Learning Part-whole Hierarchies from the Sequence of Handwriting

Meng Li [1], **David Schlangen** [1], **Dietrich Klakow** [2]
[1]Department of Linguistics, University of Potsdam
[2]Department of Language Science and Technology, Saarland University
{meng.li, david.schlangen}@uni-potsdam.de, dietrich.klakow@lsv.uni-saarland.de

Abstract

Part-whole relations and their representation play a vital role in perceptual organization and conceptual reasoning. It is critical for humans to parse visual scenes into objects and parts, and organize them into hierarchies. Few studies have examined how well neural networks learn part-whole hierarchies from visual inputs. In this paper, we introduce a new diagnostic dataset, CChar, to facilitate their understanding. It contains frame-based images of writing 6,840 Chinese characters and annotations on hierarchical structures. The results show that RNN and Transformer models could recognize a part of high-level components above strokes and illustrate a certain ability in learning part-whole hierarchies. However, these models do not have robust compositional reasoning. To identify the role of conceptual guidance in predicting hierarchical structures, we prepare visual features extracted by self-supervised and fine-tuned models, test them on generating hierarchical sequences, and observe that conceptual guidance is important to learn part-whole hierarchies. In addition, we also explore the relationship between the depth of hierarchies and model performance. It is found that RNNs perform worse as the hierarchies deepen, but the performance of Transformers becomes better with increasing depth.

Keywords: part-whole hierarchy, compositionality, neural networks, Chinese character

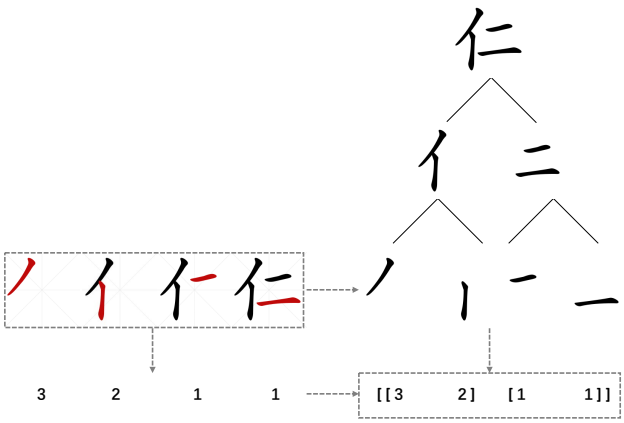

Figure 1: The process of deriving representations of part-whole hierarchies in a Chinese character, "仁" (kindheartedness): (1) represent the current stroke in each image with a categorical number; (2) parse the structure and add brackets to annotate boundaries. The current stroke in each image is highlighted in red.

Introduction

Part-whole representation is a vital aspect of how humans perceive and think about objects and their parts. The effective representation of part-whole hierarchies in artificial neural networks is still a challenge (Hinton, 2022; Culp, Sabour, & Hinton, 2022), although there is a long history of research on part-whole representations in visual perception and cognition (Wertheimer, 1938; Palmer, 1977; Marr & Nishihara, 1978; Marr, 1982; Hinton, 1990; Riesenhuber & Dayan, 1996). Symbolic AI approach and interpretable architectural approach are two general approaches for representing part-whole hierarchies in a neural network (Hinton, 2022).

Capsule Networks (Sabour, Frosst, & Hinton, 2017; Hinton, Sabour, & Frosst, 2018), GLOM (Hinton, 2022; Culp et al., 2022) and Agglomerator (Garau, Bisagno, Sambugaro, & Conci, 2022) are attempts to represent part-whole hierarchies with more interpretable architectures. Specifically, part-whole hierarchies are coupled with the architecture of neural networks and used to interpret neural networks. However, it is not easy to verify the claim that visual representations at different levels in these models directly correspond to different concepts in part-whole hierarchies.

For the symbolic approach, one solution is to create a parse graph where nodes for parts and wholes are organized in a hierarchical structure. *Scene graphs* have been proposed to describe object instances in a scene, attributes of objects, and relationships between objects (Johnson et al., 2015). Their vector representations could be injected or fused into a neural network to represent part-whole hierarchies.

Interpretable architectural approach assumes that we need additional prior designs to represent part-whole hierarchies. Does this mean that neural networks without such additional architectural designs cannot represent or learn part-whole hierarchies? And how to measure the capacity of neural networks without additional architectural designs in learning part-whole hierarchies if they could? These questions are rarely studied and may yield insights on how to better learn implicit or transparent part-whole hierarchies. To answer these questions, generating symbolic representations of part-whole hierarchies is a possible test bed. For example, the task of scene graph generation is to parse an image and generate structured representations (H. Li et al., 2024). However, in practice, the annotated scene graphs based on real-world

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

images usually have limitations (X. Li et al., 2022): (1) The manually created annotations are highly incomplete. Due to the sharp increase in the number of combination, it is unrealistic to annotate all relationships in one image. (2) The distribution of categories of objects, parts and relationships is extremely imbalanced. Therefore, we prefer a diagnostic dataset with a relatively balanced distribution, enough depth of hierarchies to compare, and interpretable symbolic representations of part-whole hierarchies.

Chinese characters exhibit "character - component - stroke" hierarchies (see Figure 1). The characters could be decomposed into components, and components could be decomposed into smaller components or strokes further. Strokes of Chinese characters can be classified into 5 types: horizontal stroke (一), vertical stroke (丨), left downward stroke (丿), dot stroke (丶), turning stroke (乙). These types can be represented with numbers (1,2,3,4,5), and each Chinese character can be represented as a sequence according to the writing order (Ministry of Education of the People's Republic of China, 2020). For example, the character "仁" could be represented as "3211". The part-whole relations could be represented with paired brackets ([]), and be viewed as bounding boxes in one dimension. There are two components in "仁", and they form a "left-right" pattern. The sequence could be "[[32][11]]" if we add brackets to mark boundaries. Therefore, the hierarchical structure of a Chinese character could be represented with sequential numbers with brackets.

For the task of learning part-whole hierarchies, the input is a sequence of images with highlighted strokes, and the output is a sequence of stroke labels with or without brackets, like [[32][11]] or 3211 (see Figure 1). There is evidence that components of Chinese characters are units of perception even when these components do not represent phonological or semantic features. Chinese children are able to recognize constituent structures of characters in elementary school, as they develop the knowledge of vocabulary and reading comprehension (Anderson et al., 2013).

We introduce a new dataset, CChar, which encodes the temporal process of writing Chinese characters and the "character - component - stroke" hierarchies.[1] CChar dataset provides a window to observe the interaction between visual features, strokes, and upper-level components in artificial neural networks. Our contributions are outlined in several aspects. (1) We propose a symbolic representation of part-whole hierarchies in Chinese characters for the first time, and introduce a new diagnostic dataset to measure the capacity of neural networks in learning part-whole hierarchies. (2) We observe that neural networks could learn a portion of part-whole hierarchies. However, their relative low accuracy in recognizing primitive components suggests that these models do not have robust compositional reasoning. (3) Fine-tuned visual features are found to be more effective than self-supervised features in generating hierarchical sequences, which demonstrates that conceptual guidance is crucial for learning part-whole hierarchies. (4) We also find that there is a negative linear relationship between the depth of hierarchies and the performance of RNNs, but not for Transformers.

[1]Data, code and appendix for this paper are available at https://github.com/limengnlp/cchar

Data

Features of Chinese Characters

Strokes The order of handwriting is prescriptive in the official document *Stroke Orders of The Commonly Used Standard Chinese Characters* (Ministry of Education of the People's Republic of China, 2020). There are some general rules for stroke orders: (1) from top to bottom (二), from left to right (孔); (2) horizontal before vertical strokes (十), left downward before dot strokes (人); (3) for character with encirclement structure, first outside, then inside, and finally closed (日); (4) for vertically symmetrical characters, first middle components, then left and right sides (小).

Components and Constituent Structures The components of Chinese characters are units comprising strokes and constructing Chinese characters. (Ministry of Education of the People's Republic of China, 1997). Different components are composed of a limited number of strokes in different patterns. There are three basic types of spatial relationship between strokes: separation, intersection, and connection. The relationship of separation means that the strokes in a component are separated from each other. The relationship of intersection means that there is a cross between the strokes in a component. The relationship of connection means the strokes in a component are connected to each other but do not cross. For example, two strokes, "丿" and "丶", can form several components like "八人入", and the difference is the way the strokes are combined with each other. In "八", there is a space between two strokes, and the relationship is separation. In "人" and "入", the strokes are connected with each other and do not cross. Components could be classified as primitive or compound components according to their levels in the constituent structures (Xing, 2007). A primitive component is the minimalist component that cannot be decomposed, such as "田" (farmland) and "力" (work force) in "男" (male). A compound component could be decomposed into smaller pieces in further. For example, "挚" (sincere) could be decomposed into "执" (hand-hold) and "手" (hand), but "执" could also be decomposed into "扌" and "丸" in further. Therefore, "执" is not a primitive component.

CChar Dataset

The CChar dataset contains 73,086 images recording the temporal process of writing 6,840 Chinese characters. For each character, there is a sequence of stroke labels with human-annotated hierarchical structures. We split the dataset randomly into train, validation, and test sets with the ratio of 70:15:15. The number of images in the training set, validation set and test set is 51,114, 11,018, 10,954. The number of characters in the training set, validation set and test set is 4,788, 1,026, 1,026. We compare our dataset with previous

datasets in Table 1. Existing part-whole datasets mostly focus on object part segmentation, and the part-whole hierarchies as abstract relations cannot be evaluated directly. The distinctive feature of CChar is explicit symbolic representations of part-whole hierarchies in Chinese characters that could be measured and interpreted directly (see more details in the Appendix at `https://github.com/limengnlp/cchar`).

Experiment

Experimental Design

All experiments in this study include two stages (see Figure 2). The first stage is to train a visual feature extractor and prepare visual features. Given the sequence of visual features from the first stage, the second stage is to generate sequences with or without hierarchical structures. There are several ways to train a visual feature extractor and get visual features in the first stage. (1) Supervised method: train the visual feature extractor on a supervised image classification dataset. For the task of stroke classification, the input is a single image with a highlighted stroke, and the output is one of the five stroke labels. (2) Self-supervised method: randomly mask patches in each image, and train models to predict them. When downstream tasks are image classification, we usually only take the encoder part as the visual feature extractor. (3) Self-supervision and supervised fine-tuning: first pre-train the visual model in a self-supervised way, then fine-tune it on a supervised dataset. In the second stage, we adopt a general encoder-decoder architecture to generate the sequence of strokes.

Experiment 1 In the first stage, we train convolutional neural networks (CNN) (LeCun et al., 1995) on the stroke classification task. The input is a single image where the current stroke is highlighted in red, and the output is the label of 5 stroke types. Then we utilize these models to extract visual features of each image. In the second stage, we compare the performance of GRU (Cho et al., 2014), LSTM (Hochreiter & Schmidhuber, 1997), and Transformer (Vaswani et al., 2017) models in generating stroke sequences with or without hierarchical annotations. The input is a sequence of images, and the current stroke in each image is highlighted in red. The output is the sequence of strokes with or without hierarchical annotations. By comparing predicted sequences and the ground truth sequence, we can investigate the capacity of neural networks in learning part-whole hierarchies. In addition, the target sequence is interpretable symbolic representations, so the accuracy of primitive components and error patterns in predicted sequences can tell us whether these models work in a robust compositional way.

Experiment 2 In the first stage, we pre-train masked autoencoders (MAE) (He et al., 2022) on the train and validation set of CChar images. The input is a single image where the current stroke is highlighted in red. Then, we fine-tune these models on stroke classification task. The input is the same setting as before, and the output is the label of 5 stroke types. We use pre-trained and fine-tuned MAE models to extract two types of visual features: one is self-supervised features, and the other is fine-tuned features. Their difference is whether feature extraction is guided by labels of stroke types. In experiment 2, we compare GRU, LSTM and Transformer in generating hierarchical sequences with two types of visual features. The input is the same as experiment 1, but the output is the sequence of strokes with hierarchical annotations only. By comparison, results could tell how important the conceptual guidance is in visual feature extraction and learning part-whole hierarchies. Part-whole hierarchies involve not only perception but also conceptual categorization and organization. We are curious about the role of conceptual guidance in visual feature extraction and the consequent effects on predicting hierarchical structures, considering part-whole hierarchies in Chinese characters are built on the stroke level.

Experiment 3 In the first stage, we have similar settings as experiment 1, and get visual features from CNN-based models. In the second stage, we control the depth of hierarchies so that data points with different hierarchies are equal in different subgroups, and randomly sample 20 different subsets from CChar. We compare the performance of GRU, LSTM, and Transformer models in generating stroke sequences with hierarchical annotations on different subsets. The input and output are the same as experiment 2. By statistical analysis, we would explore if there is a negative relationship between the depth of hierarchical structures and model performance in experiment 3.

Models

We use CNN-based models and MAE models to extract visual features.

VGG-tiny The visual feature extractors in experiment 1 and 3 are VGG-tiny models. We reduce the pyramid architecture of VGG net (Simonyan & Zisserman, 2014) to a tiny version, considering our dataset is relatively small. Images are resized to 128x128. There are only two building blocks for feature extraction, and each block has two convolutional layers and one max-pooling layer. In addition, there are also two fully-connected layers for classification, and the output dimension of the second fully-connected layer is the number of categories. VGG-tiny models are trained on CChar dataset in 30 epochs with supervised labels and are expected to classify the categories of strokes. During the training, the stroke order of each character is not reserved and the order of images is shuffled in the batch.

MAE MAE models in He et al. (2022) are trained on ImageNet-1K. To adapt our small dataset, we modify some hyperparameters and use a ViT-tiny backbone to prevent overfitting. Compared with CNN, ViT lacks inductive bias and may perform badly when the dataset is small. The input image is resized to 128x128. The patch size is 16. The embedding dimension is 256. There are 4 layers in both the encoder and decoder. The number of attention heads is 4. In

Dataset	Dim	Max hierarchy	Object	Real	Instances
Ellipse world (Culp et al., 2022)	2D	3	multiple	synthesized	505,000
Exercise (Xue, Wu, Bouman, & Freeman, 2016)	2D	3	single	real	50,500
Geo (Xu et al., 2018)	2D	2	multiple	synthesized	110,000
PartNet (Mo et al., 2019)	3D	7	single	synthesized	26,671
PTR (Hong, Yi, Tenenbaum, Torralba, & Gan, 2021)	3D	3	multiple	synthesized	70,000
CChar (ours)	2D	6	single	synthesized	6,840

Table 1: Comparison between CChar and other part-whole datasets.

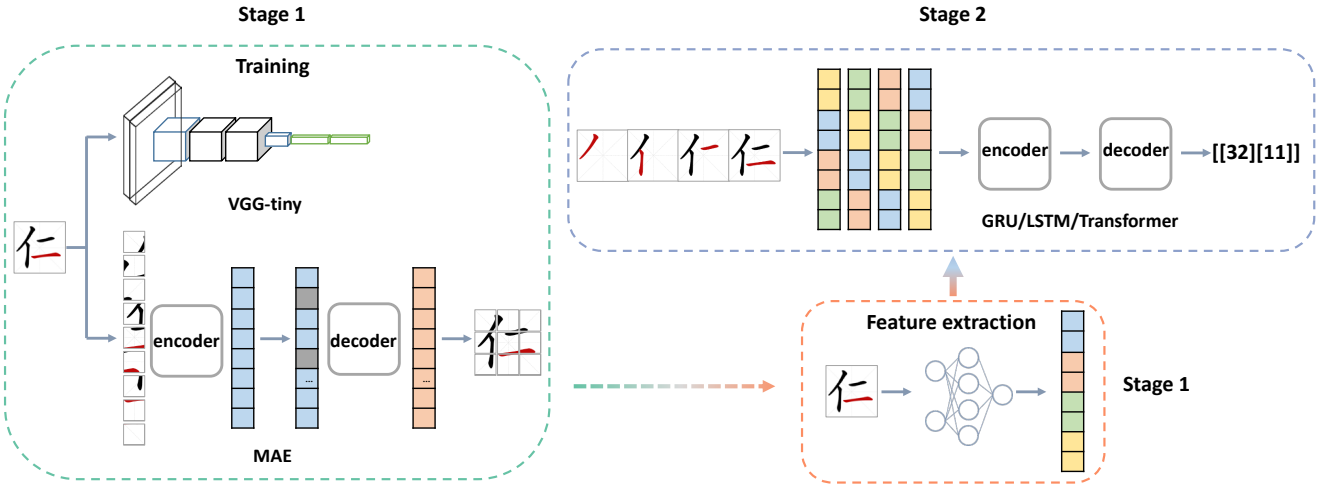

Figure 2: Two-stage experiments.

experiment 2, we build two types of visual feature extractors. First, we pre-train MAE models on CChar train and validation images in 200 epochs, with different mask ratios (0.1, 0.3, 0.5, 0.7, 0.9). Then, we load the pre-trained weights and fine-tune with supervised labels in 50 epochs. The batch size is 64. For both pre-trained and fine-tuned models, we only take the encoder as feature extractors, and use a '[CLS]' token vector with 256 dimensions, the first token vector of the encoder output, to represent the whole image. Thus, we get self-supervised features and label-guided features.

We use LSTM, GRU, and Transformer in an encoder-decoder architecture to generate sequences. In each experiment, we use the Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ and $\varepsilon = 10^{-8}$. The learning rate is 0.0005. The batch size is 32. The loss function is cross entropy. The random seed is 42. Please see the appendix for other hyperparameters of LSTM, GRU and Transformer in different experiments.

Evaluation

Visual Feature Extraction Models are evaluated by accuracy in the task of stroke classification.

Generating the Sequence of Strokes The model-generated sequences are compared against standard sequences with human annotations. It might seem that syntactic parsing is a similar task, and metrics for evaluating it could be transferred to our task. However, there are several key differences between our task and syntactic parsing tasks. First, characters will not change after text-to-text parsing, while predicting stroke labels from sequential visual inputs is more complex. In addition to misplacing the brackets, models in this task may also predict the wrong labels of images. Secondly, it is unnecessary to assign labels for larger units at high levels (e.g. types of phrase structures) in our task. Therefore, we use BLEU score (Lin & Och, 2004) and normalized Levenshtein ratio (Levenshtein et al., 1966) for automatic evaluation.

BLEU is a widely used metric in machine translation, and is calculated based on measuring the effective overlap between a reference sentence and a predicted sentence. It is a corpus-level metric for comparing different models. Here we use the BLEU-4 setting for calculation.

Levenshtein ratio, a normalized edit distance, is also adopted here. The formula is illustrated as Equation 1. The distance of insertion, deletion and substitution are relatively 1, 1, and 2. For example, when the predicted sequence is '55' and true sequence is '[5]', $ratio = 1\text{–}(1+2)/(2+3) = 0.4$.

$$Ratio(a,b) = 1 - \frac{Distance(a,b)}{|a|+|b|} \quad (1)$$

Primitive components are the smallest components that

cannot be decomposed, and the accuracy of primitive components could indirectly reflect whether models work in a robust compositional way. For each character, given the predicted sequence, extract "[number]" patterns and match them with true primitive components. The accuracy of primitive components equals the number of correctly matching primitive components divided by the total number of true primitive components. For example, if the predicted sequence is "[[252][132]]" and the true sequence is "[[23][[1][252]]]", the correctly matched primitive component is "[252]", and $accuracy = 1/3 = 0.33$. The accuracy of primitive components only counts the number of primitive components that could be matched, and does not consider the length and position of primitive components or their combinations. It is an indicator of local compositionality from stroke level to primitive component level, no matter representations are built in a top-down or bottom-up way.

Results and Discussion

Capacity of NNs to Learn Part-whole Hierarchies

In experiment 1, the accuracy of VGG-tiny is 99.71%, and we use it to extract visual features with 512 dimensions for each image. We use BLEU scores, Levenshtein ratios, and the percentage of exact match to measure the capacity of neural networks to learn the hierarchical structures. For tasks to predict target with hierarchy, we calculate the Levenshtein ratios and the percentage of exact match in two different settings: one is based on target sequences with brackets, and the other one is based on target sequences without brackets (see Table 3). The results in the group without brackets can provide the accuracy of predicting sequential images and serve as a reference. By comparing the two settings, we can know the difficulty of learning hierarchical structures and disentangle the source of errors.

The results in Table 2 and Table 3 show that neural networks could recognize a part of the components above strokes, and predict at least 30% of test data with complete accuracy. Transformers show better performance in capturing long dependency and predicting target sequences with hierarchies than GRU and LSTM. Transformers' ability to model long dependency is supported by two aspects of evidence (see the Appendix): (1) the visualization of attention illustrates how Transformers recognize the boundaries of primitive and compound components; and (2) Transformers make much fewer errors on unpaired brackets.

The results in Table 4 imply that these models do not have robust compositional reasoning, although Transformers are better than RNNs in predicting local components.

Model	BLEU-4 (hier)	BLEU-4
GRU	74.21	91.78
LSTM	79.79	93.93
Transformer	92.41	92.49

Table 2: The BLEU-4 scores of different models.

Conceptual Guidance in Visual Feature Extraction

In experiment 2, the accuracy of fine-tuned MAE models is 99.57%, 99.53%, 99.55%, 99.60%, 99.64% when mask ratio is 0.1, 0.3, 0.5, 0.7, and 0.9 respectively.

The results in Table 5 show that there is a difference of 30 to 40 points between two types of visual features. Models with self-supervised features have limited ability to generate hierarchical sequences, while models with fine-tuned features prove much better performance. This implies conceptual guidance is important for extracting visual features and learning part-whole hierarchies.

GLOM (Hinton, 2022) and Agglomerator (Garau et al., 2022) try to tackle this issue by introducing intricate architectures to improve clustering from the pixel level. However, this experiment shows that conceptual guidance of visual features is also very important for learning part-whole hierarchies, because the boundary between perception and cognition is not clear-cut. In fact, it proves unexpectedly effective that using priors of linguistic structures to improve model performance in various computer vision tasks (Narasimhan, Rohrbach, & Darrell, 2021; Rao et al., 2022; El Banani, Desai, & Johnson, 2023; Deng et al., 2023; Michel et al., 2024).

Relationship between Depth of Part-whole Hierarchies and Model Performance

In experiment 3, the accuracy of VGG-tiny is 99.74%, and we use it to extract visual features with 256 dimensions for each image.

We use simple linear regression to test if the depth of hierarchical structures significantly predicts the Levenshtein ratios of different models. The results show that the depth of hierarchies could significantly predict the Levenshtein ratios of RNNs and Transformers (see detailed regression coefficients in the Appendix). The performance of RNNs (GRU and LSTM) decreases linearly as the hierarchical structures deepen. However, the performance of Transformer models is better with an increasing depth of hierarchies, which may be explained by that attention mechanism could capture longer dependencies (see Figure 3).

RNNs outperform Transformers for characters with shallow hierarchies, which may be attributed to the small size of dataset and the inductive biases of RNNs and Transformers. The dataset in experiment 3 is a collection of subsets of CChar, and each subset only counts about 20% of CChar (see detailed distribution of hierarchical levels in the Appendix). Transformers lack local inductive biases and cannot show its advantage when they are trained on very short sequences with much fewer components. In previous studies on the inductive bias of RNNs, it is found that GRUs and LSTMs have inductive biases toward low frequent patterns (Ishii, Ueda, & Miyao, 2023). In addition, LSTM-based seq2seq learners can learn and generalize math operations from a single training example (Kharitonov & Chaabouni, 2020).

It should be pointed out that the longest sequence with hierarchical structures in the CChar dataset only has 56 tokens.

Model	Target with hierarchy				Target	
	L-ratio(hier)	Match(hier)	L-ratio	Match	L-ratio	Match
GRU	0.8947	29.43	0.8996	34.41	0.9777	76.80
LSTM	0.9144	41.13	0.9141	44.74	0.9849	84.41
Transformer	0.9661	64.91	0.9737	72.71	0.9748	71.73

Table 3: The Levenshtein ratios and the percentages of exact match predicted by different models. The "L-ratio (hier)" metric calculates the average Levenshtein ratio between predicted sequences with brackets and true sequences with brackets, while the "L-ratio" metric measures the average similarity between predicted sequences after removing brackets and true sequences without brackets. Similarly, the "Match (hier)" metric calculates the percentage of a total match between predicted sequences with brackets and true sequences with brackets, while the "Match" metric calculates the percentage of a total match between predicted sequences after removing brackets and true sequences without brackets.

Model	Avg accuracy
GRU	57.86
LSTM	62.25
Transformer	83.72

Table 4: The average accuracy of primitive components predicted by different models.

Model	Mask ratio	BLEU-4	
		pre-trained	fine-tuned
GRU	0.1	35.64	70.49
	0.3	39.62	72.41
	0.5	44.33	70.70
	0.7	44.85	68.74
	0.9	41.45	70.71
LSTM	0.1	32.83	68.32
	0.3	38.12	71.52
	0.5	44.43	68.73
	0.7	45.76	68.65
	0.9	44.14	68.30
Transformer	0.1	33.30	76.86
	0.3	31.37	74.29
	0.5	51.47	64.08
	0.7	52.13	77.18
	0.9	51.63	73.39

Table 5: The BLEU-4 scores of different models with self-supervised or fine-tuned visual features.

The length of sequences is very limited, and only characters with 1 to 4 levels of hierarchies are selected in experiment 3, which could undermine the conclusion.

Conclusion

In this paper, we introduce a diagnostic dataset to measure the capacity of neural networks in learning part-whole hierarchies. RNNs and Transformers can recognize a portion of upper-level components and learn part-whole hierarchies. However, this does not imply that these models have robust compositional reasoning. We also compared two types of visual features to understand the role of stroke labels in guiding visual feature extraction. Supervised features in generating hierarchical sequences prove to be more effective than self-supervised features, which emphasizes the importance of conceptual guidance in learning part-whole hierarchies. Lastly, the depth of hierarchies affects the Levenshtein ratios of RNNs and Transformers differently. As the hierarchical structures deepen, the performance of RNNs decreases, but Transformer models perform better.

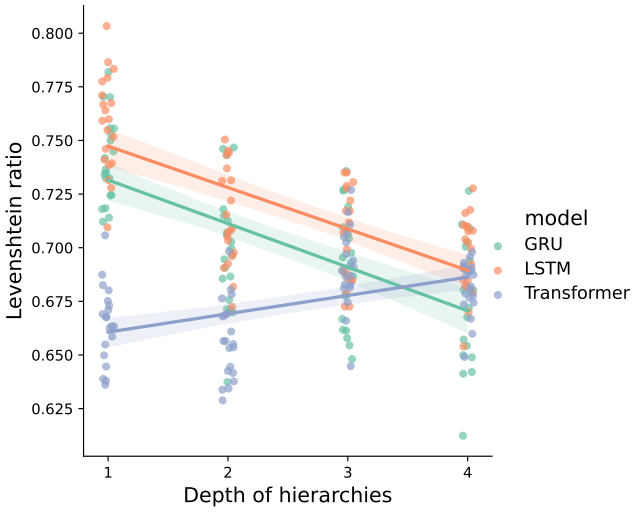

Figure 3: Relationship between depth of hierarchies and Levenshtein ratios in different models.

This study aims to analyze how different variables affect neural networks in learning part-whole hierarchies, and heavily relies on the specific structure of Chinese characters. There are still many challenges in expanding to more open-ended naturalistic visual environments and building efficient architectures to represent part-whole hierarchies. Following insights from previous experiments, we expect there will be neural architectures with conceptual guidance to play a greater role in representing part-whole hierarchies. To effectively integrate conceptual guidance, hybrid neuro-symbolic architectures or cognitive-inspired multimodal learning are to be explored in the future.

References

Anderson, R. C., Ku, Y.-M., Li, W., Chen, X., Wu, X., & Shu, H. (2013). Learning to see the patterns in chinese characters. *Scientific Studies of Reading*, *17*(1), 41–56.

Cho, K., van Merriënboer, B., Gulçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1724–1734).

Culp, L., Sabour, S., & Hinton, G. E. (2022). Testing glom's ability to infer wholes from ambiguous parts. *arXiv preprint arXiv:2211.16564*.

Deng, C., Jiang, C., Qi, C. R., Yan, X., Zhou, Y., Guibas, L., ... others (2023). Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 20637–20647).

El Banani, M., Desai, K., & Johnson, J. (2023). Learning visual representations via language-guided sampling. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 19208–19220).

Garau, N., Bisagno, N., Sambugaro, Z., & Conci, N. (2022). Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13689–13698).

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 16000–16009).

Hinton, G. E. (1990). Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, *46*(1-2), 47–75.

Hinton, G. E. (2022). How to represent part-whole hierarchies in a neural network. *Neural Computation*, 1–40.

Hinton, G. E., Sabour, S., & Frosst, N. (2018). Matrix capsules with em routing. In *International conference on learning representations.*

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hong, Y., Yi, L., Tenenbaum, J., Torralba, A., & Gan, C. (2021). Ptr: A benchmark for part-based conceptual, relational, and physical reasoning. *Advances in Neural Information Processing Systems*, *34*, 17427–17440.

Ishii, T., Ueda, R., & Miyao, Y. (2023). Empirical analysis of the inductive bias of recurrent neural networks by discrete fourier transform of output sequences. *arXiv preprint arXiv:2305.09178*.

Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M., & Fei-Fei, L. (2015). Image retrieval using scene graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3668–3678).

Kharitonov, E., & Chaabouni, R. (2020). What they do when in doubt: a study of inductive biases in seq2seq learners. In *International conference on learning representations.*

LeCun, Y., Jackel, L., Bottou, L., Brunot, A., Cortes, C., Denker, J., ... others (1995). Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks* (Vol. 60, pp. 53–60).

Levenshtein, V. I., et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, pp. 707–710).

Li, H., Zhu, G., Zhang, L., Jiang, Y., Dang, Y., Hou, H., ... Bennamoun, M. (2024). Scene graph generation: A comprehensive survey. *Neurocomputing*, *566*, 127052.

Li, X., Chen, L., Shao, J., Xiao, S., Zhang, S., & Xiao, J. (2022). Rethinking the evaluation of unbiased scene graph generation. In *33rd british machine vision conference 2022, BMVC 2022, london, uk, november 21-24, 2022*. BMVA Press.

Lin, C.-Y., & Och, F. J. (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd annual meeting of the association for computational linguistics (acl-04)* (pp. 605–612).

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information.* Henry Holt and Co., Inc.

Marr, D., & Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, *200*(1140), 269–294.

Michel, O., Bhattad, A., VanderBilt, E., Krishna, R., Kembhavi, A., & Gupta, T. (2024). Object 3dit: Language-guided 3d-aware image editing. *Advances in Neural Information Processing Systems*, *36*.

Ministry of Education of the People's Republic of China. (1997). 信息处理用*gb 13000.1* 字符集汉字部件规范*[chinese character component standard of gb 13000.1 character set for information processing].*

Ministry of Education of the People's Republic of China. (2009). 现代常用字部件及部件名称规范*[specification of common modern chinese character components and component names].*

Ministry of Education of the People's Republic of China. (2020). 通用规范汉字笔顺规范*[stroke orders of the commonly used standard chinese characters].*

Mo, K., Zhu, S., Chang, A. X., Yi, L., Tripathi, S., Guibas, L. J., & Su, H. (2019). Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 909–918).

Narasimhan, M., Rohrbach, A., & Darrell, T. (2021). Clip-it! language-guided video summarization. *Advances in neural information processing systems*, *34*, 13988–14000.

Palmer, S. E. (1977). Hierarchical structure in perceptual representation. *Cognitive psychology*, *9*(4), 441–474.

Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., ...

Lu, J. (2022). Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 18082–18091).

Riesenhuber, M., & Dayan, P. (1996). Neural models for part-whole hierarchies. *Advances in neural information processing systems*, *9*.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, *30*.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wertheimer, M. (1938). Laws of organization in perceptual forms. In W. D. Ellis (Ed.), *A source book of gestalt psychology.* (pp. 71–88). Kegan Paul, Trench, Trubner & Company.

Xing, H. (2007). 现代汉字特征分析与计算研究*[feature analysis and computation of chinese characters]*. Commerical Press.

Xu, Z., Liu, Z., Sun, C., Murphy, K., Freeman, W. T., Tenenbaum, J. B., & Wu, J. (2018). Unsupervised discovery of parts, structure, and dynamics. In *International conference on learning representations.*

Xue, T., Wu, J., Bouman, K., & Freeman, B. (2016). Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *Advances in neural information processing systems*, *29*.

Zhou, Y. (1992). 中国语文纵横谈*[a discussion on chinese language]*. People's Education Press.