LomonosovMSU at SemEval-2024 Task 4: Comparing LLMs and embedder models to identifying propaganda techniques in the content of memes in English for subtasks №1, №2a, and №2b

Gleb Skiba[1,2] **Mikhail Pukemo**[2] **Dmitry Melikhov**[2] **Konstantin Vorontsov**[1,2]
Institute of AI, Lomonosov Moscow State University[1],
The faculty of Computational Mathematics and Cybernetics,
Lomonosov Moscow State University[2]
gleb-skiba@mail.ru, iMan.assistance@gmail.com,
melikhov.dmitry.a@gmail.com, Voron@mlsa-iai.ru

Abstract

This paper presents the solution of the LomonosovMSU team for the SemEval-2024 Task 4 "Multilingual Detection of Persuasion Techniques in Memes" competition for the English language task. During the task solving process, generative and BERT-like (training classifiers on top of embedder models) approaches were tested for subtask №1, as well as an BERT-like approach on top of multimodal embedder models for subtasks №2a/№2b. The models were trained using datasets provided by the competition organizers, enriched with filtered datasets from previous SemEval competitions. The following results were achieved: 18th place for subtask №1, 9th place for subtask №2a, and 11th place for subtask №2b. The code for the solutions is available at github[1].

1 Introduction

In the modern world, memes are one of the most popular forms of delivering information to social media users. Unfortunately, memes created using a variety of rhetorical and psychological techniques are also used to conduct disinformation campaigns.

The overall goal of the SemEval-2024 Task 4 competition is to build models to detect rhetorical and psychological propaganda techniques in memes. The competition itself consists of three subtasks:

1. Build a model to detect rhetorical and psychological techniques only in the textual content of the meme. This is a hierarchical multilabel classification problem.

2a. Build a model to detect rhetorical and psychological techniques in both textual and visual contexts of the meme (multimodal task). This is a hierarchical multilabel classification problem.

2b. Build a model to identify the presence of rhetorical and psychological techniques in both textual and visual contexts of the meme in general. This is a binary classification problem.

In this work, experiments were conducted with generative and BERT-like models to solve subtask №1 and classifiers on top of multimodal models to solve subtasks №2a/№2b. BERT-like approaches refer to the creation classifiers over embedder models and will be used further in this article. All tasks were solved for datasets in English. The following results were achieved: 18th place for subtask №1, 9th place for subtask №2a, and 11th place for subtask №2b. The code for the solutions is available in the repository at GitHub[2].

2 Related Works

Supervised fine-tuning (SFT) (Ouyang et al., 2022) is one of the most popular methods for fine-tuning LLMs to solve various tasks. In this work, we conducted fine-tuning of LLMs, such as LLAMA[3] and Mistral (Jiang et al., 2023), for detecting propaganda techniques. Unfortunately, SFT is very resource-intensive, and conducting frequent experiments with fine-tuning LLMs is time-consuming and expensive. To address this problem, a less resource-intensive approach, LoRA (Hu et al., 2021), was used, but unfortunately, the results of this approach were not satisfactory.

In parallel with SFT and LoRA approaches, experiments were conducted on fine-tuning simple classifier layers on top of embedding models: debert (He et al., 2021), CLIP (Radford et al., 2021), BLIP (Li et al., 2022). The results of these experiments yielded comparable results to experiments with LLMs.

[1] https://github.com/pansershrek/Semeval2024_LomonosovMSU

[2] https://github.com/pansershrek/Semeval2024_LomonosovMSU

[3] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1544–1548
June 20-21, 2024 ©2024 Association for Computational Linguistics

3 Tasks solutions

3.1 Subtask №1

This subtask represents a hierarchical classification task. Two approaches were used to solve this task:

1. Generative approach: This approach involves training a generative model to generate explicit responses to questions in JSON format.

2. BERT-like approach: This approach involves training a simple fully connected network on top of a frozen pre-trained embedding model to solve the hierarchical classification task.

3.1.1 Generative approach

Idea. The main idea of this approach is to train a generative model, both in SFT mode (Ouyang et al., 2022) and trained using the LoRA technique (Piskorski et al., 2023), to answer the question: "Does the provided text contain any propaganda techniques, and if so, which ones?"

Dataset. The following combinations of data were used for training the models:

1. For selecting the best candidate model for the final solution, the training dataset provided by the authors was used, along with the dataset from the previous semeval competition [4]. This dataset was filtered, and only samples containing propaganda techniques matching those from the current competition were taken.

2. For training the model for the final solution, the same dataset described earlier was used, along with all samples from the gold dataset added to it.

Data Format. The following prompt phrase was used: "Your goal is to identify rhetorical and psychological techniques in the given text." The model was required to output JSON with propaganda techniques or an empty JSON. The model received texts in the following format: "Your goal is to identify rhetorical and psychological techniques in the given text.\nInput:. . . \nOutput:”.

All texts from the dataset were filtered as follows:

1. All unnecessary line breaks in the texts were removed.

2. All unnecessary "\" characters in line breaks were removed. That is, if the text contained a construction of the following form "\\n", it was replaced with "\n".

3. Texts with prompts and responses longer than 4096 characters were not included in the dataset.

Models. LLaMA-2-7b-chat [5] and Mistral-7b-Instruct-v0.2 (Jiang et al., 2023) were used as models.

Training and Inference Parameters. Both models were trained in both SFT mode and using the LoRA technique, using the huggingface transformers [6] and torch frameworks on 4 A100 GPUs for no more than 12 hours per experiment.

In SFT mode, both models were trained with the parameters, presented in Appendix A.1.

When training with the LoRA technique, the models were trained with the same parameters, but with the following LoRA parameters, presented in Appendix A.2.

For inference, the vLLM framework (Kwon et al., 2023) was used with the following parameters presented in Appendix A.3.

3.1.2 BERT-like approach

As a result of training with deberta-base-cased, comparable results to the generative approach were achieved: F1 score of 0.638.

Idea. The main idea of this approach is to train a fully connected layer for hierarchical classification on top of a frozen embedding model for the task of hierarchical multilabel classification.

Dataset. For model training, we utilized the training dataset provided by the authors. The dev dataset was used for selecting the final model. For the final prediction, the model was trained on a mixture of the train and dev datasets.

Data Format. To predict propaganda techniques, we represented the data labels in the format of an acyclic graph, where the nodes of this graph are generalized technique types ("Ethos", "Pathos", ..., etc.), and the leaves are concrete techniques ("Name calling", "Doubt", ..., etc.). Instead of predicting only specific techniques, we predict generalized techniques as well. For example, if the current sample needs to predict the technique {"Whataboutism"}, the model should predict Anc(y) = {"Whataboutism", "Distraction", "Reasoning", "Ad Hominem", "Ethos", "Logos", "Persuasion"}. Thus, datasets are formed as sets of pairs: (x, Anc(y)), where x is the text, and y is the set of techniques contained in the text x.

Model. The frozen model used to obtain embeddings was deberta-base-cased (He et al., 2021).

[4] https://propaganda.math.unipd.it/semeval2023task3/

[5] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[6] https://huggingface.co/docs/transformers/index

```
{
"id": "train_71410",
"input": "Your goal is to identify rhetorical and psychological techniques in the
     given text.\nInput:Tunnel to Schiff\nBunker where our next President will be
     chosen.\nOutput:",
"output": [
        "Causal Oversimplification",
        "Smears"
    ],
"full_input": "Your goal is to identify rhetorical and psychological techniques
    in the given text.\nInput:Tunnel to Schiff\nBunker where our next President
    will be chosen.\nOutput:[\"Causal Oversimplification\", \"Smears\"]"
}
```

Figure 1: Sample from the dataset example

The embedding of the entire text from the embedding model was taken, followed by the application of several dropout layers in parallel, and the results were averaged. At the end, trainable linear layers were used for classification.

Training Parameters. Training parameters of the discussed models can be seen in Appendix A.4.

3.1.3 Results

Generative approach. Experiments with pretraining models using the LoRA technique did not yield the desired result. Consistent responses from the models in JSON format could not be achieved, and there was not enough time to develop rules for formatting their outputs.

The results of the models trained with SFT are presented in the Table 1.

Mistral	LLaMa 2
0.56 F1	0.65 F1

Table 1: Results for subtask №1

BERT-like approach. As a result of training with deberta-base-cased, comparable results to the generative approach were achieved: F1 score of 0.638.

Final Prediction. For the final prediction, the LLaMA-2-7-chat model was fine-tuned on dataset 2 with the same parameters as dataset 1, and with the following parameters, shown in Appendix A.5.

It achieved an F1 score of 0.61339 and secured the 18th position on the leaderboard.

3.2 Subtask №2a

Idea. This task resembles subtask №1, but besides text, it involves meme images. It was tackled using a trainable linear layer for hierarchical classification atop frozen multimodal text-to-image embedding models.

Dataset. For model training, we utilized the training dataset provided by the authors. The dev dataset was used for selecting the final model. For the final prediction, the model was trained on a mixture of train and dev datasets.

Data Format. To represent propaganda techniques, we employed the format of an acyclic graph from the first subtask. The dataset is presented as triples: (img, x, Anc(y)), where img and x represent the image and text of the current meme, and y is the set of techniques contained in the current meme.

Models. CLIP (Radford et al., 2021) and BLIP (Li et al., 2023) were used as models to obtain embeddings for texts and images. The embeddings of the full text and the entire image were concatenated, and several dropout layers were applied in parallel, with the results averaged. At the end, trainable linear layers were used for classification.

Training Parameters. The models were trained on a single P100 GPU on the Kaggle platform. Parameters for BLIP/CLIP were frozen.

Other parameters are shown in Appendix A.6.

Results. The results of the models trained on dataset 1 are presented in the Table 2.

CLIP	BLIP
0.648 F1	0.633 F1

Table 2: Results for subtask №2a

3.3 Subtask №2b

This subtask differs from subtask №1 only in that it requires predicting whether there is any propaganda technique present in the text at all. We used the same approach, the same models with the same hyperparameters, trained on the same datasets as in subtask №1. The only difference is in the data format - the dataset consists of triples (img, x, y), where img and x are the image and text of the current meme, and y is a flag indicating whether the current meme contains a propaganda technique.

Results. The results of the models trained on dataset 1 are presented in the Table 3.

CLIP	BLIP
0.72 F1	0.748 F1

Table 3: Results for subtask №2b

For the final prediction, the CLIP model was selected, achieving an F1 score of 0.772 and securing the 11th position on the leaderboard.

4 Further research

We did not have time to take more powerful models and experiment with them to solve subtask №1, but we are confident that Mixtral 8x7b or Llama-2-13b-chat would yield better results. Additionally, we did not have time to add data from the PTC (He et al., 2021) corpus to the dataset, but we are sure that it would have provided an even greater improvement. We also did not have time to test generative models that take text with images as input and generate text responses, for example, RUDOLPH (Radford et al., 2021).

References

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Jakub Piskorski, Nicolas Stefanovitch, Valerie-Anne Bausier, Nicolo Faggiani, Jens Linge, Sopho Kharazi, Nikolaos Nikolaidis, Giulia Teodori, Bertrand De Longueville, Brian Doherty, Jason Gonin, Camelia Ignat, Bonka Kotseva, Eleonora Mantica, Lorena Marcaletti, Enrico Rossi, Alessio Spadaro, Marco Verile, Giovanni Da San Martino, Firoj Alam, and Preslav Nakov. 2023. News categorization, framing and persuasion techniques: Annotation guidelines. Technical report, European Commission Joint Research Centre, Ispra (Italy).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision.

A Appendix

This appendix shows the training and generation parameters for the models described above in the text.

A.1 Hyperparameters for models training in SFT mode

- BATCH_SIZE = 4
- GRADIENT_ACCUMULATION = 4
- LEARNING_RATE = 1e-5
- MAX_LEN = 4
- WARMUP_STEPS = 4
- in fp32 and for 2 epochs

A.2 Hyperparameters for models training with LoRA technique

- LORA_R = 8
- LORA_ALPHA = 16
- LORA_DROPOUT = 0.05
- TARGET_MODULES = ["q_proj", "k_proj", "v_proj", "o_proj"]

A.3 Hyperparameters for models output generation with vLLM

- TOP_K = 50
- TOP_P = 1 and 0.9
- MAX_TOKENS = 600
- TEMPERATURE = 1, 0.8, 0.6, and 0.2

A.4 Hyperparameters for BERT-like models training for subtask №1

- Models were trained on a single P100 GPU on the Kaggle platform [7].
- Optimizer: AdamW with linear scheduler
- Learning rate (LR): 2e-5
- Batch size: 8
- Warmup steps: 100

A.5 Hyperparameters for models output generation with vLLM for final prediction

- TOP_K = 50, TOP_P = 0.9
- MAX_TOKENS = 600
- TEMPERATURE = 0.8

A.6 Hyperparameters for BERT-like models training for subtask №2a and №2b

- Optimizer: Adam
- Learning rate (LR): 2e-3
- Batch size: 10

[7] https://www.kaggle.com/