Participation in the KONVENS 2021 Shared Task on Scene Segmentation Using Temporal, Spatial and Entity Feature Vectors

Florian Barth, Tillmann Dönicke
Göttingen Centre for Digital Humanities
University of Göttingen
florian.barth@uni-goettingen.de
tillmann.doenicke@uni-goettingen.de

Abstract

This paper describes the team's efforts in solving the KONVENS 2021 Shared Task on Scene Segmentation. It presents a statistical approach and puts a focus on the design of feature vectors that cover the key criteria for scene boundaries, namely the change of time, space, and/or entities between two scenes. Combining our feature set with a random forest classifier achieves micro-averaged F1's of 0.07 (in-domain) and 0.12 (off-domain), which puts our system in third place (out of five) in the shared task but does not improve over the performance of the neural model previously published by the organisers (Zehe et al., 2021a). Nevertheless, we think that handcrafted features can, in combination with distributional embeddings, improve the task of scene segmentation and this paper might inspire future work in this direction.

1 Introduction

The analysis of narratological phenomena is a major field of research within computational literary studies and focuses on aspects like time (Kearns, 2020), space (Pustejovsky et al., 2015), narrative levels (Reiter et al., 2019) as well as perspective or involvement of the narrator (Eisenberg and Finlayson, 2016). These tasks typically involve a critical reflection on the theoretical background in conjunction with an extended refinement of annotation guidelines (cf. Bögel et al. 2015), which leads to complex categories for which annotators mostly achieve moderate to substantial agreement.

The current task focuses on the detection of narrative scenes (Zehe et al., 2021b). The concept introduced by Genette (1983) describes a strong relationship or even equality of narrated time (time of *discours*) and story time (time of *histoire*) as an exclusive criterion for the definition of a scene. Zehe et al. (2021a) expand the definition to 4 criteria that constitute scenes: the equality or continuity of 1) time and 2) space, 3) the centrality of a specific action, and 4) a constant character constellation. In this paper, we present a statistical classifier that is based on a feature design including each of these criteria for scenes. This allows an analysis of feature importance and a better understanding of how the defining criteria are processed by the learning algorithm.

The observations on scene detection in fictional texts can furthermore contribute to similar tasks in other textual domains such as news stories, or it might serve as a basis for an in-depth understanding of the plot structure within a narration.

	texts	sents	S-S	S-NS	NS-S
trial	1	1,080	40	2	2
train	20	56,461	1,112	59	65
eval 1	4	–	475	18	21
eval 2	2	–	743	33	35

Table 1: Number of texts, sentences and individual boundary classes for trial, training and evaluation data. Evaluation texts were not provided, so the number of sentences is not known to us.

2 Data

The shared task data consists of 27 novels from which 20 serve as train set, 1 was given as trial data, and 6 as test set. The latter is split into two evaluation tracks: the first consists of 4 dime novels and the second contains 2 novels of contemporary high literature. All training and trial texts are dime novels. On average, each text of the training data consists of 35,801 tokens (standard deviation: 10,425) and 2,823 sentences (standard deviation: 550). The target classes of the task are the boundaries between scenes and/or nonscenes. Therefore, 3 different classes exist: Scene-to-Scene, Scene-to-Nonscene, and Nonscene-to-Scene (Nonscene-to-Nonscene is excluded by definition). The proportion of scenes within the training data is considerably higher, which is why the 2 classes involving nonscenes are rather underrepresented (cf. Table 1).

3 Preprocessing

We preprocessed the already sentencised texts with spaCy[1]. We used its default tokenizer and lemmatizer for German and added several custom preprocessing components. First, we added the Universal Dependency parser, morphological analyzer, clausizer and tense–mood–voice–modality tagger from Dönicke (2020). Second, we added a direct speech tagger that recognises text between opening and closing quotation marks. Third, we added a coreference resolver based on the algorithm from Krug et al. (2015), which we extended to create coreference clusters for all noun phrases in a text and not only character mentions. Fourth, we added a temponym tagger that recognises and normalises temporal expressions using regular expressions. For this, we used the German resource files from HeidelTime (Strötgen and Gertz, 2010)[2]. And fifth, we added a verb tagger that assigns Levin (1995)'s verb categories from GermaNet (Hamp and Feldweg, 1997) to the verb of the matrix clause, based on a disambiguation with respect to synset distances of verb–subject and verb–object(s).

4 Features

We extract features from different syntactic units in a sentence: from the sentence itself, from clauses, from noun phrases, and from tokens. When vectorising a document $D = (s_1, \dots, s_n)$, we get sentence vectors $(\vec{s}_1, \dots, \vec{s}_N)$, which we then concatenate to context-sensitive vectors $X_D = (\vec{x}_1, \dots, \vec{x}_n)$ using a window of 5 sentences: $\vec{x}_i := \vec{s}_{i-2} \circ \dots \circ \vec{s}_{i+2}$. The following subsection briefly describes the features which we extract from each sentence.

4.1 General Features

General features should catch structural markers for scene boundaries, e.g. (changes in) grammatical features such as tense and aspect, direct speech, presence of punctuation, and discourse connectives (usually the first word of a sentence).

From the matrix clause, we extract: **tense** (fut/past/pres), **aspect** (imperf/perf), **mood** (imp/ind/subj:past/subj:pres), **voice** (active/pass:dynamic/pass:static), and the lemmas of **modal verbs** (*können*/*müssen*/*wollen*/...); whether it is inside **direct speech** (no/yes); and for all verbs, the **part of speech** (AUX/VERB) and **Levin category** (Communication/Cognition/...).

From subordinate clauses, we extract: the root **dependency relation** (acl/ccomp/csubj/...); whether it is inside **direct speech**; and for all verbs, the **Levin category**.

From the sentence, we extract: the lemmas of **punctuation tokens** (,/;/*/...); the lemma of the **last token** if it is a punctuation token; whether the first, last or any other token is a **space token** (e.g. an empty line)[3]; the **part of speech** (PRON/SCONJ/...) and **dependency relation** (mark/nsubj/...) of the first non-punctuation, non-space token; and the **number of tokens** and the **number of clauses**.

Because we had the impression that nonscenes increasingly occur in the beginnings and ends of texts, we also added the **sentence's index** as feature, once counted from the start and once counted from the end. To prevent the system from simply memorising the tags for all indices, we set the feature to 10 for indices greater or equal to 10.[4]

4.2 Temporal Features

Temporal expressions are recognised and normalised by the temponym tagger. We split the **norm value** of a temporal expression at dashes and camel-cased letters into substrings that we use as substring features. For example, the expression *[am] nächsten Tag* '[the] next day' is normalised as *UNDEF-next-day* which is split into $\{UNDEF, next, day\}$.

For all temporal expressions, the temponym tagger further returns a **type** (date/duration/interval/set/time), and optionally a **modifier** (END/

[1] https://spacy.io/

[2] https://github.com/HeidelTime/heideltime/tree/master/resources/german

[3] We included this feature because headings, which are surrounded by empty lines, are usually nonscenes, and spaCy inserts special space tokens for such empty lines. However, the organisers of the shared task seem to have replaced all whitespace with single spaces beforehand, so there are no space tokens.

[4] 9/21 (43%) of the trial and training texts have a nonscene boundary among the first 10 sentences; 2/21 (10%) have a nonscene boundary among the last 10 sentences.

MID/START), a **quantifier** (EVERY), and a **frequency** (1M/1S/1W), which we also add as features.

We ignore temporal expressions within direct speech for the same reason as we ignore spatial and other mentions within direct speech (see next subsection).

4.3 Mention Features

Mentions are all noun phrases, including pronouns, in a document that are part of a coreference cluster. (These are all noun phrases with a few exceptions, e.g. expletive and interrogative pronouns.) Hereby, all mentions of a coreference cluster should denote the same entity. For the shared task, we ignore all mentions within direct speech because we only want to consider entities that are present in a scene, and direct speech frequently contains mentions of absent entities.

We differentiate between spatial entities (toponyms, nouns with inherent spatiality, e.g. buildings, inner rooms, landscapes, etc.) and other entities (characters, objects, concepts, etc.). For the distinction, we extracted a list of 18,345 spatial nouns from GermaNet based on their affiliation to a certain upper-level synset and define a mention to denote a spatial entity if its head noun is in the list. We consider times, spaces, and characters to be most relevant for the shared task but do not further sub-categorise other (i.e. non-spatial) entities into characters and non-character entities. We think that this is not necessary since we assume clusters of characters to stand out through a high rate of proper-noun mentions and include part-of-speech-based features, as we will describe below.

We extract an identical set of features for mentions of spatial entities and mentions of other entities. In the following, we describe the procedure for *either*.

First, we determine the sentence distance of each mention, i.e. how many sentences ago the corresponding entity was last mentioned. If an entity is mentioned for the first time, we set the distance to -1. Then, we take the mention with the lowest distance and the mention with the highest distance for feature selection and discard all other mentions.[5]

From the two mentions, we extract: the **sentence distance**; whether the mention's head is a **pronoun** (no/yes), **proper noun** (no/yes), or **common noun** (no/yes); the root dependency relation / **grammatical role** (iobj/nmod/nsubj/obj/obl); **case** (acc/dat/gen/nom), **person** (1per/2per/3per), **number** (plu/sing), and **gender** (fem/masc/neut); whether the mention contains a **determiner** (no/yes) or **numeral** (no/yes); and the lemmas of **determiners** (*der*/*dies*/...). Furthermore, we add a feature that indicates whether the two mentions are the **same** (if there is only one mention in the sentence).

We are aware that our coreference algorithm is not perfect and sometimes returns more than one cluster for the mentions of a single entity or, even worse, merges the clusters of two or more entities. We therefore add some features that should give a hint on the *trustworthyness* of a coreference cluster. From the clusters of the mentions, we extract: **number of mentions**; **percentage of pronoun** mentions, **percentage of proper-noun** mentions, and **percentage of common-noun** mentions; and its ***lemma-type ratio***, which we define as the fraction

$$r(C) = \frac{\max\#2_{\ell}|\{m \in C_{\text{nouns}} : L(H(m)) = \ell\}|}{\max\#1_{\ell}|\{m \in C_{\text{nouns}} : L(H(m)) = \ell\}|}$$

with $C_{\text{nouns}} = \{m \in C : \text{is\_noun}(H(m))\}$, $H(m)$ being the head of m, $L(w)$ being the lemma of w, and $\max\#nS$ being the n-th highest element in S. Thus, $r(C)$ divides the frequency of the second-most frequent lemma in C_{nouns} by the frequency of the most frequent lemma in C_{nouns}.

4.4 Statistics

To counteract overfitting, we exclude features that occur less than 5 times in the training data, reducing the number of features from 2,820 to 1,744. Categorical features are then binarised so that each feature–value combination becomes a Boolean feature. This results in 2,104 features altogether. Table 2 shows the top-45 features scored and ranked by ANOVA F-value for classifying the

[5] A sentence contains an unfixed number of mentions but a feature vector has a fixed number of dimensions, leaving two options: 1) One can extract features from every mention but makes it impossible to reallocate the feature values to individual mentions, or 2) one can create feature groups for individual mentions but only extracts features for a fixed number of mentions. We chose to go with the second option because we think that the combination of features is very important, e.g. knowing that *a (single)* mention has case = nominative and number = singular is presumably much more important than knowing that *some* mention has case = nominative and *some (possibly different)* mention has number = singular.

F	*i*	Unit	Feature	Value
747	-1	token	punct	...
586	-1	token	punct	*!*
554	0	max space m.	dep	obj
330	0	max space m.	dep	obl
308	0	temponym	substr	*PT1M*
305	-2	token	punct	‘
261	0	temponym	substr	*PXD*
259	0	max space m.	case	nom
257	0	min space m.	case	nom
245	-2	min other m.	person	1per
238	0	first token	pos	DET
233	0	min other m.	is_nn?	
205	0	temponym	substr	*REF*
198	0	first token	pos	NUM
195	0	subord. clause	dep	appos
194	0	temponym	substr	*WI*
193	0	last token	punct	*:*
193	0	max space m.	case	acc
192	0	max space m.	gender	masc
191	0	min space m.	case	acc
191	0	first token	pos	CCONJ
189	0	temponym	substr	*PT10M*
187	-1	sentence	tempon.?	
184	0	first token	pos	INTJ
181	0	temponym	substr	*P1D*
176	0	temponym	substr	*P5Y*
170	0	max space m.	num.?	
164	1	matrix clause	aspect	imperf
162	0	last token	punct	≫
160	1	first token	dep	xcomp
158	-2	first token	pos	SCONJ
158	0	last token	punct	...
151	0	max space m.	sent. dist	
150	1	first token	pos	AUX
137	0	last token	punct	*;*
136	0	token	punct	*;*
135	0	max space m.	gender	neut
134	0	max space m.	dep	nmod
132	0	min space m.	article	*seinen*
131	0	min space m.	dep	nmod
131	0	min space m.	dep	obj
130	0	min space m.	num.?	
126	0	min space m.	article	*einzig*
123	1	last token	punct	*
122	1	first token	pos	ADJ

Table 2: 45-best features ranked by F-value. Each feature represents: the offset i of the sentence from the current sentence, the syntactic unit from which the feature was extracted, the feature itself, and (only for originally categorical features:) the value of the feature.

sentences of the training set into four classes: None (no boundary), Nonscene-to-Scene, Scene-to-Nonscene, and Scene-to-Scene.

The attributes *max* and *min* in e.g. *max space mention* indicate whether it is the mention with the highest or lowest sentence distance, respectively. We can see that some minimal pairs of features receive (almost) equal F-values, e.g. *max space mention case = nom* and *min space mention case = nom*. This is due to the fact that most sentences only mention up to one spatial entity. Among the top-45 features, there are 16 features addressing mentions of spaces and 8 features addressing temporal expressions. Further 8 features cover the first token in a sentence, mainly its part of speech, 5 features cover the sentence-final punctuation in a sentence, and 4 features cover the punctuation anywhere in a sentence. The remaining features address other mentions (2×) and grammatical features of subordinate clause (1×) and matrix clause (1×). 34 features cover the current sentence; the remaining features cover the second last (3×), last (3×), and next (5×) sentence.

5 Classifier

A sentence can either be the start of a scene, the start of a nonscene, or none of both. Thus, a 3-class classification (Scene, Nonscene, None) would be sufficient for training a classifier and constructing spans for scenes and nonscenes. A span tagged with X followed by a span tagged with Y then produces the boundary class X-to-Y for the shared task's evaluation. However, to sensitise our model with respect to types of scene boundaries, we used 4 classes during training (None, Nonscene-to-Scene, Scene-to-Nonscene, Scene-to-Scene), where the class Scene is divided into Nonscene-to-Scene and Scene-to-Scene. In this classification, the first sentence of a document was treated as Scene-to-Scene or Scene-to-Nonscene boundary, depending on whether the document starts with a scene or a nonscene. A span tagged with A-to-X followed by a span tagged with B-to-Y then produces the boundary class X-to-Y in the evaluation, ignoring A and B.

We trained a random forest classifier with 100 decision trees, entropy as split criterion, a maximum tree depth of 11, and at least 3 samples per leaf. The class weights were balanced for all 4 classes. These parameters showed the best results in a cross-validation (see 6.1). We also

	mean	std	max	min
Scene-to-Scene	.075	.067	.231	.000
micro-avg	.070	.062	.202	.000
macro-avg	.026	.022	.077	.000

Table 3: Cross-validation results: F1 on the Scene-to-Scene class and micro-averaged and macro-averaged F1 for all classes.

	Precision	Recall
Scene-to-Scene	.070	.090
micro-avg	.068	.080
macro-avg	.024	.032

Table 4: Cross-validation results: mean precision and recall on the Scene-to-Scene class and micro-averaged and macro-averaged precision and recall for all classes.

tested other classification methods, including Naive Bayes, SVM and k-NN, but decision trees yielded the best results, presumably because they are able to learn dependencies between features.

6 Results

6.1 Cross-Validation

Due to the sparseness of nonscenes within the corpus, we also include the trial data for evaluating our hyperparameters. We perform a 21-fold cross-validation where each text represents one fold (leave-one-out evaluation). For the micro-averaged F1 including all classes, we achieve a mean over all folds of 0.070 with a standard deviation of 0.062 (see Table 3). The best result for an individual text/fold in the micro-averaged evaluation of all classes is 0.202. Our classifier does not detect any nonscene correctly, hence the macro-averaged F1 for the cross-validation is considerably lower. The F1 for the most-frequent class Scene-to-Scene is slightly higher than the micro-averaged F1, which reflects that our classifier is optimised on the detection of this boundary type and the micro-averaged F1 essentially depends on this class.

Overall, the number of predictions by the classifier is similar to the number of gold boundaries, which creates a balance of precision and recall (see Table 4).

6.2 Final Evaluation

In the final evaluation, the classifier achieves slightly better results than in the cross-validation.

	mean	std	max	min
eval 1	**.07**	.05	.14	.03
eval 2	**.12**	.07	.17	.07

Table 5: Final evaluation results: micro-averaged F1.

	mean	std	max	min
eval 1	**.06**	.12	.24	-.02
eval 2	**.08**	.04	.10	.05

Table 6: Final evaluation results: γ agreement.

Table 5 shows the micro-averaged F1 on both evaluation sets. Evaluation on the dime novels yields an average F1 of 0.07 whereas evaluation on high literature yields an average F1 of 0.12. This is surprising insofar that the training set consists only of dime novels which means that the off-domain performance is higher. Note, however, that the evaluation sets consist only of 4 and 2 texts, respectively, and the difference in performance lies in the range of variation that we observed in the cross-validation.

As in the cross-validation, the classifier does not make correct predictions for Scene-to-Nonscene or Nonscene-to-Scene boundaries. Thus, the performance solely relies on the majority class Scene-to-Scene.

Table 6 additionally shows the γ metric (Mathet et al., 2015), which computes the agreement of gold spans and predicted spans. The value range is $[-\infty; 1]$, where a value of 1 indicates exact agreement, a value of 0 indicates agreement by chance, and values < 0 indicate worse-than-chance agreement. We achieve an agreement slightly better than chance on both evaluation sets.

7 Discussion

With respect to the narratological foundation of the task, we present a feature design that reflects the defining criteria for scenes like the temponym extraction for the temporal dimension, spatial nouns to describe the current setting, Levin verb categories for specific actions and the mention features as representation of the character constellation. We supplement these features with a variety of linguistic features such as tense, mood, voice, modal verbs, and direct speech.

The use of a random forest algorithm intends a transparency of the classification but besides the limited success rate, a manual inspection of individual trees gave no further insights about the connection between the learned decisions and the theoretical foundation of the feature design (e.g. the assumption that changes of characters, places or time trigger a new scene). We optimised the classifier in a way that the amount of predictions approximately matches the number of each boundary class within the gold data, which yields a balanced relation of precision and recall. Despite of this ratio, the algorithm never detects nonscenes properly and only a few predictions for the boundary type Scene-to-Scene match the correct sentence position.

Apparently, the only decisive rule which the classifier has learnt about nonscenes is that they occur in the beginning of texts. The class Nonscene-to-Scene was predicted as first boundary for every text in the cross-validation and every but one text in the final evaluation. The class Scene-to-Nonscene followed by Nonscene-to-Scene was predicted only 3 times in the middle of a text in the cross-validation and never in the final evaluation. Neither Scene-to-Nonscene nor Nonscene-to-Scene boundaries were predicted as last boundary for any text. This behaviour is caused by the sentence's index feature—when we removed it, the text-initial nonscenes were not predicted anymore. We kept the feature because a lot of texts do indeed start with a nonscene[6], although the prediction for the first sentence is not relevant for the shared task (the classification of the first sentence only produces a correct boundary when position and type of the next scene are detected properly).

Considering the complexity of the task, the context window of 2 sentences for each feature seems rather small, nevertheless, we could not achieve a better performance with a broader window. As we have seen in Section 4.4, features of context sentences are underrepresented in the feature top-list, which is probably caused by their sparsity. We hypothesise that a larger context for individual features might improve the classification as well as the additional usage of contextual embeddings. In general, training a classifier with a large number of (sparsely attested) features on a small data set is prone to over-/underfitting. We expect that a larger training set would reduce the sparsity of features and possibly increase the performance of our approach.

So far, our mention features include places and characters (and other entities) but we do not verify if a character actually appears or if a place is part of a current action (if not, for example, places are not relevant as scene features as *New York* in the sentence: *In a month, she plans to go to New York with her brother*). This possibly causes the prediction of incorrect boundaries. The approach of capturing actions by Levin's verb categories might be improved by a direct syntactical relation to a considered entity, still, it is limited to a single sentence (or its clauses), and a more context-sensitive approach for capturing actions would be desirable. The lack of distinction between present and absent entities influences both the correct prediction of Scene-to-Scene boundaries and the rare occurrences of nonscenes.

8 Conclusion

This paper describes the participation in the KONVENS 2021 Shared Task on Scene Detection. The categories of scene and nonscene are based on a narratological concept which is streamlined to four concrete criteria: (change of) time, place, character constellation, and/or type of action. The training set consists of 20 dime novels, in which the classes (boundary types of scenes/nonscenes) are considerably unbalanced. We developed general, linguistic features as well as specific ones with respect to the defining criteria of scenes. Despite the detailed feature design, our random forest classifier is only able to detect some occurrences of one boundary type (Scene-to-Scene) and never finds nonscenes. Inspecting the best features and the decision trees revealed that features for time and space play an important role in the functionality of the algorithm but within the given context they cannot develop to their full potential. Given the current feature design, a stronger focus on a broader context as well as modelling actions in relation to space and character entities might improve the overall results. For that, a larger training set would be desirable.

The complete code and the presented model can be found here: `https://gitlab.gwdg.de/florian.barth/stss`.[7]

[6] 7/21 (33%) of the trial and training texts start with a nonscene.

[7] All results can be reproduced with the annotation data of the task. Since the data is under copyright, we cannot publish it within this repository.

References

Thomas Bögel, Michael Gertz, Evelyn Gius, Janina Jacke, Jan Christoph Meister, Marco Petris, and Jannik Strötgen. 2015. Collaborative text annotation meets machine learning: heurecléa, a digital heuristic of narrative. *DHCommons Journal*, 1.

Tillmann Dönicke. 2020. Clause-level tense, mood, voice and modality tagging for German. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 1–17, Düsseldorf, Germany. Association for Computational Linguistics.

Joshua Eisenberg and Mark Finlayson. 2016. Automatic identification of narrative diegesis and point of view. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 36–46.

Gérard Genette. 1983. *Narrative discourse: An essay in method*. Cornell University Press.

Birgit Hamp and Helmut Feldweg. 1997. Germanet-a lexical-semantic net for german. *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Edward Kearns. 2020. Annotating and quantifying narrative time disruptions in modernist and hypertext fiction. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 72–77.

Markus Krug, Frank Puppe, Fotis Jannidis, Luisa Macharowsky, Isabella Reger, and Lukas Weimar. 2015. Rule-based coreference resolution in German historic novels. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 98–104, Denver, Colorado, USA. Association for Computational Linguistics.

Beth Levin. 1995. English verb classes and alternations. *A preliminary Investigation*, 1.

Yann Mathet, Antoine Widlöcher, and Jean-Philippe Métivier. 2015. The Unified and Holistic Method Gamma (γ) for Inter-Annotator Agreement Measure and Alignment. *Computational Linguistics*, 41(3):437–479.

James Pustejovsky, Parisa Kordjamshidi, Marie-Francine Moens, Aaron Levine, Seth Dworman, and Zachary Yocum. 2015. SemEval-2015 task 8: SpaceEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 884–894, Denver, Colorado. Association for Computational Linguistics.

Nils Reiter, Marcus Willand, and Evelyn Gius. 2019. A shared task for the digital humanities chapter 1: Introduction to annotation, narrative levels and shared tasks. *Journal of Cultural Analytics*, 2(1).

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

Albin Zehe, Leonard Konle, Lea Katharina Dümpelmann, Evelyn Gius, Andreas Hotho, Fotis Jannidis, Lucas Kaufmann, Markus Krug, Frank Puppe, Nils Reiter, et al. 2021a. Detecting scenes in fiction: A new segmentation task. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3167–3177.

Albin Zehe, Leonard Konle, Svenja Guhr, Lea Katharina Dümpelmann, Evelyn Gius, Andreas Hotho, Fotis Jannidis, Lucas Kaufmann, Markus Krug, Frank Puppe, Nils Reiter, and Annekea Schreiber. 2021b. Shared task on scene segmentation@konvens2021. In *Shared Task on Scene Segmentation*.