

# OtterlyObsessedWithSemantics at SemEval-2024 Task 4: Developing a Hierarchical Multi-Label Classification Head for Large Language Models

Julia Wunderle<sup>†</sup> and Julian Schubert<sup>‡</sup> and Antonella Cacciatore<sup>‡</sup>

Albin Zehe<sup>†</sup> and Jan Pfister<sup>†</sup> and Andreas Hotho<sup>†</sup>

Center for Artificial Intelligence and Data Science (CAIDAS)

Data Science Chair, Julius-Maximilians-Universität Würzburg (JMU)

<sup>†</sup>{lastname}@informatik.uni-wuerzburg.de

<sup>‡</sup>{firstname.lastname}@informatik.uni-wuerzburg.de

## Abstract

This paper presents our approach to classifying hierarchically structured persuasion techniques used in memes for Task 4 Subtask 1 of SemEval 2024. We developed a custom classification head designed to be applied atop of a Large Language Model, reconstructing hierarchical relationships through multiple fully connected layers. This approach incorporates the decisions of foundational layers in subsequent, more fine-grained layers. To improve performance, we conducted a small hyperparameter search across various models and explored strategies for addressing uneven label distributions including weighted loss and thresholding methods. Furthermore, we extended our pre-processing to compete in the multilingual setup of the task by translating all documents into English. Finally, our system achieved third place on the English dataset and first place on the Bulgarian, North Macedonian and Arabic test datasets.


## 1 Introduction

Memes are widely used for communicating in the digital age, often laced with sarcasm and humor. However, beyond their role in everyday conversation, memes are increasingly recognized for their persuasive and manipulative potential. They hold power to subtly influence opinions, incite reactions, or shape public discourse and perception. Given this dual nature of memes as both funny communication tool and vehicle for manipulation, there arises a need to dissect and understand the persuasion techniques embedded within them. A proper understanding of this domain enhances the ability to reflect on and emotionally defend against manipulation. In this context, Large Language Models (LLMs) emerge as valuable assets in analyzing and deciphering the persuasive elements within

memes. Their automated, rapid processing capabilities make them well-suited for parsing through vast amounts of meme data, extracting patterns, and discerning underlying features. Recognizing the importance of this topic, the SemEval 2024 Task 4 Subtask 1 focuses on identifying persuasion techniques used in memes (Dimitrov et al., 2024). The aim of the first subtask is to classify the textual content from memes into various hierarchically structured persuasion techniques. In this paper, we provide a detailed description of our system including the custom classification head we designed in order to incorporate the hierarchy of the labels. Our system was able to achieve the third place on the English test dataset. Furthermore, we outperformed all other systems on the Bulgarian, North Macedonian and Arabic test sets. In summary, (i) we created a custom classification head well-suited for hierarchical settings, (ii) developed a strategy for languages where less training data is available, (iii) analyzed the influence of different hyperparameters and strategies in the context of multi-label classification problems. Our code is publicly available<sup>1</sup>.

## 2 Related Work

In the context of multi-label classification, the primary aim is to identify all relevant classes associated with a given sample. Additionally, in a hierarchical classification setting the labels are partially ordered, ranging from broader generic categories to narrowed specific instances (Kiritchenko et al., 2006). There is a large variety of approaches for this task. While earlier methods were based on tree-structures and graphs, more recent approaches rely on deep learning models (Liu et al., 2023). This section introduces various models adaptable to the task of hierarchical multi-label classification.

 These authors contributed equally to this work.

<sup>1</sup><https://github.com/LSX-UniWue/Semeval-2024-Task-4>

## 2.1 Models

Transformer models consist of an encoder and decoder which can individually be adapted for sequence classification tasks (Vaswani et al., 2017).

**Encoder-only Models** are well-suited for sequence classification. These models directly generate a representation of the input sequence, which is then passed through a classification head for prediction. As huggingface (Wolf et al., 2020) allows us to easily test different models, we compared a variety of encoder-only models. This includes different *BERT* (Devlin et al., 2019) and *RoBERTa* (Liu et al., 2019) models. As the memes in our dataset often contain hateful or toxic content, we include specifically pre-trained *BERT-base* models. While *hateBERT* (Caselli et al., 2021) is re-trained on explicitly hateful content from banned reddit communities, *bert-hateful-memes-expanded* (limjiayi, 2021) was fine-tuned on multiple datasets containing hateful memes.

**Decoder-only Models** like *LLaMA 2* can be adapted for sequence classification tasks by utilizing the logits of the last token from the input sequence (Huggingface). We evaluated the performance of the 7b and 13b parameter versions of *LLaMA 2* (Touvron et al., 2023).

## 3 Dataset

The organizers provided English datasets for training, validation and testing (7000/500/1500, train/val/test). Additionally a dev set (1000) was published to enable comparison of participating systems on a separate leaderboard ahead of the final submission on the test data. Each sample within these datasets consists of a unique id, the URL linking to the source of the meme, the transcribed text content and a list of associated labels. For example, the text: *Stay on high moral ground and we will win - Raphael Warnoc*, has the associated labels: *Appeal to authority* and *Glittering generalities (Virtue)*. The labels of the memes are structured hierarchically, with *Ethos*, *Pathos* and *Logos* in the first layer. In total, there are 28 labels with 20 persuasion techniques in the last layer, which we will refer to as *leaves*. It is important to note that the leaves are not distributed equally within the datasets. While *Smears* (1990), *Loaded Language* (1750) and *Name calling/Labeling* (1518) appear most frequently in the training data, *Presenting Irrelevant Data (Red Herring)* (59) and *Obfuscation*,

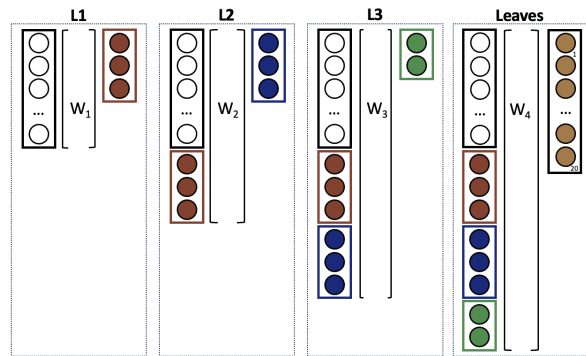


Figure 1: Illustration of our custom classification head. The depicted parts represent different layers, where **L1** corresponds to the first hierarchy layer: Ethos, Pathos, Logos. **L2** maps to the second layer, and **L3** to the third. Finally, all features are mapped to the **Leaves**. This design allows us to incorporate previously made decisions into subsequent layers. For simplicity,  $W_{1-4}$  represent fully connected layers.

*Intentional vagueness*, *Confusion* (21) occur most rarely. The final submissions were made on the English test dataset. In addition, the hosts released testing data for North Macedonian (259), Bulgarian (436) and Arabic (100) (Dimitrov et al., 2024).

## 4 System Overview

This section provides an in-depth description of our system. To integrate the hierarchical structure of the labels, we introduce a custom classification head that is designed to be applied atop various pre-trained Large Language Models.

### 4.1 Pre-Processing

We tested our system with two different pre-processing approaches: In memes, lines of text are often broken due to space limitations on the image. Therefore, we assume that most newline characters do not carry any semantic information and thus remove them in the first pre-processing variation (*cleaned*). As preliminary experiments indicated that certain LLMs might exhibit enhanced performance with all-lowercase input, the second version incorporates an additional step to convert the text to lowercase letters (*all\_lower*).

### 4.2 Custom Classification Head

The fundamental concept of our classification head entails intuitively reconstructing the hierarchy across multiple fully connected layers. As depicted in Figure 1, the basic architecture unfolds as follows: In the initial layer (**L1**), the sequence embedding provided by the backbone LLM serves as

input, producing logits for the three highest nodes of the hierarchy, namely: Ethos, Pathos and Logos. Logits for the next layer (**L2**), Ad hominem, Justification and Reasoning, are obtained by passing a concatenation of the sequence embedding and the logits of the preceding layer through another feed-forward layer. This process is repeated for the last parent nodes, Distraction and Simplification (**L3**). Finally, the logits for the 20 individual leaf nodes (**Leaves**) are obtained using another linear layer, which incorporates the concatenation of sequence embeddings and the logits of all previous layers. Accessing decisions from upper levels of the hierarchy enables logits in the fine-grained layers to be shaped by the choices made for more foundational categories. Crucially, the loss is calculated over all nodes, not solely leaf nodes, enabling the model to learn the hierarchy effectively. The head outputs logits for all 28 labels, which are then transformed into probabilities using a sigmoid function. Lastly, the probabilities are converted into labels using thresholds, where labels with a probability above the threshold are included in the final prediction. Notably, all classes in the hierarchy, including leaf nodes with low parent probabilities and vice versa, can be predicted. This design principle ensures that decisions made at higher levels serve as guidance without imposing restrictions, thereby maintaining the autonomy of lower-level decisions within the hierarchical structure.

### 4.3 Loss function

We aimed to address the unequal label distribution by testing both the standard binary cross-entropy loss as well as its weighted variant. Each class was weighted depending on their inverse frequency, assigning a higher penalty to misclassifications of minority classes, with the goal of enhancing the performance of these less represented classes.

### 4.4 Ensemble

To further enhance the robustness of our predictions we employed an ensemble approach where we utilize majority voting across four different models. Each of these models was trained with the same hyperparameters but with different random seeds. As described above (Section 4.2) our model outputs labels for each sample. To combine the suggestions of multiple models, we experimented with various boundary levels to determine the number of model predictions needed for a label to be included in the final prediction. Our experiments revealed that re-

quiring at least two of the four models to vote for a label is the most effective.

### 4.5 Handling Different Input Languages

To extend the applicability of our system for the multilingual setting, we integrated an additional pre-processing step. The provided non-English test datasets were translated into English using GPT-4 (OpenAI et al., 2023), using the following prompt: *You are a bilingual humorist, adept at translating meme text between languages while preserving the original humor, cultural nuances, and any slang or idiomatic expressions. Ensure the translation is accurate, contextually appropriate, and retains the meme’s playful tone. Avoid adding explanations or additional commentary and provide only the translation.*

## 5 Experimental Setup

In order to approximate optimal parameters for the LLaMA 2 models, we conducted a grid-search for various BERT and RoBERTa models as these require less computational resources. During training, we utilized gradient accumulation to reach a gradient update every 128 samples. All models were trained for ten epochs, with a learning rate of either  $5 \times 10^{-4}$  or  $5 \times 10^{-5}$  and the Adam optimizer (Kingma and Ba, 2014). We further included the two different pre-processing styles as well as the binary-cross entropy loss and its weighted variation as hyperparameters. Lastly, we performed all experiments with and without our custom classification head. Training was conducted on either NVIDIA GeForce RTX 4090 or NVIDIA A100 GPUs. Due to the large size of the LLaMA 2 models, we used Low-Rank-Adaptation to greatly reduce the number of trainable parameters for this model-family (Hu et al., 2021). We used the provided training dataset for training and the validation dataset to test generalization capabilities after each epoch. In the final stage, we assessed our system’s performance on the dev dataset, utilizing a hierarchical version of the F1-score metric (hF1) following (Kiritchenko et al., 2006). The full set of hyperparameters we used is shown in Table 3.

### 5.1 Determining Optimal Thresholds

For each sample, our model outputs one logit for each class. Thus, we need to decide on a threshold, determining the decision boundary for assigning labels to instances based on their predicted probabilities. As the commonly used threshold of 0.5

Table 1: Comparison of hF1-scores and averages across all languages of our system and other top-performing systems. The corresponding ranks are provided in brackets.

System	en	bg	md	ar	Avg
Ours	0.697 (3)	<b>0.568 (1)</b>	<b>0.512 (1)</b>	<b>0.476 (1)</b>	<b>0.563 (1.5)</b>
NLPNCHU	0.663 (6)	0.517 (3)	0.462 (5)	0.475 (2)	0.529 (4.0)
914isthebest <sup>2</sup>	<b>0.752 (1)</b>	0.463 (11)	0.369 (14)	0.360 (13)	0.486 (9.75)
BCAmirs <sup>3</sup>	0.699 (2)	0.448 (13)	0.393 (12)	0.396 (9)	0.484 (9.0)

appeared non-optimal for our task based on preliminary testing, we implemented different strategies aiming to find optimal thresholds on the validation dataset. To systematically find the best threshold, we predetermined a spectrum of threshold levels to investigate. We experimented with (i) picking the same global threshold for all classes, and (ii) optimizing the threshold for each class individually. We computed the accuracy and F1-score for each threshold-label combination and selected the best outcomes for both metrics respectively. For both variants, we output all classes with probabilities above the threshold as well as all parents of the selected nodes.

## 6 Results

This paragraph discusses the influence of various hyperparameters, our ranking on the leaderboard and provides a detailed error analysis.

### 6.1 Influence of Hyperparameters

We tested the influence of both pre-processing styles, the two variants of the loss calculation, different learning rates and the custom classification head we designed. As shown in Table 5, the pre-processing variant has a negligible impact, with all models performing almost identical for both *cleaned* and *all\_lower* data. Surprisingly, all models perform worse when weighting classes based on their inverse frequency in the binary cross-entropy loss. A possible reason for this is the high imbalance of our dataset (see Section 3). Weighted loss prioritizes minimizing the loss for minority classes, potentially compromising accuracy for majority classes, leading to sub-optimal overall results. The addition of our custom classification head improves our results up to eleven percent points and two percent points on average. Strikingly, *bert-large-cased* performs the worst and

models pre-trained on hateful content can outperform their foundation counterparts. While *bert-hateful-memes-expanded* achieves even better results than models with a higher parameter count, *hateBERT* performs worse than the BERT-base model. Lastly,  $5 \times 10^{-5}$  was the best learning rate for all models tested in the grid search. Nevertheless, first experiments with LLaMA 2 revealed, that a learning rate of  $5 \times 10^{-4}$  works better for this model family. Using these findings, we decided to train a LLaMA 2 13b model using the *all\_lower* pre-processing style with our custom classification head, a learning rate of  $5 \times 10^{-4}$  and no weighted loss. The LLaMA 2 models outperform the other models with the chosen parameter selection. We further observed that global thresholds consistently yielded superior performance compared to selecting single thresholds for each class. The optimal thresholds of our experiments range between 0.2 and 0.4 and vary depending on the base model and other parameters. We assume that the inferior performance of individual thresholds stems from our methodology of including all ancestors of a predicted leaf in the output, regardless of their assigned probabilities. As a result, inaccuracies at the lowest hierarchy level disproportionately affect our system’s precision due to the compounded errors in ancestor predictions.

### 6.2 Main Results

A total of 33 teams competed in the subtask. Table 1 compares our system against other top-performing systems across all evaluated languages using the official test results. Our framework consistently ranks among the top three across all languages, securing the top position for Bulgarian (bg), North Macedonian (md) and Arabic (ar) datasets. It achieves the highest average hierarchical F1-score and the highest average leaderboard ranking. This demonstrates the versatility of our approach, underlining our methodology’s effectiveness and adaptability to non-English languages. Table 2 presents

<sup>3</sup>(Dailin Li and Lin, 2024)

<sup>3</sup>(Amirhossein Abaskohi and Carenini, 2024)



hierarchical performance on the dev dataset for our four distinct models trained using varied seeds, in addition to their ensemble which was used for the final submission.

The highest performing individual model records a hF1 of 0.682, while the ensemble demonstrates an enhanced score of 0.690. This indicates that leveraging the outputs from multiple independently trained models can lead to improved results. Despite similar hF1 scores across models, variations of up to four and seven percent points in hierarchical precision (hP) and hierarchical recall (hR) respectively suggest differing error patterns and strengths among the models. This disparity highlights the efficacy of our ensemble approach, showcasing its capacity to amalgamate diverse insights from the dataset.

### 6.3 Error Analysis

In this chapter, we will dive deeper into the shortcomings of our system regarding the performance of our ensemble model on the dev dataset (Figure 2). Overall, the distribution of labels predicted by our system closely aligns with the ground truth. However, our system exhibits a bias towards predicting classes with a larger number of samples, leading to a higher frequency of these labels in our output. Conversely, labels with fewer occurrences in the training data are underrepresented in our predictions, leading to lower F1-scores in comparison. Some leaves with very few training samples such as *Presenting Irrelevant Data (Red Herring)* (59) and *Obfuscation, Intentional vagueness, Confusion* (21) are never predicted by our system, leading to a F1-score of zero. Interestingly, despite *Appeal to authority* only occurring very rarely in the training data (850), our system achieves an F1-score of 0.892 in this class. This label describes that a claim is being stated as true simply because a valid authority or expert on the issue said it was true, without any other supporting evidence offered (Dimitrov et al., 2024). We therefore assume the label to be easier to predict than other classes, as the occurrence of certain authorities or names in particular at the end of a sentence are a strong indicator for this persuasion technique. It is noticeable, that our model is able to differentiate well at the first hierarchy level: Ethos, Pathos and Logos, achieve F1-scores of over 60%. Similar observations can be made for the non-English test datasets (Figure 3, Figure 4, Figure 5).

Table 2: Hierarchical results on the dev dataset for our four distinct models trained using various seeds and the ensemble of these four models.

System	hP	hR	hF1
1	0.623	0.745	0.679
2	0.661	0.673	0.667
3	0.643	0.698	0.669
4	0.631	0.740	0.682
Ensemble	0.636	0.754	0.690

## 7 Conclusion

In this paper, we introduced a robust system to classify hierarchically structured persuasion techniques in a meme-corpus for the SemEval challenge 2024 Task 4 Subtask 1. Our system achieved a top-three ranking for each language individually and outperforms every other system averaged over all languages. A key aspect of our approach is the incorporation of the label hierarchy using a custom classification head that models the individual layers of the hierarchy. This classification head can be used atop of different LLMs and improves the performance by up to 11 percent points. We employed a grid-search across various models and hyperparameters to approximate optimal parameters for a LLaMA 2 13b model that then produces the embedding for the classification head. Interestingly, weighting the loss to increase the influence of classes with fewer samples did not improve the overall performance. In addition, picking the same classification threshold for each class worked better than searching one for each label individually.

There are multiple possibilities to build upon the success of our system: First, the organizers suggested similar data sources that could be used for pre-training. Additionally, upgrading to a bigger LLM, such as LLaMA 2 70b, known for its superior performance over smaller LLaMA 2 variants, could further elevate our system’s capabilities. Moreover, extending our hyperparameter tuning could uncover better model configurations. Our methodology for parent-node selection could be refined by discarding parent nodes selected by children if the ancestor itself has low confidence. Lastly, feature stacking could be used to create a powerful model that incorporates features generated by other models in its classification head.

## Acknowledgements

This work is partially supported by the MOTIV research project funded by the Bavarian Research Institute for Digital Transformation (bidt), an institute of the Bavarian Academy of Sciences and Humanities. Additional resources, were provided by denkbares GmbH. The authors are responsible for the content of this publication.

## References

- Lele Wang Amirhossein Abaskohi, Amirhossein Dabiriaghdam and Giuseppe Carenini. 2024. Bcamirs at semeval-2024 task 4: From visuals to word: A multimodal and multilingual exploration of persuasion in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval 2024*, Mexico City, Mexico.
- Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. [HateBERT: Retraining BERT for abusive language detection in English](#). In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.
- Xin Zou Junlong Wang Peng Chen Jian Wang Liang Yang Dailin Li, Chuhan Wang and Hongfei Lin. 2024. 914isthebest at semeval-2024 task 4: Cot-based data augmentation strategy for persuasion techniques detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval 2024*, Mexico City, Mexico.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 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 *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval 2024*, Mexico City, Mexico.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *CoRR*, abs/2106.09685.
- Huggingface. [LlamaForSequenceClassification](#). Accessed 2024-01-26.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Svetlana Kiritchenko, Stan Matwin, Richard Nock, and A. Fazel Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Advances in Artificial Intelligence*, pages 395–406, Berlin, Heidelberg. Springer Berlin Heidelberg.
- limjiayi. 2021. [limjiayi/bert-hateful-memes-expanded](#). Accessed 2024-02-11.
- Rundong Liu, Wenhan Liang, Weijun Luo, Yuxiang Song, He Zhang, Ruohua Xu, Yunfeng Li, and Ming Liu. 2023. [Recent advances in hierarchical multi-label text classification: A survey](#).
- 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](#).
- OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madeleine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor

Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. [Gpt-4 technical report](#).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Ro-

driguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

## A Hyperparameters

Table 3: Search space for hyperparameter optimization.

Parameter	Values
Model	bert-base-cased, bert-base-uncased, hateBERT, bert-hateful-memes-expanded, bert-large-cased, bert-large-uncased, xlm-roberta-base, xlm-roberta-large, llama-2-7b, llama-2-13b
Batch Size	128
Epochs	10
LR	$5 \times 10^{-5}$ , $5 \times 10^{-4}$
Style	all_lower, cleaned
Weight Loss	True, False
Custom Head	True, False

## B Grid Search results

Table 4: Results of a grid-search on the dev dataset for BERT and RoBERTa models across all hyperparameters.

Model	LR	Style	Weight Loss	Custom Head	hP	hR	hF1
bert-large-cased	$5 \times 10^{-5}$	cleaned	True	True	0.429	0.718	0.537
bert-large-cased	$5 \times 10^{-5}$	cleaned	True	False	0.488	0.651	0.558
bert-large-cased	$5 \times 10^{-5}$	all_lower	True	True	0.436	0.705	0.539
bert-large-cased	$5 \times 10^{-5}$	all_lower	True	False	0.450	0.722	0.554
bert-large-cased	$5 \times 10^{-5}$	cleaned	False	True	0.600	0.612	0.606
bert-large-cased	$5 \times 10^{-5}$	cleaned	False	False	0.540	0.638	0.585
bert-large-cased	$5 \times 10^{-5}$	all_lower	False	True	0.589	0.614	0.601
bert-large-cased	$5 \times 10^{-5}$	all_lower	False	False	0.544	0.689	<b>0.608</b>
hateBERT	$5 \times 10^{-5}$	cleaned	True	True	0.423	0.742	0.539
hateBERT	$5 \times 10^{-5}$	cleaned	True	False	0.469	0.634	0.539
hateBERT	$5 \times 10^{-5}$	all_lower	True	True	0.477	0.651	0.550
hateBERT	$5 \times 10^{-5}$	all_lower	True	False	0.420	0.732	0.534
hateBERT	$5 \times 10^{-5}$	cleaned	False	True	0.572	0.651	<b>0.609</b>
hateBERT	$5 \times 10^{-5}$	cleaned	False	False	0.551	0.661	0.601
hateBERT	$5 \times 10^{-5}$	all_lower	False	True	0.549	0.669	0.603
hateBERT	$5 \times 10^{-5}$	all_lower	False	False	0.553	0.661	0.602
bert-base-cased	$5 \times 10^{-5}$	cleaned	True	True	0.449	0.717	0.552
bert-base-cased	$5 \times 10^{-5}$	cleaned	True	False	0.477	0.624	0.541
bert-base-cased	$5 \times 10^{-5}$	all_lower	True	True	0.478	0.691	0.565
bert-base-cased	$5 \times 10^{-5}$	all_lower	True	False	0.483	0.614	0.541
bert-base-cased	$5 \times 10^{-5}$	cleaned	False	True	0.520	0.693	0.594
bert-base-cased	$5 \times 10^{-5}$	cleaned	False	False	0.510	0.654	0.573
bert-base-cased	$5 \times 10^{-5}$	all_lower	False	True	0.567	0.665	<b>0.612</b>
bert-base-cased	$5 \times 10^{-5}$	all_lower	False	False	0.533	0.674	0.595
bert-base-uncased	$5 \times 10^{-5}$	cleaned	True	True	0.458	0.723	0.561
bert-base-uncased	$5 \times 10^{-5}$	cleaned	True	False	0.417	0.737	0.532
bert-base-uncased	$5 \times 10^{-5}$	all_lower	True	True	0.490	0.664	0.564
bert-base-uncased	$5 \times 10^{-5}$	all_lower	True	False	0.426	0.721	0.535
bert-base-uncased	$5 \times 10^{-5}$	cleaned	False	True	0.579	0.659	<b>0.616</b>
bert-base-uncased	$5 \times 10^{-5}$	cleaned	False	False	0.549	0.633	0.588
bert-base-uncased	$5 \times 10^{-5}$	all_lower	False	True	0.571	0.662	0.613
bert-base-uncased	$5 \times 10^{-5}$	all_lower	False	False	0.551	0.662	0.601



Table 5: Results of a grid-search on the dev dataset for BERT and RoBERTa models across all hyperparameters. Additionally, the outcomes of LLaMA 2-models for the approximated best configurations are shown.

Model	LR	Style	Weight Loss	Custom Head	hP	hR	hF1
xlm-roberta-base	$5 \times 10^{-5}$	cleaned	True	True	0.455	0.715	0.556
xlm-roberta-base	$5 \times 10^{-5}$	cleaned	True	False	0.461	0.659	0.543
xlm-roberta-base	$5 \times 10^{-5}$	all_lower	True	True	0.449	0.707	0.550
xlm-roberta-base	$5 \times 10^{-5}$	all_lower	True	False	0.446	0.652	0.530
xlm-roberta-base	$5 \times 10^{-5}$	cleaned	False	True	0.561	0.688	<b>0.618</b>
xlm-roberta-base	$5 \times 10^{-5}$	cleaned	False	False	0.507	0.647	0.568
xlm-roberta-base	$5 \times 10^{-5}$	all_lower	False	True	0.598	0.633	0.616
xlm-roberta-base	$5 \times 10^{-5}$	all_lower	False	False	0.495	0.650	0.562
bert-large-uncased	$5 \times 10^{-5}$	cleaned	True	True	0.512	0.692	0.589
bert-large-uncased	$5 \times 10^{-5}$	cleaned	True	False	0.480	0.676	0.561
bert-large-uncased	$5 \times 10^{-5}$	all_lower	True	True	0.508	0.723	0.596
bert-large-uncased	$5 \times 10^{-5}$	all_lower	True	False	0.479	0.643	0.594
bert-large-uncased	$5 \times 10^{-5}$	cleaned	False	True	0.578	0.692	0.630
bert-large-uncased	$5 \times 10^{-5}$	cleaned	False	False	0.412	0.686	0.515
bert-large-uncased	$5 \times 10^{-5}$	all_lower	False	True	0.608	0.654	<b>0.630</b>
bert-large-uncased	$5 \times 10^{-5}$	all_lower	False	False	0.594	0.621	0.607
bert-hateful-memes-expanded	$5 \times 10^{-5}$	cleaned	True	True	0.494	0.673	0.570
bert-hateful-memes-expanded	$5 \times 10^{-5}$	cleaned	True	False	0.472	0.638	0.542
bert-hateful-memes-expanded	$5 \times 10^{-5}$	all_lower	True	True	0.502	0.666	0.573
bert-hateful-memes-expanded	$5 \times 10^{-5}$	all_lower	True	False	0.473	0.643	0.545
bert-hateful-memes-expanded	$5 \times 10^{-5}$	cleaned	False	True	0.591	0.679	<b>0.632</b>
bert-hateful-memes-expanded	$5 \times 10^{-5}$	cleaned	False	False	0.564	0.657	0.607
bert-hateful-memes-expanded	$5 \times 10^{-5}$	all_lower	False	True	0.601	0.660	0.629
bert-hateful-memes-expanded	$5 \times 10^{-5}$	all_lower	False	False	0.562	0.664	0.609
xlm-roberta-large	$5 \times 10^{-5}$	cleaned	True	True	0.480	0.662	0.557
xlm-roberta-large	$5 \times 10^{-5}$	cleaned	True	False	0.499	0.662	0.569
xlm-roberta-large	$5 \times 10^{-5}$	all_lower	True	True	0.514	0.623	0.564
xlm-roberta-large	$5 \times 10^{-5}$	all_lower	True	False	0.494	0.638	0.557
xlm-roberta-large	$5 \times 10^{-5}$	cleaned	False	True	0.662	0.639	0.650
xlm-roberta-large	$5 \times 10^{-5}$	cleaned	False	False	0.574	0.673	0.619
xlm-roberta-large	$5 \times 10^{-5}$	all_lower	False	True	0.631	0.697	<b>0.662</b>
xlm-roberta-large	$5 \times 10^{-5}$	all_lower	False	False	0.581	0.688	0.630
llama7b	$5 \times 10^{-4}$	all_lower	False	True	0.648	0.684	<b>0.666</b>
llama13b	$5 \times 10^{-4}$	all_lower	False	True	0.623	0.745	<b>0.679</b>

### C Label distribution and F1

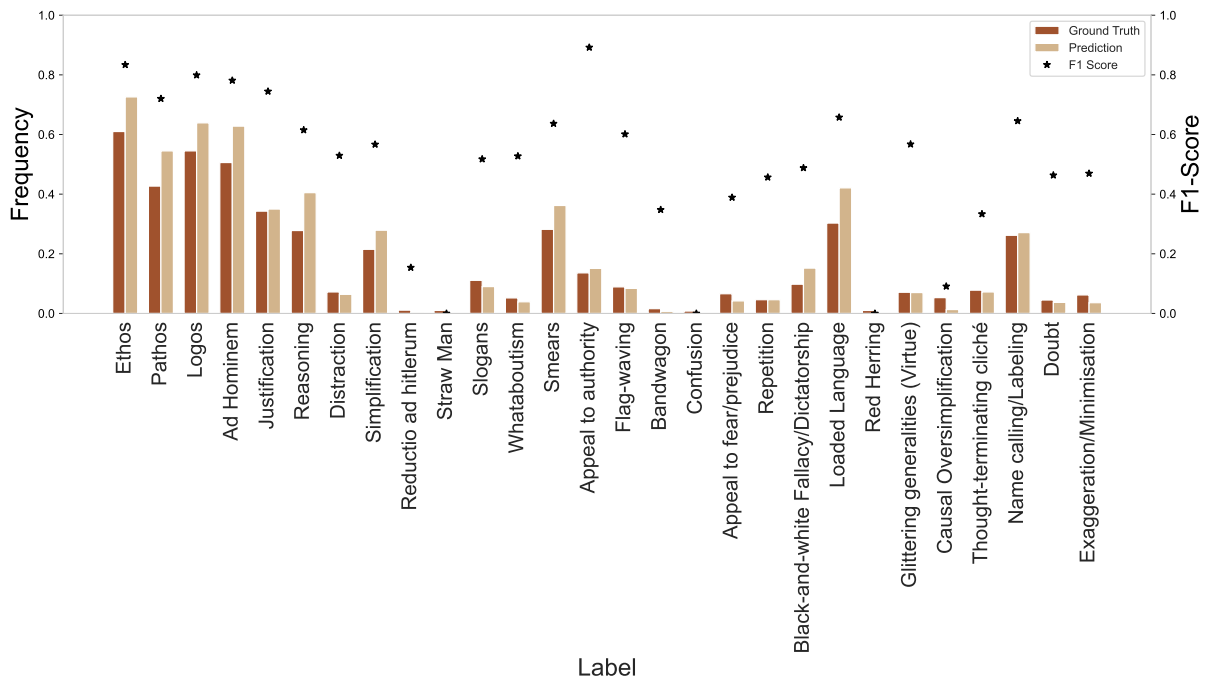


Figure 2: Distribution of labels in the **English** dev set and our system's predictions, normalized by the number of samples. The star (★) indicates the F1-Score of our system for the given label.

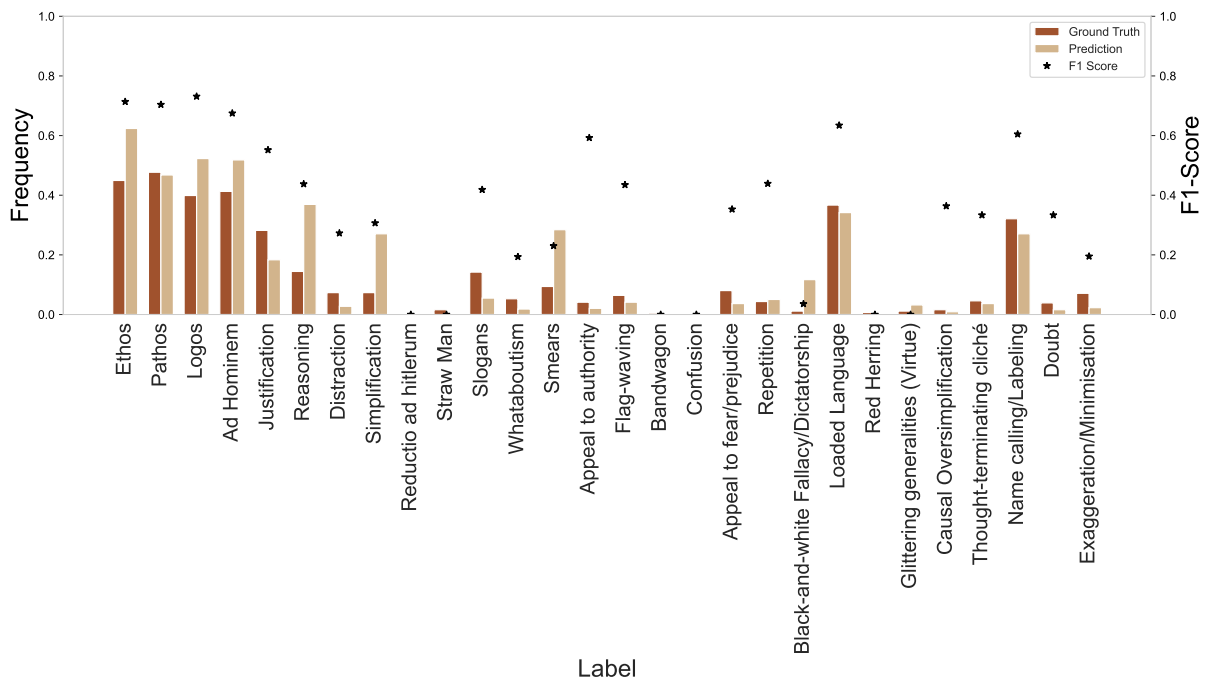


Figure 3: Distribution of labels in the **Bulgarian** test set and our system's predictions, normalized by the number of samples. The star (★) indicates the F1-Score of our system for the given label.

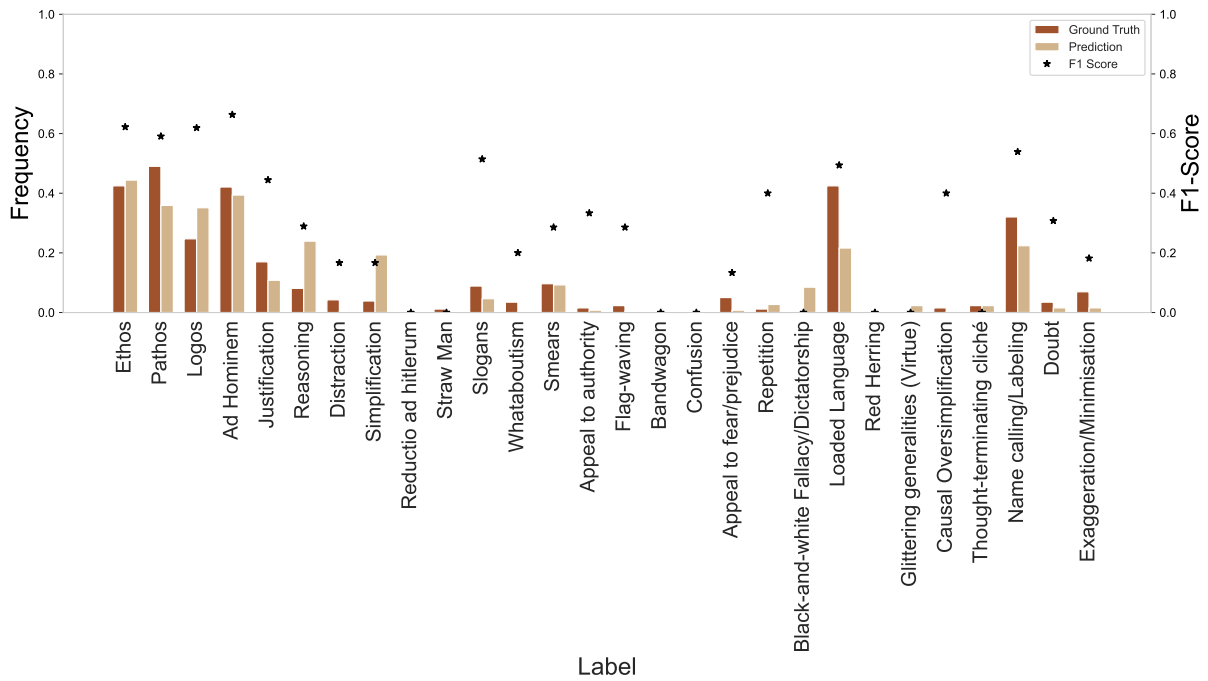


Figure 4: Distribution of labels in the **North Macedonian** test set and our system’s predictions, normalized by the number of samples. The star (★) indicates the F1-Score of our system for the given label.

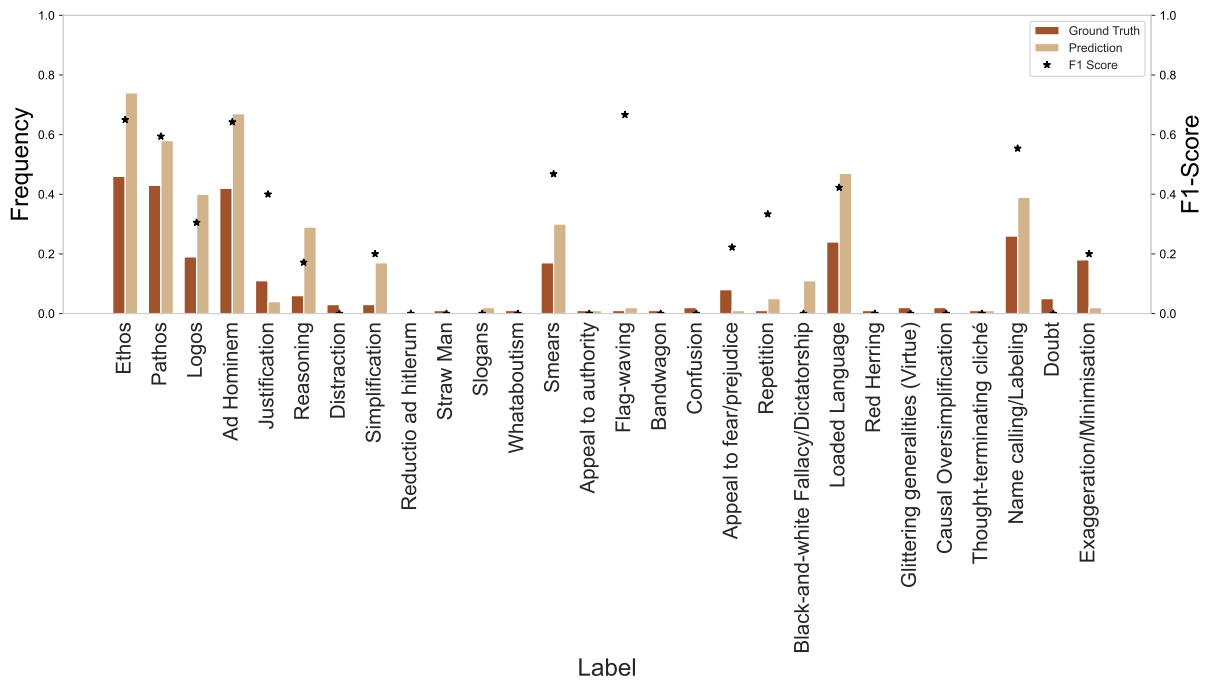


Figure 5: Distribution of labels in the **Arabic** test set and our system’s predictions, normalized by the number of samples. The star (★) indicates the F1-Score of our system for the given label.