Flauschgummi at GermEval 2025 Shared Task on Candy Speech Detection: Sentiment Analysis and Classification of Online Comments

Björn Kießwetter and Christoph Lukas and Tim Sieber
University of Regensburg
{Bjoern.Kiesswetter|Christoph1.Lukas|Tim.Sieber}@stud.uni-regensburg.de

Abstract

There has been a noticeable increase in the use of offensive language across user-generated content on social media platforms. Such language can target, harm, or marginalize individuals or entire communities. In contrast to this negative trend, we focus on the identification and classification of candy speech – supportive and emotionally positive statements. We examine the effectiveness of stacking traditional machine learning techniques like Logistic Regression, k-Nearest Neighbors, Random Forest, Multi-Layer Perceptron, Support Vector Classifier and simple word counting, and compare them to BERT, a powerful pre-trained language model. The study evaluates both individual and ensemble methods to determine the most effective strategy. Our results indicate that traditional stacking methods used without BERT achieve the highest accuracy, followed by the hybrid approach, while BERT alone performs the worst on this specific task.

1 Introduction

Anyone who visited a social media platform like Facebook, Instagram or YouTube has most likely encountered offensive posts from users in the comment section, also known as hate speech. What if we could not only remove negativity, but also actively promote and spread positivity? To explore this idea, we participate in Subtask 1 of the GermEval 2025 Shared Task on Candy Speech Detection, where we focus on identification of candy speech in YouTube comments. Candy speech refers to expression of positive attitudes on social media toward individuals or their output, e.g. videos or comments (Clausen et al., 2025).

Detecting candy speech is a very complex task, as it depends not only on sentiment but also on contextual understanding. While langue models like GPT (OpenAI, 2024) or BERT (Devlin et al., 2019) have gained enormous influence in today's era, they also come with major limitations like high computational costs and a significant carbon footprint (Li et al., 2021). To evaluate the trade-off between computational efficiency and classification performance, we compare three modeling approaches: a classical stacking model as a lightweight baseline, a BERT-based model to assess the potential accuracy gains from deep contextual embeddings, and a hybrid model combining both approaches.

2 Related Work

2.1 Offensive Language Detection

Previous studies have explored the detection of hate speech using classical machine learning methods (Ayo et al., 2020; Davidson et al., 2017). Their findings suggest that traditional approaches, such as SVMs or Logistic Regression, can achieve performance comparable to that of deep learning models, particularly in the context of hate speech recognition on Twitter (Mercan et al., 2021).

2.2 Emotion Classification in Text

In recent years, language models like GPT, BERT and RoBERTa (Liu et al., 2019) proved to be very powerful in tasks such as emotion and sentiment classification (Zian et al., 2021). Automatic recognition of emotions in text is an active research field in natural language processing. Traditional approaches often use classical machine learning methods, such as SVMs or Random Forest. With the emergence of large pretrained language models, the precision and robustness of emotion classification have improved significantly (Stigall et al., 2024).

3 Methodology

This section introduces the steps taken to obtain additional training data, clean and prepare the data sets, as well as conducting the experiments.

Proceedings of the 21st Conference on Natural Language Processing (KONVENS), Volume 2: Workshops, pages 411–415, 2025

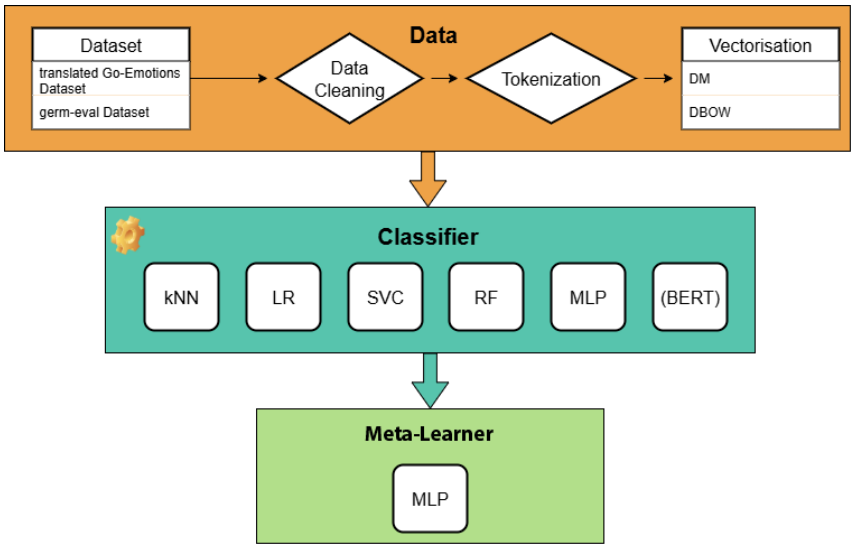

Figure 1: Pipeline

Figure 1 shows the execution flow of our code in the form of a pipeline, illustrating each step and how data is processed sequentially. First we clean the data, then tokenize it and apply our two vectorization methods: Distributed Memory (DM) and Distributed-Bag-of-Words (DBOW). These vectors are then passed on to our 5 different classifiers: k-Nearest Neighbors (kNN), Logistic Regression (LR), Support Vector Classifier (SVC), Random Forest (RF), and Multi-Layer Perceptron (MLP). The BERT model, in contrast, directly processes the raw comments, instead of vectors. The last station is the MLP. With help of the 6-dimensional vector, from the models mentioned above, it predicts whether a given comment contains candy speech or not.

3.1 Data Preparation

In an initial test, we trained our models on the official data provided by the shared task organizers, but our models did not achieve satisfactory results. Therefore, we decided to extend the training data by including additional comments from external sources (Demszky et al., 2020). Details are described in the following section. Since these comments were in English, we translated them into German using Helsinki-NLP/opus-mt-en-de (Tiedemann and Thottingal, 2020).

Data Cleaning The importance of preprocessing data in form of data cleaning is essential for improving the models' performance and ensuring consistent input for our classifier (Kalra and Aggarwal, 2018). Therefore, we performed a series of data cleaning steps. The effectiveness of this approach was also demonstrated by Lee et al. (2021) and further supported by Li et al. (2021).

Given the noisy and unstructured nature of YouTube comments, we deleted usernames, links, emojis, and excessive punctuation, which further improved model accuracy (see Figure 3).

Tokenization Since classical machine learning algorithms cannot process text directly, we converted it into numerical vectors using Doc2Vec (Le and Mikolov, 2014). This required an initial tokenization, for which we used NLTK's tokenization algorithm (Bird et al., 2009). The outcome of this procedure is a list of tokens that serve as input to Doc2Vec.

Vectorization of Comments For vectorisation, the Doc2Vec algorithm offers two main variants: DM and DBOW. Both approaches aim to learn fixed-length vector representations of given text. DM learns from word sequences and context, while DBOW focuses on predicting document words without considering word order. Both aim to produce meaningful representations of documents for downstream tasks (Le and Mikolov, 2014).

4 Models

4.1 Stacking without BERT

For a classification task, there are different approaches. We decided to use a stacking variant that combines five different machine learning algorithms. Stacking is a method in which multiple classifiers perform the same task in parallel. Their re-

sults are then combined, via a Meta-Learner reclassified, which allows them to compensate for each other's weaknesses and reinforce their strengths. This technique has shown its effectiveness in terms of accuracy and robustness (Zian et al., 2021).

For our stacking ensemble, we used the following traditional machine learning models: LR, kNN, SVC, MLP and RF. Since each of these classifiers uses random seeds, we performed the classification for each comment 10 times and returned the mean of the outcome to ensure stability and reduce variance. This resulted in a 5-dimensional vector with values ranging from 0 to 1.

4.2 Stacking with BERT

To leverage deeper semantic context, we repeated the same classification pipeline, i.e., LR, kNN, SVC, MLP, and RF, but added BERT. As before, we ran 10 iterations for every classification, resulting in a 6-dimensional vector with values again ranging from 0 to 1.

4.3 Simple Word Counting

An additional dimension for our classifier was to "rate" a comment based on present words. We used a collection of "bad", "good" and "intensifier" words to obtain a very easy semantic understanding of the given comment, inspired by the approach in (Palanisamy et al., 2013). This dictionary was manually curated from the comments in our dataset and includes several categories of words: positive and negative sentiment words, intensifiers that amplify the sentiment strength, negation words that can reverse sentiment polarity, and positive/negative emojis to capture emotional expressions.

4.4 Meta-Learner

Our main prediction model, the meta-learner, is a Multi-Layer Perceptron, which we trained with BERT on 6–7 dimensional vectors. These vectors were generated by the methods previously described to determine the semantics of each comment. The MLP was trained on 4,000 comments that were cleaned and preprocessed and not part of the comments we used to train our classifiers above. This means that our main prediction model comes into contact only with new and unseen data.

4.5 BERT

For comparison, we evaluated the classification performance of a pre-trained German BERT model (Guhr et al., 2020) applied directly to the task, without any additional stacking or feature engineering.

5 Results

5.1 Meta-Learner Performance

Model	F1	Precision	Recall
stacking without BERT	0.6972	0.8879	0.5739
stacking with BERT	0.6842	0.8765	0.5611
BERT alone	0.6852	0.6829	0.6874

Table 1: Performance comparison of the models submitted to Subtask 1.

As shown in Table 1, stacking without BERT outperforms the other approaches in terms of precision. However, when it comes to recall, BERT alone achieves the highest performance.

5.2 Model comparison

In this section, we present a comparison of all models based on the following metrics: accuracy, F1-score, Precision, and Recall.

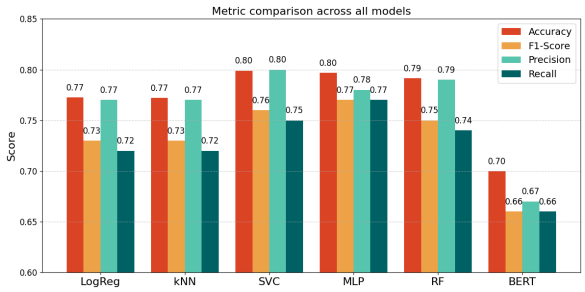

Figure 2: Metric comparison across all models.

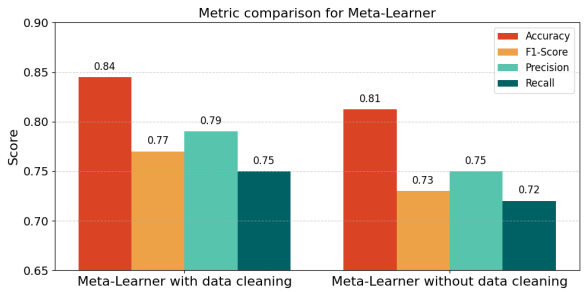

Figure 3: Metric comparison for Meta-Learner.

6 Discussion

We explored to what extent classical machine learning algorithms can remain relevant in an era increasingly shaped by the use of large language models, as discussed in (Miah et al., 2024; Zhan et al., 2024; Zhang et al., 2023). While we did not achieve state-of-the-art results with our model, it is important to

note that GPT-3.5 obtains a precision score of 0.87, compared to 0.96 for GPT-4 and 0.91 for Llama 2 in a related study that analyzes sentiment of text via large language models (Krugmann and Hartmann, 2024). Nonetheless, our findings highlight that relatively simple techniques, when combined thoughtfully, can still achieve competitive performance.

The overall accuracy, F1-score, Precision, and Recall of our model improved substantially, after we cleaned and preprocessed the training data. A direct comparison shows that our model trained on the unchanged data performed worse in classifying candy speech than the one trained on preprocessed data, as shown in Figure 3. A possible explanation could be that in the case of unchanged data, the model was trained with vectors that were distorted or semantically meaningless.

The improvements in accuracy, F1-score, Precision, and Recall can likely be attributed to preprocessing, which helped produce more consistent and informative input patterns, thereby enhancing classification performance. This trend has already been demonstrated by earlier research (Lee et al., 2021; Li et al., 2021) and appears consistent with the current findings.

A critical aspect in evaluating our model lies in the choice of the vectorization method. Although we used Paragraph Vector - Distributed Memory (PV-DM) and Paragraph Vector - Distributed Bag of Words (PV-DBOW), it remains an open question how our results might have improved, if we had used an alternative method, such as Term Frequency-Inverse Document Frequency (TF-IDF). The latter represents text by weighting each term based on its relative importance within a comment and across the entire dataset.

In related work on hate speech detection (Mercan et al., 2021), TF-IDF has proven its effectiveness by recognizing the relevance of a single word and reducing the influence of common but uninformative terms. This approach achieved higher accuracy scores using classifiers such as SVM, which is closely related to the SVC, RF, and LR models we used (Table 2).

Accuracy comparison	**SVM/SVC**	**LR**	**RF**
our study	0.76	0.73	0.75
related study	0.94	0.94	0.95

Table 2: Accuracy comparison of different classifiers in our study and a related study by Mercan et al. (2021).

As Table 2 shows, alternative vectorization methods led to noticeably better performance. This suggests that to improve our results, we should consider replacing PV-DM and PV-DBOW with TF-IDF.

Surprisingly, the best-performing model in our experiments did not include any LLM components. Instead, it relied solely on traditional ML algorithms and supervised stacking. This suggests that in tasks with strong lexical or stylistic signals, such as detecting emotionally supportive candy speech comments, deep contextual understanding may not be necessary. Simpler models can exploit these surface-level patterns effectively.

Moreover, while BERT is a powerful and general-purpose model, its embeddings did not considerably boost performance in our hybrid approaches.

7 Conclusion

In summary, with regard to candy speech detection, our results support the idea, that classical stacking ensembles can outperform both the standalone BERT-classifier as well as a hybrid model that incorporates deep contextual embeddings. We achieved substantial gains in accuracy, Precision, Recall, and F1-score trough data cleaning. Our cleaning process was tailored to the characteristics of the vectorization algorithm, which appeared to struggle with non-German words, emojis, and fragmented inputs. Moving forward, exploring alternative vectorization methods such as TF-IDF, employing data-augmentation strategies, and investigating model-distillation techniques could further enhance both performance and efficiency. Overall, our results support the idea that classical ML methods should not be dismissed outright in modern NLP workflows. They remain accessible, interpretable and surprisingly powerful.

References

Femi Emmanuel Ayo, Olusegun Folorunso, Friday Thomas Ibharalu, and Idowu Ademola Osinuga. 2020. Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. *Computer Science Review*, 38:100311.

Steven Bird, Ewan Klein, and Edward Loper. 2009. Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. Original-date: 2009-09-07T10:53:58Z.

Yulia Clausen, Tatjana Scheffler, and Michael Wiegand. 2025. Overview of the GermEval 2025 Shared Task on Candy Speech Detection. In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, Hildesheim, Germany. ACL.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for 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.

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2020. Training a Broad-Coverage German Sentiment Classification Model for Dialog Systems. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1620–1625, Marseille, France. European Language Resources Association.

Vaishali Kalra and Rashmi Aggarwal. 2018. Importance of Text Data Preprocessing & Implementation in RapidMiner. In *Proceedings of the First International Conference on Information Technology and Knowledge Management*, page 71–75.

Jan Ole Krugmann and Jochen Hartmann. 2024. Sentiment Analysis in the Age of Generative AI. *Customer Needs and Solutions*, 11(1):3.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1188–1196, Beijing, China. PMLR.

Ga Young Lee, Lubna Alzamil, Bakhtiyar Doskenov, and Arash Termehchy. 2021. A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance.

Peng Li, Xi Rao, Jennifer Blase, Yue Zhang, Xu Chu, and Ce Zhang. 2021. CleanML: A Study for Evaluating the Impact of Data Cleaning on ML Classification Tasks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 13–24.

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. *arXiv preprint arXiv:1907.11692*.

Vildan Mercan, Akhtar Jamil, Alaa Ali Hameed, Irfan Ahmed Magsi, Sibghatullah Bazai, and Syed Attique Shah. 2021. Hate Speech and Offensive Language Detection from Social Media. In *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, pages 1–5. IEEE.

Md Saef Ullah Miah, Md Mohsin Kabir, Talha Bin Sarwar, Mejdl Safran, Sultan Alfarhood, and M. F. Mridha. 2024. A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and LLM. *Scientific Reports*, 14(1):9603.

OpenAI. 2024. GPT-4 Technical Report.

Prabu Palanisamy, Vineet Yadav, and Harsha Elchuri. 2013. Serendio: Simple and Practical lexicon based approach to Sentiment Analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 543–548, Atlanta, Georgia, USA. Association for Computational Linguistics.

William Stigall, Md Abdullah Al Hafiz Khan, Dinesh Attota, Francis Nweke, and Yong Pei. 2024. Large Language Models Performance Comparison of Emotion and Sentiment Classification. In *Proceedings of the 2024 ACM Southeast Conference on ZZZ*, pages 60–68, Marietta GA USA. ACM.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Tong Zhan, Chenxi Shi, Yadong Shi, Huixiang Li, and Yiyu Lin. 2024. Optimization Techniques for Sentiment Analysis Based on LLM (GPT-3).

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023. Sentiment Analysis in the Era of Large Language Models: A Reality Check.

Seng Zian, Sameem Abdul Kareem, and Kasturi Dewi Varathan. 2021. An Empirical Evaluation of Stacked Ensembles With Different Meta-Learners in Imbalanced Classification. *IEEE Access*, 9:87434–87452.