Detection of Persuasion Techniques in Memes

Gözde Ünver, Fami Mahmud, Batıkan Özkul
Technical University of Munich

Abstract

This paper explores the concept of persuasion techniques embedded within memes and presents a solution for their detection. Persuasion techniques, or persuasive language, involve presenting an idea with the ulterior motive of convincing the recipient or consuming party. Unfortunately, this methodology has been repeatedly used to spread certain agendas, which may, in some cases, contain hate. Our study delves into a specific method we devised to detect these techniques in memes and social media posts. In the final results, our team was able to achieve a top 10 position in all three subtasks of the competition.

1 Introduction

The project explained in this paper was structured around an interacademic competition hosted by SemEval (International Workshop on Semantic Evaluation) 2024. A brief explanation of the competition's goal would be: developing models capable of detecting persuasion techniques in a given meme, respecting a hierarchical order. The final test of the competition was designed for multiple languages such as Arabic and French, but we only developed our models for English.

The hierarchy is essentially a directed acyclic graph that groups subsets of techniques sharing similar characteristics in a hierarchical structure, predetermined by SemEval. The hierarchy of persuasion techniques used was inspired by the European Commission Joint Research Centre.

There were three different subtasks in the competition, each with its unique criteria and evaluation metrics. The overall goal remains unchanged for each subtask, but the data in use as well as the implementation differ.

1.1 Subtask 1

In the first subtask of the competition we are given only the "textual content" of a meme, and are required to identify which of the 20 persuasion techniques, organized in a hierarchy, it uses. If the ancestor node of a technique is selected, only a partial reward is given. This is a hierarchical multi-label classification problem.

1.2 Subtask 2A

This time we are required to identify which of the 22 persuasion techniques, organized in a hierarchy, are used both in the textual and in the visual content of the meme (multi-modal task). meaning the training data consists not only textual content but visual too. also the increase in number of persuasion techniques are due to addition of 'Appeal to Strong emotion' and 'Transfer', which are both child nodes of Pathos.

1.3 Subtask 2B

Last subtask of the competition is a binary classification problem; identifying whether a given meme contains a persuasion or not. It can be considered a version of subtask 2A in which the hierarchy is cut at the first two children of the root node.

1.4 Evaluation

For the evaluation of models, two different scorings were used; subtask 1 and 2A were evaluated using hierarchical-F_1 score, whereas the Subtask 2B were evaluated using macro-F_1 score.

1.4.1 Hierarchical F_1 Score

The logic behind hierarchical F_1 score is as follows, please consider the 'gold label' as correct label i.e. 'y' :
1- if the prediction is a leaf node and it is the correct label, then a full reward is given. For example Red Herring is predicted and it is the gold label as well.
2- if the prediction is NOT a leaf/child node and it is an ancestor of the correct gold label, then a partial reward is given .
3- if the prediction is not an ancestor node of the correct label, then a null reward is given.

1.5 Distribution of the tasks

Gözde Ünver: Implementation of the main subtask 1 pipeline (the hierarchical classifier) and its model trainings, Word2Vec attempt, memes dataset preprocessings for subtask 1 and 2B, the complete implementation, experiments and trainings of subtask 2B, our chair's server setup for model trainings
Fami Mahmud: Implementation of the foundational subtask 1 pipeline (multilabel classification), WandB integration and its model trainings. Implementation of various experimental enhancements of the pipeline as transfer learning, ensembling with stacking method and reversed hierarchy.
Batıkan Özkul: complete implementation of Subtask 2A, experimenting with untrained zero shot models for Subtask1, analyzing and manipulating training data(adding underrepresented variables from PTC dataset, data augmentation) and other preprocessing techniques, training models.

2 Related Work

Image and Text Models After the introduction of the transformer based models, there were major performance improvements compared to the more traditional approaches like Word2Vec (Mikolov et al., 2013), TF-IDF (Das and Chakraborty, 2018), bi-LSTM (Schuster and Paliwal, 1997). Hence, in our project we explored various transformer models. Self-attention provided leveraging important information from the given sequence more effectively (Vaswani et al., 2023). One of the most prominent models that use self-attention is BERT (Devlin et al., 2019), a bidirectional transformer model. With its multiple self-attention blocks, BERT was originally pretrained for masked language modelling and next sentence prediction. With fine-tuning, it can be used as a powerful text encoder for downstream classification tasks. Successors of BERT-based models have proposed various solutions to improve the BERT performance thus, we experimented all of the following models to find the best fit to our project. Those models were; RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021), ALBERT (Lan et al., 2020), BERTweet (Nguyen et al., 2020), XLNet (Yang et al., 2020), XLM-RoBERTa (Conneau et al., 2020). These models were pretrained on datasets that were not very similar to ours. Furthermore, we were asked to implement unique pipelines that might require more than one type of model due to the difficulty of the tasks. As a result, we needed to find the best performing text model(s) by finetuning them for our purpose. We started with their base versions and then continued with their large versions to reach the optimal pipelines. On the other hand, after the success of transformer models on texts, there have been various successful applications of transformer models as image models. Convolutional models are translation equivariant, less expensive and have good performance for the image processing tasks. On the other hand, with more pretraining and large datasets transformer models offer better performance. That's why, we experimented both approaches in our tasks with ResNet (He et al., 2015) and EfficientNet (Tan and Le, 2020) as CNN models and Google-ViT (Dosovitskiy et al., 2021) and CLIP (Radford et al., 2021) as transformer models.
Previous research has predominantly focused on identifying propaganda within the textual content of news articles (Da San Martino et al., 2019; Barrón-Cedeño et al., 2019; Rashkin et al., 2017). This challenge was addressed in SemEval-2020 task 11 on Detection of Persuasion Techniques in News Articles (Da San Martino et al., 2020), as well as in SemEval-2020 task 3: Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multi-lingual Setup (Piskorski et al., 2023). The endeavor to detect persuasion techniques was further extended to analysis of textual and visual content of memes, initially introduced in the SemEval-2021 task 6 (Dimitrov et al., 2021). In subtask 1, the three highest-scoring teams all leveraged fine-tuned BERT-based transformer models (Tian et al., 2021; Feng et al., 2021; Gupta and Sharma, 2021). To additionally incorporate the visual content of memes, the teams compared two approaches: one utilizing solely multimodal representations and the other employing fusion strategies for combining text transformers and visual features. Across all experiments, the multimodal approach outperformed textual representation.
Although SemEval-2021 task 6 is closely related to our task, it ignores the hierarchy of the persuasion techniques. To the best of our knowledge, no previous research has been been conducted on a hierarchical approach to detecting persuasion techniques in memes. Our work suggests a possible pipeline in which persuasion techniques are predicted hierarchically.

3 Experimental Setup

3.1 Dataset

The dataset (Dimitrov et al., 2024) for the subtasks were provided by the organizers of the competition and the sizes of the dataset can be seen in Table 1. For the subtask 1, we only used the json files that contained the texts extracted from the memes, ids, list of labels and the links to the memes. Subtask 2A and 2B also use the images. Until the actual test set was published, we had improved our pipelines according to the development set results displayed in the leaderboard of the competition. For our final submissions on the test sets, we merged the development sets into the training sets.

	Train Set	Validation Set	Development Set	Test Set
Subtask 1	7000	500	1000	1500
Subtask 2A	7000	500	1000	1500
Subtask 2B	1200	150	300	600

Table 1: Summary of the memes dataset we used in subtasks

3.2 Modifications on training data

After analyzing the training data, we realized that some of the classes were greatly underrepresented. For example, in subtask 1, the entire size of the training data was 7000 elements. Out of these 7000, 1990 were labeled as "Smears," meaning between 30-35% of the data represented a single label, whereas there were 5 classes with fewer than 100 elements. We theorized that such a polarized dataset would undermine the training process of the models. In order to counter this, we acquired another dataset named 'PTC' from SemEval's Detection of Propaganda Techniques in News Articles competition (PTC, 2023-2024 Detection of Propaganda Techniques in News Articles). After analyzing the dataset and selecting the underrepresented classes, we merged it with the original training dataset and used this new dataset for training. However, contrary to our beliefs, after comparing two models—one trained on the original data and the second on the modified/enriched data—we observed that the models trained on the original data performed better. In the end, this dataset was not used, but it provided us with a better understanding of the task at hand and helped mitigate our concerns about the dataset.

3.3 Server Usage

In the beginning, we used free Tesla T4 GPUs of Google Colab for our implementations. Later, we required more and uninterrupted GPU time for different model trainings and hyperparameter searches because the free usage of Colab was limited for some hours. That's why we started to use 12 GB NVIDIA GeForce RTX 3080 server from the chair of our praktikum. We were not the only users of that server. We had 4 days in each week to do all of our trainings. This was also another challenge because even the subtask 1 trainings usually took one full day for one type of model.

3.4 Tools and Libraries

We used Hugging Face models, PyTorch, Weights and Biases, Jupyter notebook in our implementations and Notion for taking meeting notes and presenting our updates.

4 Methodology

Among the 3 subtasks, we started implementing on subtask 1, then subtask 2B, and finally subtask 2A which is the combination of the first two tasks. The reason for this order implementation was because we wanted to learn the best pipelines and models from 1 and 2B and use them in 2A.

4.1 Subtask 1

In Subtask 1, we initially developed a pipeline for direct multilabel prediction. Subsequently, we refined this pipeline to consider the hierarchy of persuasion techniques. The hierarchical pipeline is illustrated in Figure 1. We partitioned the prediction process into the ten nodes of the hierarchy and implemented an iterative hierarchical processing approach. For each node, we fine-tuned a BERT-based model augmented with an additional classification layer to predict its child nodes. This classification layer takes the CLS token, representing the entire content of the text, as input. If, for a given sample at a particular node, the predicted child node probability exceeds the specified threshold, the sample is passed on to the child node. Each node is assigned an individually optimized threshold. The final multilabel prediction for a sample is attained when either all child node probabilities fall below the threshold or leaf nodes are reached.

We conducted experiments with various models to identify the best-performing one for our specific task, including DeBERTa-V3(He et al., 2021),

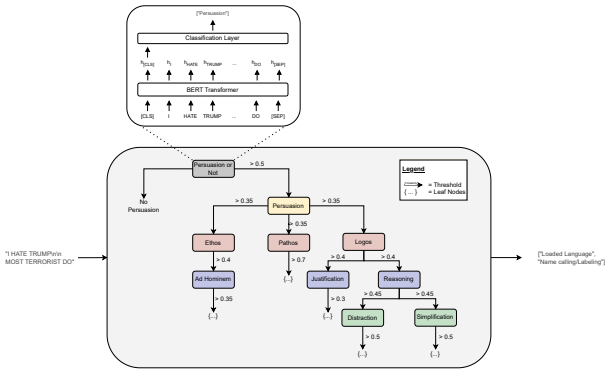

Figure 1: Subtask 1 pipeline

BERTweet(Nguyen et al., 2020), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2020), XL-Net (Yang et al., 2020), XLM-RoBERTa (Conneau et al., 2020). Among these, DeBERTA-V3-large emerged as the most effective and robust model. The pre-trained models underwent complete fine-tuning for each node over three epochs, while optimizing hyperparameters such as learning rate and threshold. Threshold values ranged from 0.35 to 0.7 and varied for each node based on the differing number of child nodes. The specific thresholds can be found in Figure 1. A comparison of these models' performance is presented in Section 5.1.

4.2 Subtask 2B

In subtask 2B, we only used one type of text model; BERTweet-Large(Nguyen et al., 2020) and not DeBERTa-V3-large (He et al., 2021) as observed in subtask 1 because together with an image model, the pipeline required too much memory space so we used the second best text transformer. We experimented with different image models and pipelines. To process the text in the meme and the actual meme image, we obtained the text and image embeddings from the text and image models separately and then concatenated them in Figure 2. The concatenated embedding was passed to two consecutive linear layers to predict the class of the meme; propagandistic or not propagandistic. The input dimension of the first linear layer was the size of the concatenated embedding vector. The second linear layer always had the dimension of 512 and we used ReLU activation function between the linear layers. Using this pipeline, we experimented with many image models. All models we used were pretrained and we fine tuned the complete pipeline end-to-end for 10 epochs using our memes dataset. We did hyperparameter search on small learning rates; 5e-6, 5e-5 and 5e-4 because this pipeline with two models performed better with smaller learning rates due its size and complexity. We evaluated the performances of our models with F_1-macro score.

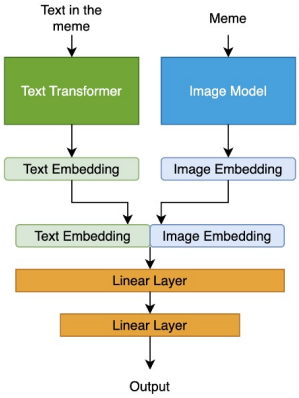

Figure 2: Subtask 2B pipeline

Even though CLS embedding of a sequence is generally used for sequence representation, we obtained the best scores on the development set when we took the average of the embeddings of all tokens that are the inputs of their respective models. The best performing image model in this approach was Google/ViT-base-patch32-224-in21k (Dosovitskiy et al., 2021). After finding the best performing pipeline, we tried out some different methods to improve the results. However, our original pipeline implementation described above, still performed the best on the development set that's why, we continued using that solution in our final test set submission. The detailed comparisons of all the model performances in each experiment and the final pipeline can bee seen in Section 5.2.

4.3 Subtask 2A

In subtask 2A, due to time constraints during development (remaining time until the deadline), we utilized only one type of text model: BERTweet-Large(Nguyen et al., 2020), which emerged as the best-performing model in subtask 1, and the best-performing image model from Subtask 2B, namely Google/ViT-base-patch32-224-in21k.
The pipeline for Subtask 2B remains the same, as illustrated in Figure 2. To process the text in the meme and the actual meme image, we obtained the text and image embeddings from text and image models separately, subsequently concatenating them. The concatenated embedding was then fed through two consecutive linear layers to enable multilabel classification. Ideally, we would

have trained a node/model for each persuasion technique, but time constraints hindered this approach. The pipeline was completed 24 hours before the deadline, and the total training process for a multi-node model would be $O(t * n)$, whereas a single model conducting multilabel classification would be $O(n)$.

5 Evaluation

	F_1 Dev	F_1 Test	Ranking Dev	Ranking Test
Subtask 1	0.6391	0.6738	12	4
Subtask 2A	0.6784	0.6771	5	6
Subtask 2B	0.8536	0.7841	1	9

Table 2: Test and development set performances of our final implementations. F_1 for subtask 1 and 2A refers to hierarchical-F_1 while for subtask 2B it is macro-F_1

As it can be seen in Table 2, we were always among the top 10 groups for all the subtasks for the test set. In the following subsections, we explain in detail which models we chose according to their validation and development set performances on each subtask.

5.1 Subtask 1

During the development phase, the models were trained exclusively on the training set, fine-tuning hyperparameters on the validation set, and assessing performance on the development set. Table 3 illustrates the performance of the various models we investigated.

	Hierarchical F_1	Hierarchical Precision	Hierarchical Recall
BERTweet-large	**0.6391**	0.5845	0.7051
DeBERTa-v3-large	0.6353	0.5902	0.6878
RoBERTa-large	0.6232	0.5852	0.6676
ALBERT-large-v2	0.5860	0.5313	0.6534
ALBERT-large-v2	0.5860	0.5313	0.6534
XLNet-base-cased	0.5760	0.5378	0.6201
XLM-RoBERTa-base	0.5647	0.5037	0.6424
BERT-base-cased	0.5577	0.5112	0.6135

Table 3: Hierarchical scores on development set of models trained on training set solely

BERTweet-large exhibited the best performance, which aligns with our expectation given its training on a sizable corpus of short text snippets resembling the linguistic style found in memes. However, after the two highest-performing models were retrained using both the training and development sets in the test phase, we opted for DeBERTa-v3-large as our final model for the pipeline. The performance of these two models is shown in Table 4.

	Hierarchical F_1	Hierarchical Precision	Hierarchical Recall
BERTweet-large	0.6806	0.6134	0.7644
DeBERTa-v3-large	**0.6918**	0.6492	0.7404

Table 4: Hierarchical scores on val set of models trained on training and development set

We supported our decision for DeBERTa with an error analysis (see Appendix B). The comparative analysis between the predictions of DeBERTa-v3-large and BERTweet-large against the gold labels of the validation set revealed that the predictions of DeBERTa-v3-large demonstrated greater robustness and closer alignment with ground truth. This observation suggests superior generalization capabilities for DeBERTa-v3-large over BERTweet-large. Our assumption regarding generalization was supported when we observed a significant improvement, advancing from 12th place in the development set to 4th place in the test set, achieving a hierarchical F1-score of 0.6738.

Throughout the development process, we explored various approaches to optimize performance. For instance, we integrated hierarchical transfer learning, following the method outlined by Banerjee et al. (Banerjee et al., 2019). This method of recursively initializing child nodes using the weights of the respective parent node marginally improved scores on the development set, but infrastructure constraints hindered its inclusion in the extended training of DeBERTa-v3-large with both training and development sets.

Additionally, we attempted ensemble learning by combining multiple models using the stacking method, although this did not yield significant performance improvements. We assume this is caused by the lack of sufficient training data for our meta-model in the ensemble pipeline.

As stated before, initially, we implemented a multilabel approach disregarding hierarchy, which yielded a respectable hierarchical F1-score of 0.6121 on the development set with DeBERTa.

To see how the traditional methods would perform on the dataset, we implemented Word2Vec (Mikolov et al., 2013) and TF-IDF (Robertson, 2004) in a multilabel classification. Unfortunately, they performed very poorly so we didn't continue

with them. We believe due to the complex semantic structure of the dataset and the limited performance of these methods, they weren't able to learn enough information.

5.2 Subtask 2B

In our final pipeline in Figure 2, we tried out many image models and different methods to represent the image and text sequences. Transformer models were almost always better in performance as can be seen in Table 2. As we were participating in a competition and eager to explore extraordinary methods, we tried out different methods for sequence representations. Those were; using CLS, averaging embeddings of all tokens and the pooler output which is the further processed CLS embedding with a layer normalization. We tested out these approaches on both the text and the image models in the pipeline at the same time, except for the CNN-based image models where we only tried out these three methods on the text transformer. Averaging the tokens gave the best development set score using the model ViT-3 from the Table 2, so we used it as such in our final pipeline. Although we were at the first place on the development set as can be seen on Table 2, we were at the 9th place on the test set. However, we would like to point out that we were still in the top 10 for this subtask.

	Avg of tokens	CLS	Pooler output
ViT-1(Dosovitskiy et al., 2021)	0.8119	0.7962	0.8074
ViT-2(Dosovitskiy et al., 2021)	0.8288	0.7751	0.7766
ViT-3(Dosovitskiy et al., 2021)	**0.8537**	0.7843	0.8052
ViT-4(Dosovitskiy et al., 2021)	0.8119	0.7801	0.8118
CLIP-patch16(Radford et al., 2021)	0.8074	0.8074	0.7997
CLIP-patch32(Radford et al., 2021)	0.7805	0.7995	0.7876
ResNet-50(He et al., 2015)	0.7366	0.7411	0.7570
ResNet-152(He et al., 2015)	0.7884	0.7572	0.7616
EfficientNet(He et al., 2015)	0.7588	0.6676	0.7518

Table 5: F_1-macro scores on development set of all image models we tried with the main pipeline using different sequence representation methods. ViT-1: Google/Vit-base-patch16-224-in21k, ViT-2: Google/vit-base-patch16-384, ViT-3: Google/vit-base-patch32-224-in21k, ViT-4: google/vit-base-patch32-384. For the EfficientNet, B5 version is used.

Furthermore, we tried out some different pipelines and compared them to our first and best pipeline in Figure 2. Firstly, we tried out ensembling text and image models with a stacking method by training them separately first and then using their predictions in a meta model. As a meta model, we tried out random forest classifier, logistic regression and 2 linear layers trained only after the base models. On the other hand, we also tried out cross attention between the text and image embeddings. Query was the text embedding, key and value were the image embedding. All these new attempts failed to obtain higher development set score than our original pipeline. Lastly, we tested the performance of our original pipeline on the new dataset whose texts were removed by blurring with a pipeline (Borella, 2021) using a Keras-OCR model. Unfortunately, removing the texts caused a slight performance decrease. The reason we believe is that in the original images, the model learned to ignore the spaces that contained texts and they seem to not contain any more valuable information about the image. When the texts were gone, there were more space to attend so a little more attention was shifted to those spaces with small information as in Figure 3. As a result, we didn't alter our dataset either. The comparison between our final pipeline and our failed attempts can be seen in table 6.

	$F_1 - macro$
ViT-3	**0.8537**
Ensemble-RF	0.8282
Ensemble-LogR	0.7778
Ensemble-LL	0.7524
CrossAttention	0.7962
ImageOnly	0.4711
TextOnly	0.6465
AlteredDataset	0.8373

Table 6: $F_1 - macro$ scores on the development set for each attempt. Ensemble-* means stacking method is applied. RF: Random forest, LogR: Logistic regression and LL: Linear layers. ImageOnly means only the same image model as ViT-3 was used for predictions and TextOnly means only BERTweet was used for predictions. AlteredDataset shows the performance of the main pipeline when it is trained on the memes dataset whose texts were removed

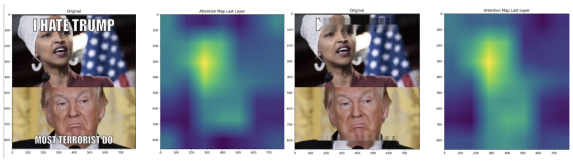

Figure 3: Attention maps on the original (with text) (Dimitrov et al., 2024) and the altered memes datasets (without text). Bright green areas represent the areas that the models attended more while dark blue spaces were interpreted as less important by them. In the altered dataset, attention was shifted a little more to the space that was used to be text.

5.3 Subtask 2A

Due to the time constraints mentioned earlier, we didn't have much time to develop multiple models and compare their performances. Instead, we trained the best-performing models from the previous two tasks. This approach followed the logic that 'since the tasks are nearly identical, with the only difference being the data in use, if we take the best-performing classifier from Subtask 1 and implement it to Subtask 2A, it should yield good results.' During training, we utilized grid search to identify the ideal learning rates, which was also the only hyperparameter we searched for, as including every new metric would increase the training time. This decision was logical as we theorized that the learning rate was the most sensitive hyperparameter, particularly considering variations in the training data. The threshold values and any other hyperparameters, on the other hand, remained the same as in Subtask 1. In the end, this approach proved to be correct, and the model's performance, both during testing and in the final results, did not suffer. As demonstrated in Table 2, the final model had a similar, if not higher, score for both the test and development datasets and exhibited the least amount of variation in final rankings, indicating a certain level of robustness.

6 Conclusion & Future Work

We successfully participated in the SemEval 2024 shared task 4 on detecting persuasion techniques in memes. Our hierarchical multilabel classification approach secured a position in the top 10 on the leaderboard across all subtasks. While detecting persuasion techniques remains challenging, the progress observed in recent years underscores its potential to mitigate the spread of propaganda campaigns.

In future work, increasing the data size could significantly improve the training of this architecture. Additionally, leveraging generative models could offer the possibility to augment data and subsequently fine-tune larger multimodal models to enhance performance and robustness.

7 Acknowledgements

We thank Daryna Dementieva for her guidance throughout our project.

References

2023-2024 Detection of Propaganda Techniques in News Articles. Ptc corpus tasks on "detection of propaganda techniques in news articles".

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical Transfer Learning for Multi-label Text Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300, Florence, Italy. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Towards Data Science Carlo Borella. 2021. Remove text from images using cv2 and keras-ocr.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

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 *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1377–1414, Barcelona (online). International Committee for Computational Linguistics.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. Fine-Grained Analysis of Propaganda in News Article. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.

Bijoyan Das and Sarit Chakraborty. 2018. An improved text sentiment classification model using tf-idf and next word negation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

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 *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 70–98, Online. Association for Computational Linguistics.

Dimitar Dimitrov, Giovanni Da San Martino, Preslav Nakov, Firoj Alam, Maram Hasanain, Abul Hasnat, and Fabrizio Silvestri. 2024. Semeval 2024 task 4 "multilingual detection of persuasion techniques in memes". https://propaganda.math.unipd.it/semeval2024task4/index.html [Accessed: (2024)].

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Zhida Feng, Jiji Tang, Jiaxiang Liu, Weichong Yin, Shikun Feng, Yu Sun, and Li Chen. 2021. Alpha at SemEval-2021 task 6: Transformer based propaganda classification. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 99–104.

Vansh Gupta and Raksha Sharma. 2021. NLPIITR at SemEval-2021 Task 6: RoBERTa Model with Data Augmentation for Persuasion Techniques Detection. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1061–1067, Online. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

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

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.

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Jakub Piskorski, Nicolas Stefanovitch, Giovanni Da San Martino, and Preslav Nakov. 2023. SemEval-2023 Task 3: Detecting the Category, the Framing, and the Persuasion Techniques in Online News in a Multi-lingual Setup. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 2343–2361, Toronto, Canada. Association for Computational Linguistics.

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.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of Varying Shades: Analyzing Language in Fake News and Political Fact-Checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.

Stephen Robertson. 2004. Understanding inverse document frequency: On theoretical arguments for idf. *Journal of Documentation - J DOC*, 60:503–520.

M. Schuster and K.K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Mingxing Tan and Quoc V. Le. 2020. Efficientnet: Rethinking model scaling for convolutional neural networks.

Junfeng Tian, Min Gui, Chenliang Li, Ming Yan, and Wenming Xiao. 2021. Mind at semeval-2021 task 6: Propaganda detection using transfer learning and multimodal fusion. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1082–1087.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020. Xlnet: Generalized autoregressive pretraining for language understanding.

A Data Analysis

A.1 Training Data in Subtask 1

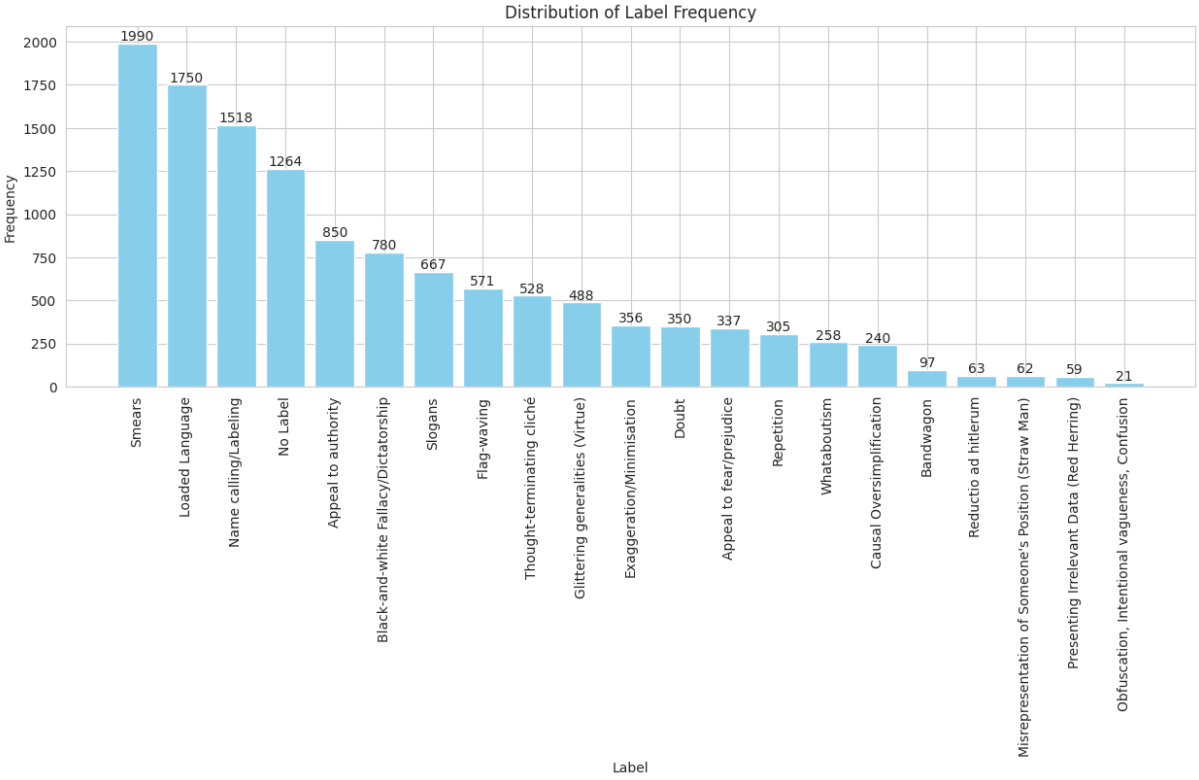

Figure 4: Label frequency

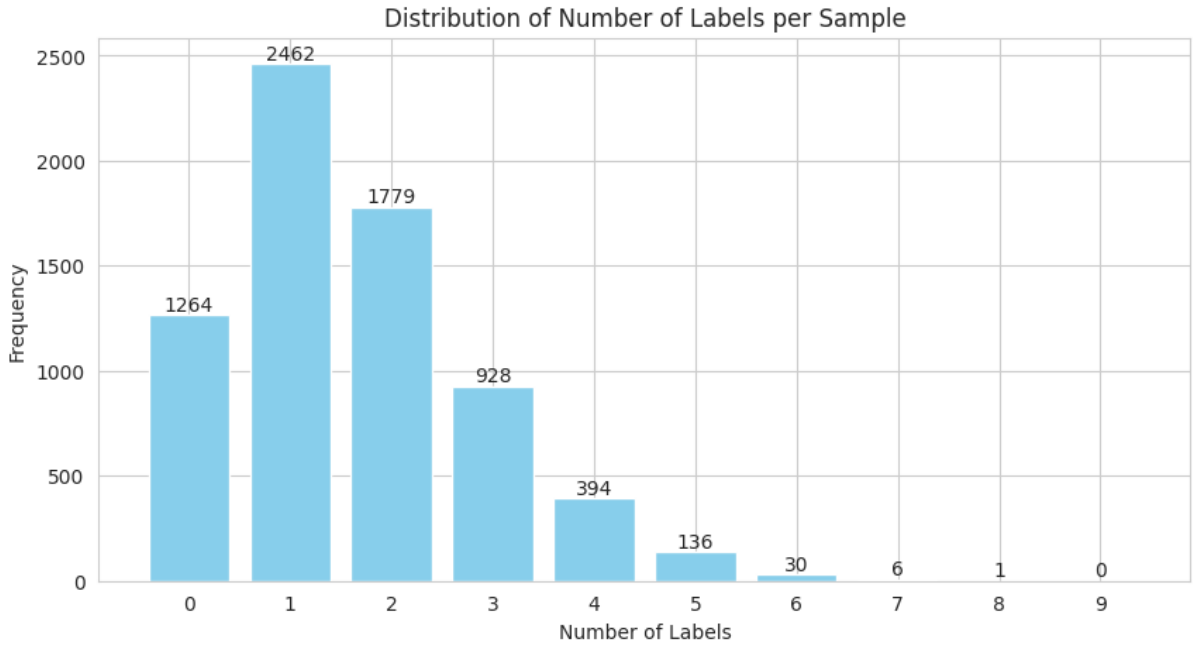

Figure 5: Number of labels per sample

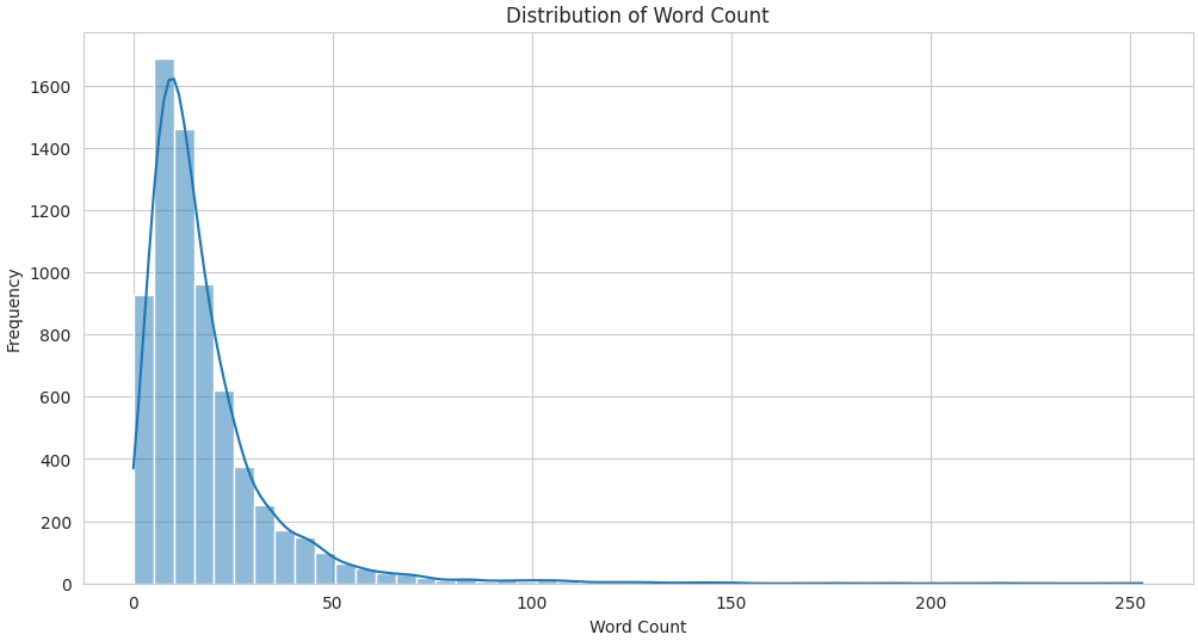

Figure 6: Distribution of word count

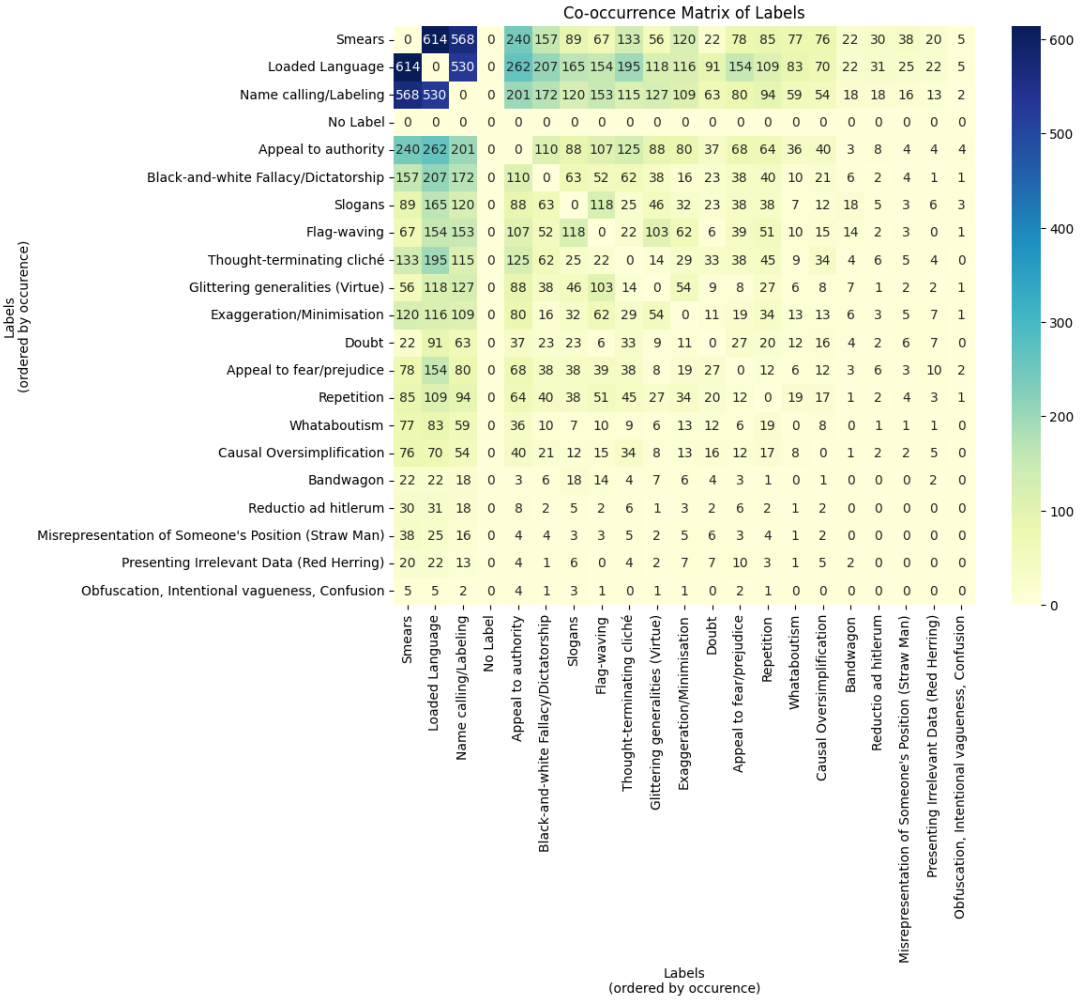

Figure 7: Co-occurence of labels

B Error Analysis

B.1 Error Analysis in Subtask 1

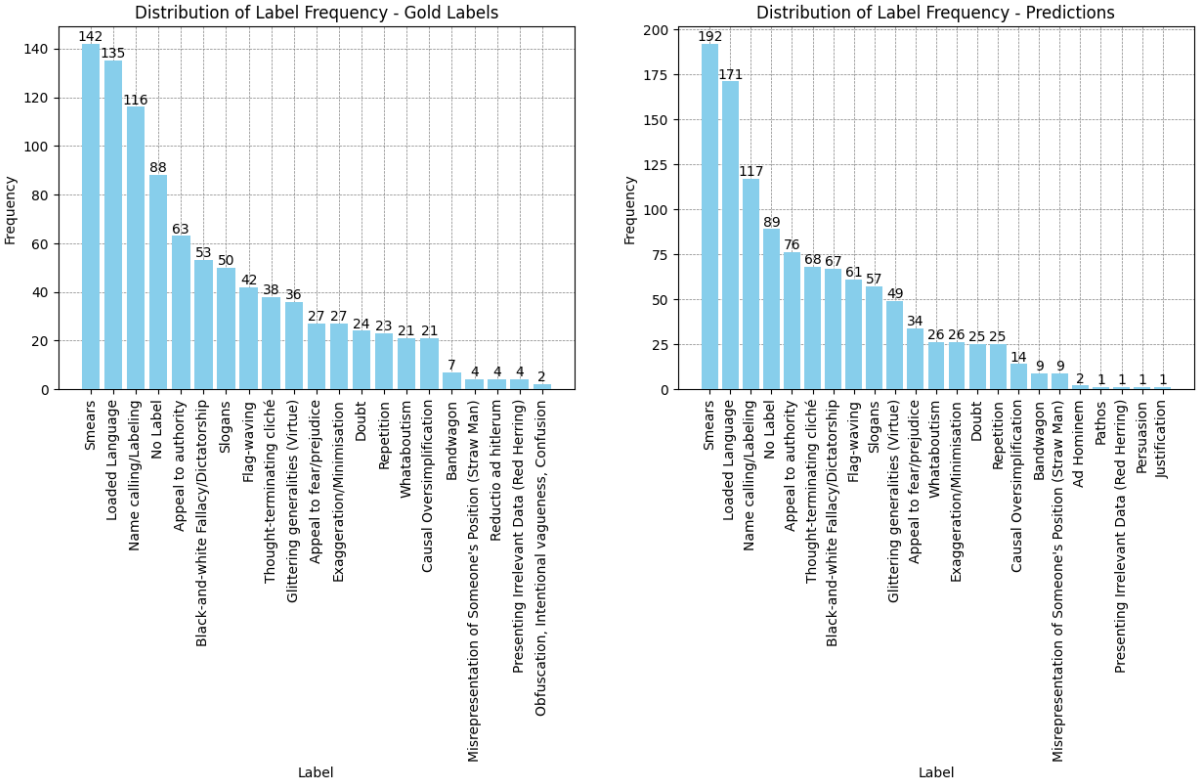

Figure 8: Label frequency - DeBERTa

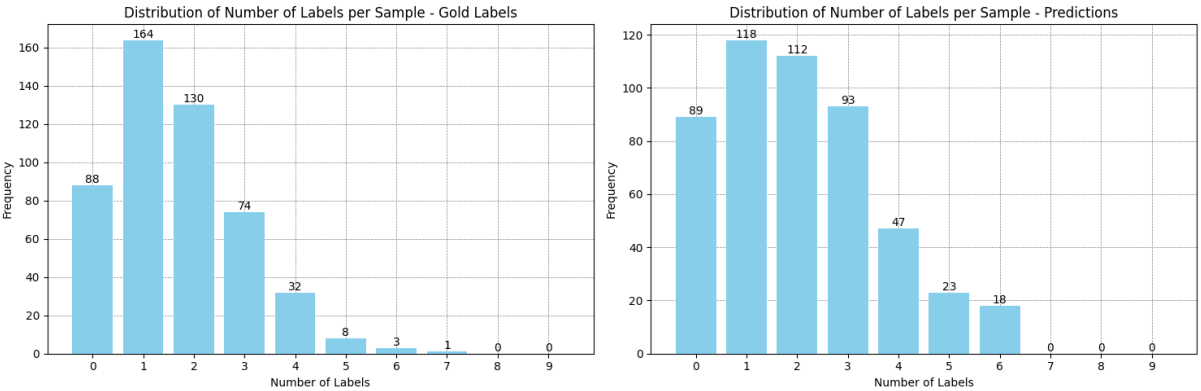

Figure 9: Number of labels per sample - DeBERTa

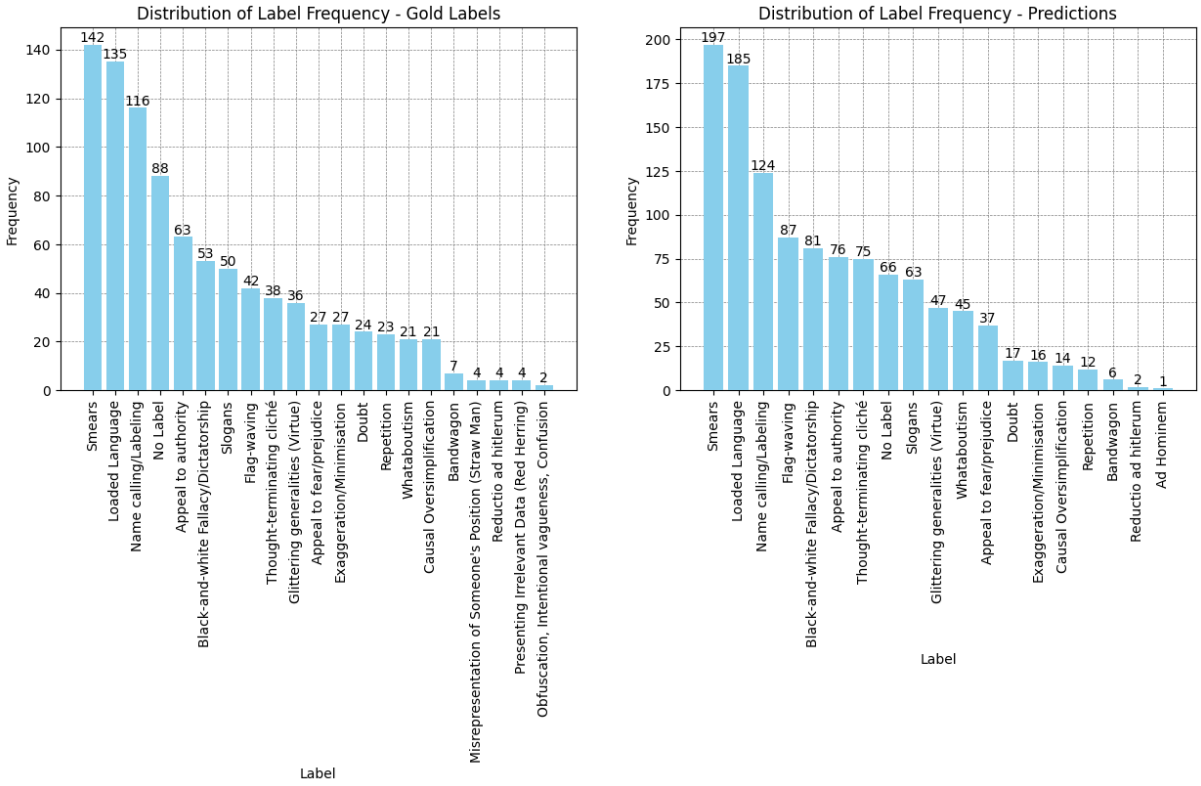

Figure 10: Label frequency - BERTweet

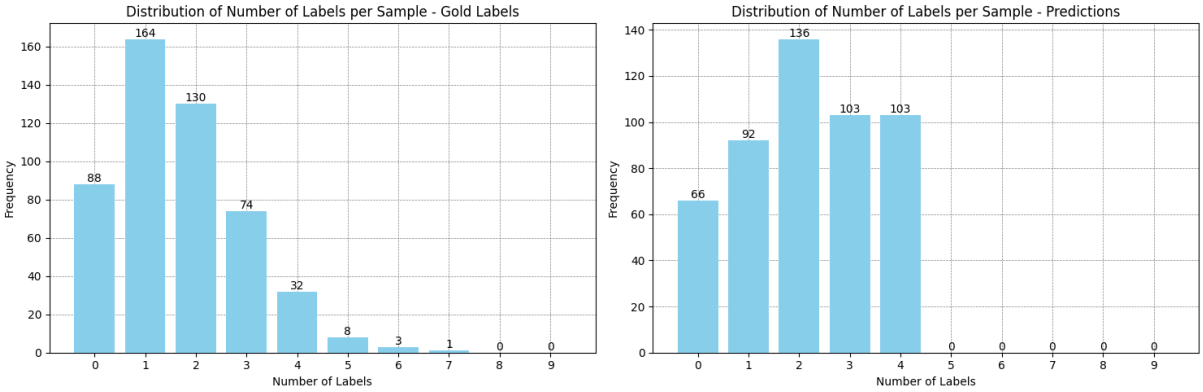

Figure 11: Number of labels per sample - BERTweet