FICODE at GermEval 2024 GerMS-Detect closed ST1 & ST2: Ensemble- and Transformer-Based Detection of Sexism and Misogyny in German Texts

Falk Maoro and **Michaela Geierhos**
University of the Bundeswehr Munich, Research Institute CODE
Werner-Heisenberg-Weg 39, Neubiberg, Germany
falk.maoro@unibw.de
michaela.geierhos@unibw.de

Abstract

In this paper, we present our solution for the shared task of GermEval 2024 GerMS-Detect. The joint task consists of two subtasks that we address in our solution. The texts in question may contain instances of sexism or misogyny and have been annotated in a multi-class classification setting. From this setting, two tasks are derived that require different binary or multi-class classifications. We propose an ensemble method using multiple sequence classification models that can be applied to both subtasks. With respect to **Subtask 1**, our approach achieves an average F1 score of **0.641**, and with respect to **Subtask 2**, our approach achieves an average Jensen-Shannon divergence of **0.354**. The code is available at the following link: https://github.com/fmaoro/germeval24

1 Introduction

The prevalence of sexism and misogyny in social media is a major concern. In order to address this issue, the GermEval 2024 GerMS-Detect shared task presents two subtasks on the identification of such misbehavior in German-language forum posts. We, the team *ficode*, propose a solution for the closed Subtask 1 and another solution for the closed Subtask 2. The shared task of GermEval 2024 GerMS-Detect provides German forum posts that have been annotated by multiple annotators to indicate the presence and strength of sexism and misogyny. Since there are multiple annotations per instance, the shared task focuses on predicting the distribution and further combined labels of the annotations. All required labels in both subtasks can be interpreted as sequence classification tasks.

Significant progress has been made in the area of language modeling tasks, such as sequence classification, with the advent of the transformer architecture proposed by Vaswani et al. (2017). In particular, Devlin et al. (2019) invented the Bidirectional Encoder Representations from Transformers (BERT), which represents an input sequence as an encoding that can be used to train multiple language modeling tasks. Since the BERT model was primarily trained on English data, the need for a German-specific BERT-like model was solved by GBERT (Chan et al., 2020). A powerful approach is needed to apply such BERT models because the tasks require a large number of predicted labels, and the classification of text into levels of sexism and misogyny is a rather complex task.

Ensemble learning, which integrates multiple models to achieve superior performance, is a robust approach to solving such complex machine learning tasks. Mohammed and Kora (2023) highlight the success of ensemble methods in various domains and their enhancement by deep learning models, despite the complexity of tuning such models. Kotary et al. (2023) introduce differentiable model selection, which optimizes ensemble composition by selecting the best performing models, thus overcoming the limitations of traditional methods. In addition, Wood et al. (2024) provide a unified theory of how model diversity reduces bias and variance, further improving ensemble performance. These works influence our approach to solving the two subtasks by highlighting the power of ensemble methods.

Therefore, we decided to use the pre-trained GBERT-large model as a baseline for fine-tuning with the available training data. The inherent German language knowledge of the model is advantageous for learning the nuances of sexism and misogyny in German texts. By training a total of six GBERT-based sequence classifiers and using them in an ensemble pipeline, we achieve an average F1 score of **0.641** for Subtask 1 and an average Jensen-Shannon divergence of **0.354** for Subtask 2.

The paper is organized as follows: Section 2 presents a brief analysis of the available training and test data. This includes an examination of the available labels and the distribution of annotations. Section 3 outlines the initial training approach. This serves as the basis for the predictions for the two subtasks. There is also a description of the methodology for using the models in an ensemble pipeline, followed by a description of the experimental setup in Section 4. An analysis of the results is presented in Section 5. Finally, Section 6 provides a concluding remark.

2 Data

The data consists of news forum posts labeled by multiple annotators. In the training subset, each post is assigned a unique ID, the text of the post itself, and a list of annotations. Each annotation contains a user pseudonym and one of the following labels: *0-Kein* (no sexism/misogyny), *1-Gering* (low sexism/misogyny), *2-Vorhanden* (present sexism/misogyny), *3-Stark* (strong sexism/misogyny), and *4-Extrem* (extreme sexism/misogyny).

In the test subset, each post is identified by a unique ID, accompanied by the text of the post and a list of the pseudonyms of the annotators who labeled the post. The training subset consists of 5,998 examples, while the test subset contains 1,986 examples. The number of annotations per example varies widely, ranging from 4 to 11 annotations. The average is 4.8 annotations per example.

Since there are multiple annotations per example, Subtask 1 defines a set of aggregating labels that need to be predicted. The first label is *bin_maj*, which is a boolean indicating that the majority of annotators assigned a label other than *0-Kein*. The label *bin_one* is also a boolean indicating that at least one annotator assigned a label other than *0-Kein*. The third binary label, *bin_all*, indicates that all annotators have assigned labels other than *0-Kein*. The only multi-class label is *multi_maj*, where the most common annotated label should be predicted. *disagree_bin* indicates if there is unanimous agreement on *0-Kein*.

The distribution of labels in Figure 1 and the distribution of labels for the class *multi_maj* in Figure 2 show a notable imbalance in all class labels, except for *bin_one*. Of particular interest is the low number of true *bin_all* labels compared to the number of positive labels. Furthermore, over 70 % of the annotated labels are *0-Kein*.

The analysis of the text data does not reveal any specific, conspicuous features. Table 1 shows the minimum, maximum and average number of characters, words, and tokens (tokenized with a *deepset/gbert-base* tokenizer) for training subset examples. In addition, an examination of random samples revealed no need for preprocessing the input texts. Therefore, in our further work we use the input texts in their original form.

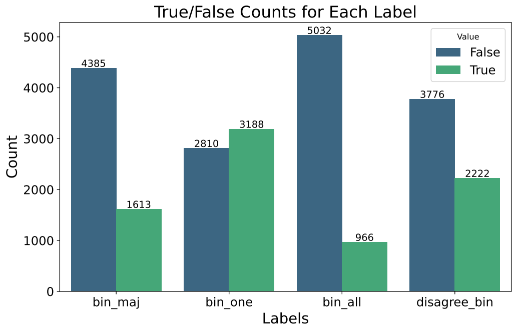

Figure 1: Label distribution for all binary labels.

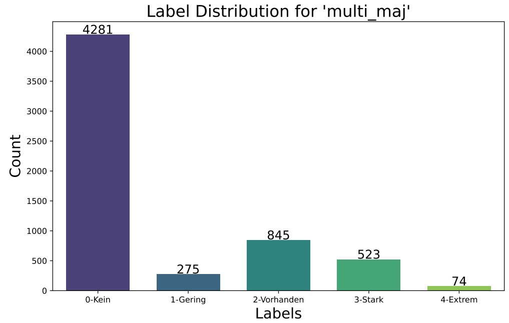

Figure 2: Label distribution for the class ‘multi_maj’.

Variable	Minimum	Maximum	Mean
Characters	3	999	216.35
Words	1	173	32.87
Tokens	3	234	50.70

Table 1: Number of characters, words, and tokens for all training examples.

3 Concept

Our approach to solve the closed Subtask 1 and the closed Subtask 2 of the GermEval 2024 GerMS-Detect shared task is to train multiple BERT models for sequence classification. The models are first trained and then used for both tasks by post-processing their outputs in different ways. The models trained for the following subtasks are described in detail in Section 3.1. Then, in Section 3.2, we present the pipeline we used to solve Subtask 1. Finally, in Section 3.3, we describe our approach to solving Subtask 2.

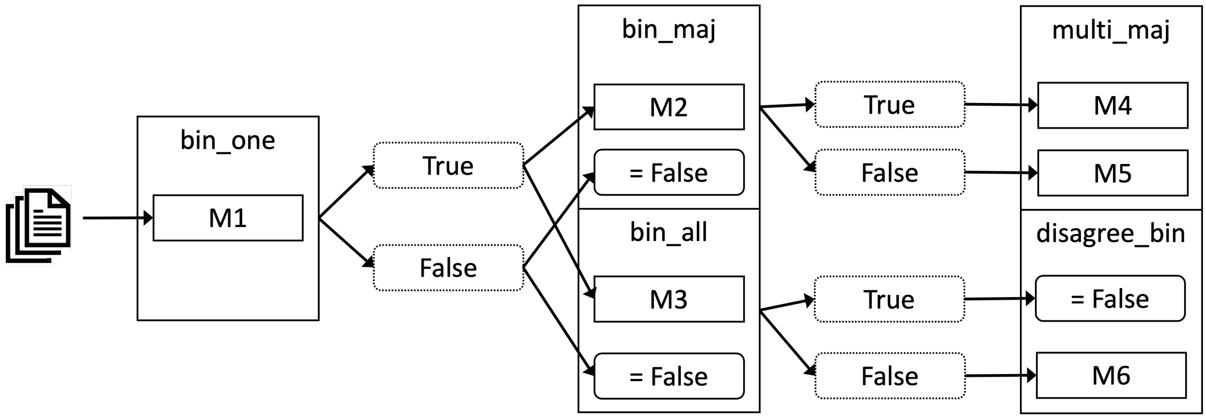

Figure 3: Pipeline for the closed Subtask 1.

3.1 Modeling

Since the subtasks require the prediction of five different binary and multi-class labels, we defined six different models.

The first model (**M1**) is a binary sequence classifier that receives all examples for training and predicts the label *bin_one*, which indicates whether there is at least one annotator who did not annotate *0-Kein*.

The second model (**M2**) receives all examples and classifies *bin_maj*. Therefore, the model has to predict whether there is a majority of annotators labeling other than *0-Kein*.

The third model (**M3**) is almost identical to **M2**, but differs in that it classifies *bin_all*, which indicates that all annotators labeled some form of sexism or misogyny.

The *multi_maj* classification is divided into two training sets. The fourth model (**M4**) is trained on examples that exhibit a clear form of sexism or misogyny, as indicated by the presence of at least one true instance of *bin_maj* or *bin_all*. In contrast, **M5** uses all available training examples to classify both distinct and indistinct examples.

The sixth model (**M6**) is applied to all examples where *bin_all* is not true and classifies *disagree_bin*.

3.2 Subtask 1

The approach for Subtask 1 uses the six models described in Section 3.1 in a sequential pipeline that is visualized in Figure 3. First, all examples in the test subset are predicted by **M1** to generate *bin_one* labels. Then, for all examples where the prediction of *bin_one* is true, **M2** predicts *bin_maj* and **M3** predicts *bin_all*. In cases where the prediction *bin_all* is true, the label *bin_maj* is also set to true. Conversely, if none of the labels are true or *bin_one* is false, both *bin_maj* and *bin_all* are set to false.

In all cases where the *bin_maj* prediction is true, **M4** predicts the *multi_maj* label. For all other examples, **M5** predicts the label *multi_maj*.

Finally, **M6** predicts the *disagree_bin* label for all instances where *bin_all* was predicted as false.

3.3 Subtask 2

Similar to our approach in Subtask 1, we use a pipeline to compute the required outputs. This pipeline is shown in Figure 4 and reuses a subset of the models from Subtask 1. Moreover, we do not extend the model training and we do not use the available data on the number of annotators per example in the training set.

For the *dist_bin* distribution, we need to predict the proportion of annotators who have labeled an example as *0-Kein* (not sexist) versus those who have labeled it as sexist or misogynist. Since our **M1** model has already been trained to predict whether there is at least one sexist vote for an example, it can be reused for this purpose. First, we take the example text and use the **M1** model to predict softmax values for both binary values (true and false). Then, instead of relying solely on the softmax scores to define the distribution, we use an algorithm that we call *Nearest Distribution Matcher*. The matcher first generates a list of evenly spaced numbers in the range of 0 to 1. The number of values in this list is equal to the sum of the number of annotators in the example plus 1. In the case of two annotators, the resulting list would contain the values [0, 0.5, 1], corresponding to 0 %, 50 %, and 100 %, respectively. The distribution is then computed using the value with the smallest difference to the softmax score for each label (true

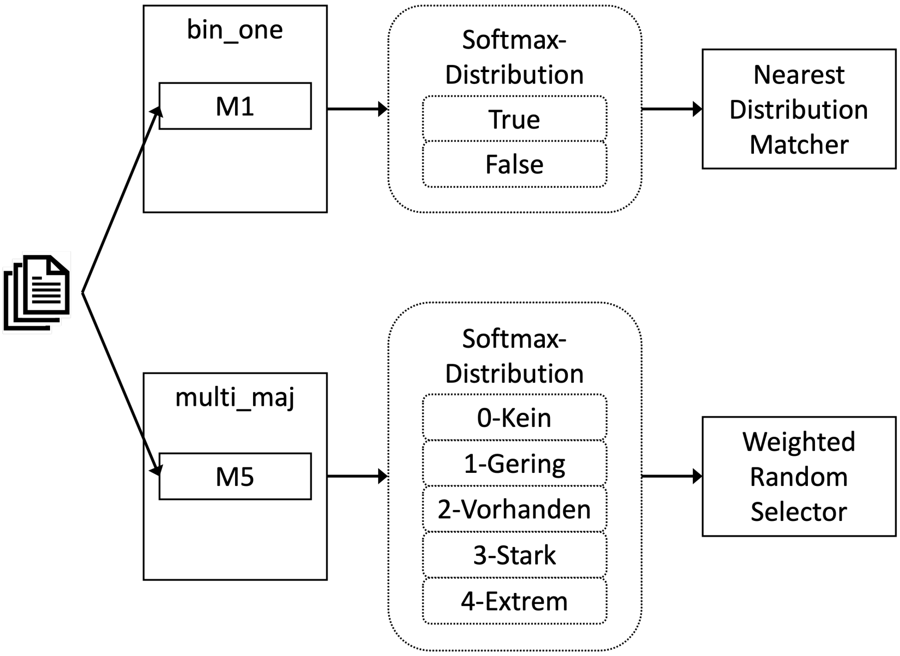

Figure 4: Pipeline for the closed Subtask 2.

and false).

The multi-score distribution, denoted *dist_multi*, is derived from the predictions of **M5**. Here we use the **M5** model to predict softmax scores for the example. To derive the distribution of annotated labels by the annotators, we use the softmax scores as probabilities for a *Weighted Random Selector*. For each annotator in the example, the selector chooses one of the five labels. Consequently, the final distribution is calculated by dividing the number of draws per label by the total number of draws for all labels in the example.

4 Training

Our training pipeline uses a pre-trained *deepset/gbert-large* model as a baseline for all six fine-tuned models. Therefore, for each task, a binary (**M1**, **M2**, **M3**, **M6**) or a multilabel (**M4**, **M5**) classification head with randomly initialized parameters is added to the encoder layer of the baseline model. For fine-tuning we use the raw texts of the training data and specify a learning rate between 2e-5 and 4e-5 and a number of epochs ranging from 8 to 30. We have manually tried to optimize the parameters in order to maximize the F1 score. The specific parameters for our models are available in the public repository[1].

In addition, the training pipeline uses all available training data for training, rather than splitting the data into training and validation subsets. We do this to maximize the number of training data points available for model training. Consequently, all models were first evaluated on the training set within the pipeline for the specific modeling tasks.

For fine-tuning our models and computing predictions, we utilized a system equipped with an NVIDIA A100 80GB PCIe GPU, supported by 128 GB of RAM and a 32-core Intel(R) Xeon(R) Gold 6248R CPU.

5 Results

After applying our ensemble method to the test data in Subtask 1, the predictions were uploaded to the shared task website for automated evaluation. The results for the five different classes and the final task score are shown in Table 2. Except for the labels *MultiMaj* and *DisagreeBin*, which achieved F1 scores of 0.414 and 0.610 respectively, all other labels achieved F1 scores of at least 0.7. This indicates that the challenging task was not unambiguous for the fine-tuned models. This may be due to the fact that the classification of text into levels of sexism or misogyny is sometimes a matter of interpretation, as even the annotators showed.

We also used the same trained models for Subtask 2, used them in the prediction pipeline for that task, and uploaded the predictions to the shared task website. The results are shown in Table 3. The results show that the distributions computed by our

[1] https://github.com/fmaoro/germeval24

Target Label	F1 Score
MultiMaj	0.414
BinMaj	0.744
BinOne	0.733
BinAll	0.705
DisagreeBin	0.610
Average Score	**0.641**

Table 2: Results for the closed Subtask 1.

pipeline have some differences, but still show substantial similarities to the distributions given by the annotated labels. Since our training process did not take into account the number of annotators or the distribution of labels, the result is rather weak. In addition, the randomness used for the weighted random selector affects each prediction, so running the pipeline again would produce different values.

In addition, the classification heads of the fine-tuned models were optimized to maximize the softmax scores for the true labels, and were not given any information about the distribution or uncertainty of the levels of sexism and misogyny.

Target Label	JS-Distance Score
Dist Multi	0.365
Dist Bin	0.343
Average Score	**0.354**

Table 3: Results for the closed Subtask 2.

6 Conclusion

In this work, we have proposed two related solutions for the two closed subtasks of the shared task GermEval 2024 GerMS-Detect. We solved the tasks by first training multiple BERT models to predict the labels of different subsets of the data. The use of six fine-tuned models (**M1**–**M6**) within a pipeline enabled strong performance for most of the classes in Subtask 1. The pipeline in Figure 3 was used to predict the labels of each example. Depending on the results of the first models, the further path in the pipeline was influenced. Thus, if an example was predicted to have no sexism / misogyny votes at all by the *binary_one* label, the further labels for *bin_all*, *bin_maj*, *multi_maj*, and *disagree_bin* were affected. This set of rules for applying models sequentially and only when necessary allowed for an efficient and effective use of the classifiers.

In addition, two of these models (**M1** and **M5**) were used to predict the distributions of annotators voting for the different labels in Subtask 2, with acceptable results. Using the softmax scores of the two classifiers in our Nearest Distribution Matcher and the Weighted Random Selector (see Figure 4), the distribution of annotators labeling the different levels was computed. Considering the uncertainty of classifying the level of sexism and misogyny in a text, the different results are understandable.

Acknowledgments

This work was funded by the German Federal Ministry of Education and Research under the grant no. 13N16242.

References

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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.

James Kotary, Vincenzo Di Vito, and Ferdinando Fioretto. 2023. Differentiable model selection for ensemble learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 1954–1962. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Ammar Mohammed and Rania Kora. 2023. A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, 35(2):757–774.

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.

Danny Wood, Tingting Mu, Andrew M. Webb, Henry W. J. Reeve, Mikel Luján, and Gavin Brown. 2024. A unified theory of diversity in ensemble learning. *J. Mach. Learn. Res.*, 24(1).