Predicting Structural Elements in German Drama

Janis Pagel[1], Nidhi Sihag[2] and Nils Reiter[3]

[1]*Institute for Natural Language Processing, University of Stuttgart, Pfaffenwaldring 5b, 70569 Stuttgart, Germany*
[2]*University of Stuttgart, Keplerstrasse 7, 70174 Stuttgart, Germany*
[3]*Department of Digital Humanities, University of Cologne, Albertus-Magnus-Platz, 50931 Cologne, Germany*

Abstract

We address the challenge of enriching plain text dramas with predicted TEI/XML elements. We use a large corpus of dramas annotated with TEI information about act/scene changes, speaker changes, and stage directions, among others. On this data, we fine-tune a pre-trained BERT transformer model on several subtasks, like predicting stage directions vs. utterances. We show that the used architecture is able to predict the learned structural elements on unseen data for several settings and models.

Keywords

TEI, Text Segmentation, Dramatic Texts, Computational Literary Studies

1. Introduction

Dramatic texts or dramas describe a type of literary text that is usually meant to be performed on stage and structured in a dialogical form. Many dramas share the following common structural aspects: i) the text is divided into acts and scenes; scene boundaries are often associated with characters entering and leaving the stage [cf. 10, p. 230]; ii) the current speaker is indicated by name and in most scenes there are at least two characters talking to each other in direct speech [cf. 10, pp. 5–6, 14]; iii) there are stage directions not meant to be spoken by characters but rather indicating certain actions to be performed on stage, like character exits and appearances or concrete actions or emotions of a character [cf. 10, p. 15]. Act, scene and speaker tags as well as stage directions are considered part of the secondary text, i.e. they are typographically marked and not spoken, while character speech is part of the primary text [cf. 10, pp. 13–14]. Figure 1 shows the beginning of Shakespeare's *Hamlet* illustrating these properties.

We present ongoing work on experiments to learn these types of structural elements from plain dramatic texts with the final goal of enriching texts automatically with markup encoded in XML/TEI. Other than for example GROBID[1], we do not rely on visual information coming from applying OCR (optical character recognition) on PDF documents, but consider plain text as our only input. While there already exist structural annotations of many of the most canonical dramatic texts, a procedure to create such annotations automatically allows to include more texts, in particular non-canonical ones, into quantitative analysis of dramatic texts.

Second Conference on Computational Humanities Research, November 17–19, 2021
janis.pagel@ims.uni-stuttgart.de (J. Pagel); st170055@stud.uni-stuttgart.de (N. Sihag); nils.reiter@uni-koeln.de (N. Reiter)
0000-0003-4370-1483 (J. Pagel); 0000-0003-3193-6170 (N. Reiter)

[1]https://github.com/kermitt2/grobid

ACT 1
Scene 1
Enter Barnardo and Francisco, two sentinels.
BARNARDO Who's there?
FRANCISCO Nay, answer me. Stand and unfold yourself.
BARNARDO Long live the King!
FRANCISCO Barnardo.
...

Figure 1: Beginning of Shakespeare's *Hamlet*, adapted from https://www.dracor.org/shake/hamlet

More generally, automatically predicting text structure from plain texts offers new ways to use texts, as layout, text formatting or structure play an important role in many historic text types and uses of them in computational humanities research. It is important to note that we do not see this work as editorial work or a form of interpreting the text by inserting markup. Instead, our goal is to create machine readable markup of structures that are already overtly in the text and can later be used for computational analyses.

2. Related Work

While there are many attempts of automatically segmenting texts into smaller and cohesive units, many of these works focus on recognizing discourse units [3, 4] and do not predict structural elements suitable for markup. On the other hand, approaches that deal with structure prediction, e.g. for deriving tables of contents [1] normally operate on a global level and do not consider local structures (such as stage directions or speaker designations).

Most recently, McConnaughey, Dai, and Bamman [6] show results of different models for recognizing segments of 1,055 historical and OCRed books. They test a conditional random fields (CRF) model, a random forest model and a bi-directional LSTM (Long Short-Term Memory) architecture on the task of detecting labels such as title, table of contents or appendix by using a feature space consisting of features such as detected keywords, alphabetical order of page contents or density of characters on a page. They operate on a page-level, but allow single pages to contain more than one label. They find the LSTM model to perform best and the two sequence based models (LSTM and CRF) to perform better than the random forest model; however, they also note that the differences in the results is not significant and that there is no apparent advantage of using a sequence based model over models that look at each instance in isolation.

Pethe, Kim, and Skiena [9] work on 9,126 books from Project Gutenberg and aim at identifying chapter boundaries by first creating ground truth data out of the Gutenberg documents and then using this data as a training and evaluation basis. They create the silver-standard ground truth dataset by using a pre-trained BERT model that retrieves a list of potential chapter heading candidates and filter this list using over 1,000 different regular expression rules. After removing the chapter headings from the data, they test different models like a fine-tuned BERT model to predict the breakpoints. The BERT model using a context window of 254 words performs best on all evaluation tasks and metrics.

Zehe et al. [11] present a dataset of scene annotations for literary prose texts. Their definition of *scene* is mostly based on narratological assumptions and hence they distinguish scenes

and non-scenes. They also present preliminary experiments on automatically detecting these segmentations using a stock BERT model. They point out that using a non-fine-tuned BERT model is not sufficient for solving the task. They specifically notice that the model cannot detect the beginning of scenes after non-scene episodes. As they also point out, non-scenes can only seldom be found in dramatic texts. We share their approach in classifying on a sentence-level and differentiating between different types of segmentation.

To our knowledge, there is no research on the automatic segmentation of dramas.

3. Model

We make use of a Bidirectional Encoder Representations from Transformers (BERT) architecture as our base model. BERT is a large neural network architecture, with a number of parameters that can range from 100 million to over 300 million.

It is usually preferable to use a pre-trained BERT model that was trained on a huge dataset. For the following experiments, we use models that were trained on English and German data. While the English models are trained on a different language than that of the dramatic texts, it allows us to show how much the model relies on structural compared to semantic information.

3.1. Architecture

Our models are based on the pre-trained BERT model. It consists of 12 layers with a deep self-attention mechanism. We feed the BERT model with the tokenized sentences which contain the input ids and attention mask. The output of the BERT model is passed on to a *dropout layer* (rate: 10 %) so that we can prevent our model from overfitting. The output of the dropout layer is passed on to a *rectified linear activation function* to overcome the vanishing gradient problem and allow our model to learn faster and perform better.

After that, two dense layers are used. The first layer has 768 states equal to the number of hidden states of the pre-trained BERT model. The layer is followed by a second dense layer with a softmax activation function.

During fine-tuning, we freeze all the layers of the model, to prevent any updating of model weights.

4. Data

We are using the German Drama Corpus (GerDraCor), a collection of German-language plays from 1730 to the 1940s [2]. These files are encoded with TEI[2] tags that represent structural properties of the plays. The following tags are relevant to our task: `<div type="act">` indicates whether a new scene or act is starting; `<head>I. Akt</head>` indicates the number of the act or scene; `<stage>` contains all the stage directions like *tall, strong man with a coal-black beard, throws the dice with a great noise*; `<speaker>` contains the name of the character that is currently speaking; and finally `<p>` and `<l>`, which contain the text that is spoken.

We use single sentences as input for the BERT model. This is a compromise, as some tags will only cover single words or incomplete sentences. Using whole sentences as input has advantages though, as it gives more context to the model and it is straightforward to later assign XML tags for sentences or groups of sentences instead of arbitrary sub-sequences.

[2] https://tei-c.org/

Some tags will also have multiple lines of text stored inside them. We therefore use the *NLTK Sentence Tokenizer* to split groups of multiple sentences into single sentences, which can be given as input to our model. The extracted plain text is passed to the tokenizer. The text coming from inside the speaker tags is also considered as a sentence, even though it usually just constitutes a name. Figure 2 shows a part of our final pre-processed dataset which we use for training and testing. While the first column *SENTENCE* contains all the tokenized sentences, the second column *Decider* contains numeric class labels for the tokenized sentences.

	SENTENCE	Decider
0	Der innere Hof der Burg Farnrode	0
1	Von links vorn schrÃ⊡Â¤g Ã⊡Â¼ber die BÃ⊡Â¼hne...	0
2	Durch die geÃ⊡Â¶ffnete ThÃ⊡Â¼r blickt man in ...	0
3	Rechts hinten in der Ecke ein Wirthschaftsgeb...	0
4	Die rechte Seite schlieÃ⊡Â¸t eine hohe Mauer ...	0
...	...	...
1395143	Beut sie ihm	2
1395144	Er ist mein Sohn;	2
1395145	Und empfah' des Vaters Segen	2
1395146	ROSAMUNDE	1
1395147	sinkt neben Flodoardo vor dem Dogen auf die Knie	0

[1395148 rows x 2 columns]

Figure 2: Dataset after preprocessing. 0 stands for stage direction, 1 for speaker tag and 2 for an utterance.

Overall, the dataset contains 1 410 783 sentences with 10 021 598 tokens and 240 794 types. Since the sentences in the dataset are of varying length, we use padding to make all sequences have the same length. Since the vast majority of sentences has ten or less tokens, we set the maximal sequence length to 10.

5. Experiments

5.1. Baseline

We implement a simple baseline to compare the results of the transformer models against. For the baseline, we choose a conditional random fields (CRF) model, which is able to consider sequential information. To make the baseline comparable to the BERT models, we also choose sentences as the input and let the model predict if a sentence belongs to one of five classes: act (0), scene (1), stage direction (3), speaker tag (4) or utterance (5). The CRF receives features extracted from each sentence, namely:

- The lower-cased surface string of the sentence.
- If the sentence contains the German word ‘Akt’.
- If the sentence contains the German words ‘Szene’ or ‘Scene’.
- If the sentence begins with an uppercase letter.

- If the sentence only contains uppercase letters.
- If the sentence contains a digit.

For training, we make use of the limited-memory BFGS algorithm and elastic net regularization.

5.2. Experimental Setup

Training For the BERT models, we use pre-trained models provided by HuggingFace[3] to fine-tune on. We use the AdamW algorithm [5] which is an improved version of Adam to train, with a batch size of 256 and we clip the norm of the gradients at 1, as an extra safety measure against exploding gradients. The model is implemented in PyTorch [7] and scikit-learn [8]. We use negative log likelihood as loss function, and apply a learning rate equal to 2e−5. The training runs for 20 epochs.

Table 1
Distribution of classes on the dataset.

Class	Count	Rel. Count in Percent
Act	1,458	0.001
Scene	11,001	0.008
Stage	175,238	0.124
Speaker	316,451	0.224
Speech	906,635	0.642

Class Weights Table 1 shows how the classes are distributed among the sentences. Some classes have much more training examples than others, introducing bias in our models. To deal with this problem, we apply class weights to the loss function. These are computed as the inverse frequency of the classes in the training set.

5.3. Evaluation

We consider accuracy, precision, recall and F1-score as metrics.

5.4. Results

In this section, we investigate the performance of our proposed model on various tasks. We split the dataset randomly into three sets: train, validation, and test, where the train set is 70% and validation and test set are both 15% of the overall data. We fine-tune the model using the train and validation set, and evaluate on the test set.

Detecting Act Boundaries We extract the data from `<stage>`, `<head>` and `<speech>`. The sentence splitter recognizes 1 208 899 sentences. The goal of the prediction is to mark all sentences: If the sentence is the first sentence of an act it is classified as 1, otherwise as 0. Hence it is a binary classification task. For this task we use **'bert-base-uncased'** (the

[3] https://huggingface.co/

identifier at HuggingFace) as a base model. In Table 2 we can see the results as classification report and confusion matrix.

We can see that the model is able to predict a non-boundary for nearly 100 % of the cases. Yet, the model is not overfitting on class 0, as the prediction for an act boundary still gets high scores with an F1-score of 0.89. From Table 2b we can see that the model makes more mistakes in wrongly classifiying non-act-boundaries as act changes than the other way around (i.e., the number of false positives is higher than the number of false negatives).

Table 2
Evaluation results for detecting act boundaries.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	181 046
1	0.93	0.85	0.89	289
accuracy			1.00	181 335
macro avg	0.96	0.93	0.94	181 335
weighted avg	1.00	1.00	1.00	181 335

(a) Classification Report.

		GS	
		0	1
SO	0	181027	19
	1	42	247

(b) Confusion Matrix. GS is the gold standard, SO the system output.

Detecting Stage Directions For this task we use the `<stage>` and `<speech>` tags, which contain 1 203 911 sentences. Each sentence is classified as to whether it is (part of a) stage direction or not, using the **'bert-base-uncased'** model. As stage directions are much more frequent and much more similar to character speech, we consider this task to be more difficult than the one discussed above: Instead of relying on lexical cues, it needs to take discourse structure and semantic information into account.

Table 3 shows the results. In this, 0 represents character speech and 1 represents stage direction. The model has indeed more difficulties with correctly predicting stage directions. With a precision of 0.7 and recall of 0.95, the detection nevertheless performs reasonably well. The confusion matrix shows that false positives are much more common than false negatives, which can probably be explained by the imbalance in the training dataset.

Table 3
Evaluation results for detecting Stage direction and character speech.

	precision	recall	f1-score	support
0	0.99	0.92	0.95	151409
1	0.70	0.95	0.81	29178
accuracy			0.93	180587
macro avg	0.84	0.94	0.88	180587
weighted avg	0.94	0.93	0.93	180587

(a) Classification Report.

		GS	
		0	1
SO	0	139324	12085
	1	1365	27813

(b) Confusion Matrix. GS is the gold standard, SO the system output.

Table 4
Evaluation results when using the model Bert_Uncased

	precision	recall	f1-score	support
0	0.97	0.97	0.97	252
1	0.98	0.99	0.98	1820
2	0.67	0.92	0.77	29591
3	0.97	0.99	0.98	52912
4	0.97	0.91	0.95	153789
accuracy			0.93	238364
macro avg	0.92	0.96	0.93	238364
weighted avg	0.95	0.93	0.94	238364

(a) Classification Report.

		GS				
		0	1	2	3	4
SO	0	244	1	2	3	2
	1	7	1795	11	1	6
	2	0	3	27281	1195	1112
	3	0	0	339	52552	21
	4	0	29	13324	366	140070

(b) Confusion Matrix. GS is the gold standard, SO the system output.

Table 5
Evaluation results when using the model BERT_German_Uncased

	precision	recall	f1-score	support
0	1.00	0.95	0.97	252
1	0.97	0.99	0.98	1820
2	0.77	0.93	0.84	29591
3	0.97	0.99	0.98	52912
4	0.99	0.95	0.97	153789
accuracy			0.95	238364
macro avg	0.94	0.96	0.95	238364
weighted avg	0.96	0.95	0.96	238364

(a) Classification Report.

		GS				
		0	1	2	3	4
SO	0	240	7	3	0	2
	1	1	1808	6	2	3
	2	0	14	27526	1149	902
	3	0	3	411	52425	7
	4	0	32	7885	237	145635

(b) Confusion Matrix. GS is the gold standard, SO the system output.

All tasks combined For this task, we extract the data from the all above mentioned tags, which in total contain 1 589 090 sentences. The task now is a 5-way classification, as we classify sentences as being (part of) a stage direction (2), name of a speaker (3), character speech (4) or act (0) or scene boundary (1).

For this task we use different types of BERT models and compare them. Table 4 shows results for **'bert-base-uncased'**. All results are still comparable to the results of classifying the tags individually. Some of the results are lower, but not by much. This is promising, as it shows that we can potentially predict the complete structure of a plain text drama at once without loosing much in predictive power over classifying the single types of structure individually.

As mentioned earlier, all models so far have been pre-trained on English data. The above evaluation shows that even on German data, they can make good predictions, which can be explained by the fact that most of the distinguishing features needed so far for prediction are structural rather than content-based. However, for the task of predicting all tags together, we now use a model trained on German data and see if the results can be further improved. Table 5 shows the results for applying the **'bert-base-german-uncased'** model. We can see that especially for predicting stage directions, the performance improves significantly by 7 percentage points F1 score. The other results are either identical or slightly higher in the case

Table 6
Evaluation results when using the model Bert_German_Cased

	precision	recall	f1-score	support
0	0.96	0.97	0.97	252
1	0.95	0.99	0.97	1820
2	0.72	0.96	0.82	29591
3	0.96	1.00	0.98	52912
4	1.00	0.92	0.96	152789
accuracy			0.94	238364
macro avg	0.92	0.97	0.94	238364
weighted avg	0.95	0.94	0.94	238364

(a) Classification Report.

		GS				
		0	1	2	3	4
SO	0	245	1	5	0	1
	1	5	1800	5	7	3
	2	5	10	28406	736	434
	3	0	1	117	52780	14
	4	0	73	10914	1575	141227

(b) Confusion Matrix. GS is the gold standard, SO the system output.

Table 7
Evaluation results of the CRF baseline.

	precision	recall	f1-score	support
0	1.0	0.88	0.93	252
1	0.99	0.91	0.95	1820
2	0.86	0.30	0.44	29591
3	0.99	0.92	0.95	52912
4	0.86	0.99	0.92	152789
accuracy			0.89	238364
macro avg	0.94	0.80	0.84	238364
weighted avg	0.90	0.89	0.88	238364

(a) Classification Report.

		GS				
		0	1	2	3	4
SO	0	172	0	0	3	20
	1	0	1375	1	15	118
	2	0	9	7108	30	16479
	3	0	0	23	46162	3789
	4	0	3	1047	41	135223

(b) Confusion Matrix. GS is the gold standard, SO the system output.

of speech with a plus of 2 percentage points. This is absolutely expected, as these two types are more content based. Still, the English model is able to pick up on enough structural cues to also predict well on German data.

Lastly, we check if it makes a difference to use a model that was trained on cased data, as all other models before were trained on uncased data. Here, the **'bert-base-german-cased'** model has been used. The results for this can be found in Table 6, and are sightly lower than in the uncased setting. This suggests that preserving case lets the model generalize less well.

Baseline We compare these final results to the baseline system. The results are shown in Table 7. The baseline performs rather well for the tasks of predicting act and scene boundaries and recognizing speaker tags. However, the BERT-based models achieve slightly higher values for all these classes. For the task of character speech identification, it performs worse than the BERT-based models in term of precision, but achieves a higher recall than all other models. For the crucial task of recognizing stage directions, it returns a rather low recall value, but the highest precision value.

5.5. Summary of the results

In all experiments, we observe that the models achieve precision and recall scores around 95 % to 99 % for most of the categories. For stage directions, the evaluation yields lower scores: The model misclassifies some of the sentences in character speech as stage directions. By experimenting with different BERT models we are able to achieve a precision of 77 % for stage directions which means that BERT German Uncased is the most suitable model for these predictions. While the CRF-based model sets a high baseline for the tasks of act, scene and speaker recognition, the BERT-based models outperform the baseline in all measures. Only for the content-based tasks of speech and stage direction recognition, the baseline achieves higher results in recall and precision, respectively. In future work, the transformer models might benefit from combining them with the CRF model.

6. Conclusion and Future work

In this paper, we have shown that the BERT model is a reasonable model for predicting and extracting structural segments from dramatic texts. Based on this finding, we have proposed a novel fine-tuned model based on BERT. From the above results we can conclude that **'BERT_German_Uncased'** is the most effective base model. We can also conclude from the above results that all the models perform quite well, whether we predict the segments with binary classification or in the full model with five classes. We were further able to show that models trained on English data are able to predict the more structural elements of German dramatic texts with high accuracy. However, for the structural elements that rely more on text content, a model trained on German data performs better.

Both recall and precision for all classes except class 2 (stage directions), are quite high which means that the model predicts these classes accurately. The recall for class 2 is 0.93 which means that the model was able to find 93 % of the stage direction sentences. However, precision is a bit lower for class 2, which means that the model misclassifies some of the class 4 sentences (character speech) as stage.

In the future, we plan to extend on the presented work to create a fully automatic mapping tool to convert plain text scans of dramatic texts into properly structured TEI/XML documents. Even if this automatic conversion is likely to contain some errors, correcting it manually is much less labor-intensive than coding the entire play by hand. We plan to add texts currently only available in plain text to the DraCor corpora once the above mentioned tool is developed and functioning. One challenge we will most likely face will be that OCRed texts usually contain mistakes which might throw of the transformer model. Hence we will also experiment with text normalization techniques. This opens a path towards large scale data analysis of plays that currently are not available as part of the DraCor repository. In addition, our analysis has shown that the trained model works reasonably well even if used across language boundaries. This suggests that it is also possible to apply a very similar model on plays from other languages, as training data for many languages is already available.

Acknowledgements

The first and third author have conducted the described research within the QuaDramA project, funded by the Volkswagen foundation and within the Q:TRACK project, funded by

the German Research Foundation (DFG) in the context of SPP 2207 *Computational Literary Studies*. We thank both for making this possible.

References

[1] A. Doucet, G. Kazai, S. Colutto, and G. Mühlberger. "Overview of the ICDAR 2013 Competition on Book Structure Extraction". In: *Proceedings of the Twelfth International Conference on Document Analysis and Recognition (ICDAR)*. Washington, D.C., US, 2013, pp. 1438–1443.

[2] F. Fischer, I. Börner, M. Göbel, A. Hechtl, C. Kittel, C. Milling, and P. Trilcke. "Programmable Corpora: Introducing DraCor, an Infrastructure for the Research on European Drama". In: *Proceedings of DH2019: "Complexities"*. Utrecht, The Netherlands, 2019. DOI: 10.5281/zenodo.4284002.

[3] M. A. Hearst. "TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages". In: *Computational Linguistics* 23.1 (1997), pp. 33–64. URL: https://www.aclweb.org/anthology/J97-1003.

[4] A. K. John, L. Di Caro, and G. Boella. "Text Segmentation with Topic Modeling and Entity Coherence". In: *Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS)*. Ed. by A. Abraham, A. Haqiq, A. M. Alimi, G. Mezzour, N. Rokbani, and A. K. Muda. Vol. 552. Advances in Intelligent Systems and Computing (AISC). Springer, 2017, pp. 175–185. DOI: 10.1007/978-3-319-52941-7_18. URL: https://link.springer.com/chapter/10.1007%5C%2F978-3-319-52941-7%5C%5F18.

[5] I. Loshchilov and F. Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[6] L. McConnaughey, J. Dai, and D. Bamman. "The Labeled Segmentation of Printed Books". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark, 2017, pp. 737–747. DOI: 10.18653/v1/D17-1077. URL: https://aclanthology.org/D17-1077.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[9] C. Pethe, A. Kim, and S. Skiena. “Chapter Captor: Text Segmentation in Novels”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 8373–8383. DOI: 10.18653/v1/2020.emnlp-main.672. URL: https://aclanthology.org/2020.emnlp-main.672.

[10] M. Pfister. *The Theory and Analysis of Drama*. Trans. by J. Halliday. European Studies in English Literature. Cambridge: Cambridge University Press, 1988. DOI: 10.1017/cbo9780511553998.

[11] A. Zehe, L. Konle, L. Dümpelmann, E. Gius, A. Hotho, F. Jannidis, L. Kaufmann, M. Krug, F. Puppe, N. Reiter, A. Schreiber, and N. Wiedmer. “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*. Association for Computational Linguistics, 2021, pp. 3167–3177. URL: https://www.aclweb.org/anthology/2021.eacl-main.276.