Seg2Act: Global Context-aware Action Generation for Document Logical Structuring

Zichao Li[1,2,*], **Shaojie He**[1,2,*], **Meng Liao**[3,†], **Xuanang Chen**[1],
Yaojie Lu[1,†], **Hongyu Lin**[1], **Yanxiong Lu**[3], **Xianpei Han**[1], **Le Sun**[1]
[1]Chinese Information Processing Laboratory, Institute of Software,
Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Search Team, WeChat, Tencent Inc., China
{lizichao2022,heshaojie2020,chenxuanang,luyaojie}@iscas.ac.cn
{hongyu,xianpei,sunle}@iscas.ac.cn {maricoliao, alanlu}@tencent.com

Abstract

Document logical structuring aims to extract the underlying hierarchical structure of documents, which is crucial for document intelligence. Traditional approaches often fall short in handling the complexity and the variability of lengthy documents. To address these issues, we introduce Seg2Act, an end-to-end, generation-based method for document logical structuring, revisiting logical structure extraction as an action generation task. Specifically, given the text segments of a document, Seg2Act iteratively generates the action sequence via a global context-aware generative model, and simultaneously updates its global context and current logical structure based on the generated actions. Experiments on ChCatExt and HierDoc datasets demonstrate the superior performance of Seg2Act in both supervised and transfer learning settings[1].

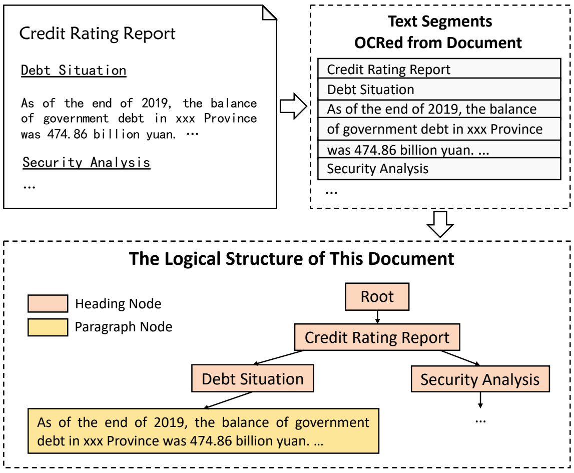

Credit Rating Report
Debt Situation
As of the end of 2019, the balance of government debt in xxx Province was 474.86 billion yuan. ···
Security Analysis
...
Text Segments OCRed from Document
Credit Rating Report
Debt Situation
As of the end of 2019, the balance
of government debt in xxx Province
was 474.86 billion yuan. ...
Security Analysis
...
The Logical Structure of This Document
Heading Node
Paragraph Node
Root
Credit Rating Report
Debt Situation
Security Analysis
As of the end of 2019, the balance of government debt in xxx Province was 474.86 billion yuan. ...
...

Figure 1: The illustration of document logical structuring task, which aims to transform text segments into a hierarchical tree structure containing the document's headings and paragraphs.

1 Introduction

Document logical structuring is an essential task for document understanding, which aims to extract the underlying logical structure of documents (Tsujimoto and Asada, 1990; Summers, 1998; Mao et al., 2003; Luong et al., 2010; Pembe and Güngör, 2015; Gopinath et al., 2018; Maarouf et al., 2021). As shown in Figure 1, document logical structuring transforms a document into a hierarchical logical tree composing of headings and paragraphs. Understanding a document's logical structure will benefit numerous downstream tasks, such as information retrieval (Liu et al., 2021), abstractive summarization (Qiu and Cohen, 2022), and assisting large language models in question answering over long structured documents (Saad-Falcon et al., 2023).

Document logical structuring is challenging due to the complexity of text segment dependencies in documents and the diversity of logical structures across various documents. Firstly, real-world documents are mostly multi-page, lengthy and with complex structures, while OCR tools often break content into short and incomplete lines rather than complete paragraphs. Such inconsistency between text segments and hierarchical structure poses a significant challenge to tracking and formulating text semantics and long-range dependencies. Secondly, due to the diversity of logical structures in various documents (e.g., financial report and scientific literature), it is very difficult to design a unified approach with strong generalization abilities, i.e., it can solve different types of documents.

Currently, most document logical structuring methods first decompose the extraction of logical structure into multiple separated subtasks (mostly including feature extraction, heading detection and nodes relationship prediction), then compose the components of different subtasks in a pipeline to predict the final document logical structure (Rahman and Finin, 2017; Bentabet et al., 2019; Hu

[*] Equal contribution.
[†] Corresponding author.
[1] The publicly available code are accessible at https://github.com/cascip/seg2act.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 18077–18088
November 12-16, 2024 ©2024 Association for Computational Linguistics

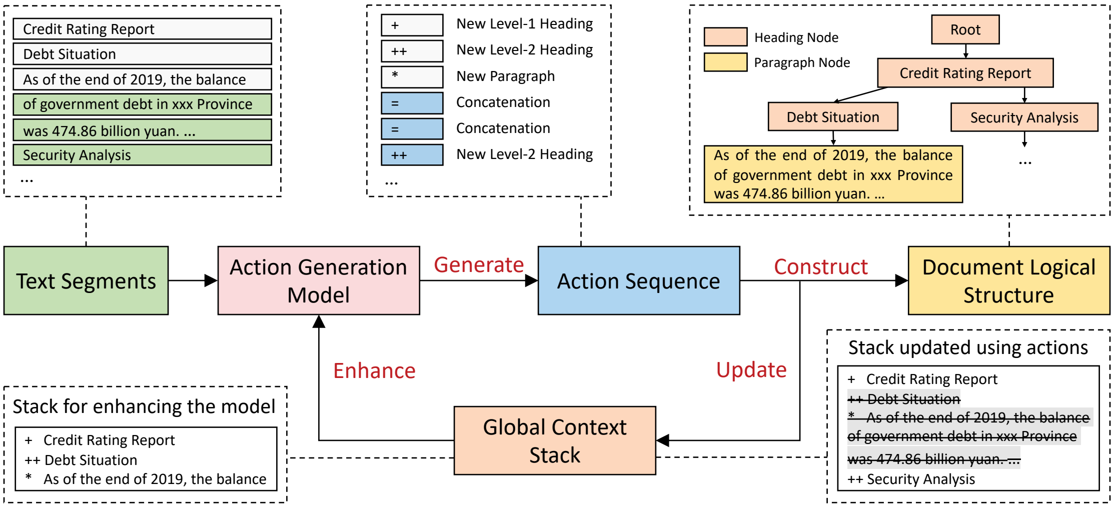


Figure 2: A generation step of SEG2ACT. The action generation model converts current text segments into actions to incrementally construct the document logical structure. A global context stack is maintained to enhance the model's global awareness, while the generated actions then being employed to update the stack.

et al., 2022b; Wang et al., 2023). The main drawbacks of these methods are: 1) By encoding fragmented text segments independently, these methods cannot capture the global information of documents and often result in semantic loss. 2) By pairwise predicting the relationship between text segments, these methods often ignore the long-range dependencies and result in sub-optimal structures. 3) The pipeline framework suffers from the error propagation problem. Due to the varieties of document structures, it is very challenging to design the optimal composition architecture manually for different types of documents.

To address these issues, in this paper, we propose SEG2ACT, a global context-aware action generation approach for document logical structuring. As illustrated in Figure 2, instead of decomposing the extraction of logical structure into subtasks, we revisit structure extraction as an action generation task. Specifically, sequentially feeding a document's text segments, a global context-aware generative model is employed to generate a sequence of actions for document logical structuring. We propose three types of actions, each corresponding to an operation that maps text segments to the logical structure, applicable across various types of documents. Furthermore, during the structuring process, SEG2ACT maintains a global context stack which selectively stores crucial parts of global document information, expressing long-range dependencies in a concentrated manner. In this way, SEG2ACT can effectively handle various document types, generate the logical structure of a document in an end-to-end manner, and leverage global document information for text segment encoding and structure generation. Experiments on ChCatExt and HierDoc datasets demonstrate that SEG2ACT achieves superior performance in both supervised and transfer settings, verifying the effectiveness and the generalization ability of the proposed method.

Our contributions are summarized as follows: 1) This is the first work to make the logical structure extraction as an one-pass action generation task, which is more generalizable and easy to implement. 2) A generation framework called SEG2ACT is proposed, which adopts a global context-aware generative model to better encode the semantics of text segments and model the long-range dependencies between them. 3) SEG2ACT significantly outperforms baselines in both supervised and transfer settings, showing its effectiveness and the generalization ability.

2 SEG2ACT: Document Logical Structuring as Action Generation

2.1 Overview

As mentioned, this work considers document logical structuring task as an action generation task. That is, given a sequence of text segments $X = x_1, ..., x_N$, the goal is to produce a sequence of actions $Y = y_1, ..., y_N$, which are further used to construct the logical structure T of the document. The overall framework of SEG2ACT is depicted in Figure 2. Specifically, given a sequence of text segments, a window with w_I segments is input to the action generation model iteratively, to obtain an action sequence consisting of three types of actions. During a generation step, the previous actions and segments are constructed as a global context stack, which can provide global information for the action generation model. After that, the generated actions update both document logical structure and global context stack simultaneously. Once all text segments have been processed, the complete logical structure of target document will be produced.

2.2 Actions for Document Logical Structuring

The logical structure is a hierarchical tree composed of heading and paragraph nodes, where the depth of a node represents its level. Before structuring, a level-0 heading node with no textual content is added as the root. To achieve one-pass structuring, we define three actions to map text segments to the logical structure:

- **New Level-k Heading**: this action signifies adding the text segment as a new level-k heading node to the current document logical structure, with the last added level-$(k-1)$ heading node serving as its parent. We use k consecutive "+" to represent it.
- **New Paragraph**: this action denotes adding the current segment as a new paragraph node to the document logical structure, with the last added heading node serving as its parent. We use an asterisk "*" to represent it.
- **Concatenation**: this action indicates that the corresponding segment is an extension of the preceding text. It appends the text of the corresponding segment to the last added node of the current document logical structure. We use an equal sign "=" to represent it.

Previous works, such as TRACER (Zhu et al., 2023), also define a series of actions, but they are performed under pairwise local transitions, and a segment may participate multiple times due to the shift-reduce operation. In contrast, SEG2ACT establishes a one-to-one relationship between segments and actions, directly mapping text segments to specific positions in the document's logical structure. This design reduces the number of necessary predictions, resulting in a more efficient process.

2.3 Action Generation Model

The action generation model refers to a generative language model, which is adopted to convert text segments into action sequence by considering the global information. Specifically, as illustrated in Table 1, this action generation model takes a global context stack and the current input text segments as input to predict actions for constructing the logical structure. In this section, we first describe the global context stack, which enhances the action generation as it provides global information. Then, we present the multi-segment multi-action strategy, wherein w_I segments are converted into w_O actions at each step, which broadens the model's perspective and accelerates the construction process.

```
### STACK:
+       Government Bonds Credit Rating Report ↩
++      Credit Quality Analysis for this Series ↩
+++     Use of Proceeds ↩
*       The funds raised from the Government Bonds are ... and
        projects related to agriculture, ↩

### SEGMENT:
        forestry, water resources and social services. ↩
        Payment Security Analysis ↩
        The proceeds for the projects funded by this bond issue
        are derived from project operational revenues ↩

### ACTION:
= ↩     +++ ↩     * ↩
```

Table 1: A demonstration example of the model template in a single prediction step. It utilizes the global context stack and multi-segment multi-action strategy. "↩" denotes a line break.

2.3.1 Global Context Stack

To keep the action generation model informed about the ongoing construction process, we design a global context stack to provide global information to aid the model in decision-making. Specifically, as shown in Figure 2 and Table 1, we utilize the same symbols ("+" and "*") as actions introduced in Section 2.2 to organize previous text.

The global context stack selectively contains a subset of nodes from the constructed logical structure. Initially, the stack contains only the root node. For each generation step, it is updated according to the generated actions: **New Level-k Heading** continuously pops nodes until the stack top is a level-$(k-1)$ heading, then pushes the new level-k heading node. **New-Paragraph** pops the paragraph node (if any) from the top of the stack, then pushes the new paragraph node. **Concatenation** appends the current text segment to the top node of the stack. Thus, the stack stores the last added node at the top, followed by all the nodes along the upward backtracking path in the hierarchical tree, which we intuitively regard as being closely related to the current structuring.

Based on this approach, the global context stack models the long-distance dependencies in a centralized manner, enabling global information to be facilitated within a limited input length.

2.3.2 Multi-segment Multi-action Strategy

Since documents are segmented at the line level, there would be a lot of text segments for a document waiting for action prediction. For example, the average number of text segments in the HierDoc dataset (Hu et al., 2022b) is 853.38. Therefore, if we process text segments one by one, it is not only insufficient for capturing the complete semantics, but also inefficient for obtaining all actions of segments. To this end, we propose a multi-segment multi-action strategy to strengthen our SEG2ACT framework to be more practical.

Specifically, we not only extend the length of the input segment window, denoted as w_I, but also extend the output action window's length, denoted as w_O. On one hand, in a single prediction step, the action generation model receives w_I consecutive text segments, which allows the input segment window to encompass a more extensive range of contextual information, facilitating informed decision-making by the model. On the other hand, we can instruct the action generation model to predict w_O actions in a single step to speed up the whole generation process, thereby reducing the required number of prediction steps to $\lceil N/w_O \rceil$, where $1 \leq w_O \leq w_I$. When $w_I = w_O$, it is our default setting, representing the one-pass mode.

2.4 Model Training and Inference

In this section, we first describe how to train the action generation model, and then introduce the inference process that includes constraints.

2.4.1 Training

The training dataset consists of a collection of documents, each denoted as D. Each document is comprised of text segments $X = x_1, ..., x_N$, along with a corresponding sequence of action annotations $Y = y_1, ..., y_N$. More details of data pre-processing can be found in Appendix A. We optimize the global context-aware action generation model using teacher-forcing cross-entropy loss, which is defined as:

$$\mathcal{L} = -\sum_{i=1}^{|D|} \log P(y_{i:i+w_I-1} | s_i, x_{i:i+w_I-1}; \Theta) \quad (1)$$

Algorithm 1: Text segments to logical structure

```
Input      : Text segments $X = x_1, ..., x_N$,
             input segment window's length $w_I$,
             output action window's length $w_O$.
Output     : Document logical tree structure $T$.
Initialize : root ← HeadingNode(),
             stack $S$ ← [root], tree $T$ ← [root].
for $i \leftarrow 1$ to $\lceil N/w_O \rceil$ do
    segments ← $[x_{(i-1)\cdot w_O+1}, ..., x_{(i-1)\cdot w_O+w_I}]$;
    actions ← Model($S$, segments);
    for $j \leftarrow 1$ to $w_O$ do
        if actions[$j$] = "New Level-$k$ Heading" then
            node ← HeadingNode(segments[$j$]);
            UpdateStackAndTree($S$, $T$, node);
        end
        else if actions[$j$] = "New Paragraph" then
            node ← ParagraphNode(segments[$j$]);
            UpdateStackAndTree($S$, $T$, node);
        end
        else if actions[$j$] = "Concatenation" then
            ConcatText($S$, $T$, segments[$j$]);
        end
    end
end
return $T$;
```

where s_i represents the global context stack associated with the text segment x_i and Θ denotes the parameters of the model. For multi-segment multi-action strategy, w_I represents the input segment window size. The model learns to predict actions aligned with w_I, namely, $w_I = w_O$, which means the number of predicted actions is equal to the number of input segments during model training.

2.4.2 Inference

Given a sequence of text segments from a document, as shown in Algorithm 1, we utilize the trained action generation model to generate actions for segments, and then parse the actions to obtain the logical structure. During inference, after setting the input size of segments w_I, we can use w_O to control the speed of the iterative action execution process. The greedy search algorithm is used to generate the action sequence. At each generation step, we parse w_O actions to update the document logical structure, as outlined in Section 2.2, and update the global context stack as described in Section 2.3.1. After all segments are processed, we can obtain the final logical structure for the document.

To ensure the validity of the generated action sequence and the effective updating of the logical structure and the stack, we apply some hard constraints. For example, tokens outside of a predefined set will be banned, and the concatenation action "=" cannot be generated when the stack contains only the root node. All constrains and the execution method can be found in Appendix B. In rare cases where the output number of actions mismatches w_I, we treat these as failures, skip these segments, and continue to the next generation step.

3 Experiments

This section evaluates Seg2Act by conducting experiments in both supervised learning and transfer learning settings.

3.1 Experimental Settings

Datasets. We conduct experiments on the following datasets: 1) ChCatExt corpus (Zhu et al., 2023), which contains text segments from 650 Chinese documents and corresponding logical structures. 2) HierDoc corpus (Hu et al., 2022b), consisting of 650 English scientific documents and corresponding Table-of-Content (ToC) structures, which contains only heading annotations.

Metrics. For evaluation, we use the same criteria in previous work, including F1-score and TEDS (Hu et al., 2022b; Zhu et al., 2023). Additionally, we add a new criterion, DocAcc, to evaluate the accuracy of logical structures at the document level.

DocAcc. A prediction is considered to be correct only when the logical structure exactly matches the ground truth; otherwise, it is judged as incorrect.

Baselines. We compare our method with the following two groups of baselines:

1) Baselines using text only: TRACER (Zhu et al., 2023) is a transition-based framework for logical structure extraction, which predicts transition actions by encoding local pairwise text segments through a pre-trained language model.

2) Baselines using text, layout and vision: MTD (Hu et al., 2022b) is a multi-modal method that utilizes pre-trained models to encode visual, textual, and positional document information, extracting ToC by attention and pairwise classification stages; CMM (Wang et al., 2023) is a three-stage framework that starts with a heuristic-based initial tree, then encodes nodes with pre-trained models, and finally refines the tree by moving or deleting nodes.

For our approach, we conduct the experiments of two settings:

1) Seg2Act. It is a global context-aware action generation method proposed in this paper, which generates the document logical structure in an end-to-end, one-pass manner.

2) Seg2Act-T. It is a modified version of TRACER, in which we utilize our proposed global context-aware generative model as the action parser, while still generating shift-reduce actions and following constraints akin to TRACER.

Method	Heading	Paragraph	Total	DocAcc
Methods using RBT3 as Backbone				
TRACER	90.49	84.33	82.39	-
TRACER*	90.04	83.96	82.07	26.15
Methods using GPT2-Medium as Backbone				
TRACER*	91.15	88.53	85.40	47.38
Seg2Act-T (Ours)	93.94	91.21	89.01	52.00
Seg2Act (Ours)	94.88	92.99	90.96	57.23
Methods using Baichuan-7B as Backbone				
TRACER*	94.91	91.62	89.55	53.85
Seg2Act-T (Ours)	**96.01**	93.98	92.39	58.46
Seg2Act (Ours)	**96.01**	**94.19**	**92.63**	**63.69**

Table 2: Overall performance on ChCatExt (Heading, Paragraph, Total nodes in F1-score and logical structure accuracy at the document level). TRACER* refers to our implemented results.

Method	Modality	Backbone	HD	ToC
MTD	T+L+V	BERT+ResNet	96.1	87.2
CMM	T+L	RoBERTa	97.0	88.1
Seg2Act (Ours)	T	GPT2-Medium	96.3	93.3
		Baichuan-7B	**98.1**	**96.3**

Table 3: Heading detection (HD) in F1-score and ToC in TEDS (%) of baselines and Seg2Act on HierDoc.

Implementations. Our implementations are built upon Pytorch (Paszke et al., 2019), Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022) libraries. For both GPT2-Medium and Baichuan-7B backbone models (Radford et al., 2019; Baichuan-inc, 2023), we employ the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 3×10^{-4}. The number of training epochs is set to 10, and the batch size is set to 128. We set the input segment window and output action window as $w_I = w_O = 3$. Experiments are conducted on an NVIDIA A100 GPU. For the transfer learning experiments, we initially pre-train models on the Wiki corpus (provided by Zhu et al. (2023)) for 10,000 steps. Besides, we utilize the LoRA (Hu et al., 2022a) technique to reduce the GPU memory overhead during Baichuan-7B training. We set the rank r to 8 and the alpha value α to 16. All experiments are averaged results obtained from five different random seeds to ensure robustness and reliability.

3.2 Results in Supervised Learning Setting

Table 2 shows the performance of text-only baselines on the ChCatExt, And Table 3 compares the

Test Set	Method	Zero-Shot	Few-Shot						Full-Shot		
			BidAnn		FinAnn		CreRat		BidAnn	FinAnn	CreRat
			3	5	3	5	3	5			
BidAnn	TRACER	2.70	66.64	14.58	2.36	21.48	1.02	12.78	88.20	25.26	11.74
	SEG2ACT-T (Ours)	42.92	96.53	97.07	86.45	85.89	88.78	89.53	99.40	95.74	**73.49**
	SEG2ACT (Ours)	**56.25**	**99.31**	**99.45**	**90.30**	**96.89**	**95.59**	**97.12**	**99.72**	**98.07**	69.92
FinAnn	TRACER	11.39	**67.04**	15.51	3.52	32.57	1.77	25.29	8.10	68.59	20.17
	SEG2ACT-T (Ours)	28.98	26.17	**28.18**	42.37	58.21	47.71	48.15	32.47	76.87	46.04
	SEG2ACT (Ours)	**43.30**	25.00	23.51	**56.17**	**75.19**	**48.38**	**55.11**	**47.92**	**85.17**	**60.25**
CreRat	TRACER	14.07	**79.03**	16.42	4.52	27.53	18.66	19.24	7.00	30.82	92.29
	SEG2ACT-T (Ours)	49.65	35.71	31.31	**47.79**	56.36	71.63	84.77	32.33	42.19	95.77
	SEG2ACT (Ours)	**67.86**	55.22	**54.75**	24.32	**65.73**	**82.77**	**86.59**	**61.20**	**70.14**	**97.76**

Table 4: Performance (F1-score of total nodes) on transfer learning experiments in zero-shot, few-shot and full-shot settings on three sub-corpora of ChCatExt: bid announcements (BidAnn) with 100 documents, financial announcements (FinAnn) with 300 documents, and credit rating reports (CreRat) with 250 documents..

Method	Heading	Paragraph	Total	DocAcc
SEG2ACT	**96.01**	94.19	92.63	**63.69**
- multi-segment multi-action	95.49	93.32	91.56	62.15
- GCS (symbol)	95.71	**94.28**	**92.69**	57.23
- GCS (text)	90.92	87.52	83.85	50.77
- GCS (both text and symbol)	89.45	85.36	81.15	44.92

Table 5: Performance on ChCatExt with ablated settings. GCS denotes the global context stack.

performance of multi-modal baselines on HierDoc with text-only SEG2ACT. We can see that:

1) **By generating the logical structure in an end-to-end manner, SEG2ACT achieves state-of-the-art performance.** In Table 2, SEG2ACT predicts the document logical structure with high accuracy, and outperforms TRACER under the same Baichuan-7B backbone by +9.84 in DocAcc. Table 3 shows that our method performs better than multi-modal methods in both HD and TEDS, even though it only uses semantic information. These above indicate that our SEG2ACT can better perceive the overall logical structures of the documents.

2) **Global contextual information plays a crucial role in document logical structuring.** In Table 2, injecting global context stack into TRACER produces a general performance improvement. In both GPT2-Medium and Baichuan-7B backbones, SEG2ACT-T surpasses TRACER in terms of F1-score for headings, paragraphs, total nodes and document-level accuracy. This highlights the significance of the global context stack for document logical structuring.

3.3 Results in Transfer Learning Setting

To assess the generalization of SEG2ACT, we first pre-train the backbone model on the Wiki corpus and then conduct a series of transfer learning experiments under zero-shot, few-shot, and full-shot settings, as shown in Table 4. For ease of presentation, we use the F1-score of total nodes as the representative metric. We observe that:

1) **The action generation framework of SEG2ACT can learn general document structures instead of capturing type-specific features.** Compared with SEG2ACT-T, SEG2ACT attains average improvements of +10.65, +6.04, +15.28 for full-shot, few-shot, and zero-shot settings, exhibiting its superiority in various scenarios.

2) **SEG2ACT can robustly resist data scarcity, displaying a quick adaptation capability.** Taking the case of 5-shot training as an example, SEG2ACT only averages a slight drop of 3.98 compared to the full-shot setting.

3.4 Ablation Study

3.4.1 Effects of Global Context Stack

Table 5 shows the impact of global information on SEG2ACT. We break down the global context stack formatted in the schema into two components: text and symbol. The symbol represents the hierarchical mark before the text, such as "+" and "*". Therefore, deleting the global context stack (symbol) means using only the texts in the schema, and deleting the global context stack (text) means using only symbols in the schema. We observe that:

1) **The structural representation schema offers an effective way to perceive the global document structure.** When hierarchical symbols are removed, SEG2ACT's ability to predict the overall document structure significantly diminishes, resulting in a decrease of 6.46 in DocAcc.

2) **There's an inherent trade-off between hierarchical prediction and paragraph concate-**

Input Segment Window	Output Action Window $w_O = 1$	 $w_O = 2$	 $w_O = 3$	 $w_O = 4$	 $w_O = 5$
$w_I = 1$	**91.56** **(10.43s)**	-	-	-	-
$w_I = 2$	**93.14** (17.61s)	92.99 **(8.79s)**	-	-	-
$w_I = 3$	**92.83** (24.73s)	91.73 (12.29s)	92.63 **(8.13s)**	-	-
$w_I = 4$	**92.32** (31.23s)	91.41 (15.83s)	91.76 (10.48s)	91.74 **(7.75s)**	-
$w_I = 5$	93.03 (38.44s)	**93.06** (19.64s)	91.40 (12.96s)	92.76 (9.76s)	92.43 **(7.63s)**
Baseline	89.55 (10.58s)				

Table 6: The F1-score of total nodes (inference time per document) of scaling the lengths of the input segment window and output action window for SEG2ACT on ChCatExt. Baseline refers to TRACER in Baichuan-7B.

nation with the use of the global context stack. We notice a slight change of -0.06, -0.09, +0.3 in F1-score for total nodes, paragraph nodes and heading nodes, respectively, when hierarchical symbols are added. These symbols encourage SEG2ACT to focus on hierarchical discrimination, slightly diminishing its ability to concatenate paragraphs and resulting in a minor decrease in F1-score.

3.4.2 Effects of Multi-segment Multi-action

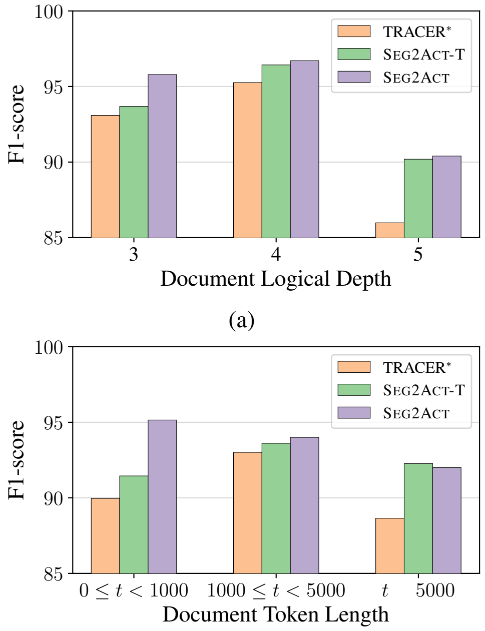
To verify the effect of the multi-segment multi-action strategy on SEG2ACT's performance and efficiency, we scale the lengths of input segment window and output action window from 1 to 5, conducting experiments on ChCatExt. We take F1-score for total nodes as the metric and measure the average inference time for each document, as shown in Table 6. We can see that:

1) **Providing insights from the following consecutive segments mitigates short-sighted issue and enhances performance.** Extending the input segment window length w_I to 2, 3, 4, and 5, the SEG2ACT method exhibits improvements in F1-score of +1.58, +1.27, +0.76, and +1.50, compared to the case where $w_I = 1$.

2) **Simultaneously generating multiple actions ensures decoding efficiency of SEG2ACT.** By increasing the output action window length w_O, SEG2ACT experiences a reduction in inference time while maintaining comparable performance. For instance, when comparing ($w_I = 3$,$w_O = 3$) with ($w_I = 1$,$w_O = 1$), SEG2ACT demonstrates a notable improvement with a +1.07 increase in F1-score and a ×0.28 boost in inference speed.

3.5 Analysis of Document Length

To analyze the impact of document length, we show the performance on different subsets of ChCatExt in Figure 3. We can observe that:



(a)

(b)

Figure 3: Results (F1-score of total nodes) for documents with different logical tree depths (a) and token lengths (b) on ChCatExt dataset.

1) **Our proposed actions are more effective for complex document logical structure than shift-reduce actions**. As the depth of the logical structure increases, the performance of all models significantly declines. However, SEG2ACT still achieves the best performance among the three models.

2) **Global contextual information improves the logical structure handling of lengthy documents**. As document token length increases, models with global context experience a smaller performance drop compared to TRACER.

3.6 Case Study

We illustrate two cases in the prediction steps, as depicted in Table 7. In the first scenario, the local pairwise method TRACER fails to predict the current input segment for the "Reduce" action due to a lack of global perspective. On the contrary, our SEG2ACT successfully predicts the correct type and level with the assistance of the global context stack. In the second case, expanding the input segment window enables the model to make more insightful decisions. These two cases highlight the effectiveness of our method.

Method	Stack	Segment	Predicted Action	
TRACER*	Risk Principle ↩	Chapter 3 Basis and Scope for Determining the Holders of Employee Stock Ownership Plans ↩	New Paragraph	✗
SEG2ACT ($w_I = w_O = 1$)	+ Summary of Employee Stock Ownership Plan (Draft) ↩ ++ Chapter 2 Purpose and Basic Principles of Employee Stock Ownership Plans ↩ +++ 2. The basic principles of employee stock ownership plans ↩ ++++ Risk Principle ↩ * Participants in this employee stock ownership plan ... equal rights and interests with other investors. ↩	Chapter 3 Basis and Scope for Determining the Holders of Employee Stock Ownership Plans ↩	New Level-2 Heading	✓
SEG2ACT ($w_I = w_O = 1$)	+ Announcement on the Inquiry Letter on Matters Related to the Company's Application for Bankruptcy ↩	— Is the early acquisition decision reasonable? ↩	New Paragraph	✗
SEG2ACT ($w_I = w_O = 3$)	+ Announcement on the Inquiry Letter on Matters Related to the Company's Application for Bankruptcy ↩	— Is the early acquisition decision reasonable? ↩ On January 20, 2021, the company announced that it would acquire 100% equity of HNA Airport Group from its related party Hainan Airlines Travel Service Co., Ltd. for 500 million yuan, with a net asset value of 34.073 million euros. The transaction ↩ appreciation rate is about 87%, and the main assets of HNA Airport Group are 82.5% equity of Hahn Airport in Frankfurt, Germany (hereinafter referred to as Hahn Airport). In the short term, the company has announced that HNA Airport Group and Hahn Airport have filed for bankruptcy. ↩	New Level-2 Heading New Paragraph Concatenation	✓

Table 7: A case study for models utilizing the Baichuan-7B backbone.

4 Related Work

Document logical structuring has received significant attention for an extended period (Tsujimoto and Asada, 1990; Summers, 1998; Mao et al., 2003; Luong et al., 2010; Pembe and Güngör, 2015; Gopinath et al., 2018; Maarouf et al., 2021; Zhu et al., 2023). Traditional methods have predominantly focused on designing heuristic or handcrafted rules to extract logical structures (Fisher, 1991; Conway, 1993). For instance, text regular matching methods can be employed to differentiate headings from paragraphs. However, a notable drawback of such rule-based approaches is their specificity to certain document types, limiting their applicability to others.

In recent years, the advent of deep learning has opened up new avenues for document logical structuring, with a particularly promising trend being multi-modal and multi-stage modeling (Bourez, 2021; Cao et al., 2022). From a multi-modal perspective, the incorporation of layout and vision modalities enhances the representation of semantic structures (Hu et al., 2022b; Wang et al., 2023). On the other hand, adopting a multi-stage approach involves decomposing the task into subtasks, which facilitates an easier and more manageable modeling process (Rahman and Finin, 2017; Bentabet et al., 2019). While multi-modal methods excel with single-page document images, they struggle to effectively model the intricate structures of lengthy, multi-page documents. Similarly, multi-stage methods encounter challenges related to error propagation when concatenating all stages in real-world applications.

Another noteworthy direction is the transition-based extraction (Koreeda and Manning, 2021; Zhu et al., 2023). Transition-based methods parse texts into structured trees from the bottom up, offering efficiency and suitability for very long documents. However, these methods focus on pairwise local context, capturing only local information while neglecting the global information of the documents.

In contrast to previous works, our research introduces an end-to-end and generation-based method. This approach minimizes error propagation and enhances generalization. Furthermore, our framework, incorporating global context information, helps the action generation process and efficiently predicts the logical structure of documents.

5 Conclusions

This paper proposes SEG2ACT, a novel method that models document logical structuring task as an end-to-end, one-pass action generation process. By leveraging the generative language model as an action generator and incorporating a global context stack, SEG2ACT achieves significant performance and strong generalization on two benchmark datasets. For future work, we plan to explore the integration of long-context language models and multi-modal language models with the SEG2ACT framework.

Limitations

First, generating indefinite-length action sequence using generative model may result in some cases that are challenging to parse, despite being constrained by hard rules. For example, in the multi-segment multi-action strategy, it cannot be guaranteed that the model will always output action sequence matching the specified w_I count.

Second, our approach does not utilize visual information, thus requiring a proper order of input text segments, making it difficult to handle sequence with disrupted text segment order. Therefore, more effort is needed to incorporate visual information, making our method more flexible and applicable in a wider range of scenarios.

Ethics Statement

In consideration of ethical concerns, we provide the following detailed descriptions:

1) All the data and backbone model weights we use come from publicly available sources. When using these resources for this study, we strictly adhere to their licensing agreements.

2) Our approach relies on large language models such as Baichuan-7B (Baichuan-inc, 2023) as its backbone. As these language models have been trained on extensive text data sourced from the Web, it may be susceptible to issues such as toxic language and bias. However, our model is further fine-tuned to only generate structural actions and can only be used for document logical structuring, significantly mitigating the impact of these concerns.

Acknowledgements

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by the Natural Science Foundation of China (No. 62306303, 62122077 and 62106251), and the Basic Research Program of ISCAS (Grant No. ISCAS-ZD-202402).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *CoRR*, abs/2309.16609.

Baichuan-inc. 2023. Baichuan-7B. `https://github.com/baichuan-inc/Baichuan-7B`.

Najah-Imane Bentabet, Rémi Juge, and Sira Ferradans. 2019. Table-of-contents generation on contemporary documents. In *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*, pages 100–107. IEEE.

Christopher Bourez. 2021. FINTOC 2021 - document structure understanding. In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 89–93, Lancaster, United Kingdom. Association for Computational Linguistics.

Rongyu Cao, Yixuan Cao, Ganbin Zhou, and Ping Luo. 2022. Extracting variable-depth logical document hierarchy from long documents: Method, evaluation, and application. *J. Comput. Sci. Technol.*, 37(3):699–718.

Alan Conway. 1993. Page grammars and page parsing. A syntactic approach to document layout recognition. In *2nd International Conference Document Analysis and Recognition, ICDAR '93, October 20-22, 1993, Tsukuba City, Japan*, pages 761–764. IEEE Computer Society.

JL Fisher. 1991. Logical structure descriptions of segmented document images. *Proceedings of International Conference on Document Analysis and Recognition*, pages 302–310.

Abhijith Athreya Mysore Gopinath, Shomir Wilson, and Norman M. Sadeh. 2018. Supervised and unsupervised methods for robust separation of section titles and prose text in web documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 850–855. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022a. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Jun Du, and Jiajia Wu. 2022b. Multimodal tree decoder for table of contents extraction in document images. In *26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022*, pages 1756–1762. IEEE.

Yuta Koreeda and Christopher D. Manning. 2021. Capturing logical structure of visually structured documents with multimodal transition parser. In *Proceedings of the Natural Legal Language Processing*

Workshop 2021, NLLP@EMNLP 2021, Punta Cana, Dominican Republic, November 10, 2021, pages 144–154. Association for Computational Linguistics.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip S. Yu. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 188–200. Association for Computational Linguistics.

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

Minh-Thang Luong, Thuy Dung Nguyen, and Min-Yen Kan. 2010. Logical structure recovery in scholarly articles with rich document features. *Int. J. Digit. Libr. Syst.*, 1(4):1–23.

Ismail El Maarouf, Juyeon Kang, Abderrahim Ait Azzi, Sandra Bellato, Mei Gan, and Mahmoud El-Haj. 2021. The financial document structure extraction shared task (FinTOC2021). In *Proceedings of the 3rd Financial Narrative Processing Workshop*, pages 111–119, Lancaster, United Kingdom. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Song Mao, Azriel Rosenfeld, and Tapas Kanungo. 2003. Document structure analysis algorithms: a literature survey. In *Document Recognition and Retrieval X, Santa Clara, California, USA, January 22-23, 2003, Proceedings*, volume 5010 of *SPIE Proceedings*, pages 197–207. SPIE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

F. Canan Pembe and Tunga Güngör. 2015. A tree-based learning approach for document structure analysis and its application to web search. *Nat. Lang. Eng.*, 21(4):569–605.

Yifu Qiu and Shay B. Cohen. 2022. Abstractive summarization guided by latent hierarchical document structure. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5303–5317. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Muhammad Mahbubur Rahman and Tim Finin. 2017. Deep understanding of a document's structure. In *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies, BDCAT 2017, Austin, TX, USA, December 05 - 08, 2017*, pages 63–73. ACM.

Jon Saad-Falcon, Joe Barrow, Alexa F. Siu, Ani Nenkova, Ryan A. Rossi, and Franck Dernoncourt. 2023. Pdftriage: Question answering over long, structured documents. *CoRR*, abs/2309.08872.

Kristen Maria Summers. 1998. *Automatic Discovery of Logical Document Structure*. Ph.D. thesis, Cornell University, USA.

Shuichi Tsujimoto and Haruo Asada. 1990. Understanding multi-articled documents. In *10th IAPR International Conference on Pattern Recognition, Conference A: Computer Vision & Conference B Pattern recognition systems and applications, ICPR 1990, Atlantic City, NJ, USA, 16-21 June, 1990, Volume 1*, pages 551–556. IEEE.

Xinyu Wang, Lin Gui, and Yulan He. 2023. A scalable framework for table of contents extraction from complex ESG annual reports. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13215–13229. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Tong Zhu, Guoliang Zhang, Zechang Li, Zijian Yu, Junfei Ren, Mengsong Wu, Zhefeng Wang, Baoxing Huai, Pingfu Chao, and Wenliang Chen. 2023. CED: catalog extraction from documents. In *Document Analysis and Recognition - ICDAR 2023 - 17th International Conference, San José, CA, USA, August 21-26, 2023, Proceedings, Part III*, volume 14189 of *Lecture Notes in Computer Science*, pages 200–215. Springer.

A Data Pre-processing

Currently, most datasets of document logical structuring are labeled with logical tree structure. In order to train our model, we convert the logical tree structure to our training corpus using preorder traversal, as illustrated in Algorithm 2.

Algorithm 2: Logical structure to training data

```
Input      : Document logical tree structure $T$.
Output     : Text segments $X = x_1, ..., x_N$,
             action sequence $Y = y_1, ..., y_N$.
Initialize : $X \leftarrow [\,]$, $Y \leftarrow [\,]$.
Procedure Travel(node):
    $X$.extend( node.content );
    if node.type = "Heading" then
        $Y$.append( "+" * len(node.depth) );
    else
        $Y$.append( "*" );
    end
    segment_num ← len(node.content);
    if segment_num > 1 then
        for $i \leftarrow 2$ to segment_num do
            $Y$.append( "=" );
        end
    end
    child_num ← len(node.children);
    for child ∈ node.children do
        Travel(child);
    end
return $X, Y$ after Travel($T$.root);
```

B Action Constraints

For SEG2ACT-T, we conduct the same constraints as TRACER (Zhu et al., 2023), which includes four actions: *Sub-Heading*, *Sub-Text*, *Reduce*, *Concat*. The constraints are as follows:

- The action between Root node and the first input text segment can only be *Sub-Heading* or *Sub-Paragraph*;
- The paragraph nodes can only be leaf nodes in the logical tree structure. Thus, if the last segment is predicted to be a paragraph node, only *Reduce* and *Concat* actions are permitted for the prediction of current segment.

For SEG2ACT, the constraints are as follows:

- The predicted token must be in the predefined action set. We only allow token prediction in predefined set {"+", "*", "=", "\n"} and ban all other predictions through `LogitsProcessor` of the Transformer library (Wolf et al., 2020), which supports forcibly setting token prediction probability to 0;
- The *Concatenation* action cannot be performed when the stack contains only the root node. Therefore, the action for the first input text segment can only be *New Level-1 Heading* or *New Paragraph* (indicating that the initial predicted token can only be "+" or "*"). We also utilize `LogitsProcessor` to execute this constraint;
- Heading nodes are prohibited from skipping levels, and if they do so, they are constrained to be at the current maximum level plus 1 (for example, if the generated action is "++++" but the maximum level of the heading nodes in the global context stack is only 2, we modify the decoded action to be *New Level-3 Heading*). This constraint ensures that the parent node for newly added nodes can be found within the stack and the tree structure.

For the first constraint of SEG2ACT, different models may use different tokenizers, resulting in different token prediction strategies. In addition, the tokens allowed to be predicted are also related to the model's last generated tokens. Table 8 shows the allowed token predictions for the GPT2-Medium and Baichuan-7B models, respectively.

Last Token	Next Token						
	\n	+	++	++++	*	=	</s>
\n		√	√	√	√	√	√
+	√	√	√	√			
++	√	√	√	√			
++++	√	√	√	√			
*	√						
=	√						

(a) The allowed token predictions in GPT2-Medium model.

Last Token	Next Token					
	\n	+	++	*	=	</s>
\n		√	√	√	√	√
+	√	√	√			
++	√	√	√			
*	√					
=	√					

(b) The allowed token predictions in Baichuan-7B model.

Table 8: The allowed token predictions for models with different tokenizers.

C Effects of Model Size

In this section, we explore the impact of model size on our proposed framework.

Method	Heading	Paragraph	Total	DocAcc
Methods using Baichuan-7B as Backbone				
TRACER*	94.91	91.62	89.55	53.85
SEG2ACT-T (Ours)	96.01	93.98	92.39	58.46
SEG2ACT (Ours)	96.01	94.19	92.63	63.69
Methods using Baichuan-13B as Backbone				
TRACER*	94.79	92.49	90.39	54.15
SEG2ACT-T (Ours)	95.97	93.73	92.06	60.62
SEG2ACT (Ours)	**96.25**	**94.40**	**92.83**	**67.08**

Table 9: The result on ChCatExt (Heading, Paragraph, Total nodes in F1-score and logical structure accuracy at the document level).

Model	Total	DocAcc	TimeCost
Qwen1.5-0.5B	92.22	57.54	**4.01s**
Qwen1.5-1.8B	**92.99**	63.69	4.27s
Qwen1.5-4B	92.93	**65.23**	7.06s
Baichuan-7B	92.63	63.69	8.13s

Table 10: The result of SEG2ACT on ChCatExt (Total nodes in F1-score, logical structure accuracy at the document level and time cost per document).

As demonstrated in Table 9, enlarging models can boost performance, and among models of equal size, those integrating global context typically exhibit superior performance.

However, the performance gains from increasing the model size are not cost-effective compared to the expenses of training larger models. Additionally, larger models result in longer inference times, making efficiency a critical concern in practical applications. Therefore, we also discuss the performance of our proposed SEG2ACT framework when decreasing the model size. Since there is no version of the Baichuan model smaller than 7B size, we choose Qwen1.5 model (Bai et al., 2023) for experiments. As shown in Table 10, we can observe that:

1) **Backbone model choice affects performance**. Comparing the Qwen1.5 and Baichuan backbone models, the Qwen1.5-4B outperforms the Baichuan-7B in F1-score and document-level accuracy, while also being smaller in model size.

2) **The action generation framework may not necessarily require an oversize model**. For instance, in the Qwen1.5 series of models, the Qwen1.5-1.8B model achieves similar performance to the Qwen1.5-4B, but is ×0.65 faster.