KlarTextCoders at StaGE: Automatic Statement Annotations for *German Easy Language*

Akhilesh Kakolu Ramarao[1], Wiebke Petersen[1], Anna Sophia Stein[1], **Emma Stein**[2], Hanxin Xia[1]

{kakolura,petersew}@hhu.de, emma.stein@stud.uni-goettingen.de

13th September
KONVENS 2024

[1]Heinrich-Heine-Universität, Düsseldorf, Germany
[2]Georg-August-Universität, Göttingen, Germany

Overview

1. Introduction
2. Subtask 1: Determining the number of statements
3. Subtask 2: Identifying statement spans
4. Conclusion and Future work

Introduction

German Easy language

- Easy Language is a simplified linguistic form that excludes complex grammatical and lexical features (MaaSS and Bredel, 2017)
- 6.2 million Germans have low literacy levels (Buddeberg and Grotlüschen, 2020)
- Since 2011, governance bodies must include mandatory Easy Language information on websites (Bundesministerium des Innern und für Heimat, 2011)

Task

Statement Segmentation in German Easy Language (StaGE)

1. Predicting number of statements
2. Identifying statement spans

Subtask 1: Determining the number of statements

Overview

- Multi-faceted approach to statement segmentation task
- Combines rule-based, machine learning, and deep learning methods

Rule-based parser

- Built based on annotation guidelines provided by organizers
- Uses SpaCy's German language model (de_dep_news_trf)
- Performs token-level analysis of Parts of Speech (POS) tags, dependency labels, morphological features
- Handles special cases like parentheses, adjectives, prepositional phrases
- Groups tokens into clauses based on dependencies and POS tags

Machine learning classifiers

Features extracted

- Syntactic features: Dependency trees, Parts-of-speech
- Semantic features: Abstract Meaning Representation (AMR). AMR captures the core meaning of sentences, abstracting away from syntactic variations.

  'Viele Zeitungen haben über den Hund berichtet.'
  'Many newspapers have reported about the dog.'

```
(r / report-01
      :ARG0 (n / newspaper
            :quant (m / many))
      :ARG1 (d / dog))
```

- Additional linguistics features: Number of tokens, Number of adjectives/propositions and so on

Machine learning classifiers

Features extracted

- Length features most important (AMR length, dependency chain length, token count)
- BERT features (all 10 UMAP dimensions) in top 20 most important features
- Dependency tree and AMR features are less important

Machine learning classifiers

Classifiers

- Random Forest (RF), Support Vector Machine (SVM), Multilayer Perceptron (MLP), Logistic Regression (LR)
- Random Forest classifier with all features performed best on the eval set. MAE: 0.28, Accuracy: 0.75
- Data augmentation did not improve results
- Combining diverse feature types (syntactic, semantic, embeddings) beneficial

LLM prompting

- Llama model for counting the number of statements
- Iterative prompt design
- 66.83% accuracy on the test set, primarily due to overpredicting sentences with one statement
- Struggled with predicting statement spans due to complexity

Submission

- Pre-trained German BERT model (SpaCy *de_core_news_lg*[1]) for contextual embeddings
- Extend BERT with parts-of-speech (POS) information
- Model components:
  - Pre-trained German BERT model
  - POS Encoder
  - Classifier
- POS encoder using one-hot encoding and linear layer
- Classifier combines BERT outputs with encoded POS features

[1] https://spacy.io/models/de

Submission

Forward Pass

- Extract [CLS] token representation
- Get POS tags using SpaCy
- Encode POS tags using one-hot encoding and linear layer

Training Details

- Combine trial, train, and test datasets
- Remove zero-statement sentences
- Split dataset into 90% training and 10% test datasets
- Train model for 100 epochs with batch size of 16

Submission

Model Performance

- Accuracy on test set: 80%
- Precision: 0.65
- Recall: 0.68
- F1-score: 0.65

Subtask 2: Identifying statement spans

Token classification task

- Transform attention span labels into token-level classifications
- Use tokenized phrase and assign labels to each token
- Example:
  Convert attention span [[0, 1, 4], [3]] to [1, 1, 0, 2, 1]
- Use fine-tuned BERT model from subtask 1
- Further fine-tune for token classification task
- Leverage transfer learning from statement-level classification

Model Overview

Model Architecture

- Retain pre-trained BERT layers
- Add part-of-speech (POS) encoder for syntactic information
- Implement new classification layer for token-level predictions

Forward Pass

- Process input through BERT to get the last hidden state
- Encode POS tags using POS encoder
- Concatenate outputs along last dimension
- Pass combined representation through classifier

Model Overview

Training

- Use 90% train and 10% test split.
- Train model for 10 epochs with batch size of 16

Model Performance

- chrF score: 0.36
- Jaccard similarity: 0.29

Conclusion and Future work

Conclusion

- BERT with POS information effective for both subtasks
- Exploration of various methods provided insights into strengths and limitations
- Statement segmentation in German Easy Language is a challenging task
- Continued exploration of different approaches is necessary for improved performance

Future work

- Postprocessing to minimize the divergence between the predictions of subtask 1 and 2
- Investigate multi-task learning approach for improved performance
- Compare performance of BERT with other pre-trained language models
- Explore use of shared linguistic knowledge across tasks

Resources

- Slides: https://akkikek.xyz/presentations/konvens24.pdf
- Codebase: https://github.com/ansost/easy-to-read [WIP]
- Models: https://huggingface.co/akki2825/klartextcoder

References

Klaus Buddeberg and Anke Grotlüschen, editors. 2020. *LEO 2018: Leben mit geringer Literalität*. wbv Publikation, Bielefeld.

Bundesministerium des Innern und für Heimat. 2011. Barrierefreie-informationstechnik-verordnung 2.0. https://www.barrierefreiheit-dienstekonsolidierung.bund.de/Webs/PB/DE/gesetze-und-richtlinien/bitv2-0/bitv2-0-artikel.html. accessed: 17.04.2024.

Christiane MaaSS and Ursula Bredel. 2017. *Ratgeber Leichte Sprache: Die wichtigsten Regeln und Empfehlungen für die Praxis*. Duden. ISBN: 9783411912360.

Appendix

LLM prompting

System prompt

You are an expert in German Easy Language.

User prompt

Give the statement spans of the sentence below.
For your decisions rely on the annotation guidelines provided below.
Provide your output in the form of a nested list. Return nothing but that list or the string “None” if the sentence only has one statement or zero statements.
Think step by step.
Three example sentences:
sentence: Eine sehr bekannte Alchemisten war Maria die Jüdin
statements: 1, statement spans: None;
(We provided two more examples but did not include them here for the sake of brevity.)

The sentence you should annotate is the following:

{sentence}

Annotation guidelines:

{annotation_guidelines}

AMR: abstract meaning representations

‘Viele Zeitungen haben über den Hund berichtet.’
‘Many newspapers have reported about the dog.’

```
(r / report-01
      :ARG0 (n / newspaper
            :quant (m / many))
      :ARG1 (d / dog))
```

extracted AMR-features:

attr-arg0:, attr-arg0:attr-instance:, attr-arg0:attr-instance:newspaper,
attr-arg0:, attr-arg0:attr-quant:, attr-arg0:attr-quant:attr-instance:,
attr-arg0:attr-quant:attr-instance:many,
attr-arg1:, attr-arg1:attr-instance:, attr-arg1:attr-instance:dog,
attr-instance:, attr-instance:report_01