MLOpt – 15.095
Final Report

Context-Aware LLMs for Financial Data

by

Alessandro Morosini: morosini@mit.edu
Andrea Valacchi: andval33@mit.edu

Date of submission
December 6, 2024

1 Introduction

1.1 Problem Statement

Business is a creative endeavor. As such, it entails a fundamental component of risk. In our project, we elected to focus on the most extreme of them all: Bankruptcy Risk. Effective early prediction of this risk is a crucial component of Business Valuation, given the ramifications of its impact on every Company Stakeholder, from Investors (retail and institutional), to Regulatory Bodies.

To that end, our project explored the applications of the TabText Carballo et al. (2022) framework to the realm of Financial Ratios Analysis, specifically investigating its potential in augmenting the information stored in the tabular data alone by leveraging contextual information.

1.2 Current Approaches and Project Motivation

The methods we considered as incumbent benchmarks consisted of training state-of-the-art Machine Learning models on the tabular data alone, which is described more in detail in the relevant section.

Two main issues arise when following this approach:

- Working with tabular data is labor-intensive and can be highly time-consuming process.
- Critical information could be lost by ignoring the broader context in which the analysis is performed.

More specifically, working with tabular data (especially in the context of high-dimensional datasets) requires laborious cleaning and preprocessing, such as handling missing values (highly consequential modeling choice), outliers, or inconsistent formats. All these factors demand some degree of domain knowledge to extract meaningful insights from the data and ultimately hinder the scalability of the entire Analysis.

As discussed in class, the TabText framework has the potential not only to bypass the shortcomings of all the preprocessing phase, but also to augment the information contained in the tabular data by leveraging the "knowledge" learned by the LLM used to generate the text embeddings during the training phase. The results section provides detailed insights into how TabText scored along both these dimensions.

2 Data

2.1 Dataset Description

In our analysis, we utilized the UCI Polish Companies Bankruptcy dataset Tomczak (2016), available on the UCI Machine Learning Repository.

It consists of a wide array of financial records gathered from 43,405 Polish Companies over a five-year period, which were classified based on having defaulted or not (our target binary variable) in a given year. We selected this particular dataset because of its balanced pool of 65 different financial indicators, which we grouped in the following categories:

- Profitability & Cost Effectiveness (Most notably, Return on Assets and Gross Profit Margin...)
- Debt Management (Financial Leverage)
- Liquidity (Net Working Capital Ratio)

- Operational Efficiency (Asset Turnover)

This categorization was an essential prerequisite step to develop the function used to create a single, cohesive string of text to extract the embeddings from, as discussed in the following section.

3 Methodology

3.1 Preprocessing

In our experiment design, we divide preprocessing into two steps. The first has the goal of solving the issues of missing values and categorical values, which regular algorithms for tabular data cannot handle. The second is carried out to transform the data in a natural language, in a way that TabText is able to process.

3.1.1 Under sampling

As emerged from exploratory data analysis, the dataset is heavily unbalanced: only 5% of the company reports bankruptcy. This motivated the use of under-sampling techniques of the majority class. This method was preferred to oversampling the minority class mainly due to computational constraints derived from the creation of the embeddings. Two approaches were implemented and tested: random and optimized.

Random For random subsampling, we divide the data into buckets, defined by the year of data collection. Then, within each bucket, we randomly sampled companies from the ones not reporting bankruptcy, so that they matched, in terms of quantity, the ones that had filed for bankruptcy.

Optimized For the optimized approach, we adapt the techniques used in Bertsimas and Weismantel (2005). Specifically, we implement the formulation

$$\begin{aligned}
\min_z \quad & \sum_{i\in S_1^t}\sum_{j\in S_0^t} z_{ij}\|x_i - x_j\|^2 \\
\text{s.t.} \quad & \sum_{j\in S_0^t} z_{ij} = 1, \quad \forall i \in S_1^t, \\
& \sum_{i\in S_1^t} z_{ij} \le 1, \quad \forall j \in S_0^t, \\
& z_{ij} \in \{0,1\}, \quad \forall i \in S_1^t,\ j \in S_0^t,
\end{aligned}$$

where S_1^t and S_0^t respectively represent the company going bankrupt and not going bankrupt in year t. This problem aims to find matches of similar companies within a given year such that the total normalized distance of the covariance is minimized.

Once the process is completed, we then join the data from the different buckets, obtaining a train data set in which the label distribution was balanced.

3.1.2 Tabular

Two main challenges were to be addressed when preparing the data: missing values and categorical variable encoding.

Missing Values The data set contains a relatively small amount of missing values, around 1%. This motivated the use of imputation techniques rather than an analysis devoted to understanding whether to drop specific samples or features. For numerical variables, we performed median imputation, while for categorical variables we used the mode.

Encoding The original data set did not contain any categorical variables. However, after feature engineering, which involved the quantization of some numerical features, categorical variables were introduced. For encoding, we explored both ordinal encoding and one-hot encoding. Ordinal encoding seemed to perform much better, which was expected given the ordinal nature of the quantization problem. Therefore, the results are reported only for ordinal encoding.

3.1.3 Embeddings

The use of TabText requires two steps for processing the tabular data: the conversion of sample-level information into strings, and the creation of embeddings.

For the first phase, we divided the features into four different categories, based on financial intuition. For each of these, we introduced a string representing the content, to add value through metadata. These were used to create a paragraph for each of the sample, introduced by a context-providing sentence, containing all the financial information. Additional details and an example are provided in Appendix 7.1.

The second phase involved the creation of embeddings from the paragraph. For this task, we initially used BERT. However, recognizing the potential benefits of using an LLM fine-tuned on financial data, we transitioned to FinBERT Huang et al. (2023), which incorporates domain-specific knowledge and improved overall performance. The resulting 768 embeddings per sample were then used to train TabText.

3.2 Experiments

Several experiments were carried out, each with different purposes. These included:

- Comparing FinBERT's capabilities against BERT
- Evaluating TabText versus tabular and multimodal approaches
- Analyzing the impact of missing values on performance
- Assessing the importance of context information provided by language

In this section, we outline the setup.

3.2.1 BERT vs FinBERT

This experiment involved the comparison of BERT with FinBERT. This was run as a preliminary experiment to verify our hypothesis that FinBERT would perform better thanks to the pre-training on financial data.

3.2.2 Tabular vs Embeddings

This experiment involved the comparison of an XGBoost model trained on tabular data with an XGBoost trained on contextual embeddings according to the TabText framework. The former was fitted to processed data, while the latter was trained on raw and clean data.

3.2.3 Multimodality

This experiment aimed to explore the potential of augmenting tabular data with embeddings. This involved the comparison of the two models presented in the previous section, with a multimodal framework combining both datasets. The multimodal model was fitted to both raw and clean data.

3.2.4 Missing Value Analysis

In this experiment, our objective was to evaluate the impact of missing values on TabText performance, compared to normal models trained on tabular data. This investigation was motivated by one of the key advantages of TabText: its minimal pre-processing requirements, which eliminate the need to handle null values, unlike traditional models. We generated five different datasets, each with an increasing proportion of missing values: starting with 10% missing values in the first dataset, followed by increments of 10%, up to a maximum of 50% missing values. For each of these, we compare TabText's performance over traditional models.

3.2.5 Context Information Analysis

This experiment was meant to quantify the importance of context information for TabText. It was divided into two parts, one testing the value of column names, which was tested by running a TabText on anonymized column, and one testing the value coming from metadata information, which was tested by removing the contextual information.

4 Results

All of the obtained by performing a train-test split based on year, with train being the first three years of data and test being the last two, to avoid any look-ahead bias. The models are cross-validated using a 5-fold cross-validation, and all reported results are out-of-sample. Moreover, the data is undersampled following the randomized approach, as this yield to much greater out-of-sample performance. This is probably due to the fact that the optimization approach removes a great part of diversity in the train, which is needed by the model to generalize to unseen data whose distribution is otherwise unkown.

4.1 BERT vs FinBERT

The results show that our hypothesis was confirmed. FinBERT improves on BERT, though the improvement in performance is moderate. We believe this is due to the nature of the dataset, which contains very complex ratios that might be hard to interpret. For the rest of the analysis, FinBERT was used as the go-to model to generate embeddings.

4.2 Tabular vs TabText

In this section, we present the results of the comparative analysis of the fitting of an XGBoost model on tabular data versus contextual embeddings. A performance metric report is provided in Table 1.

From the results, we can see a significant gap in performance between the two models, with XGBoost on tabular data outperforming the others. Notably, this difference is not due to the preprocessing applied to the tabular data, as the performance of TabText remains impressively consistent across both raw and clean data, having been tested on both. Therefore, we attribute this gap to the inability of the embeddings to capture a deep representation of the context effectively. Further analysis is provided later to confirm this hypothesis.

Metric	FinBERT (Raw)	FinBERT (Clean)	Tabular
Test Accuracy	0.668	0.663	0.871
Test AUC	0.752	0.753	0.942
Test Sensitivity	0.732	0.734	0.858

Table 1: Comparison of performance metrics across models.

4.3 Multimodal

Embeddings per se seem to have much less information than the processed tabular data. But maybe the combination of two might have more value. With this idea in mind, we explored training a multi-modal model, which combines the tabular data with the embeddings generated from the paragraphs. For this, we used the embeddings generated from raw data, given that the power of TabText also lies in the need of minimal preprocessing. The results are contained in Table 2

	Metric	FinBERT (Raw)	Tabular	Multi-Modal
0	Test Accuracy	0.668	0.871	0.815
1	Test AUC	0.752	0.942	0.912
2	Test Sensitivity	0.732	0.858	0.819

Table 2: Comparison of performance metrics across models, including multimodality

The multimodal model achieves good predictive performance. However, while its metrics are higher than those of TabText, they remain lower than those of the XGBoost model trained on clinical data. This suggests that the embeddings do not add significant value. We tested this hypothesis in two different ways. First, by running PCA on the embeddings and then by training a multimodal model that combines tabular data with the principal components from PCA. Interestingly, the results show that increasing the number of components deteriorates out-of-sample performance. This suggests that the embeddings likely encode information already captured by the tabular data.

This observation is further confirmed by analyzing the SHAP values for the multimodal model. As shown in Figure 1, among the 20 most relevant features of the model, only one is an embedding, while the rest are financial ratios. Moreover, similar conclusions can be drawn by looking at the OCT splits **??** where embeddings do not appear in the decision rules. Therefore, we conclude that the embeddings do not provide additional value and might even introduce noise.

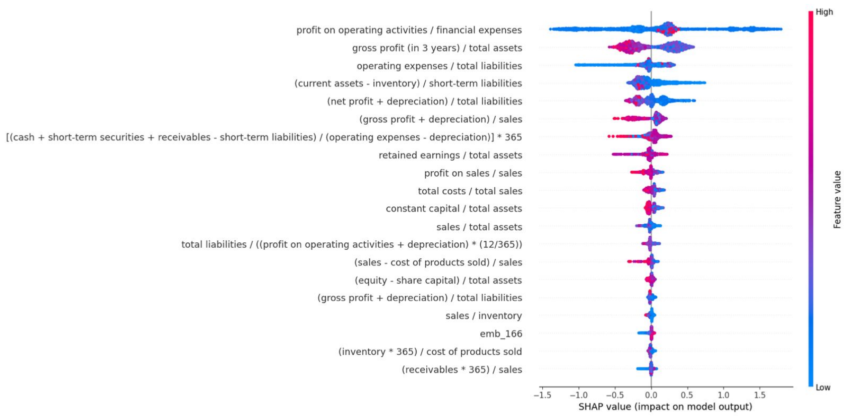

Figure 1: SHAP Values for Multi-Modal Model

4.4 Missing Value Analysis

The experiments show very interesting results, as it seems that TabText is more resistant to missing values. Figure 2 shows the performance of TabText compared to XGBoost on tabular data, as a function of varying the number of missing values. Results are reported on AUCs; however, similar performance can also be observed in other metrics.

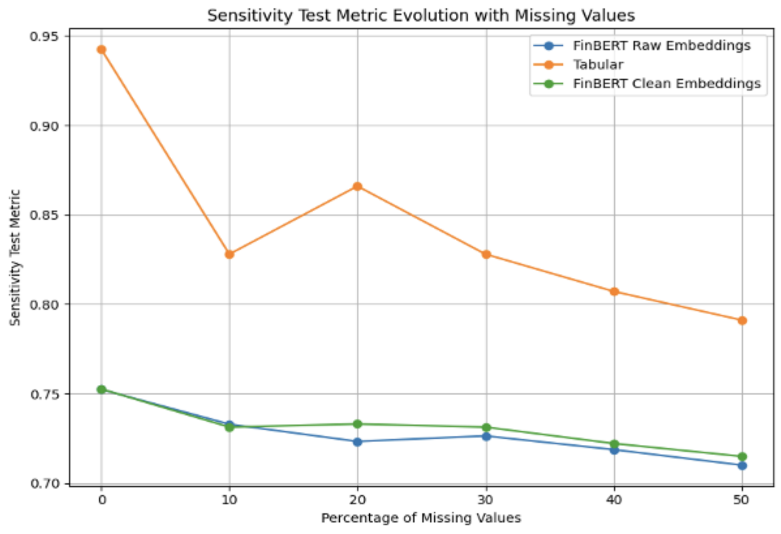

Figure 2: Comparison of Model AUCs for Different Levels of Missing Values

The model trained on tabular data outperforms the model trained on the embeddings for experiment zero (no addition of missing values). This is in line with the results in Section 4.2. Interestingly, we see that the gap in the two models tends to shrink as the number of missing values increases. In fact, XGBoost on tabular data performance appears to drop dramatically. There is an exception at the level of 20% missing values, where it seems that imputation techniques hit a lucky spot, as confirmed by the increase of FinBERT on clean embeddings; this could be worth investigating further. On the other side, the performance of TabText on embeddings is plateauing, both on clean and raw data, suggesting that the model is weakly influenced by the quality of the data. The gap between the two models is more than halved when using poor-quality data. This raises questions for future research, some of which were already explored briefly, as presented in Section 4.6.

4.5 Context Information Analysis

This experiment was motivated by the fact that FinBERT outperformed BERT, but not by a large margin. This lead us to think that the names of the dataset features might be too complex for the model to understand. In fact, most of the features are precomputed financial ratios, some of which being highly complex. For instance, consider the feature:

*[(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365.*

It is worth emphasizing that FinBERT is fine-tuned on financial reports, which primarily deal with basic financial quantities rather than such complex ratios. This discrepancy might also explain why the performance of TabText is lower than that of the tabular model, as the financial context is not fully captured by the model due to this added complexity.

To better understand this, we ran an experiment in which we anonymized the column names to better understand their impact on performance. Table 3 summarizes the results.

Metric	Non-Anonymized	Anonymized	Change (%)
Test AUC	0.752	0.738	-1.86
Test Sensitivity	0.732	0.709	-3.14

Table 3: Comparison between non-anonymized and anonymized ratios, and percentage changes.

As we can see, anonymization of column names resulted in a slight drop in performance for both metrics. On the one hand, this supports the hypothesis that the feature names provide contextual cues that the model utilizes. On the other hand, however, we note that the drop in performance is limited. This suggests that the model still relies heavily on the underlying tabular data rather than on the feature names themselves.

4.6 Ensembling

Results from Section 4.4 highlight how TabText can be valuable when dealing with large quantities of missing values. We believed that this finding could be highly consequential and worthy of further investigation, hence we decided to conduct some experiments on the matter, focusing on the extreme scenario with 50% missing values. For this setting, we fit all the models discussed so far and evaluated their performance. Specifically, we considered: the metrics of XGBoost on tabular data, those of XGBoost on embeddings created from raw data (similar performance for clean) and those of a multimodal model that combined the two.

Specifically, we explored whether the ability of TabText to handle missing values could be used to enhance the predictions of tabular models. In essence, we wondered if TabText could 'correct' predictions from tabular models. Although the multimodal approach provides all the information available for a single model, we hypothesized that an ensemble approach could better combine the strengths of each model.

For this purpose, we implemented an ensemble strategy. Specifically, we used logistic regression, Random Forest, and XGBoost classifiers as meta-models, trained on the probability outputs of the tabular and embedding-based models. Among these, XGBoost performed the best, achieving the highest metrics. The results of this analysis are presented in Table 4.

Metric	Raw Data	Tabular	Multimodal	Ensemble (XGBoost)
Test AUC	0.699	0.792	0.780	0.818
Test Sensitivity	0.725	0.770	0.764	0.777

Table 4: Comparison of models in the 50% missing values scenario.

The ensemble model outperformed both individual models and the multimodal approach, demonstrating the value of combining the distinct strengths of tabular and embedding-based models in the presence of many missing values.

5 Conclusion

In conclusion, we developed and tested models to predict bankruptcy risk, with XGBoost on tabular data achieving the best performance, reaching an AUC of 0.942. While we hypothesized that incorporating contextual financial information through the TabText framework and FinBERT embeddings would enhance performance, the embeddings did not add significant value. We attribute this to two factors: (i) the financial ratios used as features are too complex for the language model to effectively capture their meaning, and (ii) the dataset is relatively clean, whereas TabText appears better suited for handling more "dirty" data.

These findings suggest two promising directions for future research: (i) evaluating TabText on datasets with simpler, more self-explanatory features, and (ii) testing its performance on noisier datasets where its robustness to missing values and minimal preprocessing requirements may provide an advantage. Our ensemble experiments demonstrated that TabText can complement tabular models, particularly in scenarios with high levels of missing data, showcasing its potential to add value in specific contexts.

6 Contributions

Alessandro conducted the dataset research, while Andrea focused on the LLM research. For preprocessing, Alessandro handled under sampling and tabular data processing, while Andrea was responsible for generating embeddings for the different models. During the experiments, both contributed equally by "parallelizing" their efforts. Alessandro fit the initial models and performed missing value analysis and ensembling, whereas Andrea focused on anonymization analysis, multimodality, and visualizations. Side experiments, such as PCA over embeddings and SHAP values analysis, were carried out collaboratively. Both team members contributed equally to creating the slides and writing this report.

References

Bertsimas, D. and R. Weismantel (2005): "Optimization over integers," *Dynamic Ideas.*

Carballo, K. V., D. Bertsimas, and al. (2022): "TabText: A Flexible and Contextual Approach to Tabular Data Representation," .

Huang, A. H., H. Wang, and Y. Yang (2023): "FinBERT: A Large Language Model for Extracting Information from Financial Text*," *Contemporary Accounting Research*, 40.

Tomczak, S. (2016): "Polish Companies Bankruptcy," UCI Machine Learning Repository, DOI: https://doi.org/10.24432/C5F600.

7 Appendix

7.1 String Creation

To enhance TabText's ability to understand a representation of the context, we divided the features into four categories: "Profitability Ratios", "Liquidity Ratios", "Leverage Ratios" and "Efficiency Ratios". Moreover, we also introduced the paragraph with the following context-giving sentence: "We are analyzing the financial health of a company to assess its risk of bankruptcy. These are the key financial ratios and indicators of the company". Below is an example of how the paragraph looks like for an individual, after string processing but before embedding creation.

We are analyzing the financial health of a company to assess its risk of bankruptcy. These are the key financial ratios and indicators of the company. Profitability Ratios: The ratio of Net profit divided by total assets is 0.126. [...] The ratio of Profit on operating activities divided by sales is 0.0598. **Liquidity Ratios:** The ratio of Working capital divided by total assets is 0.1876. [...]. The ratio of Sales divided by short-term liabilities is 5.4586. **Leverage Ratios:** The ratio of Total liabilities divided by total assets is 0.5422. [...] The ratio of Total liabilities divided by profit on operating activities + depreciation multiplied by 12 divided by 365 is 0.0832. **Efficiency Ratios:** The ratio of Inventory multiplied by 365 divided by sales is 56.248. [...] The ratio of Total sales divided by total assets is 2.9581.

7.2 OCT splits

We trained an Optional Classification Tree (OCT) on the multimodal dataset. Our goal was not only to achieve strong out-of-sample accuracy, but also to gain insights into feature importance by examining the tree's structure. Figure 3 shows the OCT truncated at a depth of three levels for clarity. Notably, embeddings do not appear among the top splits, nor do they feature at deeper levels of the tree (which cannot be observed by the plot). This absence confirms that embeddings may provide limited value in this context.

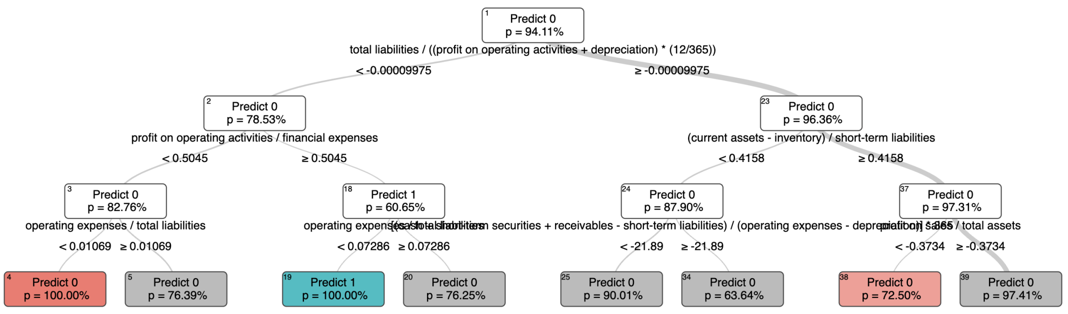

Figure 3: (Truncated) Optimal Classification Tree