FTRI at the NTCIR-18 FinArg-2 Task: Identify Temporal Reference in Earnings Conference Calls

Adhitia Erfina and Phuong Le-Hong
FPT Technology Research Institute – Vietnam National University

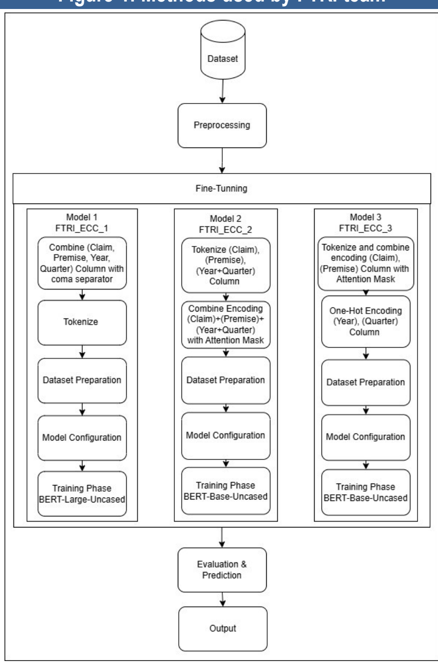
Introduction

Argument mining, an extension of opinion mining, has rapidly emerged as a hot research topic in recent years. Not only to capture someone's opinion, but also argument mining aims to investigate the reason behind the opinion[1]. FinArg-2 aims to introduce "Temporal Inference of Financial Arguments" focusing on the assessment of temporal information, which is a distinct phenomenon in financial opinions. FinArg-2 will continue utilizing the same resources as FinArg-1, including analyst reports, earnings conference calls, and social media data [2]. Earnings Conference Calls (ECCs) are structured quarterly meetings where company executives communicate financial updates, discuss past performance, and outline future business expectations. These calls serve as a crucial platform for maintaining transparency with investors, analysts, and stakeholders by providing direct insights into a company's financial health and strategic direction. ECCs are divided into three key sections [2], [3]:

1. **Safe Harbor Statement** – A disclaimer addressing potential uncertainties in forward-looking statements.
2. **Presentation** – A review of financial outcomes and future expectations by company executives.
3. **Q&A Session** – Analysts pose questions, and executives provide clarifications and justifications.

FTRI participates in NTCIR-18 FinArg-2 on the ECC subtask, where models must identify the temporal reference associated with an argument. Each participant was given the opportunity to submit the 3 best results based on the model that we had created. As for the evaluation results shown in table 1, our 3 output models were in the top 4 among other participants based on Micro F1 and Macro F1 [4].

Figure 1: Methods used by FTRI team



Result (2)

FTRI_ECC_2
Considering that the results obtained from previous experiments were less than optimal, we conducted improvement experiments.

We do BERT encoder on the Claim to have vector x1, BERT encoder on the Premises to have vector x2, concatenate x1 and x2 to have a vector x. And BERT encoder on the Year to have vector y1, BERT encoder on the Quarter to have vector y2, concatenate y1 and y2 to have vector y. Analyze x+y to Label with attention mask.

Table 3: FTRI_ECC_2 Result

Model	Epoch	Micro F1	Macro F1
BERT	5	69.05%	65.76%
BERT	8	68.97%	67.06%
BERT-Large	3	58.33%	54.07%
BERT-Large	6	66.43%	68.58%
RoBERTa	4	59.21%	57.26%
RoBERTa	5	68.48%	66.10%

Table 1: ECC Evaluation Result

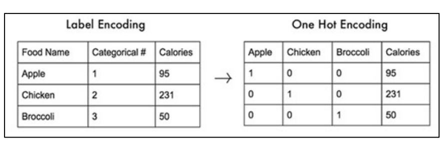
Team Name	Micro F1	Macro F1
FTRI_ECC_3	77.38%	75.07%
FTRI_ECC_1	71.43%	68.58%
SCaLAR IT Team_ECC_1	70.24%	67.85%
FTRI_ECC_2	69.05%	65.76%
IMNTPU_ECC_1	69.05%	67.06%
TMUNLPG1_ECC_2	69.05%	66.13%
AIDAVANCE_ECC_1	69.05%	67.11%
AIDAVANCE_ECC_3	69.05%	66.10%
SCUNLP-1_ECC_3	67.86%	64.94%
SCUNLP-1_ECC_1	66.67%	63.06%
AIDAVANCE_ECC_2	66.67%	61.05%
SCUNLP-2_ECC_2	66.67%	63.41%
SCUNLP-2_ECC_3	66.67%	63.37%
IMNTPU_ECC_3	65.48%	62.44%
TMUNLPG1_ECC_3	65.48%	64.45%
Trustworthy_ECC_1	63.10%	58.67%
IMNTPU_ECC_2	63.10%	57.87%
SCUNLP-2_ECC_1	63.10%	59.54%
TMUNLPG1_ECC_1	61.90%	56.32%
Trustworthy_ECC_2	60.71%	52.75%
SCUNLP-1_ECC_2	58.33%	52.07%
SCaLAR IT Team_ECC_2	35.71%	32.27%

Method (2)

Fine-tuning
The fine-tuning stage in Model 1 combines the Claim, Premise, Year, and Quarter columns into one separated by commas. After merging the text, it is tokenized with BertTokenizer. The results of several experiments show that BERT-Large-Uncased performs slightly better than the other models with the configuration train_batch=8, eval_batch=8, train_epochs=5.

While in Model 2 we tokenized the Claim, Premise, and Year + Quarter columns. The next stage is combine-encoding the entire column with attention mask added. The configuration we use is train_batch=8, eval_batch=8, train_epochs=7 with BERT-Base-Uncased. The purpose of the attention mask is to tell the model which tokens to pay attention to and which ones to ignore.

And in Model 3 we tokenize the Claim, and Premise columns and combine encoding with an attention mask. In addition, in the Year and Quarter columns we added an extra one-hot vector encoding feature to represent categorical data into binary numeric form. Considering they are categorical variables of several values, which will make it more sense explicitly. The configuration we use is train_batch=8, eval_batch=8, train_epochs=7 with BERT-Base-Uncased.



Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

Figure 3: One-hot Encoding Example [13]

Result (3)

FTRI_ECC_3
We conducted encoder like previous experiment plus TF-IDF. And one-hot encoding on the Year to have vector y1, one-hot encoding on the Quarter to have vector y2, concatenate y1 and y2 to have vector y. Analyze x+y to label with attention mask. Table 4 shows the result.

Table 4: FTRI_ECC_3 Result

Model	Epoch	Micro F1	Macro F1
BERT	5	77.38%	75.07%
BERT	10	71.07%	69.66%
BERT-Large	5	60.33%	54.07%
BERT-Large	8	70.43%	69.43%
RoBERTa	6	70.06%	68.21%
RoBERTa	10	68.26%	63.06%

Method (1)

The first step is to find the best pretrained model as the baseline model that we will fine-tune with several experiments using variations in the preprocessing and training stages. In addition, realizing that the quantity of training data is not much, we added validation data to the training data to make the amount of training data larger.
We conducted several experiments with various models that are considered to perform well in argument mining such as DistilBERT [5], BERT [6], RoBERTa [7], FinBERT [8], DeBERTa [9], as well as the large version of each model [7], [10]. The chosen model is part of a family of autoencoder models that focus on text comprehension, using encoder-only architecture. The overall method used is described in Figure 1.

Dataset
In this subtask, we were given 600 train data and 150 validation data. The given dataset consists of:

- **Claim**: A statement of fact, opinion, or belief regarding a company's performance, prospects, or other aspects of business.
- **Premise**: Reasons, evidence, or justifications that support the claim.
- Year: 2015 to 2019
- **Quarter**: Q1-Jan to Mar, Q2-Apr to Jun, Q3-Jul to Sep, Q4-Oct to Dec.
- **Label**: 0-No time reference, 1-Long past (more than half a year), 2-Short past (less than half a year): during this quarter or up to 2 quarters.

Result (1)

At this stage, we have conducted several experiments with various models that are considered to perform well in argument mining. We use the same configurations for all models and only differ in the train epoch.
FTRI_ECC_1
In this model we use BERT encoder on the Claim to have vector x1, BERT encoder on the Premises to have vector x2, concatenate x1 and x2 to have a vector x. Analyze x to Label.
According to several experiments conducted on multiple models, BERT-Large, BERT, and RoBERTA models have better accuracy scores compared to other models. In some models, the accuracy results are also influenced by how many epochs of training are performed in the system. The following table explains the results of several experiments.

Table 2: FTRI_ECC_1 Result

Model	Epoch	Micro F1	Macro F1
BERT-Large	5	71.43%	68.58%
BERT-Large	3	58.33%	54.07%
BERT	10	68.97%	67.06%
BERT	5	59.71%	51.25%
RoBERTa	3	67.06%	64.94%
RoBERTa	4	68.48%	66.10%
RoBERTa-Large	3	58.33%	56.07%
RoBERTa-Large	4	66.67%	63.06%

Conclusion

Based on the experiments we conducted in the NTCIR-18 FinArg-2 ECC Subtask, we generated 3 outcomes based on the 3 plans in our model with the following results:

A. **FTRI_ECC_1:** This model ranked 2nd overall with a score of 71.43% MicroF1 and 68.58% MacroF1.
B. **FTRI_ECC_2:** This model ranked 4th overall with a score of 69.05% Micro F1 and 65.75% Macro F1.
C. **FTRI_ECC_3:** BERT encoder on the Claim to have vector x1, BERT encoder on the Premises to have vector x2, concatenate x1 and x2 to have a vector x plus TF-IDF. And one-hot encoding on the Year to have vector y1, one-hot encoding on the Quarter to have vector y2, concatenate y1 and y2 to have vector y. Analyze x+y to label with attention mask. This model ranked 1st overall with a score of 77.38% Micro F1 and 75.07% Macro F1.

ACKNOWLEDGMENTS

We would like to express our deepest gratitude to FPT Technology Research Institute and FPT Smart Cloud for providing us with their full support so that we can achieve the expected results.