Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 7102–7111
Marseille, 11–16 May 2020
© European Language Resources Association (ELRA), licensed under CC-BY-NC

Improving Sentence Boundary Detection for Spoken Language Transcripts

Ines Rehbein, Josef Ruppenhofer, Thomas Schmidt

Data and Web Science Group
University of Mannheim
Mannheim, Germany

Archive for Spoken German
Leibniz Institute for the German Language
Mannheim, Germany

ines@informatik.uni-mannheim.de, {ruppenhofer|thomas.schmidt}@ids-mannheim.de

Abstract

This paper presents experiments on sentence boundary detection in transcripts of spoken dialogues. Segmenting spoken language into sentence-like units is a challenging task, due to disfluencies, ungrammatical or fragmented structures and the lack of punctuation. In addition, one of the main bottlenecks for many NLP applications for spoken language is the small size of the training data, as the transcription and annotation of spoken language is by far more time-consuming and labour-intensive than processing written language. We therefore investigate the benefits of data expansion and transfer learning and test different ML architectures for this task. Our results show that data expansion is not straightforward and even data from the same domain does not always improve results. They also highlight the importance of modelling, i.e. of finding the best architecture and data representation for the task at hand. For the detection of boundaries in spoken language transcripts, we achieve a substantial improvement when framing the boundary detection problem as a sentence pair classification task, as compared to a sequence tagging approach.

Keywords: Spoken language transcripts, sentence boundary detection, corpus creation

1. Introduction

Being able to structure natural spoken discourse into sentence-like units (SLUs) is desirable not only from a theoretical point of view, but is also a key requirement for enabling research in corpus linguistics as well as the application of Natural Language Processing tools (e.g. PoS-taggers, syntactic parsers) to transcripts of spoken language.

While various proposals have been made for how to divide spoken language in corpora into smaller units, typically these divisions were not guided by syntactic considerations. Instead, division into inter-pausal units is common (e.g. Hamaker et al. (1998) for the Switchboard corpus (John J. Godfrey, Edward Holliman, 1993)).

For German, the SegCor project presented a proposal and guidelines for dividing transcribed speech into sentence-like units based on Topological Fields (Westpfahl and Gorisch, 2018; Westpfahl et al., 2019). The Topological Fields Model (Drach, 1937; Höhle, 1986) is a descriptive grammar formalism that captures regularities in German word order by dividing sentences in different verbal and non-verbal fields and describing their position with regard to the main verb. In a corpus-based study, Schmidt and Westpfahl (2018) then investigated how well the length of gaps between utterances can predict the syntactic boundaries annotated in the SegCor corpus. They showed that while there is a correlation between gap length and surface syntax, gap length on its own is not sufficient for a reliable prediction of SLU boundaries.

Our work builds on previous work on automatic boundary detection in German spoken language transcripts (Ruppenhofer and Rehbein, 2019) and tries to further improve the accuracy for SLU boundary detection. Ruppenhofer and Rehbein (2019) modelled the problem as a sequence tagging task and showed that neural models with contextual string embeddings (Akbik et al., 2018), based on the Flair library of Akbik et al. (2019), outperform a classical feature-based CRF classifier. This paper presents new experiments where we (i) test different neural architectures and task setups for SLU boundary detection, and (ii) investigate the potential benefits of additional training data from a different source of spoken language.

This paper proceeds as follows. We discuss related work in Section 2. and present our dataset in Section 3. Our experiments and their results are described in sections 4. and 5. While we show that training data expansion for this task is not straightforward even with data from the same domain (Section 4.), we present substantial improvements for SLU boundary detection when framing the task as sentence pair classification (Section 5.2.). We conclude and outline avenues for future work in Section 6.

2. Related Work

In this section, we report on previous work on sentence boundary detection in written language and on the detection of sentence-like units in spoken language.

2.1. SLU detection in written text

In the realm of medially written language, the most closely related task is sentence boundary detection. Typically, this has been framed as deciding for a closed class of interpunctuation symbols (mainly '.','?','!') whether they represent the end of a sentence or not, with abbreviations constituting one of the key sources of error. While traditionally very high accuracies were reported, Read et al. (2012) show in their overview of SLU detection that performance can be significantly worse on text other than news, with machine learning-based systems often being less robust than rule-based or hybrid sytems. Comparing Wikipedia pages to topically related blogs, they also show that within the same domain, sentence-boundary detection performs less well the more informal the text type is.

Recently, sentence boundary detection has also come into focus due to the rise of social media, which often include text where standard punctuation conventions are both ignored and/or extended. Posts on Twitter or Facebook may, for instance, 'end' a sentence with an emoticon, an asterisk or a pipe symbol rather than a punctuation symbol. Rudrapal et al. (2015) test the limitations of rule-based sentence boundary detection and investigate three machine learning algorithms as possible alternatives.

2.2. SLU detection in spoken language transcripts

In the domain of medially spoken language, the detection of sentence-like units is a much harder problem, given the lack of punctuation and case information, and the high number of disfluent utterances.

Stevenson and Gaizauskas (2000) investigated the upper bound for human performance on such data and measured agreement for inserting punctuation (also including commas) in transcriptions of a BBC news program. Instead of transcriptions created by an ASR system, they used human-created transcription but removed punctuation and case information. On that data, human annotators showed a precision in the range of 84-93%. Recall, however, was much lower with 68-78%.

Westpfahl and Gorisch (2018) measured human agreement for the syntactically motivated SegCor segmentation scheme that also distinguished different sentence types (see Section 3.). They report an average kappa of 0.69 for the agreement of two annotators for the segmentation across 8 transcripts. While Westpfahl and Gorisch (2018) give no breakdown of which confusions among boundary types are most frequent for their human annotators, they do show a further complication of the task: the different sentence types are distributed differently across different text types and their specific properties also vary by text type. For instance, in so-called expert talk, simple sentences are longer than in other texts.

Taken together, these experiments underline the challenge in the task we tackle by showing that sentence boundary labeling cannot be done perfectly by humans and that its diffculty is variable across text types.

To make up for missing punctuation and case information, some studies have made use of both prosodic features to augment the lexical information from the transcripts. Gotoh and Renals (2000) performed experiments with HMMs on reference transcripts from BBC radio and tv programs which included repeated and incorrect speech as well as disfluencies. They also constructed an alternative pause duration model alone based on speech recogniser output aligned with the transcripts. The pause duration model outperformed the language modelling approach, while a combination of the two models provided further performance gains. Precision and recall scores of over 70% were attained for the task of deciding for each word whether it represents the last word of a sentence. In his work on sentence boundary detection on Czech radio news and discussion programs, Kolář (2008) similarly finds that combining several models works best.

Liu et al. (2005) evaluate the performance of a CRF-model on two English corpora (conversational telephone speech and broadcast news speech) on both human transcriptions and automatic speech recognition output. Their experiments show that the use of prosody improves performance over the use of word n-grams alone and that the addition of further features e.g. on pos-tags provides another improvement.

Roark et al. (2006) apply a re-ranking approach to the detection of SLU boundaries. In a two stage approach, they first fix a subset of the word boundaries as points of division, yielding subsequences betwen fixed points, which they call fields. In the second stage, candidate boundaries within the fields are generated and then ranked.

Zribi et al. (2016) predict sentence boundaries in transcriptions of spoken Tunisian Arabic. Their best system combines a rule-based approach with partial decision trees (PART) and achieves an F1 of around 82% on their data. Importantly, their results show that automatic sentence boundary detection can improve the accuracy of a PoS tagger for transcribed Tunisian Arabic.

In previous work (Ruppenhofer and Rehbein, 2019), we have experimented with various features and task parameters, showing that the right context is far more important for SLU detection than the left context, and that information on speaker turns considerably improves results. We experimented with a feature-rich classification setup based on Conditional Random Fields (CRF) that allowed us to easily include additional information, such as PoS tags or lemmas. However, we also showed that the CRF classifier can be outperformed by a simpler neural model that incorporates contextualised string embeddings (Akbik et al., 2018; Akbik et al., 2019). Given the success of the neural model, we would like to test whether further improvements can be obtained with transfer learning based on BERT's contextualised word embeddings (Devlin et al., 2019) (Section 5.).

3. Data

This section presents our gold standard for the segmentation of German oral corpora, created in the SegCor ("Segmentation of Oral Corpora") project,[1] as well as the additional spoken language data we use in our training data expansion experiments.

3.1. SegCor

The SegCor data has some features that distinguish it from most previous work. Our data represents conversational speech with two or more speakers that was recorded in non-laboratory settings. Since tools based on the automatic processing of the audio signal do not work all that well on our data, we instead work with the transcripts only. Our dataset consists of 33 documents with more than 54,000 lexical tokens originating from the FOLK corpus (Schmidt, 2014) that were divided into sentence-like units by the SegCor project. This data set was doubly annotated and disagreements were adjudicated (Westpfahl and Gorisch, 2018). Note that to avoid confusion, we reserve the term *segment* and related forms for the division of speech into chunks by the transcribers that was guided by silences in the speech signal. For the division of the material into *sentence-like units* we will use the term "SLU boundary detection".

[1] https://www1.ids-mannheim.de/prag/muendlichekorpora/segcor.html

The raw FOLK transcripts, which we take as our input and which lack SLU-boundaries, follow the cGAT conventions (Schmidt et al., 2015). Accordingly, the data uses "contributions" and "segments" as the fundamental units in the data structure. Segments of speech are the original units of transcription: transcribers are instructed to select them as chunks that can be transcribed in one go given cognitive load and usability of the transcription environment. Crucially, segment boundaries should be placed at word boundaries or at the beginning or end of pauses. Like segments, contributions are defined without any reference to syntactic considerations (Schmidt et al., 2015, 8):

> 'A contribution in a cGAT transcript comprises all immediately consecutive segments attributed to a speaker. Contributions should not be confused with sentences, which are units of written language. Instead, they are to be understood as dialogue contributions.

`Pauses` (silences up to 0.2s) may occur between separate contributions but also within a contribution. `Gaps`, silences longer than 0.2s, always separate contributions in cGAT.

The relation between the input representation in terms of contributions and the intended output representation in terms of sentence-like units is not always one to one. Common deviations are as follows. First, a contribution may correspond to several SLUs as illustrated by (1).

(1) 1 contribution : n SLUs

a. $< c >$h ich weiß net ich glaub eher nich h h$< /c >$

b. $< SLU >$h ich weiß net$< /SLU >$
$< SLU >$ ich glaub eher nich h h$< /SLU >$

c. 'I don't know. I rather think not.'

Second, several contributions may jointly correspond to one SLU.

(2) n contributions : 1 SLU

a. $< c >$der beschäftigt sich$< /c >$
$< c >$(0.85) $< /c >$
$< c >$zwei minuten mit dem$< /c >$

b. $< SLU >$ der beschäftigt sich (0.85) zwei minuten mit dem $< /SLU >$

c. 'He occupies himself with that one for two minutes.'

Both situations may also occur in combination so that we get $n : m$-relations between contributions and SLUs.

To decide on SLU boundaries, we can not only make use of the transcribed word forms but can also include further information. While we do not use acoustic features such as word durations and pitch contours, the transcript does give us access to temporal information that has proved useful in previous work (Gotoh and Renals, 2000). We encode pause length and, since we know which tokens are produced by which speaker, we also introduce turn boundaries into our representation.

3.2. KiDKo

In addition to the rather small SegCor dataset we also have access to a much larger corpus of informal spoken youth language, the KiezDeutsch-Korpus (KiDKo) (Wiese et al., 2012; Rehbein et al., 2014). KiDKo contains spontaneous peergroup dialogues of adolescents from multiethnic Berlin-Kreuzberg (around 266,000 tokens) and a supplementary corpus with adolescent speakers from monoethnic Berlin-Hellersdorf (around 111,000 tokens, excluding punctuation). On the normalisation layer where punctuation is included, the token counts add up to around 359,000 tokens (main corpus) and 149,000 tokens (supplementary corpus).

On the normalisation layer, the data includes punctuation and is segmented into sentence-like units. On top of the normalisation, the corpus comprises additional annotation layers with Parts of Speech (PoS), syntactic chunks and Topological Fields (Drach, 1937; Höhle, 1986).

As the segmented utterances in KiDKo serve as the basis for the Topological Field annotations, both corpora have been segmented according to similar principles. However, some differences remain. One of them concerns coordinated clauses where each conjunct includes a finite verb. These coordinated sentences are split into seperate clauses in SegCor (3). In KiDKo, the decision whether to segment coordinated sentences or not is left to the transcribers who have access to the audio file and decide for each individual utterance, based on prosodic and semantic cues. As a result, a coordinated clause might either be separated as in (3) or might be left as one complex clause, as in (4).

(3) $< SLU >$sie trinkt kaffee$< /SLU >$
"she drinks coffee"
$< SLU >$ und er trinkt tee $< /SLU >$
"and he drinks tea"

(4) $< SLU >$sie trinkt kaffee und er trinkt tee $< /SLU >$
"she drinks coffee and he drinks tea"

To be able to use the corpus as additional training data for SLU detection in the SegCor corpus, we segmented all coordinated sentences in KiDKo not containing an ellipsis into seperate sentences, as in (3). We also merged tag questions, which are tagged as a separate SLU in KiDKo (5), with their preceeding sentence, as was done in SegCor (6).

(5) $< SLU >$ gurke is n gemüse$< /SLU >$
"cucumber is a vegetable"
$< SLU >$ oder $< /SLU >$
"right"

(6) $< SLU >$ gurke is n gemüse oder $< /SLU >$
"cucumber is a vegetable right"

Other adaptations concern the transcription of hesitation particles and interjections, where we converted frequent word forms so that they followed the conventions used for annotating SegCor.

Table 1 gives an overview of the two datasets after the conversion and Table 2 shows the distribution of SLU types in

	# tokens		# SLU	
	SegCor	**KiDKo**	**SegCor**	**KiDKo**
train	38,293	230,166	7,756	61,524
dev	5,578	33,265	1,213	8,985
test	10,841	65,845	1,771	16,634
total	**54,712**	**329,276**	**10,740**	**87,143**

Table 1: Statistics for train/dev/test data from SegCor and KiDKo (no. of tokens and no. of sentence-like units).

the two datasets.[2] While the general trend looks similar, we can see that KiDKo has a much higher number of non-sentential SLUs than SegCor. SegCor, on the other hand, has a larger number of non-boundary tokens. This confirms that many of the non-sentential units in KiDKo are short answers (*yes, no, ok* etc.). SegCor also has a higher percentage of pauses. This is due to the fact that SegCor transcribers also recorded micropauses shorter than 2 milliseconds while in KiDKo these micropauses have not been transcribed.

Overall, the distribution in the two corpora seems to be similar enough to suggest that KiDKo is a suitable dataset for our data expansion experiments. We thus expect to see an increase in results for SLU boundary detection on the SegCor testset when training on the combined training sets from both corpora.

4. Training Data Expansion

In our first set of experiments we want to test whether additional training data can boost results for SLU detection in SegCor. Given that our supplementary dataset was created in a similar fashion, namely spoken multi-party dialogues recorded in non-laboratory settings, we expect to see an improvement when adding the KiDKo data to the SegCor training set.

We first run experiments in a classical feature-based setting with a Conditional Random Fields (CRF) classifier. Even though the CRF was outperformed by a neural sequence tagger with contextual string embeddings (Akbik et al., 2018; Akbik et al., 2019) in our previous work, the short training times for the CRF allow us to run many experiments in a short period of time in order to explore the potential of the training data expansion approach and to compute learning curves.

We split the SegCor training data into 10 samples of equal size and train 10 separate classifiers, the first on sample 1, the second on samples 1 and 2, the third on the first three samples, and so on. Sample 10 accordingly includes all data from the SegCor training set. Next, we also split the KiDKo training data into samples of the same size as the SegCor samples and add them incrementally to the training set. We evaluate each model on the SegCor and KiDKo test sets, separately, and report F1 for each class (**B**oundary, n**O**n-boundary), as accuracies on the highly imbalanced datasets are not very informative.

SLU type	**KiDKo (%)**	**Segcor (%)**
uninterpretable	0.07	0.26
aborted	0.20	1.39
complex	2.00	1.75
simple	8.89	5.30
pauses/turns	10.10	18.02
non-sentential	19.65	7.42
no boundary	59.08	65.86

Table 2: Distribution of SLU types in the two corpora.

Table 3 and Figure 1 show F1 scores and learning curves for the CRF models trained on data of increasing size. We can see that the KiDKo test set seems to be somewhat easier to segment, with results being nearly 3% higher than the ones for the SegCor test set when training on all 10 samples from SegCor. When adding additional samples from KiDKo (samples 11-29), the learning curve for the KiDKo test set rises steeply, with final results of 90% F1(B) being more than 10% higher than the ones for the SegCor test set. This gap in performance might be partly due to the greater number of short answers in KiDKo that are easy for the classifier to predict. However, we also observe a performance gap for longer sentence-like units. It is not obvious

	SegCor			**KiDKo**		
Sample	**F1**	**F1 B**	**F1 O**	**F1**	**F1 B**	**F1 O**
1	83.4	70.8	96.0	85.8	76.7	95.0
2	84.1	72.0	96.1	87.4	79.4	95.4
3	84.8	73.5	96.2	87.4	79.4	95.4
4	86.2	76.0	96.5	88.1	80.6	95.6
5	86.6	76.6	96.5	88.5	81.3	95.7
6	87.2	77.6	96.7	88.2	80.7	95.6
7	87.5	78.3	96.8	88.6	81.5	95.8
8	87.6	78.5	96.8	89.0	82.1	95.9
9	87.9	79.0	96.8	88.9	81.9	95.8
10	88.0	**79.1**	96.9	88.9	81.9	95.8
11	87.9	79.0	96.8	90.3	84.2	96.3
12	87.8	78.9	96.8	91.1	85.6	96.6
13	87.6	78.5	96.8	91.3	86.0	96.6
14	87.7	78.6	96.8	91.6	86.5	96.8
15	87.7	78.7	96.8	91.9	86.9	96.8
16	87.7	78.6	96.8	92.3	87.6	97.0
17	87.9	78.9	96.8	92.7	88.3	97.1
18	87.8	78.8	96.8	92.8	88.5	97.2
19	87.7	78.6	96.8	92.9	88.6	97.2
20	87.7	78.7	96.8	93.0	88.7	97.2
21	87.8	78.8	96.8	93.1	89.0	97.3
22	87.8	78.8	96.8	93.3	89.2	97.3
23	88.0	**79.1**	96.8	93.4	89.4	97.4
24	87.9	79.0	96.8	93.5	89.5	97.4
25	87.8	78.8	96.8	93.5	89.5	97.4
26	87.7	78.7	96.8	93.5	89.6	97.4
27	87.9	**79.1**	96.8	93.6	89.8	97.5
28	87.9	79.0	96.8	93.7	89.9	97.5
29	87.8	78.8	96.8	93.8	**90.0**	97.5

Table 3: CRF: results for training on samples from SegCor of increasing size (samples 1-10), with additional training data from KiDKo (samples 11-29).

[2] Both corpora encode additional information either on the level of PoS or chunks/topological fields that allows us to retrieve this statistic.

whether this gap is caused by annotation inconsistencies in the dataset or whether the content and interaction types of the conversations in SegCor are, in fact, harder to segment.

We tested an additional setting where we filtered out those training instances in KiDKo that were most unsimilar to the ones in the SegCor dataset. This was done with the help of a language model (LM) that was trained on PoS sequences in SegCor. We then ranked the instances in KiDKo by their perplexity per PoS tag sequences, according to the LM. Then we removed those instances that had the highest perplexity, thus the ones being the least similar to the SegCor data.[3] We experimented with removing different portions of KiDKo and obtained small improvements. However, none of the settings we tried managed to improve results over training only on the SegCor data, thus showing that it is hard for the classifier to learn new information from KiDKo that is useful for SLU detection on SegCor.
This is evidenced by the learning curves (Figure 1). The curve for SegCor levels out after the first nine samples and shows only a very slight improvement for sample 10 which achieves the highest F-score on the SegCor testset. After that, results do not improve further and even decrease for most samples. For KiDKo, however, the learning curves show small but steady improvements until the end. Here we obtain our best result of 90% F1(B) when training on all samples from both corpora.
While it is not surprising that results on the KiDKo testset improve when adding training data from the *same* corpus (i.e. in-domain data), we were surprised that the large KiDKo training set did not help at all to improve results for SLU detection in SegCor.
The learning curves cast some doubt on whether annotating more training data from the same domain might help to improve results on the SegCor data, given that we only see a very slight improvement from sample 9 to 10. This finding is consistent with previous experiments where we added new transcripts from the SegCor corpus but did not see any improvements.[4] Therefore, in our next set of experiments, we focus on testing different architectures and representations for improving SLU detection in spoken language transcripts, based on multi-layer bi-directional transformers.

5. SLU detection with BERT

Recently, transformers have pushed the state of the art for many NLP applications by learning context-sensitive embeddings with different optimisation strategies and then fine-tuning the pre-trained embeddings in a task-specific setup. The BERT embeddings have been trained on large datasets by incorporating word embeddings with positional information and self-attention in different tasks, i.e. by predicting masked words based on their left and right context and by classifying two sentences based on how probable it is that the second one immediately succeeds the first one in a text document. As a result, the learned embeddings encode the left and right context for each word which makes them superior to previous representations for many NLP tasks.

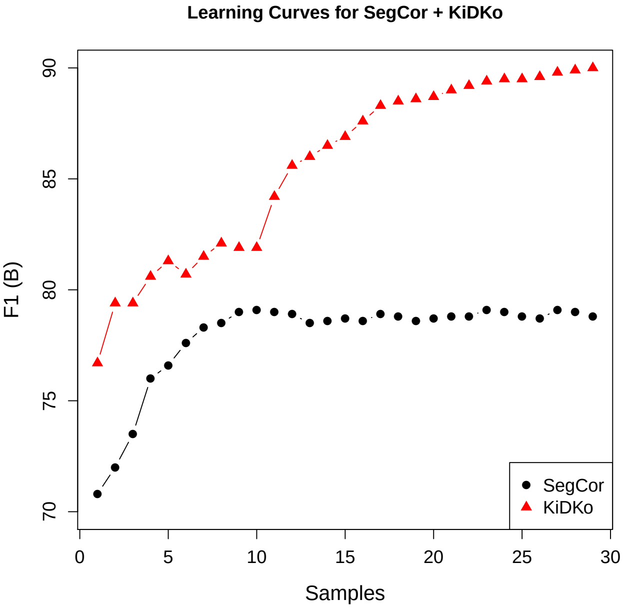

Figure 1: CRF: learning curves (F1 B) for SegCor when using additional training data from KiDKo coarse-grained SLU detection (first 10 samples are from SegCor, samples 11-29 from KiDKo).

Given the tremenduous success of the transformer model, it seems only natural to test pre-trained BERT embeddings in the sentence boundary detection task. However, there are two potential problems for using BERT for SLU detection on spoken language transcripts. First, most available transformer models have been trained on large amounts of written text, and we therefore might expect a high OOV rate for non-normalised transcribed spoken language. Second, it is not clear how to represent the input data when fine-tuning the model, given that we do not have sentence boundaries in the first place.
The most straightforward way to fine-tune BERT for our task is to model SLU detection as a sequence tagging problem as we have done before, but this time replacing the character-based contextual string embeddings of Akbik et al. (2018; Akbik et al. (2019) by the BERT embeddings.

5.1. SLU detection as sequence tagging

In our experiments, we use the HuggingFace transformers library (Wolf et al., 2019) that provides pre-trained transformer models for different languages and tasks. As our input transcripts are lower-cased, we use the pre-trained German uncased BERT model (bert-base-german-dbmdz-uncased).[5]

[3] A similar method was used successfully in Søgaard (2011) for cross-lingual unsupervised parsing, and in Rehbein (2011) for self-training of monolingual parsers.

[4] These transcripts, however, had only been annotated by one annotator so it was not clear whether the results might reflect a lower quality in the annotations.

[5] The model has been trained by the MDZ Digital Library team (dbmdz) at the Bavarian State Library on 2,350,234,427 tokens of raw text, including Wikipedia, the EU Bookshop corpus, Open Subtitles, CommonCrawl, ParaCrawl and News Crawl. For details see `https://github.com/dbmdz/german-bert`.

O	was	bei	muttersprachlern	untersucht	wird
G	what	in	native speakers	investivated	will
B	was	bei	mutter ##sprach ##lern	untersucht	wird
E			*"that's studied in native speakers"*		

Figure 2: Subword tokenisation by BERT (bert-base-german-dbmdz model; O: orig. transcript, G: english gloss, B: BERT tokenisation, E: English translation).

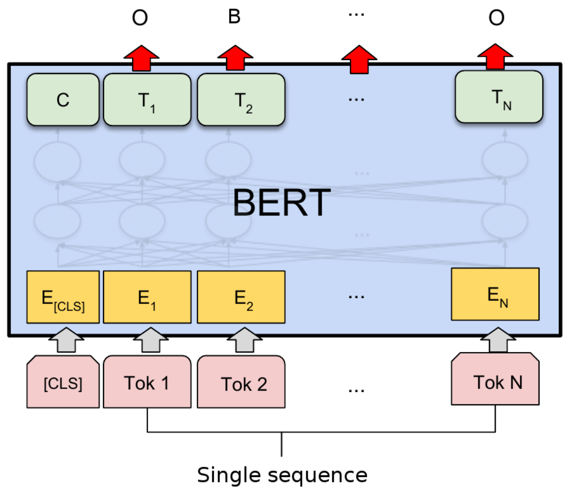

Figure 3: Fine-tuning BERT for sequence tagging tasks.

		Macro Acc	Macro F1	F1 B	F1 O	Embeddings/ Features
ID				*BERT sequence tagging*		
1		95.3	90.3	83.4	97.3	dbmdz- german- uncased
2		95.1	89.6	82.0	97.2	
3		95.2	89.7	82.2	97.2	
4		95.0	89.5	81.8	97.1	
5		95.1	89.6	82.0	97.1	
avg.		**95.1**	**89.7**	**82.3**	**97.2**	
CRF		94.7	88.3	79.7	96.9	+/-2 word,pos
Flair		92.3	83.4	71.3	95.5	FastText (FT)
		95.1	89.6	82.0	97.1	FT+Flair
		94.8	89.3	81.6	97.0	FT+custom
		95.4	90.2	**83.1**	97.4	FT+Flair+custom

Table 4: Results for SLU detection as a sequence labelling task (training/dev/test data from SegCor). Baseline results for CRF + Flair are from Ruppenhofer & Rehbein (2019) on the *same* dataset.

One limitation of BERT is the sequence length constraint, a parameter set during pretraining, that restricts the length of the input sequences for the pre-trained model to 512 tokens. While this still seems to be quite long, in practice BERT provides its own tokenisation of the input text into subtokens that results in much higher token counts per sequence, as compared to the original sequence length (see the example in Figure 2 above). Because of this, we have to reduce the length for our input sequences that was set to five merged speaker contributions in previous experiments (for a discussion of different representations of spoken language transcripts for ML, see Ruppenhofer and Rehbein (2019).

To accomodate BERT's length restrictions, we extract sequences with a maximum length of 80 tokens as follows. We iterate over each token in the input data, looking for probable starting points of new utterances. We consider pauses, speaker turns or discourse markers such as 'also' (well) as probable starts.[6] For each potential starting point, we then extract at least 10 tokens to the right (or, if there are less than 10 tokens, we add all of them). This gives us sequences with a length of at least 10 tokens. Then we extend this sequence by adding additional tokens from the right context, up to the next pause, speaker turn or discourse marker, again assuming that those are probable SLU boundaries. Finally, we make sure that the length of the extracted sequence does not exceed 80 tokens. If the sequence is too long, we cut off after the first 20 tokens.

This procedure ensures that our input sequences have a length of at most 80 tokens. It also means that during training and test, some tokens are presented to the tagger more than once, but with different context window sizes. To obtain the final prediction for each token, we collect all predictions made by the tagger and use a simple majority vote to determine the final label.

Once we have extracted our input sequences from SegCor, we train the sequence tagger provided by the HuggingFace library with a batch size of 32 and a learning rate of 5e-5 for three iterations on our data.[7] The sequence tagging model was originally intended for NER and similar tasks. In our setup, however, we input an unsegmented sequence from our spoken transcripts and let the model predict for each token whether or not this token is followed by an SLU boundary. Figure 3[8] illustrates the BERT model for the sequence tagging task.

Results on the SegCor data for this architecture are shown in Table 4. When comparing our results to our previous work on the same dataset (Table 4, lower part), we see that the BERT model outperforms all previous models (CRF, Flair-FastText, Flair-FastText+Flair embeddings) except for the Flair model that was trained with our own customised embeddings (Flair-FastText+Flair+custom), in addition to the FastText and Flair embeddings provided by the Flair library (Akbik et al., 2019).

These custom embeddings are character-based Flair embeddings that have been trained on ca. 11 million 'sentences' extracted from the open subtitles corpus (Lison and Tiedemann, 2016) and an in-house twitter dataset. All sentences with a length $>$ 60 characters have been removed, as have sentences that contained more than one comma and one period, question mark or exclamation mark. The punc-

[6] The disambiguation between discourse markers and other adverbial forms was done based on automatically predicted fine-grained PoS (Westpfahl and Schmidt, 2016).

[7] We also experimented with other learning rates which gave inferior results on the development set.

[8] Taken from Devlin et al. (2019) and adapted to our setup.

tuation marks were removed before training and the data was lowercased. The motivation for this setup was to train embeddings on text that is more similar to our spoken language transcripts.

As shown above, our BERT sequence tagging model achieves results in the same range as the model stacked with the original FastText, Flair and our Flair-custom embeddings, with considerably shorter training times. It does, however, fail to outperform our best previous model (FT+Flair+custom). We thus decided to test a different BERT architecture where, instead of modelling SLU detection as sequence tagging, we present the model with two separate strings and fine-tune it on the task of predicting whether or not there should be a SLU boundary at the end of the first string.

5.2. SLU detection as sentence pair classification

Devlin et al. (2019) present a BERT model for Recognising Textual Entailment (RTE), framed as a sentence pair classification task where the model is presented with two input sentences and learns to predict a label, i.e. whether the second sentence is entailed by the first one or not. The model architecture uses pre-trained BERT embeddings to encode the two sentences, separated by a special token $E_{[SEP]}$ (Figure 4). In addition, the model encodes on a separate segmentation embedding layer for each token whether it belongs to the first or to the second sentence. The resulting representation is then fed to a logistic regression that predicts the class label. It has been shown that this model outperforms previous models for RTE (Devlin et al., 2019) and also for similar semantic tasks such as Multi-Genre Natural Language Inference (MNLI).

We hypothesize that explicitly representing the left and right context for each potential SLU boundary as two separate sequences to the model might make it easier for BERT to learn relevant features for SLU detection from the data.

Thus, in our next experiment, we model SLU detection as a sentence pair classification task. As we do not have sentence boundaries in the first place but our goal is to learn them, we extract the input sequences for the model in a similar fashion as before (Section 5.1.). We iterate over each token t in the input corpus and extract a training (dev/test) instance where the task is to predict whether t is followed by an SLU boundary (Table 5). For each token t, we thus extract two separate sequences where t is the last token of input sequence 1 while input sequence 2 encodes the right context of token t.

CLASS	Sentence 1	Sentence 2
O	TURN	äh so um die ja äh PAUSE ...
O	TURN äh	so um die ja äh PAUSE ...
O	TURN äh so	um die ja äh PAUSE ...
O	TURN äh so um	die ja äh PAUSE ...
B	TURN äh so um die	ja äh PAUSE ...
B	TURN äh so um die ja	äh PAUSE ...
Engl.:	*uh so about the yes*	*uh*

Table 5: Generation of training instances for SLU detection as sentence pair classification.

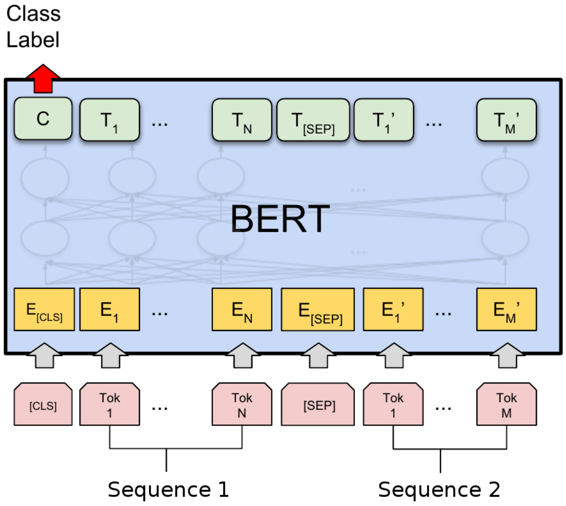

Figure 4: Fine-tuning BERT for sentence pair classification.

We extract the left context for token t by adding at least 10 tokens to the left of t to sequence 1. After adding the 10th token, we check, for each additional token, whether it is a pause or new speaker turn and, that being the case, we stop adding more context. The motivation behind this is that we would like to present the model with sequences that have starting and end points similar to real SLU boundaries. We generate input sequence 2 in the same fashion, starting from token$_{t+1}$ and adding at least 10 tokens to the right, stopping when we encounter a potential SLU boundary.

Figure 4 illustrates the BERT architecture for sentence pair classification that we apply for the SLU detection task.[9] The two input strings are separated by an artificial token and the model explicitly encodes the sequence id for each token (i.e. whether the token belongs to the first or to the second sequence) on a separate segmentation embedding layer. We train a sentence pair classifier on our training data and predict SLU boundary labels for each token in the test set.

Table 6 shows results for modelling SLU detection as sentence pair classification. For ease of comparison, we repeat our results for sequence tagging (Table 4) in the first row of Table 6. As can be seen, explicitly representing the left and right context relevant for SLU boundary prediction as separate representations results in an increase of more than 3% F1 for the boundary class (F1 B). F1 over both classes increases from 89.7% to 91.5%. Our model also outperforms the best Flair model trained with additional custom embeddings (Table 4) by 2.3% (F1 B).

We also notice that the sentence pair classification model seems to be more robust, compared the the BERT sequence tagging model from Section 5.1. We observe less variation in results between different runs (for sequence tagging, standard deviation for F1 (B) was 0.64 while the standard deviation for the sentence pair classification model is some-

[9] Figure taken from Devlin et al. (2019) and adapted to our task.

	Macro Acc	Macro F1	F1 B	F1 O
	BERT sequence tagger			
avg.	**95.1**	**89.7**	**82.3**	**97.2**
ID	*BERT sentence pair classifier*			
1	95.9	91.4	85.1	97.6
2	96.1	91.7	85.6	97.7
3	96.0	91.4	85.1	97.7
4	96.0	91.5	85.4	97.7
5	96.0	91.5	85.3	97.7
avg.	**96.0**	**91.5**	**85.3**	**97.7**

Table 6: Results for SLU detection with BERT in different task setups (sequence tagging, sentence pair classification).

	Macro Acc	Macro F1	F1 B	F1 O
ID	*BERT sentence pair classifier*			
avg.	**96.0**	**91.5**	**85.3**	**97.7**
ID	*Transfer learning on KiDKo*			
1	96.3	92.2	86.5	97.9
2	96.2	92.0	86.1	97.8
3	96.3	92.0	86.2	97.8
4	96.2	92.0	86.2	97.8
5	96.2	92.0	86.2	97.8
avg.	**96.3**	**92.0**	**86.2**	**97.8**

Table 7: Results for SLU detection as sentence pair classification, with transfer learning on the KiDKo dataset.

what lower with 0.38). In addition, we noticed that the model seems to be less sensitive to the choice of learning rate than the sequence tagging model trained on the same data.

5.3. SLU detection with transfer learning

In Section 4. we saw that adding data from another corpus of spoken language transcripts failed to improve results on SegCor, despite the similarity of content (informal multiparty dialogues) and annotations. Therefore, we now want to test whether we can benefit from the auxiliary training data when using it to initialise the model weights in a transfer learning setup.

So far, we have used the pre-trained BERT embeddings to leverage the information learned from large amounts of data to help for our specific task, SLU detection on spoken language transcripts. In our final experiment, we add another learning step where we first fine-tune a pretrained BERT model on the KiDKo training data and then fine-tune the same model again on the SegCor data. This should allow the model to transfer useful information from the larger KiDKo dataset while still being fine-tuned to the target data.

Table 7 shows that this additional learning step allows us to benefit from the auxiliary training data. For ease of comparison, we repeat our best result from Table 6 in the first row of Table 7. The additional transfer learning step further increases results on SegCor from 85.3% to 86.2% F1(B) on average, yielding another improvement of nearly 1%. We can also see that this procedure results in a more robust classifier, with hardly any variation between the results for the different runs. In contrast, our previous BERT models are highly sensitive to initialisation, with differences in results between the highest and the lowest score of 1% for sentence pair classification and 1.6% for sequence tagging (F1 for the boundary class).

6. Conclusion & Future Work

The goal of this paper was to improve results for SLU detection in spoken language transcripts. To that end, we experimented with different architectures, based on the Bi-directional Encoder Representations from Transformers (BERT). We showed, however, that transfer learning with pre-trained BERT embeddings does not always outperform other neural architectures and that choosing the right model and data representation is crucial. For the task of SLU boundary detection, we showed that explicitly encoding the left and right context is important, and that this can be done using BERT's segmentation embedding layer. Modelling SLU detection as sentence pair classification obtained 3% improvement (F1 B) over a BERT sequence tagger trained on the same data.

Expanding the training set with additional data from another spoken language corpus, however, did not yield the expected results. Instead, we observed even a slight decrease in F1(B) when training on the combined datasets. We showed that this problem can be overcome in a transfer learning setup where we succeeded to make use of the auxiliary data without suffering from out-of-domain effects. The final classifier not only outperforms our previous models on the SegCor data by another 0.9% F1(B) but also proved to be far more robust than models trained only on the smaller SegCor dataset.

Based on our previous results, we suggest that multi-task learning might be another way to improve results for SLU detection in spoken language transcripts. We leave this avenue to future work.

7. Acknowledgements

This work was conducted withing the project ”Segmentation of oral Corpora” (SegCor, DFG project number 281693063), funded by the Agence Nationale de la Recherche (ANR) and by the German Research Foundation (DFG).

8. Bibliographical References

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of* *the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *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.

Drach, E. (1937). *Grundgedanken der Deutschen Satzlehre*. Diesterweg, Frankfurt am Main.

Gotoh, Y. and Renals, S. (2000). Sentence boundary detection in broadcast speech transcripts. In *in Proc. of ISCA Workshop: Automatic Speech Recognition: Challenges for the new Millennium ASR-2000*, pages 228–235.

Hamaker, J., Zeng, Y., and Picone, J. (1998). Rules and guidelines for transcription and segmentation of the switchboard large vocabulary conversational speech recognition corpus. Technical report, Mississippi State University.

Höhle, T. N. (1986). Der Begriff "Mittelfeld". Anmerkungen über die Theorie der topologischen Felder. In *Akten des VII. Internationalen Germanisten-Kongresses, Bd. 3*, pages 329–340.

Kolář, J. (2008). *Automatic Segmentation of Speech into Sentence-like Units*. Ph.D. thesis, PhD Thesis, University of West Bohemia, Pilsen, Czech Republic.

Lison, P. and Tiedemann, J. (2016). Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In Nicoletta Calzolari (Conference Chair), et al., editors, *The Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Liu, Y., Stolcke, A., Shriberg, E., and Harper, M. (2005). Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 451–458. Association for Computational Linguistics.

Read, J., Dridan, R., Oepen, S., and Solberg, L. J. (2012). Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India. The COLING 2012 Organizing Committee.

Rehbein, I., Schalowski, S., and Wiese, H. (2014). The kiezdeutsch korpus (kidko) release 1.0. In *The Ninth International Conference on Language Resources and Evaluation*, LREC 2014, Reykjavik, Iceland, May 26-31, 2014, pages 3927–3934.

Rehbein, I. (2011). Data point selection for self-training. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 62–67, Dublin, Ireland. Association for Computational Linguistics.

Roark, B., Liu, Y., Harper, M., Stewart, R., Lease, M., Snover, M., Shafran, I., Dorr, B., Hale, J., Krasnyanskaya, A., and Yung, L. (2006). Reranking for sentence boundary detection in conversational speech. In *The 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, 12.

Rudrapal, D., Jamatia, A., Chakma, K., Das, A., and Gambäck, B. (2015). Sentence boundary detection for social media text. In *The 12th International Conference on Natural Language Processing*, pages 254–260, Trivandrum, India. NLP Association of India.

Ruppenhofer, J. and Rehbein, I. (2019). Detecting the boundaries of sentence-like units in spoken german. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 130–139, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Schmidt, T. and Westpfahl, S. (2018). A study on gaps and syntactic boundaries in spoken interaction. In Adrien Barbaresi, et al., editors, *The 14th Conference on Natural Language Processing*, KONVENS 2018, pages 40–49. Austrian academy of sciences, Vienna, Austria.

Schmidt, T., Schütte, W., and Winterscheid, J. (2015). cgat. konventionen für das computergestützte transkribieren in anlehnung an das gesprächsanalytische transkriptionssystem 2 (gat2). Working paper, IDS Mannheim, Mannheim.

Schmidt, T. (2014). The research and teaching corpus of spoken german – folk. In *The 9th conference on international language resources and evaluation*, LREC 2014, pages 383–387, Reykjavik. European Language Resources Association (ELRA).

Søgaard, A. (2011). Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 682–686, USA. Association for Computational Linguistics.

Stevenson, M. and Gaizauskas, R. (2000). Experiments on sentence boundary detection. In *The Sixth Applied Natural Language Processing Conference*, pages 84–89, Seattle, Washington, USA. Association for Computational Linguistics.

Westpfahl, S. and Gorisch, J. (2018). A syntax-based scheme for the annotation and segmentation of german spoken language interactions. In *The Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions*, LAW-MWE-CxG 2018, pages 109–120. Association for Computational Linguistics, Stroudsburg, PA, USA.

Westpfahl, S. and Schmidt, T. (2016). Folk-gold – a gold standard for part-of-speech-tagging of spoken german. In Nicoletta Calzolari, et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, LREC 2016, pages 1493–1499, Paris. European Language Resources Association (ELRA).

Westpfahl, S., Schmidt, T., Borlinghaus, A., and Strub, H. (2019). Guideline: syntaktische segmentierung in folker. Working paper, Leibniz-Institut für Deutsche Sprache (IDS), Mannheim.

Wiese, H., Freywald, U., Schalowski, S., and Mayr, K. (2012). Das kiezdeutsch-korpus. spontansprachliche daten jugendlicher aus urbanen wohngebieten. *Deutsche Sprache*, 40:97–123.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zribi, I., Kammoun, I., Ellouze, M., Belguith, L. H., and Blache, P. (2016). Sentence boundary detection for transcribed tunisian arabic. In *The 13th Conference on Natural Language Processing*, KONVENS'16.

9. Language Resource References

John J. Godfrey, Edward Holliman. (1993). *Switchboard-1*. Linguistic Data Consortium, 2.0, ISLRN 988-076-156-109-5.