



An Automatic Soundtracking System for Text-to-Speech Audiobooks

Zikai Chen, Lin Wu, Junjie Pan, Xiang Yin

Bytedance AI-Lab

{chenzikai, wulin.5050, panjunjie.jeff, yinxiang.stephen}@bytedance.com

Abstract

Background music (BGM) plays an essential role in audiobooks, which can enhance the immersive experience of audiences and help them better understand the story. However, well-designed BGM still requires human effort in the text-to-speech (TTS) audiobook production, which is quite time-consuming and costly. In this paper, we introduce an automatic soundtracking system for TTS-based audiobooks. The proposed system divides the soundtracking process into three tasks: plot partition, plot classification, and music selection. The experiments shows that both our plot partition module and plot classification module outperform baselines by a large margin. Furthermore, TTS-based audiobooks produced with our proposed automatic soundtracking system achieves comparable performance to that produced with the human soundtracking system. To our best of knowledge, this is the first work of automatic soundtracking system for audiobooks. Demos are available on <https://acst1223.github.io/interspeech2022/main>.

Index Terms: audiobook, soundtracking, plot partition, plot classification, text-to-speech

1. Introduction

With the great development of text-to-speech (TTS) technology in recent years, neural network-based TTS systems [1, 2, 3, 4] can generate high-quality and natural speech. As a result, TTS systems have been applied for audiobooks production due to high quality and productivity compared to traditional manual recordings. In addition, Pan et al. [5] first proposed a chapter-wise understanding system to extract speaker and emotion tags from texts, which can automatically generate multi-voice emotional audiobooks with a multi-speaker emotional TTS system [6].

Background music (BGM) can significantly improve the intelligibility of audiobooks from audiences' feedback in recent market research. Moreover, previous approaches on film scoring demonstrate that BGM can not only set moods and tonalities for the foreground media [7], but also expose the inner feelings and thoughts of characters to the audience and help them better get immersed into artworks [8]. To obtain well-designed BGM soundtracks for TTS-based audiobooks, the common method is manual soundtracking, which select BGM based on the sentiment of story plots. Specifically, a story plot means a continuous, sentimentally close segment that depicts a complete story event in the novel texts. However, manual soundtracking is quite time-consuming and costly, which is impractical in commercial TTS-based audiobooks production.

To address those issues, we propose an automatic soundtracking system for TTS-based audiobooks, which can automatically select BGM for audiobooks according to the sentiment of corresponding plots in texts within a few seconds. Firstly, a plot partition module is applied to split a novel chapter into several plots. Secondly, a plot classification module is employed to

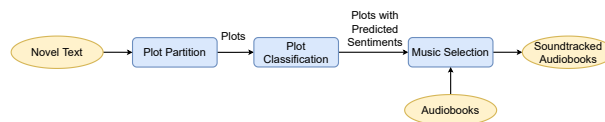


Figure 1: An overview of the automatic soundtracking system.

identify the sentiment of those plots. Lastly, appropriate BGM pieces are selected according to the sentiment of corresponding plots.

There has been considerable literature in text sentiment analysis, which aims to identify the subjectivities of text. The common task for text sentiment analysis is to analyze the polarity of text [9, 10], but in more complex tasks, there can be more than two classes [11]. Recent methods adopt some pre-trained networks to incorporate outside knowledge and finetune the network on particular sentiment classification tasks [12, 13, 14]. Our proposed system also includes sentiment analysis, which is more complicated. Different from previous approaches, we need to partition a novel chapter into several plots first, and then classify sentiment on the plot level.

To our best known, this is the first automatic soundtracking system for TTS-based audiobooks. The experiments shows that each module in our proposed system significantly outperform baselines. In addition, TTS-based audiobooks with our soundtracking system perform comparably to that with the human system.

2. System architecture

As shown in Figure 1, our proposed audiobook soundtracking system consists of three modules: plot partition, plot classification, and music selection. Specifically, the plot partition module is firstly applied to split chapter-level texts into several plots. And then the plot classification module is employed to predict sentiment tag for each plot. Finally, the music selection module is applied to automatically select pieces of music from a music library and mix audios together according to the sentiment tags and plot lengths.

2.1. Plot partition

The plot partition module aims to split chapter-level texts into continuous plots. In our system, a plot contains more than two paragraphs, and plot boundaries locate between paragraphs. Meanwhile, plot boundaries must meet one of the following conditions: 1). a main role first shows up or takes some actions; 2). the time or the location changes; 3). the atmosphere of events or the emotion of main roles changes. Paragraphs adjacent to plot boundaries are called the head and the tail of plots, respectively.

An example of plot partition is shown in Figure 2. We take plot partition as a sequence labeling task on the paragraph level,

where the head/tail paragraphs are assigned $[SEP]$ tags, and the others are assigned $[NON]$ tags.

2.1.1. BERT-RNN-CNN structure

As shown in Figure 3, the structure of our plot partition model is a 12-layer BERT [15], followed by a 512-unit Bidirectional GRU [16] (BiGRU) layer and a 1D-CNN [17] layer. The Chinese BERT model is pre-trained on massive Chinese novels with the Whole Word Masking [18] strategy and fine-tuned in the plot partition training process. The BiGRU is employed to capture the sequential dependency among paragraphs. The CNN layer is set with $kernel = 3$ and $stride = 1$, aiming to extract the correlation between adjacent paragraphs.

In addition, a similarity layer is applied to model the alternation between paragraphs before the output layer. Formally,

$$o'_{ij} = h_{ij} \oplus \text{FF}_s \left((\text{sim}(c_{ij}, c_{i,j-1}), \text{sim}(c_{ij}, c_{i,j+1}))^T \right) \quad (1)$$

$$o_{ij} = \text{FF}_f(o'_{ij}) \quad (2)$$

where the similarity function is the cosine similarity between two vectors. c_{ij} is the output from the 1D-CNN layer, \oplus is the concatenation of two vectors, and FF_s and FF_f are corresponding feed-forward layers.

2.1.2. Multi-task learning

Since the atmosphere alternation of events or the emotional state change of main roles can occur in the plot boundary, it is desirable that the representations of paragraphs also include such sentimental information. To achieve this, we set up two training objectives for the plot partition model. The first one is the partition objective. For each paragraph p_{ij} , if it is a $[SEP]$ paragraph, then its label $y_{ij} = 1$, otherwise $y_{ij} = 0$. Assume the output probability for y_{ij} is o_{ij} , then the objective aims at minimizing the negative log-likelihood

$$\mathcal{L}_{pp} = \sum_{i=1}^N \sum_{j=1}^{n_i} (-y_{ij} \log o_{ij} - (1 - y_{ij}) \log(1 - o_{ij})) \quad (3)$$

The second one is the classification objective. With the objective, the model learns to recognize the sentiment of each paragraph with the output hidden state h_{ij} of BiGRU:

$$p_{ij} = \text{softmax}(\text{FF}_c(h_{ij})) \quad (4)$$

$$\mathcal{L}_{pc} = - \sum_{i=1}^N \sum_{j=1}^{n_i} \sum_{c \in \mathcal{C}} l_{ijc} \log p_{ijc} \quad (5)$$

FF_c is a feed-forward layer. \mathcal{C} is the set of sentiment categories, which are the same as in Subsection 3.3. $l_{ijc} = 1$ if the sentiment label of the j -th paragraph in the i -th chapter is c , otherwise $l_{ijc} = 0$. p_{ijc} is the predicted probability for the same paragraph to be identified in the category c .

With multi-task learning, the loss function of the plot partition model is

$$\mathcal{L}_p = (1 - \lambda)\mathcal{L}_{pp} + \lambda\mathcal{L}_{pc} \quad (6)$$

Here λ is a hyperparameter for adjusting the weight of the two objectives.

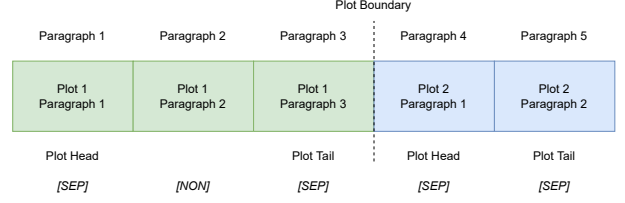


Figure 2: An example of plot partition, which consists of two plots. The plot border (dash line) is between Paragraph 3 and 4.

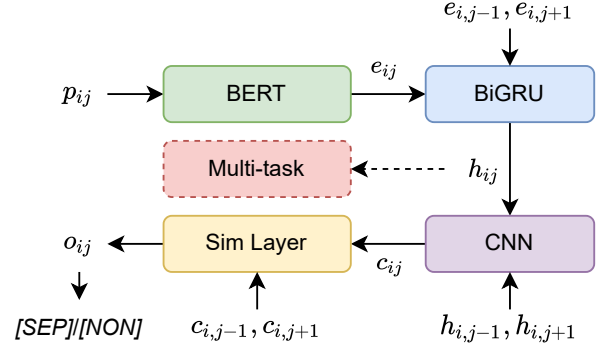


Figure 3: Structure of the plot partition model. The model learns to identify whether a paragraph is assigned the $[SEP]$ tag or $[NON]$ tag. p_{ij} is the j -th paragraph of the i -th chapter in the dataset. Its embedding e_{ij} is obtained by BERT. h_{ij} and c_{ij} are outputs from BiGRU and 1D-CNN, respectively. Finally, the output o_{ij} from the similarity layer is used for $[SEP]/[NON]$ identification.

During inference phase, only the $[SEP]/[NON]$ prediction is needed. We do not need the predicted sentiment because single paragraph contain little sentiment information, which can easily lead to wrong predictions. We leave the sentiment prediction to the plot classification module in Subsection 2.2, which utilizes all the paragraphs in the plot to recognize the corresponding sentiment.

2.1.3. Head-tail tactic

Intuitively, the plot partition module only needs to predict all the heads or the tails considering that a plot boundary can be identified by a head or a tail paragraph alone. However, our module is designed to predict both heads and tails. We call it the head-tail tactic. We argue that there is no bias towards the plot head or tail in the definition of the plot boundary, and only predicting one kind of them may confuse the model and downgrade the performance. Ablation experiments in Subsection 3.2 demonstrate the validity of this argument. In addition, a post-processing rule is applied to handle successive $[SEP]$ (≥ 3) situations. Only the last two $[SEP]$ paragraphs are regarded as the tail and the head, respectively, and the others are forced to $[NON]$.

2.2. Plot classification

The plot classification module aims to predict sentiment tags for each plot in the text. Normally a plot is composed of several

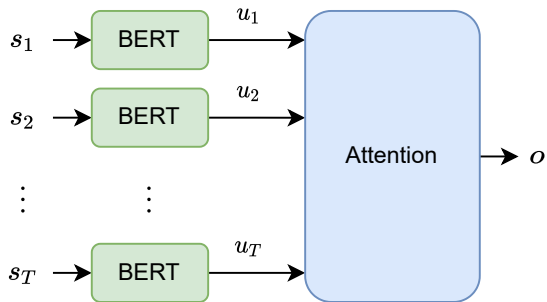


Figure 4: Structure of the plot classification model. The inputs are T segments (s_1, s_2, \dots, s_T) from a plot. The embedding u_i of segment s_i is obtained by BERT and sent into the self-attention part. The output o from the self-attention is used to predict the sentiment tag of the plot.

paragraphs. As a single paragraph contain very little contextual information, we use this module to recognize the plot sentiment from multiple longer text segments in a plot. Moreover, the self-attention mechanism [19] is introduced to focus on more important paragraphs in the input text segments.

The inputs are T segments $\{s_1, s_2, \dots, s_T\}$ from a plot, and u_i is the character-level 768-dimension embedding of s_i generated by a 12-layer BERT. The BERT is first pre-trained same as that used in plot partition, and fine-tuned in the training process. Subsequently, a modified multi-head self-attention layer is applied to the embedding u_i to focus on more important segments. The output layer is a feed-forward layer with softmax activation. Cross-entropy loss is used to predict 12 categories: *warmth, happiness, romance, highlight, threat, sadness, injury, misunderstanding, conflict, positive background, negative background, neutral event*.¹

The segments are decided empirically. We first treat the whole plot as the initial segment. Then we recursively split the longest unsplit segment into two halves until we have acquired more than 8 segments to ensure that all important segments have been covered. We name this tactic *Halving*. Other segmentation tactics are also explored in Subsection 3.3.

2.3. Music selection

After identifying the sentiment of each plot, we finally select BGM and mix it into audiobooks based on plot sentiment tags and plot lengths. We build a BGM library by using an internal music generation model to generate various pieces of music according to the sentiment categories and expected durations. A heuristic strategy (Algorithm 1) is adopted to soundtrack each plot based on its sentiment tag and text length. Selected music can be compressed, faded, trimmed, looped, silence-added to ensure the consistency of the whole soundtrack of a chapter. Specifically, to enhance the auditory experience, we leave extremely short plots that contain less than 80 characters unsoundtracked.

Algorithm 1 Music selection strategy

Input: Text of a plot p and the sentiment tag c of p

Output: Selected music for p

- 1: **if** p is a short plot (less than 200 characters) **then**
 - 2: Select a short piece of music that fits category c
 - 3: **else**
 - 4: **if** p contains more dialogue than narration **then**
 - 5: $q \leftarrow$ the dialogue part of p
 - 6: **else**
 - 7: $q \leftarrow$ the narration part of p
 - 8: **end if**
 - 9: **if** q is long (more than 480 characters) **then**
 - 10: Select and concatenate multiple pieces of music that fits category c
 - 11: **else**
 - 12: Select a piece of music that fits category c
 - 13: **end if**
 - 14: **end if**
 - 15: **return** the selected music
-

Table 1: Details of the dataset

	# of chapters	# of plots	# of paragraphs
Train	4,500	23,241	281,507
Test	500	2,577	32,253

3. Experiments and results

3.1. Dataset

As there is no open-source data, we sample 5,000 chapters (4,500 chapters for the training set and 500 chapters for the test set) of Chinese novels and conduct plot annotation on them. We train 8 people for the plot annotation. For each chapter, we randomly assign two of them to annotate the plots. If their results are different, we further ask a novel specialist to do the annotation and adopt the specialist’s result. Details of the dataset are listed in Table 1.

3.2. Objective evaluation of the plot partition model

For plot partition, we report the partition accuracy. As there is ambiguity on plot border position, it is actually very difficult for models to identify these borders exactly. Meanwhile, it is not required to predict the plot border perfectly to produce satisfying soundtracks. Therefore we define $\text{acc-}E$ ($0 \leq E \leq 3$) as our metric. $\text{acc-}E$ indicates the percentage of predicted plots with at most E paragraphs different from the corresponding gold labels. $\text{acc-}0$ is the standard accuracy. When E grows larger, more ambiguity is tolerated.

We also train a baseline to compare with our model. The baseline model utilizes a pre-trained BERT model with a 2-layer 768-unit dense on top of it to judge whether each paragraph is a plot head. No context information is provided.

The Ablation study is organized as follows:

- Ours w/o head-tail: only plot heads are regarded as [SEP] paragraphs.
- Ours w/o cos-sim: the cosine similarity layer is removed, which means $o'_{ij} = c_{ij}$.

¹Note that these are all translated versions because our work is conducted on Chinese novels, and these tags are also in Chinese.

- Ours w/o multi-task: λ in Equation 6 is set to 0 (The multi-task loss is removed).

Table 2: Objective evaluation result of the plot partition model

	Acc-0	Acc-1	Acc-2	Acc-3
Baseline	0.161	0.244	0.292	0.334
Ours	0.260	0.397	0.451	0.513
Ours w/o head-tail	0.263	0.365	0.428	0.484
Ours w/o cos-sim	0.263	0.395	0.448	0.511
Ours w/o multi-task	0.257	0.382	0.438	0.498

The result is in Table 2. Our model outperforms the baseline model with a significant margin, indicating that context information is critical in predicting the plot border.

In the ablation study, our model performs best on acc-1 to acc-3 and comparably on acc-0, only 0.3% lower than the best score. The head-tail tactic is the most critical of all three tactics, contributing to 3.2% and 2.9% gain on acc-1 and acc-3, respectively. This indicates that the plot tail is also very informative in predicting the plot border. The multi-task learning also contributes to 1.5% gain on acc-1 and acc-3, which reveals the importance of understanding the sentiment of each paragraph.

3.3. Objective evaluation of the plot classification model

For plot classification, we report the macro F1 score. For comparison, we also train a 2-layer 768-unit dense network as our baseline model. The inputs of the network are paragraph representations obtained with the 12-layer pre-trained BERT.

To justify the effectiveness of *Halving*, we also attempt other segmentation tactics. The details of each tactic are as follows.

- *Full*: 1 segment, the whole plot.
- *Same*: segments made up of the same number of paragraphs. By hyperparameter tuning, we use segments of 4 paragraphs to achieve the best result.

Table 3: Objective evaluation result of the plot classification model

Combination tactic name	Macro F1 score
Baseline	0.371
<i>Full</i>	0.518
<i>Same</i>	0.529
<i>Halving</i> (ours)	0.535

The result is in Table 3. Again, our model vastly outperforms the baseline, showing that it is far from enough with only paragraph representations. In addition, *Halving* tactic is proved more effective than other segmentation tactics.

3.4. End-to-end subjective evaluation

To assess the performance of the whole soundtracking system, we also conduct a manual evaluation. We sample 80 chapters of multi-role dubbed audiobooks synthesized by TTS and assess three systems with these chapters:

- *Baseline system*: baseline plot partition model + baseline plot classification model + heuristic music selection strategy
- *Model system*: proposed plot partition model + proposed plot classification model + heuristic music selection strategy
- *Human system*: groundtruth plot segmentations + groundtruth plot tags + heuristic music selection strategy

40 scorers are asked to rate each mixed audiobook chapter from each system on a scale of 1 (very poor) to 5 (excellent). A detailed explanation of each score level is explained in Table 4.

Table 4: Details of each score level

Score	Details
1	Very poor, greatly worsens listening experience
2	Poor, slightly conflicts with the story
3	Fair, some problems exist but overall acceptable
4	Good, not perfect but matches the text well
5	Excellent, greatly boosts listening experience

After the rating, the qualified rate (score ≥ 3) and excellent rate (score ≥ 4) are calculated respectively for each system.

Table 5: Manual evaluation result

System	Qualified rate (%)	Excellent rate (%)
Baseline	63.75	23.75
Model	88.75	45.00
Human	88.75	52.50

The result is shown in Table 5. Our model system greatly outperforms the baseline system, and on the qualified rate it even ties the human system. However, there is still a gap between our model system and the human system on the excellent rate, indicating that there exists room for improvement of our system. For example, our classification model is still not good enough at detecting the emotional intensity, which is strongly related to the excellent rate but not the qualified rate.

The qualified and excellent rates for the human system are not 100%. This is because we use the heuristic music selection strategy rather than complete manual soundtracking in the human system. It has to be admitted that there are still many deficiencies in our music selection strategy. Due to the lack of previous work about music selection and limited input features, it is unavoidable that the strategy cannot be perfect. We leave the improvement of the strategy as future work.

4. Conclusions

In this paper, we introduce an automatic audiobook soundtracking system for TTS audiobook production. The objective evaluation shows that our plot partition and plot classification sub-modules outperform baselines by a large margin. Furthermore, the subjective evaluation of the whole system shows that our automatic system can achieve comparable performance to the human system. In future work, we can improve the system by enhancing its capability in capturing emotional intensity and optimize the music selection strategy to realize finer control to the subtle emotional changes within a plot.

5. References

- [1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomvrgiannakis, R. A. J. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTER-SPEECH*, 2017.
- [2] J. Shen, R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 04 2018, pp. 4779–4783.
- [3] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *NeurIPS*, 2019.
- [4] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *ArXiv*, vol. abs/2006.04558, 2021.
- [5] J. Pan, L. Wu, X. Yin, P. Wu, C. Xu, and Z. Ma, "A chapter-wise understanding system for text-to-speech in chinese novels," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6069–6073.
- [6] P. Wu, J. Pan, C. Xu, J. Zhang, L. Wu, X. Yin, and Z. Ma, "Cross-speaker emotion transfer based on speaker condition layer normalization and semi-supervised training in text-to-speech," 10 2021.
- [7] C. Gorbman, "Narrative film music," *Yale French Studies*, no. 60, pp. 183–203, 1980. [Online]. Available: <http://www.jstor.org/stable/2930011>
- [8] JESSICA GREEN, "Understanding the Score: Film Music Communicating to and Influencing the Audience," *The Journal of Aesthetic Education*, vol. 44, no. 4, pp. 81–94, 2010, publisher: University of Illinois Press. [Online]. Available: <http://www.jstor.org/stable/10.5406/jaesteduc.44.4.0081>
- [9] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: <https://aclanthology.org/D13-1170>
- [10] P. Yanardag and S. V. N. Vishwanathan, "Deep graph kernels," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.
- [11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," 2019.
- [12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.
- [13] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *NeurIPS*, 2019.
- [14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>
- [15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [18] Y. Cui, W. Che, T. Liu, B. Qin, and Z. Yang, "Pre-training with whole word masking for chinese bert," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, p. 3504–3514, 2021. [Online]. Available: <http://dx.doi.org/10.1109/TASLP.2021.3124365>
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.