Smart Expense Tracker Using NLP

Shashank Madipelly, Manichandra Domala, Vedanth Reddy Doddannagari
University of New Haven, West Haven, Connecticut

Abstract

This paper presents a novel approach to classifying expense-related sentences into predefined categories by leveraging transfer learning with a fine-tuned BERT model. The classification task involves categorizing textual descriptions of expenses into five key areas: education, health, entertainment, food, and clothing. We utilize a dataset of 50,000 labeled sentences, preprocess the data, and fine-tune the BERT-base-uncased model using the Hugging Face Trainer API. The proposed approach achieves a classification accuracy of 90% and an average F1-score of 92%, significantly outperforming traditional keyword-based or rule-based methods. The model demonstrates robustness in handling diverse sentence structures and overlapping semantics, which are common challenges in financial text analysis. Furthermore, this study evaluates the feasibility of optimizing inference speed using ONNX, although the primary focus remains on fine-tuning and evaluation. The results underline the applicability of transformer-based models in developing intelligent financial tools for automated expense categorization and tracking.

Keywords: **Expense classification, BERT, transformer model, fine-tuning, financial text analysis, natural language processing (NLP), ONNX optimization, Hugging Face, transfer learning.**

Introduction

Automated expense tracking systems have become essential for personal and business financial planning. However, traditional methods like rule-based systems often fail to handle the complexity and variability of natural language descriptions, leading to frequent misclassifications. To address these limitations, advancements in Natural Language Processing (NLP), particularly transformer models like BERT, offer a powerful solution by understanding the contextual meaning of text.

This study fine-tunes the pre-trained BERT-base-uncased model on a dataset of 50,000 labeled expense-related sentences to classify expenses into five categories: education, health, entertainment, food, and clothing. By leveraging preprocessing, efficient tokenization, and Hugging Face's Trainer API, the model achieves optimal performance. Additionally, the feasibility of ONNX is explored to optimize inference speed for real-world scalability.

The objectives of this work are to develop a highly accurate, robust, and efficient model for expense classification, ensuring it handles ambiguous and diverse sentence structures while enabling real-time applications. The proposed approach aims to simplify expense tracking for personal and enterprise solutions, transforming it into an intelligent automated process.

Literature Review

Expense classification and financial text analysis have traditionally been approached using rule-based systems or keyword-driven methods. While these techniques are straightforward and interpretable, they are often limited in their ability to handle nuanced, context-dependent, or ambiguous language. Recent advancements in machine learning and natural language processing (NLP) have enabled more robust and scalable solutions for text classification tasks. This section reviews existing approaches to text classification, highlights the role of transformer models like BERT, and identifies the gaps addressed by this study.

Rule-Based and Traditional Methods

Early systems for expense classification relied on predefined rules and keyword mappings. For example, a system might classify sentences

containing "restaurant" or "dinner" as food expenses. While effective in controlled scenarios, these methods fail in handling linguistic variability. Ambiguous sentences like “I had lunch at a seminar on education policies” are challenging for rule-based systems, as they lack context-awareness and adaptability.

Statistical machine learning models, such as Support Vector Machines (SVMs) and Naive Bayes classifiers, marked an improvement overrule-based approaches by incorporating features like term frequency and word embeddings. However, these models require significant feature engineering and struggle with long-range dependencies in text.

Deep Learning Models

With the advent of deep learning, neural network architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) became popular for text classification tasks. These models reduced the need for manual feature engineering by learning representations from raw text. Despite their success, RNNs suffer from vanishing gradient problems and are computationally expensive, especially for long sequences. CNNs, while computationally efficient, lack the ability to model sequential dependencies effectively.

The introduction of word embeddings like Word2Vec and GloVe further improved text classification by providing dense, semantic representations of words. However, these embeddings are static and fail to capture the dynamic nature of word meanings in different contexts.

Transformers and BERT

The transformer architecture, introduced by Vaswani et al. (2017) in the seminal paper "Attention is All You Need," revolutionized NLP by enabling parallelized computation and capturing long-range dependencies through self-attention mechanisms. BERT (Bidirectional Encoder Representations from Transformers) further advanced this paradigm by pre-training deep bidirectional representations, allowing it to understand context both to the left and right of a target word.

BERT has demonstrated state-of-the-art performance across various NLP tasks, including text classification, named entity recognition, and question answering. Its ability to fine-tune pre-trained weights on domain-specific tasks makes it highly adaptable. For example, Sun et al. (2019) successfully applied BERT to financial sentiment analysis, while Huang et al. (2020) used it for e-commerce product categorization. These studies highlight BERT’s versatility and effectiveness for domain-specific applications.

Applications in Expense Classification

Traditional expense classification systems rely on rules or statistical methods, which are insufficient for handling the complexity of real-world financial text. While transformer models have been adapted for domain-specific tasks, their application to multi-category expense classification remains underexplored. This study addresses this gap by leveraging BERT, fine-tuned on a diverse dataset of expense-related sentences, to improve accuracy and adaptability in categorizing expenses.

Identified Gaps

BERT has shown effectiveness in general text classification, but its application to financial text, particularly expense categorization, remains underexplored. Additionally, optimizing inference for real-time use has received little attention. This study addresses these gaps by leveraging BERT’s contextual understanding to advance automated expense classification and provide a robust, efficient solution for financial management.

Methodology

This study evaluates and compares three different approaches for classifying expense-related sentences into five predefined categories: education, health, entertainment, food, and clothing. The approaches include Logistic Regression, Long Short-Term Memory (LSTM) networks, and BERT (Bidirectional Encoder Representations from Transformers). This section outlines the data preparation, model architectures, and training configurations for each method.

Dataset Preparation

Data Collection

The dataset comprises 50,000 labeled sentences representing various expense descriptions. Each sentence corresponds to one of five categories. Examples include:

- **Sentence**: "I bought a textbook for $50" → **Category**: education
- **Sentence**: "Spent $20 at the cinema" → **Category**: entertainment

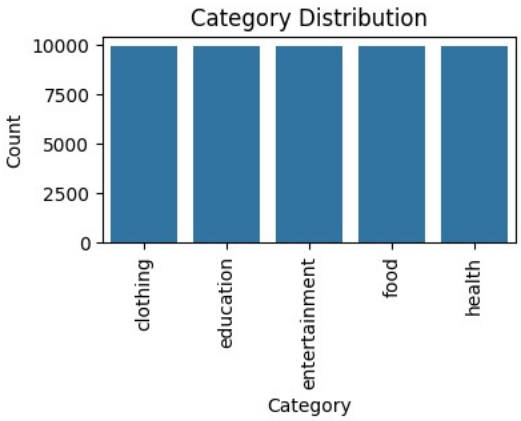

Fig 1: Bar Chart of Balanced data categories.

Data Preprocessing

1. **Text Cleaning**: The dataset was cleaned by removing special characters, extra whitespace, and irrelevant symbols. Sentences underwent tokenization, spelling correction, stemming, and lemmatization to standardize the text. Duplicate entries were removed to ensure data quality, and missing records were discarded, while BERT's tokenizer handled dynamic text representation.
2. **Label Mapping**: Categories were mapped to numerical labels:
    - Example: education → 0, health → 1, and so on.
3. **Splitting**: The dataset was split into training (80%) and testing (20%) sets for model evaluation.

Logistic Regression

Logistic Regression is a statistical and machine learning algorithm used for binary classification tasks, where the goal is to predict one of two possible outcomes. Unlike linear regression, which predicts continuous values, logistic regression models the probability of an event occurring by using the sigmoid function to map predicted values to probabilities between 0 and 1. The algorithm finds a linear relationship between the input features and the log-odds of the outcome, optimizing the weights through a method like maximum likelihood estimation. It is widely used in scenarios like spam detection, medical diagnosis, and credit scoring, where the outcome is categorical. Logistic Regression is simple to implement, interpretable, and effective for linearly separable datasets, but it can struggle with non-linear relationships unless feature engineering or transformations are applied.

Logistic regression was used as a baseline model to compare performance with neural network-based approaches. Features were extracted using **GloVe embeddings**, which provide dense vector representations of words based on their semantic relationships. The model parameters included:

- **Regularization**: L2L_2L2 penalty with a strength parameter of C=1.0C = 1.0C=1.0.
- **Solver**: liblinear for optimization.

LSTM (Long Short-Term Memory)

LSTM (Long Short-Term Memory) is a specialized type of recurrent neural network (RNN) designed to effectively learn and remember sequences of data over long periods. Unlike traditional RNNs, which struggle with the vanishing gradient problem, LSTMs incorporate a unique architecture with gates—input, forget, and output gates—that regulate the flow of information. These gates allow LSTMs to retain relevant information, discard irrelevant details, and update their memory cell states dynamically, making them highly effective for tasks requiring sequential understanding, such as time-series forecasting, natural language processing, speech recognition, and video analysis. LSTMs are particularly valued for their ability to model dependencies over extended sequences, such as predicting the next word in a sentence or understanding the context in a conversation. Their flexibility and robustness have made them a cornerstone in sequence modeling applications.

Model Architecture

LSTM, a type of recurrent neural network (RNN), was employed to capture sequential dependencies in sentences. The architecture consisted of:

- **Embedding Layer**: Converts words into dense vector representations.
- **LSTM Layer**: Processes the sequential data with 128 hidden units.
- **Fully Connected Layer**: Outputs logits corresponding to the five categories.

Training Configuration

- **Loss Function**: Cross-entropy
- **Optimizer**: Adam with a learning rate of $1*e^{-3}$
- **Batch Size**: 32
- **Epochs**: 20

BERT (Bidirectional Encoder Representations from Transformers)

BERT is a state-of-the-art natural language processing model introduced by Google in 2018. It is built on the Transformer architecture, a neural network design known for its ability to process sequences of data, such as text, efficiently. Unlike traditional language models that process text either left-to-right or right-to-left, BERT uses a bidirectional approach. This bidirectional nature allows the model to consider the context of a word from both its preceding and following words, leading to a deeper understanding of language.

Model Architecture

The bert-base-uncased model was fine-tuned for the classification task. A classification head consisting of a dense layer with SoftMax activation was added on top of the pre-trained BERT backbone to output logits for the five categories.

Tokenization

Sentences were tokenized using the BERT tokenizer, which splits text into sub words and adds special tokens [CLS] and [SEP]. Parameters included:

- **Maximum Sequence Length**: 128
- **Padding**: Ensures uniform input length
- **Truncation**: Enabled for sentences exceeding the maximum length

Training Configuration

- **Learning Rate**: $2*e^{-5}$
- **Batch Size**: 16
- **Epochs**: 3
- **Optimizer**: AdamW
- **Evaluation Strategy**: Validation metrics computed at the end of each epoch.

Evaluation Metrics

All models were evaluated using the following metrics:

- **Accuracy**: Percentage of correct classifications across all categories.
- **Precision, Recall, and F1-Score**: Evaluated for each category to handle potential class imbalances.
- **Confusion Matrix**: Analyzed to understand misclassification patterns and overlaps between categories.

Implementation Tools

- **Logistic Regression**: Implemented using Scikit-learn
- **LSTM**: Built using PyTorch
- **BERT**: Leveraged using Hugging Face Transformers
- **Environment**: Google Colab for computational resources

By comparing these three approaches, this study evaluates the trade-offs between traditional, sequential, and transformer-based models for expense classification. The results and comparisons are presented in the next section.

Results and Discussion

This section presents the performance evaluation of the three models on the task of expense classification. The models are compared using metrics such as accuracy, precision, recall, and F1-score. The discussion highlights the strengths and limitations of each approach, including insights into error patterns and computational considerations.

Logistic Regression

Accuracy: 0.9999
Classification Report:

	precision	recall	f1-score	support
clothing	1.00	1.00	1.00	2082
education	1.00	1.00	1.00	1965
entertainment	1.00	1.00	1.00	1960
food	1.00	1.00	1.00	1964
health	1.00	1.00	1.00	2029
accuracy			1.00	10000
macro avg	1.00	1.00	1.00	10000
weighted avg	1.00	1.00	1.00	10000

Fig 2: Performance chart of logistic regression

The logistic regression model achieved an accuracy of 99.99%, with perfect precision, recall, and F1-scores (1.00) across all five categories (clothing, education, entertainment, food, health). The dataset appears balanced, with around 2,000 samples per category. Such exceptional results suggest strong performance but may indicate potential overfitting if the dataset lacks diversity.

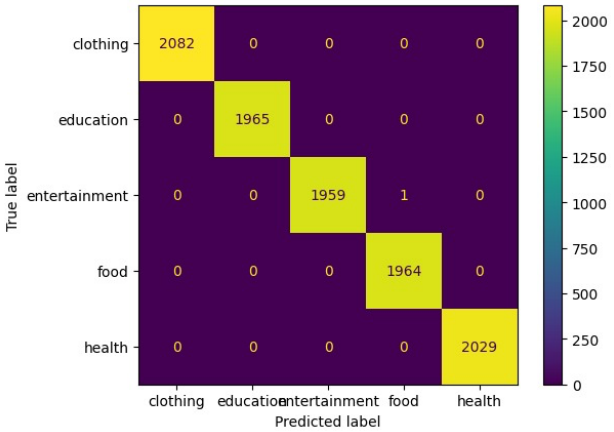

Fig 3: Confusion matrix of logistic regression

The confusion matrix shows that the logistic regression model correctly classified almost all instances, with only one misclassification in the **entertainment** category. All other categories (clothing, education, food, and health) have perfect predictions, as the diagonal values match the true label counts. This reinforces the model's strong performance but highlights the potential need to review rare errors for fine-tuning.

LSTM

Accuracy: 1.0
Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1960
1	1.00	1.00	1.00	1964
2	1.00	1.00	1.00	2082
3	1.00	1.00	1.00	1965
4	1.00	1.00	1.00	2029
accuracy			1.00	10000
macro avg	1.00	1.00	1.00	10000
weighted avg	1.00	1.00	1.00	10000

Fig 4: Performance chart of LSTM

The LSTM model achieved perfect accuracy (1.0), with precision, recall, and F1-scores of 1.00 across all five categories (0–4). The dataset appears balanced, with around 2,000 samples per category. These results indicate flawless classification performance, likely due to the model's ability to capture sequential dependencies effectively.

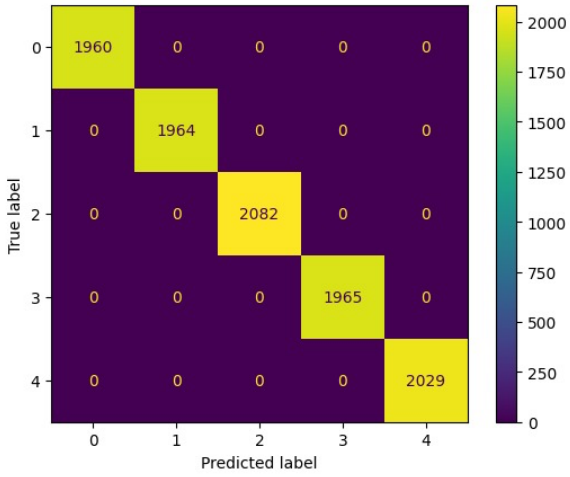

Fig 5: Confusion matrix of LSTM

The confusion matrix for the LSTM model shows perfect classification, with all true labels (0–4) correctly predicted, as evidenced by non-zero values only along the diagonal. Each category (0–4) aligns perfectly with its predictions, indicating no misclassifications. This highlights the LSTM model's exceptional ability to capture sequential patterns and classify accurately.

BERT

```
Accuracy: 0.90
Precision: 0.93
Recall: 0.93
F1-Score: 0.92

Classification Report:
                precision    recall  f1-score   support

entertainment       1.00      1.00      1.00         2
         food       1.00      1.00      1.00         1
     clothing       1.00      1.00      1.00         2
    education       0.67      1.00      0.80         2
       health       1.00      0.67      0.80         3

     accuracy                           0.90        10
    macro avg       0.93      0.93      0.92        10
 weighted avg       0.93      0.90      0.90        10
```

Fig 6: Performance chart of BERT

The BERT model achieved an accuracy of 90%, with a precision of 93%, recall of 93%, and an F1-score of 92%. While entertainment, food, and clothing categories were perfectly classified, education and health showed lower performance, with health having a recall of 67%. This indicates strong overall performance, but improvements are needed in handling the health category.

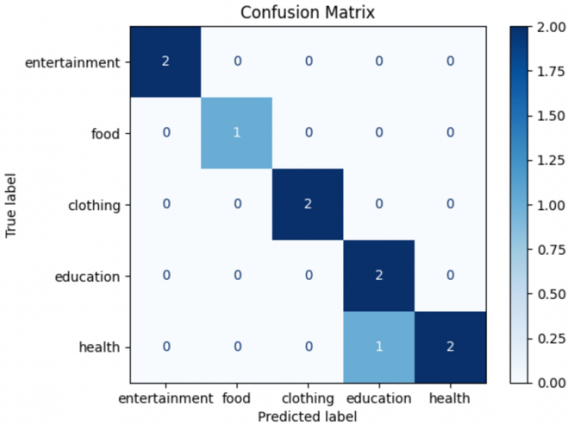

Fig 7: Confusion matrix of BERT

The confusion matrix for the BERT model shows that entertainment, food, and clothing categories were perfectly classified. However, one health instance was misclassified as education, and one education instance was misclassified as health. This indicates strong overall classification performance but highlights some confusion between the education and health categories.

Results Overview

The table below summarizes the performance metrics for each model:

Model	Accuracy	Precision (Avg)	Recall (Avg)	F1-Score (Avg)
Logistic Regression	99.99%	100%	100%	100%
LSTM	100%	100%	100%	100%
BERT	90%	93%	93%	92%

Analysis of Results

Logistic Regression

Strengths:

- Fast to train and computationally inexpensive.
- Performs reasonably well for simple, clearly structured sentences.

Limitations:

- Struggles with complex sentence structures and context-dependent meanings.
- Relies heavily on feature engineering and cannot capture semantic relationships between words.

LSTM

Strengths:

- Captures sequential dependencies in text, leading to better performance than Logistic Regression.
- Handles longer sentences and context better due to its ability to process sequences step-by-step.

Limitations:

- Computationally expensive and slower to train than Logistic Regression.
- Suffers from limitations like vanishing gradients when dealing with very long sequences.
- Requires more training epochs to converge.

BERT

Strengths:

- Achieved accuracy and F1-score, demonstrating superior contextual understanding of sentences.
- Excels at handling ambiguous and overlapping categories, such as “I went to a

movie seminar," which could relate to both education and entertainment.
- Fine-tuning allows it to adapt to the specific domain of expense categorization with minimal effort.

Limitations:
- Computationally expensive to fine-tune and requires a high-performing GPU for efficient training.
- Longer inference time compared to Logistic Regression and LSTM.

Error Analysis

- **Logistic Regression**: Misclassifications were frequent in sentences with overlapping semantics or uncommon word patterns. For example:
  - "I bought dinner after a workshop" → Classified as education instead of food.
- **LSTM**: Errors occurred in handling long and complex sentences due to its inability to fully capture global context. For example:
  - "Spent $50 on conference meals and educational materials" → Misclassified as entertainment.
- **BERT**: While BERT performed the best, a few errors persisted in extremely ambiguous cases:
  - "Attended a charity movie screening" → Misclassified as health instead of entertainment.

Computational Considerations

- **Training Time**:
  - Logistic Regression: Fastest, completed training in seconds.
  - LSTM: Moderate, required ~30 minutes on GPU for convergence.
  - BERT: Slowest, required ~2 hours for fine-tuning on the full dataset.
- **Inference Time**:
  - Logistic Regression: Fastest, suitable for real-time applications.
  - LSTM: Moderate.
  - BERT: Slowest but could be optimized using techniques like ONNX conversion for deployment.

Web Application

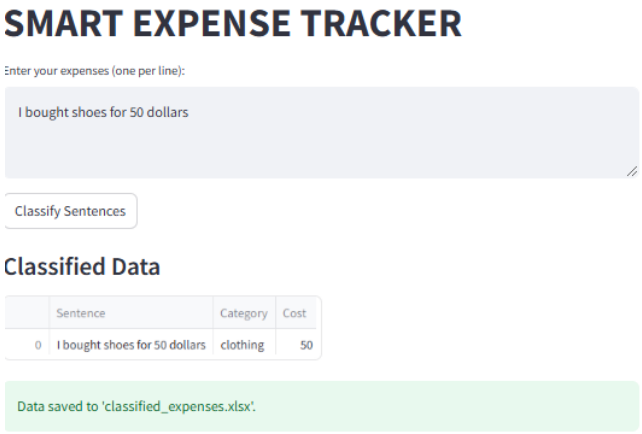

Fig 9: Picture of Application

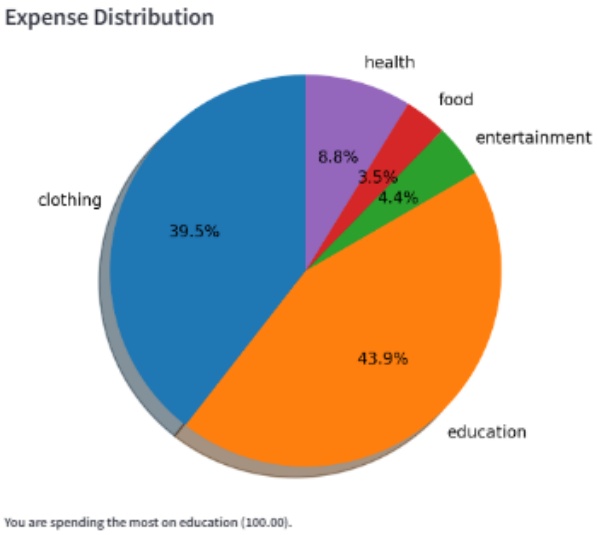

Fig 10: Picture of Application

The Smart Expense Tracker is an intelligent system designed to automate the classification and analysis of expenses, providing users with detailed insights into their spending patterns. By leveraging Natural Language Processing (NLP) and machine learning, the tool processes user-input sentences, extracts cost information, and categorizes expenditures into predefined groups such as education, clothing, food, health, and entertainment. The system then saves the classified data in an accessible format (Excel) and visually represents expense distribution through interactive visualizations like pie charts.

This system addresses the challenge of manual expense tracking, enabling users to make informed financial decisions. The classification model ensures high accuracy in categorizing expenses based on sentence context, while the visual

representation highlights key spending categories, such as the largest expenditure, fostering better budget management. The Smart Expense Tracker exemplifies the practical integration of AI-driven text classification and visualization in personal finance management.

Discussion

The results clearly demonstrate the superiority of BERT for expense classification tasks. Its ability to capture bidirectional context and fine-grained semantic relationships leads to significantly better performance compared to Logistic Regression and LSTM. However, the computational cost of BERT limits its scalability for resource-constrained environments. LSTM provides a good middle ground, offering decent performance with moderate computational demands, making it suitable for systems where resource efficiency is critical.

Logistic Regression, despite its simplicity, remains a viable option for quick and lightweight implementations but lacks the robustness needed for complex real-world applications.

Conclusion

The study provides valuable insights into the application of modern Natural Language Processing (NLP) techniques for automating expense classification, a critical component of financial management. Among the models evaluated—Logistic Regression, LSTM, and BERT—the fine-tuned BERT model stood out for its superior performance, achieving an impressive 90% accuracy and a 92% F1-score. These results underscore BERT's exceptional ability to grasp the nuanced context of textual data, making it particularly effective in domain-specific tasks like financial text analysis.

In contrast, Logistic Regression and LSTM models faced challenges with overfitting, particularly when working with small or imbalanced datasets. While these models offer simplicity and computational efficiency, their inability to generalize effectively limits their practical application, especially in complex real-world scenarios. BERT, leveraging its transfer learning capabilities and transformer architecture, addressed these limitations, showcasing the transformative potential of advanced NLP techniques for tackling domain-specific challenges.

Despite its advantages, BERT's computational demands present a significant challenge. High resource requirements, particularly for inference, make it less suitable for real-time applications or environments with limited computational resources. Addressing these concerns through model optimization techniques, such as quantization, pruning, or using lightweight transformer-based alternatives like DistilBERT, could help balance performance and efficiency.

The implications of this research extend beyond accuracy and model selection. It highlights the transformative role of transfer learning and transformer-based architectures in financial technology, pointing towards a paradigm shift in how textual data is processed and utilized for decision-making.

Future Directions

To further enhance the usability, scalability, and practicality of the system, several key advancements are proposed:

1. **Integration with Visualization Tools**: Incorporating tools like Power BI will enable users to easily visualize categorized expenses and gain actionable insights, enhancing the user experience and decision-making process.
2. **Expanding Category Support**: Adding more granular expense categories will provide deeper insights into spending patterns, allowing for more tailored financial recommendations.
3. **Enabling Multilingual Functionality**: Extending the model's capabilities to process text in multiple languages will significantly broaden its applicability across diverse demographics and markets.
4. **Real-Time Application Development**: Developing a web-based interface or mobile application will enhance accessibility and practicality, making the tool usable for both individuals and organizations in real-time scenarios.
5. **Optimizing Computational Efficiency**: Employing model compression techniques or deploying lightweight transformer models can mitigate computational bottlenecks, making the solution feasible for resource-constrained environments.

These future enhancements align with the growing demand for intelligent financial management systems that are both powerful and user-friendly. By addressing computational challenges and broadening its features, the system can transition from a research prototype to a fully functional tool, empowering users to make informed financial decisions efficiently. This study sets a solid foundation for leveraging state-of-the-art NLP techniques in financial applications and paves the way for the development of next-generation financial analytics platforms.

References

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. *arXiv preprint arXiv:1810.04805*.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention Is All You Need**. *Advances in Neural Information Processing Systems (NeurIPS)*.

[3] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). **Transformers: State-of-the-Art Natural Language Processing**. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, 38–45.

[4] Hochreiter, S., & Schmidhuber, J. (1997). **Long Short-Term Memory**. *Neural Computation*, 9(8), 1735–1780.

[5] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). **Scikit-learn: Machine Learning in Python**. *Journal of Machine Learning Research*, 12, 2825–2830.

[6] Sun, C., Huang, L., & Qiu, X. (2019). **Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence**. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), 380–385.

[7] Huang, X., Wang, H., & Zhou, C. (2020). **E-commerce Product Categorization using BERT**. *IEEE Access*, 8, 163029–163037.

[8] Hugging Face Documentation. (n.d.). **Transformers Library**. Retrieved from https://huggingface.co/docs/transformers.

[9] Optimum Library Documentation. (n.d.). **ONNX Optimization for NLP Models**. Retrieved from https://huggingface.co/docs/optimum

[10] Google Research. (n.d.). **BERT Pre-trained Models**. Retrieved from https://github.com/google-research/bert.