Context-Aware Content Moderation for German Newspaper Comments

Felix Krejca[0009−0002−0549−0865] [1,2], Tobias Kietreiber[0009−0003−4396−1135] [1], Alexander Buchelt[0000−0003−3851−6320] [1], and Sebastian Neumaier[0000−0002−9804−4882] [1]

[1] St. Pölten University of Applied Sciences, Austria
firstname.lastname@fhstp.ac.at
[2] Der Standard, Vienna, Austria

Abstract. The increasing volume of online discussions requires advanced automatic content moderation to maintain responsible discourse. While hate speech detection on social media is well-studied, research on German-language newspaper forums remains limited. Existing studies often neglect platform-specific context, such as user history and article themes. This paper addresses this gap by developing and evaluating binary classification models for automatic content moderation in German newspaper forums, incorporating contextual information. Using LSTM, CNN, and ChatGPT-3.5 Turbo, and leveraging the One Million Posts Corpus from the Austrian newspaper Der Standard, we assess the impact of context-aware models. Results show that CNN and LSTM models benefit from contextual information and perform competitively with state-of-the-art approaches. In contrast, ChatGPT's zero-shot classification does not improve with added context and underperforms.

Keywords: Automatic Content Moderation · Context-Aware Text Classification · Hate Speech Detection.

1 Introduction

With the rising quantity of text data in different kinds of media, the theoretical understanding and the practical implications of automatic content moderation based on machine learning techniques is more important than ever. Recent developments, such as the EU Digital Services Act (DSA) and other regulatory efforts, highlight the growing pressure on platforms to implement transparent and effective content moderation systems.[1] Manual moderation, while thorough, often exposes moderators to distressing material, leading to significant emotional and psychological challenges. Studies have documented that content moderators frequently encounter graphic and harmful content, resulting in conditions such

[1] https://digital-strategy.ec.europa.eu/en/policies/dsa-impact-platforms#ecl-inpage-lo5phy1n, last accessed 2025-02-05

as post-traumatic stress disorder, anxiety, and depression [27]. AI-driven moderation has the potential to diminish the emotional toll on moderators by handling straightforward decisions, thereby reducing their exposure to harmful material.

While much of the recent research has focused on detecting hate speech and offensive language in social media (e.g., [1,16,19]), limited studies have explored automated content moderation within German-language newspaper forums. Moreover, most existing research in this area does not utilize platform-specific context information, such as user posting history or details about the article under which a comment is posted (e.g., [2,25,26,31,21,13,12]).

This paper contributes to this branch of research by providing novel approaches for binary automatic content moderation with new ways of using platform-specific contextual information, to decide if a comment should stay online (0) or be removed (1), for German newspaper comments, using LSTM, CNN and LLM (ChatGPT: GPT-3.5-Turbo)[2] models. We compare the models and prompts to the state-of-the-art classification models for automatic content moderation and offensive language classification of German newspaper comments (e.g., [2,26,29]). We address the following central research question: *How do LSTM, CNN, and LLM models incorporating contextual information perform in binary automatic content moderation tasks for German newspaper comments compared to previous classification approaches?*

To explore this question, we use the One Million Posts Corpus [26], which comprises 1,000,000 posts from the Austrian newspaper Der Standard.[3] Our findings show that CNN and LSTM models enhanced with contextual information achieve competitive performance compared to state-of-the-art transformer-based models such as BERT [2,31,21,12]. However, we also demonstrate that incorporating contextual information does not significantly improve the zero-shot classification performance of ChatGPT 3.5 Turbo in terms of accuracy and F1-score.

The remainder of this paper is structured as follows: Section 2 provides a review of the relevant literature and theoretical foundations, defining key concepts and presenting a taxonomy of existing methods in the field. Section 3 outlines the methodology, including data preprocessing, the architectures of the CNN and LSTM models, and the prompt configurations used for ChatGPT. Section 4 presents the results, comparing our findings to existing studies. Finally, Section 5 concludes the paper by summarizing key insights and highlighting directions for future research.

2 Background & Related Work

2.1 Hate Speech, Offensive Language, and Platform Guidelines

In line with existing literature, we define *automatic content moderation* as the use of computational decision programs to determine whether user-generated content

[2] https://platform.openai.com/docs/models/gpt-3-5-turbo/, last accessed 2025-02-06

[3] https://www.derstandard.at/, last accessed 2025-02-06

(e.g., a post) complies with platform-specific rules. Based on this assessment, an appropriate moderation action is taken (e.g., the post is removed) [7]. In contrast, manual content moderation refers to human moderators making these decisions instead of an automated system.

In the literature, several terms are closely linked to content moderation decisions, including *hate speech* [5, p. 85], abusive language, cyberbullying, and *offensive language* [30]. Understanding these concepts is essential for analyzing recent research and, in particular, the annotation guidelines used for manual labeling in content moderation studies [5,30].

Fortuna & Nunes [5] provide an overview of various hate speech definitions, highlighting key aspects: (1) targeting specific groups based on characteristics like ethnicity or religion, (2) intent to incite violence or hatred, (3) language that demeans or attacks these groups, and (4) challenges in distinguishing hate speech from humor, particularly on platforms like Facebook. Waseem et al. [30] define abusive language along two dimensions: (1) whether it targets individuals or generalized groups, and (2) whether the abuse is overt or implicit. Explicit abuse includes clear slurs, while implicit abuse is often obscured by sarcasm or ambiguous language, making detection more challenging. Since hate speech is a subset of abusive language, broader definitions such as offensive language may be more applicable in content moderation. Zampieri et al. [32] define offensive language as "any form of non-acceptable language (profanity) or a targeted offense, whether veiled or direct."

Hate speech and offensive language are just one part of content moderation, which in newspaper forums often follows platform-specific rules. For example, Der Standard moderates not only for offensiveness but also for relevance, spam, personal data, and legal compliance. [4] This work explores how machine learning can be trained to reflect such platform-specific guidelines for more effective automated moderation.

2.2 Datasets for Automatic German Newspaper Moderation

The One Million Posts Corpus [26] is a German-language dataset featuring user comments from the Austrian newspaper DerStandard, annotated for content moderation. It includes 1,000,000 binary-labeled posts (online/offline) and 11,773 multi-labeled posts, categorized by professional moderators based on factors such as sentiment, relevance (on-/off-topic), inappropriate language, discrimination, feedback, personal stories, and argumentation. The dataset provides rich contextual information, including comment structure, timestamps, article metadata, and user interactions. The dataset is widely used in research on automated moderation in German newspaper forums [26,2,31]. Given its large-scale binary-labeled data, it serves as a robust benchmark for training and evaluating models in this study.

[4] Der Standard Community Guidelines, https://www.derstandard.at/communityrichtlinien, last accessed 2025-02-06

The Rheinische Post dataset [2] consists of user comments from the Rheinische Post website,[5] annotated by moderators and crowdworkers using a custom labeling schema that includes offensive categories (e.g., sexism, racism, insults) and organizational labels (e.g., advertisement, meta-discussions). Unlike the One Million Posts Corpus, the Rheinische Post dataset lacks contextual information for the postings. The NDR Dataset [31] includes labeled comments from NDR news articles,[6] The dataset has only binary labels (offline/online) for each comment and no contextual details, besides the article title and article URL. AustroTox contains Austrian news forum comments annotated for offensiveness, including token-level spans marking vulgar language and targets of offensive statements. The dataset contains the title of the articles as additional contextual information [21]. HOCON34k comprises over 34,000 German newspaper comments labeled for hate speech and the absence or presence of enough context and contains no additional contextual information [12]. GERMS-AT includes 8,000 user comments from Austrian Newspaper forums, annotated on a 0–4 scale for sexist and misogynistic content, and has no contextual information beyond that [13].

2.3 Taxonomy of Content Moderation Methods

Hate speech and offensive language detection are typically framed as supervised machine learning tasks, requiring large manually labeled datasets [2]. Based on literature reviews on content moderation [1,16,19] the approaches can be categorized into three primary groups:

(1) Traditional (Shallow) Methods. Shallow models rely on conventional word representation techniques like TF-IDF and n-grams [16]. Sentiment lexicons and linguistic features can enhance classification accuracy. Popular classifiers include Support Vector Machines (SVM), Naïve Bayes (NB), logistic regression, decision trees, and K-Nearest Neighbors [19].

(2) Advanced Methods. Word embeddings generate vectorized representations that capture semantic relationships between words [16]. Common methods include word2vec [17], fasttext [10], and GloVe [22]. These embeddings are used in combination with traditional classifiers or deep neural networks, improving performance compared to shallow methods.

Deep learning models process input using either traditional feature encoding (e.g., TF-IDF) or pre-trained embeddings. Established architectures include:

- Long Short-Term Memory (LSTM) models: Introduced by Hochreiter & Schmidhuber [6], LSTMs effectively retain long-term dependencies. They have been applied in German newspaper comment moderation [26].
- Convolutional Neural Networks (CNNs): CNNs have been successfully applied to hate speech detection in English datasets [1] but remain underexplored in German-language moderation.

[5] https://rp-online.de/, last accessed 2025-02-06

[6] Dataset currently not accessible, https://www.ndr.de/, last accessed 2025-02-06

- Transformer-based Models: Introduced in Attention Is All You Need [28], finetuned transformers, based on e.g., BERT, ELECTRA, or ALBERT, outperform earlier deep learning models in hate speech classification [16,24,9].

(3) Generative Pretrained Transformer-based LLMs. Generative Large Language Models (LLMs), such as ChatGPT 3.5, leverage transformers and allow zero-shot or few-shot learning via prompt engineering [4,14,15,18,8,3,21]. OpenAI's API enables automated moderation pipelines; on the other hand, open source alternatives like Mistral 7B provide local deployment options.[7] Effective prompt design is crucial for achieving optimal classification performance [14,21].

2.4 Offensive language detection in German newspaper comments

Despite the growing importance of automatic moderation, research on German-language newspaper comment moderation remains limited, particularly when focusing on a binary online/offline decision [31]. Instead, many studies center on detecting specific categories such as offensive or abusive language, rather than addressing content moderation as a whole [2,26]. This distinction is crucial, as different definitions and conceptual scopes lead to variations in annotation guidelines and dataset labeling practices in this research field [5].

Paper	Classifier	Word Representation
Schabus et al. 2017 [26]	Support Vector Machine	Countvectorizer (Bag of Words)
	Multinomial Naive Bayes	Countvectorizer (Bag of Words)
	Support Vector Machine	naive Bayes log-count ratios
	Bag of Cluster	word2vec
	Support Vector Machine	doc2vec
	LSTM	pre-trained embedding
Assenmacher et al. 2021 [2]	Multinomial N. Bayes	tf-idf / fasttext
	Logistic Regression	tf-idf / fasttext
	Gradient Boosted Trees	tf-idf / fasttext
	AutoML	tf-idf / fasttext
	BERT	/
Yadav & Milde 2021 [31]	Logistic Regression	Countvectorizer (Bag of Words)
	Logistic Regression	doc2vec
	Neural Network	doc2vec
	BERT	/
Pachinger et al. 2024 [21]	BERT	/
	Electra	/
	GPT-3.5	/
	GPT-4	/
	LeoLM	/
	Mistral	/
Krenn et al. 2024 [13]	BERT	/
Keller et al. 2025 [12]	BERT	/

Table 1. Overview of classifiers and word representations used in studies on German newspaper comment moderation.

[7] https://mistral.ai/en/news/announcing-mistral-7b, last accessed 2025-02-07

There are six major papers in the scientific literature [26,2,31,21,13,12]; Table 1 gives an overview of the classifiers and word representations used in the papers. Notably, Yadav & Milde [31] incorporate contextual information by using both the article title and comment, as well as splitting data by topic. Pachinger et al. [21] use the article title as contextual information in their experiments, while Keller et al. [12] consider the absence of enough contextual information during their data annotation process.

3 Methods

3.1 Data Overview and Preprocessing

We first give an overview of the data recorded in the One Million Posts Corpus in Table 2 for posts and Table 3 for articles. We use the columns Path, Title of the article, Headline and Body of the post (merged into a Comment variable) as well as two engineered features R_o^s and R_o^f described further down below to predict the Status of the post, i.e. whether it is still "online" or was "deleted" by a moderator.

Name	Description	Type	Example
Post ID	Identifier of the Post	integer	81085
Article ID	The ID of the article the comment was posted under	integer	1212
Parent Post	NULL for top-level comments, otherwise the ID of the parent comment	integer or NULL	80997
User ID	The (anonymized) user ID	integer	7721
Headline	The headline of the post	text or NULL	NULL
Body	The main text of the post	text or NULL	Drei Packerl Karten á 36 Blatt pro Jahr
Time Stamp	When the post was created	date	2015-07-02 12:25:53.553
Positive Votes	upvotes by other community members	integer	0
Negative Votes	downvotes by other community members	integer	0
Status	Whether the post is "online" or "deleted"	text	online

Table 2. The data recorded in the One Million Posts Corpus for each post.

There is a large class imbalance between "online" and "deleted" posts in the One Million Posts Corpus, with 1,011,773 labeled as "online" and 62,320 labeled as "deleted". This can lead to a worse classification performance of the resulting models, so following the recommendations of [23], we balance the data by randomly dropping "online" comments, resulting in 124,640 total entries, 62,320 "online" and 62,320 "deleted".

The Path, Title and Comment variables are text variables, so we will need further preprocessing on them. Following the suggestion of [11], we make use of encoding, cleaning, tokenization, stop word removal, and lemmatization. After importing, the data is further cleaned by converting all text to lowercase and

Name	Description	Type	Example
`Article ID`	Identifier of the article	`integer`	1212
`Path`	Breadcrumbs to the article	`text`	Newsroom/Panorama/Chronik
`Date`	Date of Publishing	`date`	2015-07-02 05:30:00.00
`Title`	Title of the article	`text`	Damenstift in Innsbruck: Adelsfrauen, die täglich für den Kaiser beten
`Body`	Content of the article	`text`	*omitted for brevity*

Table 3. The data recorded in the One Million Posts Corpus for each article.

removing non-alphabetic characters. After that, lemmatization is done. Lemmatization brings the words to a more general form. Then the text is tokenized into single words. In case a model receives multiple text variables as input, the specific variables are concatenated with special tokens. "LINK" for the topic path,"TITEL" for the title of the article, and "KOMMENTAR" for the user comment.

We also construct two additional features from the `User ID`, a simple online ratio R_o^s, and the full online ratio R_o^f, as follows:

$$R_o^s := \frac{C_{\text{online}}^{\text{train}}}{C^{\text{train}}},$$

$$R_o^f := \frac{C_{\text{online}}^{\text{train}} + C_{\text{online}}^{\text{ds}}}{C^{\text{train}} + C^{\text{ds}}},$$

where C^{train} is the number of comments of the user in the training set, C^{ds} is the number of comments of the user in the part of the dataset lost to downsampling, $C_{\text{online}}^{\text{train}}$ is the number of online comments of the user in the training set, $C_{\text{online}}^{\text{ds}}$ is the number of online comments of the user in the part of the dataset lost to downsampling. R_o^s therefore represents the percentage of user comments in the downsampled dataset that are still online, while R_o^f represents the same percentage in the whole dataset. Note that C^{ds} and $C_{\text{online}}^{\text{ds}}$ do not use any data points from the test or validation sets to prevent any form of data leakage.

3.2 Architectures

The model architectures used in this paper can be categorized into three groups: Traditional shallow methods as baselines, deep learning based approaches and generative pretrained transformers. A graphical comparison of the deep learning architectures is shown in Figure 1.

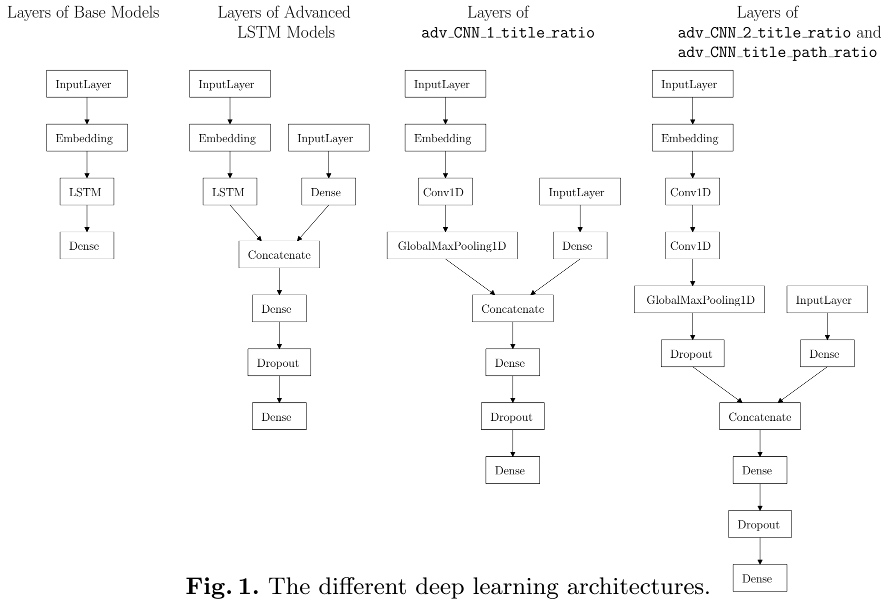

Fig. 1. The different deep learning architectures.

Traditional Shallow Methods. Traditional shallow models are utilized as baseline models, with no additional context information, thus predicting the binary classification task solely with the cleaned text of the user comment.

After the data preprocessing explained in the last section, it is crucial to transform the text into a numerical representation. For this a Countvectorizer is used and on top of that a traditional classifier. We use a multinomial naive Bayes and logistic regression.

Deep Learning Methods. The deep learning methods additionally use LSTM and CNN architectures. As embeddings we use pre-trained word embeddings from fasttext [10]. We use three models with simple LSTM architectures, three with more advanced LSTM architectures, and three with CNN architectures, which all differ in the inputs they take to make their predictions. These are summarized in Table 4.

Generative Pretrained Transformer-based LLMs. For the generative transformer-based models we use the OpenAI API [20] and the model "ChatGPT: GPT-3.5-Turbo" to do the experiments, following the discussion in Li et al. [14]. Building

Name	Inputs Used
`base_LSTM`	`Comment`
`base_LSTM_title`	`Comment`+`Title`
`base_LSTM_title_path`	`Comment`+`Title`+`Path`
`adv_LSTM_Title_simple_ratio`	`Comment`+`Title`+R_o^s
`adv_LSTM_Title_ratio`	`Comment`+`Title`+R_o^f
`adv_LSTM_Title_Path_ratio`	`Comment`+`Title`+`Path`+R_o^f
`adv_CNN_1_title_ratio`	`Comment`+`Title`+R_o^f
`adv_CNN_2_title_ratio`	`Comment`+`Title`+R_o^f
`adv_CNN_title_path_ratio`	`Comment`+`Title`+`Path`+R_o^f

Table 4. The inputs for the different deep learning models.

on recent findings in prompt design, we focus on providing richer contextual information to improve classification. Comments are input as raw text. Hyperparameters are tuned following Li et al., with temperature set to 0 for determinism and top-p to 0.95. To assess the impact of context, we test multiple prompt variations, starting from the following base prompt:[8]

> Du bist ein Forenmoderator und dafür zuständig, Kommentare unter einem Zeitungsartikel zu moderieren. Mache eine Prediction zur Moderationsentscheidung ob das Kommentar Online bleiben soll ‘‘0’’ oder Offline genommen werden soll ‘‘1’’. Das Kommentar ist: ‘‘[*comment*]’’. Antworte ausschließlich im Json Format {‘‘Moderationsentscheidung’’: prediction}

In total, we use 7 different variants, summarized in Table 5, including the use of the newspaper's forum rules.[9]

Name	Information in Prompt
`GPT_base`	base prompt only
`GPT_mod_title`	base prompt modified to include the article title
`GPT_mod_title_strength`	base prompt + article title + request for output of strength of prediction
`GPT_mod_title_ratio`	base prompt + article title + R_o^f
`GPT_mod_title_path`	base prompt + article title + path
`GPT_mod_title_erklaerung`	base prompt + article title + request for explanation of decision
`GPT_mod_title_forenregeln_kurz_erklaerung`	base prompt + article title + short summary of forum rules + request for explanation of decision

Table 5. The different prompt configurations of the GPT models.

[8] English translation: `You are a forum moderator and are responsible for moderating comments under a newspaper article. Make a prediction for the moderation decision whether the comment should remain online ‘‘0’’ or be taken offline ‘‘1’’. The comment is: ‘‘[`*comment*`]’’. Reply exclusively in Json format {‘‘Moderation decision’’: prediction}`

[9] https://about.derstandard.at/agb/#Forum, last accessed 2025-02-11

4 Results

This section presents the evaluation of various machine learning models for binary classification tasks, using performance metrics such as accuracy, F1-score, precision, recall, and AUROC, as outlined by [23]. The models assessed include advanced LSTM and CNN architectures, simpler classifiers such as naive Bayes and logistic regression, and models based on GPT prompts.

Model	**Accuracy**	**AUROC**	**F1-Score**	**Precision**	**Recall**	**Missing Answer**
`adv_LSTM_Title_path_ratio`	0.733	0.809	0.713	0.734	0.692	/
`adv_LSTM_Title_ratio`	0.729	0.808	0.718	0.714	0.723	/
`adv_CNN_1_title_ratio`	0.728	0.804	0.699	0.741	0.663	/
`adv_CNN_title_path_ratio`	0.726	0.804	0.703	0.730	0.681	/
`adv_CNN_2_title_ratio`	0.726	0.802	0.701	0.733	0.673	/
`adv_LSTM_Title_simple_ratio`	0.725	0.802	0.725	0.695	0.757	/
`base_LSTM_title_path`	0.703	0.777	0.698	0.678	0.720	/
`base_LSTM_title`	0.699	0.772	0.687	0.683	0.693	/
`naive_bayes`	0.678	0.743	0.679	0.677	0.681	/
`logistic_regression`	0.663	0.726	0.656	0.670	0.643	/
`base_LSTM`	0.661	0.728	0.667	0.629	0.710	/
`GPT_mod_title_path`	0.635	/	0.651	0.609	0.700	75
`GPT_base`	0.632	/	0.652	0.601	0.712	56
`GPT_mod_title_ratio`	0.631	/	0.643	0.605	0.687	158
`GPT_mod_title_erklaerung`	0.629	/	0.654	0.597	0.725	44
`GPT_mod_title`	0.626	/	0.646	0.598	0.702	63
`GPT_mod_title_forenregeln_kurz_erklaerung`	0.613	/	0.672	0.571	0.816	53
`GPT_mod_title_strength`	0.606	/	0.673	0.563	0.835	39

Table 6. Model Performance Metrics

Model Performance Overview. The results summarized in Table 6 indicate that the advanced LSTM and CNN models consistently outperform simpler approaches. The `adv_LSTM_Title_path_ratio` model achieves the highest accuracy (0.733) and AUROC (0.809). Conversely, simpler models such as logistic regression and naive Bayes perform at a lower level, with accuracy scores of 0.663 and 0.678, respectively. The GPT-based models exhibit the poorest overall performance, with accuracy scores ranging from 0.606 to 0.635, and high rates of "missing answer" – instances where the model fails to classify comments due to incorrect formats or refusal to provide a response.

Accuracy and the Role of Contextual Information. As seen in Table 6, adding contextual variables significantly enhances the performance of self-trained models, whereas this does not hold for GPT-based models. For example, `base_LSTM`, which uses only cleaned comment text as input, is outperformed by `base_LSTM_title` and `base_LSTM_title_path`, which incorporate the article title and topic path as additional input features. Interestingly, GPT models do not show similar improvements when contextual information is added, indicating that the increased input complexity incurs additional computational costs without a corresponding increase in accuracy.

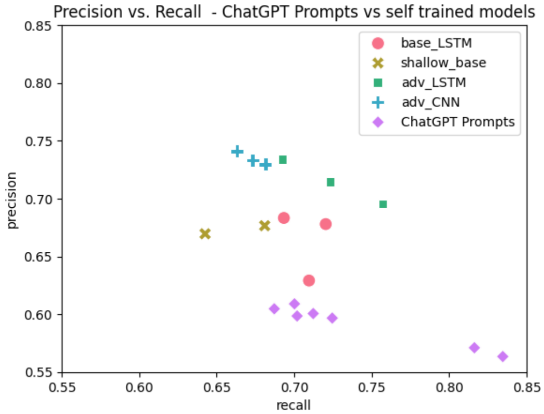

Fig. 2. Precision-Recall comparison of the different models.

AUROC and F1-Score Analysis. The AUROC values for the models mirror the trends observed in accuracy. The advanced LSTM models demonstrate slightly superior AUROC scores compared to other models. In terms of F1-score, the advanced LSTM model using the simple ratio input achieves the highest score. Among the GPT models, `GPT_title_strength` achieves a relatively high recall but at the expense of precision, leading to a higher rate of false positives. These findings suggest that such models may be more suitable for human-assisted moderation systems, where maximizing recall is preferred over precision.

Precision and Recall Comparison. Figure 2 presents a comparison of precision and recall, revealing that the GPT-based models exhibit high recall but lower precision, implying a higher false positive rate. In contrast, the self-trained models demonstrate a more balanced performance between precision and recall, with the advanced LSTM and CNN models outperforming other approaches. Models incorporating user history, in the form of the Online Ratio variable, show the best performance, while the inclusion of the `Path` variable contributes minimally to performance improvements.

Comparison of Self-Trained and GPT Models. Overall, the self-trained models outperform GPT-based models, particularly in terms of accuracy and F1-score. The advanced LSTM and CNN models exhibit superior results, with the `adv_LSTM_Title_path_ratio` model achieving the highest accuracy and AUROC values. Although computationally less expensive, simpler classifiers such as logistic regression and naive Bayes outperform the `base_LSTM` model but are consistently outperformed by more advanced models such as `base_LSTM_title` and `base_LSTM_title_path`. It is also worth noting that there are no significant differences in performance between the advanced LSTM and CNN architectures.

Comparison with Previous Literature. When compared to previous work in the field, such as [2], [21],[12] and [31], the self-trained models presented in this study exhibit competitive performance. While Assenmacher's best model (BERT) achieves a superior AUROC score (0.914), the highest-performing model in this study (`adv_LSTM_Title_path_ratio`) achieves a strong AUROC of 0.809. Notably, when comparing models based on professional moderator labels, the advanced CNN and LSTM models developed in this study outperform the professional-moderator-based models from [2]. Similarly, compared to [31], our best model achieves higher F1-score, precision, and recall, although it underperforms in terms of accuracy.

5 Conclusion

This study explored the potential of incorporating contextual information into automatic content moderation systems for German-language newspaper comments. By integrating context variables such as user posting history, forum rules, and article titles into LSTM, CNN, and large language models like ChatGPT 3.5 Turbo, the results show that LSTM and CNN models with context can compete with state-of-the-art transformer models like BERT in terms of performance.

Another finding is that adding context to GPT-3.5 Turbo prompts did not significantly improve classification accuracy, with simpler, context-free prompts performing comparably, underscoring the need to balance model complexity and cost. Additionally, the study highlights the potential of context-enriched models to improve moderation. A limitation of our work is the continuous advancement of LLMs, meaning that more recent models may improve the performance and integration of contextual information.

Future research could explore adding more detailed contextual data, fine-tuning transformer models, and developing models capable of explaining their classification decisions, which could be important for legal and ethical considerations. The cost-effectiveness of using contextual models and maintaining fairness in variables like the Online Ratio are critical concerns for further exploration.

References

1. Alkomah, F., Ma, X.: A literature review of textual hate speech detection methods and datasets. Information **13**(6) (2022). https://doi.org/10.3390/info13060273
2. Assenmacher, D., Niemann, M., Müller, K., Seiler, M.V., Riehle, D.M., Trautmann, H.: RP-Mod & RP-Crowd: Moderator- and Crowd-Annotated German News Comment Datasets (2021). https://doi.org/10.5281/zenodo.5242915, https://doi.org/10.5281/zenodo.5242915, [dataset]
3. Chiu, K.L., Collins, A., Alexander, R.: Detecting hate speech with gpt-3 (2022), https://arxiv.org/abs/2103.12407
4. Dehghan, S., Yanikoglu, B.: Evaluating ChatGPT‘s ability to detect hate speech in Turkish tweets. In: Proceedings of the 7th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (2024). pp. 54–59. Association for Computational Linguistics, St. Julians, Malta (2024), https://aclanthology.org/2024.case-1.6/

5. Fortuna, P., Nunes, S.: A survey on automatic detection of hate speech in text. ACM Comput. Surv. **51**(4) (2018). https://doi.org/10.1145/3232676
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
7. Horta Ribeiro, M., Cheng, J., West, R.: Automated content moderation increases adherence to community guidelines. In: Proceedings of the ACM Web Conference 2023. p. 2666–2676. WWW '23, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3543507.3583275
8. Jaremko, J., Gromann, D., Wiegand, M.: Revisiting implicitly abusive language detection: Evaluating llms in zero-shot and few-shot settings. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 3879–3898 (2025)
9. Jose, M., Anthony, J., Joseph, J.V., Thomas, J., Thomas, S.B.: A review of machine learning and deep learning approaches for offensive text detection. International Journal on Emerging Research Areas (IJERA) **04**(02), 47–50 (2025). https://doi.org/10.5281/zenodo.14651005, https://doi.org/10.5281/zenodo.14651005
10. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics, Valencia, Spain (2017), https://aclanthology.org/E17-2068/
11. Kang, Y., Cai, Z., Tan, C.W., Huang, Q., Liu, H.: Natural language processing (nlp) in management research: A literature review. Journal of Management Analytics **7**(2), 139–172 (2020). https://doi.org/10.1080/23270012.2020.1756939
12. Keller, M.E., Auch, M., Döschl, A., Vlk, F., Quernheim, J., Hartmann, M., Mandl, P., Kaul, A., Franz, M., Greguš, M., Kotsis, G., Delir Haghighi, P., Khalil, I.: Hocon34k: A corpus of hate speech in online comments from german newspapers. In: Information Integration and Web Intelligence, pp. 212–226. Springer Nature Switzerland (2025). https://doi.org/10.1007/978-3-031-78090-5_18, https://doi.org/10.1007/978-3-031-78090-5_18
13. Krenn, B., Petrak, J., Kubina, M., Burger, C.: Germs-at: A sexism/misogyny dataset of forum comments from an austrian online newspaper. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). pp. 7728–7739 (2024)
14. Li, L., Fan, L., Atreja, S., Hemphill, L.: "hot" chatgpt: The promise of chatgpt in detecting and discriminating hateful, offensive, and toxic comments on social media. ACM Trans. Web **18**(2) (2024). https://doi.org/10.1145/3643829
15. Lu, J., Ma, K., Wang, K., Xiao, K., Lee, R.K.W., Xu, B., Yang, L., Lin, H.: Unveiling the capabilities of large language models in detecting offensive language with annotation disagreement. https://doi.org/10.48550/arxiv.2502.06207 (2025), arXiv:2502.06207
16. Malik, J.S., Qiao, H., Pang, G., van den Hengel, A.: Deep learning for hate speech detection: a comparative study. International Journal of Data Science and Analytics pp. 1–16 (2024)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013), https://arxiv.org/abs/1301.3781
18. Mnassri, K., Farahbakhsh, R., Chalehchaleh, R., Rajapaksha, P., Jafari, A.R., Li, G., Crespi, N.: A survey on multi-lingual offensive language detection. PeerJ. Computer Science **10**, e1934–e1934 (2024). https://doi.org/10.7717/peerj-cs.1934, https://doi.org/10.7717/peerj-cs.1934

19. Mullah, N.S., Zainon, W.M.N.W.: Advances in machine learning algorithms for hate speech detection in social media: A review. IEEE Access **9**, 88364–88376 (2021). https://doi.org/10.1109/ACCESS.2021.3089515
20. OpenAI: Openai api (2025), https://mistral.ai/en/news/announcing-mistral-7b
21. Pachinger, P., Goldzycher, J., Planitzer, A.M., Kusa, W., Hanbury, A., Neidhardt, J.: Austrotox: A dataset for target-based austrian german offensive language detection. arXiv preprint **arXiv:2406.08080** (2024), https://arxiv.org/abs/2406.08080
22. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
23. Raschka, S., Liu, Y.H., Mirjalili, V.: Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python. Packt Publishing Ltd (2022)
24. Saumya, S., Kumar, A., Singh, J.P.: Filtering offensive language from multilingual social media contents: A deep learning approach. Engineering Applications of Artificial Intelligence **133**(Part B), 108159 (2024). https://doi.org/10.1016/j.engappai.2024.108159, https://doi.org/10.1016/j.engappai.2024.108159
25. Schabus, D., Skowron, M.: Academic-industrial perspective on the development and deployment of a moderation system for a newspaper website. In: Proceedings of the eleventh international conference on language resources and evaluation (2018)
26. Schabus, D., Skowron, M., Trapp, M.: One million posts: A data set of german online discussions. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1241–1244. SIGIR '17, New York, NY, USA (2017). https://doi.org/10.1145/3077136.3080711
27. Spence, R., Bifulco, A., Bradbury, P., Martellozzo, E., DeMarco, J.: The psychological impacts of content moderation on content moderators: A qualitative study. Cyberpsychology: Journal of Psychosocial Research on Cyberspace **17**(4) (2023)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017), https://doi.org/10.48550/arxiv.1706.03762
29. Vidgen, B., Derczynski, L.: Directions in abusive language training data, a systematic review: Garbage in, garbage out. PLOS ONE **15**(12), 1–32 (2021). https://doi.org/10.1371/journal.pone.0243300
30. Waseem, Z., Davidson, T., Warmsley, D., Weber, I.: Understanding abuse: A typology of abusive language detection subtasks. In: Proceedings of the First Workshop on Abusive Language Online. pp. 78–84. Association for Computational Linguistics, Vancouver, BC, Canada (2017). https://doi.org/10.18653/v1/W17-3012
31. Yadav, A., Milde, B.: forumBERT: Topic adaptation and classification of contextualized forum comments in German. In: Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021). pp. 193–202. Düsseldorf, Germany (2021), https://aclanthology.org/2021.konvens-1.17/
32. Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., Kumar, R.: SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In: Proceedings of the 13th International Workshop on Semantic Evaluation. pp. 75–86. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019). https://doi.org/10.18653/v1/S19-2010