HHUflauschig on Candy Speech Detection: Hybrid Approaches for Binary Classification and Span Typing

Wiebke Petersen, Lara Eulenpesch
Heinrich Heine Universität
Düsseldorf, Germany
{wiebke.petersen, lara.eulenpesch}@hhu.de

GermEval 2025 Shared Task on Candy Speech Detection, Hildesheim

Subtask 1: Task definition

Task:

- **Objective**: Identify whether a German YouTube comment contains *Flausch* (affectionate, positive language).
- **Task Type**: Binary classification.
- **Challenges**: Subjective definitions, informal language, spelling variation.

Subtask 1: System Overview

Data preparation:

- added spelling corrected comments
- added translations
- create held-out 10% split of training data for evaluation

Subtask 1: System Overview

Data preparation:

- added spelling corrected comments
- added translations
- create held-out 10% split of training data for evaluation

Approach:

- Hybrid architecture combining:
  - Linguistically motivated features
  - Fine-tuned transformer models
- Final prediction made by meta-classifier (logistic regression).

Fine-tuned LLMs

Results on held-out evaluation data

Model	Input	F1
gbert-large	original	**0.906**
gbert-large	spelling corrected	0.896
bert-base-german	original	0.885
bert-base-german	spelling corrected	0.880
roberta-large	translated	0.875

Results:
- original input > corrected text
- large models > base models
- German models > English model on translations

Features

- **Softmax scores of fine-tuned LLMs**: for original, spelling corrected and translated comments
- **Sentiment Polarity:** via `TextBlob` and `TextBlobDE`
- **Ekman's Emotions Scores:** via English translations and avaiable fine-tuned RoBERTa model
  (*anger, disgust, fear, happiness, sadness, surprise*)
- **Positive Lexicon features:**
  - Lists of positive words, tokens, emojis, emoticons (via ChatGPT-4o)
  - Tokens filtered out by frequency in non-Flausch comments
  - Absolute count and ratio features
- **Surface Features:**
  - Number of words with consecutive capital letters (2+)
  - Number of repeated characters (3+)

Results of meta-classifiers (logistic regression)

Results on held-out evaluation data

Features	**F1**	**Rec.**	**Prec.**
all non-BERT features	0.694	0.785	0.621
all BERT features	0.926	**0.944**	0.908
all features	0.932	0.936	0.927
winning configuration	**0.938**	0.929	**0.947**
`gbert-large` on orig.	0.906	0.881	0.932

winning configuration = `gbert-large` orig. comment + all sentiments (Ekman + polarity) + positive word count + positive token count + positive token ratio

Results on competition test data

winning configuration	0.887	0.900	0.875

BA-thesis of Eulenpesch (with improved features):

Results: all > BERT > positive counts > sentiments

Subtask 2: Task Definition

- **Goal:** Identify text spans expressing Flausch and assign one of 10 *Flausch-types*.
- **Evaluation:**
  - **Strict F1:** Correct span boundaries *and* correct type
  - **Span F1:** Span only
  - **Type F1:** Type only
- **Challenge:** Both accurate segmentation and subtle type classification

Subtask 2: System Overview

- We explored two paradigms:
  1. **End-to-End**: single model for joint span+type prediction (fine-tuned `gbert-large`)
  2. **Two-Step Pipeline**:
     - Step 1: span segmentation (rule-based or with LLM)
     - Step 2: type classification (with LLM)

Subtask 2: System Overview

- We explored two paradigms:
  1. **End-to-End**: single model for joint span+type prediction (fine-tuned `gbert-large`)
  2. **Two-Step Pipeline**:
     - Step 1: span segmentation (rule-based or with LLM)
     - Step 2: type classification (with LLM)
- **Best System:** Two-step pipeline based on `gbert-large` for segmentation and classification

Two-Step Pipelines: Span Segmentation

LLM Approach: Token-level BIO tagging with BERT

Rule-Based Approach: : We apply heuristics over SpaCy dependency trees:

Token 1 Token 2 Token 3 Token 4 Token 5

Two-Step Pipelines: Span Segmentation

LLM Approach: Token-level BIO tagging with BERT

Rule-Based Approach: : We apply heuristics over SpaCy dependency trees:

- for each token, we traverse upward until reaching a root, which is either the syntactic root, *reported speech* (rs), *coordinating conjunction* (cd), or *junctor* (ju).

Root A	Root A	Root B	Root B	Root C
↑	↑	↑	↑	↑
Token 1	Token 2	Token 3	Token 4	Token 5

Two-Step Pipelines: Span Segmentation

LLM Approach: Token-level BIO tagging with BERT

Rule-Based Approach: : We apply heuristics over SpaCy dependency trees:

- for each token, we traverse upward until reaching a root, which is either the syntactic root, *reported speech* (rs), *coordinating conjunction* (cd), or *junctor* (ju).
- Consecutive tokens sharing the same root form a span.

Root A	Root A	Root B	Root B	Root C
↑	↑	↑	↑	↑
Token 1	Token 2	Token 3	Token 4	Token 5

Two-Step Pipelines: Span classification

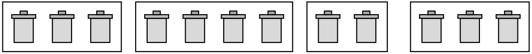
`gbert-large` fine-tuned to classify spans into

- 10 Flausch types (trained on typed flausch spans) or
- 10 Flausch types + not-Flausch class (trained on typed flausch spans + non-Flausch spans)

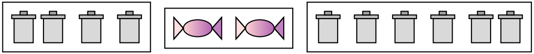
Two-Step Pipelines: Span classification

`gbert-large` fine-tuned to classify spans into

- 10 Flausch types (trained on typed flausch spans) or
- 10 Flausch types + not-Flausch class (trained on typed flausch spans + non-Flausch spans)
- non-Flausch spans are generated from
  - non Flausch comments using our SpaCy heuristics



  - Flausch spans by splitting at Flausch spans



Results

Results on held-out evaluation data

System	Strict	Span	Type
gbert-end-to-end	0.647	0.682	0.792
gbert 2-step	**0.728**	**0.769**	**0.833**
gbert 2-step + not-flausch	0.693	**0.769**	0.785
spacy 2-step	0.370	0.389	0.733

Results on competition test data

gbert 2-step	0.615	0.668	0.766

- additional `not-flausch` label for rejecting non-Flausch spans → no improvement
- rule-based approach does not identify correct spans

improved rule-based approach in BA-thesis (still weaker than `gbert 2-step`)

Some limitations

- **Data Split:** Held-out evaluation not stratified by video $\Rightarrow$ possible leakage

Some limitations

- **Data Split:** Held-out evaluation not stratified by video ⇒ possible leakage
- **Distribution shift between train and test set:**
  - comments in test set are longer (68.6 vs. 58.3 tokens)
  - comments in test set contain higher proportion of Flausch comments (41.3% vs. 29.1%)
  - comments in test set contain more annotated spans per comment (0.65 vs. 0.43).
  - test and train differ in span type distribution.


Thank you!
… and thanks to the organizers for such a sweet challenge!