Twin BERT Contextualized Sentence Embedding Space Learning and Gradient-Boosted Decision Tree Ensembles for Scene Segmentation in German Literature

Sebastian Gombert
Information Center for Education
DIPF: Leibniz Institute for Research and Information in Education
Frankfurt am Main, Germany
gombert@dipf.de

Abstract

This paper documents a submission to the shared task on scene segmentation hosted at KONVENS 2021 (Zehe et al., 2021b). The aim of this shared task was to find methods for segmenting narrative texts into different scenes – segments of text where location, time and the constellation of characters stay more or less coherent. This task is formulated as a sentence classification task where sentences bordering the scenes have to be distinguished from in-scene sentences. The approach presented in this paper is based on two steps. In the first one, a twin BERT training setup is used to learn a sentence embedding space in which sentences functioning as scene borders are well-separated from ones that are in-scene. In the second one, the sentence embeddings generated by this model are used as feature vectors to feed a gradient-boosted decision tree ensemble which conducts final predictions. In the shared task leaderboard, the system ranked second in track 1 and first in track 2.

1 Introduction

Scene segmentation in narrative texts is a novel task in natural language processing introduced by Zehe et al. (2021a). The aim of this task is to segment pieces of literature into scenes – sections of text where the relation of story time and discourse time, the location and character constellations stay more or less the same. From a formal point of view, this problem can be interpreted as a sentence in context classification task where sentences separating scenes have to be distinguished from in-scene ones. This is needed as the typical length of longer narrative texts such as novels prevents techniques such as co-reference resolution useful for proceeding steps of analysis from functioning well (Zehe et al., 2021a). With a text being segmented into coherent scenes, each scene can be processed separately improving the performance for such follow up processing.

This paper presents an a participating system at the *KONVENS 2021 shared task on scene segmentation* (Zehe et al., 2021b) and relies on two steps. For the first one, a *BERT*-based (Devlin et al., 2019) neural network trained in a twin network setup is used to predict embeddings for respective input sentences (Reimers and Gurevych, 2019). This network was trained to provide an embedding space in which sentences bordering scenes are well-separated from in-scene ones. For the second step, gradient-boosted decision tree ensembles (Mason et al., 1999) are then fed these sentence embeddings as feature vectors to carry out final predictions.

For shared task evaluations, this system was trained on a data set consisting of various German dime novels where scene borders had been previously annotated. Participating systems were evaluated in two tracks using F1 scores. In the first track, the models were evaluated using a test set consisting of additional dime novels. In this track, the system presented in this paper achieved the second place with an F1 of 0.16. In the second track, domain-adaptability was probed by evaluating the systems on a set of German contemporary highbrow literature. Here, the system presented performed better and was ranked first with an F1 of 0.26.

2 Background

2.1 Task Description

In Zehe et al. (2021a), the authors interpreted the task of *scene segmentation* as a sentence classification task. They defined four different classes of sentences: *no border*, *scene-to-scene*, *scene-to-nonscene* and *nonscene-to-scene*. The three latter of these are used to mark the different kinds of textual borders among the sentences. They trained a *BERT*-based (Devlin et al., 2019) classifier utilising a sliding windows over multiple sentences for context encoding to carry out sentence classification.

This approach was evaluated against the unsupervised *TextTiling* (Hearst, 1997) and *TopicTiling* (Riedl and Biemann, 2012) methods on a corpus consisting of 15 German dime novels using cross validation. While the supervised *BERT* model achieved superior results (γ 0.15) compared to the unsupervised methods (γ 0.01; γ 0.02), the overall results turned out subpar which led the authors conclude that *scene segmentation* can be regarded as an inherently hard task.

For the KONVENS 2021 shared task, the organizers provided an expanded version of the data set presented by Zehe et al. (2021a). This data set is composed of various German dime novels. The authors chose this genre as they deemed it easier for potential models to deal with.

2.2 Related Work

While segmenting text into smaller units such as tokens, sentences or spans is one of the oldest and most researched topics in natural language processing, the task of semantically segmenting narrative texts into scenes is a new one. In this form, *scene segmentation* was first introduced by Zehe et al. (2021a). From a problem-centric point of view, Zehe et al. (2021a) relate *scene segmentation* to the task of *topic segmentation*, the task of segmenting a text by topic changes, as changes of time, place and character constellation can be interpreted as a special cases of topic changes.

Most of the more recent work in this area (Riedl and Biemann, 2012; Misra et al., 2011) is built upon *latent Dirichlet allocation* (Blei et al., 2003). This method discovers fields of words consistently co-occuring in the same contexts. By monitoring changes in their distribution throughout a text, one can define topic-wise section borders. Another related topic according to Zehe et al. (2021a) is discourse coherence. Recent approaches in this area rely on neural networks to detect textual coherence in various setups and use cases (Li and Jurafsky, 2017; Pichotta and Mooney, 2016). Changes in these coherence scores can be used for detecting borders within texts, as well.

3 System Description

My code can be found under [1].

3.1 Adjustments to the Tag Set

While Zehe et al. (2021a) used a quaternary tag set which distinguished scene to scene- and nonscene to scene borders which is also used for official shared task evaluations, my system internally relies on a tertiary tag set consisting of the tags *O*, *SCENE* and *NONSCENE*. The latter two refer to the first sentence of an according section. The reason for this adjustment is that the number of border sentences is low compared to the number of non-border sentences. My tertiary tag set is the smallest classification setup which can be used to distinguish scenes and non-scenes. Using this tertiary tagset results in all scene to scene- and nonscene to scene sentences being grouped under the *SCENE* task, and all scene-to-nonscene ones under the *NONSCENE*.

3.2 Twin BERT Embedding Space Learning

My system is built around the idea of neural embedding space learning. Reimers and Gurevych (2019) introduced the idea of using twin and triplet network-based training setups for fine-tuning transformer language models to map sentences into meaningful semantic vector spaces under the name *Sentence Transformers*. In their training setup, two or three different sentences are fed into the same transformer language model. These pairs and triplets of sentences are assigned scores such as cosine similarity or concrete training labels. A prediction head which is fed the output of the transformer language model for all two or three sentences is trained to predict the assigned scores or labels. After this training process, the transformer language model can embed sentences into a vector space where they are well-separated according to the respective training objective.

The idea behind the system presented in this paper is to combine this approach of twin network embedding space learning with the sliding window-based approach from Zehe et al. (2021a). More precisely, my approach is to utilise a twin network-based training setup to learn an embedding space encoding information about a sentence as well as the sentences surrounding it. The goal here is that,

[1] `https://github.com/SGombert/ssts-2021-sego`

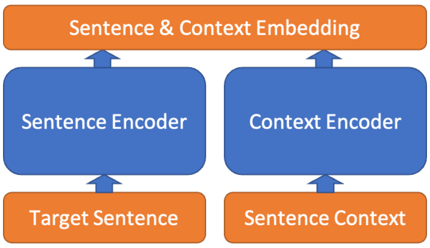

Figure 1: The architecture of the neural network model in prediction mode when generating contextualized sentence embeddings.

within this vector space, the embeddings of sentences bordering scenes are well-separated from them of in-scene ones.

Instead of a single *BERT* model as (Reimers and Gurevych, 2019), it uses two of them with one functioning as *sentence encoder* and one as *context encoder*. In both cases, the regular pooling layer output of these networks is used to encode given input sentences. While the *sentence encoder* is only used to predict a sentence embedding for a given target sentence, the context encoder also predicts sentence embeddings for a context window of n sentences to the left and to the right around this target sentence. The output of both encoders is concatenated to acquire the final embeddings for embedding a sentence and its context into vector space.

$$m(s_t) = e_{sent}(s_t) \oplus e_{cont}(s_t) \tag{1}$$

$$e_{sent}(s_t) = B_1(s_t) \tag{2}$$

$$e_{cont}(s_t) = c_{left}(s_t) \oplus B_2(s_t) \oplus c_{right}(s_t) \tag{3}$$

$$c_{left}(s_t) = B_2(s_{t-n}) \oplus \cdots \oplus B_2(s_{t-1}) \tag{4}$$

$$c_{right}(s_t) = B_2(s_{t+1}) \oplus \cdots \oplus B_2(s_{t+n}) \tag{5}$$

In these equations, s_t is a given sentence at time step (position in text) t. $m(s)$ refers to the function used for predicting embeddings. $e_{sent(s)}$ and $e_{cont}(s)$ are the two different encoder networks. B_1 and B_2 refer to the two underlying *BERT* networks, and $c_{left}(s_t)$ and $c_{right}(s_t)$ are the functions used for acquiring the context of a given sentence s_t. n determines the size of this context.

For training such a sentence embedding model, I randomly sampled 15000 pairs of sentences which were both either scene- or non-scene borders and 15000 pairs where both sentences were from different categories, the majority of them being pairs of scene border and in-scene sentences, from the training set. While the prior set of pairs is assigned a score of 1, the pairs from the latter set are assigned a score of -1.

$$m_{concat}(p) = m(s_1(p)) \oplus m(s_2(p)) \tag{6}$$

$$f(p) = L(m_{concat}(p)) \tag{7}$$

In these equations, p refers to a triple of two sentences from the training set and an according score (-1 or 1, depending on class equality), $s_1(p)$ and $s_2(p)$ are functions retrieving the first respectively second sentence from a given training input triple. $f(p)$ refers to the final output score calculated by the network during training and L to a linear feed-forward layer. During training both sentences of a triple and their according local context sentences are propagated through both the *sentence* respectively the *context encoders*. Their pooling layer outputs for both sentences are concatenated and propagated into a linear layer whose single output neuron is trained to predict the according score using *hinge embedding loss*:

$$f(x,y) = \begin{cases} x & \text{if y = 1} \\ max(0, \delta - x) & \text{if y = -1} \end{cases} \tag{8}$$

Within this function, x is a predicted score, y a gold standard one and δ the so-called margin, a hyper parameter which can be used to control the distances between the vectors a given model learns. This function is used to learn a maximum margin-like embedding space which separates scene borders from in-scene sentences.

The *GermanBERT* variant provided by Huggingface Transformers (Wolf et al., 2020) under the id *bert-base-german-dbmdz-uncased*[2] is used as a base for both *sentence encoder* and *context encoder*. The reason for choosing this model was that the data it was pre-trained on includes narrative texts which makes it an appropriate basis for a model dealing with literary data. The model was trained using *AdamW* (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) with the *learning rate*

[2] https://huggingface.co/bert-base-german-dbmdz-uncased

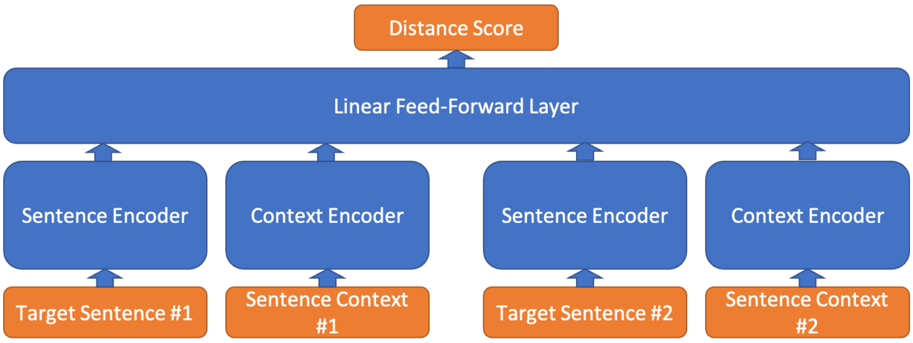

Figure 2: A visualisation of the twin network-based training setup.

set to 0.000001 and *weight decay* to 0.0001. The embedding model was trained for one epoch using a constant *warm up schedule* with a constantly increasing learning rate for the first 1000 iterations. No batch processing was used during training.

As visible in figure 3, the model indeed learned to embed sentences into a vector space in which they were well-separated into two distinct clusters. However, it does not seem that the model generalized the idea of what exactly is a scene border well from the training data. While for '*Der kleine Chinesengott*', the German dime novel provided as trial corpus, the majority of scene borders is located in the smaller of the two clusters, there are also borders located in the larger cluster, and, moreover, many in-scene sentences are also sorted into the smaller cluster. This phenomenon was visible after multiple training runs with different sampled pairs of sentences which implies that drawing clear distinctions between scene borders and in-scene sentences is hard for solely *BERT*-based models.

3.3 Gradient Boosted Decision Tree Ensembles

As the embedding model did seemingly not learn a precise enough distinction between scene borders and in-scene sentences, using *maximum margin* classification with the resulting embeddings as feature vectors was no option. Instead, I chose gradient boosted decision tree ensembles (Mason et al., 1999) as classification algorithm because of its ability to select distinctive features and ignore less distinctive ones.

During training, this algorithm creates an ensemble of weak regression trees trained to predict the logits within a specialized logistic regression setup. Combining enough of such trees results in a strong learner. This is conducted by means of gradient descent and decision tree learning. Each subsequent tree is trained to correct erroneous predictions of the previous ones. As each of them is limited to use only a small subset of the input features provided in given input feature vectors, the trained ensemble can automatically isolate features which globally distinguish scene borders from in-scene sentences the best within the training set.

For implementing this part of the system, I used *Catboost* (Prokhorenkova et al., 2018) as framework. The model is based upon its multi class classification mode. The tree growth policy is set to *lossguide* and *class weights* are used. The following formula is used for calculating them:

$$w_c = 1 - \frac{num(c)}{\sum_{c'}^{C} num(c')} \tag{9}$$

w_c is a respective class weight, c a class, C the set of all classes, c and c' classes and $num(c)$ a function which returns the number of training examples for a given class. Additionally, I used early stopping to prevent overfitting. For this, I set the number of training iterations to 5000, let the framework choose a learning rate automatically, and then used the checkpoint of the model which performed best on the trial dime novel.

4 Evaluation

4.1 Results

Shared task evaluations were carried out on two different corpora resulting in two different evaluation tracks. The first of these corpora consisted of 5 more dime novels similar to the ones systems were trained on to address in-domain transfer capabilities of the participating systems. The corpus used for the second track consisted of two pieces of highbrow German literature. The aim of this track was to evaluate out-of-domain transfer capabilities

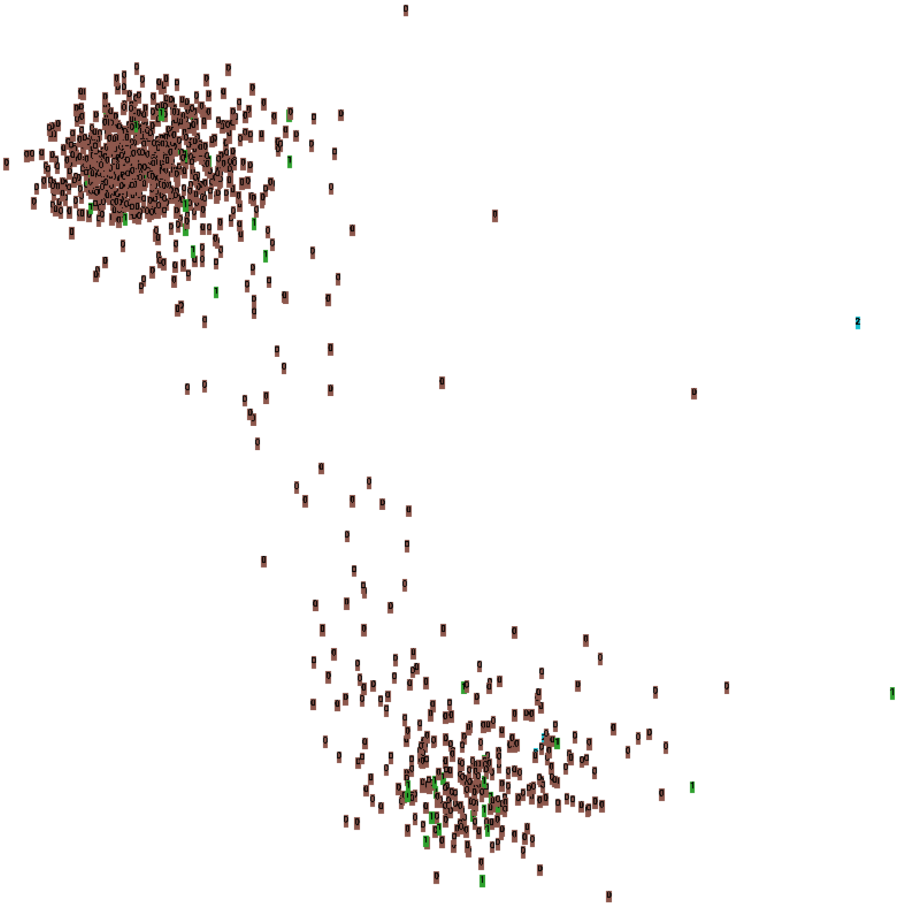

Figure 3: The embeddings predicted for the sentences from the dime novel '*Der kleine Chinesengott*' used as trial data in the shared task visualized in 2D using *principal component analysis* (Pearson, 1901). *0/brown* corresponds to in-scene sentences, *1/green* to scene borders and *2/blue* to non-scene borders.

of the participating systems. My system ranked second out of four in the first track reaching a micro F1 of 0.16 and first out of five in the second track reaching a micro F1 score of 0.26. These results confirm the difficulty of this task observed by Zehe et al. (2021a).

Track	F1	γ	Rank
Dime Novels	0.16	0.085	*2/4*
Highbrow Literature	0.26	0.175	*1/5*

Table 1: The shared task evaluation results of my system.

4.2 Qualitative Error Analysis

To further analyze the results of my system, I turned to qualitative error analysis. For this purpose, I collected the false negative and false positive scene border sentences detected by my system for the trial corpus and analyzed a selection of them with regard to common structural patterns. 128 of the sentences marked as scene borders within the trial corpus were false positives. What became quickly visible was that some false positives contained changes of time, character constellations and/or location. As these function as important signals for a scene change, the model seems to have overgeneralized such cases. The following utterances are examples for a signified change in time from false positives:

Langsam verstrich die Zeit.
Natürlich kamen wir zu spät.
unendlich langsam verstrich die Zeit [...].
Ich wartete also noch eine Weile, dann aber [...]
Gerade in dem Moment vernahm ich [...]

Examples for a change in character constellation are the following:

Bills Alarmruf hatte den Spitzbuben verscheucht.
Der Verfolger war [...] untergetaucht.
Da hörte ich Tom plötzlich aufstehen [...].
Tom erhob sich jetzt und entschuldigte sich [...].
Dem herbeieilenden Portier berichtete ich [...].
Ich war wieder allein [...].
Bill meldete in diesem Moment den Besuch Dr. Türks.
Ich fand ihn ohnmächtig auf dem Fußboden liegen.

The following utterances are examples for a location change:

Wir verließen unser Häuschen [...].
"Schnell, zu Wertheim," raunte Tom mir zu.
Wir trafen uns erst wieder draußen in der Linienstraße.
Wir durchsuchten noch einmal das Arbeitszimmer [...].
Endlich erreichten wir den kleinen Antiquitätenladen.
Ich fuhr zur Linienstraße.
Dann aber schlich ich mich in den dunklen Hausflur.

Most false positive sentences mention time, characters or location without explicitly signifying a change. This speaks for the assumption that the model might have overgeneralized these signals:

In der Nähe des schlesischen Bahnhofs.
"Tom, was tust Du, mußte das sein!"
Bill lag wieder still.
Auch Tom lauschte und schien unschlüssig zu sein.
Isaak Kornblum besaß Telephon.
Ich tat es.

On the other hand, many of the false negatives contain similar signals. This puts the assumption that the model might have overgeneralized upon such signals into question. Of course, one needs to consider that the majority of dimensions of the respective embeddings encode sentences from the context of a particular target sentence. Given this fact in combination that with the observation that false positives and false negatives share similar patterns, it seems very likely that these local context sentences have played a major role for classification. The following utterances are examples for false negatives:

Tom eilte jetzt die Treppe empor [...].
Mein Weg ging über die Gartenmauer.
Dann verschwand er lautlos durch die Vordiele.
Wir [...] verließen schnell den Laden.
dann stieg er die Leiter empor.
Tom verschwand schnell durch die Verbindungstür [...].

5 Conclusion & Outlook

I presented my submission to the shared task on scene segmentation at KONVENS 2021, a system aimed at segmenting German narrativew texts into distinct scenes, spans of text where character constellations, discourse- and story time, and locations stay more or less the same. For its implementation, the task was interpreted as a sentence in context classification task. For solving this task, I first trained a neural model consisting of two *German-BERT* networks, the *sentence encoder* and *context encoder*, which, in conjunction, predict contextualized sentence embeddings. This was conducted in a twin network setup where triplets of two sentences and an according score were fed to a a linear layer responsible for predicting such an according score.

The goal behind this was to train a model which would be able to embed sentences into a vector space in which sentences functioning as scene borders would be well-separated from in-scene ones which could then be used as feature vectors in regular classification. While the model indeed learned a vector space in which sentences were more or less sorted into two distinct clusters, these clusters did not seem to capture a general understanding of the concept of scene borders. This is shown by the observation that gold standard scene borders from the trial set were sorted into both clusters when embedded by the model.

For this reason, *gradient boosting* was chosen as a subsequent classification algorithm for its ability to isolate a subset of features which would still be able to separate classes well. Early stopping was used during training, meaning that the model was trained for 5000 iterations on the shared task training data and the iteration of the model which achieved best results on the trial data set was chosen as final. This achieved comparably poor results with micro F1 scores of 0.16 for track 1 respectively 0.26 for track 2. Nonetheless, these results were sufficient for ranks 2/4 respectively 1/5 in the two tracks.

It is an interesting observation that my system performs better for highbrow literature in spite of the fact that its training data consisted solely of dime novels as it contradicts the assumption of the authors that dime novels would be potentially easier to deal with for participating systems compared to highbrow literature. A possible explanation for this could lie in the more formal nature of highbrow literature which might result in more regularities that are useful for successful classification. However, without further inspection, this remains speculation.

Further work could be the optimization of the architecture and training procedure of the contextualized sentence embedding model presented in this paper. This might lead to improved downstream training results. Moreover, as gradient boosting functions as feature-based learning algorithm, it could be an option to combine contextualized sentence embeddings with statistical and hand-crafted features for representing sentences in context. In general, it can be said that the problem is far from solved as sugggested by the poor results. However, the idea of learning contextualized sentence embeddings and the optimization of the according training procedure could be a useful option to for future work on the topic.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of 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.

Marti A. Hearst. 1997. Text tiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 198–209, Copenhagen, Denmark. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. 1999. Boosting algorithms as gradient descent. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 512–518, Cambridge, MA, USA. MIT Press.

Hemant Misra, François Yvon, Olivier Cappé, and Joemon Jose. 2011. Text segmentation: A topic modeling perspective. *Information Processing & Management*, 47(4):528–544.

Karl Pearson. 1901. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Karl Pichotta and Raymond J. Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2800–2806. AAAI Press.

Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6639–6649.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Martin Riedl and Chris Biemann. 2012. TopicTiling: A text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42, Jeju Island, Korea. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. 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, Annekea Schreiber, and Nathalie Wiedmer. 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, Online. Association for Computational Linguistics.

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*.