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/10.20736/0002001281

LIPI at the NTCIR-17 FinArg-1 Task: Using Pre-trained Language Models for Comprehending Financial Arguments

Swagata Chakraborty
Anubhav Sarkar
swagatac652@gmail.com
sarkaranubhav2001@gmail.com
Christ (Deemed to be University)
Bangalore, India

Dhairya Suman
dhairyasuman@gmail.com
Indian Association for the Cultivation of Science
Kolkata, India

Sohom Ghosh
Sudip Kumar Naskar
sohom1ghosh@gmail.com
sudip.naskar@gmail.com
Jadavpur University
Kolkata, India

ABSTRACT

Comprehending arguments from financial texts helps investors in making data driven decisions. The FinArg tasks of NTCIR-17 deal with mining arguments related to finance from Research Reports, Earnings Conference Calls, and Social Media. In this paper, we describe our team's approach to solve the three such problems - Argument Unit Classification, Argument Relation Detection & Classification, and Identifying Attack and Support Argumentative Relations. We obtained best performance using pre-trained language models (like BERT-SEC and FinBERT) and cross-encoder architecture.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Applied computing** → **Document management and text processing**; *Economics*; • **Computing methodologies** → **Information extraction**.

KEYWORDS

argument analysis, financial natural language processing, large language models

TEAM NAME

LIPI

SUBTASKS

FinArg-1: Argument Identification (Argument Unit Classification, Argument Relation Detection and Classification), Identifying Attack and Support Argumentative Relations in Social Media Discussion Threads

1 INTRODUCTION

Earning call transcripts are an important source to know more about the financial performance of any organization. With the advent of social media, investors tend to discuss various investment strategies online. The FinArg-1 shared task [4] co-located with NTCIR-17 deals with mining arguments from financial texts. In this paper, we discuss various approaches we followed for identifying argument units and relations in earning call transcripts and social media posts. This corresponds to Task-2 and Task-3 as mentioned in [4]. The dataset for Task-2 was in English while that for Task-3 was in Chinese. Furthermore, for the task of Argument Relation Identification, we also explored the applicability of Large Language Models under zero-shot and few-shot settings.

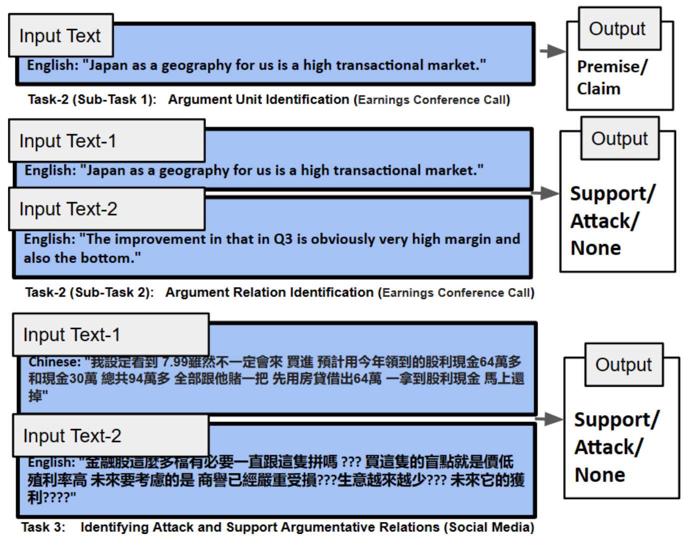

Figure 1: Argument Analysis in Financial Texts

2 PROBLEM STATEMENT

<u>**Task 2, Sub-Task 1:**</u> Given a financial argumentative text in English, we want to classify it as premise or claim.
<u>**Task 2, Sub-Task 2:**</u> Given two financial argumentative texts in English, our aim is to detect the relation between them. The relation can be 'Support', 'Attack', or None.
<u>**Task 3:**</u> Given two argumentative social media posts relating to finance in Chinese, the objective is to classify the relation between them. The relation can be 'Support', 'Attack', or None.
Chen et al. [4] and Alhamzeh et al. [1] described the tasks and datasets in more detail. We present this in Figure 1.

3 SYSTEM DESCRIPTIONS

In this section, we discuss our best performing systems.

3.1 Task 2: Argument Identification

The Argument Identification task consists of two sub-tasks: Argument Unit Classification and Argument Relation Identification.

3.1.1 Sub Task 1: Argument Unit Classification. In this task, we had to identify and classify whether the given sentence was a claim or a premise. The training data had 7,753 sentences, and validation data had 969 sentences. In the training data given, the distribution was quite balanced, with 52.4% of the sentences labelled as claims and the remaining as premises. A similar distribution was seen with the validation data as well. After experimenting with various models, we found that a BERT-SEC [10] model trained for 5 epochs and a batch size of 32 performed the best (Micro-F1: 73. 89%, Macro-F1: 73. 86% in the test set).

3.1.2 Sub Task 2: Argument Relation Detection and Classification. In the given training dataset, we had 5,521 pairs of labelled sentences from which we had to identify and classify the relationship between them as support, attack, or none. In the validation data set, we had 690 pairs of labelled sentences. We identified the high class imbalance in the given dataset, so we had made an attempt to up-sample the minority class by paraphrasing the existing sentence pairs. The counts of each of the classes are given in table 1. We had used Contextual Word Embedding Augmenter and Synonym Augmenter from NLPAUG [11] library and FLANG-roberta model [15] for paraphrasing sentences. However, paraphrasing was performed only on the training dataset, and hence the validation dataset remains the same.

We further fine-tuned the best performing FinBERT model of Task-1 Sub-Task-2 for classification using the cross-encoder architecture [13]. This fine-tuning was done for 5 epochs with a batch size of 16 on the original dataset. This outperformed all other models we trained (Micro-F1: 79.42%, Macro-F1: 60.22% in the test set).

3.2 Task 3: Identifying Attack and Support Argumentative Relations

In the training dataset, we had 6,518 pairs of labelled Chinese sentences. In the validation dataset, we had 815 pairs of labelled Chinese sentences. To increase the number of instances in the minority class, we paraphrased them. The distribution is presented in Table 1. Our aim was to infer from a sentence pair of social media posts if the argumentative posts were supportive, attacking, or neutral. Since the posts were in Chinese, we divide our work into 2 parts. Firstly, we translated Chinese texts into English using Google Translate. Secondly, we worked with the raw Chinese texts as it is.

By fine-tuning a BERT-SEC [10] model using cross encoder architecture on English texts obtained through translation, we obtained the best results on the test set (Micro F1: 64.79%, Macro F1: 69.45%). This fine-tuning was done with a batch size of 8, for 5 epochs on the original i.e. non-paraphrased dataset.

4 EXPERIMENTS AND RESULTS

In this section, we mention the experiments we performed and their results.

4.1 Task 2: Argument Identification

4.1.1 Sub Task 1: Argument Unit Classification. For this task, we experimented with various classification techniques. We first used a simple Recurrent Neural Networks (RNNs) with a fully connected layer along with the Spacy [8] tokenizer to get our outputs, but this model did not perform well. The second experiment was using the FastText [9] model. This model had far fewer parameters than the previous model. It first calculated the word embedding for each word using the Embedding layer, then calculated the average of all the word embeddings and fed it to the linear layered Neural Network (NN). Next, we replaced the existing embeddings with Glove [12] and fed our embeddings into 3 convolutional layers and then finally to a fully connected layer to get the labels. We used a drop-out of 50%. Finally, we fine-tuned a few pre-trained language models like BERT [7], BERT-SEC [10], and FinBERT [3]. This led to significant improvement in performance. The results are mentioned in Table 2.

4.1.2 Sub Task 2: Argument Relation Identification. Firstly, we concatenated the texts in a given pair with separator ([SEP]) token in between them. We fine-tuned several encoder based pre-trained language models for classification. They are DistillBERT [14], Flang-RoBERTa [15], and BERT-SEC [10]. Subsequently, we fine-tuned the cross encoder [13] architecture with BERT [7], BERT-SEC [10], and FinBERT [3] previously fine-tuned for Task-2 Sub-Task-1 embeddings. The scores of all the models are given in the result section, Table 3. We further tried to adapt the models to the given domain using Masked Language Modelling (MLM). However, this didn't improve the performance. Each model was trained with a batch size of 16 and 5 epochs. All the experiments were performed on the original as well as the paraphrased datasets.

Leveraging Large Language Models. Large Language Models (LLMs) have been re-defining the state of the art in Natural Language Processing. We experimented like Dolly v2 [6] (a LLM) under zero shot and few shot settings.

Few shot learning is a method where we ask a language model to do a task and provide the model with a few examples of the task. Initially, we experimented with a static prompt where we choose one example from each classification category: 'Support', 'Attack', and 'None'. A static prompt is a prompt where the few shot examples are kept fixed with different query. But the performance was not satisfactory. This inspired us to come up with a novel dynamic prompt engineering algorithms where the few shot examples would not be fixed unlike static prompting. The motive of our algorithms is to dynamically choose such examples with each validation query which are similar to the query, hence giving the language model a better understanding of the classification task. Our algorithms have two steps and three steps, respectively. Our first proposed algorithm (Algorithm-1) has two steps, (1) Tweet Topic Classification and (2) Semantic Similarity. The algorithm initially finds the tweet topic of each instances present in train set and validation set. These topics were extracted using pre-trained model [2]. We append these tweet topics to the train set and validation set as columns. Now, we iterate through the validation set. For each validation instance, we choose a sample from the train set whose topic is equal to the topic of validation instance. Since, this task is like Natural Language Inference (NLI), and we have a pair of sentences whose relationship has to be determined, we merge the two sentences for simplicity. This gives us a train corpus whose embedding is found. Similarly, for the validation instance, we merge the sentences and find the embedding. Now, we find the cosine similarity between all the instances present in the train corpus and the validation instance. From this we choose top k sentences having maximum semantic similarity. These two steps ensures that the training examples provided with the validation instance belongs to the same topic as well as have

TASK	DATASET	LABEL	# ORIGINAL	# PARAPHRASED
2-2	Train	0	1600	3200
		1	3859	3859
		2	62	372
	Val	0	200	200
		1	482	482
		2	8	8
3	Train	0	684	4104
		1	3676	3676
		2	2158	3676
	Val	0	85	85
		1	460	460
		2	270	270

Table 1: Count (#) before & after paraphrasing. 2-2 refers to (Task-2, Sub-Task-2)

MODEL	MACRO-F1 (VALIDATION SET)	MICRO-F1 (VALIDATION SET)
RNN + Spacy Tokenizer	0.3571	0.5270
FastText + NN	0.7155	0.7173
GloVe Embeddings + CNN	0.6952	0.6957
BART-BASE-CASED + BERT TOKENIZER	0.7336	0.7337
BERT-SEC	**0.7426**	**0.7430**
FinBERT	0.7398	0.7401

Table 2: Results of Task 1, Sub-Task 1: Argument Unit Identification

Algorithm 1: Dynamic Prompt Engineering Algorithm 1

Data: Train, Validation
Result: Few-shot examples
Step 1: Train Topic ⟵ Tweet Topic Classification (Train)
Step 2:
Val Topic ⟵ Tweet Topic Classification (Validation)
Step 3: Train ⟵ Train Topic
Step 4: Validation ⟵ Val Topic
Step 5: for *each instance in the validation dataset* **do**
  if *Train['Topic'] == Validation['Topic']* **then**
    Train Sample ⟵ sample from the Train record
  else
    Do not sample Train record
  end
  for *each record in the train sample* **do**
    Paragraph ⟵ Merge the two posts of the instance
    Train Corpus ⟵ Paragraph
  end
  Train Embed ⟵ Embeddings (Train Corpus)
  Query ⟵ Merge the two posts of the record
  Query Embed ⟵ Embeddings (Query)
  cosine-sim ⟵
   semantic similarity(Query Embed, Train Embed)
  Extract the top k results from cosine-sim having the
   highest semantic similarity, where
   $k = \min(5, \text{length(train sample)})$
  Use these top results as few-shot examples
end

the highest semantic similarity. However, in this algorithm, we are not ensuring whether each of the examples comes from different classification categories.

Our second proposed algorithm (Algorithm-2) has three steps, (1) Tweet Topic Classification using [2] (2) Semantic Similarity (3) Class Filter. This algorithm overcomes the limitation of the previous algorithm by making sure that the examples provided with the validation query comes from different classes ('Attack', 'Support', 'None') and has similar topic with high semantic similarity.

The results of static prompts and dynamic prompts is provided in the Table 5. Sample prompts have been provided in the Appendix section. From the results, we can observe that prompts curated from Algorithm 1 are performing better than static prompts as well as prompts curated from Algorithm 2. From this observation, we can conclude that it is unnecessary to provide examples from different classes. This would simply add redundant information and noise to the language model, resulting in miss-classification. Thus, Algorithm 1 which curates prompts without the class filter works better than Algorithm 2. This is the first finding. Another observation is, language models tends to predict the first occurring classification category from the example for majority of the validation queries. We have performed three experiments with Algorithm 2, "v1" had "None" category as the first example, "v2" had "Attack" category as the first example, "v3" had "Support" category as the first example. In each of three experiments, the category of the first example became the majority category for prediction. This can be noticed from the confusion matrix given in the table. This is happening possibly because internally the language model is getting biased towards the first mentioned class. This is also a reason why algorithm 2 does not perform as good as algorithm 1. This is the other finding.

Algorithm 2: Dynamic Prompt Engineering Algorithm 2

```
Data: Train, Validation
Result: Few-shot examples
Step 1: Train Topic ← Tweet Topic Classification(Train)
Step 2: Val Topic ← Tweet Topic Classification(Validation)
Step 3: Train ← Train Topic
Step 4: Validation ← Val Topic
Step 5: for each record in the Validation do
    if Train['Topic'] == Validation['Topic'] then
        Train Sample ← sample the Train record
    else
        Do not sample from train set
    end
    for each label in {'Attack', 'Support', 'None'} do
        if Train Sample['label'] == label] then
            Train Sample_label
              ← sample the Train Sample record
        else
            Do not sample Train Sample record
        end
        if length(Train Sample_label>0) then
            for each record in the Train Sample_label do
                Paragraph ←
                  Merge the two posts of the record
                TrainCorpus_label ← Paragraph
                TrainEmbed_label ←
                  Embeddings(TrainCorpus_label)
            end
            Query ← Merge the two posts of the record
            Query Embed ← Embeddings(Query)
            cosine-sim ←
              semanticsimilarity(QueryEmbed, TrainEmbed_label)
            Extract the top k results from cosine-sim having
              the highest semantic similarity, where k =
              min(5, length(train sample))
            Use these top results as few-shot examples
        else
            Do nothing
        end
    end
end
```

4.2 Task 3: Identifying Attack and Support Argumentative Relations

Firstly, we translated the Chinese texts to English so that we could comprehend them. TO address the class imbalance, we paraphrased the English texts belonging to the minority classes. We fine-tuned several encoder based models like BERT-base-Uncased [7] and Flang-RoBERTa [15] for classification after concatenating the texts in a given pair with a separator ([SEP]) token. We experimented with both the original and paraphrased data. Subsequently, we used cross-encoder architecture [13] with embeddings from Distil-RoBERTa [14], FLANG-Roberta [15], and BERT-SEC [10] for both the original and paraphrased datasets. We further used Masked Language Modelling (MLM) to adapt these models to the given domain.

To avoid the loss due to translation, we experimented with the original Chinese Texts as well. Firstly, we converted the raw Chinese text to simplified traditional Chinese texts using zhconv library.[1]. We trained a SBERT-Chinese[2] model for classification. We used the original dataset for training, as we couldn't find and validate a paraphraser suitable for Chinese texts. Subsequently, we replaced the embeddings in the cross-encoder architecture with SBERT-Chinese[3] embeddings and fine-tuned the model further. Each of the cross-encoder models were trained batch size of 8 and 5 epochs to train our dataset. The results are presented in Table 4.

Leveraging Large Language Models. Our experimentation focused on few-shot learning scenarios. Initially, we employed a static prompt strategy, selecting one example from each classification category ('Support', 'Attack', 'None'). However, this approach yielded unsatisfactory performance. This led us to innovate novel dynamic prompt engineering algorithms. The core idea behind these algorithms was to dynamically choose examples during validation that closely resembled the query, providing the LLM with a more profound understanding of the classification task.

As Algorithm-1 performed better than Algorithm-2 for Task 2-2, we experimented only with Algorithm-1 for Task-3. We translated the Chinese texts to English and evaluated Large Language Models like flan-t5-small [5], mpt-1b-redpajama-200b-dolly [4], and dolly-v2-3b [6] under various settings. The results are mentioned in Table 5. More details regarding the prompts are mentioned in Apendix.

5 CONCLUSION

In this paper, we shared our team, LIPI's approach for Argument Unit Classification, Argument Relation Detection, and Identifying Attack & Support Argumentative Relations in English and Chinese financial texts. We observed FinBERT[3] and BERT-SEC [10] based models when fine-tuned using cross encoder architecture performed the best for relation identification. Paraphrasing and pre-fine-tuning using MLM did not help much in improving the performance of the model. LLMs under zero shot and few shot setting did not do as good. For Task-2, our team was ranked 13th and 2nd in sub-task-1 and sub-task-2 respectively. For task-3, we were ranked 4th.

Regarding the limitations, it is necessary to mention that we have not considered semantic loss due to paraphrasing. In future, we would definitely try to improve it and we want to extend this solution to low resources Indian languages and create a user-friendly tool to help investors.

[1] https://pypi.org/project/zhconv/ (accessed on 15th August, 2023)
[2] https://huggingface.co/DMetaSoul/sbert-chinese-qmc-finance-v1 (accessed on 16th August, 2023)
[3] https://huggingface.co/DMetaSoul/sbert-chinese-qmc-finance-v1 (accessed on 16th August, 2023)
[4] https://huggingface.co/mosaicml/mpt-1b-redpajama-200b-dolly (accessed on 23rd August, 2023)

MODEL	DATA	VALIDATION	
		MICRO F1	MACRO F1
DistilBERT	Original	0.7913	0.5321
DistilBERT	Paraphrased	0.7942	0.4811
Flang-Roberta	Original	0.7971	0.5653
Flang-Roberta	Paraphrased	0.7971	0.5456
BERT-SEC	Original	0.813	0.5647
BERT-SEC	Paraphrased	0.7880	0.4900
Cross Encoder (BERT)	Original	0.7898	0.5383
Cross Encoder (BERT)	Paraphrased	0.7913	0.4956
Cross-Encoder (BERT-SEC)	Original	0.7695	0.476
Cross-Encoder (BERT-SEC)	Paraphrased	0.7681	0.4807
Cross Encoder (FinBERT Finetuned)	**Original**	**0.8275**	**0.5298**
Cross-Encoder (MLM-FinBERT)	Original	0.8000	0.5054
Cross-Encoder (MLM-FinBERT)	Paraphrased	0.7913	0.5482

Table 3: Results of Task 2, Sub-Task 2: Argument Relation Identification

LANGUAGE	MODEL	DATA	VALIDATION	
			MICRO F1	MACRO F1
English	BERT-base	Original	0.6453	0.6783
English	BERT-base	Paraphrased	0.6319	0.6568
English	FLANG-RoBERTa	Paraphrased	0.6392	0.6754
English	Cross Encoder (SBERT)	Original	0.6404	0.6796
English	Cross Encoder (SBERT)	Paraphrased	0.6500	0.6880
English	Cross Encoder (DistilROBERTA)	Original	0.7055	0.7472
English	Cross Encoder (DistilROBERTA)	Paraphrased	0.6920	0.7374
English	Cross Encoder (Flang-Roberta)	Original	0.6932	0.7342
English	Cross Encoder (Flang-Roberta)	Paraphrased	0.6858	0.7314
English	**Cross Encoder (BERT-SEC)**	**Original**	**0.6932**	**0.7342**
English	Cross Encoder (BERT-SEC)	Paraphrased	0.6800	0.7000
English	Cross Encoder (MLM on BERT-SEC)	Original	0.6846	0.7160
English	Cross Encoder (MLM on BERT-SEC)	Paraphrased	0.6871	0.7180
Chinese	SBERT-Chinese	Original	0.6321	0.6450
Chinese	Cross Encoder (SBERT-Chinese)	Original	0.6503	0.6432

Table 4: Result of Task 3: Identifying Argumentative Relation in Social Media Discussion

REFERENCES

[1] Alaa Alhamzeh, Romain Fonck, Erwan Versmée, Elöd Egyed-Zsigmond, Harald Kosch, and Lionel Brunie. 2022. It's Time to Reason: Annotating Argumentation Structures in Financial Earnings Calls: The FinArg Dataset. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 163–169. https://doi.org/10.18653/v1/2022.finnlp-1.22

[2] Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Vitor Silva, Leonardo Neves, and Francesco Barbieri. 2022. Twitter Topic Classification. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 3386–3400. https://aclanthology.org/2022.coling-1.299

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

[4] 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.*

[5] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. https://doi.org/10.48550/ARXIV.2210.11416

[6] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. *Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM.* https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[8] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. (2017). To appear.

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* (2016).

[10] Lefteris Loukas, Manos Fergadiotis, Ilias Chalkidis, Eirini Spyropoulou, Prodromos Malakasiotis, Ion Androutsopoulos, and Paliouras George. 2022. FiNER: Financial Numeric Entity Recognition for XBRL Tagging. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)* (Dublin, Republic of Ireland). Association for Computational Linguistics. https://arxiv.org/abs/2203.06482

Task	Model	Prompt Type	Validation			Confusion Matrix
			Micro F1	Macro F1	Accuracy	NA SP AT
2-2	Dolly 3B V2	Static (ZS)	0.5214	0.2925	0.5215	$\begin{bmatrix}15 & 70 & 0 \\ 46 & 408 & 6 \\ 33 & 235 & 2\end{bmatrix}$
		Static (FS)	0.4985	0.3248	0.4986	$\begin{bmatrix}51 & 121 & 28 \\ 121 & 290 & 71 \\ 2 & 3 & 3\end{bmatrix}$
		Dynamic - Algo. 1	0.6652	0.4114	0.6652	$\begin{bmatrix}54 & 145 & 1 \\ 72 & 404 & 6 \\ 1 & 6 & 1\end{bmatrix}$
		Dynamic - Algo. 2 (v1)	0.455	0.3346	0.4551	$\begin{bmatrix}120 & 72 & 8 \\ 256 & 192 & 34 \\ 4 & 2 & 2\end{bmatrix}$
		Dynamic - Algo. 2 (v2)	0.2362	0.1886	0.2362	$\begin{bmatrix}17 & 61 & 122 \\ 40 & 141 & 301 \\ 1 & 2 & 5\end{bmatrix}$
		Dynamic - Algo. 2 (v3)	0.6449	0.2939	0.6449	$\begin{bmatrix}13 & 186 & 1 \\ 42 & 432 & 8 \\ 0 & 8 & 0\end{bmatrix}$
3	Dolly 3B V2	Static (FS)	0.3042	0.2766	0.3043	$\begin{bmatrix}37 & 32 & 16 \\ 198 & 160 & 102 \\ 136 & 83 & 51\end{bmatrix}$
		Dynamic - Algo. 1	0.5632	0.3914	0.5123	$\begin{bmatrix}62 & 21 & 2 \\ 13 & 385 & 62 \\ 21 & 202 & 47\end{bmatrix}$
	MPT 1B	Static (FS)	0.5650	0.5558	0.5681	$\begin{bmatrix}52 & 28 & 5 \\ 3 & 357 & 100 \\ 0 & 216 & 54\end{bmatrix}$
		Dynamic - Algo. 1	0.6312	0.4918	0.6809	$\begin{bmatrix}73 & 12 & 0 \\ 9 & 392 & 59 \\ 2 & 52 & 216\end{bmatrix}$
	Flan T5 Small	Static (FS)	0.3374	0.2141	0.3374	$\begin{bmatrix}6 & 2 & 77 \\ 10 & 9 & 441 \\ 10 & 0 & 260\end{bmatrix}$
		Dynamic - Algo. 1	0.4614	0.3136	0.4614	$\begin{bmatrix}46 & 35 & 4 \\ 107 & 348 & 5 \\ 0 & 198 & 72\end{bmatrix}$

Table 5: Prompt Engineering Results. 2-2 refers to Task-2, Sub Task-2 and 3 refers to Task-3. NA, SP, AT refer to classes None, Support & Attack class respectively. ZS - Zero Shot, FS - Few Shot.

[11] Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.
[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162
[13] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084
[14] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv* abs/1910.01108 (2019).
[15] Raj Sanjay Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. When FLUE Meets FLANG: Benchmarks and Large Pretrained Language Model for Financial Domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

A APPENDICES

A.1 Prompts of Task 2-2

The below prompts has been made to Dolly V2 3B.

- **Zero Shot Prompt**: Determine the one word relationship between the following two sentences. Choose one relationship from: [none, support, attack]. Response in one word.
  Sentence 1: ‘query sentence 1‘
  Sentence 2: ‘query sentence 2‘
  Response:

  Here the pair of query sentence is taken from the validation set.

- **Few Shot Static Prompt**: Determine the one word relationship between the following two sentences. Choose one relationship from: [none, support, attack]. Response is a one word relationship.
  Sentence 1: ‘What I can say, the biggest dynamic going on again is that Amazon fulfilled unit growth of nearly 40%, which was last year and carrying into this year.‘
  Sentence 2: ‘We are matching that with just over 30% increase in square footage, and you’re right, that does include some shipping sort centers and things that are incremental and new functions for us, if you will.‘
  Response: none
  Sentence 1: ‘Japan as a geography for us is a high transactional market.‘
  Sentence 2: ‘The improvement in that in Q3 is obviously very high margin and also the bottom.‘
  Response: support
  Sentence 1: ‘I think there’s a tendency in this industry to call everything new the next computer platform.‘
  Sentence 2: ‘However, that said, I think AR can be huge.‘
  Relationship: attack
  Sentence 1: ‘query sentence 1‘.
  Sentence 2: ‘query sentence 2‘.
  Response:

  Here the pair of query sentence is taken from the validation set.

- **Few Shot Dynamic Prompt from Algorithm 1**: Determine the one word relationship between the following two sentences. Choose one relationship from: [none, support, attack]. Response is a one word relationship.
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘query sentence 1‘
  Sentence 2: ‘query sentence 2‘
  Response:

  Here the pair of sentences are taken from the train set having highest semantic similarity and the pair of query sentence is taken from the validation set.

- **Few Shot Dynamic Prompt from Algorithm 2**: Determine the one word relationship between the following two sentences. Choose one relationship from: [none, support, attack]. Response is a one word relationship.
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: none
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: none
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: support
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: support
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: attack
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: attack
  Sentence 1: ‘query sentence 1‘
  Sentence 2: ‘query sentence 2‘
  Response:

  Here the pair of sentences are taken from the train set having highest semantic similarity and the pair of query sentence is taken from the validation set. In different version of the prompt, the placement of classification categories differ.

A.2 Prompts of Task 3

The below prompts has been made to all the models.

- **Zero Shot Prompt**: Determine the one word relationship between the following two sentences. Choose one relationship from: [none, support, attack]. Response in one word.
  Sentence 1: ‘query sentence 1‘
  Sentence 2: ‘query sentence 2‘
  Response:

  Here the pair of query sentence is taken from the validation set.

- **Few Shot Dynamic Prompt from Algorithm 1**: Determine the one word relationship between the following two sentences. Choose one relationship from: [none, support, attack]. Response is a one word relationship.
  Sentence 1: Taishi Electric, January 01, 103 to March 31, 103, the comprehensive profit and loss table per share is 0.95 yuan 2014/05/12 14:49 Karishi information should be empty and daily tomorrow
  Sentence 2: 1. Public Information Observation Station 2. It is not intentional. IFRS3 forced listed cabinet companies to perform acquisitions must recognize the premium or discount of the equity of the purchase of the stake in the purchase of the price of the purchase of the equity.
  Response: support
  Sentence 1: FRS helped the Taiwan Steamor’s help last year to help the original profit of only 930 million, and the under IFRS became 1.26 billion stars. How much does the increase in maintenance costs affect?
  Sentence 2: Da sees you so seriously studying 8926 to pat your hands !! The younger brother has always focused on the management of the stock that has always focused on the management of the management.
  Sentence 1: ‘Chunghwa Telecom ADR rose 0.16 US dollars by 0.53 percent to 93.76 yuan‘
  Sentence 2: oday, China Electric Taiwan Darian Biography has risen Foreign capital also buy super But half of China Dian’s transaction volume was bought by foreign capital The increase is not even half of the big or far -reaching increase in Taiwan.
  Relationship: none
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘sentence 1‘
  Sentence 2: ‘sentence 2‘
  Response: ‘response‘
  Sentence 1: ‘query sentence 1‘
  Sentence 2: ‘query sentence 2‘
  Response:

  Here the pair of sentences are taken from the train set having highest semantic similarity and the pair of query sentence is taken from the validation set.