# BAI-Arg LLM at the FinLLM Challenge Task: Earn While You Argue - Financial Argument Identification

**Varad Srivastava**
Barclays
varad.srivastava@barclays.com

## Abstract

Previous studies have shown that analyst decisions that can influence investors to buy or sell in markets, are based on statements in Earnings Conference Calls (ECC). In this study, we present our LLMs (BAI-Arg Alpha and Beta) dedicated to the task of financial argument identification in sentences from ECC transcripts. Our experiments involved using in-context zero-shot and semantically similar few-shot learning, along with QLoRA-based fine-tuning methods. Our model BAI-Arg Alpha was able to out-perform all other proposed models, to achieve 1st rank on the leaderboard of FinLLM challenge (IJCAI'24). Furthermore, using our BAI-Arg Beta model, we were able to achieve micro-F1 and macro-F1 scores of 76.68% and 76.66% respectively, which are state-of-the-art, and out-perform all previously proposed models and approaches on the task. By being able to categorize arguments in ECC with a high degree of accuracy through our model, we hope to offer stakeholders enhanced clarity on financial sentiments, which can enable them to make more informed decisions in the economic markets.

## 1 Introduction

Predicting movements in market is a challenging problem, even with the recent growth of data and advance algorithms in the field of finance. This is because several factors and environments can influence its movements, which makes it difficult to get a very accurate estimate of stock prices in the future. According to the "efficient market hypothesis" (Fama, 1970), since the market is efficient (everything is fairly priced according to their value), it is not possible to outperform the overall market all the time even by using technical analysis to predict trends and select market timings. However, it is a widely accepted view that most of the investment decisions are influenced by cognitive bias and experience of a person, as humans are not known to be rational decision-makers (Tversky and Kahneman, 1974). Past research works have extensively studied the impact of sentiments and events from online news, and social media platforms like tweets, as well as the semantics of language and recommendations used by forecasters and professional analysts, which can influence investors decision to buy or sell in markets. Findings by Keith and Stent (2019) in particular have shown that statements on Earnings Conference Calls (ECC) are reflective of analysts' decisions.

ECC are organized during every fiscal quarter and consist of the following three parts: a safe harbor statement, a presentation and question answering (Q&A) session. During presentations, executives present their statements about the performance of the company in last quarter as well as expectations about the future quarters. Professional analysts posit their questions and demand clarifications from the company's representatives during the Q&A session. The company executives present their arguments as answers in order to justify their opinions and convince people to believe in them. Previous studies have shown that discussions during the Q&A session have the most influence on the shifts in market (Matsumoto et al., 2011; Price et al., 2012).

While most of the previous works have encapsulated the use of semantic or syntactic analyses, argument mining can be used to extract a deeper interpretation of the language used to make statements in these sessions which in turn can help understand what people expect of the markets. This information can be used to drive investment decisions.

ECC transcripts are more favourable to extract arguments for two reasons. One, social media platforms are often restricted by number of characters. Two, people tend to post their opinions and views rather than structured premises or claims. For example, most of the tweets only have claims, which

|  | Train | Test | Whole |
|---|---|---|---|
| Premise | 4,062 | 508 | 4,570 |
| Claim | 3,691 | 461 | 4,152 |
| Total | 7,753 | 969 | 8,722 |

Table 1: Data statistics

assert a conclusion or viewpoint without providing the required reasoning or evidence.

Additionally, even though language models have been extensively used for this task, use of LLMs on financial tasks such as these is still under-explored and under-utilized. Therefore in this paper, we experiment with various LLMs, utilizing methods like in-context learning and fine-tuning to investigate the arguments stated in the answers of company executives to questions of analysts. We finally propose LLM models (BAI-Arg), to leverage their state-of-the-art capabilities to classify these statements on the basis of argumentative function they represent - premise, or claim.

## 2 Dataset

The FinArg dataset (Alhamzeh et al., 2022) was used for the task of argument unit classification and was made available as part of shared task of the Fin-LLM challenge. See Appendix:A.1 for examples from the dataset. Here, the task is to use the capabilities of LLMs to interpret the argument units in statements from ECC transcripts by classifying them into "premise" or "claim". 7,753 statement texts and their gold labels were provided as training data, and the models were evaluated on 969 texts of test data. See Table 1 for more details.

## 3 Related Work

On the FinArg-1 challenge task of argument unit identification in NTCIR-17 (Chen et al., 2023), various language models were examined with either prompting or fine-tuning. The best model was submitted by TMUNLP (Lin et al., 2023) which was based on assembling outputs of ELECTRA and Roberta models using a voting mechanism, and achieved 76.55% macro-F1 score. The second ranked model by IDEA (Tang and Li, 2023) combined BERT hidden state embeddings with a Convolutional Neural Network (CNN), while the third ranked model by TUA1 (Chen et al., 2023) used the T5 model with prompt-based learning and instruction tuning. Other submitted approaches included leveraging GPT-3.5 Turbo for in-context learning as well as generating more similar data to augment the dataset.

Sy et al. (2023) experimented with a BERT-based ensemble learning approach using a majority-voting mechanism to achieve a macro-F1 score of 76.62% on the task. More recently, Xie et al. (2024) in their work on the FinBen benchmark, evaluated several state-of-the-art LLM models like GPT-4, Gemini, LLaMA-70B, FinMA-7B, Falcon-7B, ChatGLM3-6B, FinGPT-7b-lora. InternLM-7B, Mixtral-7B, and CFGPTsft-7B-Full, on the Financial Argument Classification (FinArg-ACC) task, with GPT-4 out-performing all others with a macro-F1 score of 60.0%.

## 4 Methodology

This section provides descriptions of the various approaches we experimented for the challenge.

### 4.1 In-Context Learning

For in-context learning, we use LLMs like:

- `Llama-3`: We used Llama-3 8B parameter model (AI@Meta, 2024), which has context length of 8,192.
- `Mistral`: We used Mistral-7B model version 0.2 (Jiang et al., 2023; MistralAI) which has a context window of 32,768.
- `Gemma`: We used Gemma 7B model (Google), which has context length of 8,192.
- `GPT`: We used GPT-3.5 Turbo (OpenAI), which has a context window of 16,385 tokens.

These pre-trained chat models have been further fine-tuned to follow instructions with Reinforcement Learning from Human Preferences (RLHF). Therefore, we use the instruction-tuned versions of each of the models.

### 4.2 Prompt Engineering

Articulate prompt engineering is crucial in steering behaviour and response of the LLMs, by providing them the appropriate instructions and context for a task. Our prompt template, which went through various iterations of experiments, is provided in Appendix:A.3. The prompt starts with an instruction which encompasses the context of the task including a knowledge base detailing the classification criteria and short description of each of the classes. The test statement is then provided as an input by the user.

### 4.2.1 Zero Shot and Random Few Shot Learning

For our initial approach, we experimented with zero-shot learning and in-context learning with 1, 5 and 10 examples per class, chosen randomly from the training set.

### 4.2.2 Semantically Similar Few Shot Learning

In this approach, we select those examples for in-context learning from the training set, which are semantically similar to the test statement at inference. This is achieved by first training a sentence-transformer (MPNet) on the training set, which learns to encode the statements in the embedding space, based on whether their class is similar or dissimilar. In this work, we select one of its variations - 'all-mpnet-base-v2', which also ranks among the top in the HuggingFace sentence transformers leaderboard. Therefore based on this idea, for each test sentence to be classified, we use the all-mpnet-base-v2 vector embeddings and the cosine similarity metric (for distance calculation) to retrieve the 5,10 and 20 most similar examples at inference time, while performing in-context learning. For more details on the MPNet model, and its hyper-parameter tuning, refer to Appendix:A.2.

### 4.3 Fine-Tuning of Instruction Tuned LLMs

Based on the performance of models during in-context learning, we select Llama-3 8B model for fine-tuning to enhance model performance further. Each sample from the training set was converted into a prompt which included the test statement as a user input and the true label as the reply expected from the chat assistant. We experimented with two prompts here, with differences in only the structure and language of instruction (see Appendix:A.3).

QLoRA (Quantized Low-Rank Adaptation) (Dettmers et al., 2023) was used to efficiently fine-tune the model. We first quantized the pre-trained model to 4-bit and then added a set of learnable low-rank adapter weight matrices with rank 64, that are tuned using backpropagation for upto 3 epochs. This was able to significantly reduce trainable parameters to 167M, hence reducing GPU memory requirements. The details of hyper-parameters are shown in Table 2. For the metrics reported in the Section 5, the model with "Prompt-1" was trained for three epochs, while the model with "Prompt-2" for two epochs. Hereafter, we refer to earlier model as "BAI-Arg Alpha", and the latter as "BAI-Arg Beta".

| Hyperparameters | Value |
|---|---|
| Gradient Accumulation Steps | 4 |
| Learning Rate | 2e-4 |
| Epoch | 2 |
| LoRA-Rank | 64 |
| LoRA-Alpha | 128 |
| LoRA-Dropout | 0 |
| Optimizer | Adam |

Table 2: QLoRA Hyper-parameter Details

## 5 Results

### 5.1 Performance on FinLLM Challenge Task

We report the performance of our modelling approaches through the metrics: micro-F1 (µ-F1) and macro-F1 (m-F1), as shown in Table 3. We observed that although all models perform poorly on zero-shot and random few-shot in-context learning, Llama-3 here still edges out Mistral and Gemma models. Additionally, when Retrieval-Augmented Generation (RAG) is used by augmenting in-context learning with semantically similar examples, there is a significant increase in performance for all models. Notably, here as well, Llama-3 is able to outperform other models, barring the 10-shot similar examples setting, where GPT-3.5 outperforms even Llama-3, by 0.33 percentage points (pp) in macro-F1. Nevertheless, since this does not hold on other settings like the 20-shot, we selected Llama-3 model for fine-tuning to investigate if performance can be further enhanced. Indeed, fine-tuning was able to enhance performance significantly, with macro-F1 increasing by upto 4.39 pp on the BAI-Arg Beta[1] model. Additionally, see Appendix:A.4 and A.5 for details on ablation studies conducted on few-shot learning and fine-tuning approaches respectively, and A.6 for error analysis.

Our model was ranked 1st on the FinLLM challenge leaderboard [2] for this task, when compared against the performance of other submitted models, as shown in the first section of the Table 4.

### 5.2 Performance Comparison with Existing Models

We also compared the performance of our model against the performance of LLMs in previous works such as that of Xie et al. (2024), and performance of the top-3 models proposed during

[1] https://huggingface.co/varadsrivastava/BAI_Arg_Beta

[2] https://huggingface.co/spaces/TheFinAI/IJCAI-2024-FinLLM-Learderboard

Table 3: Classification results for all models on the test data, with N-Shot indicating the number of samples used during learning. FT-n indicates fine-tuned using Prompt 'n'

| **Methods** | **Setting** | $\mu - \mathrm{F}_1$ | $\mathrm{m}{-}\mathrm{F}_1$ |
|---|---|---|---|
| Gemma | 0-shot | 55.41 | 50.93 |
| Lllama-3 | 0-shot | **59.44** | **56.74** |
| Mistral | 0-shot | 53.56 | 47.26 |
| Gemma (random) | 1-shot | 50.57 | 40.04 |
| Llama-3 (random) | 1-shot | **58.93** | **54.21** |
| Mistral (random) | 1-shot | 58.10 | 53.76 |
| Gemma (random) | 5-shot | 53.97 | 49.13 |
| Llama-3 (random) | 5-shot | 61.61 | 60.16 |
| Mistral (random) | 5-shot | 53.56 | 39.35 |
| Gemma (similar) | 5-shot | 64.09 | 62.22 |
| Llama-3 (similar) | 5-shot | **71.00** | **70.87** |
| Mistral (similar) | 5-shot | 67.91 | 66.65 |
| GPT-3.5 (similar) | 5-shot | 69.04 | 68.83 |
| Gemma (random) | 10-shot | 52.94 | 45.83 |
| Llama-3 (random) | 10-shot | 61.09 | 57.55 |
| Mistral (random) | 10-shot | 55.73 | 47.88 |
| Gemma (similar) | 10-shot | 66.98 | 66.20 |
| Llama-3 (similar) | 10-shot | 70.69 | 70.65 |
| Mistral (similar) | 10-shot | 70.90 | 70.13 |
| GPT-3.5 (similar) | 10-shot | **71.10** | **70.98** |
| Gemma (similar) | 20-shot | 69.35 | 68.58 |
| Llama-3 (similar) | 20-shot | **72.34** | **72.27** |
| Mistral (similar) | 20-shot | 71.93 | 71.36 |
| GPT-3.5 (similar) | 20-shot | 70.69 | 70.51 |
| BAI-Arg Alpha | FT-1 | 76.26 | 76.12 |
| **BAI-Arg Beta** | FT-2 | **76.68** | **76.66** |

NTCIR-17 (2023) (Chen et al., 2023). These comparison results are shown in Table 4.

We observe that our model BAI-Arg Beta outperforms all others in it's ability to identify the argument unit, achieving micro-F1 and macro-F1 scores of 76.68% and 76.66%.

### 5.3 Model Cheating Detection

Due to concerns around data leakage in LLMs, a perplexity-based metric - Data Leakage Test (DLT), has been proposed by the FinLLM challenge organizers building on existing research (Wei et al., 2023). For details about the metric, refer to Appendix:A.7.

The DLT values are shown in Table 5. We observed that both of our models have a high enough DLT value, and even though there's a drop in the Beta version, the DLT metric value is still significantly higher than the reference baseline from the leaderboard. This indicates that our models have a very low likelihood of cheating from data leakage.

Table 4: Comparison of our model's performance against other proposed models

| **Models** | $\mu - \mathrm{F}_1$ | $\mathrm{m}{-}\mathrm{F}_1$ |
|---|---|---|
| Albatross [2] | 75.75 | - |
| L3iTC [2] | 75.44 | - |
| Wealth Guide [2] | 75.13 | - |
| GPT-4 (Xie et al., 2024) | 60.0 | - |
| Gemini (Xie et al., 2024) | 31.0 | - |
| LLaMA2-70B (Xie et al.) | 58.0 | - |
| FinMA-7B (Xie et al., 2024) | 27.0 | - |
| Falcon-7B (Xie et al., 2024) | 23.0 | - |
| TMUNLP-1 (Lin et al., 2023) | 76.57 | 76.55 |
| IDEA-1 (Tang and Li, 2023) | 76.47 | 76.46 |
| TUA1-1 (Chen et al., 2023) | 76.37 | 76.36 |
| Sy et al. (2023) | - | 76.62 |
| BAI-Arg Alpha (Ours) | 76.26 | 76.12 |
| **BAI-Arg Beta (Ours)** | **76.68** | **76.66** |

Table 5: Data Leakage Test Results

| **Models** | **DLT** |
|---|---|
| L3iTC [2] | 2.2565 |
| **BAI-Arg Alpha** | **28.8399** |
| **BAI-Arg Beta** | **14.6049** |

## 6 Conclusion

In the rapidly evolving field of research using LLMs in finance domain, this shared task of FinLLM presented a unique opportunity to leverage LLM-based approaches for financial argument identification in quarterly Earnings Conference Calls (ECC) as premise or claim. In this paper, we presented our model, BAI-Arg LLM, based on well-articulated instruction prompts and fine-tuned Llama-3 8B model, which ranked first on the task in the IJCAI'24 FinLLM challenge leaderboard. It is able to out-perform all the other model submissions in the challenge, as well as the models proposed in previous literature. Therefore, by being able to categorize arguments in ECC with a high degree of accuracy through our model, we are able to offer stakeholders enhanced clarity on financial sentiments, which can enable them to make more informed decisions in the economic markets.

## References

AI@Meta. 2024. Llama 3 model card.

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)*, pages 163–169, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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. NII Institutional Repository.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Eugene F. Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, 25(2):383–417.

Google. Gemma 7b instruct model card on huggingface. `https://huggingface.co/google/gemma-7b-it`. [Accessed 21-06-2024].

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Katherine Keith and Amanda Stent. 2019. Modeling financial analysts' decision making via the pragmatics and semantics of earnings calls. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 493–503, Florence, Italy. Association for Computational Linguistics.

Heng-Yu Lin, Eugene Sy, Tzu-Cheng Peng, Shih-Hsuan Huang, and Yung-Chun Chang. 2023. Tmunlp at the ntcir-17 finarg-1 task. NII Institutional Repository.

Dawn Matsumoto, Maarten Pronk, and Erik Roelofsen. 2011. What Makes Conference Calls Useful? The Information Content of Managers' Presentations and Analysts' Discussion Sessions. *The Accounting Review*, 86(4):1383–1414.

MistralAI. Mistral 7b instruct v0.2 model card on huggingface. `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`. [Accessed 21-06-2024].

OpenAI. `https://platform.openai.com/docs/models/gpt-3-5-turbo`. [Accessed 21-06-2024].

S. McKay Price, James S. Doran, David R. Peterson, and Barbara A. Bliss. 2012. Earnings conference calls and stock returns: The incremental informativeness of textual tone. *Journal of Banking and Finance*, 36(4):992–1011.

Eugene Sy, Tzu Cheng Peng, Shih Hsuan Huang, Hen You Lin, and Yung Chun Chang. 2023. Fine-grained argument understanding with bert ensemble techniques: A deep dive into financial sentiment analysis. ROCLING 2023 - Proceedings of the 35th Conference on Computational Linguistics and Speech Processing, pages 242–249. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Shaopeng Tang and Lin Li. 2023. Idea at the ntcir-17 finarg-1 task: Argument-based sentiment analysis. NII Institutional Repository.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. 2023. Skywork: A more open bilingual foundation model. *Preprint*, arXiv:2310.19341.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, Yijing Xu, Haoqiang Kang, Ziyan Kuang, Chenhan Yuan, Kailai Yang, Zheheng Luo, Tianlin Zhang, Zhiwei Liu, Guojun Xiong, Zhiyang Deng, Yuechen Jiang, Zhiyuan Yao, Haohang Li, Yangyang Yu, Gang Hu, Jiajia Huang, Xiao-Yang Liu, Alejandro Lopez-Lira, Benyou Wang, Yanzhao Lai, Hao Wang, Min Peng, Sophia Ananiadou, and Jimin Huang. 2024. The finben: An holistic financial benchmark for large language models. *Preprint*, arXiv:2402.12659.

# A Appendix

## A.1 Dataset Examples

Humans often use argumentations to express themselves during a communication, and to think or deliberate about a situation or choice, which forms the core part of human decision making. A simple form of argument has two parts: a 'premise' (which provides some evidence or reason), and it supports a 'claim' (which is a conclusion).

An example of each from the provided dataset is provided as follows.

Premise:

```
"But another area that's growing incredibly
    quickly is private messaging, right, where
    between Messenger and WhatsApp, I think we'
    re around 60 billion messages a day, which I
     think is something like three times more
    than the peak of global SMS traffic."
```

Claim:

```
"And what we're doing on Messenger and on
    WhatsApp are really making sure that
    businesses can connect with people, and then
     in the early stages of testing messaging."
```

Recognizing arguments from a text involves two sub-tasks: firstly, identifying and separating the argumentative units from the non-argumentative text units; secondly, classifying argument units into premises and claims. However, it is possible that a sentence is not a separate argument unit, rather encompasses several argument units. Because of this, argument units in the dataset were originally annotated at a minimum of clause-level and a maximum of sentence-level. Various clauses within the same sentence were considered different argument components if there was an inference relation between them (for e.g., appeared in forms like "claim because of premise", "Since premise then claim.", "In view of the fact premise that it follows that claim"), rather than a conjunction (for e.g. "and", "or"), or conditional (for e.g. "if, then"). However, this resulted in a few counter-intuitive clauses in the dataset, which might not make much sense in themselves, unless seen together with their original sub-clauses. Some examples of these instances are:

Premises:

```
"The second thing is video."
"because of the FX situation, right."
```

Claims:

```
"So, first on head count."
"One is just the format."
```

Therefore, these noisy examples make the task more challenging than it seems. Assuming the distribution of such instances in the test set to be similar to the training set, in our prompts - we decided to rely on instructions based on the function of the argument unit (premise or claim), rather than its structure.

### A.2 MPNet Model and Hyper-parameter Tuning

MPNet is a transformer-based model which uses permuted language modelling to learn dependency among predicted tokens, as well as uses auxiliary position information as input. It is pre-trained on a text corpora of over 160 GB and fine-tuned on downstream tasks like GLUE, and SQuAD. Hyper-parameter tuning for MPNet-v2 was performed using Optuna framework. Over 10 trials, validation micro-F1 was maximized by having search spaces over body's learning rate (1e-5, 5e-3), as well as the batch size [4,8,16,32].

### A.3 Prompt Templates

The prompt template for the BAI-Arg Beta model is shown below.

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
You are an expert assistant which can analyze
    sentences from earnings conference call and
    identify their argumentative function. Your
    task is to classify the sentence after
    <<<>>> into one of the following predefined
    classes:
premise
claim
A sentence is a premise if it offers an evidence
     or reasoning. A sentence is a claim if it
    asserts a conclusion or viewpoint. You will
    only respond with the name of the class. In
    case you reply with something else, you will
     be penalized. Do NOT provide explanations
    or notes.<|eot_id|><|start_header_id|>user<|
    end_header_id|>
<<<
Sentence: {Text}
>>>
<|eot_id|><|start_header_id|>assistant<|
    end_header_id|>
Class: {Class}<|eot_id|>
```

A similar template was used to generate final results (shown in Table 3) for all other models (Mistral, Gemma and GPT) as well. The prompt for each of the specific models only differed in the special tokens they use to identify the instructions, user input or model's reply. For e.g. in Gemma, we use "*<start_of_turn>userinstruction<end_of_turn>*" to specify the instruction and context, and "*<start_of_turn>model*" to indicate that we expect a reply from the model.

For the initial challenge submissions, we had worked on an earlier version of prompt, which we refer to as "Prompt-1". This was used in the model, BAI-Arg Alpha. However, during our later experimentations, we came up with "Prompt-2" (which was used in the BAI-Arg Beta model). We were able to achieve significantly better performance on in-context learning with this prompt. The performance of the Llama-3 model on the 'Prompt-1' is shown in Table 6, for reference.

The prompt template for the BAI-Arg Alpha[3] model is shown below.

```
<|begin_of_text|>
<|start_header_id|>system<|end_header_id|>
```

[3] https://huggingface.co/varadsrivastava/BAI_Arg_Alpha

Table 6: Classification results comparison for Llama-3 on the test data using Prompts 1 and 2, with N-Shot indicating the number of samples used during in-context learning.

| **Prompt** | **Setting** | $\mu - \mathrm{F}_1$ | $\mathrm{m}{-}\mathrm{F}_1$ |
|---|---|---|---|
| Prompt-1 | 0-shot | 54.80 | 51.44 |
| Prompt-2 | 0-shot | 59.44 | 56.74 |
| Prompt-1 (random) | 1-shot | 57.48 | 55.91 |
| Prompt-2 (random) | 1-shot | 58.93 | 54.21 |
| Prompt-1 (random) | 5-shot | 59.65 | 58.66 |
| Prompt-2 (random) | 5-shot | 61.61 | 60.16 |
| Prompt-1 (similar) | 5-shot | 61.61 | 60.68 |
| Prompt-2 (similar) | 5-shot | 71.00 | 70.87 |
| Prompt-1 (random) | 10-shot | 60.06 | 57.21 |
| Prompt-2 (random) | 10-shot | 61.09 | 57.55 |
| Prompt-1 (similar) | 10-shot | 64.81 | 64.16 |
| Prompt-2 (similar) | 10-shot | 70.69 | 70.65 |

```
You are an expert assistant, helping to analyze
     sentences from earnings conference calls and
      identify their argumentative function.
     Given a sentence which will be provided to
     you by the user from a earnings conference
     call, decide whether it is a premise or
     claim, described respectively as follows:
premise: A sentence which offers evidence or
     reasoning.
claim: A sentence which asserts a conclusion or
     viewpoint.
Reply with only one word (premise or claim).<|
     eot_id|>|start_header_id|>user<|
     end_header_id|>
Sentence: {Text}<|eot_id|>|start_header_id|>
     assistant<|end_header_id|>
Class: {Class}<|eot_id|>
```

### A.4 Few Shot Learning: Ablation Study

Since we observed significantly better results when using semantically similar few-shot learning as compared to random few-shot learning, we investigated if the models are doing better because of the inherent biasness in examples that were retrieved (based on semantic similarity). To test this, we analysed how likely it is for the majority of labels of the few examples to match the target class.

We observed that the majority of the classes of the most semantically similar examples matched the target class on upto 76.68% of the test instances at inference. See Table 7 for more details. At first look, this does hint that the similar examples might be biasing the model into doing better. However, since the model performance of all models (except GPT-3.5) increases considerably as the similar examples are increased from 5 to 20, with the number of biased examples falling down. Hence, it could be possible that the higher number of examples are also improving the argument understanding of the model, and the model might not just be resorting to the biasness of the examples for its good performance.

Table 7: Analysis of biasness in Few-shot learning approach: The table shows how likely it is for the majority of the labels of the few examples to match the target class of test sentence.

| **Setting** | **Instances of biased examples** | **% of Test** |
|---|---|---|
| 5-shot (similar) | 743 | 76.68% |
| 10-shot (similar) | 713 | 73.58% |
| 20-shot (similar) | 729 | 75.23% |

To investigate how significant of a role the biasness of the semantically similar examples are playing in few-shot learning, we perform an ablation study, wherein we investigated the performance of models by 'de-biasing' the example classes by sampling top-k examples from each class. For this, we sampled examples from top 500 semantically similar sentences (using the same methodology as described in Section 4.2.2), to retrieve top-5 and top-10 examples from each class, for 10-shot and 20-shot learning, respectively.

Table 8 shows the scores of the models when this equitable distribution of examples by class were retrieved for each test sentence at inference.

We observed that the performance of models drop significantly when the example classes are 'de-biased', by sampling top-k examples from each class. Therefore, this indicates that the biasness of the examples had a major role to play in the significant gains in performance of the models. Since, retrieval of such semantically-similar examples could be difficult in a noisy, real world use-case of this approach which could make the model less stable. Therefore, this provides support to the fine-tuning approach we used subsequently and the robustness of our final proposed model, BAI-Arg Beta.

### A.5 Fine-tuning: Ablation Study

We performed an ablation study with the fine-tuning approach as well. We investigated two questions - One, whether training the BAI-Arg model with few examples improves the performance or the few shots are only helpful before fine-tuning; Two, if few examples do help in fine-tuning, whether there is a difference if the model is trained on ran-

Table 8: Classification results for all models using similar and de-biased examples for in-context learning, with N-Shot indicating the number of samples used during learning.

| **Methods** | **Setting** | $\mu - \mathrm{F}_1$ | $\mathrm{m}{-}\mathrm{F}_1$ |
|---|---|---|---|
| Gemma (similar) | 10-shot | 66.98 | 66.20 |
| Llama-3 (similar) | 10-shot | 70.69 | 70.65 |
| Mistral (similar) | 10-shot | 70.90 | 70.13 |
| GPT-3.5 (similar) | 10-shot | **71.10** | **70.98** |
| Gemma (debiased) | 10-shot | 62.33 | 60.29 |
| Llama-3 (debiased) | 10-shot | 65.63 | 64.94 |
| Mistral (debiased) | 10-shot | 59.86 | 53.34 |
| GPT-3.5 (debiased) | 10-shot | 67.39 | 67.39 |
| Gemma (similar) | 20-shot | 69.35 | 68.58 |
| Llama-3 (similar) | 20-shot | **72.34** | **72.27** |
| Mistral (similar) | 20-shot | 71.93 | 71.36 |
| GPT-3.5 (similar) | 20-shot | 70.69 | 70.51 |
| Gemma (debiased) | 20-shot | 61.40 | 58.53 |
| Llama-3 (debiased) | 20-shot | 65.94 | 63.78 |
| Mistral (debiased) | 20-shot | 60.99 | 55.91 |
| GPT-3.5 (debiased) | 20-shot | 67.70 | 67.68 |

Table 9: Classification results for models trained on few-shot (five) examples

| **Methods** | $\mu - \mathrm{F}_1$ | $\mathrm{m}{-}\mathrm{F}_1$ |
|---|---|---|
| In-context (random ex) | 61.61 | 60.16 |
| In-context (similar ex) | 71.00 | 70.87 |
| Fine-tuned (random ex) | 72.34 | 71.30 |
| Fine-tuned (similar ex) | 74.51 | 74.38 |

dom examples versus similar examples.

In order to investigate these, we performed an "active few-shot fine-tuning" where we included random and semantically similar (five) examples in the training of the Llama-3 model. The QLoRA hyper-parameters used were the same as shown in Table 2, and the model was trained for two epochs. The results obtained are shown in Table 9.

We observed that the "active few-shot fine-tuning" with randomly selected examples significantly improves the performance by upto 9 pp on micro-F1. as compared to the in-context learning with random examples. Additionally, semantically similar examples improve the performance even further, although the gains over in-context learning are not as significant, here. Interestingly, fine-tuning without few-shot examples still outperforms fine-tuning with examples, indicating that the examples might only help the model improve its understanding of arguments upto a limit.

Therefore, few examples do improve the performance and are helpful not just before, but during training as well. Additionally, here too, similar examples out-perform randomly selected ones in model performance.

### A.6 Error Analysis

We performed a qualitative error analysis of our BAI-Arg Beta model to understand the model's behaviour by observing what it gets wrong. This model made 122 errors on the 'premises' and 104 errors on the 'claims'.

Although its difficult to figure out the model's exact heuristics for arriving at the decision, here are some observations we made:

- *Errors in premises being identified as claims:* These could be a result of the evidence or reasoning being expressed as view-points or lacking any key metrics. For e.g.:

```
"In terms of overall ad tech world, I think
      a lot is happening and there's a lot
     that's going to evolve in the whole
     ecosystem.
And if you go beyond that, I feel good
     about our gaming business sequentially
     ."
"The iPhone SE, we are thrilled with the
     response that we've seen on it."
"I feel confident in our ability to produce
      gross margin improvement across all
     those services."
"So we have great relationships with third
     party carriers."
"So we've said often that we think that
     virtual reality and augmented reality
     could be the next big computing
     platform."
```

  Also, we observed certain errors where premises were rather expressed as past actions or planned ones in future, which might be the reason, the model classified them as claims. For e.g.:

```
"And we're continuing to invest across the
     board in terms of our core R&D and
     innovation efforts in terms of
     headcount growth there."
"And so we don't enter into those with no
     experience, although we will enter into
      them humbly."
```

- *Errors in claims being identified as premises:* These errors could have been caused due to addition of specific metrics, which the model might be mistaking for being part of evidence or reasoning. For e.g.:

```
"In the last 18 months, we've doubled the
    number of paid Prime member, which we'
    re very excited about."
"So all of those trailing 12-month metrics
    actually stayed the same or slightly
    declined in Q1."
"But in general, inclusive of LinkedIn, I'm
     still around 100 bps."
```

Also, we observed certain errors where the missing context of the sentence might have confused the model in mistaking the sentences for facts, rather than a conclusion. For e.g.:

```
"We did see ARPU growth this quarter."
```

A claim like above might have been preceded by a context which is likely to include premise clauses, probably something like "although users decreased by xx%..." which might then make more sense for the below example to be percieved as a claim (with an inference relation like "in view of the fact premise").

Some other such examples are:

```
"And people had, again the ability to see
    the benefit that Prime membership save
    incremental dollars, because of it at
    Whole Foods."
"And the other one is the on-premises
    server number which is very good in
    terms of hybrid demand this quarter
    also with high margin."
"In the United States, which is usually the
     most advanced market, 35% of small
    businesses have no web presence at all
    ."
```

Although, improving performance on such examples with somewhat overlapping argumentation intents is difficult without providing the context for each argument clause; for future work, we will try to leverage Chain-of-Thought reasoning in our prompts to mitigate them.

### A.7 Data Leakage Test

The DLT metric calculates the difference in perplexity of the LLMs between the training and test data to determine its data generation tendencies. A larger difference implies that the LLM is less likely to have seen the test set during training compared to the training set and suggests a lower likelihood of the model cheating, and vice versa. The formula for the DLT metric is as follows:

$$DLT = PPL\left(\mathrm{D}_{\text{test}}\right) - PPL\left(D_{\text{train}}\right)$$

$$PPL\left(D_{\text{train}}\right) = \frac{1}{\left|D_{\text{train}}\right|} \sum_{x \in D_{\text{train}}} P(x)^{-\frac{1}{N}}$$

$$= \frac{1}{\left|D_{\text{train}}\right|} \sum_{x \in D_{\text{train}}} P\left(w_1 w_2 \cdots x_N\right)^{-\frac{1}{N}}$$

$$= \frac{1}{\left|D_{\text{train}}\right|} \sum_{x \in D_{\text{train}}} 2^{-\frac{1}{N} \log P(w_i w_2 \cdots\cdots x_N)}$$

$$= \frac{1}{\left|D_{\text{train}}\right|} \sum_{x \in D_{\text{train}}} 2^{\mathrm{Cross-Entropy}(x)}$$

DLT values have been calculated for one other submitted model as well, to establish a reference baseline of Model Cheating, and minimize the impact of generalization on the metric.