ANNUAL REPORT ANALYSIS OF IBM FOR YEAR (2022) USING NLP

1st Anthoni Sagarika Pamisetty 2nd Darin Johnson
Data Science, University of Maryland Baltimore County, Maryland, United States
anthonp1@umbc.edu, darinj@umbc.edu

***Abstract*—Everyone is now interested in what to do with stocks,investing in crypto and the entire internet is full with financial information about companies available to investors. Many tries and attempts were made to gather the information from emerging news,newspaper,online resources. These texts contain valuable information in a more complex structure and vocabulary than a regular article. The aim of this paper is to make use of Natural language processing and various methods of text and data mining in order to extract relevant informa- tion about changes in the company(IBM) here, such as peo- ple, new products or stores, from its annual re- ports as well as extracting the general sentiment of these texts. When evaluated on IBM yearly report(2022), all of these techniques revealed insightful and worthwhile data that was hidden from plain sight. This system could serve as an excellent foundation for a more robust and feature-rich investment program. This project is a component of the University of Maryland Baltimore County Master's in Professional Studies in Data Science,Natural language processing Course**

***Index Terms*—NLP,data mining,information,annual reports**

I. INTRODUCTION

The trending aspect of data science involves analyzing data and producing precise insights. Businesses are currently focusing more of their efforts on extracting textual information; this interest has been growing over the past few years. In the finance industry, quantitative analysis is insufficient to draw conclusions. The primary goal of this paper is to assist investors in locating accurate information from annual reports so they can make the right decisions.

Even though stocks are the major topic of the paper, I want to concentrate mostly on the analysis and findings to tell investors if it is safe to invest or not. I will also conclude at the end of the paper as to whether the stocks performed well or poorly by 2023.Here, my decision is driven by my personal interests in IBM.I'm going to examine the annual report for the most recent year, 2022.

The primary goals of this article are to determine the attitudes expressed in the annual reports—whether they are good, negative, or neutral.The second goal is to employ sophisticated pre-trained models for auto summarization; the eventual objective is to use Named Entity Recognition to create a chatbot that responds to inquiries about annual reports (IBM).The format of the paper is as follows. Section 2 provides an overview of earlier research in this field. Section 3 outlines the approach that was looked into. Section F reports the experimental results; sections 4 and 5 provide conclusions and recommendations for future work.

II. LITERATURE REVIEW

The concept put forth here is not new; in fact, it has been researched for over 50 years from a variety of angles. According to a chronological timeline, the initial research that indicated interest in examining annual reports or other financial publications was on determining the veracity of these businesses that might be attempting to conceal their records using cryptic terminology. The transparency and honesty of the reports have been the subject of numerous studies. Henry[2], among others, found that these texts demonstrated a high reading level comparable to the more technically written financial statements [2] and that the reports' readability did not seem to significantly improve over time [3].
Wang [4] introduced NewsCATS to trade

immediately when news stories are published. Next, classify news articles using the normal three-tag method (good, neutral, and poor) in classification problems.Then, associate each article with fluctuations in stock index.

Of course, advancements in Natural Language Processing and Machine Learning quickly outgrew financial predictions. Butler et al. combined Word N-grams and Readability functions into a bag-ofwords strategy that reached an average accuracy of 69%. Falinouss focused on prediction of intraday stock prices with vector space modeling, support vector machines, and the tfidf term weighting scheme. They notably reached an accuracy of 83%.

Sentiment analysis in financial news is arguably the best studied approach in this topic [7, 8, 9, 10]. It attempts to comprehend how news affects investors' decisions; regrettably, it is impossible to predict with precision how traders will respond to material published in newspaper columns. As The Wolf of Wall Street once said, "It's too late when you see it on the Wall Street Journal." "NegExpander" produced the best Kappa values for all of the experiments Turegun. [7] conducted on negative detection in sentiment analysis. They conclude that "regular expression and syntactic processing based algorithms have bet-ter agreements with human reviewers than the classification-based algorithms."

Financial reports have also been grouped for quantitative analysis, such as in the works of Wang [4] and Kloptchenko [6], with the current effort focusing only on the forecast of stock prices. Here, the reports were visualized by the authors using collocational networks and prototype-matching text clustering. enhanced with self-organizing maps later in [14]. The textual portion of the reports provides some insight into the company's financial performance. The groupings derived from the qualitative and quantitative analyses (historical performance) did not align in each instance.

Lastly, I would like to thank Turegun [7] for his work, which included a comprehensive collection of earlier studies in this area and a solid introduction to financial text mining.

III. Methodology and Process

This section has the techniques and steps followed. Everything has been coded using Python3 and also I used Google colab pro so as to handle pre-trained models, one can use regular vs code as well but one have to check the system specifications as well.

A. *Pre-processing*

Since all of the data is freely accessible in PDF format, the first thing that needs to be done is convert everything to plain text. This conversion may result in textual inconsistencies and, frequently, the loss of data, including tables, figures, and other graphic representations. Fortunately, there are aesthetic standards that an annual report must adhere to in order to prevent garish and superfluous representations.To do this I used PyPDF2 to convert the pdf file to text, also I used pycryptodome to get the pdf file converted.I took the IBM file 2022 annual report from IBM website itself.

Later I did proceed doing the basic pre-processing steps in NLP.This pre-processing step is necessary for any NLP application in order to prepare the data—text in this case—that will be used in the future. The procedures required to carry out this preparation are explained in this section. For these kinds of projects, the processes are planned in a standard pipeline that goes like this: normalization $\rightarrow$ tokenization $\rightarrow$ sentence duplication $\rightarrow$ POS tagging.

Mounting Google Drive: Use google.colab to access files in a user's Google Drive.
Text Extraction: The PdfReader is implemented from the PyPDF2 library to read and extract text from every page of the PDF files.
Tokenization: That will help in chopping the extracted text into individual words using the wordtokenize from the NLTK library.
Stopword removal: Filtering using NLTK's predefined common English stopwords list so that the analysis will be focused on meaningful words.
The Word Frequency Analysis: It uses the Counter class from the collections library to count instances of each word for insight into the most common

terms mentioned in the documents.

Stopword removal: Filtering common English stopwords with the predefined list in NLTK to keep focused on the word.
Word Frequency Analysis: This will be done using the Counter class in Python to perform word frequency analysis and identify the most common words in the report.

I have used regex operations to remove special characters in the PDF.

B. Post-Preprocessing

We move now to the current project's post-processing phase, which hinges on three different applications that are independent from each other: Sentiment Analysis, Auto-summarization, and Name Entity Recognition(CHATBOT) The data is ready for use in operations and analysis. The selected multi-task pre-trained CNN to perform the NER classification was "en core web sm".

C. SENTIMENTAL ANALYSIS

Numerous attempts have been made to analyze sentiment in financial literature, as mentioned in section 2. I decided to use the bot-tom's strategy and divide statements into three groups: positive, neutral, and negative. The initial stage was gathering pre-labeled text that fit these categories; for this, I used a little, off-topic dataset made up of just six sentences, and I verified that the finbert was operating as intended. The model did a fantastic job, producing positive, negative, and neutral phrases.Next, I trained the dataset independently using various popular filters, which produced the graph shown in the image.
Following are the reasons why I chose finbert.

- Contextual Understanding: A BERT (Bidirectional Encoder Representations from Transformers) version, FinBERT fully captures contextual links between sentences. It is essential for effectively analyzing financial records when context is critical because it may pick up on minute details and mood swings.
- FinBERT's design provides for the ability to fine-tune on particular tasks, allowing for additional flexibility and optimization for sentiment analysis in financial domains. FinBERT can adjust its representations to better capture the nuances of financial language by fine-tuning on datasets relevant to a given domain.
- State-of-the-Art Performance: When applied to financial writings, extensive empirical evaluations have shown that FinBERT performs better than generic sentiment analysis models. Higher accuracy and effectiveness in recording financial emotion are attributed to its domain-specific pre-training and fine-tuning capabilities.

D. AUTO SUMMARIZATION

To perform auto summarization i went with using the pretrained model where I have used large facebook cnn model, I used hugging face and transforms to do such,I have created a pipeline that helped for auto summary.I have created a function capitalize that performed the following operations.

- I have split-ted the sentences and I capitalized the first character in the each word.I have joined them accordingly
- I decided to apply the pre-trained model on these sentences,i chose the maximum summary length as 100.I have loaded the model and tokenizer.I used encode to generate the summary.I have later decoded and generated the summary.
- Since the length for facebook bart is 1024,I had to choose the maximum input length as 1024.Later on I have joined the sentences thus generated from 1 to 1024.If the input length is more than 1024 then truncate it.
- I later on called the function to generate the summary.

E. NAME ENTITY RECOGNITION

Through the usage of a pre-trained model named "deepset/roberta-base-squad2" and the tokenizer that goes along with it, the code establishes a reliable framework for handling user inquiries and producing precise answers. The model selection is important since it is ideally suited for this application because it has been trained specifically for answering questions. Once the model and tokenizer are loaded, the code creates a question-answering pipeline that expedites the process of obtaining pertinent data from user inquiries and delivering succinct responses.

In actuality, the code asks users to enter questions while running in a nice while loop. Users can ask questions about a variety of topics with ease thanks to this interactive interface. When the code receives a question, it creates an input object with the user's question and any pertinent context. The model uses this background as a backdrop to find relevant data in order to provide a response. The model's capacity to comprehend the context and derive pertinent insights facilitates the generation of precise responses, hence augmenting the user experience through informative and significant responses.

The iterative process by which the code operates guarantees continuous interaction, allowing users to keep asking questions until they decide to end the session by entering 'q'. This adaptability increases the usefulness of the question-answering system by enabling users to investigate different subjects and get timely responses. Overall, the code provides a user-friendly interface for obtaining meaningful responses to a variety of queries, demonstrating the practical implementation of pre-trained language models in natural language understanding tasks. The code demonstrates how AI-powered question-answering systems may revolutionize information retrieval and improve user experiences by seamlessly integrating cutting-edge natural language processing algorithms.

F. Experimentation and Results

The tests were carried out at IBM a firm listed on the SP 500 stock exchange.This company was chosen because it piques my curiosity and because it has recently made strides in the fields of machine learning and artificial intelligence, which makes it a perfect fit for the scope of my job.

G. Sentimental analysis

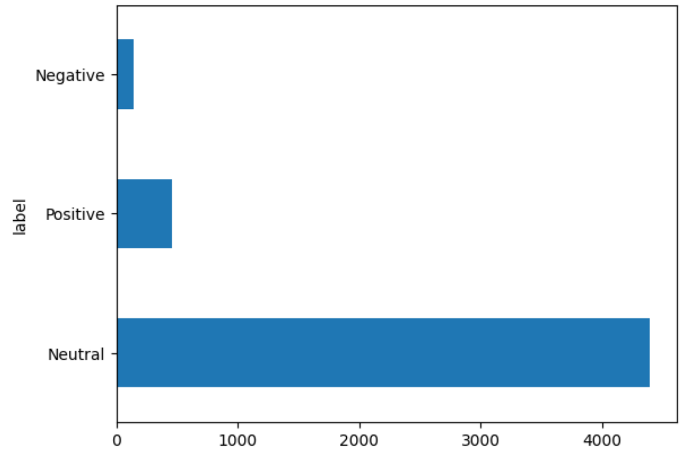

Fig. 1. SENTIMENTAL TONES FOR THE ANNUAL REPORT

From the above fig 1 we can say the tone which is neutral is higher since any writer who writes the annual reports writes them in a neutral,thus we can say the tone in IBM annual report for year 2022 is neutral,the negative tones are also almost 500 which may includes some keywords like downfall,loss etc,we must notice the negative tone in this case clearly.

H. Summarization

As mentioned in the section D above the summary is limited to 1024.I had to choose this because of the facebook model.Choosing other models can also help a lot.Following words are the output sentences

1) output-autosummary: Summary: Ibm is facing a series of specific business challenges: inflation,..supply chain disruption, tight labor markets, sustainability, and an ever-evolving cybersecurity threat..

From the above generated summary we can conclude that IBM faced several business challenges in the year 2022, to analyze this I went deep down to conclude using the stocks and its information.I felt doing a current ratio analysis helps in this case.

The current ratio is around 1 between 2022-2023. Current assets seem to be not very strong in relation to current liabilities. This is simply because, A current ratio of less than 1 means that the company's current liabilities exceed its current assets, suggesting that it may have difficulties meeting its short-term financial obligations. However, in the first quarter of 2023, the current ratio looked recovering from low point. This may be heralding a good current ratio in the future.

So, the current ratio analysis was not good that year, hence we can say the summary was so apt and best to my knowledge.

I. Q & A chatbot

This is a regular Q & A bot that is been built out of my interest,the outputs can be seen in the figure below.

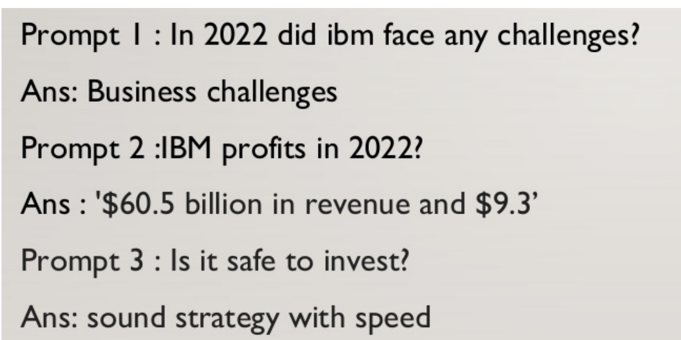

Fig. 2. Questions and Answers

From the above Q & A section we can conclude IBM saw both profits and loss in the calendar year 2023.

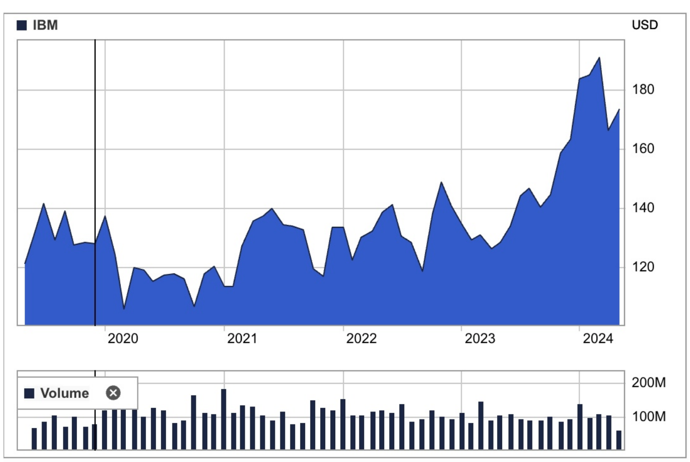

Fig. 3. IBM Stocks during year 2022

Summary

We can conclude that IBM faced several challenges during the year 2022-2023,using Sentimental analysis, analysis was done that most of the tones is neutral,I did use Autosummarization to get the summary where it mostly stated that may downfalls were faced by IBM, we then compared with some financial methods to prove it right.Using chatbot analysis also helped us in analysis the profits and loses faced by IBM.With this we can conclude that IBM faced pretty much challenges and saw a good growth during the start of financial year 2023.

IV. Future Work

This endeavor has been highly illuminating and of great personal interest. I would like to keep developing and finishing this project by increasing 10 more years of IBM to include a more in-depth research and further exploration of the potential applications of such a system. This study, in particular, still has a lot of potential for improvement, especially in the sentiment analysis section, where a variety of sentiments instead of just the three tags and negation handling utilized in the classification might be used.Additionally, it may combine several company-related materials to obtain the viewpoints of journalists and publications as well as the company's intended message for investors and customers.

References

1) Tetlock, P. C., Saar-Tsechansky, M., & Macskassy, S. (2008). More than words: Quantifying language to measure firms' fundamentals. *The Journal of Finance*, 63(3), 1437–1467.
2) Henry, E., & Leone, A. J. (2016). Measuring qualitative information in capital markets research: Comparison of alternative methodologies to measure disclosure tone. *The Accounting Review*, 91(1), 153–178.
3) Goryachev, S., Sordo, M., Zeng, Q. T., & Ngo, L. (2006). Implementation and evaluation of four different methods of negation detection. Technical report, DSG.
4) Wang, B., Huang, H., Wang, X., & Chen, W. (2009). An ontology-based NLP approach to semantic annotation of annual report. In *2009 International Conference on*

Computational Intelligence and Security (pp. 180–183). IEEE.

5) Kloptchenko, A., Magnusson, C., Back, B., Vanharanta, H., & Visa, A. (2002). Mining textual contents of quarterly reports. *Turku Center for Computer Science Technical Reports*.
6) Kloptchenko, A., Eklund, T., Karlsson, J., Back, B., Vanharanta, H., & Visa, A. (2004). Combining data and text mining techniques for analyzing financial reports. *Intelligent Systems in Accounting, Finance & Management: International Journal*, 12(1), 29–41.
7) Turegun, N. (2019). Text mining in financial information. *Curr. Anal. Econ. Finance*, 1, 18–26.
8) Hassan, T., & Baumgartner, R. (2006). Using graph matching techniques to wrap data from PDF documents. In *Proceedings of the 15th International Conference on World Wide Web* (pp. 901–902).
9) Oro, E., & Ruffolo, M. (2008). Xonto: An ontology-based system for semantic information extraction from PDF documents. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence* (pp. 118–125). IEEE.
10) Hassler, M., & Fliedl, G. (2006). Text preparation through extended tokenization. *WIT Transactions on Information and Communication Technologies*, 37.