stock movement prediction works [5–7, 35]. MCC is a useful measure even if there is class imbalance and considers all four categories - true positives, false negatives, true negatives, and false positives. It is given by the formula:

$$MCC = \frac{tp * tn - fp * fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}$$

7 BASELINE

BertSum [20] uses a document level encoder based on Bert [4]. BertSumExt model introduced an inter-sentence transformer layer using [CLS] output and finally a linear classification layer to classify a sentence to include or exclude from the summary. BertSumExtAbs model uses a two-stage fine-tuning in which the encoder was trained on extractive summarization task and then trained model was used to further train an encoder-decoder framework for abstractive summarization task. We include BertSumExt and BertSumExtAbs as baselines. We also include three unsupervised extractive summarization techniques as baselines.

LexRank [11] is an unsupervised graph based approach inspired by PageRank and HITS (hyperlink-induced topic search) algorithms. It uses lexical centrality approach to find salience among sentences and eigenvector centrality in graph representation of sentences.

TextRank [22] is also inspired by PageRank algorithm and employs a ranking scheme on graph where each node represents a sentence in the graph. Once the algorithm converges, we have scores for each sentence in the graph from which summary can be extracted.

LSA [26] explores the use of Latent Semantic Analysis in text summarization task. It identifies semantically important sentences in the document through matrix decomposition techniques.

8 EXPERIMENTAL SETUP

The model was implemented in Pytorch[5] framework. Due to the comparatively small size of the dataset, the model dimensions were chosen to be relatively small. For H-BERT model, SBERT gives a sentence representation of 768 dimensions. The document level Bi-LSTM hidden state dimensions was set to 20 dimensions which resulted in a document representation of 40 dimensions. Note that SBERT is not fine-tuned during the process. For HAN model, we used GloVe pre-trained embeddings of 50 dimensions. Hidden state dimension for word encoder and sentence encoder module was set as 20 and 40 dimensions respectively. This resulted in a sentence and document level representation of 40 and 80 dimensions, respectively. We used a learning rate of 0.0005 with Adam optimizer [16].

[3] S&P 500 or Standard & Poor's 500 Index is a stock market index that measures the stock performance of 500 largest publicly-traded companies in the U.S.
[4] https://github.com/kavgan/ROUGE-2.0
[5] https://pytorch.org

Table 2: Intrinsic evaluation using ROUGE- metric.

	ROUGE-1			ROUGE-2			ROUGE-L		
	P	R	F1	P	R	F1	P	R	F1
BertSumExtAbs	**37.23**	9.88	15.41	**19.53**	4.03	6.44	35.77	8.34	13.16
BertSumExt	30.93	10.54	15.60	16.86	4.08	6.36	29.60	9.21	13.63
LexRank	33.92	34.60	34.21	14.78	14.91	14.82	34.89	31.99	33.10
TextRank	31.02	35.65	33.10	12.61	14.29	13.30	31.00	32.66	31.60
LSA	32.46	33.26	32.83	12.87	13.60	13.22	28.27	34.44	30.99
HAN	35.13	36.56	35.81	19.29	15.15	16.90	35.61	30.33	32.68
H-BERT	34.62	38.83	**36.06**	17.21	**20.72**	**18.81**	**36.85**	32.98	34.32
H-BERT + Index History	34.07	**39.13**	35.97	16.97	20.53	18.26	36.01	**33.56**	**34.64**

Table 3: Test Accuracy and Matthews Correlation Coefficient(MCC) for different summarization methods along different additional features.

(a) Only Summaries

Method	Accuracy(%)	MCC
Full MD&A Section	62.32	0.2318
BertSumExtAbs	57.60	0.1268
BertSumExt	59.14	0.1604
LexRank	60.64	0.2003
TextRank	60.11	0.1945
LSA	58.20	0.1453
HAN	61.67	0.2167
H-BERT	61.86	0.2241
H-BERT + Index History	**62.42**	**0.2422**

(b) Summaries + SIC

Method	Accuracy(%)	MCC
Full MD&A Section	61.48	0.2128
BertSumExtAbs	57.22	0.1153
BertSumExt	59.23	0.1625
LexRank	61.47	0.2177
TextRank	61.29	0.2110
LSA	60.45	0.1904
HAN	62.04	0.2276
H-BERT	**62.89**	**0.2579**
H-BERT + Index History	62.79	0.2568

(c) Summaries + Index

Method	Accuracy(%)	MCC
Full MD&A Section	62.70	0.2395
BertSumExtAbs	59.00	0.1622
BertSumExt	60.82	0.1988
LexRank	62.79	0.2468
TextRank	62.23	0.2336
LSA	60.26	0.2029
HAN	64.20	0.2714
H-BERT	**65.23**	**0.2924**
H-BERT + Index History	63.92	0.2756

(d) Summaries + SIC + Index

Method	Accuracy(%)	MCC
Full MD&A Section	63.82	0.2680
BertSumExtAbs	58.63	0.1506
BertSumExt	60.54	0.1949
LexRank	61.57	0.2202
TextRank	61.89	0.2342
LSA	61.14	0.2309
HAN	62.98	0.2467
H-BERT	**63.82**	**0.2728**
H-BERT + Index History	62.51	0.2450

Note that for HAN model, same setting was used for training the summary extraction module as well as for evaluation(Section 6). For all cases, we use a validation set to decide an early stopping point while training. We take the model that gives the least validation loss to be the best model.

9 RESULTS AND DISCUSSION

9.1 Intrinsic Evaluation

Table 2 gives the results for ROUGE-1, ROUGE-2 and ROUGE-L metrics. The scores are computed as average over 50 human extracted gold summaries. It is observed that the supervised summarization methods like BertSumExtAbs and BertSumExt give good precision score but low recall. This is because of the fact that they are trained on CNN/DailyMail [14] dataset. These datasets are small and have summaries with an average of around 3.59 to 3.86 sentences [20]. Because of this fact the models are trained to extract very few sentences and thus result in poor recall score. Precision is also boosted because of the same reason that few sentences are extracted as summaries.

We observe that H-BERT, with and without Index History gives comparable results. This could be because of the fact that the Index History feature is useful to improve the stock movement prediction capabilities (Section 9.3) but does not have much effect on the sentence it pays attention to.

H-BERT along with HAN gives better results compared to other summarization techniques indicating that the sentences that are being paid more attention to are good summary candidates.

9.2 Extrinsic Evaluation

Table 3 compares the results of the proposed method of summarization with other baseline summarization methods. Results are computed as per the evaluation methodology given in Section 6.

Accuracy and MCC on the test set for the stock movement prediction task are shown when summaries using different techniques are used as input. Results for the full MD&A section is also added for the reference to compare with summarization techniques. We can see that summaries, which are only 10% of full MD&A section data, can model the stock movement task with similar or even better results meaning that the model parameters are learned in a way that gives us a useful summary for this task. Table 3a gives results when only summaries (no extra features) are used for stock movement prediction task. We observe that summaries extracted by H-BERT with and without Index History are better able to model the task of stock movement prediction with respect to other baseline methods.

In addition to the summary, there could be other factors that drive the prediction task, such as the sector (industry) or index[6] in which the company belongs. To study this phenomenon we include different features along with summaries to train the model for stock movement prediction task and observe results. We append extra feature with the document representation vector d (Equation (7)) to classify stock movement prediction. This allows us to evaluate how adding extra information with the summaries can be useful in the prediction task.

In Table 3b, we test the effect of the sector by adding SIC code[7] features with summaries. As 414 unique SIC codes accommodate companies in our dataset, we concatenate a 414-dimensional one-hot-vector to document vector d. Here we see improvements in results in general. This indicates that given a summary and sector, model is better able to predict stock movements.

A similar analysis is done by concatenating a 2-dimensional one-hot-vector to document vector d to capture the effect when a company belongs to an index or not. We consider S&P 1500 [8] stocks to be index companies and all others as non-index companies. The reason for using this information is that index companies tend to be more stable and less risky in general, which could be helpful in stock movement prediction. Table 3c shows results for this case. We observe that the proposed method performs the best in this case, meaning that the model trained on only the summaries extracted by the proposed method gives an underrated or overrated view of companies. So adding an index feature might have normalized this bias towards neutrality. Finally, in Table 3d, we show results by concatenating both SIC and Index features vectors along with summary features. The results fell in this case, which could be because additional information may be introducing noise and deteriorating the patterns present in the data, thus compromising evaluation scores.

The best result in Table 3 is given by the H-BERT and some cases H-BERT with Index History. This suggests that by considering summaries extracted by these models can lead to better predictability of stock movement. Moreover it gives similar or better prediction capabilities compared to full MD&A section. Which implies that similar or better decisions can be made by consuming less information(summaries) compared to full document.

Table 4: Comparison of CAGR and Average returns with benchmark indexes and portfolio constructed by summary extractor model.

Asset Class	CAGR	Average Returns
S&P 100	6.94%	8.72%
S&P 500	7.15%	8.86%
Russell 1000	7.29%	9.03%
DJIA	7.63%	9.05%
HAN (P = 10)	9.41%	11.38%
HAN (P = 25)	11.75%	13.64%
HAN (P = 35)	11.07%	12.93%
HAN (P = 50)	11.12%	12.79%
H-BERT (P = 10)	11.32%	12.71%
H-BERT (P = 25)	14.06%	15.18%
H-BERT (P = 35)	13.25%	14.77%
H-BERT (P = 50)	12.27%	13.74%
H-BERT + Index Hist. (P = 10)	15.61%	17.24%
H-BERT + Index Hist. (P = 25)	15.05%	16.34%
H-BERT + Index Hist. (P = 35)	14.22%	15.60%
H-BERT + Index Hist. (P = 50)	12.93%	14.20%

9.3 Model Prediction and Reliability Analysis

We have considered the attention weights of the model to extract summaries but have not analyzed actual predictions made by the model. In this section, we evaluate the summary extractor model's prediction capabilities by constructing a portfolio of stocks. We compare the returns of this constructed portfolio with traditional stock market indexes over a span of 25 (1994 to 2018) years. We consider the complete evaluation split of the data and sort the prediction probabilities given by the trained model for each year. We pick the top P most probable buy label by the model to construct an equal-weighted portfolio of all the stocks. We compare the CAGR[9] and Average Returns of constructed portfolio with some major indexes in the US - S&P 100, S&P 500, Russell 1000 and DJIA . From Table 4, we observe that the portfolio constructed by the summary extractor model gives better returns compared to various market indexes in the long run. This implies that the predictions made by summary extractor model and the corresponding attention weights are reliable.

10 CONCLUSION AND FUTURE WORK

In this paper, we address the problem of information overload in the financial domain. We propose a Goal Guided Summarization framework for annual financial reports. The summarization task is guided by the goal of making buy or sell decisions of the company's stocks. We accomplish this task by training hierarchical neural models to predict stock movements and extract the summary by ranking sentences using the attention weights. By using intrinsic and extrinsic evaluation, we observe that the summaries extracted by the proposed method can help analysts in decision making. It

[6] A market index is a hypothetical portfolio of investment holdings that represents a segment of the financial market.

[7] The Standard Industrial Classification (SIC) are four-digit codes that categorize companies into various industries (sectors) w.r.t. their business activities.

[8] The S&P 1500 or Standard & Poor's 1500 Index is a market-capitalization-weighted index of the 1500 largest publicly-traded companies in the U.S. covering 90% of the market capitalization.

[9] Compound annual growth rate (CAGR) is the rate of return that would be required for an investment to grow from its beginning balance to its ending balance, assuming the profits were reinvested at the end of each year of the investment's lifespan.

also achieves the goal of stock movement prediction better than existing summarization techniques and the full document itself.

This summarization framework can be useful in different domains where the annotated dataset is not available, but the purpose of summary consumption can be defined. Further, we also conclude that the trained summary extractor model can be used for portfolio construction, which generated better returns than benchmark market indexes implying that the model and extracted summaries are reliable and are practical. There are a few other things that can be explored as future works. (1) Proposed method extracts summaries w.r.t. analyst or investor. Similarly, summaries w.r.t. other stakeholders like legal teams and managers can be extracted by designing appropriate goals. (2) We can also examine other items in the 10-K report like Risk Factors and Legal Proceedings along with aggregated news articles and global events related to companies over a specific window to extract the goal specific summaries.

REFERENCES

[1] Torben G Andersen and Tim Bollerslev. 1997. Intraday periodicity and volatility persistence in financial markets. *Journal of empirical finance* 4, 2-3 (1997), 115–158.

[2] Ziqiang Cao, Wenjie Li, Sujian Li, and Furu Wei. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 152–161.

[3] Eddy Cardinaels, Stephan Hollander, and Brian J White. 2018. Automatic summaries of earnings releases: Attributes and effects on investors' judgments. *Available at SSRN 2904384* (2018).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[5] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1415–1425.

[6] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

[7] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2016. Knowledge-driven event embedding for stock prediction. In *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*. 2133–2142.

[8] Xin Du and Kumiko Tanaka-Ishii. 2020. Stock Embeddings Acquired from News Articles and Price History, and an Application to Portfolio Optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3353–3363.

[9] Mahmoud El-Haj. 2019. MultiLing 2019: Financial Narrative Summarisation. In *Proceedings of the Workshop MultiLing 2019: Summarization Across Languages, Genres and Sources*. 6–10.

[10] Mahmoud El-Haj, Ahmed AbuRa'ed, Marina Litvak, Nikiforos Pittaras, and George Giannakopoulos. 2020. The Financial Narrative Summarisation Shared Task (FNS 2020). In *Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation*. COLING, Barcelona, Spain (Online), 1–12. https://www.aclweb.org/anthology/2020.fnp-1.1

[11] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479.

[12] Eugene F Fama. 1965. The behavior of stock-market prices. *The journal of Business* 38, 1 (1965), 34–105.

[13] Kavita Ganesan. 2015. ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks. (2015).

[14] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 1693–1701.

[15] Hanqi Jin, Tianming Wang, and Xiaojun Wan. 2020. SemSUM: Semantic Dependency Guided Neural Abstractive Summarization.. In *AAAI*. 8026–8033.

[16] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG]

[17] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. 2009. Predicting Risk from Financial Reports with Regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Boulder, Colorado, 272–280. https://www.aclweb.org/anthology/N09-1031

[18] Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural Text Summarization: A Critical Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 540–551. https://doi.org/10.18653/v1/D19-1051

[19] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[20] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 3730–3740. https://doi.org/10.18653/v1/D19-1387

[21] Corentin Masson and Syrielle Montariol. 2020. Detecting Omissions of Risk Factors in Company Annual Reports. In *Proceedings of the Second Workshop on Financial Technology and Natural Language Processing*. -, Kyoto, Japan, 15–21. https://www.aclweb.org/anthology/2020.finnlp-1.3

[22] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[23] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[24] Jun Ping Ng, Praveen Bysani, Ziheng Lin, Min-Yen Kan, and Chew Lim Tan. 2012. Exploiting category-specific information for multi-document summarization. In *Proceedings of COLING 2012*. 2093–2108.

[25] Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why We Need New Evaluation Metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2241–2252. https://doi.org/10.18653/v1/D17-1238

[26] Makbule Gulcin Ozsoy, Ferda Nur Alpaslan, and Ilyas Cicekli. 2011. Text summarization using Latent Semantic Analysis. *Journal of Information Science* 37, 4 (2011), 405–417. https://doi.org/10.1177/0165551511408848

[27] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank citation ranking: Bringing order to the web.* Technical Report. Stanford InfoLab.

[28] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[29] Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*. 1054–1059.

[30] Stephen J Taylor. 2008. *Modelling financial time series.* world scientific.

[31] Paul C Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of finance* 62, 3 (2007), 1139–1168.

[32] Paul C Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. 2008. More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance* 63, 3 (2008), 1437–1467.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[34] Wenting Xiong and Diane Litman. 2014. Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews. In *Proceedings of coling 2014, the 25th international conference on computational linguistics: Technical papers*. 1985–1995.

[35] Yumo Xu and Shay B. Cohen. 2018. Stock Movement Prediction from Tweets and Historical Prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1970–1979.

[36] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 1480–1489. https://www.aclweb.org/anthology/N16-1174

[37] Shuang (Sophie) Zhai and Zhu (Drew) Zhang. 2019. Forecasting Firm Material Events from 8-K Reports. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*. Association for Computational Linguistics, Hong Kong, 22–30. https://doi.org/10.18653/v1/D19-5104