

# TopoLedgerBERT: Topological Learning of Ledger Description Embeddings using Siamese BERT-Networks.

Sander Noels<sup>1,2</sup>, Sébastien Viaene<sup>1</sup>, Tijn De Bie<sup>2</sup>

<sup>1</sup>Silverfin, Gaston Crommenlaan 12, 9050 Ghent, Belgium

<sup>2</sup>Department of Electronics and Information Systems, Ghent University, 9000 Ghent, Belgium

Correspondence: [sander.noels@ugent.be](mailto:sander.noels@ugent.be)

## Abstract

This paper addresses a long-standing problem in the field of accounting: mapping company-specific ledger accounts to a standardized chart of accounts. We propose a novel solution, TopoLedgerBERT, a unique sentence embedding method devised specifically for ledger account mapping. This model integrates hierarchical information from the charts of accounts into the sentence embedding process, aiming to accurately capture both the semantic similarity and the hierarchical structure of the ledger accounts. In addition, we introduce a data augmentation strategy that enriches the training data and, as a result, increases the performance of our proposed model. Compared to benchmark methods, TopoLedgerBERT demonstrates superior performance in terms of accuracy and mean reciprocal rank.

## 1 Introduction

Ledger accounts are essentially the building blocks of accounting that provide detailed records of all financial activities within a company. They collectively form a hierarchical structure known as the chart of accounts (COA), which offers a comprehensive view of a company's financial activities. However, companies often customize their COA to accommodate their specific business needs and regulatory requirements, resulting in a lack of standardization and comparability (Jørgensen and Igel, 2021; Wang, 2023). This scenario has brought about various standardization initiatives aiming to enhance the quality of COAs. A standardized COA not only improves the accuracy of financial data, but also promotes comparability and improves the overall quality of the financial information reported (Rylee, 2017). Moreover, it paves the way for machine learning solutions that can generalize financial information between companies (Bergdorf, 2018).

The mapping of company-specific ledger accounts to a common COA remains a challenge. Existing academic and industry solutions typically create custom machine learning models for individual companies, or limit data to a predefined set of accounts (Bergdorf, 2018). Furthermore, the hierarchical information inherent in a COA, which could significantly enhance the performance of the mapping solution, has never been used.

We hypothesize that the hierarchical structure of the ledger accounts within the COA and their interrelationships can enhance the performance of the mapping solution. Considering the meaning and hierarchical nature of the ledger account descriptions, we believe that a model that understands these relationships should deliver superior ledger mapping results.

We propose TopoLedgerBERT, a novel sentence embedding method for ledger account mapping that overcomes the limitations of existing methods by exploiting the hierarchical nature of ledger accounts. To create the mapper, we take a ledger description from the custom COA, compute its embedding, and then calculate its distance to embeddings of standard ledger accounts. The standard ledger account having the minimum distance is considered the mapping for the custom ledger account. TopoLedgerBERT incorporates a graph distance concept to include hierarchical information from the COA during the sentence embedding process. By doing so, it captures not only the semantic similarity of ledger accounts but also their hierarchical structure. Additionally, this method is capable of making predictions for previously unseen companies and COAs.

Our main contributions are summarized as follows:

- We propose TopoLedgerBERT, a novel sentence embedding method specifically designed for the mapping of ledger accounts in accounting.
- Unlike existing solutions that require cus-

tomized machine learning models for individual companies, TopoLedgerBERT can adapt to new companies and COAs without any prior setup.

- We demonstrate that the TopoLedgerBERT model outperforms benchmark methods in terms of accuracy and mean reciprocal rank.
- Our empirical evaluation reveals that TopoLedgerBERT effectively captures hierarchical relationships between different ledger accounts.
- Finally, we demonstrate the importance of fine-tuning pre-trained embeddings and employing data augmentation strategies for improving the model's performance.

The remainder of this paper is as follows: Section 2 reviews the related work. Section 3 delves into the background and explains how a COA can be represented as a graph. Section 4 details our proposed TopoLedgerBERT model, followed by an empirical evaluation in Section 5. Finally, Section 6 concludes the paper.

## 2 Related Work

### 2.1 Ledger Accounts and the Chart of Accounts

Ledger accounts serve as units that record and summarize a company's transactions. A ledger account typically includes an ID that possibly reflects the hierarchy and a short description that can sometimes be challenging to decipher. Each ledger account also contains a set of bookings for each financial year. A company generally comprises around 100 ledger accounts.

These accounts collectively form a hierarchy known as the "chart of accounts" (COA), a classification system for financial information. The COA segregates expenditures, revenues, equity, assets, and liabilities into distinct categories, providing a comprehensive breakdown of all financial transactions during a specific accounting period. However, a long-standing issue in this domain is the lack of standardization in the COA.

### 2.2 Factors Driving the Customization of COAs

Customizing COAs is driven by several factors. Companies often require specialized ledger accounts to support specific financial transactions relevant to their sector (Jorge et al., 2022). In addition to sector-specific needs, the complexities of tax law

and individual company requirements contribute to the customization of COAs. These complexities underline the subjective nature of bookkeeping, where companies vary in their practices, making standardization a challenging endeavor (Liu et al., 2021). Two examples can illustrate these complexities:

- **Tax Compliance:** US tax laws dictate varying rates at which different expenses are deductible. For example, expenses for business vehicles and office snacks are 100% deductible, while business meals are only 50% deductible. Consequently, separate accounts for these expenses become necessary for accurate tracking and compliance.
- **Business Needs:** Companies may have different needs for account specificity. For example, company A might be satisfied with a generalized account named "cars and trucks" for all its automobile-related expenses. However, company B might prefer more granular accounting and create separate accounts for "fuel", "vehicle maintenance", and "vehicle insurance".

Although customization can support specific business needs and regulatory compliance, it can also lead to inconsistencies and difficulties in comparing financial statements between companies (Dhole et al., 2015; Noels et al., 2022, 2023a). Therefore, while supporting the freedom for companies to customize their COAs, it is essential to develop methods for translating these custom COAs into a unified structure.

### 2.3 The Importance of Standardization

Standardization initiatives have been introduced, both on national and international scale, to improve various aspects of accounting (EUROSTAT, 2017; Jorge et al., 2022). For example, in 2008, the European Commission recommended the use of standard COAs for small enterprises in a report that highlights best accounting practices (Commission, 2008). Standardization initiatives are beneficial for various reasons. Primarily, they enhance the accuracy of financial data and, consequently, financial reporting. A unified COA allows for the uniform sorting and aggregating of financial data, facilitating consistent accounting practices and routines. This, in turn, helps to develop effective IT and control systems (Jorge et al., 2022). According to Rylee (2017), the standardization of COAs ensures a consistent and streamlined reporting of financial

data, thus preventing inaccuracies that might result from too complex COAs.

## 2.4 Standardization Initiatives in Related Work

The "multiple charts of accounts problem" is a longstanding issue in both academia and industry due to the unique customization of COAs by different companies. The challenge lies in the need for a system that can effectively generalize and merge financial information across various companies and sectors, even with minimal historical data (Bergdorf, 2018).

Past strategies have tried to circumvent this problem using company-specific classifiers or the use of uniform ledger accounts or COAs. Jørgensen and Igel (2021) propose a system capable of mapping transactions to the appropriate ledger accounts across different companies. However, this approach required one to first manually map specific account codes into a unified space. Munoz et al. (2022) limit themselves to a set of accounts used by different companies, employing hierarchical classifiers to predict the correct account code for specified invoice line items. Despite the innovative aspect of their approach, this method is limited to the account codes present in their training data, making it ineffective for accounts added after their training period. Noels et al. (2022, 2023a) define a COA with all possible financial accounts hierarchically structured within a financial statement, allowing them to represent every company. This allows them to compute the similarity between the financial statements of each company. However, due to the lack of uniformity, the use of this method for automated financial statement analysis remains limited, indicating the need for a method capable of automating the translation to a standardized COA.

All the methods discussed face the challenge known as the "cold-start problem", which occurs when a system has to make predictions for ledger accounts that were not present in the training dataset. This limitation has inspired efforts toward the automation of COA standardization. In response to this challenge, a viable solution lies in automating the translation of a company's unique financial structure into a unified accounting chart. Such a chart should be capable of accounting for all potential company transactions and could therefore be effectively implemented in machine learning systems. According to Jørgensen and Igel (2021), the most significant opportunity for improvement

lies in the automated transformation to a standardized COA, while still allowing company-specific COAs tailored to individual company needs or tax regulations.

## 2.5 Account Mapping in Academia and Industry

Companies like Oracle and Silverfin provide their own mapping solutions. Oracle's solution includes a COA mapping feature for data consolidation, but this rule-based system requires substantial maintenance due to the need for continuous adaptation. Silverfin uses machine learning in its AI mapping feature for better adaptability. The company approaches account mapping as a classification problem that depends on textual information as well as transaction details associated with a custom ledger account. Given this approach of considering the problem as a classification tied to a COA, it inevitably faces difficulties when encountering new COAs. Both aim to automate the process of mapping custom ledger accounts, highlighting their importance in the industry.

There is a growing academic interest in automated account mapping solutions, as demonstrated by a system proposed by Wang (2023). The system standardizes custom ledger accounts by automatically mapping them to standard ones using three pre-trained embedding methods (TF-IDF, Word2Vec, FinBERT). To the best of our knowledge, this is the first academic work that offers a solution to the laborious process of mapping custom ledger accounts to a standardized taxonomy.

The rise of large language models opens up new opportunities in financial data applications. The rationale is that these methods could be better adapted to the unique terminology used in financial data, potentially improving the accuracy of mapping custom ledger accounts (Noels et al., 2023b). According to Wu et al. (2023), the complexity of the financial domain, the unique terminology, scarce data, and privacy regulations have limited the success of large language models in providing valuable insights. Therefore, fine-tuning pre-trained embeddings, such as FinBERT, could potentially overcome these hurdles and offer more accurate representations of financial data, thus improving the mapping process (Araci, 2019).

Liu et al. (2021) introduce another promising approach, using siamese network-inspired neural networks to perform binary matching of transaction and account information. This method uses

pre-trained embedding models to place transactions and accounts in the same vector space. This approach could be applied to the account mapping problem, where custom ledger account descriptions are paired with their standardized ledger account descriptions. This approach offers a distinct advantage over classification-based methods, as it moves away from handling a fixed COA.

In conclusion, the siamese network-inspired approach appears to be a promising avenue for future research, as it is not restricted to a predefined number of accounts and can accommodate unseen ledger accounts and COAs. Additionally, the idea of fine-tuning language models on financial data presents a compelling opportunity. Given the unique terminology, models trained specifically for this mapping problem would be better equipped to handle the complexity and nuances of ledger descriptions. These insights inspire further exploration of the development of more robust, adaptable, and accurate systems to automate the account mapping process in financial accounting.

### 3 Background

This section provides an overview of the COA as a graph and the application of siamese BERT-networks in account mapping.

#### 3.1 Chart of Accounts as a Graph

As stated by Yang and Cogill (2013), a vertex-labeled tree is a natural representation of the ledger accounts present within a COA.

As an example, we consider the *assets* section of a balance sheet. The *assets* section can be divided into *fixed* and *current assets*. Both of these ledger accounts can be subdivided into more detailed accounts, as shown in Figure 1. Note that the vertex-labeled representation of a COA is not limited to this specific example. A subset of ledger accounts and their reciprocal relationship are given for exemplary purposes.

This representation method clearly preserves the structural property of a COA. This means that a COA for every company can be represented by a vertex-labeled tree, where the vertex labels are the ledger account descriptions.

#### 3.2 Siamese BERT-Networks

Siamese BERT-networks are a type of sentence embedding method that can embed a sentence, taking into account the semantic and contextual information of a sentence. In the context of account

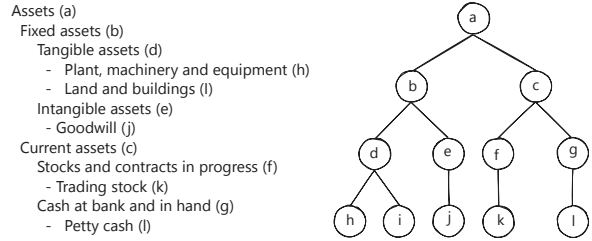


Figure 1: Left: Assets subsection of the balance sheet. Right: A vertex-labeled tree representation of the assets subsection of the balance sheet.

mapping, these networks use the bi-encoder transformer architecture of the Sentence-BERT network (Reimers and Gurevych, 2019) to learn a joint representation space of the ledger account descriptions and their standardized counterparts.

Training a sentence embedding method involves inputting pairs of sentences into the network. These pairs can be either positive pairs, which consist of semantically similar sentences, or negative pairs, which consist of semantically dissimilar sentences. The network then calculates the distance between the two inputs. It attempts to minimize the distance for similar inputs (e.g., identical account descriptions) and maximize the distance for dissimilar inputs (e.g., different account descriptions).

### 4 TopoLedgerBERT

#### 4.1 Rationale Behind the Proposed Solution

The rationale behind the proposed solution is to address the limitations of current sentence embedding methods in the context of ledger account mapping. As demonstrated in previous studies, the hierarchical relationships between the ledger accounts play a significant role in understanding the financial position of a company (Yang and Cogill, 2013; Noels et al., 2022, 2023a). Therefore, the embedding method we propose aims to capture not only the semantic similarity of the ledger accounts, but also their hierarchical composition within the COA. In doing so, we hypothesize that the proposed method can provide better embeddings for ledger accounts, leading to improved mapping accuracy and efficiency in the accounting domain.

Let us consider a specific example. The ledger account for *land and buildings* can be subdivided into *agricultural land* and *residential land*. In this context, the proposed method should consider *agricultural land* and *residential land* as more similar to each other than to other ledger accounts that are located farther apart on the graph. This is due to

their direct hierarchical relationship and position in the COA.

## 4.2 TopoLedgerBERT

As mentioned in Section 3.1, a COA can be symbolized by a vertex-labeled tree, represented as  $T = (V, E, L, \phi)$ . In this context,  $V(T) = \{1, \dots, n\}$  stands for a set of  $n$  vertices,  $E \subseteq V \times V$  indicates the set of edges, and  $L$  is a collection of unique labels corresponding to each vertex in  $V$ . These labels in  $L$  define the standardized ledger account descriptions within a target COA. Therefore, we introduce a function  $\phi : V \rightarrow L$  that maps every vertex  $v$  in  $V$  to a unique label  $l$  in  $L$ .

We also define the shortest path distance  $d(i, j)$  between the vertices  $i$  and  $j$  in  $T$  and construct the distance matrix  $D(T)$  of  $T$ . Using  $D(T)$ , we construct a similarity matrix  $S(T)$ , where each value

$$s_{ij} = 1 - \frac{d_{ij}}{\max(D)} \quad (1)$$

represents the similarity between the vertices  $i$  and  $j$ . If the value of  $s_{ij}$  is high, it signifies close proximity within the tree  $T$ , while low values indicate a significant distance.

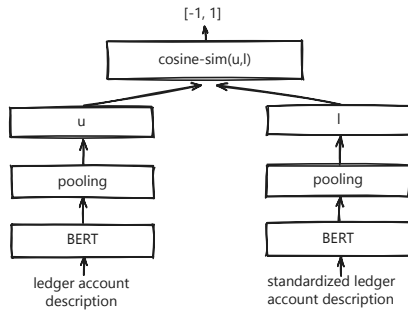


Figure 2: Diagram of the Sentence-BERT architecture for computing ledger account description similarity scores for the ledger account mapping problem.

We create positive training samples by pairing each original ledger account description with its corresponding ledger account from the COA, assuming that there is an appropriate standardized ledger account for each account description. We denote the set of original ledger account descriptions as  $U$ , and the corresponding standardized ledger account descriptions as  $L$ . Each sample in the positive training dataset  $D_+$  is a triplet  $(u, l_u, s_i) \in U \times L \times \{1\}$ , where  $s_i = 1$  indicates the highest similarity score.

To define the set  $D_+$ , we introduce  $v_u$  which represents the vertex in  $V$  corresponding to the

original ledger account description  $u$ . The function  $\phi : V \rightarrow L$  then maps each vertex  $v_u$  to its corresponding unique label  $l_u$ . Given these definitions, we can formally define  $D_+$  as follows:

$$D_+ = \{(u, \phi(v_u), 1) | u \in U\}, \quad (2)$$

where  $u$  represents an original ledger account description, and  $\phi(v_u)$  denotes its corresponding label in the COA.

For each tuple  $(u, \phi(v_u), 1) \in D_+$ , we generate  $K$  additional negative training samples. First, let us define the set  $V_{\setminus v_u}^K \subseteq V \setminus \{v_u\}$  with  $|V_{\setminus v_u}^K| = K$ , a randomly selected subset of  $K$  vertices from  $V$  different from  $v_u$ . Subsequently, for each  $v \in V_{\setminus v_u}^K$ , we calculate the similarity score  $s_{v_u v}$  using the similarity matrix  $S(T)$ . The set of negative training samples  $D_-$  is defined as:

$$D_- = \{(u, \phi(v), s_{v_u v}) | u \in U, v \in V_{\setminus v_u}^K\}, \quad (3)$$

with  $|D_-| = K|D_+|$ .

Finally, the augmented data set  $D_{aug}$  is the union of  $D_+$  and  $D_-$ :

$$D_{aug} = D_+ \cup D_- \quad (4)$$

$D_{aug}$  functions as the input for the TopoLedgerBERT model, as shown in Figure 2. The dataset augmentation method is visualized in Figure 3.

## 5 Empirical Evaluation

This section provides a thorough empirical evaluation of TopoLedgerBERT’s effectiveness in mapping ledger account descriptions to a standardized taxonomy, based on experiments using real-world financial data.

### 5.1 Data Description

The TopoLedgerBERT model uses a confidential dataset from Silverfin<sup>1</sup>, an accountancy cloud service provider. This dataset comprises financial statement data from 5,608 UK companies from various industrial sectors and market sizes. Each company has mapped their original ledger account descriptions to the corresponding ledger accounts of one of six possible COAs, resulting in 521,265 unique mappings. The dataset is split into a 90%-10% training and testing set.

For each unique COA in the dataset, a vertex-labeled tree  $T_{config} =$

<sup>1</sup><https://www.silverfin.com>

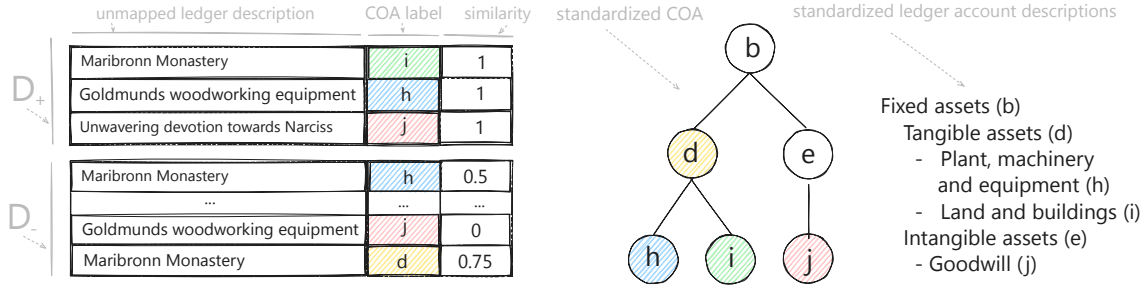


Figure 3: Example construction of an augmented dataset,  $D_{aug}$ , by TopoLedgerBERT for ledger account mapping.

$(V_{config}, E_{config}, L_{config}, \phi_{config})$  with  $config \in \{1, 2, 3, 4, 5, 6\}$  was constructed. Each chart has  $\{282467, 151524, 40183, 28573, 10311, 8207\}$  mappings, respectively.

## 5.2 Methods Evaluated

To assess the performance of our proposed TopoLedgerBERT model, we benchmark it against several established models. The following methods are considered:

- Standard SBERT: Utilizes the sentence-transformer model (*all-MiniLM-L6-v2*<sup>2</sup>) pre-trained on over 1B English sentence pairs (Reimers and Gurevych, 2019).
- Fine-tuned SBERT: An improved version of the Standard SBERT, incorporating multiple negatives ranking loss with in-batch negatives. The model’s training hyperparameters inspired by Decorte et al. (2023) are found in Appendix A.
- FinBERT: A BERT model variant specialized for financial texts (Araci, 2019). FinBERT is not utilized to generate sentence embeddings but rather to understand the financial context at the token level. A mean pooling strategy is used to derive sentence representations.
- TopoLedgerBERT: Our proposed model. It starts from the same sentence-transformer model (*all-MiniLM-L6-v2*) and employs a cosine similarity loss function that acknowledges hierarchical distances between different ledger accounts. Training hyperparameters can be found in the Appendix A.

## 5.3 Performance Metrics

In assessing the effectiveness of the proposed TopoLedgerBERT model, we employ four performance metrics: accuracy (Acc), mean reciprocal

<sup>2</sup><https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

rank (MRR), and two novel measures, the Mean Misprediction Distance (MMD) and the Mean Overall Distance (MOD).

The MMD quantifies the average hierarchical distance between the predicted and the actual label for mispredicted instances. A lower MMD implies that, although the prediction is incorrect, the predicted label is still relatively close to the true label in terms of the hierarchical structure of the COA. This metric provides a comprehensive overview of the model’s performance in preserving the hierarchical relationships in the COA.

Let  $MD(v_i, v_j)$  be the distance of misprediction between the predicted vertex  $v_i$  and the true vertex  $v_j$ , which is defined as the length of the shortest path between  $v_i$  and  $v_j$  in the COA tree  $T$ . If the prediction is correct, i.e.,  $v_i = v_j$ , then  $MD(v_i, v_j) = 0$ . Otherwise,  $MD(v_i, v_j) > 0$ .

The MMD is then defined as the average misprediction distance over all mispredicted instances in the test set, calculated as follows:

$$MMD = \frac{\sum_{v_i \neq v_j} MD(v_i, v_j)}{|\{(v_i, v_j) | v_i \neq v_j\}|} \quad (5)$$

where  $\{(v_i, v_j) | v_i \neq v_j\}$  is the set of mispredicted instances.

Similarly, the Mean Overall Distance (MOD) can be defined, which also takes into account correctly predicted instances:

$$MOD = \frac{\sum_i MD(v_i, v_j)}{|\{(v_i, v_j)\}|}, \quad (6)$$

where  $\{(v_i, v_j)\}$  is the set of all instances.

## 5.4 Experiment 1: Performance Evaluation of TopoLedgerBERT and Impact of Negative Sampling

In this experiment, we aim to investigate two main research questions:

1. How does the TopoLedgerBERT model perform in comparison to other established methods?
2. Does increasing the number of negative samples in the augmented dataset  $D_{aug}$  impact the performance of the TopoLedgerBERT model?

For the second question, we experiment with different values of  $K$ , where  $K$  represents the number of extra negative samples generated per positive sample in  $D_+$ . The values we consider for  $K$  are  $\{5, 10, 15, 20\}$ . We denote the different TopoLedgerBERT models trained on augmented datasets with different negative sample sizes by TopoLedgerBERT@ $K$ .

Metric	Acc	MRR
FinBERT	28.91	34.57
Standard SBERT	44.79	53.07
Fine-tuned SBERT	64.15	72.69
TopoLedgerBERT@5	65.30	72.62
TopoLedgerBERT@10	66.13	73.23
TopoLedgerBERT@15	66.80	73.73
TopoLedgerBERT@20	<b>67.01</b>	<b>73.78</b>

Table 1: Comparison of the performance of TopoLedgerBERT and benchmark models. Models are evaluated based on Acc and MRR. The best performing model for each metric is in **bold**.

The results of the empirical evaluation, summarized in Table 1, validate the superior performance of our proposed TopoLedgerBERT model over the benchmark methods in terms of accuracy and mean reciprocal rank.

The FinBERT model, despite being trained on financial data, does not perform as well as the other models. This is likely due to its token-level focus, which does not lend itself to generating meaningful sentence representations. On the other hand, the Standard SBERT and Fine-tuned SBERT models, which are sentence-transformer models, perform significantly better. This highlights the importance of sentence representation methods in the context of ledger account mapping.

When comparing the TopoLedgerBERT@ $K$  models with different values of  $K$ , we observe a general trend of increasing performance with increasing  $K$ . The best performance is seen at  $K = 20$ , where the model achieves an accuracy of 67.01% and a MRR score of 73.78%. This suggests that augmenting the training data with more diverse negative samples containing information about the

ledger account taxonomy can effectively enhance the model performance. Learning from negative samples helps the model better distinguish between different ledger accounts and enhance its mapping abilities.

However, the impact of a further increase in  $K$  on the model’s performance is not straightforward. Excessive negative samples could potentially dilute the impact of the positive samples. Further investigations could be done by testing a wider range of  $K$  values.

## 5.5 Experiment 2: Hierarchical Relationship Understanding of TopoLedgerBERT

In this experiment, we evaluate the ability of the TopoLedgerBERT model to capture hierarchical relationships between different ledger accounts. We compare it with the Fine-tuned SBERT model to determine its effectiveness. The comparison is based on two metrics: MOD and MMD.

Metric	MOD	MMD
Fine-tuned SBERT	1.44	4.41
TopoLedgerBERT@20	<b>1.02</b>	<b>3.90</b>

Table 2: Comparison of MOD and MMD for Fine-tuned SBERT and TopoLedgerBERT@20. The best results are **bold**.

As shown in Table 2, the TopoLedgerBERT@20 model outperforms the Fine-tuned SBERT model in terms of both MOD and MMD. This suggests that the TopoLedgerBERT model is more effective at capturing hierarchical relationships in the COA and accurately mapping ledger accounts, thus leading to semantically better predictions.

The MOD value for the TopoLedgerBERT@20 model is significantly lower than the value for the Fine-tuned SBERT model. This indicates that the TopoLedgerBERT@20 model’s predictions are generally closer to the true labels in the COA, including both correct and incorrect predictions.

Regarding the MMD value, the TopoLedgerBERT@20 model achieves a lower value. This suggests a more effective use of hierarchical information by the TopoLedgerBERT@20 model, which leads to more accurate predictions, even for mispredicted samples.

Subsequently, we analyze the performance of TopoLedgerBERT@20 by comparing its misprediction distance distribution with that of Fine-tuned SBERT. This analysis provides additional insights

into the model’s ability to capture and leverage the hierarchical information from the COA.

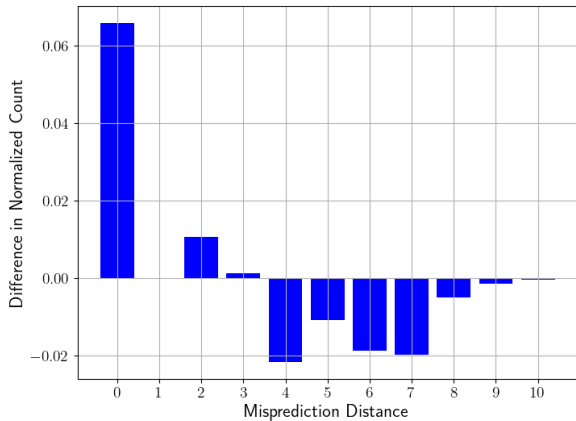


Figure 4: Misprediction Distance (MD) difference distribution between TopoLedgerBERT@20 model and Fine-tuned SBERT model.

The plot, as shown in Figure 4, shows the difference in the distribution of the MD between the TopoLedgerBERT@20 and the Fine-tuned SBERT models. A positive difference indicates that the TopoLedgerBERT@20 model has more predictions at that specific distance, while a negative difference suggests the opposite. Looking at the misprediction distance distribution difference plot, it is clear that the TopoLedgerBERT@20 model outperforms the Fine-tuned SBERT model in terms of both the count of correct predictions and the distribution of mispredictions.

TopoLedgerBERT@20 outperforms Fine-tuned SBERT by around 6% in correct predictions. While both models show a similar pattern in one-edge-away predictions, reflecting a preference for specificity, TopoLedgerBERT@20 has more two-edges-away mispredictions. This implies that TopoLedgerBERT@20 leans towards making close-by predictions.

TopoLedgerBERT@20 significantly performs better than Fine-tuned SBERT for predictions that are four edges away or more. This implies that the TopoLedgerBERT@20 model not only increases the count of correct predictions but also reduces the number of far-off predictions, hence preserving the hierarchical relationships in the COA better.

## 5.6 Discussion

The empirical evaluation shows that TopoLedgerBERT outperforms the benchmark models in terms of accuracy and mean reciprocal rank, validating its effectiveness in mapping custom ledger account

descriptions to a standardized taxonomy. The increased performance of the TopoLedgerBERT model is attributed to its innovative approach of incorporating hierarchical information from the COA into the training process and employing data augmentation strategies.

Additionally, the TopoLedgerBERT model performs better in capturing hierarchical relationships between different ledger accounts. The lower MOD and MMD scores indicate that the model can make more accurate predictions that are closer to the true labels in the COA.

The results confirm that the TopoLedgerBERT model shows a significant advance toward the ledger account mapping problem. Future research could explore further enhancements to the model, such as incorporating additional contextual information or exploring more complex data augmentation strategies.

## 6 Conclusion

In this paper, we address the significant problem of mapping custom ledger accounts to a standardized COA, a task that is crucial to improving the comparability of financial data between companies. We propose a unique method, TopoLedgerBERT, that leverages hierarchical information in the COA and fine-tuned sentence embeddings to create a mapping solution. Our empirical evaluation shows that TopoLedgerBERT not only improves the mapping accuracy, but also effectively captures the hierarchical relationships between different ledger accounts. Furthermore, the TopoLedgerBERT model shows its effectiveness in addressing the cold-start problem, generally seen when a system must predict for ledger accounts absent from the training dataset. Overcoming this issue not only solves the "multiple charts of accounts problem", as the model is not bound by a predefined set of ledger accounts or a COA, but it also paves the way for more streamlined and efficient automation of COA standardization. The empirical evaluation provides valuable insights into the proposed model’s potential benefits and areas for further improvement, ultimately contributing to improving the efficiency and precision of the accounting domain.

In the future, this work could be extended in several directions. Firstly, our proposed method can be applied to various sectors where target labels are hierarchically structured. Secondly, the incorporation of additional hierarchical informa-



tion or features may enhance the performance of the model. In addition, a detailed analysis could be performed to understand to what extent the effectiveness of our method depends on the quality and consistency of the COAs used. Third, a test of our proposed method across a larger, more diverse range of industries or countries may provide deeper insights into the model’s generalizability and robustness. Fourth, a more advanced strategy for selecting negative samples could supplement the current approach in generating more representative training samples. Lastly, incorporating evaluation metrics from the accounting domain can provide additional validation of the model’s utility in accounting applications.

## Acknowledgments

This research received funding from the Flemish government, through Flanders Innovation & Entrepreneurship (VLAIO, project HBC.2020.2883) and from the Flemish government under the program “Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen”. We also acknowledge the invaluable feedback provided by Simon De Ridder, Karen Dedecker, and Pieter De Koninck of the AI Team at Silverfin.

## References

- Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.
- Johan Bergdorf. 2018. Machine learning and rule induction in invoice processing: Comparing machine learning methods in their ability to assign account codes in the bookkeeping process.
- European Commission. 2008. Final report of the expert group: Accounting systems for small enterprises—recommendations and good practices. Luxembourg: European Commission.
- Jens-Joris Decorte, Severine Verlinden, Jeroen Van Haute, Johannes Deleu, Chris Develder, and Thomas Demeester. 2023. Extreme multi-label skill extraction training using large language models. *arXiv preprint arXiv:2307.10778*.
- Sandip Dhole, Gerald J Lobo, Sagarika Mishra, and Ananda M Pal. 2015. Effects of the sec’s xbrl mandate on financial reporting comparability. *International Journal of Accounting Information Systems*, 19:29–44.
- EUROSTAT. 2017. Epsas issue paper on the national approaches to harmonisation of chart of accounts. Working paper no. epsas wg 17/12, European Commission, Luxembourg. Paper by PwC on behalf of Eurostat.
- Susana Jorge, Giovanna Dabbicco, Caroline Aggestam-Pontoppidan, and Diana Vaz de Lima. 2022. New development: The development of standardized charts of accounts in public sector accounting. *Public Money & Management*, 42(7):530–533.
- Rasmus Kær Jørgensen and Christian Igel. 2021. Machine learning for financial transaction classification across companies using character-level word embeddings of text fields. *Intelligent Systems in Accounting, Finance and Management*, 28(3):159–172.
- Juan Liu, Lei Pei, Ying Sun, Heather Simpson, Jocelyn Lu, and Nhung Ho. 2021. Categorization of financial transactions in quickbooks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3299–3307.
- Justin Munoz, Mahdi Jalili, and Laleh Tafakori. 2022. Hierarchical classification for account code suggestion. *Knowledge-Based Systems*, 251:109302.
- Sander Noels, Simon De Ridder, Sébastien Viaene, and Tijl De Bie. 2023a. An efficient graph-based peer selection method for financial statements. *Intelligent Systems in Accounting, Finance and Management*, 30(3):120–136.
- Sander Noels, Adriaan Merlevede, Andrew Fechey, Maarten Vanhalst, Nick Meerlaen, Sébastien Viaene, and Tijl De Bie. 2023b. Automated financial analysis using gpt-4. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science and Demo Track*, pages 345–349, Cham. Springer Nature Switzerland.
- Sander Noels, Benjamin Vandermarliere, Ken Bastiaensen, and Tijl De Bie. 2022. An earth mover’s distance based graph distance metric for financial statements. In *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFER)*, pages 1–8. IEEE.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.
- Carol D Rylee. 2017. So, you need a new chart of accounts. *Planning for Higher Education*, 45(3):60.
- Richard Wang. 2023. Standardizing xbrl financial reporting tags with natural language processing. *Available at SSRN 4613085*.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Steve Yang and Randy Cogill. 2013. Balance sheet outlier detection using a graph similarity algorithm. In *2013 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFER)*, pages 135–142. IEEE.

## A Details of the Training Process

The strategies for learning sentence representations were executed using the widely used S-BERT<sup>3</sup> implementation (Reimers and Gurevych, 2019). In the multiple negative ranking loss, the 'scale' hyperparameter was kept at the default value of 20. The training was always done for 1 epoch. Positive pairs were mixed randomly into groups of 64. The AdamW optimizer was used as the default optimizer, with a learning rate of 2e-5, and a 'WarmupLinear' learning rate schedule that includes a warm-up period covering 5% of the training data. Automatic mixed precision was employed to make the training process faster. We utilized mean pooling as the pooling technique.

---

<sup>3</sup>[https://www.sbert.net/docs/package\\_reference/losses.html](https://www.sbert.net/docs/package_reference/losses.html)