Interspeech 2022
18-22 September 2022, Incheon, Korea

DocLayoutTTS: Dataset and Baselines for Layout-informed Document-level Neural Speech Synthesis

Puneet Mathur[*1], *Franck Dernoncourt*[2], *Quan Hung Tran*[2], *Jiuxiang Gu*[2], *Ani Nenkova*[2], *Vlad Morariu*[2], *Rajiv Jain*[2], *Dinesh Manocha*[1]

[1]University of Maryland, College Park
[2]Adobe Research
[*]puneetm@umd.edu

Abstract

We propose a new task of synthesizing speech directly from semi-structured documents where the extracted text tokens from OCR systems may not be in the correct reading order due to the complex document layout. We refer to this task as layout-informed document-level TTS and present the DocSpeech dataset which consists of 10K audio clips of a single-speaker reading layout-enriched Word document. For each document, we provide the natural reading order of text tokens, its corresponding bounding boxes, and the audio clips synthesized in the correct reading order. We also introduce DocLayoutTTS, a Transformer encoder-decoder architecture that generates speech in an end-to-end manner given a document image with OCR extracted text. Our architecture simultaneously learns text reordering and mel-spectrogram prediction in a multi-task setup. Moreover, we take advantage of curriculum learning to progressively learn longer, more challenging document-level text utilizing both DocSpeech and LJSpeech datasets. Our empirical results show that the underlying task is challenging. Our proposed architecture performs slightly better than competitive baseline TTS models with a pre-trained model providing reading order priors. We release samples of the DocSpeech dataset[1].
Index Terms: document-level speech synthesis, text-to-speech, reading order detection, curriculum learning.

1. Introduction

Text-to-speech (TTS) is an important task in speech language processing to enable human-machine interaction that is intelligible and indistinguishable from human speech. Prior works in neural TTS have achieved near human-level speech synthesis ability using recent attention-based autoregressive methods such as Tacotron 2 [1] and Transformer-based end-to-end speech synthesis models (eg. Transformer-TTS) [2]. However, synthesizing coherent speech from text in documents remains a challeng problem due to two reasons: (1) long sequence length of input text; (2) lack of correct reading order provided by off-the-shelf Optical Character Recognition (OCR) engines that tend to arrange all recognized tokens in a top-to-bottom and left-to-right manner, and disregard the layout of the long-form text [3].

The capability to perform long-context speech synthesis is needed for several tasks such as singing voice synthesis (SVS) systems [4], document readers and screen reading systems [5], reading out audio-based online content (e.g. news articles, audiobooks, and podcasts), and conversational speech generation [6]. Typically, the raw input text comprising of large coherent speech units such as paragraphs is pre-processed into utterances (e.g., sentences) and are treated independently from each other, thus discarding the original ordering. Moreover, directly concatenating parts of independently synthesized audio units can make it unnatural due to various reasons. These include lack of spontaneous prosodic phenomena like filled pauses, prolongations, voice modulation, variations in fundamental frequency through time, or the speech rate peaking around the middle of a larger unit [7]. Prosodic variation is governed by context at different levels, and contextual information is expressed through prosody [8].

Current TTS systems assume that the reading order sequence of input text tokens is correct. However, the reading order itself depends on the structure of the document and is not known apriori. In fact, current OCR systems cannot infer this correctly from complex spatial documents. Figure 1 shows how top-down, left-right linearizing text tokens based on bounding box coordinates is not optimal for certain document types, such as multi-column templates, tables, forms, and invoices, where text maybe structured spatially in a layout. Synthesizing speech from a scrambled sequence of text tokens may result in unacceptable results, thereby deteriorating the quality of human computer interactions and making documents inaccessible for people with visual disabilities [5]. Recently, some deep learning based methods have been proposed to perform reading order detection [9]. However, pre-processing raw text sequences discards most orthographic knowledge of the structure of the original text data, making it harder to accurately model prosodic variations in speech. Additionally, mere reordering of text tokens followed by synthesis of isolated short sentences distorts the natural *phrasing* [10] of text, and may discard the larger context and structure of the long-form text.

Main Contributions: We propose the task of document-level layout-informed text-to-speech synthesis that aims to generate human-level speech corresponding to the correct reading order of the text present in a semi-structured document. Furthermore, we release DocSpeech, a benchmark dataset of 10K speech samples corresponding to Word documents with a wide variety of semi-structured layouts.

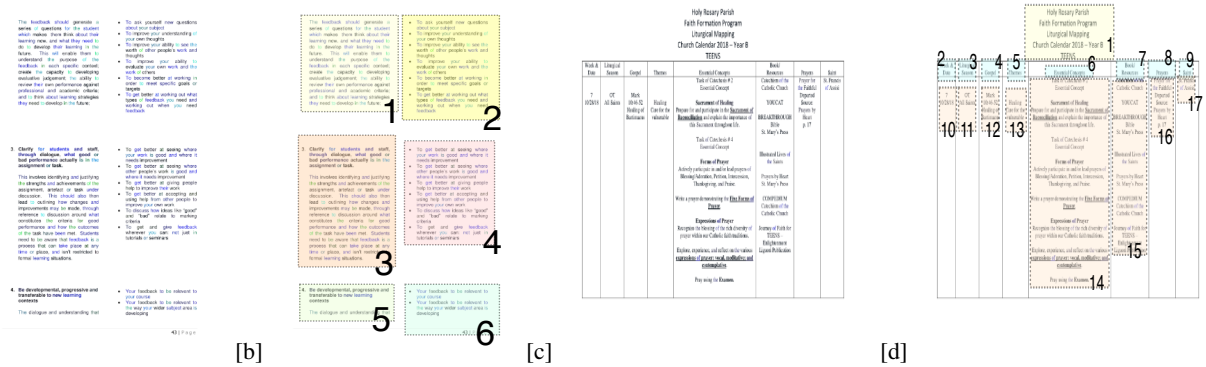
The process of synthesizing layout-informed speech for semi-structured documents can benefit from solving the reading order unscrambling and long-form audio generation tasks simultaneously. Therefore, we present a strong neural model, DocLayoutTTS, to jointly model text reading order detection as well as speech mel-spectrogram synthesis using Transformer encoder-decoder. Using curriculum learning [11], our DocLayoutTTS model demonstrates competitive performance on the task of layout-informed document-level TTS. Some novel aspects of our work include:

1. We propose a new task for layout-informed documet-level TTS to generate speech from text present in semistrcutured documents. We curate a public dataset, DocSpeech, which consists of 10K audio clips of a single speaker reading documents with complex layouts. We provide OCR reading order as well as unscrambled text transcription for each clip. The resulting audio clips have a total length of approximately 830 hours, with average clip duration of 5 minutes, compared to 10 seconds in LJSpeech dataset.
2. We present DocLayoutTTS, a neural baseline archi-

[1]https://github.com/doclayouttts/DocLayoutTTS-Dataset

Copyright © 2022 ISCA

10.21437/Interspeech.2022-574



[a] [b] [c] [d]

Figure 1: *Document image examples (a,c) in* `DocSpeech` *with the reading order information (b,d). The colored areas show the paragraph ordered as per correct reading order. Reading order detection is important to align the input sequence to TTS in the correct order. Text lying in the same spatial unit needs to be read together for proper context.*

tecture that simultaneously learns text reordering, newline prediction and mel-spectrogram prediction for synthesizing speech from documents in our proposed dataset in a multi-task fashion.

3. Our proposed model uses curriculum learning to learn increasingly long document-level speech synthesis, starting with short speech utterances from LJSpeech dataset. We compare the performance of `DocLayoutTTS` with other strong baselines and find that `DocSpeech` is a challenging dataset for the proposed task.

2. Related Work

Long-form document-level TTS: Recurrent neural network models such as Tacotron [12] use attention-mechanism to align the target text and output a spectrogram. On the other hand, Tacotron 2 [1] system uses location-sensitive attention to extend the alignment between the encoder and decoder to the information of the previous time step. However, they are still limited to synthesizing few sentences of text into speech due to constraints on long-range input sequences. Transformer-based end-to-end text-to-speech synthesis models such as Transformer-TTS [2] use multi-headed attention to solve the long range dependency problem. However, as the sequence length of the input increases, the computational complexity of training the Transformer model rises quadratically. [13] used attention-masking along with curriculum learning to extend the maximum synthesis length to 5 minutes. However, most of these prior works are limited by the fact that they rely on well-formed phoneme sequences as input to generate the mel-spectrograms. We hypothesize that a multi-task objective that simultaneously learns reading order detection, newline prediction, and mel-spectrogram generation can help exploit latent layout signals for recovering the correct reading order, thereby preserving the natural prosody required for human-like document-level speech.

Reading Order Detection in Text: Several previous studies have explored reading order prediction in text. [14] was one of the earliest works to propose a rule-based learning method for identifying reading order sequence in text. [15] applied domain knowledge to determine the reading order relationship between logical document components. More recently, deep learning based methods have been widely used for this task. [16] used visual layout features encoded through a graph neural network to reorganize OCR text into a proper sequence. Most recently, [9] provided an seq2seq model for text reordering based on semantic, visual and spatial signals. In contrast, our work is the first attempt in terms of studying the necessity of reading order detection for text-to-speech tasks.

Statistics	LJSpeech	DocSpeech
Total Clips	13,100	10,000
Total Words	225K	1800K
Total Characters	1308K	10500K
Total Duration	24 hr	830 hr
Average Clip Duration	6.57 sec	5 min
Min Clip Duration	1.11 sec	1.05 min
Max Clip Duration	10.10 sec	10.2 min
Mean Words per Clip	17.23	156

Table 1: *Dataset statistics comparing the proposed* `DocSpeech` *with LJSpeech dataset.* `DocSpeech` *has fewer total speech samples but significantly larger total and mean clip duration.*

3. `DocSpeech` Dataset

We create `DocSpeech`, a synthetic dataset by re-purposing the open-source ReadingBank dataset [9] which provides semi-structured Word documents with the reading sequence of words as extracted from DocX files, correct reading order sequence based on structured layout, as well as corresponding bounding boxes for each text token extracted from the PDF versions of the DocX files. We sample a subset of 10K documents from ReadingBank such that each file has more than 50 words.

We used the Gentle Forced Aligner[2], a Kaldi[3]-based audio alignment tool to perform forced alignment of words and audio snippets on LJSpeech[4] dataset. We obtain the word-level audio alignment from the generated time-marked conversation file which is used to construct an audio mapping of each unique word with its corresponding mel-spectrogram. We combine mel-spectrograms corresponding to each token in the correct reading order of the document text file. However, if simply joined, an unnatural voice may be generated due to the audio concatenation step. To prevent this, we insert an empty m-token (mel spectrogram token) between each consecutive word-level mel-spectrogram. The m-token, allows the speech to pause naturally between consecutive mel-spectrograms, giving the effect of naturally linked words. `DocSpeech` contains 10,000 document-speech pairs with an average clip duration of 5 minutes, out of which 100 files are used for testing and remaining for training. Table 1 summarizes the dataset statistics about the

[2]https://lowerquality.com/gentle/
[3]https://kaldi-asr.org/
[4]https://keithito.com/LJ-Speech-Dataset/

DocSpeech dataset.

4. Our Approach

In this section, we describe the problem statement, individual components of DocLayoutTTS model as illustrated in Figure 2, and training paradigm for optimizing the model.

4.1. Problem Formulation

We formally define the document-level layout-informed text-to-speech task. Given a semi-structured document $\mathcal{D}$ with words w_i acquired through an OCR along with their corresponding bounding box coordinates (x_1, y_1, x_2, y_2) (where (x_1, y_1) and (x_2, y_2) are the top-left and bottom-right coordinates, respectively), we aim to synthesize speech mel-spectrogram $\mathcal{S}$ such that the constituent words are sorted into their correct reading order in the speech output. We derive the ground truth reading order from the embedded XML metadata of Word documents. Further, the WORD documents are converted into the PDF format to extract the 2D bounding box of each word using Google Tesseract[5].

4.2. Textual Layout Encoder

Inspired by Transformer-TTS architecture [2], we include a text-to-phoneme converter to learn the mapping between different regularities between the text syllabi and phonemes. Each incoming phoneme is passed through an encoder prenet to embed the phoneme input into a trainable embedding of 512 dimensions, followed by a batch normalization, ReLU activation, and a dropout layer. We add positional encoding (PE) [17] scaled by a factor of α to the processed phoneme input to take advantage of the relative token sequence of input.

Additionally, we add four 2D-positional encoding ($PE^{2D}_{x_0}$, $PE^{2D}_{x_1}$, $PE^{2D}_{y_0}$, $PE^{2D}_{y_1}$) to the phoneme input for learning the relative spatial position in a document. The four 2D-positional embedding layers correspond to the upper (y_0), lower (y_1), left (x_0) and right (x_1) coordinate directions, respectively. Each input phoneme h_i being fed to the encoder is represented by the following Equation:

$$h_i = prenet(phoneme_i) + \alpha * PE(i) + \beta * (PE^{2D}_{x_0}(i) + PE^{2D}_{x_1}(i) + PE^{2D}_{y_0}(i) + PE^{2D}_{y_1}(i))$$

4.3. Decoder

Reading Order Sequence Decoder: In the sequence decoding stage, the source and target are reordered sequences. We constrain the target sequence prediction to be the correctly ordered indices in the source sequence. Additionally, we also predict if a particular input position indicates start of a newline in the text document. Specifically, we perform a binary classification at each decoding step to check if the token denotes the end of reading order line due to layout constraints or end of page width.
Melspectrogram Decoder: Similar to TransformerTTS [2], we use a Transformer decoder using multi-head attention to integrate the encoder hidden states in multiple perspectives. We experiment with a larger embedding space of $d = \{1024, 2048, 4096\}$ compared to 512 embedding space of Transformer TTS to better model long-range context vectors. We employ a WaveNet vocoder to synthesize audios from the generated mel-spectrograms.

4.4. Multi-task Training

We use mean absolute error (MAE) to predict the mel-spectrogram. Reordered sequence index classification uses categorical cross-entropy loss, while newline prediction uses a weighted binary cross-entropy loss to adjust for class imbalance. All three tasks are correlated and reinforce each other, so we use multi-task training to optimize them simultaneously. The final optimization uses a weighted sum of the link prediction loss and element classification loss where the weighting factors λ and γ are hyperparameters as shown in the follwoing Equation:

$$\mathcal{L}_{total} = \lambda \mathcal{L}_{mel} + \gamma \mathcal{L}_{reorder} + (1 - \lambda - \gamma)\mathcal{L}_{newline}$$

[5]https://github.com/tesseract-ocr/tesseract

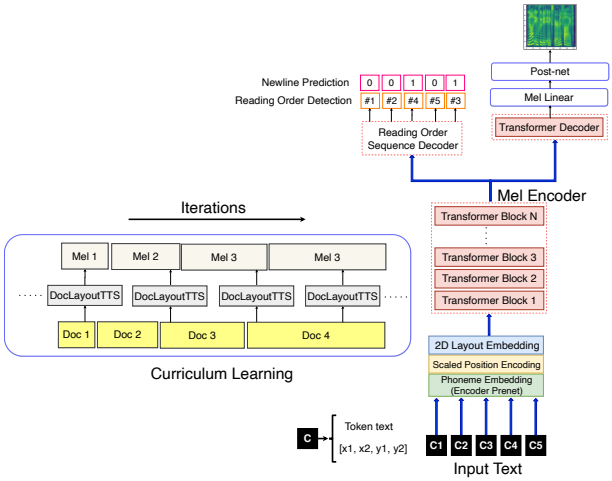


Figure 2: *DocLayoutTTS model takes a sequence of text tokens as input along with their bounding box coordinates. Encoder Prenet converts the input into a sequence of phonemes which are passed through a phoneme embedding. Scaled Position encoding and Layout encoding are added to the input and passed through the Transformer encoder. Mel decoder predicts the mel spectograms while the reading order sequence decoder predicts the indices corresponding to each word. We also predict the newline which denotes a break in the left-to-right traversal of reading order sequence in the document. Curriculum learning feeds increasingly long text sequences as input to gradually train the model with more difficult input samples.*

4.5. Curriculum Learning

Inspired by [13], we utilize curriculum learning [18] to improve the training process for document-level TTS. Curriculum learning is a deep learning training process where the difficulty of learning becomes gradually more complex. We apply curriculum learning to train the encoder-decoder network to help generate longer sequence inputs without losing long-range context. We utilize LJSpeech and our proposed DocSpeech to achieve an increasing difficulty curriculum in terms of the input sequence length. We initially start with sentence level input of LJSpeech. In the subsequent iterations, we input document text with increasing lengths, until all text input sequences have been exhausted. In order to fit the data in the limited GPU capacity, the model was set to automatically reduce the batch size to 1/2 whenever the GPU capacity limit was reached. This process continued untill the batch size reduced to 1 and could not go down further.

5. Experiments

In this section, we detail our experiments to test the proposed DocLayoutTTS model with the DocSpeech dataset. We compare our method with strong baselines and evaluate the synthesized audio quality in terms of MOS score.

5.1. Baselines

We compare DocLayoutTTS with two state-of-the-art pre-trained TTS models - Tacotron 2 [1] and TransformerTTS [2]. The input to these models is the original sequence of text extracted by the OCR. LayoutReader [9] is a state-of-the-art reading order detection model. We also test using a seperate model for text reordering and using the reordered output as input to TTS models. In this direction, we use a pre-trained LayoutReader for reading order detcetion, followed by Tacotron 2 and TransformerTTS for TTS. This provides an opportunity to analyze the effects of decoupling the reading order detection process from speech synthesis. We also compare our method to DLTTS, an RNN-based document-level TTS model.

5.2. Evaluation

We select 100 testing examples from the DocSpeech dataset. Each test sample consists of the document image, extracted text tokens along with their bounding boxes, and ground truth speech output. We evaluate mean option score (MOS) on these 100 documents generated by different models. We randomize the speech samples from baselines, our model, and the ground truth, followed by equal sized sampling to ensure that the expert testers don't know the sources of speech files. MOS is performed by 5 fluent English speakers who are experts in the domain of audio processing. MOS evaluation involves testers rating the quality of audio on a scale from 1 to 5 with 0.5 point increments. We also compare the mel-spectrograms generated by our model with baseline models to visualize the effect of our contributions to the quality of generated audio.

5.3. Training Details

We used Pytorch framework for deep learning models. We used two Nvidia Tesla P100 GPU's to train the models. We enabled multi-GPU training to enlarge the batch size. In order to accomodate large input sequences, we used dynamic batch sizes to maximize GPU utilization. DocLayoutTTS inputs phoneme sequence. Hence, the input text was pre-processed to get the phoneme sequences by following sentence separation, text normalization, word segmentation and pronunciation. Similar to TransformerTTS [2], we use a WaveNet vocoder conditioned on mel-spectrograms and trained it simultaneously using teacher forcing with a sample ground truth rate of 16000 and frame rate of ground truth mel-spectrogram equal to 80. Hyperparameters $\alpha, \beta, \gamma, \lambda$ were sampled between 0 and 1, with equal intervals of 0.1. We use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-6}$ and a learning rate of 10^{-3}.

6. Results and Analysis

Comparison with Baselines: Table 2 shows the comparison of our proposed DocLayoutTTS with baseline models. Tacotron 2 and TransformerTTS perform poorly on the DocSpeech dataset due to their inability to handle text sequences longer than few sentences as well as lack of layout information for token reordering. Hence, they suffer from missing words, repetitions and the output speech is unintelligible. This is also evident in the mel-spectogram visualizations where parts of the generated spectrogram is destroyed at multiple locations indicating failure in decoding step. Using pre-trained LayoutReader [9] to unscramble the reading order and feeding that as input to TTS models performs significantly better. However, the long-form text still hurts the performance despite being fed the reordered input. Additionally, we experiment with DLTTS model, which claims to be able to synthesize speech sequences upto 5 minutes. LayoutReader + DLTTS forms the strongest baseline with a remarkable improvement in MOS score compared to others. However, it still suffers from error propagation from decoupled reading order prediction and mel-spectogram prediction. We observe that DocLayoutTTS outperforms other strong baseline models. This improvement can be attributed to its ability to encode layout information in an end-to-end fashion as well as the document-level training using curriculum learning. However, it can also be observed that the outperformance of our architecture is not quite large in magnitude, and significantly falls short compared to the ground truth recordings. We attribute this to the inherent difficulty of that the task. Further research in more sophisticated models may help improve the performance on DocSpeech.

System	MOS
Tacotron2 [1]	1.75 $\pm$0.04
TransformerTTS [2]	1.82 $\pm$0.02
LayoutReader [9] + Tacotron2 [1]	2.05 $\pm$0.08
LayoutReader [9] + TransformerTTS [2]	2.08 $\pm$0.03
LayoutReader [9] + DLTTS [13]	2.25 $\pm$0.02
DocLayoutTTS (Ours)	**2.32 $\pm$0.02**
GroundTruth	4.65 $\pm$0.08

Table 2: ***Quantitative Results:*** *Comparison of MOS score with baseline models on DocSpeech dataset. Our proposed model improves MOS score by* 0.07 *over the layoutReader + DLTTS.*

System	MOS
DocLayoutTTS (Ours)	**2.32 $\pm$0.07**
w/o newline prediction loss	2.15 $\pm$0.05
w/o curriculum learning	2.04 $\pm$0.06
w/o reorder detection loss	1.79 $\pm$0.04

Table 3: ***Ablation Results:*** *Comparison of MOS score with ablation models on DocSpeech dataset. We perform ablation experiments to show the usefulness of different components of DocLayoutTTS model highlighted in red .*

Ablation Results: We perform a detailed study the contributions of each component in our proposed architecture to attribute the source of improvements. We experiment by training the model by removing the newline prediction loss and reorder detection loss, one at a time. We observe that text reorder detection is crucial for our model to successfully produce coherent speech clips with correct word order. Newline prediction loss helps the model to take advantage of the geometric information provided by visually-rich documents. Without the newline prediction loss, the model struggles to generate speech samples with naturalness in pauses and voice modulations. Finally, the addition of curriculum learning helps the model to smoothly extend the hidden space to learn long-range text sequences, preventing catastrophic forgetting at the decoding step.

7. Conclusion and Future Work

We present a new dataset titled as DocSpeech for the task of synthesizing speech directly from semi-structured documents where the text tokens may not be present in correct reading order. We also present DocLayoutTTS, a strong Transformer-based TTS model that can simultaneously learn to predict reading order of the document-level text and synthesize speech corresponding to the same. Further, our approach uses curriculum learning to extend the self-attention based audio alignment to long-form document-level input sequences. Although experiments on the proposed dataset display the effectiveness of our contributions, the task is challenging due to the large gap in speech quality between the ground truth recording and model generated audio. In future, we aim to explore non-autoregressive solutions that are not impacted by sequential inference nature of the current approaches so as to scale well to document-level input text lengths.

8. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[3] C. Clausner, S. Pletschacher, and A. Antonacopoulos, "The significance of reading order in document recognition and its evaluation," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 688–692.

[4] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2803–2815, 2021.

[5] D. Pradhan, T. Rajput, A. J. Rajkumar, J. Lazar, R. Jain, V. I. Morariu, and V. Manjunatha, "Development and evaluation of a tool for assisting content creators in making pdf files more accessible," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 15, no. 1, pp. 1–52, 2022.

[6] J. Cong, S. Yang, N. Hu, G. Li, L. Xie, and D. Su, "Controllable context-aware conversational speech synthesis," *arXiv preprint arXiv:2106.10828*, 2021.

[7] J. Cambre, J. Colnago, J. Maddock, J. Tsai, and J. Kaye, "Choice of voices: A large-scale evaluation of text-to-speech voice quality for long-form content," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[8] J. Cole, "Prosody in context: a review," *Language, Cognition and Neuroscience*, vol. 30, no. 1-2, pp. 1–31, 2015.

[9] Z. Wang, Y. Xu, L. Cui, J. Shang, and F. Wei, "Layoutreader: Pre-training of text and layout for reading order detection," *arXiv preprint arXiv:2108.11591*, 2021.

[10] V. Klimkov, A. Nadolski, A. Moinet, B. Putrycz, R. Barra-Chicote, T. Merritt, and T. Drugman, "Phrase break prediction for long-form reading tts: Exploiting text structure information." in *Interspeech*, 2017, pp. 1064–1068.

[11] Y. Kong, L. Liu, J. Wang, and D. Tao, "Adaptive curriculum learning," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5047–5056, 2021.

[12] 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. Agiomyrgiannakis, R. A. J. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH*, 2017.

[13] S.-W. Hwang and J.-H. Chang, "Document-level neural tts using curriculum learning and attention masking," *IEEE Access*, vol. 9, pp. 8954–8960, 2021.

[14] M. Aiello, A. Smeulders *et al.*, *Bidimensional relations for reading order detection*. University of Groningen, Johann Bernoulli Institute for Mathematics and . . . , 2003.

[15] D. Malerba, M. Ceci, and M. Berardi, "Machine learning for reading order detection in document image understanding," in *Machine Learning in Document Analysis and Recognition*, 2008.

[16] L. Li, F. Gao, J. Bu, Y. Wang, Z. Yu, and Q. Zheng, "An end-to-end ocr text re-organization sequence learning for rich-text detail image comprehension," in *ECCV*, 2020.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML '09*, 2009.