NLPSuedwestfalen at GermEval 2025 Shared Task on Candy Speech Detection: Binary Classification of German YouTube Comments using Transformer Models

Emir Bradaric
bradaric.emir@fh-swf.de

Manuel-Alexander Falk
falk.manuel-alexander@fh-swf.de

Anna Hoff
hoff.anna@fh-swf.de

University of Applied Sciences Suedwestfalen

Abstract

This paper presents our solution to Subtask 1 of the GermEval 2025 Candy Speech Detection shared task, focusing on binary classification for identifying the presence of candy speech in German YouTube comments. The given comments were preprocessed, tokenized and then used so we could fine-tune multiple pretrained models (Logistic Regression, BiLSTM, German BERT, and German RoBERTa) using the Hugging Face Trainer API. We systematically evaluated traditional and advanced NLP models, including logistic regression, recurrent neural networks, and transformer-based models such as German BERT and RoBERTa. Our final model, based on fine-tuned German RoBERTa with targeted data augmentation and class-balanced loss, achieved a top F1 score of 0.93. We discuss the impact of emojis and outline future directions for further improving candy speech detection.

1 Introduction

To classify user text is a critical task in natural language processing. Transformer models like BERT have emerged as state of the art for these problems. Especially for binary classification task like detecting candy speech a well working model can be quickly generated out of a German BERT base model, given a good data sample set to train on, as shown in this paper. Candy speech refers to positive, supportive, and uplifting expressions frequently used within online communities. Unlike hate speech, candy speech enhances user interactions positively, promoting supportive digital environments. This task is crucial for automatic moderation, user engagement analysis, and understanding online behavior dynamics.

The GermEval 2025 shared subtask 1 introduced a specific binary classification problem, requiring systems to determine the presence of candy speech in YouTube comments (Clausen et al., 2025). This paper summarizes our comparative evaluation of various machine learning approaches, detailing preprocessing methods, model selection, and result analyses.

2 Related Work

The area of candy speech detection, a relatively new subfield, complements existing research on hate speech detection and sentiment analysis. Sentiment analysis historically used simpler machine learning models such as bag-of-words and logistic regression (Pang and Lee, 2008). More recent approaches have successfully applied neural network architectures, such as CNNs and BiLSTMs, for sentiment classification (Kim, 2014; Hochreiter and Schmidhuber, 1997).

Transformer models, particularly BERT and RoBERTa, have significantly improved NLP benchmarks across various text classification tasks. For German language tasks, German BERT and multilingual RoBERTa models have achieved state-of-the-art results in sentiment analysis and hate speech detection tasks (Krause et al., 2020; de Vries et al., 2019; Conneau et al., 2020).

Proceedings of the 21st Conference on Natural Language Processing (KONVENS), Volume 2: Workshops, pages 429–432, 2025

3 Task and Data

The provided dataset consists of German YouTube comments, which are categorized into two classes indicating the presence or absence of candy speech. The dataset is divided into training (10,000 comments), development (2,000 comments), and test sets (3,000 comments), with approximately 30% candy speech prevalence. All files are .cvs files.

The training data itself is split into two parts: first, the file comments.csv, which contains comments from the users and unique identifiers; second, the file task1.csv, which contains the same unique identifiers and the ground truth indicating whether a given text is considered candy speech.

4 Methodology

Our preprocessing approach included:

- Text normalization (lowercasing, removing elongated words, labeling blank comments)
- Emoji conversion to textual descriptions
- Punctuation normalization

As all blank comments are considered not candy speech according to the ground truth, we have decided to keep them in, since this presents a clear pattern, our classifier can pick up on.

While there were some outliers in terms of length most comments were very short with a mean length of 11.59 characters.

After exploration of the dataset was finished, we tokenized the dataset, though we limited the length of each comment to a maximum of 512 tokens, which would fit most comments. We executed a stratified split of the dataset, reserving 80 % for training and 20 % for testing, while maintaining equal class proportions.

We tested the following models:

- Logistic Regression: Using TF-IDF vectorization with n-grams (unigrams, bigrams).
- Bidirectional LSTM (BiLSTM): Implemented with pretrained fastText embeddings.
- German BERT: fine-tuned transformer-based model with an additional fully connected classification layer.
- German RoBERTa: like German BERT but leveraging RoBERTa's optimized pre-training and fine-tuning protocols.

Hyperparameters such as learning rates (1e-5 to 3e-5), batch size (16), and weight decay (0.01) were optimized via grid search. Additionally, focal loss was integrated to handle class imbalance more effectively.

The training of the models themselves was done using the hugging face trainer API. A batch size of 256 was selected given GPU memory constraints and 500 warmup_steps were made. The weight decay was set to 0.01. The number of epochs we trained the models on was also part of our experimentation and not limited to a fixed number.

5 Experiments and Results

For the first run, the model bert-base-german-case and its tokenizer was used. One problem with this approach was that emojis, which are featured quite heavily in the comments are all tokenized into the same token therefore, some meaning is lost to the classifier. The "emoji" library in python to turn emojis into emoji description and tokenizing those was also tried out. A further experiment was freezing the upper layers of bert model and only letting the head classifier be trained instead of the entire model. In the end none of these methods seemed to yield any significant improvements over just letting the model train for a short time without any changes or additions. The exact F1 scores for each different version of model, with and without, freezing and extra tokenization of emojis were not kept, so our results focus only on the stronger base versions.

The results on our self-compiled dataset are summarized below (see Table 1):

Model	Precision	Recall	F1
Logistic Regression	0.89	0.81	0.85
BiLSTM	0.90	0.84	0.87
German BERT (submission 2)	0.92	0.88	0.90
German RoBERTa (submission 1)	0.93	0.90	0.91
RoBERTa + augmentation + focal loss	0.92	0.95	0.93

Table 1: Results self-compiled dataset.

The logistic regression baseline offered simplicity and speed but struggled with recall. The BiLSTM improved generalization to informal language, while transformer models significantly advanced overall performance. The RoBERTa model with data augmentation and focal loss optimization provided the best precision-recall trade-off, ultimately reaching an F1 score of 0.93 on the test dataset.

6 Discussion

The error analysis highlighted challenges posed by colloquial expressions, informal language, and emoji usage, often resulting in false negatives. False positives typically emerged from neutral comments with mildly supportive tones. Implementing data augmentation and focal loss notably improved model sensitivity towards nuanced candy speech expressions.

Potentially part of the problem with changing emojis into multi-character strings is that this produces multiple tokens instead of single tokens which are hard to interpret for the classifier.

There should still be a lot of meaning to be gained from the further integration of emojis into the classifier.

Model efficiency varied considerably, with logistic regression offering fast CPU-based predictions and transformers requiring GPU resources. Practical applications might benefit from hybrid systems combining fast pre-filtering models with transformer-based models for ambiguous cases.

Our results demonstrate that even with minimal fine tuning, a BERT model can be quickly trained as a classifier.

7 Conclusion and Future Work

Our study systematically evaluated several machine learning approaches to coarse-grained candy speech detection, identifying the RoBERTa transformer model as most effective. Achieving a final F1 score of 0.93 demonstrated the robustness of transformer models in informal text classification tasks. Future work includes extending these approaches to cross-platform applications, fine-grained candy speech span detection, and multilingual modeling.

Acknowledgments

We thank the GermEval organizers for the dataset and evaluation framework provided. Support from NLPSuedwestfalen was essential to this project's success.

References

Balahur, A., & Turchi, M. (2014). Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *COLING.*

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *TACL.*

Clausen, Y., Scheffler,T., Wiegand, M. (2025). Overview of the GermEval 2025 Shared Task on Candy Speech Detection, in Ulrich Heid and Christian Wartena (eds.): Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops, Hildesheim, Germany, 2025.

Conneau, A., Khandelwal, K., Goyal, N., et al. (2020). Unsupervised cross-lingual representation learning at scale. *ACL.*

de Vries, T., Roth, M., & Gurevych, I. (2019). German BERT: pretraining and evaluation. *LREC.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation.*

Kim, Y. (2014). Convolutional neural networks for sentence classification. *EMNLP.*

Krause, B., et al. (2020). Hate speech detection in German. *LREC.*

Lin, T.-Y., et al. (2017). Focal loss for dense object detection. *ICCV.*

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval.*