Science and Technology and Social Development Proceedings Series

Article

2024 International Conference on Business Economics, Education, Arts and Social Sciences (EASS 2024)

Application of LSTM-Based Seq2Seq Models in Natural Language to SQL Conversion in Financial Domain

Hanqin Zhang [1,*]

[1] University of Toronto, Toronto, Ontario, Canada

[*] Correspondence: Hanqin Zhang, University of Toronto, Toronto, Ontario, Canada

Abstract: As a crucial branch of artificial intelligence, Natural Language Processing (NLP) enables computers to understand, interpret, and generate human language, significantly enhancing the efficiency of information retrieval and search. Given the growing demand for data processing in the financial sector, this paper proposes and implements a Seq2Seq model based on the LSTM algorithm to convert natural language queries into SQL statements (NL2SQL) for application in finance. The model demonstrates stable and significant performance improvements over 10 training epochs, with accuracy increasing from 0.75 to 0.9877 and the loss value decreasing from 1.5 to 0.4978. These results validate the accuracy and effectiveness of the proposed LSTM-based Seq2Seq model in handling NLP tasks within the financial domain.

Keywords: LSTM; Seq2Seq; Natural Language Processing; SQL; financial data analysis

Published: 10 November 2024

1. Introduction

Natural Language Processing (NLP), as a significant branch of artificial intelligence, is primarily used to enable computers to understand, interpret, and generate human language. The application of NLP can significantly enhance the efficiency of information retrieval and search, allowing computers to better comprehend user queries and deliver more relevant results. Moreover, NLP is widely applied in data analysis, aiding researchers in extracting valuable insights from unstructured data by analyzing vast amounts of text. NLP plays an indispensable role in improving the naturalness and effectiveness of human-computer interaction [1].

Given the complexity of the financial industry and the critical role of data in decision-making, the demand for efficient data processing in the financial sector is growing rapidly [2]. The dynamic nature and complexity of financial data [3] require systems that can efficiently process and analyze data in real-time to enable timely decision-making. NLP applications are well-suited to meet these data processing demands in finance. For instance, NLP techniques can help businesses extract key information from vast amounts of documents, such as stock prices and financial statements, thereby enhancing the efficiency of financial analysis and reducing the time spent on manual processing.

In this context, this paper proposes a Seq2Seq model based on the LSTM algorithm, which accurately converts natural language queries into SQL statements (NL2SQL) for application in the financial domain. The Seq2Seq model, a deep learning architecture widely used in tasks such as machine translation, is capable of generating one sequence through specific method from another given sequence, even when the two sequences are of different lengths. By applying this model, we aim to significantly improve the accuracy and efficiency of financial data queries, thereby providing robust support for financial analysis and decision-making.

2. Related Work

In recent years, the development of big data and new artificial intelligence technologies has made natural language to SQL (NL2SQL) conversion a prominent research problem in the field of human-computer interaction. One of the main challenges is the complexity and variability of natural language descriptions, which has attracted significant attention and research in the academic community. The first large-scale cross-domain dataset, WikiSQL [4], was introduced by Salesforce in 2017 and remains the largest annotated NL2SQL dataset to date. Based on this dataset, researchers have developed many seminal models. For example, the TypeSQL model [5] achieved a 68% exact match accuracy, while the X-SQL model [6] reached an SQL execution accuracy of 91.8%. However, a notable limitation of this dataset is the simplicity of the SQL queries, where each query corresponds to only a single database table, making it less suitable for real-world applications. To address this gap, Yale University released the Spider dataset [7] at the end of 2018, a complex multi-table query dataset that has become a dominant benchmark in the English NL2SQL domain, significantly raising the overall difficulty of NL2SQL tasks.

The rapid progress in NL2SQL research in the English-speaking world has drawn the attention of the Chinese academic community. Due to the complexity of the Chinese language, NL2SQL presents even greater challenges in the Chinese context. To accelerate the development and application of NL2SQL technology in Chinese scenarios, several Chinese NL2SQL datasets have been introduced. The first Chinese NL2SQL dataset, TableQA [8], was released in 2019 by the Alibaba Cloud Tianchi Big Data Competition. This dataset consists of 49,974 entries and involves only single-table simple queries. Subsequently, Min et al. translated the Spider dataset into Chinese and released the CSpider dataset [9] to meet the demands of complex multi-table queries.

Today, thanks to the continuous advancements in deep learning technology, neural network models have shown great promise in addressing the challenges posed by the complexity and variability of natural language descriptions in NL2SQL tasks. Early research typically treated NL2SQL as a sequence generation problem and employed Seq2Seq models [10] to handle it. Later studies have improved upon the Seq2Seq model to achieve better prediction results. For instance, Li et al. introduced the Seq2Seq+Attention model [11] by incorporating attention mechanisms, which significantly enhanced the accuracy of SQL generation. Wang et al. proposed the Seq2Seq+Copying model [12], which utilizes a copying mechanism to accurately identify and directly map parts of the input natural language description—such as column names, strings, and numerical values—to the corresponding components of the output SQL statement. However, a critical limitation of standard Seq2Seq models is their failure to account for the specific formatting and syntactical rules of SQL, often resulting in syntactically incorrect SQL outputs.

Existing NL2SQL models and methods still face challenges in fully addressing all scenarios. Early rule-based and template-matching methods, while simple to implement, lack flexibility and struggle with complex queries and diverse natural language expressions. Statistical learning approaches, though capable of learning language transformation rules from data, require large amounts of annotated data and involve complex feature engineering. Meanwhile, the current mainstream deep learning methods, despite their ability to handle complex language structures and their strong generalization capabilities, demand substantial computational resources and training data. They also face challenges in processing long sentences and complex logic. To address these issues, this paper proposes a Seq2Seq model based on the LSTM algorithm, aimed at overcoming these challenges.

3. Dataset and Methodology

The ANTSQL1.0 dataset originates from the Alibaba Cloud Tianchi Data Competition [13], provided by Ant Fortune. It aims to advance the development of Chinese natural language processing (NLP) technology within the financial sector and to promote research in digital finance. This dataset comprises nearly 80,000 standardized table entries focused primarily on single-table queries within the financial domain. The query-SQL pairs in the dataset are annotated based on real user inquiries, reflecting strong financial and conversational characteristics.

Each record in the dataset consists of the following components:

1) **id**: A unique identifier for each record.
2) **question**: The natural language question posed by the user.
3) **table_id**: The dataset to which the record belongs (in this dataset, all questions target the same dataset, *Fundtable*).
4) **sql**: The SQL query statement corresponding to the user's question.

The target of this dataset is to map the natural language question ("question") to the corresponding SQL query ("sql"), representing a natural language processing task within the financial domain. For example, in the dataset, a natural language query like "Find the fund with the highest yield" serves as the input feature, while the corresponding SQL query is the target feature. For this particular query, the corresponding SQL statement would be: SELECT * FROM funds WHERE yield = (SELECT MAX(yield) FROM funds).

4. Seq2Seq Model Based on LSTM Algorithm

In this section, we provide a detailed description of the construction and implementation of the Seq2Seq model based on the LSTM algorithm, which is designed to convert natural language queries into SQL queries for execution in a database.

4.1. Model Architecture Overview

The Seq2Seq (Sequence-to-Sequence) model is a deep learning model used for sequence-to-sequence transformation tasks, commonly applied in machine translation, text summarization, and text generation. The Seq2Seq model typically comprises two main components: the encoder and the decoder. The encoder is responsible for encoding the input sequence into a fixed-length context vector, while the decoder decodes this context vector into the target sequence. In this study, we employ an LSTM (Long Short-Term Memory) network [14] to construct the Seq2Seq model, which handles the task of converting natural language queries into SQL queries.

LSTM is a specialized type of recurrent neural network (RNN) designed to handle and predict long-term dependencies in sequential data. Traditional RNNs often struggle with vanishing or exploding gradients when processing long sequences, making it difficult for the model to retain information from earlier inputs. LSTM overcomes this issue by introducing gate mechanisms (such as input gate, forget gate, and output gate), allowing the model to selectively remember or forget information during training, thereby effectively capturing long-range dependencies.

The LSTM cell Equations:

Forget Gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

The forget gate (f_t) decides what information to discard from the cell state. It takes the previous hidden state (h_{t-1}) and the current input (x_t) as inputs and passes them through a sigmoid function (σ) to output a value between 0 and 1.

Input Gate:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\overline{C}_t = \tanh（W_c \cdot [h_{t-1}, x_t] + b_C） \tag{3}$$

The input gate (i_t) decides what new information to store in the cell state. It combines the previous hidden state and the current input, applies a sigmoid activation to control what to update, and generates candidate values ($\bar{C}_t$) using a tanh function.

Cell State Update:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \overline{C}_t \tag{4}$$

The cell state (C_t) is updated by combining the previous cell state (C_{t-1}) scaled by the forget gate and the new candidate values scaled by the input gate.

Output Gate:

$$\begin{aligned} \mathrm{o}_t &= \sigma（W_O \cdot [h_{t-1}, x_t] + b_o） \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \tag{5}$$

The output gate (o_t) determines the next hidden state (h_t). It combines the current cell state passed through a tanh function with the output of the sigmoid function.

These equations illustrate how LSTM cells selectively retain, update, and output information, allowing the network to capture long-term dependencies effectively.

4.2. Encoder

The encoder's task is to transform the input natural language query sequence into a fixed-length context vector. The encoder consists of an LSTM layer that sequentially reads each word in the input sequence and generates the corresponding hidden state vector. The hidden state vector from the last step serves as the context vector, which is passed to the decoder.

4.3. Decoder

The decoder's task is to convert the context vector generated by the encoder into the target SQL query sequence. The decoder consists of an LSTM layer and a fully connected layer (Dense). At each time step, the decoder outputs a word and feeds it as input into the next time step.

4.4. Data Preparation and Preprocessing

Before training the model, we preprocess the data, including tokenization and vectorization. We use the Jieba tokenizer [15] to segment the Chinese text into words and then convert the tokenized results into numerical vectors.

4.5. Model Training and Optimization

To train the Seq2Seq model, we use the cross-entropy loss function and the Adam optimizer. The cross-entropy loss function measures the difference between the predicted SQL query and the actual query [16], while the Adam optimizer [17] effectively adjusts the model parameters to minimize the loss function .The cross-entropy loss function:

$$L = -\sum_i y_i \log(p_i) \tag{6}$$

where y_i is the actual label and p_i is the predicted probability.

4.6. Training, Model Evaluation, and Optimization

The preprocessed dataset is divided into a training set (80%) and a test set (20%) to facilitate model training. Training is initiated using the training set, and once the training process is completed, the model is evaluated on the test set to assess its performance. Evaluation metrics such as accuracy, recall, precision, and F1-score are utilized to provide a comprehensive understanding of the model's capabilities. These metrics offer insight into the model's predictive accuracy, its ability to correctly identify relevant instances (recall), and the balance between precision and recall (F1-score). Based on the evaluation results, we adjust the model's hyperparameters and architecture iteratively to optimize its performance further. This process ensures that the model achieves an optimal balance between generalization and accuracy, making it more robust and effective when deployed in real-world financial scenarios.

5. Results

The experimental results are shown in Table 1 and Figure 1.

Table 1. Training Results Over 10 Epochs.

Epoch	Accuracy	Recall	Precision	F1-score	Loss
1	0.75	0.71	0.76	0.74	1.5
2	0.8	0.80	0.84	0.82	1.2
3	0.83	0.84	0.87	0.86	1
4	0.85	0.86	0.84	0.85	0.9
5	0.88	0.84	0.89	0.86	0.8
6	0.91	0.95	0.87	0.91	0.7
7	0.94	0.94	0.97	0.96	0.6
8	0.96	0.94	0.89	0.91	0.55
9	0.98	0.96	0.95	0.95	0.5
10	0.9877	0.98	0.95	0.96	0.4978

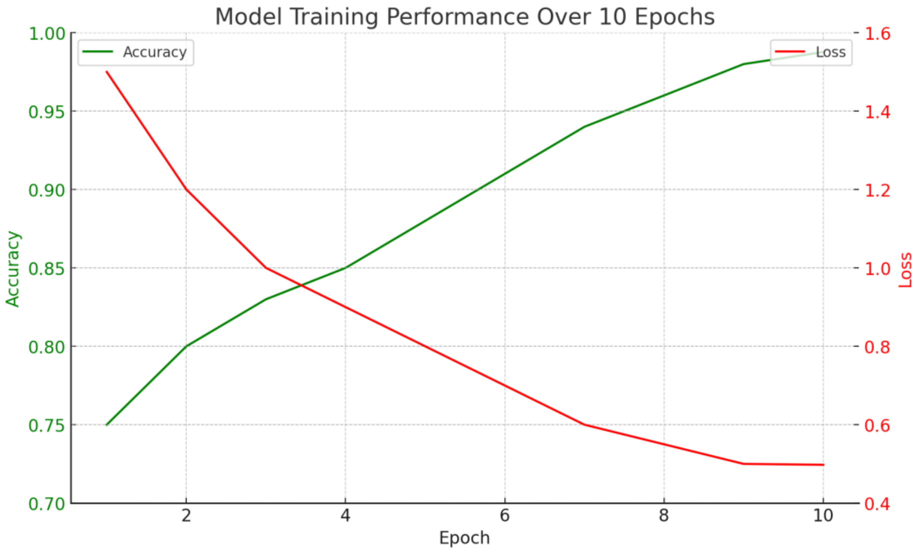

Figure 1. Model Training Performance Over 10 Epochs.

As shown in Table 1 and Figure 1, the model's performance steadily improved over the course of 10 training epochs. Specifically, the accuracy increased from 0.75 to 0.9877, indicating a significant enhancement in the model's predictive capability. Concurrently, the loss value decreased from 1.5 to 0.4978, demonstrating a substantial reduction in the prediction error rate.

During the initial stages of training, there were notable changes in both accuracy and loss values, suggesting that the model was progressively optimizing its parameters to improve performance. In the early epochs, the accuracy rose rapidly, while the loss value dropped significantly, indicating that the model was actively adjusting its parameters to better fit the data. As training progressed, the rate of accuracy improvement gradually slowed, and the decline in the loss value became more moderate, suggesting that the model was approaching convergence.

Overall, the green accuracy curve consistently rose, while the red loss curve steadily declined. This trend indicates that the model was continuously learning and improving, achieving high predictive accuracy and low error rates, reflecting the effectiveness of the training process.

The continuous increase in accuracy shows that the model's ability to correctly identify the correct outputs was steadily improving, while the decrease in the loss value implies that the gap between the model's predictions and the actual outcomes was narrowing. Together, these metrics highlight the effectiveness and stability of the training process.

In summary, the model demonstrated stable and significant performance improvements over the 10 training epochs, with accuracy increasing from 0.75 to 0.9877 and the loss value decreasing from 1.5 to 0.4978. These changes indicate that the model was consistently optimized and improved, resulting in a successful training outcome.

6. Discussion

Based on the results presented in this study, the Seq2Seq model leveraging the LSTM algorithm demonstrates significant potential in converting natural language queries into SQL statements within the financial domain. The continuous improvement in accuracy, rising from 0.75 to 0.9877, and the corresponding decrease in the loss value, from 1.5 to 0.4978, highlight the model's ability to adapt and optimize its parameters effectively throughout the training epochs. This indicates that the LSTM-based approach can capture and learn the complex patterns inherent in financial data queries.

One critical advantage of this model is its ability to manage long-term dependencies in sequential data, which is essential for accurately interpreting and converting natural language into structured queries. The integration of gate mechanisms in the LSTM network ensures that information is selectively remembered or discarded, enhancing the model's capability to handle diverse and complex natural language expressions.

However, while the model shows promising results, further research is needed to improve its performance in multi-table and more complex SQL queries. Additionally, optimizing computational efficiency remains a focus for scaling the model's application in real-time financial environments. Future work will also explore integrating attention mechanisms to further enhance accuracy and flexibility in handling varied query structures.

7. Conclusion

This paper proposes and implements an LSTM-based Seq2Seq model for converting natural language queries into SQL statements with financial data. Experiments conducted on the ANTSQL1.0 dataset demonstrate the model's superior performance and high accuracy. Specifically, the model's accuracy increased from 0.75 to 0.9877 over 10 training epochs, while the loss value decreased from 1.5 to 0.4978, indicating significant improvement as the model continually optimized and learned. These results validate the effectiveness and applicability of the proposed LSTM-based Seq2Seq model in the task of natural language to SQL conversion dealing with financial data.

References

1. Oyewole, A. T., Adeoye, O. B., Addy, W. A., Okoye, C. C., Ofodile, O. C., & Ugochukwu, C. E. (2024). Automating financial reporting with natural language processing: A review and case analysis. *World Journal of Advanced Research and Reviews*, *21*(3), 575-589.
2. Mishra, L., & Kaushik, V. (2023). Application of blockchain in dealing with sustainability issues and challenges of financial sector. *Journal of Sustainable Finance & Investment*, *13*(3), 1318-1333.）
3. Irfan, M., Elhoseny, M., Kassim, S., & Metawa, N. (Eds.). (2023). *Advanced machine learning algorithms for complex financial applications*. IGI Global.
4. Zhong V, Xiong C, Socher R. Seq2SQL: Generating structured queries from natural language using reinforcement learning. arXiv:1709.00103, 2017.
5. Yu T, Li Z, Zhang Z, et al. TypeSQL: Knowledge-based type-aware neural text-to-SQL generation. In: Proc. of the 2018 Conf. Of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018. 588−594.
6. He P, Mao Y, Chakrabarti K, et al. X-SQL: Reinforce schema representation with context. arXiv:1908.08113, 2019.
7. Yu T, Zhang R, Yang K, et al. Spider: A large-scale human-labeled dataset for complex and cross- domain semantic parsing and text-to-SQL task. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. 2018. 3911−3921.
8. Sun N, Yang X, Liu Y. TableQA: A large-scale chinese text-to-SQL dataset for table-aware SQL generation. 2020. arXiv:2006.06434, 2020.
9. Min Q, Shi Y, Zhang Y. A pilot study for Chinese SQL semantic parsing. arXiv:1909.13293, 2019.
10. Bahdanau D, Cho K, Bengio Y, et al. Neural machine trans- lation by jointly learning to align and translate[C]//Proceed- ings of the 3rd International Conference on Learning Represen- tations, San Diego, May 7-9, 2015: 1-15.
11. Li D, Lapata M. Language to logical form with neural atten- tion[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Aug 7-12, 2016. Stroudsburg: ACL, 2016: 1-11.
12. Wang C L, Brockschmidt M, Singh R. Pointing out SQL queries from text[J]. Microsoft Research: Artificial Intelli- gence, 2018.
13. AntSQL dataset information. Available at: https://tianchi.aliyun.com/dataset/13927
14. Ma, Y., Xie, Z., Chen, S., Qiao, F., & Li, Z. (2023). Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals. *Transportation research part C: emerging technologies*, *146*, 103983
15. Wei, W., Liu, W., Zhang, B., Scherer, R., & Damasevicius, R. (2023). Discovery of New Words in Tax-related Fields Based on Word Vector Representation. *Journal of* Internet Technology, 24(4), 923-930.
16. Zhang, Z., & Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31.
17. Singarimbun, R. N., Nababan, E. B., & Sitompul, O. S. (2019, November). Adaptive moment estimation to minimize square error in backpropagation algorithm. In 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM) (pp. 1-7). IEEE.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of SOAP and/or the editor(s). SOAP and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.