A Novel Natural Language Processing Strategy to Improve Digital Accounting Classification Approach for Supplier Invoices ERP Transaction Process

Wei Wen Chi, Tiong Yew Tang(✉), Narishah Mohamed Salleh, and Ha Jin Hwang

Sunway Business School, Sunway University, 47500 Subang Jaya, Selangor, Malaysia
{tiongyewt,narishahm,hjhwang}@sunway.edu.my

Abstract. Natural language processing (NLP) is a developing field that offers increasing potential to simplify accounting-related tasks. This research studies a novel NLP approach to classify invoice categories based on the invoice text description. The preprocessing steps can be divided into three parts, namely text cleaning, semantic enrichment using the labels as an information source, and text augmentation. A total of 12 different training datasets were prepared based on the raw invoice data, each reflecting an output of a unique combination of the preprocessing steps. Each training dataset was then sent for modelling with one traditional classifier and two deep learning classifiers, namely Linear Support Vector Machine (LSVM), Bi-directional Long Short-Term Memory (Bi-LSTM) and Bidirectional Encoder Representations from Transformers (BERT). Overall, the best approach yielded an improvement of up to 6.7 percentage points (ppts) for accuracy and 20 ppts for macro F1 score. Noise and overfitting were successfully reduced when only English text was retained for modelling. Using label data to semantically enrich invoice text descriptions improved the model's generalizability. The lexical synonym substitution approach proved more effective in preserving semantics compared to the word embedding approach for short text augmentations. BERT outperformed Bi-LSTM and LSVM and performance improved further with an increase in training data, confirming the superiority of deep learning classifier performance compared to traditional classifiers. Multi-class balancing by lexical-based data augmentation improved the model generalizability, evidenced by a high macro F1 score. This novel discovery contributes to the area of automating invoice text classification, which up until today has remained largely a manual task in practice. The classification approach is well suited to be integrated with other artificial intelligence solutions like Optical Character Recognition (OCR) and Robotic Process Automation (RPA) to form a completely automated invoice processing system. Since invoice classification is a repetitive and non-value-added process, the combination of this novel text classification method with RPA can reduce overhead costs by approximately 90%.

Keywords: Deep Learning · Natural Language Processing · Machine Learning · Invoice · Robotic Process Automation · RPA · Intelligent Process Automation · IPA

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13956, pp. 581–598, 2023.
https://doi.org/10.1007/978-3-031-36805-9_38

1 Introduction

1.1 Background

The amount of unstructured enterprise data is increasing at an alarming rate of up to 65% per year, but the problem with unstructured data is that it lacks structure, making analysis and extraction of insights complex and costly [1]. The advent of data management, processing, and analysis solutions such as Hadoop and Google Colab have enabled the use of unstructured data. Unstructured data has the potential to lift enterprises above their competitors in customer engagement, operational excellence, and product leadership if used effectively [2].

Textual data is unstructured and requires structuring, so its analysis is equivalent to textual data analysis. Holsti's 1970 study described the manual classification of genuine and fake suicide texts, which led to the recommendation that machines be used due to the resource-intensive nature of manual classification. NLP is the process of transforming textual input into usable and actionable information. NLP imposes structure on textual data, allowing computers to understand it, and is critical to the text conversion process [1, 3]. Even though NLP is still a relatively new pitch in terms of applications, businesses have already used it with some success [4].

NLP can automatically extract essential risk results, reducing auditor workload and shortening the review process from half a year to a month while achieving capacity freedom [5]. Deloitte and EY used natural language processing (NLP) to automate the review of large numbers of contracts, detect changes in control provisions, extract critical accounting information from current merger and acquisition agreements, and evaluate existing leases to determine the impact on lease accounting rules [6]. NLP and deep learning are gaining research traction, but they have yet to be adopted in the finance organization.

To summarize, unstructured data presents a significant challenge for enterprises; however, the emergence of data management, processing, and analysis solutions, as well as NLP, has enabled enterprises to extract valuable insights from unstructured data, giving them a competitive advantage.

1.2 Objective

The purpose of this study is to investigate the practical applications of NLP in automating daily business tasks and lowering operational costs. NLP techniques and tools can assist machines in reading unstructured data, allowing businesses to scale their big data processing efforts while freeing up human capacity for more intelligent tasks [7]. The accounting domain's lack of automation is also a motivator, with many repetitive tasks that could be automated using NLP to avoid human error. Prior research has focused on sentiment analysis in external financial reports, but despite the presence of unstructured data in these processes, little emphasis has been placed on NLP processing for internal accounting processes such as invoice classification and bank reconciliations [8]. Deloitte and EY's successful implementation of NLP and RPA serves as a reminder of the significant benefits of a well-implemented NLP technique.

1.3 Research Approach

The research focuses on the accounting domain and aims to develop a practical and innovative method for categorizing based on Chart of Account (COA) for invoices based on their text descriptions. Short text issues and class imbalance issues, on the other hand, pose challenges in invoice classification. Traditional machine learning methods struggle to extract important features from a large feature space in short text classification, and a lack of data makes deep learning models difficult to use [9, 10]. Furthermore, most short-text vocabularies are informal and field-specific, resulting in misspellings, abbreviations, and typos. The multi-class classification of unbalanced data complicates the classification problem even more, and optimizing experiments are required to find the best classifier configurations [11].

The study suggests a new way to balance class distribution and add meaning to short text data. If this approach works, it can be used in many ways to classify text in the finance and accounting fields. The study's contribution is the creation of a novel approach to invoice text classification that has the potential to help organizations reduce overhead costs for repetitive accounting tasks. The research also aims to provide a useful technique for data augmentation that preserves semantic information while producing cutting-edge results.

The next section discusses the related work for key components of this novel approach. Section 3 discusses the methodology employed for the novel experiment. Section 4 shows the results and discussion of the experiment. Section 5 discusses the conclusion. Lastly, Sect. 6 presents the research limitations and future works are presented.

2 Literature Summary

The literature review focuses on methods and techniques for classifying text in the accounting domain, such as semantic enrichment, text data augmentation, and learning models.

2.1 Semantic Enrichment

The article discusses various methods for preserving or improving a document's semantic meaning, particularly in cases of short text like social media [12]. One common way to give more semantic information is to use extra sources like the internet or texts from previous conversations. Researchers have come up with models like Dual Latent Dirichlet Allocation (DLDA) and label embedding to deal with differences in meaning and make it easier to understand the meaning. The article talks about a new way to combine original text with augmented label data to improve semantic information and make data less sparse [13, 14]. Aside from label embedding, this method is novel and has received little attention.

2.2 Text Data Augmentation

The Data Augmentation (DA) Technique is a process that makes training data bigger without collecting more data. It does this by making different versions of real datasets. There are different techniques, including Easy Data Augmentation (EDA), which enhances original content by simple operations such as synonym substitution, arbitrary insertion, swapping, and deletion [15]. Other augmentation methods include TABAS, adversarial embeddings, and generative model-based augmentation [17]. Word embeddings convert text into numbers, and different embedding methods exist. Most research employs word embeddings to train LSTM or BERT models, but no augmentation study has been conducted in the context of invoice data, either to increase training data or to manage class imbalance [18, 19]. Some studies have shown that augmentation techniques can improve sentiment detection and text classification [17, 20].

2.3 Learning Models

The article compares traditional and deep-learning text categorization models. Traditional machine learning methods, such as Support Vector Machines (SVM), are effective in text analysis, but they are being supplanted by deep learning models such as bidirectional long short-term memory (Bi-LSTM) and bidirectional encoder representations from Transformers (BERT) [21]. Bi-LSTM can manage long context memories more efficiently than recurrent neural networks (RNNs), and BERT is a cutting-edge model that can learn bidirectionally from unlabeled text and capture context [21, 22]. Despite its benefits, BERT has limitations due to its large scale and high latency, making it unsuitable for some real-time scenarios. No study has used bi-LSTM or BERT to classify invoice text, and traditional methods such as LSVM are still effective as a baseline for comparison [23].

3 Methodology

The Cross-Industry Standard Process for Data Mining (CRISP-DM) framework was used in this section, and it will proceed in the following order: business understanding, data understanding, data preparation, modelling, and evaluation. Amani and Fadlalla [24] mentioned the use of this paradigm in data mining implementations. According to Sharda et al. (2018) [25], text mining initiatives necessitate more involved preprocessing steps than a typical data mining project. The next section explains in detail the steps and procedures:

3.1 Business Understanding

The business understanding of processing supplier invoices from capturing the information in the ERP system to classifying it to the correct chart of accounts and translating it into the financial profit and loss definition is explained. The next step is to describe the current process flow before suggesting a text classification automation approach.

The current procedure involves the accounts staff receiving a supplier invoice and entering it into the ERP system before making payment. The staff summarizes the invoice

details in the invoice description, identifies the associated account code, enters it, and saves the data into the ERP system. The invoice classification process flow is shown in Fig. 1, which includes several human-dependent steps, as depicted in Fig. 2.

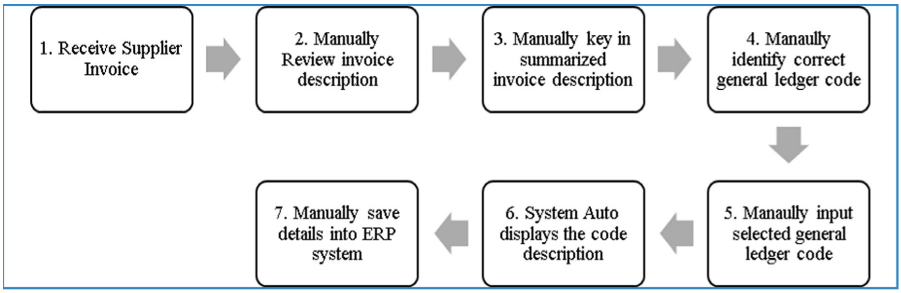

Fig. 1. "As-Is" Invoice Classification Process Flow

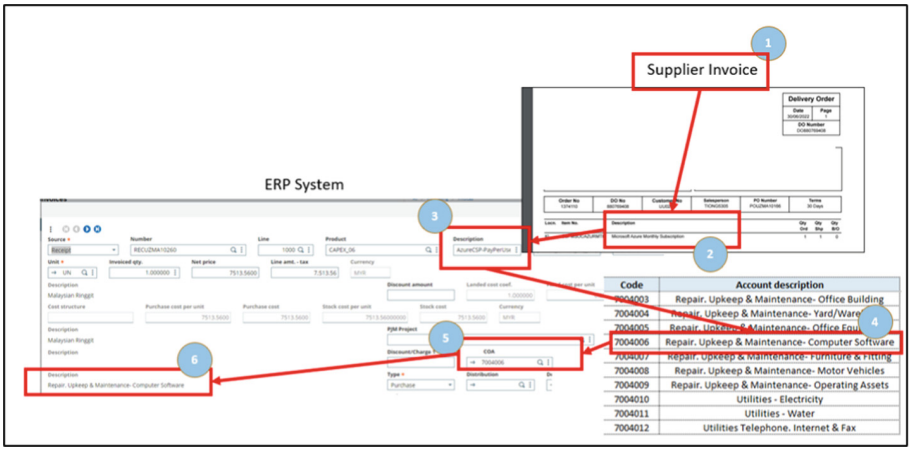

Fig. 2. Visualization of the "As-Is" Invoice Classification Steps in the ERP system.

The ERP interface and its relevant fields for entering supplier invoice information are displayed in Fig. 2. The labelled fields are required to be stored in the database for reporting purposes, which is crucial for financial reporting analysis by the finance team. Therefore, comprehending the process flow and relevant steps of the ERP User Interface can help identify areas for automation improvement.

Steps 2 through 6 have the potential to be automated, with steps 2–3 and 5–6 using RPA technology. Step 4 involves the automatic classification of the account code based on the invoice text description obtained, which will be explained in the following sections outlining the methodology for the text classifier scope. Figure 3 shows the proposed process steps for the invoice classification which will be done by RPA.

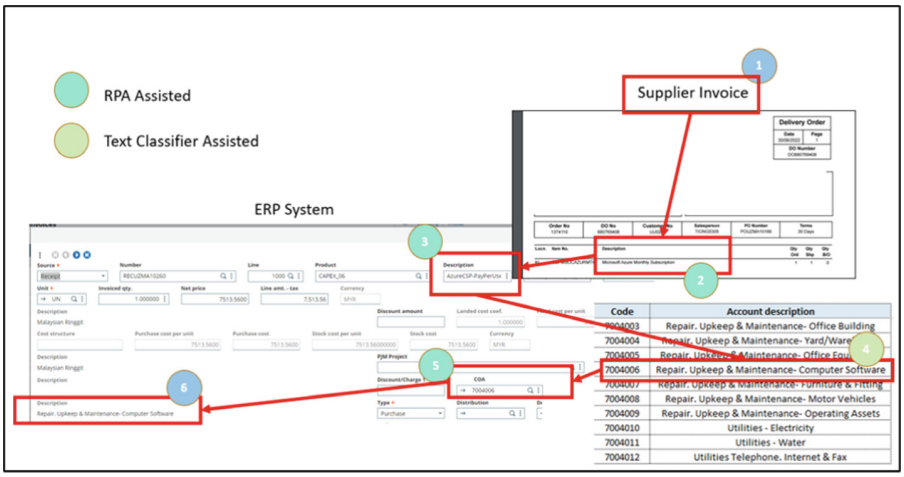

Fig. 3. "To-Be" Invoice Classification Process Steps which will be handled by RPA and a new assisted algorithm.

3.2 Data Understanding

The data understanding phase is a crucial part of the text classifier process, which is necessary for data preparation before deploying the text classification algorithm. This phase involves understanding the origin of the data, how it is stored in the ERP database, and the relationship of the master data in retrieving the source of COA code into the ERP interface.

The data is generated from the invoice process and stored in the ERP system. The raw dataset, consisting of 13,933 rows and 2 columns, was generated from this source between 2019 and 2021, with a total of 77 classes. The data definitions are presented in Table 1.

Table 1. Invoice Data Description

Data	Data Type	Description
Description	String	The invoice text description
Label	String	The general ledger code description

Initial data observations revealed the presence of uppercase letters, special characters, digits, and reference numbers, which should be removed as they are noise and negatively impact the modelling process. Stop words were also found and should be removed. Homogeneous class labels with slight differences, named entities, abbreviations, and misspellings were also detected. Invoices descriptions appearing for multiple classes will be removed from the dataset. The data is imbalanced due to the multi-class nature of the dataset, as shown in Fig. 4, which will help prioritize text classification automation development.

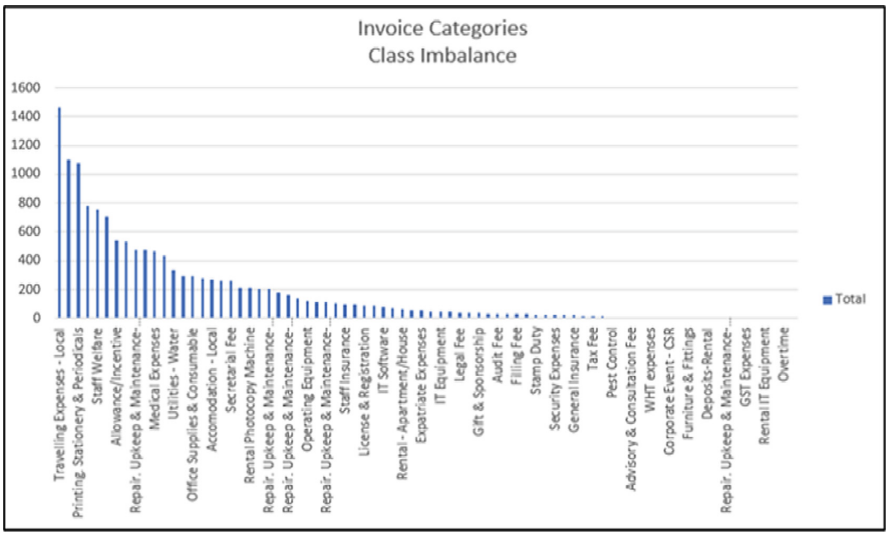

Fig. 4. The Invoice Categories Class Imbalance

From the data preparation phase up to the evaluation phase, the scope is based on the overall framework displayed in Fig. 5. Briefly, the major steps taken in these phases include text cleaning, sentiment enrichment, text augmentation, text categorization and classification, and finally evaluation. Most of the operational steps were performed using Python.

3.3 Data Preparation

The data preparation process involves text cleaning, semantic enrichment, and text augmentation of the original raw dataset named 'D1'. After text cleaning, the dataset 'D2' is split into train and test datasets and then used for semantic enrichment to produce the 'D3' dataset. 'D3_train' is further augmented using WordNet, GloVe, and BERT to create three training datasets of incremental sizes. At the end of the process, there will be 12 unique sets of training data and 3 unique sets of test data, which are paired accordingly. The details of the train-test pairing sets are presented in Table 2 while Table 3 summarizes the techniques applied sequentially.

The total number of datasets produced from this phase will be 12 datasets. Each data set contains a training set, a validation set and a test set. See Fig. 6.

The "Text Representation and Classification" phase falls under "Modeling." Here, each dataset is subjected to three classifiers, resulting in a total of 36 outputs. The "Evaluation" step concludes the CRISP-DM framework's evaluation phase, where the 36 results are evaluated. Python is the primary analysis tool employed in this study. To utilize their available graphics processing unit (GPU), the Python codes were executed on Google Colab and Kaggle.

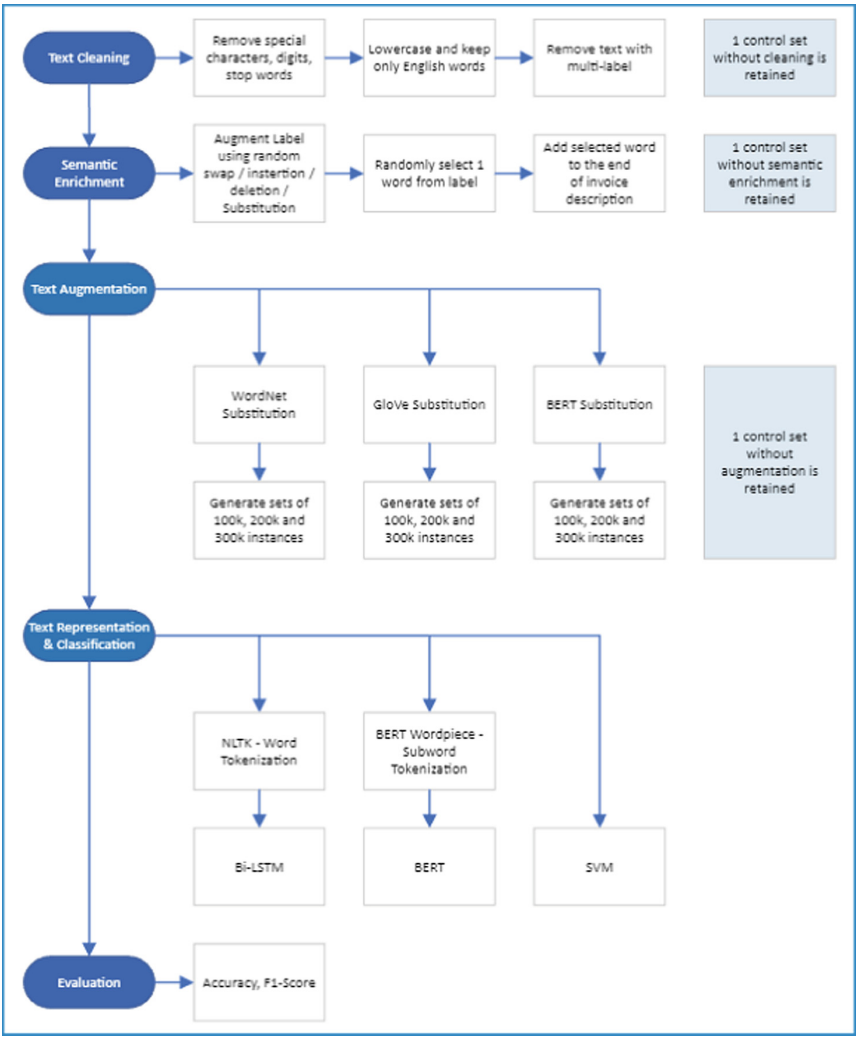

Fig. 5. The Overview Framework

3.4 Experimental Design

In this modelling phase, the three classifiers involved are LSVM, Bi-LSTM and BERT. The total result outputs from this phase will be 36 outputs. Table 4 Summarized the setting for the experiments:

In this phase, the 36 outputs produced will be evaluated using the accuracy and macro F1-score. The macro F1 scores are the average F1 scores for all classes. This score does not consider the weightage of each class and is a good indicator of whether the F1 scores are balanced across all the classes without accounting for weightage. A high score will indicate that the model generally is not biased towards any class. The weighted average F1 scores are also F1 scores but already accounted for the weightage.

Table 2. Train-Test Pairing Datasets

Set	Training Set	Testing Set
D01	D1_train	D1_test
D02	D2_train	D2_test
D03	D3_train	D3_test
D04	D3_WNtrain100k	D3_test
D05	D3_GLtrain100k	D3_test
D06	D3_BTtrain100k	D3_test
D07	D3_WNtrain200k	D3_test
D08	D3_GLtrain200k	D3_test
D09	D3_BTtrain200k	D3_test
D10	D3_WNtrain300k	D3_test
D11	D3_GLtrain300k	D3_test
D12	D3_BTtrain300k	D3_test

Table 3. Summary of Data Preparation Process

Technique	Summary of the process
1.0 Text Cleaning	It is the process of removing special characters, digits, stop words, and single characters to keep only English words. Entirely blank rows and multi-labelled instances are removed, except when a minority label has only one sample, in which case it is reclassed to the majority label. The dataset is split into train-validation-test sets
2.0 Semantic Enrichment	A new label column is generated with augmented text using random swap, deletion, or insertion, and then tokenized. One random word is selected from the tokens, and a new column is generated by concatenating it with the original description text to create semantically enriched synthetic instances, and a new dataset is generated by appending it to the "D2" train set to create a "D3" train set with double the training data. The "D3" train set is split into train validation, and the test set of "D2" is used to test the model trained with the "D3" train set
3.0 Text Augmentation	This is to examine the impact of larger training data sets on classification performance. The required number of augmented samples is determined to reach approximately 300,000 samples with equal distribution for WordNet, GloVe, and BERT substitution. The "D3" train dataset is loaded, and instances are augmented iteratively to achieve approximately 300,000 samples for each substitution

Text Cleaning	Semantic Enrichment	Increase Train Volume		Dataset Code
None				D01
Yes	None			D02
	Yes	None		D03
		100000	WN	D04
			GLV	D05
			BERT	D06
		200000	WN	D07
			GLV	D08
			BERT	D09
		300000	WN	D10
			GLV	D11
			BERT	D12

Fig. 6. Total Datasets Produced from Data Preparation Phase

This score is closer to the overall accuracy score. 80% was used as a benchmark for good classification performance.

4 Results

The detailed results from the modelling stage are collated in Fig. 7 based on Accuracy and F1-Score:

Figure 8 displays the visualization chart to compare Accuracy and Macro F1 scores by Dataset Codes and Classifier.

The results show that the top three accuracy scores were from datasets D04, D07 and D10, all of which were WordNet-augmented datasets trained with the BERT classifier. The top three Macro-F1 scores were driven by datasets D10, D11 and D12, all of which were characterized by a dataset of 300,000 instances generated by WordNet, GloVe and BERT augmenters respectively, and trained with the BERT classifier. The BERT classifier achieved above 80% accuracy for all datasets and is the best classifier. The results also show BERT having the lowest Macro F1 scores for non-augmentation datasets but becoming the best performer for all the augmented datasets. For pre-augment observations, dataset D03, which was semantically enriched but without augmentation, significantly improved classifier performance across all three classifiers compared to dataset D02 which was not semantically enriched. The D03 dataset also performed better than some of the augmented datasets. The biggest improvement in accuracy scores can be seen in the Bi-LSTM experiment with an improvement of 6.7 percentage points (ppts) while the biggest improvement in Macro F1 scores can be observed in the BERT experiment with an improvement of up to 20 ppts. Overall, the deep learning classifiers BERT and Bi-LSTM performed better than LSVM.

Table 4. Summary of Experiment's Configuration

No	LSVM	Bi-LTSM	BERT
1	Default Setting	The training and testing set is loaded, labelled as X_train, X_test, y_train, and y_test	Split Train set into training and validation set (15% of Train set)
2	Training set and test data uploaded	The train set text is tokenized using NLTK library word-based tokenization	The Train, Validation and Test sets are loaded, and labels are encoded
3	12 Experiments	Convert the train and test description into a sequence using a token	Replace Train, Validation and Test Sets with the encoded labels
4		Pad all the sequences so all vectors are of the same length. Create and Embedded the layer	Load BERT Tokenizer. The Tokenizer is based on pre-trained 'Bert Base Uncased'. Encode Training, Validation and Testing Sets
5		Build and Compile the LSTM model. The learning rate is set to 0.01	Load Pretrained Model. Load Data loaders, Optimizer and Scheduler. The learning rate is set at 1e-5 and batch size 16 to cater for lower memory processing
6			Define evaluation function Run the Training Loop. Manual early stopping is done when the validation loss increases 3 times from the last lowest validation loss measure
7			Run the full evaluation

Reference Codes:
a. Full LSVM codes at https://github.com/cwwdaniel/invoice-text-classification/blob/main/lsvm
b. Full Bi-LSTM codes at https://github.com/cwwdaniel/invoice-text-classification/blob/main/bi-lstm
c. Full BERT codes at https://github.com/cwwdaniel/invoice-text-classification/blob/main/bert

Text Cleaning	Semantic Enrichment	Increase Train Volume		Dataset Code	LSVM Accuracy	Bi-LSTM Accuracy	BERT Accuracy	LSVM Macro F1-Score	Bi-LSTM Macro F1-Score	BERT Macro F1-Score
None				D01	77.9%	78.3%	81.1%	67.0%	61.0%	61.0%
Yes	None			D02	84.4%	83.2%	84.1%	75.0%	68.0%	63.0%
	Yes	None		D03	84.6%	85.7%	86.4%	76.0%	78.0%	73.0%
		100000	WN	D04	81.4%	83.8%	86.7%	72.0%	78.0%	75.0%
			GLV	D05	79.6%	81.3%	84.2%	67.0%	71.0%	77.0%
			BERT	D06	78.5%	80.2%	82.8%	67.0%	68.0%	75.0%
		200000	WN	D07	82.9%	84.7%	86.9%	74.0%	76.0%	78.0%
			GLV	D08	80.6%	83.8%	86.1%	69.0%	74.0%	78.0%
			BERT	D09	79.3%	83.1%	85.0%	69.0%	71.0%	76.0%
		300000	WN	D10	82.7%	85.0%	87.7%	74.0%	75.0%	81.0%
			GLV	D11	80.4%	81.8%	86.6%	68.0%	70.0%	79.0%
			BERT	D12	80.0%	78.5%	86.6%	70.0%	66.0%	79.0%

Fig. 7. Model Output Results. Top performers are highlighted in green. (Color figure online)

Figure 9 and Fig. 10 display the visualization chart to compare Accuracy and Macro F1scores by Augment Volume and Classifier or Augmenter.

Figure 9 shows that the BERT classifier consistently achieved accuracy scores of more than 80% on average at different augmentation volumes in an increasing trend. Bi-LSTM and LSVM trended inconsistently with the increase of synthetic data, ultimately trending downwards or flat at the 300,000-instance dataset. Macro F1 scores trended upwards for the BERT classifier, and just slightly short of 80% at 300,000 augmented instances. A significant improvement was observed in BERT classifier Macro-F1 results with the 100,000-instance dataset over the non-augmented results. It is also observed in Fig. 10 that the WordNet augmenter has the best performance among the augmenters, also increasing in performance as the augmented volume increases. The GloVe and BERT augmenter performance had similar trends where the non-augmented results were superior to the augmented dataset results.

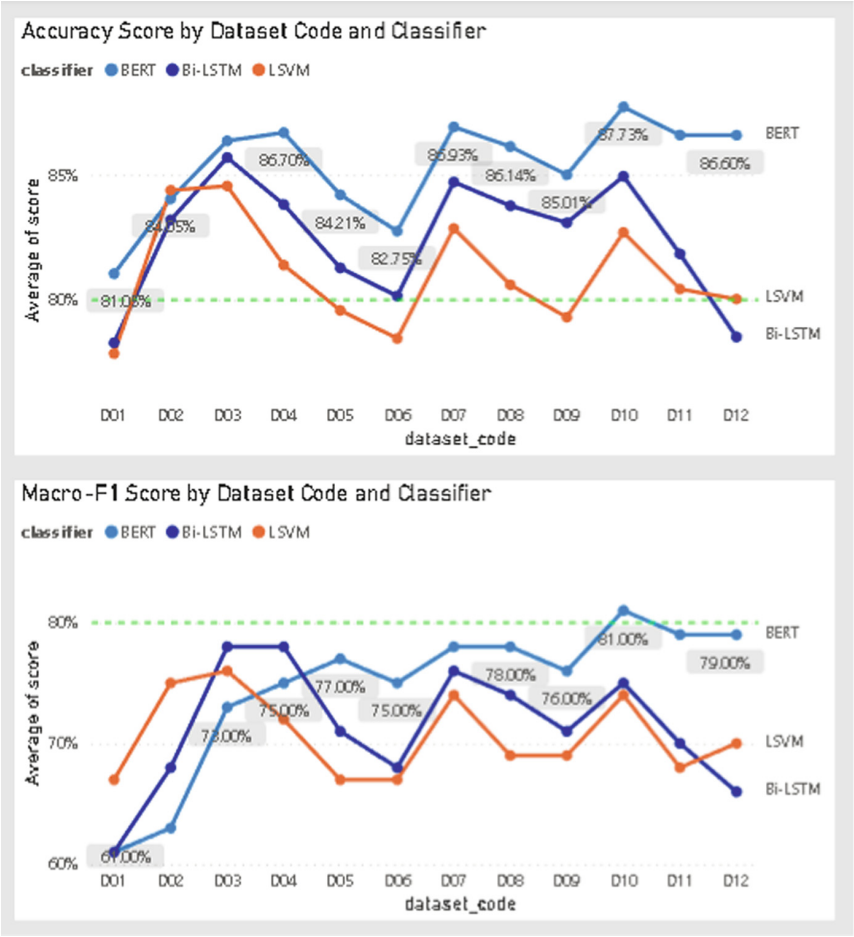

Fig. 8. Accuracy & Macro F1 scores by Dataset Code and Classifier

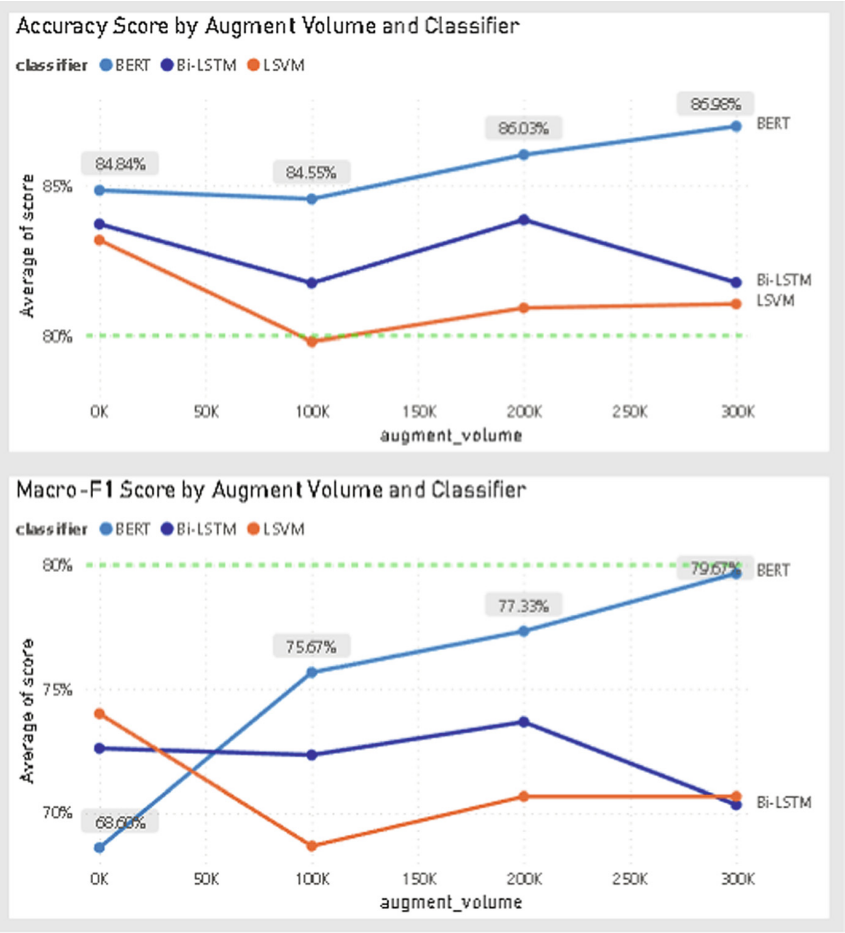

Fig. 9. Accuracy & Macro F1 scores for different levels of augmentation volume by Classifier

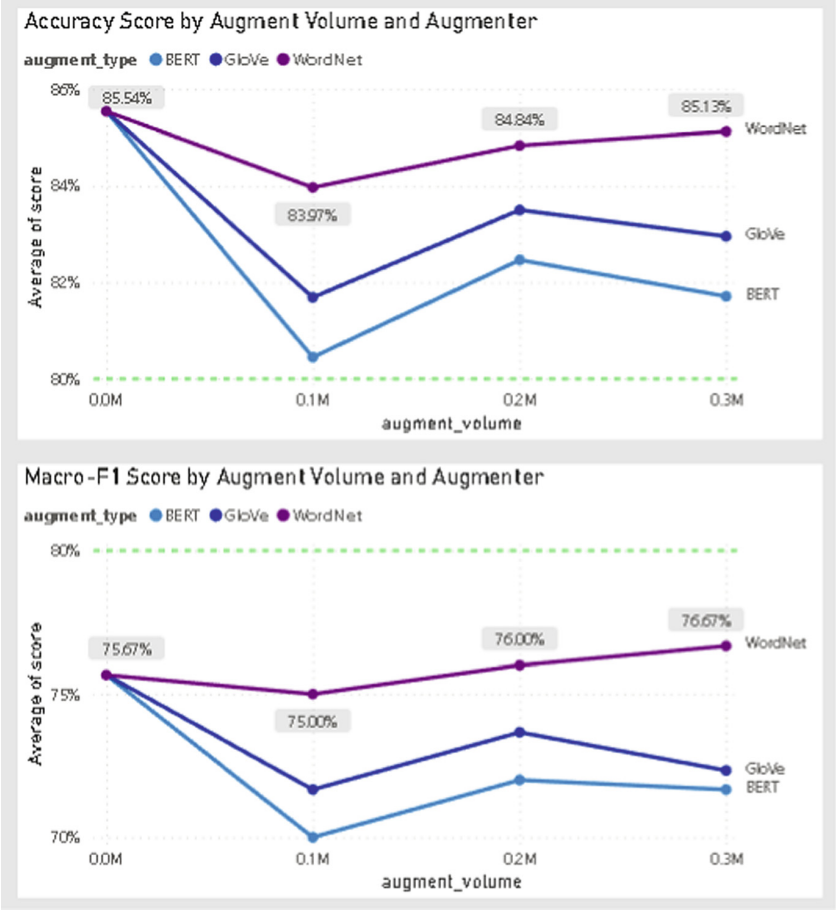

Fig. 10. Accuracy & Macro F1 scores for different levels of augmentation volume by Augmenter

Figure 11 displays the visualization chart to compare the augmenter-classifier combination Accuracy and Macro-F1 scores in descending order.

The data shows that the BERT method consistently achieves accuracy scores of more than 80% for each augmenter-classifier. The top performer is the BERT-WordNet augmenter-classifier. The BERT classifier and WordNet augmenter were frequently found in the top five augmenter-classifiers in terms of performance, indicating the significance of these two techniques on the performance of the classification task. Consistently It is observed that the larger difference in performance for augmenter-classifiers was in the Macro F1 scores rather than the accuracy.

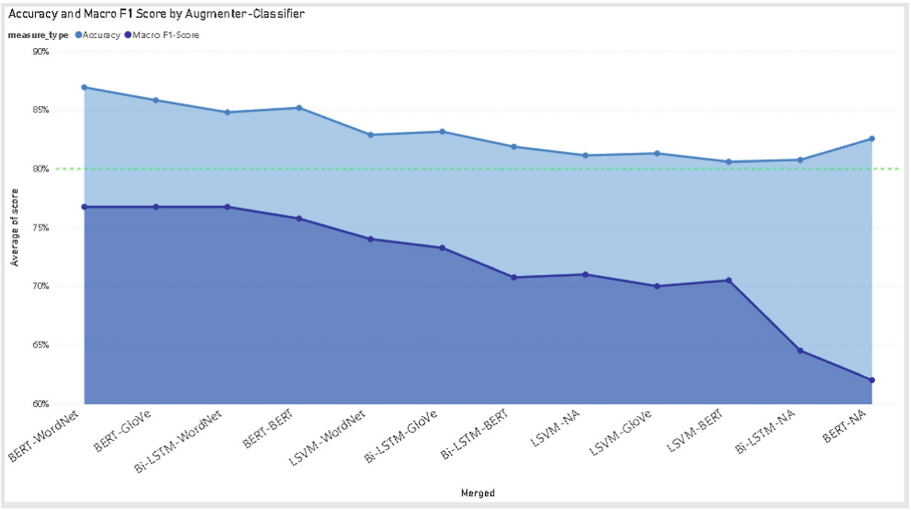

Fig. 11. Accuracy and Macro F1 Scores for the different augmenter-classifier combinations by descending order of combined scores.

5 Discussions and Conclusions

The passage talks about what happens to data before and after it is cleaned, as well as how semantic enrichment affects how well text classification works. The pre-cleaning baseline results revealed that the models were heavily biased towards majority classes, resulting in lower macro-F1 scores for minority classes. However, post-cleaning increased overall accuracy by up to 7%, indicating that removing noise and industry-specific terms from raw text improves classification performance.

Post-semantic enrichment improved accuracy by up to 2 percentage points, but more importantly, it greatly improved macro F1 scores for minority classes. This shows that additional semantic information from labels helps reduce data sparsity and improve text classification performance. The results support the idea that it helps classification performance to get additional semantic information from labels.

The objective of the research was to improve how invoices are classified by text by using semantic enrichment and data augmentation strategies. The results showed that minority classes were more likely to be misclassified than majority classes. However, once the data was cleaned up, the accuracy of classification went up to more than 80%. It was found that using BERT as a classifier and the WordNet synonym augmentation as a data enlargement were both good ways to classify short texts. The findings of this study have several implications for corporate practice, one of which is the reduction of overhead costs associated with routine and repetitive accounting procedures. The unique method of short-text semantic enrichment that was utilized in this investigation is an approach that has the potential to add to the expanding body of research that focuses on semantically enhancing short-text data. In conclusion, the findings of the study demonstrated that an effective augmentation strategy that maintains semantic information can produce state-of-the-art outcomes and can make a significant contribution to the research community.

6 Limitations and Future Works

In this study, only one traditional learner was used to compare against deep learning models. It is recommended that more traditional learners be compared against the deep learning model for more concrete findings on the question of the effectiveness of traditional versus deep learning models.

Acknowledgement. This work was supported in part by Sunway University and Sunway Business School under Kick Start Grant Scheme (KSGS) NO: GRTIN-KSGS-DBA[S]-02-2022. This work is also part of the Sustainable Business Research Cluster and Research Centre for Human-Machine Collaboration (HUMAC) at Sunway University. We also wish to thank those who have supported this research.

References

1. Sharda, R., Delen, D., Turban, E., Aronson, J. E., Liang, T.-P., King, D.: Business Intelligence, Analytics, and Data Science: A Managerial Perspective (Fourth). Pearson (2018)
2. Taylor, C. Structured vs Unstructured Data. Datamation. https://www.datamation.com/big-data/structured-vs-unstructured-data/. Accessed 21 May 2021
3. Guo, L., Shi, F., Tu, J: Textual analysis and machine learning: crack unstructured data in finance and accounting. J. Finance Data Sci. **2**(3), 153–170 (2016)
4. Zhou, Y., Cui, S., Wang, Y.: Machine learning based embedded code multi-label classification. IEEE Access **9**, 150187–150200 (2021)
5. Zhang, Y., Xiong, F., Xie, Y., Fan, X., Gu, H.: The impact of artificial intelligence and blockchain on the accounting profession. IEEE Access **8**, 110461–110477 (2020)
6. Li, L., Feng, Y., Lv, Y., Cong, X., Fu, X., Qi, J.: Automatically detecting peer-to-peer lending intermediary risk - top management team profile textual features perspective. IEEE Access **7**, 72551–72560 (2019)
7. Baviskar, D., Ahirrao, S., Potdar, V., Kotecha, K.: Efficient automated processing of the unstructured documents using artificial intelligence: a systematic literature review and future directions. IEEE Access **9**, 72894–72936 (2021)
8. Korhonen, T., Selos, E., Laine, T., Suomala, P.: Exploring the programmability of management accounting work for increasing automation: an interventionist case study. Acc. Audit. Accountability J. **34**(2), 253–280 (2021)
9. Samant, S.S., Bhanu Murthy, N.L., Malapati, A.: Improving term weighting schemes for short text classification in vector space model. IEEE Access **7**, 166578–166592 (2019)
10. Balakrishnan, V., Shi, Z., Law, C.L., Lim, R., Teh, L.L., Fan, Y.: A deep learning approach in predicting products' sentiment ratings: a comparative analysis. J. Supercomput. **78**(5), 7206–7226 (2021). https://doi.org/10.1007/s11227-021-04169-6
11. Garcia-Mendez, S., Fernandez-Gavilanes, M., Juncal-Martinez, J., Gonzalez-Castano, F.J., Seara, O.B.: Identifying banking transaction descriptions via support vector machine short-text classification based on a specialized labelled corpus. IEEE Access **8**, 61642–61655 (2020)
12. Mehanna, Y.S., Mahmuddin, M.B.: A semantic conceptualization using tagged bag-of-concepts for sentiment analysis. IEEE Access **9**, 118736–118756 (2021)
13. Subedi, B., Sathishkumar, V.E., Maheshwari, V., Kumar, M.S., Jayagopal, P., Allayear, S.M.: Feature learning-based generative adversarial network data augmentation for class-based few-shot learning. Math. Probl. Eng. **2022**, 1–20 (2022)

14. Xiang, R., Chersoni, E., Lu, Q., Huang, C.R., Li, W., Long, Y.: Lexical data augmentation for sentiment analysis. J. Am. Soc. Inf. Sci. **72**(11), 1432–1447 (2021)
15. Wan, C., Wang, Y., Liu, Y., Ji, J., Feng, G.: Composite feature extraction and selection for text classification. IEEE Access **7**, 35208–35219 (2019)
16. Wang, J., Li, Y., Shan, J., Bao, J., Zong, C., Zhao, L.: Large-scale text classification using scope-based convolutional neural network: a deep learning approach. IEEE Access **7**, 171548–171558 (2019)
17. Luo, J., Bouazizi, M., Ohtsuki, T.: Data augmentation for sentiment analysis using sentence compression-based SeqGAN with data screening. IEEE Access **9**, 99922–99931 (2021)
18. Liu, C.-L., Fink, G.A., Govindaraju, V., Jin, L.: Special issue on deep learning for document analysis and recognition. Int. J. Doc. Anal. Recogn. (IJDAR) **21**(3), 159–160 (2018). https://doi.org/10.1007/s10032-018-0310-5
19. Somayajula, S.A., Song, L., Xie, P.: A multi-level optimization framework for end-to-end text augmentation. Trans. Assoc. Comput. Linguist. **10**, 343–358 (2022)
20. Tan, K.L., Lee, C.P., Lim, K.M., Anbananthen, K.S.M.: Sentiment analysis with ensemble hybrid deep learning model. IEEE Access **10**, 103694–103704 (2022)
21. Yan, C., Chen, Y., Zhou, L.: Differentiated fashion recommendation using knowledge graph and data augmentation. IEEE Access **7**, 102239–102248 (2019)
22. Lee, S., Liu, L., Choi, W.: Iterative translation-based data augmentation method for text classification tasks. IEEE Access **9**, 160437–160445 (2021)
23. El-Alami, F.-Z., El Alaoui, S.O., En Nahnahi, N.: Contextual semantic embeddings based on fine-tuned AraBERT model for Arabic text multi-class categorization. J. King Saud Univ.-Comput. Inf. Sci. **34**(10), 8422–8428 (2022)
24. Amani, F.A., Fadlalla, A.M.: Data mining applications in accounting: a review of the literature and organizing framework. Int. J. Acc. Inf. Syst. **24**, 32–58 (2017)
25. Sharda, R., Delen, D., Turban, E.: Business Intelligence, Analytics, and Data Science: A Managerial Perspective. Pearson (2017)