Assessing the State of the Art in Scene Segmentation

Albin Zehe
University of Würzburg

Elisabeth Fischer
University of Würzburg

Andreas Hotho
University of Würzburg

{zehe,e.fischer,hotho}@informatik.uni-wuerzburg.de

Abstract

The detection of scenes in literary texts is a recently introduced segmentation task in computational literary studies. Its goal is to partition a fictional text into segments that are coherent across the dimensions time, space, action and character constellation. This task is very challenging for automatic methods, since it requires a high-level understanding of the text. In this paper, we provide a thorough analysis of the State of the Art and challenges in this task, identifying and solving a problem in the training procedure for previous approaches, analysing the generalisation capabilities of the models and comparing the BERT-based SotA to current Llama models, as well as providing an analysis of what causes errors in the models. Our change in training procedure provides a significant increase in performance. We find that Llama-based models are more robust to different types of texts, while their overall performance is slightly worse than that of BERT-based models.

1 Introduction

Research in the area of computational literary studies (CLS), where one of the main goals is to provide a computational understanding of literary texts, often has to deal with texts that are much longer than other areas of NLP. This leads to challenges for tasks like co-reference resolution, where models scale very badly with input length (Lee et al., 2017; Joshi et al., 2020) and their performance deteriorates on longer texts. Therefore, splitting these texts into coherent segments can be very helpful for further analysis. Additionally, a segmentation that is rooted in narrative theory can also be used to gain insights into the development of the plot of a story (Konle and Jannidis, 2022).

In this paper, we address several questions with regard to the State of the Art in Scene Segmentation. Our main contributions are as follows: (1) We identify a problem in the current state of the art model based on Sequential Sentence Classification and propose a modified training sample generation scheme designed to alleviate this issue. (2) We analyse how the relation between training and test data influences the performance of Scene Segmentation by employing training and test sets composed of different types of literature (dime novels, high literature and an additional out of distribution test set). (3) We question the previous evaluation metrics used for Scene Segmentation, arguing that the exact F1-score underestimates the performance of the models. (4) We evaluate the performance of current LLMs, specifically versions of Llama-3 and GPT-4o, in both a zero-shot prompting and a fine-tuning setting to see how they compare to the SotA in Scene Segmentation, which is still based on BERT. (5) Finally, we analyse the errors made by the best model and identify open challenges.[1]

2 Task Description

The task of automatic scene segmentation was formally defined by Zehe et al. (2021a). The goal is to segment a literary text into scenes, which are parts of the text with a consistent pattern in the four dimensions time, space, character and action. A break in these dimensions corresponds to a scene change. In addition to these scenes, texts can also contain *non-scenes*, which are (usually short) parts of the text without a consistent pattern in the four dimensions. These non-scenes are commonly used for introductory segments before a scene or to summarise the events of a longer period of time. Zehe et al. (2021a) define the task of scene segmentation as one of two sentence-level classification tasks: The simpler, binary version only uses the classes `NOBORDER` and `BORDER` (ignoring the distinction between scenes and non-scenes), while the full task has the four classes `SCENE-SCENE`,

[1] Code and data are available at `https://github.com/LSX-UniWue/scene-segmentation`.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 9922–9941
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

SCENE-NONSCENE, NONSCENE-SCENE or NOBORDER. In this paper, we focus on the binary task for several reasons: (a) the number of non-scenes in a text is usually very small compared to the number of scenes (cf. Table 5), meaning that the dataset for fine-grained classification is even more imbalanced than for binary classification, (b) previous work has shown that even the binary task is very challenging (Zehe et al., 2021a,b), (c) the distinction between scenes and non-scenes is not necessary for most subsequent analyses like co-reference resolution and the tracking of developments over time and (d) the classification into scenes and non-scenes can be done in an independent second step. Therefore, we define scene segmentation as a binary classification task of each sentence in a text as either a scene border or no scene border.

Note that our labels can directly be mapped to IOB2 labels (Tjong Kim Sang and Veenstra, 1999): The scene-begin label B is identical to our BORDER label and the within-scene label I corresponds to NOBORDER. We decided to use the BORDER/NOBORDER labels for consistency with previous work (Zehe et al., 2021a).

3 State of the Art

The concept of scenes has existed in narratology for a long time (Genette, 1983) as a part of the narration where the amount of time that passes in the narrative (story time, or *histoire*) and the amount of time covered by its narration (narrated time, or *discours*) are roughly equal. While there have been some earlier computational approaches to scene segmentation (Kozima and Furugori, 1994; Reiter, 2015), the task has more recently been discussed in detail (Gius et al., 2019) and finally been formalised, along with the introduction of an annotated dataset and experiments based on a BERT-model (Zehe et al., 2021a). Consequently, there has been a shared task (Zehe et al., 2021b) introducing several new ideas. The existing approaches can be divided into multiple categories:

Unsupervised Baseline Approaches Zehe et al. (2021a) present two baseline approaches using standard text segmentation techniques, namely TextTiling (Hearst, 1997) and TopicTiling (Riedl and Biemann, 2012). Unsurprisingly, both of these approaches are not able to capture the notion of scenes at all, as they are not designed with the notion of scenes in mind.

End-to-End Deep Learning Approaches The next group of models are based purely on end-to-end deep learning with no additional features.

The first approach in this group is the BERT-based model presented as a supervised baseline by Zehe et al. (2021a). A pre-trained BERT-model[2] is fine-tuned in a leave-one-text-out fashion to classify, for each sample, whether there is a scene boundary before *sentence*.

Another approach purely based on an end-to-end-model was proposed by Kurfali and Wirén (2021) and also used by (Ehrmanntraut et al., 2023), who apply the sequential sentence classification model proposed by Cohan et al. (2019) to scene segmentation. We directly build on this approach and provide a detailed description in Section 4.1.

Multi-Stage Approaches The models in this group use different kinds of multi-stage approaches, that is, they convert the text into some vector representation by use of a first model and then use this representation as input to a second model, which is trained to predict scene boundaries.

Gombert (2021) aim at building a sentence representation that distinguishes between sentences forming a scene border and sentences within a scene. To this end, they first train a model based on sBERT (Reimers and Gurevych, 2019), taking into account both a sentence and its context. Since they notice that this representation on its own is not enough to distinguish between the classes reliably, they they use it as input for a gradient-boosted decision tree ensemble for classification.

Schneider et al. (2021) introduce an "Embedding Delta Signal", which is based on a sliding window of word embeddings clusters. For each window, they build a vector representation counting and normalising the number of word embeddings assigned to a set of clusters within the first and the second window. They define the cosine distance between these two vectors as a measure of the topical difference between the two parts and select the windows with the highest cosine distance as scene borders. In an additional step, they train an SVM to classify each of their detected segments as either a scene or a non-scene.

Knowledge-Enriched Approaches The final set of models is based on the knowledge of which text elements are relevant to the definition of scenes, specifically time, place and character constellation.

[2] https://deepset.ai/german-bert.

Barth and Dönicke (2021) manually design general (tense of verbs, POS tags, etc.) and scene-specific (temporal and locational expressions and entity mentions) features and use these as input to a random forest classifier.

Hatzel and Biemann (2021) focus on the character constellation, using entity features extracted from the text by use of a co-reference model (Schröder et al., 2021) in addition to a BERT-based encoder. Entity features and the BERT-representation are fed into a linear layer, which is trained to detect scene borders. Since this model tends to predict too many scene boundaries in close vicinity, the authors add an algorithm to aggregate the local decisions of the model (which only takes into account a small area of the text) to global decisions, where these very short scenes can be penalised harshly.

4 Approaches to Scene Segmentation

We compare different approaches to our task, based on either a BERT-style model or a current LLM.

4.1 BERT – Sequential Sentence Classification

Our initial model, following (Kurfali and Wirén, 2021; Ehrmanntraut et al., 2023), is based on the Sequential Sentence Classification (SSC) architecture introduced by Cohan et al. (2019). The main idea is to use a pre-trained language model (e.g., BERT) to get a representation for a sequence of sentences and then assign each of these sentences to a set of classes C. To this end, the input sequence is pre-processed in a specific way, adding a `[SEP]` token after every sentence. For example, the input 'This is a sample. It consists of two sentences' would be transformed to '[CLS] This is a sample. [SEP] It consists of two sentences. [SEP]'. After passing this input through a language model, building contextualised representations for each token, the representations of these `[SEP]`-tokens are then used as input to a classifier, which predicts a class label for each `[SEP]` token (i.e., each sentence) individually. Figure 1 provides a visualisation of the model. We follow Kurfali and Wirén (2021); Ehrmanntraut et al. (2023) in adapting this architecture for scene segmentation, since it provides a natural way of detecting whether a sentence starts a new scene while taking into account the context provided by the surrounding sentences.

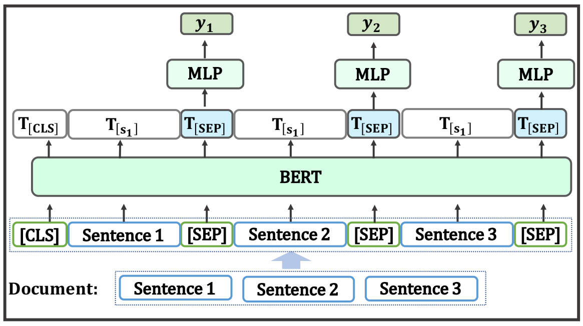

Figure 1: The Sequential Sentence Classification model proposed by Cohan et al. (2019). Visualisation copied from the original paper.

4.2 Llama – Generative Scene Segmentation

Since the current State of the Art in NLP is usually based on pre-trained Large Language Models like Llama 3 (Dubey et al., 2024), we want to evaluate how these models compare to the performance of our BERT-based classifier. The SSC approach described in the previous section cannot be adapted for decoder-based models like Llama in a straightforward way, mostly for two reasons: First, Llama does not contain a separator-token in the sense of the BERT-model. While it is easily possible to add a special token for this purpose, this token would not be pre-trained to serve as a separator between multiple sentences, as is the case for the BERT model. Secondly, using Llama to encode the input in the same way as the BERT-model would give the representations of the SEP-tokens access to only the tokens up to the respective sentence, not the following sentences, due to the decoder-based nature of the model. This is problematic for scene segmentation, where the model needs to decide whether there is a significant change in the text at the location of the SEP token.[3] Due to these issues, we decided to not employ the Llama-models for the Sequential Sentence Classification approach, but in two ways based on the text generation task they are usually used for.

Prompting First, we employ different variations of the Llama 3 model in a prompting setting. We build a prompt (cf. A.2) by giving the model a short description of what we define as a scene and a snippet of text with one sentence marked and then asking it whether the marked sentence starts a new scene, as well as to provide a reason for its answer.

[3] Approaches to alleviate this like LLM2Vec (BehnamGhader et al., 2024) are promising directions for future work.

Fine-tuning In addition to the pure prompting task for the pre-trained models, we also fine-tune the models for our task, comparing different strategies for building the prompt: **No-CoT**: The first prompt is the same as used in the prompting setting. It does not employ Chain of Thought (Wei et al., 2022) prompting, but rather asks the model to first provide a yes/no answer and then follow up with a reason. **CoT-List**: The prompt is designed to guide the model in its reasoning process, asking it to go over each of the dimensions used in our scene definition explicitly, answering "There is [a/no] significant change in [dimension]" and finally provide a classification. The full prompts and response templates are given in Appendix A.2.

5 Data

We use an extended version of the dataset from Zehe et al. (2021b), containing annotations for 41 texts or text fragments, with Table 5 (Appendix A.8) providing a full list of annotated texts as well as statistics about text length, the number of (non-)scenes and their average length for each text. Note that, in this paper, we make no distinction between scenes and non-scenes and only provide the statistics separately for informative purposes. We can see that the number of sentences in a text is much higher than the number of scenes, meaning that scene borders are indeed rare, leading to an imbalanced classification problem. The texts that are not part of the previous dataset (Zehe et al., 2021b) are marked in bold in Table 5. They have been annotated by student assistants and university employees with background in computational literary studies according to the same guidelines (Gius et al., 2021) and using the same annotation procedure as the original dataset. Following (Zehe et al., 2021a), we calculate Inter Annotator Agreement using Mathet's γ (Mathet et al., 2015), reaching a value of 0.827 for the newly annotated texts.[4]

We form several different training and test datasets from these annotated texts, in order to evaluate the generalisation capabilities of our models: **STSS-Train**: the original training dataset from the STSS, consisting entirely of dime novels, **Train-with-High**: a new dataset consisting of STSS-Train and high literature texts, **Train-Full**: Train-with-High with additional dime novels, **STSS-Test-1**: the original test dataset for track 1 of STSS, also consisting entirely of dime novels, **STSS-Test-2**: the original test dataset for track 2 of STSS, consisting entirely of high literature texts, **OOD-Test**: a new out-of-distribution test dataset consisting of texts that are neither dime novels nor high literature **Test-Full**: STSS-Test-1 + STSS-Test-2 + OOD-Test.

The assignment of texts to the different sets is given in Table 6. Since many of the texts both in the original dataset as well as in our extension are copyrighted, we cannot publish the full dataset and choose the same access modality as (Zehe et al., 2021a,b): We provide access to standoff annotations for research purposes and provide help in merging them to the full texts upon request.

6 Experiments

6.1 Evaluation Metrics

The evaluation of scene segmentation poses some challenges of its own, wherefore multiple evaluation metrics have been used in the past. In their original task description paper, Zehe et al. (2021a) propose to use both the F1-score for an exact match of predicted and annotated scene boundaries, and one component of Mathet's γ (Mathet et al., 2015) for a more lenient score. In this paper, we propose the use of a less strict modification of the F1-score, trying to capture the advantages of both metrics. We provide a discussion of additional possible metrics in Appendix A.3.

Exact F1-Score An exact F1-score can be computed for scene segmentation by labelling each sentence in the text as either BORDER or NOBORDER and computing an F1-score for these classifications. This exact F1-score is a very strict metric for the evaluation of scene segmentation, since it does not allow even a very small deviation of the predicted boundaries from the gold annotations. That means that moving the boundary by one sentence in either direction would be counted as a complete miss, which is not well-aligned with the task of scene segmentation: missing the annotated border by a few sentences may in some cases not even be an error at all, since the essence of the scene is still captured by the predicted border.

[4] Refer to Appendix A.1 for a discussion of the problems of this metric. We use the implementation from https://github.com/bootphon/pygamma-agreement/ with default parameters. This gives us $\gamma = 0.608$ for the annotations from (Zehe et al., 2021a), which is notably lower than what is reported in the paper. However, since our new annotations reach a higher γ than the original dataset, their quality can be considered sufficient.

Mathet's γ Zehe et al. (2021a) use Mathet's γ as both an inter-annotator agreement score as well - in a modified way - as an evaluation metric. While γ seems fitting due to its capability of dealing with multiple segment types (scenes and non-scenes) and alleviating the very harsh requirement of the F1-Score to match the exact right sentence, we encountered several problems with it in practice: First, the metric requires setting multiple hyperparameters whose optimal values are not intuitively clear. Secondly, we found multiple implementations of the metric, which sometimes return significantly different values of γ for the same data (cf. Appendix A.1). Due to these problems, we decided to drop the use of γ as an evaluation metric.

Relaxed F1-score We propose to use a variant of the F1-score, which we call relaxed. For this score, we apply a *tolerance* t to the predictions made by the model: We add a post-processing step to the predicted labels, where, if a predicted scene border is within t sentences of a gold annotated scene border, we move the predicted scene border to the correct position. After that, we calculate the sentence-level F1-score as usual. We use the relaxed F1-score for evaluation and set the tolerance to $t = 3$, meaning that a predicted scene border is considered correct if it is no more than three sentences next to a gold annotated scene border. All reported scores are for the minority class `BORDER`. We compute the dataset-level score as the unweighted average of the scores for all texts in the dataset.

6.2 BERT – Sequential Sentence Classification

Embedding Model We evaluate three different embeddings models for the Sequential Sentence Classification architecture: Kurfali and Wirén (2021) originally used a `GBERT-Large` model[5]. We compare this to a smaller `GBERT-Base` model[6] as well as the `Fiction-GBERT-Large`[7] used in Ehrmanntraut et al. (2023) that was additionally fine-tuned on German literature.

Sample Generation Kurfali and Wirén (2021) train the model by splitting a novel into non-overlapping context windows of a size corresponding to the language model used. Each of these windows corresponds to one training sample, where the model is trained to predict, for each sentence in the window, whether there is a scene border at this sentence. We call this strategy `full-stride`, since it always advances the current position by a full window.

We argue that this introduces a problem where, for some sentences, the model has access to very limited context on one side. In the extreme cases (first and last sentence of a window), the model does not have any information about the text preceding or following this sentence and needs to make the prediction purely based on the context on the other side. This is undesired, since scene borders are defined by a change in the dimensions space, time, action or character constellation and therefore can only be reliably detected by taking into account the context on both sides of the potential border. We start by analysing whether our intuition is correct and this does indeed cause a notable amount of errors. To this end, we train an SSC model with `Fiction-GBERT-Large` as its embedding model, analogous to (Ehrmanntraut et al., 2023) on `Train-Full` and evaluate it on `Test-Full`. Figure 2 shows the count of true positives and false negatives grouped by distance to the closest window border. We see that the full stride (bars labelled with `f`) used in previous research leads to many false negatives directly at the window border, where no or very limited context is available on one side of the current sentence.

In order to address this issue, we propose to build the context windows in a different manner: We build context windows with overlap, starting window $n + 1$ at the sentence in the middle of window n, and refer to this strategy as `half-stride`. Since this means that a sentence is part of multiple windows, we also get multiple predicted labels for each sentence. We select the prediction which has the most balanced context (measured by number of sentences on either side of the target sentence) as the final prediction. Training the same model with the `half-stride` strategy naturally addresses this, since samples with very limited context on one side of the current sentence are only used for classification if the sentence is at the beginning or the end of the full text. The bars labelled with `h` in Figure 2 show the results for the same model trained with the `half-stride` strategy. We see that the errors directly at the window borders vanish, as expected. We will see in Section 7 that this training strategy also improves the results overall.

[5] `https://huggingface.co/deepset/gbert-large`.

[6] `https://huggingface.co/deepset/gbert-base`.

[7] `https://huggingface.co/lkonle/fiction-gbert-large`.

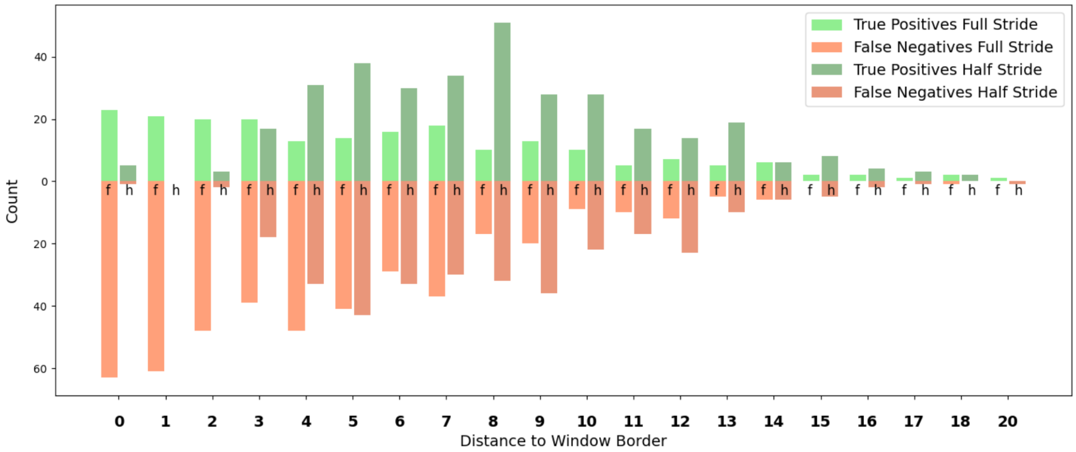

Figure 2: Correctly and incorrectly classified scene borders by distance to the closest window border using full (f) and half (h) stride. Sentences with 0 distance are directly at the border of the context window used for classification.

Hyper-Parameters Unless otherwise specified, we use a random seed of 1 and train all models for 5 epochs with learning rate 1×10^{-6} using the default HuggingFace trainer. We fine-tune the embedding model along with the classification head.

6.3 Llama – Generative Scene Segmentation

We use Llama models in the two ways described above: Prompting and fine-tuning.

Prompting For prompting, we use an Ollama API[8] and the three different models Llama 3 8b[9], 70b[10] and the recently released Lllama 3.1 405b[11].

We iterate over the texts in the test dataset, constructing a sample from each sentence by wrapping it in a `<sentence>...</sentence>` marker and surrounding it with context on either side up to a total of up to 512 tokens, balanced by the number of sentences. We keep the context size at 512 to allow a fair comparison to the BERT-based models. Note that, while only the Llama 3.1 models are officially marked as being multilingual, initial experiments confirmed that Llama 3 is able to understand our German input texts well enough.

Fine-Tuning We use the Unsloth library[12] for fine-tuning the models. We fine-tune a 4bit-quantised version of Llama 3 8b[13] using QLoRA adapters, with the config listed in Appendix A.4. We do not fine-tune the 70b and 405b version due to lack of computational resources. We generate a training sample for each sentence as in the prompting approach. However, due to the high computational cost of training the Llama models and the very imbalanced label distribution, we sub-sample the non-border sentences: We select all positive samples (label `BORDER`) and 10% of the negative samples (label `NOBORDER`). For evaluation, we build samples for each sentence in the test sets in the same way (without sub-sampling). We use regular expressions to match the answer templates for the respective prompt to extract the classification and reasoning from the text generated by the model.

7 Results

7.1 BERT – Sequential Sentence Classification

The first two lines in Table 1 show the results of an SSC model analogous to Ehrmanntraut et al. (2023) (LLPro) trained on `Train-Full` and evaluated on the different test splits, as well as the same model trained with our modified sample generation (LLPro + Half Stride). We see that, across all datasets, the `half-stride` strategy provides clear improvements in performance. The next four lines provide results for using `GBERT-Large` and `GBERT-Base` as an embedding model. Note that the `GBERT-Large` setting is a reimplementation of the best model from the STSS (Kurfali and Wirén, 2021). Again, `half-stride` consistently improves results across all datasets. We also see that performance between `GBERT-Large` and `Fiction-GBERT-Large` is generally comparable, while `GBERT-Base` clearly performs worse. Interestingly, contrary to the findings in Ehrmanntraut et al. (2023), `GBERT-Large` outper-

[8] https://ollama.com/.
[9] https://ollama.com/library/llama3:8b.
[10] https://ollama.com/library/llama3:70b.
[11] https://ollama.com/library/llama3.1:405b.
[12] https://github.com/unslothai/unsloth.
[13] https://huggingface.co/unsloth/llama-3-8b-Instruct-bnb-4bit.

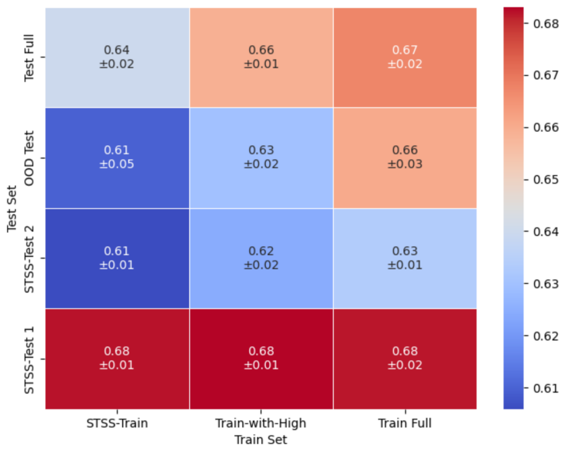

Figure 3: Median F1-Scores over 5 seeds for BERT SSC models trained and evaluated on different test sets. E.g., the lower left cell shows the result for the model trained on `STSS-Train` and evaluated on `STSS-Test-1`

forms `Fiction-GBERT-Large` in our experiments.

To analyse the relation between training and test data, that is, the generalisation capabilities of the models and the stability, we train the best-performing model with 5 different seeds on each training dataset and evaluate the models on each test dataset. Figure 3 shows the median results and standard deviation over the 5 seeds for each combination of training and test dataset. Our main takeaways from this figure are as follows: (a) The performance on `STSS-Test-1` (bottom row in Figure 3) does not change with the additional training from the expanded sets. This is very interesting, as more training data, either from another domain (`Train-with-High`) or even from the same domain (`Train-Full`) does not seem to make the model better. (b) Providing more training data, including high literature texts (both `Train-with-High` and `Train-Full`), leads to a slight improvement on the high literature test set `STSS-Test-2` (second row from the bottom) and a notable improvement on `OOD-Test` (second row from the top). Consequently, performance on `Test-Full` also improves. (c) Overall, the model seems to not generalise well from dime novels in the training data to different types of literature in `STSS-Test-2` and `OOD-Test`. Specifically, as expected from literary theory, the high literature texts seem to be the hardest.

We also analyse the influence of our relaxed F1-Score on the results by computing the score with different values for the tolerance t. Figure 4 shows the relaxed F1-Scores for $t \in \{0, 1, 2, 3\}$. Note that $t = 0$ corresponds to a regular, strict F1-Score, where a border needs to be predicted at exactly the right position. We find that, while the values increase with tolerance, the ranking of the models always stays the same. It is also notable that the better models seem to benefit more from the tolerance. This suggests that these models frequently make good predictions that are off by some sentences, while the worse models just make completely wrong predictions. We decided to focus on the evaluation with the Relaxed F1-score since, as argued in section 6.1, it provides the more realistic estimate of the model's performance.

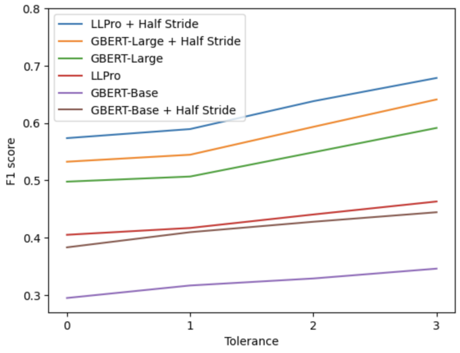

Figure 4: Relaxed F1-Scores with different tolerance. $t = 0$ corresponds to exact F1-Score.

7.2 Llama – Generative Scene Segmentation

Prompting Zero-short prompting of Llama 3 models does not perform well overall. Llama3:8b reaches a relaxed F1-score of 0.13 on `Test-Full`. Manual inspection of the predictions suggests that the model does not understand the task and predicts a scene border for almost every sentence because of minor plot progression. Llama3:70b performs significantly better, reaching an F1-score of 0.34, close to the results achieved by a fine-tuned `GBERT-Base` model, but still clearly behind our best BERT-based models. Llama3.1:405b performs almost the same as the 70b model, also reaching an F1-score of 0.34. For comparison, we also evaluate the prompting approach with ChatGPT.[14] While 4o-mini performs comparably to llama3:8b (Relaxed F1-score of $14\,\%$), 4o performs notably better than the -mini version and also than llama3:70b ($45\,\%$ Relaxed F1). All prompting results are given in Table 2 However, since these results are still quite a bit off from the best BERT models, we do not analyse the prompting approach further.

[14] Specifically, the models gpt-4o-2024-08-06 and gpt-4o-mini-2024-07-18.

Table 1: Relaxed F1-Scores (tolerance $t = 3$) for SSC models trained on `Train-Full` and evaluated on all test sets.

Model \ Test Dataset	STSS-Test-1	STSS-Test-2	OOD-Test	Test-Full
LLPro + Half Stride	0.62	0.57	0.66	0.62
LLPro	0.51	0.47	0.43	0.47
GBERT-Large + Half Stride	**0.68**	**0.66**	**0.69**	**0.68**
GBERT-Large	0.59	0.61	0.60	0.60
GBERT-Base + Half Stride	0.37	0.46	0.46	0.42
GBERT-Base	0.26	0.37	0.33	0.31

Table 2: Relaxed F1-Scores (tolerance $t = 3$) for Llama and ChatGPT models prompted on `Test-Full` without fine-tuning

Model	Relaxed F1-score
llama3:8b	0.13
llama3:70b	0.34
llama3.1:405b	0.34
gpt-4o-mini	0.14
gpt-4o	**0.45**

Fine-Tuning As described in Section 4.2, we use two different prompt templates (`No-CoT` and `CoT-List`) for fine-tuning the Llama model. Table 3 shows the performance of these models. We refer to Appendix A.10 for full details. Three things stand out: (a) models trained with `CoT-List` very clearly and consistently outperform models trained with `No-CoT`, (b) the model trained with `CoT-List` performs comparably to the best BERT models and (c) the Llama-based models are less sensitive to the different types of literature, generalising better from dime novels to high literature than the BERT-based models. Overall, the Chain of Thought reasoning is very beneficial to the models. However, the best Llama-based models (relaxed F1-score of 0.62) are still slightly worse than the best BERT-based models (0.68).

8 Analysis

We are interested in finding out what causes our models to either miss (false negative) or add (false positive) scene borders. To this end, we analyse our best model's predictions on the test data in detail.

Visual Error Analysis As a first step for this analysis, we build a visualisation of the predictions from the SSC model based on `GBERT-Large` with `half-stride` and their errors. Figure 5 (Appendix A.9) shows this visualisation for two texts. For the Harry Potter chapter, the model reaches an F1-Score of 0.77 (Precision: 0.63, Recall: 1). We see that all gold scene borders are detected correctly, but the model additionally predicts three false positives, provided verbatim in Appendix A.5. In all three cases, there is a change in character constellation and to a lesser extent in narration, which has not been deemed as significant enough to warrant a scene change by the annotators, but could be argued to actually start a new scene. For the second text (F1-Score: 0.76, Precision: 0.78, Recall: 0.75), the model produces both false negatives and false positives. Interestingly, 8 of the 14 false negatives in this text are borders between scenes and non-scenes: non-scenes are often merged to one of the surrounding scenes.

False Positives We manually went over all false positives produced by the model on the datasets `STSS-Test-1` and `OOD-Test`. In almost all cases, there was a clear reason visible for why the model predicted a scene border. In many cases, these are valid markers for scene changes that have been judged by the annotators as not significant enough to start a new scene. In two cases, a scene starts with a few lines of dialogue followed by an introductory, narrative sentence and the model detects the border at the introductory sentence. In several cases, either a plan for the future or a memory from the past was described, making it seem like a new scene has started (due to time, location and characters changing). Another source of errors, which we already discovered in the previous analysis, are time jumps in non-scenes, which are detected by the model as scene borders. In one case, the predicted border was correct, but had been missed by the annotators. We provide (informal) notes for each false positive in Appendix A.6. Overall, we consider 25 of the inspected false positives possible scene borders, and 14 as clearly wrong.

Table 3: Relaxed F1-Scores (tolerance $t = 3$) for Llama3:8b models trained on `Train-Full` with different Chain of Thought (CoT) prompting strategies and evaluated on all test sets.

CoT Config \ Test Dataset	`STSS-Test-1`	`STSS-Test-2`	`OOD-Test`	`Test-Full`
No-CoT	0.05	0.04	0.18	0.09
CoT-List	**0.55**	**0.63**	**0.69**	**0.62**

False Negatives For false negatives, we analyse the reasons given for scene borders by the annotators of the gold data. We count the number of false negatives and true positives annotated with each of the possible reasons (narrative action, time, space and location), expecting that, for example, temporal markers should be easier to detect and therefore have a high true positive rate. However, we find no notable differences between the different reasons, with the true positive rate for all reasons ranging from 55.92% (location) to 58.88% (time). In addition, we manually analyse the texts from `OOD-Test` and one text from `STSS-Test-1`, "Im Bann der Vampire". Again, informal notes for possible false negative reasons are given in Appendix A.7. We found five cases where it is not clear why the border was missed. The most frequent apparent reason for missing borders was that the model merged a non-scene to the previous or following scene. Apart from that, the model sometimes misses implicit markers for scene changes, where no direct reference to time, location or characters is provided. In the case of "Hänsel & Grethel", the narrative style is very different from that in the other texts, with large parts of the story being narrated in almost a non-scene style. This seems to cause several errors where small jumps in time are not annotated as scene changes. At the end of this text, there is a remark made by the narrator, breaking out of the story "Mein Märchen ist aus, dort lauft [sic!] eine Maus, wer sie fängt, darf sich eine große große Pelzkappe daraus machen."[15], which is not annotated by the model as a new scene. Overall, we consider 8 of the inspected false negatives as arguably non-borders and 12 as definitely missing.

9 Discussion

Our experiments reveal several interesting findings regarding scene segmentation. First, we find that building samples for SSC using the `half-stride` strategy greatly improves the performance. Secondly, we find that BERT-based models are sensitive to the type of texts in training and test set – mixing dime novels and high literature in the training set leads to better performance on non-dime novels. However, we also see that the model's performance on dime novels does not improve with the additional training data. This is very interesting, as it suggests that, contrary to our assumption, the performance of scene segmentation is not limited by the amount of available training data, but rather by the capabilities of the model. Our analysis reveals challenging cases and suggests some changes in the way the task is approached: Since borders to non-scenes cause a notable amount of errors, our decision to not include this distinction may have to be revised: as it is, the model does not have any indication that there are structurally different segments in the texts. Distinguishing between scenes and non-scenes may therefore actually be beneficial. Manual analysis of the errors shows that, in many cases, the predictions of the model can be argued to be correct, depending on how fine-grained the segmentation is supposed to be. This suggests that, even though the metrics still leave room for improvement, the produced segmentation can be sufficient for downstream tasks.

10 Conclusion

We have explored the State of the Art for automatic scene segmentation in fictional texts. We achieve a new best result for the task, with a relaxed F1-Score of 0.68 on a diverse test set with the best model. While this still leaves room for improvement, it shows that it is feasible to train models that achieve good results for automatic scene segmentation. Our analysis shows that many of the best model's wrong predictions are still understandable and could be argued to be correct in many cases, depending on how fine-grained one wants the scene segmentation to be. We have also demonstrated that the task is challenging even for current large language models, making it a valuable addition to benchmarks aiming to explore the limits of these models.

[15] "My story is over, there is a mouse running, who catches it may make a big big fur hat of it."

Limitations

Scenes vs Non-Scenes In the current paper, we only perform scene segmentation by detecting borders between segments, ignoring the distinction between scenes and non-scenes. For the full task as defined by Zehe et al. (2021a), this distinction needs to be considered. However, the classification of a segment as either scene or non-scene is expected to be much easier than the segmentation (Zehe et al., 2021a).

Dataset Our experiments are only performed on one dataset of German texts, potentially raising questions about the generalisability of the results. However, to the best of our knowledge, this is currently the only existing dataset for the task of scene segmentation.

Ethics Statement

We perform experiments on an extended version of an existing dataset of fictional texts that are publicly available (although partially only for purchase). We do not see any ethical concerns regarding the task of scene segmentation in itself, or its future applications. The models presented in this paper are purely intended to enable the analysis of fictional texts.

Acknowledgements

This publication was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the project LitBERT, project no. 529659926.[17]

References

Florian Barth and Tillmann Dönicke. 2021. Participation in the konvens 2021 shared task on scene segmentation using temporal, spatial and entity feature vectors. In *Shared Task on Scene Segmentation*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Dan Weld. 2019. Pretrained language models for sequential sentence classification. 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 3693–3699, Hong Kong, China. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor

[17] `https://gepris.dfg.de/gepris/projekt/529659926`.

Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Anton Ehrmanntraut, Leonard Konle, and Fotis Jannidis. 2023. LLpro: A literary language processing pipeline for German narrative texts. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 28–39, Ingolstadt, Germany. Association for Computational Linguistics.

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

Evelyn Gius, Fotis Jannidis, Markus Krug, Albin Zehe, Andreas Hotho, Frank Puppe, Jonathan Krebs, Nils Reiter, Nathalie Wiedmer, and Leonard Konle. 2019. Detection of scenes in fiction. In *Proceedings of Digital Humanities 2019*.

Evelyn Gius, Carla Sökefeld, Lea Dümpelmann, Lucas Kaufmann, Annekea Schreiber, Svenja Guhr, Nathalie Wiedmer, and Fotis Jannidis. 2021. Guidelines for detection of scenes.

Sebastian Gombert. 2021. Twin bert contextualized sentence embedding space learning and gradient-boosted decision tree ensembles for scene segmentation in german literature. In *Shared Task on Scene Segmentation*.

Hans Ole Hatzel and Chris Biemann. 2021. Applying coreference to literary scene segmentation. In *Shared Task on Scene Segmentation*.

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

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Leonard Konle and Fotis Jannidis. 2022. Modeling plots of narrative texts as temporal graphs. In *CHR*, volume 3290 of *CEUR Workshop Proceedings*, pages 318–336. CEUR-WS.org.

Hideki Kozima and Teiji Furugori. 1994. Segmenting narrative text into coherent scenes. *Literary and Linguistic Computing*, 9(1):13–19.

Murathan Kurfali and Mats Wirén. 2021. Breaking the narrative: Scene segmentation through sequential sentence classification. In *Shared Task on Scene Segmentation*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. Cite arxiv:1908.10084Comment: Published at EMNLP 2019.

Nils Reiter. 2015. Towards Annotating Narrative Segments. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 34–38, Beijing, 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. Association for Computational Linguistics.

Felix Schneider, Börn Barz, and Joachim Denzler. 2021. Detecting scenes in fiction using the embedding delta signal. In *Shared Task on Scene Segmentation*.

Fynn Schröder, Hans Ole Hatzel, and Chris Biemann. 2021. Neural end-to-end coreference resolution for german in different domains. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 170–181.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen, Norway. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Albin Zehe, Leonard Konle, Lea 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: Volume 1, Long Papers*. ACL.

Albin Zehe, Leonard Konle, Svenja Guhr, Lea Dümpelmann, Evelyn Gius, Andreas Hotho, Fotis Jannidis, Lucas Kaufmann, Markus Krug, Frank Puppe, Nils Reiter, and Annekea Schreiber. 2021b. Shared task on scene segmentation @ konvens 2021. In *Shared Task on Scene Segmentation @ KONVENS 2021*, pages 1–21.

A Appendix

A.1 γ Implementation Issues

The authors of γ provide both an online and an offline application.[18] For both of these applications, it is not clear how the hyper-parameters are set and they yield very different values of γ for the examples provided in the online version. There is an additional re-implementation,[19] which explains some issues with the original implementation[20] and their own implementation choices. However, even with these notes, we were unable to reproduce the values from the original implementation.

[18] https://gamma.greyc.fr/. Currently unavailable, archived version at https://web.archive.org/web/20230607124122/https://gamma.greyc.fr/.

[19] https://pygamma-agreement.readthedocs.io/.

[20] https://pygamma-agreement.readthedocs.io/en/latest/issues.html.

A.2 Llama Prompts

The gold annotations partially contain a reason (location, characters, . . .) for the scene change provided by the annotators. We use this information to fill in the templates provided below.

No-CoT Does the sentence in <sentence>...</sentence> introduce the beginning of a new scene and a significant break in time, location or characters? Answer ’True’ or ’False’ and provide a reason for your decision. A scene is defined as a segment of text with a coherent structure across the dimensions ’characters’ (which characters are present in the narration), ’location’ (where does the narration take place), and ’time’ (continuous time in the narration). A significant break in any of these dimensions corresponds to a scenes change.

→ [True/False], because there is [a/no] significant change in [narrative action, location, time or characters].

CoT-List A scene is defined as a segment of text with a coherent structure across the dimensions ’characters’ (which characters are present in the narration), ’location’ (where does the narration take place), and ’time’ (continuous time in the narration). A significant break in any of these dimensions corresponds to a scenes change. Does the sentence in <sentence>...</sentence> introduce the beginning of a new scene? Think step by step: a) Does the sentence introduce a significant change in narrative action? b) Does the sentence introduce a significant change in location? c) Does the sentence introduce a significant change in time? d) Does the sentence introduce a significant change in characters? e) Does the sentence therefore start a new scene?

→ a) There is [a/no] significant change in narrative action, b) there is [a/no] significant change in location, c) there is [a/no] significant change in time, d) there is [a/no] significant change in characters, e) therefore, the sentence [starts/does not start] a new scene.

For example, if a sentence has a scene border which is annotated with the reasons “narrative action” and “character", our target for the Llama model becomes “a) There is a significant change in narrative action, b) there is no significant change in location, c) there is no significant change in time, d) there is a significant change in characters, e) therefore, the sentence starts a new scene.”.

If no reason is annotated for a scene border, we train the model to predict that "there may be a significant change" for each dimension, and to conclude that this causes a scene border.

For the extraction of the labels from the model’s output, we use two strategies: For No-CoT, we simply check whether the response string starts with “True” or “False” and assign the class label accordingly. For CoT-List, we use the following regular expression to parse the model’s output:

```
r"a\) (?P<action>.*)"
r"b\) (?P<location>.*)"
r"c\) (?P<time>.*)"
r"d\) (?P<characters>.*)"
r"e\) (?P<border>.*)"
```

We then check for the presence of the phrase “therefore the sentence starts a new scene” in the group “border”.

A.3 Metrics

Previous work as well as the discussion with reviewers after submission of this paper has shown that there are several suitable metrics for the evaluation of scene segmentation, each with their advantages and disadvantages. In addition to the scores described in the main paper, we considered the following alternatives:

Intersection over Union Schneider et al. (2021) adapt the Intersection over Union (IoU) to scene segmentation. They describe the computation of the measure as follows: “For every ground truth part we find the detected part with the biggest overlap and assign it to the ground truth part if it has not been assigned yet. We then add the length of all the overlapping regions and normalize them by the total length of the text, resulting in an intersection over union score value for the document.” Similarly to γ this measure has the desirable property of assigning a high score to scenes that have been detected “almost” correctly (i.e., only deviate a few sentences from the gold annotation). For comparison, we also evaluated the BERT models using IoU on Test-Full, the results are given in Table 4. We see that the ranking of the approaches stays the same as with the F1-score.

Precision/Recall@k Our proposed relaxed F1-score is similar in spirit to the metrics of Precision and Recall@k that are used in Information Retrieval and Recommendation (Manning et al., 2008). In these settings, there is a list l of correct *items* and a list $\hat{l}$ of items returned by a model, sorted by their score according to the model. Precision@k then counts the number of items in the first k entries of $\hat{l}$ that are also part of l, while Recall@k counts how many of the items in l are covered by the first k entries in $\hat{l}$. Since we don't have lists of relevant items, but rather labels for each sentence, these metrics are not directly applicable. Instead, we allow the model to deviate some sentences from the location of the gold annotated border.

A.4 PEFT config

We use the configuration from the example notebook[21] provided by the Unsloth library for Llama fine-tuning:

```
model = FastLanguageModel.get_peft_model(
    model,
    r=16,
    target_modules=["q_proj", "k_proj",
                    "v_proj", "o_proj",
                    "gate_proj", "up_proj",
                    "down_proj", ],
    lora_alpha=16,
    lora_dropout=0,
    bias="none",
    use_gradient_checkpointing="unsloth",
    random_state=random_seed,
    use_rslora=False,
    loftq_config=None,
)
```

A.5 False Positive Predictions in Harry Potter

- "With a sudden exclamation she pointed at the clock's face. Mr. Weasley's hand had switched to "traveling." "He's coming!" **And sure enough, a moment later there was a knock on the back door.** Mrs. Weasley jumped up and hurried to it;"

- "But his question was answered before he could finish it. The bedroom door flew open again, and Harry instinctively yanked the bedcovers up to his chin so hard that Hermione and Ginny slid off the bed onto the floor. **A young woman was standing in the doorway, a woman of such breathtaking beauty that the room seemed to have become strangely airless.** She was tall and willowy with long blonde hair and appeared to emanate a faint, silvery glow. To complete this vision of perfection, she was carrying a heavily laden breakfast tray. "’Arry," she said in a throaty voice. "Eet ’as been too long!""

- ""You lot had better come down quickly too," she said as she left. **Harry took advantage of the temporary silence to eat more breakfast.** Hermione was peering into Fred and George's boxes, though every now and then she cast sideways looks at Harry. Ron, who was now helping himself to Harry's toast, was still gazing dreamily at the door."

A.6 False Positive Reasons

Suspected reasons for false positive predictions along with their counts (most of the reasons are too specific to occur more than one time).

- ’A new character arrives at the scene, interrupting the action’: 1

- ’A new character arrives at the scene’: 2

- ’A character leaves the scene and a short break in time occurs (zeitweilige Stille)’: 1

- ’Two characters walk away from the scene’: 1

- ’The character arrives at a new location after a short time, but the narration continues’: 1

- ’The character is relaxing and a phone call interrupts’: 1

- ’The characters arrive at a new location after a short time’: 1

- ’A new character arrives after a short time’: 1

- ’A memory is narrated, making it seem like time and location change’: 1

- ’A memory is narrated’: 1

- ’The characters walk to a different room in the same house’: 1

[21] https://colab.research.google.com/drive/135ced7oHytdxu3N2DNe1Z0kqjyYIkDXp?usp=sharing.

Table 4: Comparison of Relaxed F1-score (tolerance $t = 3$) and Intersection over Union for SSC models trained on `Train-Full` and evaluated on `Test-Full`.

Model \ Metric	F1-Score	Precision	Recall	IoU
LLPro + Half Stride	0.62	0.80	0.53	0.51
LLPro	0.47	0.82	0.35	0.35
GBERT-Large + Half Stride	0.68	0.78	0.62	0.57
GBERT-Large	0.60	0.78	0.52	0.50
GBERT-Base + Half Stride	0.42	0.80	0.31	0.33
GBERT-Base	0.31	0.80	0.20	0.22

- ’A new character is mentioned, but was already present before’: 1
- ’A car has left, but was not part of the main cast of the scene’: 1
- ’The character goes to the toilet’: 1
- ’A character has left’: 1
- ’A character is joining the scene’: 1
- ’A character is leaving and a short time passes’: 1
- ’A remote conversation is started, not really introducing new characters’: 1
- ’A short amount of time (10 minutes) has passed’: 1
- ’A short amount of time passes without dialogue’: 1
- ’A new location is reached after a short time’: 2
- ’The character moves a (probably very small) way’: 1
- ’The character is surprised by a noise while moving down a corridor’: 1
- ’A signal arrives, which does not introduce a new scene’: 1
- ’A short amount of time passes’: 1
- ’A character leaves the scene’: 1
- ’A medium amount of time passes, but the narration continues directly’: 1
- ’A character leaves and a short amount of time passes’: 1
- ’The scene has started several sentences earlier’: 1
- ’A future scene is described’: 1
- ’This should have been marked as a scene border in the gold annotations’: 1
- ’The scene started serveral sentences earlier. However, as before, there is an introductory sentence only now after some initial dialogue’: 1
- ’Border predicted slightly too early’: 1
- ’A character leaves the scene, but the narration continues’: 1
- ’This is within a non-scene, where major time jumps can occur. If it was in a scene, this would be a border’: 1
- ’This is still within a non-scene’: 1
- ’Non-scene again’: 1

A.7 False Negative Reasons

- ’There is a non-scene before this sentence, which the model has merged to the scene’: 2
- ’No obvious reason for missing the border’: 5
- ’Only a small jump in time occurs, the overall location stays the same (the train) and the narration continues’: 1
- ’There is a non-scene after this sentence, which the model has merged to the scene’: 4
- ’There is no change in time or overall location, but the focus of the narration shifts’: 1

- 'Minor time jump, same characters, same narration': 1

- 'Very uncommon case, where the narrator deviates from the main story at the very end and makes a personal remark': 1

- 'Minor, implicit time jump, no location change, same characters, similar narration': 1

- 'The scene border was predicted earlier, which is ok. There is a short non-scene-like narration in between which could be included in either scene': 1

A.8 Dataset Statistics

Table 5: Statistics about the texts in our dataset. Average (non-)scene length is measured in tokens (i.e, words or punctuation marks), document length is measured in sentences as detected by our parser. Texts marked in **bold** are new additions over the previous dataset (Zehe et al., 2021b).

Name	Scenes	Non-Scenes	Average Scene Length (Tokens)	Average Non-scene Length (Tokens)	Document Length (Sentences)
Harry Potter IV - Der Slug Club	11	3	756.55	105	454
Harry Potter IV - Schleim	5	0	1671.80	-	463
Hänsel und Gretel	18	5	154.56	90.40	126
Die Begegnung	33	5	752.85	186.60	1967
Hochzeit wider Willen	48	12	574.62	292.33	2435
Bomben für Dortmund	44	2	786.77	253.50	2977
Im Bann der Vampire	24	1	982.83	91	1811
Aus guter Familie	145	74	352.64	458.11	5025
Effi Briest	172	55	587.08	314.02	5906
Der Turm der 1000 Schrecken	58	0	535.52	-	3016
Wechselhaft wie der April	70	2	426.61	148.50	2790
Lass Blumen sprechen	52	2	564.88	522	2414
Deus Ex Machina	43	0	723.56	-	2430
Der Sohn des Kometen	39	2	789.10	782.50	2234
Die Abrechnung	56	0	508.55	-	2352
Die Widows Connection	54	12	546.93	396.17	2900
Hetzjagd durch die Zeit	50	2	682.56	282	2256
Als der Meister starb	80	2	942.46	194.50	4492
Prophet der Apokalypse	64	1	484.05	85	2212
Ein sündiges Erbe	42	1	766.83	377	3048
Immer wenn der Sturm kommt	71	10	489.79	245.30	3179
Wir schaffen es - auch ohne Mann	46	7	696.67	70.71	3262
Tausend Pferde	103	1	435.60	30	3196
Widerstand zwecklos	31	6	773.19	316.67	1859
Bezaubernde neue Mutti	44	0	760.91	-	2904
Die hochmütigen Fellmann-Kinder	74	5	524.68	69.60	3048
Ein Weihnachtslied für Dr. Bergen	67	3	457.99	223.33	2485
Verschmäht	77	0	502.06	-	2781
Griseldis	68	7	551.29	395.71	2741
Krambambuli	13	7	256.85	176.14	206
Das Erdbeben in Chili	10	4	501	373.75	191
Die Braut des irischen Kriegers	17	4	740.82	284.25	904
Der Schimmelreiter	113	0	415.73	-	2769
Im Dschungel der Lust	14	1	815.14	200	938
Agenten und Spione	42	6	673.95	269	2226
In den Dreck getreten	42	4	825.38	458	2227
Die Verwandlung	20	8	835.05	699.38	718
Wenn Tote plötzlich wieder sprechen	61	9	613.11	169.33	3516
Die Judenbuche	30	18	420.87	415.94	1031
Der Geisterfelsen im Baikal-See	23	10	323.74	554.50	540
Nur noch eine heiße Nacht mit dir!	6	2	1833.50	142	885
Average	50.73	7.15	659.47	284.48	2266.20
Sum	2080	293	27038.10	9672.25	92914

Table 6: Train and test dataset split for texts.

Dataset	Name
STSS-Train	Der Turm der 1000 Schrecken Wechselhaft wie der April Lass Blumen sprechen Deus Ex Machina Der Sohn des Kometen Die Abrechnung Die Widows Connection Hetzjagd durch die Zeit Als der Meister starb Prophet der Apokalypse Ein sündiges Erbe Immer wenn der Sturm kommt Wir schaffen es - auch ohne Mann Tausend Pferde Widerstand zwecklos Bezaubernde neue Mutti Die hochmütigen Fellmann-Kinder Ein Weihnachtslied für Dr. Bergen Verschmäht Griseldis
Train-with-High	STSS-Train + Krambambuli Das Erdbeben in Chili Die Verwandlung Die Judenbuche Der Schimmelreiter
Train-Full	Train-with-High + Im Dschungel der Lust Agenten und Spione Wenn Tote plötzlich wieder sprechen Der Geisterfelsen im Baikal-See Die Braut des irischen Kriegers Nur noch eine heiße Nacht mit dir! In den Dreck getreten
STSS-Test-1	Bomben für Dortmund Die Begegnung Hochzeit wider Willen Im Bann der Vampire
STSS-Test-2	Aus guter Familie Effi Briest
OOD-Test	Harry Potter VI - Kapitel Der Slug Club Harry Potter VI - Kapitel Schleim Hänsel und Gretel
Test-Full	STSS-Test-1 + STSS-Test-2 + OOD-Test

A.9 Visual Error Analysis

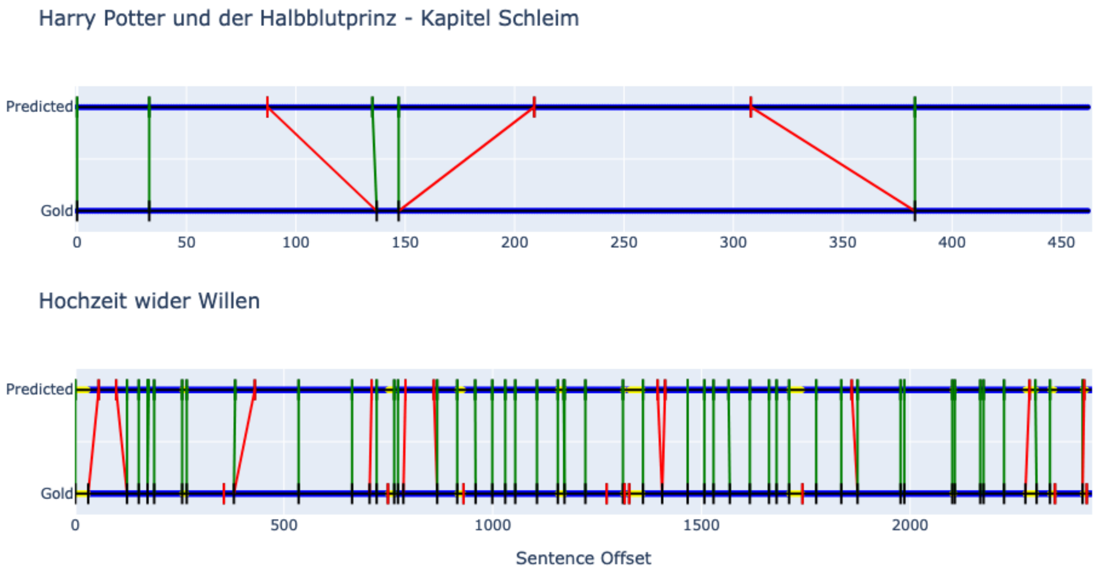

Figure 5: Visualisation of errors made by the best model on one of the annotated chapters from Harry Potter and the novel "Hochzeit wider Willen". The lower line shows scene borders annotated in the gold data, the upper line scene borders predicted by the model. For visualisation, the lines are blue for parts of the text that are labelled as scenes in the gold annotation and yellow for parts that are labelled as non-scenes (the predicted labels make no such distinction). Predicted borders are always mapped to the closest annotated gold border, as shown by the connecting lines. If a prediction is counted as correct (no more than three sentences from the closest gold annotated scene border), the connecting line is green (true positive), otherwise it is red (false positive). Undetected scene borders (false negatives) are shown in red on the lower line.

A.10 Llama Results

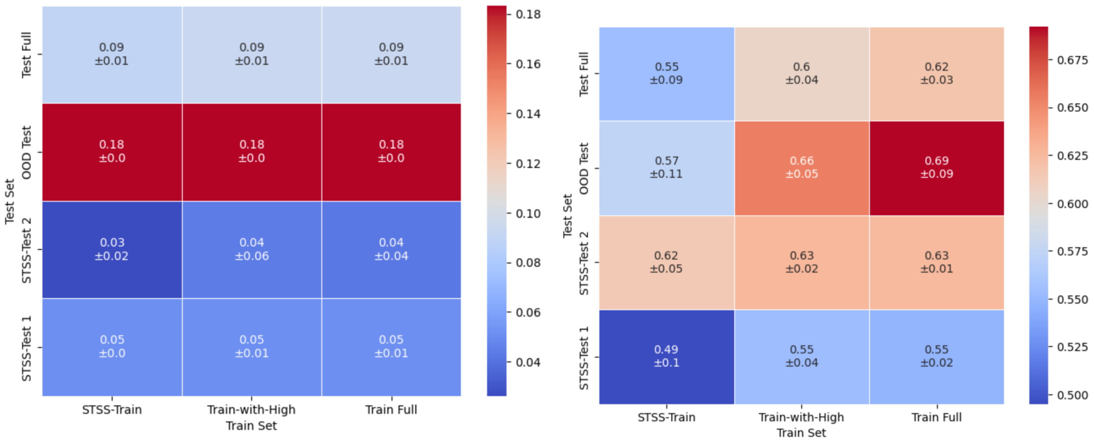

Figure 6: Llama-3-8b fine-tuned with the `No-CoT` (left) and `CoT-List` (right) strategy