EURECOM at SemEval-2024 Task 4: Hierarchical Loss and Model Ensembling in Detecting Persuasion Techniques

Youri Peskine and **Raphael Troncy** and **Paolo Papotti**
EURECOM, France
firstname.lastname@eurecom.fr

Abstract

This paper describes the submission of team EURECOM at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. We only tackled the first sub-task, consisting of detecting 20 named persuasion techniques in the textual content of memes. We trained multiple BERT-based models (BERT, RoBERTa, BERT pre-trained on harmful detection) using different losses (Cross Entropy, Binary Cross Entropy, Focal Loss and a custom-made hierarchical loss). The best results were obtained by leveraging the hierarchical nature of the data, by outputting ancestor classes and with a hierarchical loss. Our final submission consist of an ensembling of our top-3 best models for each persuasion techniques. We obtain hierarchical F1 scores of 0.655 (English), 0.345 (Bulgarian), 0.442 (North Macedonian) and 0.177 (Arabic) on the test set.

1 Introduction

Online misinformation is a complex research topic. It can appear in many shape and forms (text, images, videos, etc.), within different contexts (political debates, news articles, social media posts, etc.). As memes generate high engagement on social media, they are also used for disinformation campaign by exploiting rhetoric persuasion. This year's "SemEval-Task4: Multilingual Detection of Persuasion Techniques in Memes" task aims at detecting named persuasion techniques in memes. The full overview of the task and its sub-tasks are detailed in (Dimitrov et al., 2024).

As a brief summary, SemEval-2024 Task 4 consist of detecting persuasive techniques in Memes. The task breaks down into 3 sub-tasks; sub-task 1 use the textual content of the meme to detect the persuasion techniques, sub-task 2a use the whole image and text to detect the persuasive techniques, while sub-task 2b only consist of binary detection. In sub-task 1, a total of 20 persuasion techniques are used. In this work, we only describe our solution to tackle this sub-task 1.

Our approach consists of an ensembling model of our top-3 best models for each persuasion techniques. In our experiments, we reached the best results leveraging the hierarchical nature of the data, with hierarchical loss, and outputting ancestor classes. Our method can be reproduced using the code at https://github.com/D2KLab/semeval-2024-task-4.

2 System Description

In this section, we describe the system used in our submission. We also present approaches that were considered but not kept in our final submission.

2.1 Models

We experimented with multiple transformer-based models to tackle persuasion detection in the textual content of the memes.

- **BERT** (Devlin et al., 2019): First introduced in 2018, this model is based on the bidirectional transformer encoder architecture (Vaswani et al., 2023) trained with masked language model and next sentence prediction tasks.
- **BERT-HarMe**[1]: This model is a fine-tuned version of BERT on multiple datasets[2] (Kiela et al., 2021; Suryawanshi et al., 2020) about harmful/hateful speech in memes.
- **RoBERTa** (Liu et al., 2019): This model changes the BERT pre-training approach, making it more robust.
- **AlBERT** (Lan et al., 2020): AlBERT focuses on reducing the number of parameters

[1] https://huggingface.co/limjiayi/bert-hateful-memes-expanded

[2] https://github.com/di-dimitrov/harmeme

Dataset	Size
SemEval-2024 Train	7000
SemEval-2021 Train+Validation+Dev	951
PTC (sampled)	427

Table 1: Datasets considered for training our models.

of BERT to increase the training speed and lower memory requirements.

- **DistilBERT** (Sanh et al., 2020): This model uses knowledge distillation during pretraining to reduce the size of BERT.
- **DeBERTa** (He et al., 2021): DeBERTa improves on BERT and RoBERTa by introducing a disentangled attention mechanism and an enhanced mask decoder.

2.2 Datasets

In this task, we use multiple training datasets. We experimented adding the train, validation and dev sets from SemEval-2021 Task 6 (Dimitrov et al., 2021) and the PTC corpus (Da San Martino et al., 2020) to the training data. Table 1 shows the datasets and their respective sizes.

- **SemEval-2021 Task 6**: This dataset also annotates memes with regards to the same 20 persuasion techniques. The train, validation and dev sets are appended to the training set of this task without any modification.
- **PTC Corpus**: This dataset contains news articles annotated at the span level with regards to 18 propaganda techniques. We first split the articles into sentences and transfer the span-level label to sentence-level. In this dataset, some labels are the same as this year's task, and can be aligned in a straightforward manner. However, when propaganda labels are different, they often correspond to multiple persuasion techniques. To align these labels, we add all the corresponding persuasion techniques valid for the propaganda. We only appended sentences that contain a propaganda technique to the training set of this task (around 5% of the total number of sentences).

2.3 Outputting ancestor classes

In this task, the goal is to detect the 20 persuasion techniques, but they appear in a hierarchical framework. The official metrics of the challenge are hierarchical F1 (**F1H**), hierarchical precision (**PreH**) and hierarchical recall (**RecH**), which all take into consideration the hierarchical nature of the data. Since ancestor nodes are inherently outputted when detecting child nodes, we also tried to directly detect the ancestor classes. This raises the number of classes to 28 (instead of 20). Thus, the ancestor node can still be outputted even if it's child node has not been detected, resulting in better performing models.

2.4 Losses

We also experimented with different training losses, which address multiple aspects of the data. For example, balancing the classes misrepresentation in the data with class weights, or using hierarchical loss to reflect the hierarchical nature of the data.

- **Binary Cross Entropy (BCE) Loss**: This loss computes BCE losses for each class, weighted with the inverse frequency of its label, and sum them. This loss requires the output layer to have the size of number of classes.
- **Cross Entropy (CE) Loss**: We used 20 different CE losses for each class, weighted according to the inverse frequency of each label. Each loss computes the performance of the model at detecting a specific class. The final loss is the sum of the 20 losses. This loss requires the output layer to have twice the size of number of classes.
- **Focal Loss (FL)** (Lin et al., 2020): This loss addresses class imbalance by down-weighting the loss assigned to well-classified examples. We used the implementation proposed by (Edgar et al., 2020). This loss requires the output layer to have the size of number of classes.
- **Custom Hierarchical Loss (HL)**: In order to reflect the hierarchical nature of the data, we implemented a custom hierarchical loss function. This function uses max pooling on logits x^c from children classes of the same ancestor a (e.g. Name Calling, Doubt, Smears, Reductio ad Hitlerum and Whataboutism are all children of the Ad hominem ancestor). The newly created logit correspond to the output of the model on the corresponding ancestor. Thus, we can compute the BCE Loss between

this output and the true label y^a of the ancestor. We can iterate by max-pooling all the logits in the next ancestor. Note that logits can correspond to children or ancestor classes (e.g. the Logos ancestor pools the logits of Justification, Repetition, Intentional Vagueness, and Reasoning, even though the logits of Justification and Reasoning are also pooled from other child classes). We then sum all these BCE losses together, which measure how well the model performs to detect the ancestor rather than each persuasion techniques. Before summing this loss to the original classification loss of the techniques (CE, BCE or FL), we apply a normalization factor α. In practice, we found best results when α is equal to 0.5. Equations 1 and 2 describe the computation of this loss. $\mathcal{A}$ describes the ensemble of all ancestor techniques.

$$\mathcal{L}_{HL} = \mathcal{L}_{CE,BCE} + \alpha \cdot \sum_{a \in \mathcal{A}} \mathcal{L}^a_{BCE} \tag{1}$$

$$\begin{aligned} \mathcal{L}^a_{BCE} &= y^a \cdot log\sigma(\max(\{x^c\}_{c \in child(a)})) \\ &+ (1 - y^a) \cdot log(1 - \sigma(\max(\{x^c\}_{c \in child(a)}))) \end{aligned} \tag{2}$$

2.5 Data augmentation

Some persuasion techniques have very little training data available in the datasets. We tried generating new samples for the bottom 5 classes with different methods.

- **Round Translation**: We translated every sample in French and translated them back to English. This can generate new sentences similar to the original ones. However, this new data is very limited and will not be varied.

- **GPT-4-Turbo Generation** (et al., 2023): We used GPT-4-Turbo to generate completely new sentences corresponding to a persuasive technique. As showed in (Peskine et al., 2023), definitions of the class label have a significant impact in the performance of GPT models. We provided the definition of the persuasive technique provided by the organizers[3] in the system prompt, along with 5 randomly selected samples. We then used few-shot prompt technique with 5 more randomly selected samples, and finally asked the model to generate a new sentence. We generated two sets of 30 and 50 examples for five classes. For reproducibility measures, the full prompt is available in Appendix A.

2.6 Training process

For training our models, we use the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 2e-5, and a weight decay of 0.01. We also use a ReduceLROnPlateau Learning rate scheduler, reducing the earning rate by a factor of 0.7 if results have not improved in 4 epochs. Most experiments are done on 10 epochs, saving the best model (according to F1H) on the validation set. We also experimented with freezing the first few layers of the pre-trained BERT-based model to keep its acquired knowledge when trained on massive amount of data.

2.7 Ensembling

We trained many models according to different combinations of the previous parameters. Our final submission consists of a majority voting among the top-3 models for each persuasion technique evaluated on the dev set and according to the F1-score. These models are not necessarily the best models overall according to hierarchical F1, but demonstrate effectiveness in detecting specific persuasion technique. We also perform majority-voting on ancestor classes with models that output them (Section 2.3).

3 Results

We share our results on the dev set provided by the organisers in Table 2. These results show the performance of some single models as well as the performance of the ensembling used in the final submission. Table 4 shows the performance of each class on the dev set, using the ensembling model for classification. Table 3 shows the results of our final submission on the 4 test languages: English, Bulgarian, North Macedonian and Arabic. We translate non-English languages using `py-googletrans`[4] to English in order to run our models and obtain the predictions. We would like to note that our official submission for the Arabic language was incorrect, due to Arabic-to-English translation errors on our

[3] https://propaganda.math.unipd.it/semeval2024task4/definitions.html

[4] https://github.com/ssut/py-googletrans

Model	Data	Classes	Loss	F1H	PreH	RecH
BERT	2024	20	CE	0.612	0.603	0.621
BERT	2024+2021	20	BCE	0.623	0.561	**0.700**
BERT	2024+2021	28	HL	**0.640**	0.626	0.654
BERT	2024+2021+PTC	28	HL	0.633	**0.647**	0.618
BERT	2024+2021	28	FL	0.629	0.638	0.620
BERT	2024+2021	20	FL	0.611	0.635	0.588
BERT	2024	28	CE	0.629	0.612	0.646
RoBERTa	2024+2021	20	CE	0.619	0.610	0.628
RoBERTa	2024+2021	28	CE	0.631	0.610	0.653
BERT-HarMe	2024+2021	20	CE	0.625	0.599	0.652
BERT-HarMe	2024+2021	28	CE	0.639	0.651	0.627
BERT-HarMe	2024+2021	28	HL	0.634	0.634	0.634
BERT-HarMe	2024+GPT-augmented	28	CE	0.634	0.605	0.666
AlBERT	2024+2021	20	CE	0.604	0.600	0.607
DeBERTa	2024+2021	20	CE	0.617	0.617	0.618
DistilBERT	2024+2021	20	CE	0.602	0.622	0.584
Ensembling	Top-3 best models			**0.675**	**0.650**	**0.702**

Table 2: Results on the dev set of some of the models we tried. Other models with different combination of parameters are used in the ensembling and not showed here due to space, but obtain similar performances.

Language	F1H	PreH	RecH
English	0.655	0.628	0.685
Bulgarian	0.345	0.367	0.325
North Macedonian	0.442	0.520	0.384
Arabic	0.177	0.343	0.119
Arabic (unofficial)	0.439	0.369	0.544

Table 3: Results on the test set with our ensembling model, translating non-English languages to English.

end. We corrected the error and also show the performance of the model, albeit being an unofficial result.

4 Discussion

Model-wise, our best results were obtained using BERT, RoBERTa and BERT-HarMe. We ultimately did not use any of AlBERT, DeBERTa and DistilBERT models in our final submission as those were not in any top-3 best performing models of any persuasion techniques. The BERT-HarMe models were the best-performing on the detection of 'Slogans', 'Appeal to Authority', 'Flag-waving', 'Appeal to fear/prejudice', 'Black-and-white Fallacy/Dictatorship', 'Thought-terminating cliché', 'Presenting Irrelevant Data (Red Herring)', 'Glittering generalities (Virtue)', 'Doubt', 'Logos', 'Justification' and 'Distraction' classes. RoBERTa models were the best-performing for 'Repetition', 'Bandwagon', 'Ethos'.

We also noticed a slight performance increase by adding the 2021 dataset during training, which was not necessarily true when adding the PTC corpus. This is probably due to the fact that the PTC Corpus is about news articles and not memes. Our data-augmentation experiments on round-translation did not improve the results at all, while the GPT-4-Turbo augmentation experiments provided a very slight boost, but not for the augmented classes.

The hierarchical nature of the task and the evaluation metrics were reflected in the results, as most of our best performing models are outputting 28 classes by including the ancestors and/or are trained with Hierarchical Loss (**HL**). However, best models at detecting 'Causal-Oversimplification' are using BCE Loss.

We can see in Table 4 that some persuasive techniques are easier to detect than others. For example, 'Appeal to authority' seems to be the easiest class to detect, and 'Obfuscation, Intentional vagueness, Confusion' the hardest. Training data seems to lightly correlate with performance results, with some strong outliers like 'Smears' under-performing comparing to it's high number of training samples, and 'Bandwagon' over-performing. As for the ancestor classes, the highest-level 'Logos', 'Ethos' and 'Pathos' have the highest performance, while those composed of the hardest persuasive techniques to detect like ‘Simplification’, ‘Distraction’ and ‘Reasoning’ have lower performance.

Technique	F1H
Repetition	0.516
Obfuscation	0.000
Slogans	0.495
Bandwagon	0.583
Appeal to authority	0.891
Flag-waving	0.623
Appeal to fear/prejudice	0.425
Causal Oversimplification	0.304
Black-and-white Fallacy	0.549
Thought-terminating cliché	0.330
Straw Man	0.286
Red Herring	0.182
Whataboutism	0.442
Glittering generalities (Virtue)	0.562
Doubt	0.437
Name calling/Labeling	0.617
Smears	0.583
Reductio ad hitlerum	0.526
Exaggeration/Minimisation	0.492
Loaded Language	0.682
Logos	0.773
Reasoning	0.552
Justification	0.727
Simplification	0.496
Distraction	0.389
Ethos	0.810
Ad Hominem	0.742
Pathos	0.704

Table 4: Results of our ensembling model on the dev set, per-class.

5 Conclusion

In this paper, we describe the system team EURECOM used for sub-task 1 at SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. We explore multiple BERT-based models, training datasets, losses, data augmentation procedures, and training process. Our final submission consists of an ensembling model that performs majority voting between our top-3 best performing models for each persuasive technique. We find that some pre-trained models on harmful meme data are competitive, and that incorporating hierarchical information in the training process, such as outputting the whole 28 classes (including the ancestors) or using a hierarchical loss significantly improves the results. We obtain a hierarchical F1 score of 0.675 on the dev set and 0.655 (English), 0.345 (Bulgarian), 0.442 (North Macedonian), 0.177 (Arabic) on the test set.

Acknowledgements

This work has been partially supported by CHIST-ERA within the CIMPLE project (CHIST-ERA-19-XAI-003) and by ANR within the ECLADATTA project (ANR-22-CE23-0020).

References

Giovanni Da San Martino, Alberto Barrón-Cedeño, Henning Wachsmuth, Rostislav Petrov, and Preslav Nakov. 2020. SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles. In *14th International Workshop on Semantic Evaluation (SemEval)*, pages 1377–1414. International Committee for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. SemEval-2024 Task 4: Multilingual Detection of Persuasion Techniques in Memes. In *18th International Workshop on Semantic*

Evaluation (SemEval), SemEval 2024, Mexico City, Mexico.

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. SemEval-2021 Task 6: Detection of Persuasion Techniques in Texts and Images. In *15th International Workshop on Semantic Evaluation (SemEval)*, pages 70–98. Association for Computational Linguistics.

Riba Edgar, Mishkin Dmytro, Ponsa Daniel, Rublee Ethan, and Gary Bradski. 2020. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch. In *Winter Conference on Applications of Computer Vision*.

OpenAI et al. 2023. GPT-4 Technical Report.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2021. The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization.

Youri Peskine, Damir Korenčić, Ivan Grubisic, Paolo Papotti, Raphael Troncy, and Paolo Rosso. 2023. Definitions Matter: Guiding GPT for Multi-label Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4054–4063, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Shardul Suryawanshi, Bharathi Raja Chakravarthi, Mihael Arcan, and Paul Buitelaar. 2020. Multimodal Meme Dataset (MultiOFF) for Identifying Offensive Content in Image and Text. In *Second Workshop on Trolling, Aggression and Cyberbullying*, pages 32–41, Marseille, France. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need.

A GPT-4-Turbo Prompts

For reproducibility, we share the exact prompt used to generate new examples using GPT-4-Turbo (as of January 2024):

```
[system] Your task is to generate short sentences that contains the <current_propaganda_technique> propaganda technique.
The definition of the <current_propaganda_technique> propaganda technique is the following: <current_propaganda_technique_definition>


Here are some examples:
 - <Random example x5>

([user] Please generate a short sentence that contains the <current_propaganda_technique> propaganda technique similar to the examples, on similar topics.
[assistant] <Random example>) x5
[user] Please generate a short sentence that contains the <current_propaganda_technique> propaganda technique similar to the examples, on similar topics.
```