Quack at the NTCIR-17 FinArg-1 Task: Boosting and MLM Enhanced Financial Knowledge Sequence Classification

Zih-An Lin
Department of CSE,
National Chung Hsing University
Taiwan
hua10155174@gmail.com

Hsiao-Min Li
Department of CSE,
National Chung Hsing University
Taiwan
siaomin@smail.nchu.edu.tw

Adam Lin
Department of CSE,
National Chung Hsing University
Taiwan
s109056024@mail.nchu.edu.tw

Yun-Ching Kao
Department of CSE,
National Chung Hsing University
Taiwan
s110056002@smail.nchu.edu.tw

Chia-Shen Hsu
Department of CSE,
National Chung Hsing University
Taiwan
s110056015@mail.nchu.edu.tw

Yao-Chung Fan*
Department of CSE,
National Chung Hsing University
Taiwan
yfan@nchu.edu.tw

ABSTRACT

In the exploration of the task "Identifying Attack and Support Argumentative Relations in Social Media Discussion Threads" (SMDT) [1], we aim to discern differences between proponents and adversaries in financial discussions on the internet. For classification tasks such as these, fine-tuning Transformer models like BERT [2] is an intuitive approach. In this study, we build upon this foundation by incorporating the Masked Language Model technique to enrich the model's domain knowledge within the financial field. Furthermore, we optimize the model's performance by adjusting the weights in the loss function. Experimental results confirm that both methods effectively enhance the model's performance. This research introduces three simple yet effective methods to improve the Transformer model's ability for SMDT. The code and model for this study are available at https://github.com/leonardo-lin/NTCIR.

KEYWORDS

financial, Adaboost, sequence classification, Mask Language model

TEAM NAME

Quack

SUBTASKS

[Identifying Attack and Support Argumentative Relations in Social Media Discussion Threads (SMDT) (Chinese)]

1 INTRODUCTION

The financial market is a highly dynamic and ever-changing environment influenced not only by global political and economic factors but also by the subtle effects of participants' psychology. To capture the true pulse of the financial market, experts and scholars continually explore and analyze various types of data, from macroeconomic data to individual transaction data, as well as news reports, trading volumes, and the buying and selling trends of institutional investors. However, in the digital era, we observe that public opinions on the internet are increasingly becoming important indicators of market sentiment.

These online opinions often differ from traditional news sources as they are more immediate and diverse. Ranging from comments by professional analysts to casual remarks by the general public, these opinions constitute a vast yet heterogeneous data store. To extract valuable information from these opinions, the NTCIR-17 SMDT task attempts to create a mechanism for classifying these opinions based on their relative relationships: either supporting or opposing each other.

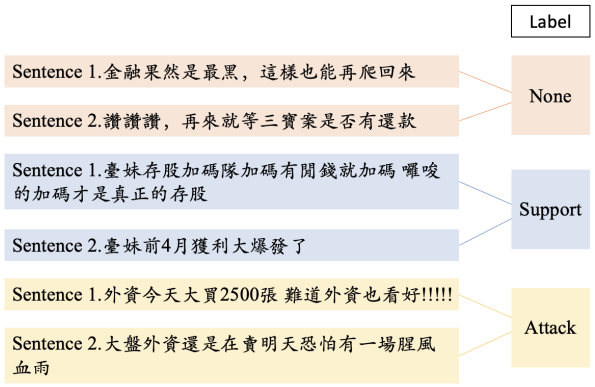

Figure 1: Instances of Support and Opposition in the Dataset

Given two comments found online, the challenge lies in determining their relationship and how they collectively influence the overall market sentiment. Figure 1 describes three examples from the dataset, illustrating the supportive and opposing relationships between different comments.

In the following sections, we detail our methods in Section 2, adopting a progressive enhancement strategy to improve the model performance in financial analysis. Section 3 showcases our research outcomes. Subsections 3.1 and 3.2 elucidate our experimental settings, while Subsections 3.3 and 3.4 explain how we enhance the model's performance through the Masked Language Model and adjusted loss function.

2 METHODOLOGY

In this study, we present three methods for judging the relationships between sentences:

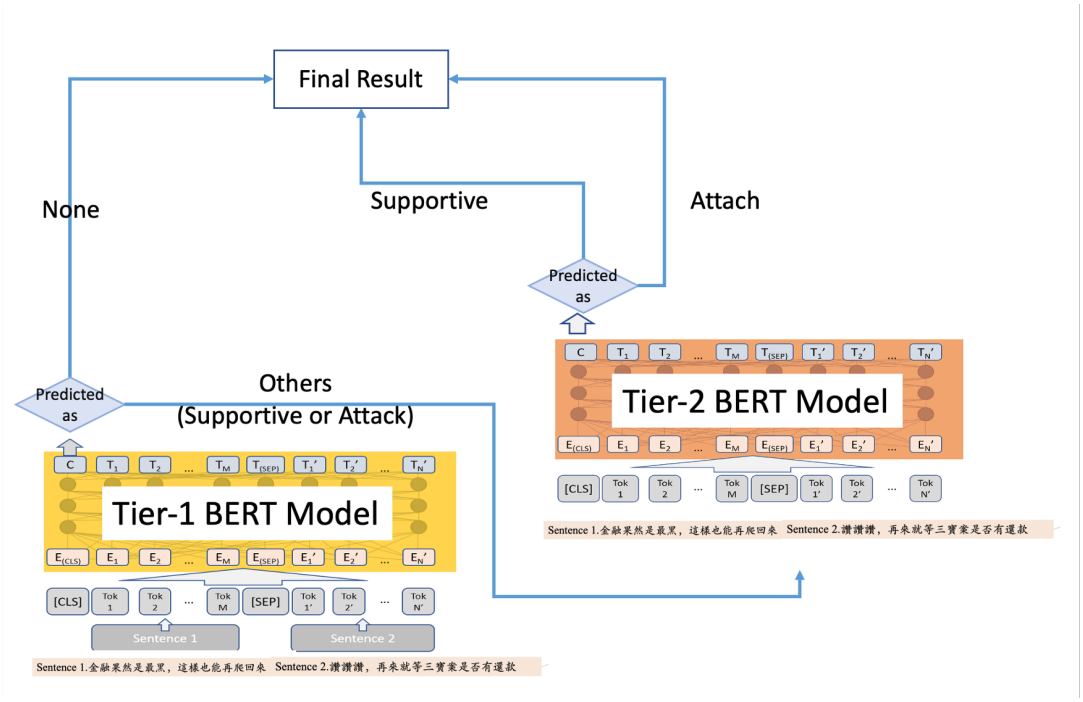

Figure 2: Process of two layered Bert Model

- Two-tiered Bert Classification (Submission-1: (SMDT_1))
- Mask LM (Submission-2: (SMDT_2))
- Boosting (Submission-3: (SMDT_3))

These three methods contribute to exploring sentence relationship judgment and provide a solid foundation for subsequent comparisons and analyses. We will delve into the ideas and training procedures of each method separately. Through these method comparisons, we aim to gain a better understanding of their strengths and limitations, offering valuable references for the final research results.

2.1 Two-tiered Bert Classification (SMDT_1)

Our approach involves hierarchical prediction using two binary classification models. Refer to Figure 2 for a visual representation. Initially, a BERT model processes the input sentence pair to determine their entailment relationship—whether they are related or unrelated. If the sentences are considered unrelated, we label the output as "none." If they are deemed related, the second-tier model comes into play. This model classifies the sentences as either supporting or attacking.

2.2 Mask LM (SMDT_2)

In SMDT_2, we explore the idea of using Masked Language Modeling (MLM) mechanism to enhance the domain knowledge of the BERT model in the financial domain. The MLM is a common language model pre-training method that involves randomly masking certain words in the text, requiring the model to predict the masked words, thereby encouraging the model to learn contextual relevance and word associations.

In our study, we use data from the well-known "STOCK" board in Taiwan's "ptt" forum as the basis, which exhibits similarities with the dataset provided by NTCIR-17. We acquire approximately 32,000 forum articles and conduct MLM fine-tuning training. This process helps the BERT model become more familiar with domain-specific vocabulary in the financial realm (such as stock codes, company aliases, forum terminology). This training method contributes to enhancing the performance of the BERT model in financial sentence comprehension. By reinforcing its understanding of financial-related content, we anticipate that BERT will demonstrate even more remarkable capabilities in the financial domain.

2.3 Boosting (SMDT_3)

We propose an Adaboost-inspired approach to enhance our classification model and improve overall performance scores. Adaboost is an ensemble learning technique in which the weights of samples misclassified by the previous classifier are increased. In contrast, the weights of samples correctly classified are decreased, and these adjusted weights are then used to train the classifier in the next round. A departure of our method from the original Adaboost lies in the interpretation of weights, where we replace the focus on the proportion of different samples with adjusting proportions of loss. We employ this method to differentiate between "attack" and "support". The working steps are as follows.

In the training data, each data pair is considered as (x_i, y_i), where x and y represent two sentences, and each (x_i, y_i) pair is associated with a corresponding label l_i is either support (0) or attack (1). Our boosting training procedure consists of a total of M iterations. In each iteration, we train a classifier G_m ($0 < m < M$). During classifier

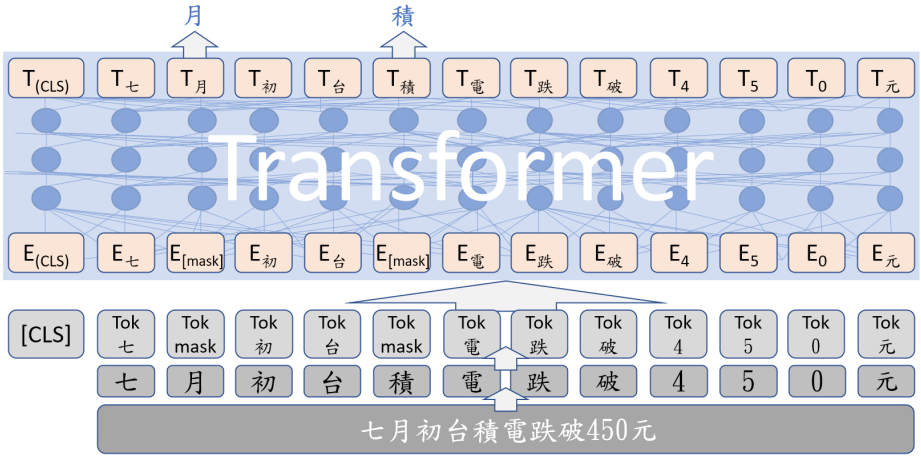

Figure 3: Process of MLM prediction

training, each data pair is assigned a weight to calculate the loss function. The weights for each data pair are adjusted based on the predictions of the previous model G_{m-1}. We use a list of weights $D_m(i)$ to represent the weights of each data pair in Iteration m. Initially, all weights are set to 1 for $D_1(i)$. In each iteration, we train the corresponding model and adjust the sampling weights as follows.

- First, we incorporate D_m into the loss function, yielding the weighted loss function F_m, which is then employed for model training. The loss function employed in this context is the binary cross-entropy. Through this procedure, we obtain a classifier $G_m : X \rightarrow \{0, 1\}$ along with its corresponding error e_m.

$$F_m = -\sum_{i=1}^{N}(D_m(i) \cdot (l_i \log(\hat{l}_i) + (1 - l_i) \log(1 - \hat{l}_i))) \quad (1)$$

$$e_m = \frac{\sum_{i=1}^{N} D_m(i) I(G_m(x_i, y_i) \neq l_i)}{\sum_{i=1}^{N} D_m(i)} \quad (2)$$

- Subsequently, we calculate the coefficient α_m for the classifier $G_m(x, y)$, which represents the weight assigned to predicting the output.

$$\alpha_m = \frac{1}{2} ln \frac{1 - e_m}{e_m} \quad (3)$$

- After calculating α_m, we use it to update the weights of each data point for the next round of training, denoted as $D_{m+1}(i)$ where $i = 1, 2...N$.

$$D_{m+1}(i) = \begin{cases} D_m(i) \exp(-\alpha_m), & \text{if } l_i = G_m(x_i, y_i) \\ D_m(i) \exp(\alpha_m), & \text{otherwise} \end{cases} \quad (4)$$

In the end, we enable all classifiers to make joint predictions based on their respective weights α_m:

$$G(x, y) = \begin{cases} 1, & \text{if } \frac{\sum_{m=1}^{M} \alpha_m G_m(x, y)}{\sum_{m=1}^{M} \alpha_m} \geq 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

By employing this approach, we can iteratively boost the weak classifiers and allow these weak classifiers to come together as a stronger classifier. In our experimentation, a total of five classifiers were trained and combined for predictions.

3 EXPERIMENTS

3.1 Setup

We trained all the models using the training dataset ($Train$) officially released by NTCIR-17 on March 31. During this period, we randomly extracted 300 records from the training dataset to create a test dataset ($Test_{informal}$). Ultimately, for the final evaluation, we employed the official dataset released by NTCIR-17 on June 30 as our ultimate test dataset ($Test_{official}$). The data quantities for these three datasets are presented in Table1.

	None	Support	Attack	total
$Train$	667	3505	2046	6218
$Test_{informal}$	17	171	112	300
$Test_{official}$	85	460	270	815

Table 1: The distribution of labels across our dataset.

3.2 Evaluation Matrics

In this study, we use precision, recall, F1-score, and accuracy as our evaluation metrics.

In the experimental phase, we perform training and fine-tuning on the SMDT_1, SMDT_2, and SMDT_3 models using the $Train$. Subsequently, we assess the performance using the $Test_{informal}$, evaluating the models' performance across various levels based on pre-selected evaluation metrics. The corresponding results are presented in Table 2.

In the final stage, we evaluate the SMDT_1, SMDT_2, and SMDT_3 models using the $Test_{official}$. The results of these evaluations are also displayed in Table 3.

	pre	rec	f1	acc
Baseline	0.68	0.68	0.68	0.64
SMDT_1	0.70	0.73	0.71	0.68
SMDT_2	0.70	0.69	0.70	0.67
SMDT_3	0.78	0.77	0.77	0.75

Table 2: Result of precision, recall, f1, accuracy on $Test_{informal}$ in different way.

	pre	rec	f1	acc
Baseline	0.73	0.71	0.72	0.68
SMDT_1	0.71	0.70	0.70	0.68
SMDT_2	0.76	0.72	0.74	0.72
SMDT_3	0.69	0.66	0.67	0.66

Table 3: Result of precision, recall, f1, accuracy on $Test_{official}$ in different way.

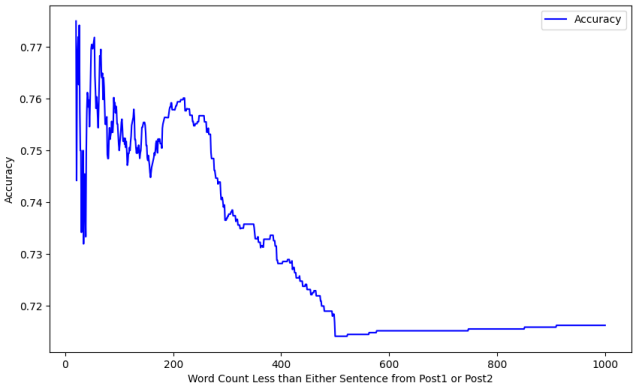

Figure 4: Correlation Chart between SMDT_2 and Sentence Length

3.3 Two-tiered Bert Classification

Using the same training dataset as the other models, we fine-tuned BERT into a three-class text classification model. We used this as a baseline for comparison experiments. Based on the data in Table 2, we can observe that the SMDT_1 method outperforms the baseline in all evaluation metrics. In particular, the SMDT_1 method shows significant superiority in terms of recall and F1-scores. This result emphasizes the effectiveness of our proposed framework in the classification task.

3.4 Mask LM

In Table 2, concerning the comparison between SMDT_1 and SMDT_2, we observed that the application of the Masked Language Model (MLM) did not seem to yield pronounced improvement effects. In fact, there was a slight decrease in both recall and F1-scores. However, a notable enhancement was evident on the testing data provided by NTCIR-17, as reflected in the score statistics shown in Table 3. Specifically, our precision, recall, F1 scores, and accuracy all showed relative improvements compared to the initial SMDT_1 model, with increases of 0.05, 0.02, 0.04, and 0.04, respectively. This illustrates that the efficacy of MLM might vary across different testing data scenarios, and it performs admirably on the $Test_{official}$.

3.5 Boosting

According to the results in Table 2, we observe that the SMDT_3 approach exhibits significant performance enhancement. Across all four evaluation metrics, nearly all scores demonstrate improvements of over 0.05 compared to the other two methods. We are highly content with such outcomes, as they underscore the exceptional performance achieved by the SMDT_3 approach in the experiment.

However, in the results presented in Table 3, we notice that the SMDT_3 approach does not bring about score improvements; in fact, there are instances of score decreases. Across the various evaluation metrics, scores have declined by varying degrees of 0.2 to 0.3.

Taking into consideration the experimental outcomes from these two tables, we draw the following conclusion: the SMDT_3 approach exhibits significant variability across different contexts. While achieving remarkable improvements in certain scenarios, it falls short in others. This highlights the substantial variation in method performance across different datasets and contexts. Overall, this underscores the complexity of model evaluation, necessitating the consideration of multiple factors to arrive at a comprehensive judgment.

3.6 Discussion

During our error analysis of the highest-scoring approach, SMDT_2, we discovered an increasing error rate in its judgments as the input length grew. As depicted in Figure 4, we validated this phenomenon by experimenting with data of varying lengths. Interestingly, while retaining longer-length data, a significant decrease in accuracy was observed.

This occurrence may be attributed to several factors. On one hand, longer inputs can make it challenging for the model to capture crucial contextual information, resulting in inaccuracies in judgments. Additionally, excessively long inputs might exacerbate model complexity, potentially leading to overfitting or information loss issues.

Based on these observations, we conjecture that SMDT_2 faces difficulties when processing lengthy inputs. This could be due to the model's limited capacity to handle longer input contexts, leading to a decline in judgment performance. Despite demonstrating excellence in certain short input scenarios, challenges persist when dealing with longer inputs. In the future, we aim to enhance the model's comprehension of longer texts by exploring alternative methods that enable a better understanding of the semantic context preceding and following lengthy texts, thereby improving overall comprehension capabilities.

4 CONCLUSIONS

This paper provides a detailed account of the QUACK team's participation in the Identifying Attack and Support Argumentative Relations in Social Media Discussion Threads (SMDT) subtask of NTCIR-17. To address this task, we introduce an enhanced approach with a two-level binary classification framework to augment the performance of the foundational three-class BERT model. Furthermore, we employ the Masked Language Model technique to enhance the BERT model's comprehension abilities in the financial domain. Finally, we perform fine-tuning through the loss function re-weighting for the model.

ACKNOWLEGEMENT

This work is supported by NSTC Taiwan Project under grant No. 112-2221-E-005-075-MY3, ITRI, and Delta Research Center, Delta Electronics, Inc.

REFERENCES

[1] Chung-Chi Chen, Chin-Yi Lin, Chr-Jr Chiu, Hen-Hsen Huang, Alaa Alhamzeh, Yu-Lieh Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2023. Overview of the NTCIR-17 FinArg-1 Task: Fine-Grained Argument Understanding in Financial Analysis. In *Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, Tokyo, Japan.*

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).