Extracting Key Insights from Financial Text: A Summarization Approach Project Update I

Parisha Desai
Computer Science Department
Purdue University Fort Wayne
2101E Coliseum Blvd,
Fort Wayne, IN 46805
desap01@pfw.edu

Manvitha Chowdari Gottipati
Computer Science Department
Purdue University Fort Wayne
2101E Coliseum Blvd,
Fort Wayne, IN 46805
gottm01@pfw.edu

1 Introduction

In today's fast-paced financial world, extracting key insights from extensive textual sources like financial reports, news articles, and disclosures is critical for informed decision-making. Financial institutions, investors, and rating agencies depend on this data to assess risks and maximize returns, but manually processing large volumes of information is both time-consuming and error prone.

Natural Language Processing (NLP), specifically text summarization, offers a solution by creating concise summaries that capture essential details, allowing users to quickly grasp key information without reviewing the entire document. This project explores summarization techniques aimed at condensing financial reports while preserving the core message and critical details. The goal of this report is to outline the progress of the implementation thus far, challenges faced, and the next steps.

2 Our Approach and Implementation Methodology

We initially started with the dataset for Financial News Summarization, which is available on HuggingFace: ragha92/FNS_Summarization.

We understood and analyzed the dataset to identify the kind of text preprocessing and normalization required. The dataset consists of 3 different files: one for training, one for validation, and one for testing.

1. **Dataset Overview**: Table 1 captures the key important parameters describing the dataset and its complexities. Each dataset consists of 2 columns: 'Annual Reports' and 'Gold Summaries'

2. **Text Preprocessing**: As part of the text processing, we have followed removing white space and any extra trailing space. We have not removed numbers as they would be a key component of our summarization task. We have also not removed short words, numbers, stop words or lowercased the characters as this being financial dataset we wanted to preserve the integrity and meaning of the text which would otherwise be lost if this text processing is applied.

3. **Summarization Methodologies**: We considered two primary methodologies for summarization:

1. **Extractive Summarization:** This methodology involves selecting and extracting key sentences directly from the original text to create a summary. It works by identifying the most important parts of a document and combining them to form a concise version. A few examples of models using this summarization methodology are TextRank and TF-IDF.

2. **Abstractive Summarization:** This methodology involves generating new sentences that capture the essence of the original content rather than directly extracting from the source. It uses Natural Language Processing (NLP) techniques and deep learning models to interpret the underlying meaning of the text and then rephrase it in a condensed form. Transformer-based models like T5, BART, PEGASUS, and BERT fall under this category.

4. **Model Selection and Implementation**:
To begin our analysis, we selected representative models from both extractive and abstractive methodologies

1. Under Abstractive summarization we identified using TextRank (unsupervised learning),

2. Under Extractive summarization we selected using T5 transformer model (supervised learning) and BART (supervised learning).

Implementing these models, we are creating techniques for text summarization on financial

Dataset Type	Number of Records	Avg word length of Annual report of records	Avg word length of Gold Summaries of records
Train	2057	40734	1016
Validation	257	41065	1124
Test	259	39980	1025

Table 1: Dataset Overview

dataset such that the models can accurately generate the text summarization by improvising the model to generate text that would closely match the Gold Summaries.

1. **TextRank Implementation**: TextRank is used essentially for unsupervised learning and is graph based ranking algorithm wherein each node in our implementation is a sentence and the edge would represent similarity between sentences based on content overlap.

   Since this is an unsupervised learning algorithm the training data is not processed, and the model summarization ability is computed using validation and test data. The importance of each sentence is computed using pyTextRank. Initially we started with ranking the top 3 sentences from text and basis these sentences a summary is generated.

   The generated summary and gold summaries for validation and test data are then compared to derive the Rouge score for unigram, brigram and entire length of the sentences. The BLEU is also computed to get a view of the performance of the model. The average of the metrics is derived from the value computed for each text to arrive at the performance of the model.

2. **T5 (Text to Text transfer Transformer) model Implementation**: A pretrained T5 transformer model is used to summarize the dataset, and the generated summary is then compared with the gold summary.

   The T5 model is first finetuned using the train data where the cross-entropy loss function is used for text generation and the Adam optimizer is used to update the weights. The goal was to minimize the loss during text generation across 5 epochs. In each epoch, the training data was divided into batches to improve learning efficiency.

   The T5 model initially converts the input article into token IDs that are processed by model; a summary is generated using token IDs. The token IDs generated are then converted back to human readable text. This step is performed first for validation and then for test dataset to compute the Rogue and BLEU score. While computing the Rogue and BLEU score the summary generated by the model and the gold summary of the validation and test data is compared to compute scores for respective dataset.

3. **BART Model implementation**: Similar to T5, the BART model is used to summarize the dataset and compare the generated summaries with gold summaries. The BART model, which is an autoencoder for pretraining sequence-to-sequence models, is fine-tuned on the training data. The cross-entropy loss function is employed to optimize text generation, and the Adam optimizer is used to update the model weights, minimizing the loss over 3 epochs. During each epoch, the model processes batches from the training dataset to improve its summarization ability.

   The BART model first converts the input text into token IDs, which are passed through the encoder-decoder architecture. The decoder generates a summary from the token IDs, which are then transformed back into human-readable text. This process is applied to both the validation and test datasets. The generated summaries are then evaluated by comparing them to the gold summaries, and the Rouge and BLEU scores are calculated to measure the model's performance.

   While computing these scores, the model's generated summary and the gold summary from the validation and test datasets are compared to evaluate its ability to capture key details accurately.

5. **Hyperparameter Analysis and Execution Time**: The table 2 below summarizes the hyperparameters used for each model and the corresponding execution times. This analysis is critical for our next steps, as it will guide us in hyperparameter tuning to improve model performance.

1. The execution time for TextRank model was 2 hours, which is efficient given the simplicity of the algorithm and the limited number of parameters to tune.

2. Both T5 and BART models have more extensive hyperparameter tuning requirements due to their complex architecture, which would have led to longer execution times.

3. The T5 model had the longest execution time (3 hours), which suggests that its settings, such as the number of epochs and beam search, may require optimization to improve efficiency while maintaining performance.

4. The BART model was the most efficient in terms of execution time at 1 hour 30 minutes, likely due to fewer epochs (3) compared to T5 (5 epochs)

5. The hyperparameter like the use of beam search and length penalties in T5 and BART, have been included to focus on generating high-quality summaries, although they add to increased computational demands.

Implementation of our code available on Github:Text-Summarization-Financial-Text

3 Current Results

Below are the key metrics we used for evaluating the performance of our models. These scores help assess the effectiveness of each model in capturing key information and generating coherent summaries.

1. **ROUGE-1 Score**: measures the overlap of unigrams (individual words) between the generated summary and the reference (gold) summary. It helps evaluate how well the summary captures essential words from the original content.

2. **ROUGE-2 Score**: evaluates overlap of bigrams (two consecutive words) between the generated and reference summaries. This metric helps provides insight into how well the model captures word pairs and maintains context between adjacent words.

3. **ROUGE-L Score**: measures the longest common subsequence (LCS) between the generated summary and the reference summary, focusing on the sequence of words that would help maintain the logical flow of information.

4. **BLEU Score**: evaluates the accuracy of a generated summary by comparing its n-grams (up to 4-grams) with the reference summary. It penalizes overly short or repetitive summaries and measures how well the generated summary matches the reference in terms of fluency and relevance.

   Table 3 captures the evaluation results

4 Current Analysis

- The highest ROUGE-1 score was achieved by the TextRank model (0.1980 on the test set), indicating its effectiveness in capturing key unigrams.

- The T5 model recorded the lowest ROUGE-1 score (0.1583), suggesting it struggled to identify significant terms from the text.

- The TextRank model outperformed others with a ROUGE-2 score of 0.0908 on the test set showing better performance in capturing meaningful word pairs.

- The BART model excelled in this metric with a ROUGE-L score of 0.1182, indicating it maintained a better sequence of words in its summaries.

- The T5 model had the lowest ROUGE-2 score (0.0842) and ROUGE-L score of 0.1151, reflecting less effective logical flow.

- The highest BLEU score was also from the TextRank model (0.0484), showing better fluency in the generated summaries.

- The BART model had the lowest BLEU score on test data (0.0198), indicating it may lack some precision despite good coherence.

Model	Hyperparameters	Execution Time
TextRank	Top Ranking sentences: 3 A smoothing function for BLEU score adjusts the score calculation for very short generated text.	2hrs
T5 Transformer Model	Model-name: t5-base Tokenization: min-length=512 and max-length=150 Summarization: min-length=30, num-beams=4 length-penalty=2.0 Fine-Tuning: Learning-rate=5e-5 epochs=5 batch-size=8 early-stopping =True Optimizer: AdamW Loss Function: CrossEntropyLoss()	3hrs
BART Transformer Model	Model-name: facebook/bart-large-cnn Tokenization: max-length=1024 Summarization: max-length=150, min-length=30 num-beams=4 length-penalty=2.0 Fine-Tuning: Learning-rate= 5e-5 epochs=3 batch-size=8 early-stopping =True Optimizer: AdamW Loss Function: CrossEntropyLoss()	1hr 30mins

Table 2: Hyperparameters And Execution Time

Model Name	**Dataset**	**Rogue-1 Score**	**Rogue-2 Score**	**Rogue-L Score**	**BLEU Score**
TextRank	Validation	0.1932	0.0846	0.0998	0.0451
	Test	0.198	0.0908	0.1051	0.0484
T5	Validation	0.1645	0.0849	0.1188	0.029
	Test	0.1583	0.0842	0.1151	0.0291
BART	Validation	0.1811	0.0913	0.1244	0.0214
	Test	0.1715	0.0865	0.1182	0.0198

Table 3: Evaluation Results

5 Key Insights

- The TextRank model consistently outperformed the other models across most metrics, particularly in ROUGE-1 and ROUGE-2, making it the strongest candidate for extracting key phrases.

- The BART model showcased its strength in maintaining logical coherence, as evidenced by its highest ROUGE-L score outperforming both T5 and TextRank, making it particularly useful when the flow of information is crucial.

- T5 seems to struggle with key term capture, as can be observed by its lower ROUGE-1 scores, although it maintains reasonable coherence in terms of sequence.

- The current results show that overall performance metrics are relatively low, with ROUGE and BLEU scores far from optimal (closer to 1 indicates stronger summarization ability). This highlights the need for further model refinement.

- Our immediate goal is to improve the performance of these models through hyperparameter tuning, additional training, and potential model adjustments, aiming to enhance their ability to generate summaries that closely match the gold standards.

6 Key Challenges

To measure the success of our models, we will use standard evaluation metrics for text summarization, focusing primarily on the following:

- **TF-IDF Implementation**: Initially, we aimed to implement TF-IDF as our baseline model. However, we encountered difficulties in scaling the implementation to the entire dataset. While the model performed well on smaller datasets, it failed to handle larger data volumes effectively. We are currently experimenting with ways to optimize it for the full dataset to get successful run using TF-IDF

- **Compute Limitations in Google Colab**: A major challenge was the lack of sufficient compute units in Google Colab, where we consistently maxed out our GPU resources. This bottleneck significantly slowed down our training processes, especially for larger models like T5 and BART, which are computationally intensive. This limitation hindered our ability to experiment more effectively with different configurations and hyperparameters.

- **Memory Consumption and Session Crashes**: Another issue was high memory consumption, which frequently caused our Colab sessions to crash. To work around this, we resorted to creating multiple Google accounts to access additional resources.

7 Next Steps: Upcoming Results and Analysis

Based on the evaluation of our current models, we have identified several key improvements that could significantly enhance the performance of our summarization models. Moving forward, our focus will be on the following strategies:

- **TextRank Model Optimization**: For the TextRank model, we will experiment with increasing the number of top sentences used in the summarization. Expanding from 3 to a larger number of sentences could possibly allow for better summarization ability, especially for longer texts, to maintain balance between capturing relevant details and conciseness.

- **T5 and BART Model Refinement**: To enhance the T5 and BART model's performance, we will explore adjusting key hyperparameters such as the maximum length of tokens during the encoding phase. Additionally, increasing the length of the summaries generated and reducing the penalty for generating longer texts could possibly yield more accurate and insightful summaries. We expect these changes to significantly improve the model's ability to capture and present complex information.

- **Lemmatization and Stop Word Analysis**: Initially, we avoided lemmatization and stop word removal due to their presence in the gold summaries. However, given the current performance results, we will now experiment and implement these techniques and analyze their impact on the model's output. Removing redundant stop words and normalizing words to their root forms could help streamline the summaries and improve model metrics.

- **Handling Missing Data**: While the missing data count in our initial dataset was minimal (1-2 records) and we didn't handle for it initially, however we now plan to remove all rows with missing data across the entire dataset and recompute the model performance. This step would help us eliminate any potential inconsistencies and ensure more reliable evaluation metrics.

- **Automated Hyperparameter Search**: Using automated hyperparameter tuning techniques, such as grid search or random search, could help us identify the optimal settings for each model, particularly for complex transformers like T5 and BART.

- **Evaluation of Alternative Summarization Techniques**: We will also explore alternative summarization methods, including hybrid approaches that combine extractive and abstractive techniques. By testing new models and comparing them with our current implementations, we aim to discover new strategies for handling complex financial text data. We would compare the performance of these new models with our improved model implementation (TextRank, T5 and BART).