NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001286

TMUNLP at the NTCIR-17 FinArg-1 Task

Heng-Yu Lin
Graduate Institute of Data Science,
Taipei Medical University
Taipei, Taiwan
m946111008@tmu.edu.tw

Eugene Sy
Graduate Institute of Data Science,
Taipei Medical University
Taipei, Taiwan
m946111012@tmu.edu.tw

Tzu-Cheng Peng
Graduate Institute of Data Science,
Taipei Medical University
Taipei, Taiwan
m946111003@tmu.edu.tw

Shih-Hsuan Huang
Graduate Institute of Data Science,
Taipei Medical University
Taipei, Taiwan
m946110008@tmu.edu.tw

Yung-Chun Chang
Graduate Institute of Data Science,
Taipei Medical University
Taipei, Taiwan
changyc@tmu.edu.tw

ABSTRACT

The TMUNLP team participated in the FinArg-1 Task of NTCIR-17, focusing on Argument Unit Identification and Argument Relation Identification in the finance domain using social media and earnings call datasets. Notably, the team ranked 1st and 3rd in these subtasks, respectively. This paper presents the team's methodologies, results, and conclusions. For Earning Conference Call (ECC) Argument Unit Identification, an ensemble strategy combining diverse pre-trained models achieved a Macro F1 score of 0.766231, with significant contributions from models like ELECTRA, RoBERTa, BERT-base-uncased, and FinBERT. In ECC Argument Relation Identification, a combination of pretrained models and sampling strategies, along with voting mechanisms, improved natural language inference tasks. Future research opportunities include optimizing integration methods for semantic inference efficiency. Finally, in Social Media (SM) Argument Relation Identification, ChatGPT's keyword features positively impacted model performance. Challenges of translation and data imbalance were addressed through category-weighted sampling methods and soft voting, showcasing adaptable strategies. This study highlights the efficacy of ensemble strategies and diverse models in NLP tasks and emphasizes potential advancements in the field.

KEYWORDS

Ensemble Technique, Unbalanced Dataset, Natural Language Inference, Text Keyword Extraction, Argument Classification, Balancing Technique

TEAM NAME

TMUNLP

SUBTASKS

Argument Unit Identification (ECC)

Argument Relation Identification (ECC and Social Media)

1 INTRODUCTION

The TMUNLP participated in three subtasks of Argument Unit Identification and Argument Relation Identification under the FinArg-1 Task of NTCIR-17 that focuses on the domain of finance with datasets that are from social media and earning calls [1]. The team ranked 1st and 3rd in Argument Unit Identification (ECC) and Argument Relation Identification (ECC), respectively. All models' source code developed in this paper are all accessible through a GitHub Repository and can be found in each subtask's method section. This paper discusses the methods, results, and discussion in three different sections, one for each subtask, and a separate section for conclusion.

2 ECC ARGUMENT UNIT SUBTASK

In this section, we outline the approach employed by the TMUNLP team for the Earning Conference Call argument unit subtask (ECCAU). The primary objective of this task is to categorize argumentative sentences as either "Premise" or "Claim." Thus, the subtask is considered as a Binary Classification problem. Section 2.1 provides a concise overview of the dataset and the preprocessing steps we undertook. In Section 2.2, we detail the methods variation for each of our submissions. Lastly, in Section 2.3, we delve into the experimental outcomes, conducting a comprehensive analysis and discussion of the results.

2.1 Dataset and Preprocessing

Dataset Overview. The dataset [2] for this subtask demonstrates balanced proportions, comprising 5,078 entries for "Premise" and 4,613 entries for "Claim." The data is divided into train, dev, and test sets, with a ratio of 80% for training, 10% for development, and 10% for testing. In the training phase, we adopt an approach to dataset augmentation. Instead of evaluating our models on the train and development sets separately, we treat them as a unified set. Employing a stratified 10-Fold approach enhances the robustness of our evaluation methodology.

Preprocessing. In ECC Argument Unit Identification, minimal preprocessing was performed on the text to preserve both its grammatical structure and linguistic features. Therefore, only conventional preprocessing techniques were applied to prepare the text data for a Transformer-based Language Model. These techniques include tokenizing the text, adding [CLS] and [SEP] tokens, padding or truncating the text to a pre-determined length (512), and obtaining the input IDs and attention masks for model training.

2.2 Methods

Given the specialized nature of the domain (earnings call transcripts), the language used is tailored for businesses, with references to companies and financial terms. Initially, we assumed that employing a language model pre-trained on financial domain texts, like FinBERT, would yield the best performance[3]. While the results did surpass those of traditional machine learning methods, within the realm of deep learning, they did not show improvement compared to certain other benchmark language models we finetuned. We hypothesize that using an ensemble technique could leverage the strengths of various high-performing deep learning approaches.

Experiment Setup. In general, all submissions use a straightforward voting mechanism that integrates different versions of pre-trained language models. Each individual language model undergoes initial fine-tuning using the provided dataset, determining the optimal epoch through testing. Then, the models are ranked based on their performance, as measured by the Macro-F1 score. The final prediction is made through an ensemble voting technique, achieved by selecting a combination of the Macro-F1 ranked Top-k models, with increasing values of k.

Two distinct types of voting are employed across different combinations:

(a) **Soft Voting:** In this approach, the final prediction is determined by a threshold, >0.5, and the mean of the prediction scores from the fine-tuned models.
(b) **Hard Voting (Majority Voting):** Operating under a similar threshold framework, this approach utilizes the mean of binary predictions from the fine-tuned models to arrive at the final prediction.

After collecting results from various top-k and voting combinations, we identified the top three best-performing models. These models were subsequently retrained using the entire dataset with the same configuration. Their predictions on the test set were then utilized as our final submission.

To ensure the reproducibility of our method in ECCAU, you can access the source code through our GitHub repository at nlptmu/FinArg-1_AUC_FinSeq.

Table 1: Pretrained Language Models (PLM) Utilized for Subtasks and Parameter Configuration.

PLMs	Parameter Configuration				
	Epochs	LF	Optimizer	Dropout Rate	LR
ECCAU					
ALBERT*	3*				
BERT-base-uncased					
DistilBERT	2	MSE	AdamW	0.35	2e-05
ELECTRA					
FinBERT					
RoBERTa					
ECCAR					
BART					
BERT	30	CE	AdamW	0.3	3e-07
DEBERTA					
FINBERT					
SMAR					
Macbert	5	CE	Adam	0.3	1e-5
Bard	7	CE	Adam	0.35	2e-05

Where: *LF – Loss Function, MSE – Mean Square Error, CE – Cross Entropy, LR – Learning Rate*

2.3 Results and Discussion

Pretrained Models. Six pretrained transformer-based language models were taken into consideration for this study. Each model underwent individual fine-tuning for binary classification tasks. The optimization of parameters and hyperparameters was accomplished through a stratified 10-fold approach applied to the unified dataset, a combination of Train and Development Set, with configurations chosen as indicated in Table 1. The findings of the experiment demonstrate that ELECTRA [4] achieved the most noteworthy performance, yielding a Macro F1 Score of 76.1179%. Subsequently, the other models followed in descending order of performance: RoBERTa [5], BERT-base-uncased [6], FinBERT [3], ALBERT [7], and DistilBERT [8].

Voting Performance on Training Set. Analyzing the performance of the voting approach using predictions from the top-k models in Table 2 reveals a marginal improvement in results during our experimentation. Our findings underscore the effectiveness of leveraging diverse pre-trained language models. Notably, our method demonstrates that the most optimal outcome is achieved by Top 4 (H), which employs hard voting on a combination of ELECTRA, RoBERTa, BERT-base-uncased, and FinBERT. This ensemble model achieved a notable 76.6231% on the Macro F1 score. Remarkably, the next two high-performing models also emerged from the voting mechanism, with Top 6 (S) and Top 6 (H) achieving Macro F1 scores of 76.5017% and 76.4809% respectively. Consequently, we selected these top three models, along with their corresponding methodologies, for evaluation on the test set.

Table 2: Language Models and Top-K Voting (Hard & Soft) Results on Unified Set (10-Fold CV)

Methods	Train + Dev Set	
	Macro-F1 (%)	Micro-F1 (%)
Finetuned Language Models		
1. ELECTRA	76.118	76.111
2. RoBERTa	75.912	75.911
3. BERT-base-uncased	75.825	75.802
4. FinBERT	75.653	75.646
5. ALBERT	75.453	75.385
6. DistilBERT	75.281	75.242
Hard Voting		
Top 2 (H)	76.218	76.278
Top 3 (H)	76.411	76.416
Top 4 (H)	**76.623**	**76.668**
Top 5 (H)	76.446	76.462
Top 6 (H)	**76.481**	**76.531**
Soft Voting		
Top 2 (S)	76.185	76.187
Top 3 (S)	76.330	76.336
Top 4 (S)	76.478	76.485
Top 5 (S)	76.356	76.370
Top 6 (S)	**76.502**	**76.519**

Table 3: Language Models and Submissions Results on Test Set

Methods	Test Set	
	Macro-F1 (%)	Micro-F1 (%)
Submissions		
Top 4 (H) [TMUNLP1]	76.551	76.574
Top 6 (H) [TMUNLP2][1]	75.826	75.851
Top 6 (S) [TMUNLP3]	76.036	76.058
Finetuned Language Models		
3. BERT-base-uncased	76.424	76.471
1. ELECTRA	76.420	76.471
2. RoBERTa	76.049	76.058
4. FinBERT	75.718	75.748
6. DistilBERT	74.489	74.510
5. ALBERT	73.032	73.168

[1]Value is different from the overview paper, as there was a bugged and has been resolved.

Submission Performance on Test Set. For ECC argument unit classification, the final models - TMUNLP1, TMUNLP2, and TMUNLP3 - mirror the configurations of Top 4 (H), Top 6 (H), and Top 6 (S) respectively.

Table 3, results illustrate that the Top 4 (H) model [TMUNLP1] excels, achieving a Macro F1 score of 76.5513%. Notably, this result outperforms both Top 6 (H) [TMUNLP2] and Top 6 (S) [TMUNLP3] with a difference of 0.1271% from BERT-base-uncased, the highest-performing among the finetuned language model. Despite ELECTRA's superior performance during training, designating it as the final model would not have yielded optimal results on the test set, potentially overlooking BERT-base-uncased's strong performance. Thus, the incorporation of voting mitigates the risk of disregarding the potential of other models, resulting in an improved score compared to individual models' performance.

Table 4: Unsubmitted Top-K Voting (Hard & Soft) Results on Test Set

Methods	Test Set	
	Macro-F1 (%)	Macro-F1 (%)
Hard Voting		
Top 2 (H)	76.658	76.780
Top 3 (H)	76.155	76.161
Top 5 (H)	75.746	75.748
Soft Voting		
Top 2 (S)	76.345	76.367
Top 3 (S)	76.348	76.367
Top 4 (S)	77.083	77.090
Top 5 (S)	76.656	76.677

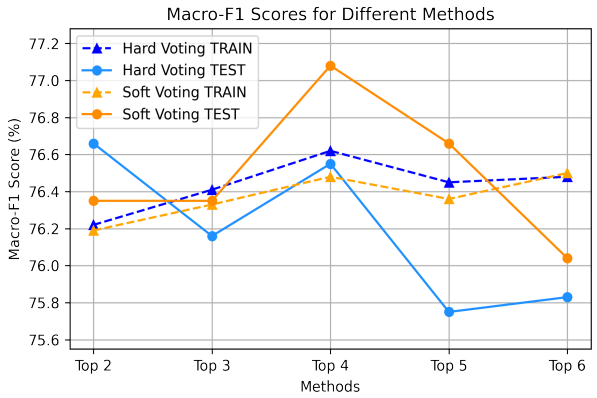

Figure 1: Line Graph of Different Top-K Voting (Hard & Soft) Macro-F1 Score on Training Phase and Testing Phase

Remaining Top-k Performance on Test Set. To delve deeper into the efficacy of voting and its variants, both the Hard and Soft methods, we conducted an internal evaluation of additional Top-k models from the training phase that were not submitted and recorded it in Table 4. This evaluation aimed to assess their performance on the test set.

Figure 1 shows a clear contrast between the train and test set results for both voting techniques, indicating some instability. We conclude that one reason is due to the fact that the top-k language models were ranked based on the training phase, but their rankings changed when they were retrained on the whole dataset and evaluated on new unseen data. As table 2 and 3 demonstrate, only top-3, 4, and 6 had the same set of language models in both phases,

while top-2 and top-5 had different combinations, resulting in higher variance between the train and test set results.

Moreover, the comparison between soft and hard voting methods reveals interesting insights. Although soft voting achieved the best score on top-4 in the test set, it also had a large gap from its train set performance, suggesting that it is less generalizable and more sensitive to data changes. This is also evident in top-6, where both soft and hard voting performed poorly in the test set. However, another perspective is that soft voting outperformed hard voting in all of top-3, 4, and 6, implying that it has a better ability to capture the nuances of new data than the decisive hard voting, which uses binary values to classify texts.

In summary, the analysis highlights (1) the importance of selecting the right models for voting, which should have consistent combinations across phases, (2) the importance of choosing the appropriate voting method, which can have significant differences in performance, and (3) the need for further experiments to investigate the advantages of soft voting.

3 ECC ARGUMENT RELATION SUBTASK

In this section, we provide an overview of the methods employed by the TMUNLP team in ECC Argument Relation subtask (ECCAR). The primary goal of this task is to detect the relationship between two given sentences in the dataset. This subtask is treated as a multi-class classification problem. Section 3.1 briefly outlines the dataset and the preprocessing steps we undertook. In Section 3.2, we delve into the variations in methods for each submission. Finally, in Section 3.3, we conduct an in-depth examination of the experimental results, offering a comprehensive analysis and discussion of the results.

3.1 Dataset and Preprocessing

Dataset Overview. The provided official dataset [2] exhibits a distinct imbalance in label distribution. In this task, a three-class classification is conducted based on two given sentences. Specifically, the training dataset consists of 1600 instances labeled as 'no detected relation', 3859 instances labeled as 'Support', and a mere 62 instances labeled as 'Attack'. Similarly, the development dataset displays an imbalanced nature, with 482 instances labeled as 'no detected relation', 200 instances labeled as 'Support', and only 8 instances labeled as 'Attack'. It's worth emphasizing that, to improve the training process, we opted to merge the training and development datasets, leading to a combined dataset where the label proportions remain significantly skewed: 'no detected relation' at 28.98%, 'Support' at 69.89%, and 'Attack' merely accounting for 1.13%.

General Preprocessing. In the pursuit of natural language inference tasks, we employed a variety of pretrained language models. During the preprocessing stage, akin to ECCAU, we employed the tokenizers of these pretrained language models for tokenization in order to preserve the original syntactic structure. This process involves tokenizing the text, adding [CLS] and [SEP] tokens, and padding or truncating the text to a predetermined length (512). Following this preprocessing, input_ids and attention_mask, which are utilized as inputs for subsequent model training, were acquired.

3.2 Methods

Log Likelihood Ratio (LLR)[1]. In the context of analyzing the dataset's content, particularly concerning the relationship between sentence1 and sentence2, we formulated the concept of employing the LLR (Log-Likelihood Ratio) method. This approach serves the purpose of determining the relevance of words within the text of both sentence1 and sentence2, subsequently computing their corresponding relationship scores.

LLR, a measure of word relationships, provides valuable insights into the connections between labels and tokens within datasets. In contrast to conventional word relationship methods, the LLR method we employed in this specific task is a customized variant developed by our team. Rather than focusing solely on word relationships, this methodology is anchored in the labels themselves. It assesses the relationship between labels and tokens present in sentences, identifying words that hold higher relevance to specific labels. These words are then ranked based on the strength of their relationship with the textual content, sorted from high to low relationship strength.

Considering the task's scope and the unique characteristics of ECCAR, we initially pair words from the texts of sentence 1 and sentence 2. Subsequent to this pairing, we derive LLR scores from these word pairs for each corresponding label category. Given the inherent imbalance in data distribution, we prioritize significant words for label categories with fewer instances. Specifically, for label "0," we select the top 500 significant words. For label "1," we choose the top 300 significant words, and for label "2," we opt for the top 2000 significant words.

Building on the selection of these noteworthy keywords for each label, we proceed to search through the text. Any words that match these crucial keywords are assigned a '1' in a newly created column. Conversely, words that do not match receive a '0' designation. This step results in the generation of several new datasets, each augmented with additional columns corresponding to these significant keywords.

Lastly, the identified significant words are utilized as features within the framework of a DNN (Deep Neural Network) model. To enhance the final prediction accuracy of the task, our team employs an ensemble technique involving the predictions of the LLR-DNN model alongside predictions from other model techniques. This concerted approach aims to elevate the task's predictive accuracy and overall stability, thereby facilitating more precise and consistent predictions throughout the task's execution.

Table 5: Comparison of Different Sampling Strategies

Methods	Micro-F1 (%)			Macro-F1 (%)		
	OG	RS	CL	OG	RS	CL
BART	80.87	75.65	**81.30**	50.69	51.15	**51.51**
LLR	**68.99**	51.30	52.32	34.58	38.98	**40.19**
DEBERTA	72.03	67.25	**77.25**	44.98	45.57	**47.34**

Model Selection. To address the task of determining the relationship between two sentences, the team employed a strategy of selecting pretrained models that have been trained on widely used large-scale natural language inference datasets, such as Multi_nli[10] and Snli[11], to ensure high performance in the subtask. This selection not only contributes to enhancing the baseline performance of the models but also exhibits high performance in semantic inference. Consequently, we decided to utilize two outstanding pretrained models, BART[12] and DEBERTA[13], as our primary frameworks. The strength of the BART model lies in its encoder-decoder architecture, enabling it to comprehend textual semantics and generate coherent text. This capability empowers BART to effectively handle textual correlations, extracting underlying semantic information. On the other hand, DEBERTA is renowned for its attention mechanism-based network structure, which facilitates capturing both intra-sentence and inter-sentence correlations, thereby further elevating the precision of semantic inference.

In addition to model selection, we integrated diverse sampling methods and voting mechanisms into the overall architecture. This integrated strategy ensures that our models excel in handling various types of sentence relationships. By combining the strengths of the LLR-DNN, BART and DEBERTA models, we are able to leverage the advantages of different models, offering a more comprehensive and accurate approach to information extraction from multiple perspectives. This, in turn, enables a more comprehensive understanding and assessment of the relationship between the two sentences.

Imbalanced Data. When dealing with the ECCAR task data, we also encountered the issue of data imbalance. To address this concern, we employed three sampling strategies: Random Sampling[14], Class Weighting[15], and SMOTE (Synthetic Minority Over-sampling Technique)[16].

Random Sampling. We utilized the Random Sampling method to address data imbalance. This common and straightforward approach involves randomly selecting samples from different classes to create a balanced training set. The advantage of this approach is that it ensures each class has sufficient representation, allowing the model to comprehensively learn the features and differences among different classes.

SMOTE generates synthetic samples for minority classes to balance the class distribution in the dataset. This method effectively augments training samples for minority classes, thereby enhancing the model's understanding and classification capabilities for these classes.

Table 6: Final Performance of our Submissions on Test Set

Methods	Micro-F1 (%)	Macro-F1 (%)
BART (CL) & LLR (RS) [TMUNLP-2]	82.03	57.90
BART(CL) & LLR (RS) & DEBERTA (CL) [TMUNLP-1]	81.88	57.36
BART(CL) [TMUNLP-3]	81.88	56.72

Through the combined utilization of these three sampling methods, we were able to train more generalized models in the presence of data imbalance. Random Sampling ensured that each class's samples received sufficient attention, Class Weighting reinforced the model's ability to learn from minority classes and SMOTE further increased training samples for minority classes. As a result, we successfully improved the model's performance in handling data imbalance challenges, effectively overcoming the hurdles posed by data imbalance.

Class Weighting Sampling. This approach involves adjusting the weights of different classes in the loss function. For classes with fewer samples, we increased their weights in the loss function, directing the model's focus towards these minority classes during training and emphasizing their importance. This treatment aids in balancing the model's training across all classes and enhancing its performance in classifying minority categories.

Model Weighting. To enhance overall performance, the implementation incorporated model weighting. Optimal weights for each model were determined by maximizing the F1 Score in the Validation Set. The identified weight combination for the candidate models was then applied in the Final Model for the test set.

Voting. The model construction involved fine-tuning the mentioned models to achieve optimal performance, complemented by the introduced weighting. Evaluation was based on the macro F1-score. The soft voting method outlined by ECCAU was employed, organizing different sampling strategies for each model and combining them for voting. Notably, no threshold was applied; instead, the Argmax of the sum of probabilities by the weighted models in a given class was determined as the final prediction. The top-performing three combinations were selected for submission as the answer.

3.3 Results and Discussion

In our final submission, we primarily utilized three models, with configurations founds in Table 1, in various combinations with different sampling strategies. Table 5 presents the performance of

each model when the train/dev datasets were merged and then split into an 8/2 ratio. The model names are explained as follows: OG (Original) without any sampling strategy, RS (Re-Sampling – Random Sampling: BART & DEBERTA, SMOTE: LLR), and CL (Class Weight).

The final arrangement of the voting combinations is based on the top three sets from the table, ranked by performance as follows: BART (CL) & LLR (RS), BART (CL) & LLR (RS) & DEBERTA (CL), BART (CL).

It is noteworthy that in Table 6, although the performance of the LLR sampling strategies may not outshining the performance of the two pretrained models, they possess the advantage of capturing features from the minority labels in imbalanced data. This superior capability, combined with BART's contextual understanding and DEBERTA's attention mechanism within its architecture, contributes to an enhanced performance in natural language inference tasks.

To enable the replication of our approach in ECCAR subtask, you can find the source code on our GitHub repository under the name nlptmu/FinArg-1_ARC_BDF4NLI

4 SM ARGUMENT RELATION SUBTASK

In this section, we provide an overview of the methods employed by the TMUNLP team in Social Media Argument Relation Subtask (SMAR). The primary goal of this task is to detect the relationship between two given sentences in the dataset. This subtask is treated as a multi-class classification problem. Section 4.1 briefly outlines the dataset and the preprocessing steps we undertook. In Section 4.2, we delve into the methods for our submission. Finally, in Section 4.3, we conduct an in-depth examination of the experimental results, offering a comprehensive analysis and discussion of the results.

4.1 Dataset and Preprocessing

Dataset Overview. The dataset for this subtask demonstrates balanced proportions, comprising 4,596 entries for "Support", 2,698 entries for "Attach" and 854 entries for "Other." The data is divided into train, dev, and test sets, with a ratio of 80% for training, 10% for development, and 10% for testing. In the training phase, we adopt an approach to dataset augmentation. Instead of evaluating our models on the train and development sets separately, we treat them as a unified set. Employing a stratified 10-Fold approach enhances the robustness of our evaluation methodology.

Preprocessing. Upon reviewing the data, we eliminated "\n", URLs, and emoji symbols apart from the "🐔" emoji. The chicken emoji, being symbolic of significant corporate entities and carrying distinct importance. Consequently, we substituted all instances of the chicken emoji with the character "積". Furthermore, we detected a blend of simplified Chinese and English content within the data. To uphold data uniformity, we utilized Google Translate to translate the entire content into traditional Chinese.

4.2 Methods

Data Augmentation. In the training stage of our model, to enhance the model's learning from training data, we employed a straightforward duplication approach for the content of both Post1 and Post2. This doubling of training material aimed to effectively amplify data diversity within the constraints of limited information[17]. By augmenting the training data in this manner, we aimed to improve the model's capacity to capture textual correlations and contextual features.

ChatGPT. Furthermore, we propose an innovative approach, in response to the prevalent use of ChatGPT, we also endeavored to enrich our research by leveraging this innovative technique. We employed ChatGPT to extract the most relevant finance-related keywords from the textual content of Post1 and Post2. Given the inherent length of our sentences, we sought to identify the most helpful 'n' keywords for model learning within the shortest word count. After several keyword quantity experiments, we concluded that extracting five keywords would be optimal. These ten keywords, collectively obtained from Post1 and Post2, were amalgamated into a single text string, forming a novel training feature. This allowed the model to learn significant keywords from the text, thereby enhancing predictive accuracy.

Ultimately, we formatted the three segments – Post1, Post2, and the keyword string – by adding [CLS] and [SEP] markers. This formatting facilitated the truncation of the text into multiple sentences, enabling tokenization for input into the model[18].
Throughout the 10-fold training process, multiple models were generated. To determine the most proficient model for predictions, we employed the Macro F1 score, a competition evaluation criterion, as the standard for model selection. This rigorous approach ensured that the predictive performance of the final model reached optimal levels[19].

To support the reproducibility of our methodology for the SMARsubtask, you can access the source code from our GitHub repository at nlptmu/FinArg-1_ARI_MacB2.

4.3 Results and Discussion

In the final submission, we primarily utilized two models, with configurations found in Table 1, along with Soft Voting. Table 7 showcases the performance of various models trained using the train dataset and predicted on the development dataset. The explanation of the model methodology is as follows: "add_feature" represents the inclusion of keyword features generated using Chat-GPT, while "no_add" indicates the absence of these Chat-GPT-generated keyword features. We opted to use the pre-trained BERT model from Bardai[20], as well as Macbert[21] , for fine-tuning our BERT models to enhance classification performance. We chose Bardai because its design aligns closely with the requirements of our competition; it is a sentiment analysis language model

Table 7: Comparisons of Models with & without Keyword Feature

Methods	Dev Set	
	Macro-F1 (%)	Micro-F1 (%)
Macbert_add_feature	75.93	74.36
Macbert _no_add	70.32	68.83
Bard_ add_feature	71.84	68.10
Bard _ no_add	71.63	67.73
Voting_ add_feature	76.01	74.36
Voting_ no_add	71.28	69.57

specifically tailored for the financial domain. The reason for selecting Macbert is its superior performance in sentiment analysis tasks on Simplified Chinese corpora compared to Bert, RoBERTa, and ELECTRA[21]. Additionally, it exhibited the best performance within our dataset. Ultimately, we employed soft voting to combine the outputs of these two models and find a balanced answer.

Given that the chosen models were optimized for Simplified Chinese, we took steps to ensure accurate vocabulary embeddings[22]. To accomplish this, we employed Google Translate to fully convert the Traditional Chinese content within the dataset to Simplified Chinese.

A random sampler was mistakenly applied on the test data, which shuffled the index of the results. Due to a human error, we submitted an incorrect result file that did not reflect the real performance of our final model. Table 8 represents the result of our revision into a sequential sampler.

Based on the insights gained from the dev set, we observed improved results by incorporating keyword features and utilizing the distinct strengths of the two models through soft voting. This enhancement contributed to an overall improvement in predictive accuracy, leading us to choose the model with keyword features for predictions on the test set (refer to Table 8 for details).

However, the results on the test set indicated a slight decrease in performance when employing soft voting compared to without. This discrepancy might stem from insufficient model selection or contrasting characteristics between the chosen models, lacking the diversity observed in Section 2 ECCAR, where we experimented with various BERT model combinations and voting methods. Another factor could be the non-optimization of the voting method, unlike the approach taken in Section 3 ECCAU, where we conducted and tested Model Weighting.

This highlights the importance of thorough experimentation to fully harness the benefits of voting for performance improvement. While it demonstrated efficacy in the dev set, its performance in the test set may not be as consistent, emphasizing the need for careful consideration and optimization in the application of voting methods.

Table 8: Results of Submissions on Test Set

Methods	Test Set	
	Macro-F1 (%)	Micro-F1 (%)
Submissions		
Bard [TMUNLP1]	71.10	67.61
Macbert [TMUNLP2]	73.39	73.13
S. Voting [TMUNLP3]	73.12	70.18

5 CONCLUSIONS

5.1 ECC Argument Unit Identification

In conclusion, the ECC Argument Unit Identification methodology effectively utilizes a voting approach to leverage the strengths of various pre-trained language models.

The model's impressive performance is from the combination of predictions derived from multiple finetuned language models, utilizing both the nuanced soft and decisive hard voting techniques. By thoughtfully selecting the most effective models and merging their collective predictions, the methodology significantly enhances predictive accuracy.

Of particular note is the combination of ELECTRA, RoBERTa, BERT-base-uncased, and FinBERT models using the robust hard voting strategy, resulting in a noteworthy Macro F1 score of 0.766231. Further evaluation on the test dataset highlights the superiority of Soft Voting in this context. Soft Voting's ability to excel on the test set emphasizes its adaptability to new and varied data, a trait where Hard Voting falls short.

This study underscores the advantage of the ensemble strategy employed, showcasing its potential to improve ECC Argument Unit Identification outcomes. Moreover, its implications may extend to enhancing various other NLP tasks by harnessing the diverse capabilities of different language models.

5.2 ECC Argument Relation Identification

This study focuses on the task of natural language inference and combines various pretrained language models with different sampling strategies for development. In the preprocessing stage, we maintain syntactic structures and incorporate the BART and DEBERTA models for the task. Through the Log Likelihood Ratio (LLR) method, we capture word relationships and combine them with the features of BART and DEBERTA.

Regarding the issue of data imbalance, we employ two strategies: random sampling and class weighting, ensuring thorough training of the model across all categories. By merging models through a voting mechanism, we enhance their performance in natural language inference tasks.

In the future, further research could explore the integration of preprocessing, model selection, and sampling strategies to enhance

semantic inference. Additionally, optimizing the integration methods will contribute to improving the efficiency and stability of models in handling semantic inference tasks.

5.3 SM Argument Relation Identification

In conclusion, our exploration of various models consistently underscores the positive impact of integrating ChatGPT-generated keyword features, showcasing its potential for academic advancements.

Notably, insights from the dev set reveal improved results through the strategic use of soft voting, leveraging the strengths of two models. While this approach enhances predictive accuracy, the test set demonstrated a slight decrease in performance compared to an alternative.

Potential factors include insufficient model selection and inherent contrasts between models, lacking the diversity observed in previous experiments. Additionally, non-optimization of the voting method may contribute to this discrepancy.

These findings emphasize the importance of thorough experimentation for optimal model performance. While soft voting proves effective in the dev set, its inconsistent performance on the test set highlights the need for careful consideration and optimization, reflecting the iterative process of refining models for specific tasks.

Acknowledgements

This study was supported by the National Science and Technology Council under grant 112-2410-H-038-007- and 112-2622-E-038-001-, as well as the National Health Research Institutes NHRI-12A1-PHCO-1823244. Yung-Chun Chang is the corresponding author.

REFERENCES

[1] C hen, C. C., Lin, C. Y., Chiu, C. J., Huang, H. H., Alhamzeh, A., Huang, Y. L., Takamura, H., & Chen, H. H. (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. *doi: 10.20736/0002001323*

[2] Alhamzeh, A.a., et al. (2022), It's Time to Reason: Annotating Argumentation Structures in Financial Earnings Calls: The FinArg Dataset. Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP), p. 163--169.

[3] Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

[4] Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

[5] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

[6] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[7] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

[8] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[9] Chang, Y. C., Ku, C. H., & Le Nguyen, D. D. (2022). Predicting aspect-based sentiment using deep learning and information visualization: The impact of COVID-19 on the airline industry. *Information & Management*, *59*(2), 103587.

[10] Williams, A., Nangia, N., & Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

[11] Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

[12] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

[13] He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

[14] Johnson, J. M., & Khoshgoftaar, T. M. (2020). The effects of data sampling with deep learning and highly imbalanced big data. *Information Systems Frontiers*, *22*(5), 1113-1131.

[15] Xu, Z., Dan, C., Khim, J., & Ravikumar, P. (2020, November). Class-weighted classification: Trade-offs and robust approaches. In *International Conference on Machine Learning* (pp. 10544-10554). PMLR.

[16] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[17] Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. Array, 100258.

[18] Cheng, X. (2021, July). Dual-view distilled bert for sentence embedding. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2151-2155).

[19] Harbecke, D., Chen, Y., Hennig, L., & Alt, C. (2022). Why only micro-f1? class weighting of measures for relation classification. arXiv preprint arXiv:2205.09460.

[20] Manyika, J. (2023). An overview of Bard: an early experiment with generative AI. Technical report, Google AI.

[21] Cui, Y., Che, W., Liu, T., Qin, B., & Yang, Z. (2021). Pre-training with whole word masking for chinese bert. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29, 3504-3514.

[22] Zhang, X., Li, P., & Li, H. (2020). AMBERT: A pre-trained language model with multi-grained tokenization. arXiv preprint arXiv:2008.11869.