The Impact of Visual Information in Chinese Characters: Evaluating Large Models' Ability to Recognize and Utilize Radicals

Xiaofeng Wu[1] **Karl Stratos**[2] **Wei Xu**[1]
[1]Georgia Institute of Technology [2]Rutgers University
xwu414@gatech.edu, karl.stratos@rutgers.edu, wei.xu@cc.gatech.edu

Abstract

The glyphic writing system of Chinese incorporates information-rich visual features in each character, such as radicals that provide hints about meaning or pronunciation. However, there has been no investigation into whether contemporary Large Language Models (LLMs) and Vision-Language Models (VLMs) can harness these sub-character features in Chinese through prompting. In this study, we establish a benchmark[1] to evaluate LLMs' and VLMs' understanding of visual elements in Chinese characters, including radicals, composition structures, strokes, and stroke counts. Our results reveal that models surprisingly exhibit some, but still limited, knowledge of the visual information, regardless of whether images of characters are provided. To incite models' ability to use radicals, we further experiment with incorporating radicals into the prompts for Chinese language processing (CLP) tasks. We observe consistent improvement in Part-Of-Speech tagging when providing additional information about radicals, suggesting the potential to enhance CLP by integrating sub-character information.

1 Introduction

Visual information embedded in Chinese characters is important, as most Chinese characters convey a meaning equivalent to an entire word in English with a complex glyphic structure. Multiple writing strokes form the *radicals*,[2] which often carry information about semantic meaning and pronunciation; the radicals are then visually combined to form Chinese characters. When encountering unfamiliar characters, Chinese speakers rely on semantic and phonetic hints from radicals, much like how English speakers use sub-words such as prefixes or suffixes, to approximate the meaning and pronunciation of unfamiliar words. For example, the Chinese character "花" (meaning "flower"; pronounced as "huā") in Figure 1 has "艹" (meaning "herbal") on the top, contributing to its semantic meaning, and "化" (pronounces as "huà") on the bottom, indicating its pronunciation. By utilizing the radical information, one can infer that "花" is related to herbs and has a pronunciation similar to "huà" without prior knowledge of the character.

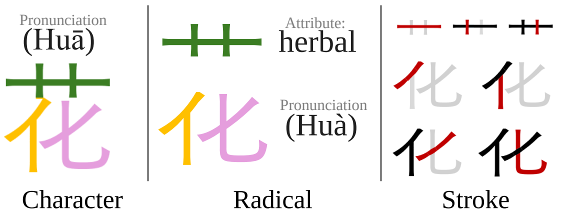

Figure 1: Chinese character "花" displayed at the character, radical, and stroke levels from left to right. Different radicals are shown in **green**, **yellow**, and **pink** colors, while the writing order of the strokes is indicated by **red** (current), **gray** (upcoming), and **black** (completed).

Although radicals contain rich information, they have received little attention in digital text processing. Contemporary typefaces treat Chinese characters, radicals, and strokes as indivisible units, disregarding their compositional relationships. Consequently, most language models follow this approach, under-utilizing the rich visual and semantic information embedded in Chinese characters. While limited prior works (Sun et al., 2021; Si et al., 2021; Stratos, 2017) have attempted to address this issue by incorporating visual embeddings, such as strokes or font images, into smaller-scale models (i.e., BERT or Neural MT), there remains a lack of research investigating whether these visual features can be recognized and utilized by models in light of the significant advancements in LLMs and VLMs, especially in inference methods (e.g., prompting).

To determine whether pre-trained LLMs recognize or can acquire the visual knowledge embedded

[1]Our benchmark can be accessed through GitHub.

[2]A comprehensive definition of Chinese radicals can be found on Wikipedia: https://en.wikipedia.org/wiki/Chinese_character_radicals. For simplicity, this paper refers to any large components within a character as radicals.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 331–350
April 29 - May 4, 2025

in Chinese characters, we establish a benchmark by collecting over 14,000 Chinese characters from the Chinese, Japanese, and Korean (CJK) Unified Ideographs,[3] considering four visual elements: radicals, composition structures, strokes, and stroke counts. As shown in Figure 3, the *composition structure* refers to the visual arrangement of a character's radicals (e.g., top-to-bottom or left-to-right). *Stroke composition* provides an essential way to represent not typable radicals; As shown in Figure 2, some radicals cannot be typed using standard input methods but can still be accurately depicted through stroke compositions (see more details in Appendix D.2). Lastly, *Stroke count* offers a measure for Chinese characters' visual complexity, similar to word length in English.

We conduct experiments on four tasks: structure recognition, radical recognition, stroke count identification, and stroke identification. We evaluated a series of LLMs and VLMs (e.g., GPT-4, Gemini-1.5, Ernie-4, Aya-command, QWen-7B, etc.), and found that all models possess some visual knowledge of Chinese characters, even without image inputs; however, it is only to a limited extent. In particular, the models tend to perform well in recognizing the first radical of a Chinese character, such as "艹" (herbal) in "花"(flower), but often fail with subsequent ones. We also demonstrate that the pixel-based encoder PIXEL (Razzhigaev et al., 2022) has the ability to capture structural information effectively after fine-tuning. As a language model pre-trained only on an English corpus, PIXEL achieved an F1 score of 84.57, significantly higher than the second-best score of 54.30 achieved by Ernie-4[4] and 23.29 by GPT-4 when provided with images of characters, indicating its potential for CLP as it naturally captures visual information.

We further investigate whether models can utilize radicals to improve performance on understanding tasks (e.g., POS tagging and NER) by prompting them to use radicals when encountering unfamiliar words. Our experiments show that radical information yields promising results in downstream tasks, particularly in POS tagging. We observe consistent improvement across models and datasets when the information about radicals is provided. Notably, Ernie-Lite-8K's POS tagging F1 score on GSD(Qi and Yasuoka, 2023) decreases by 2.1

[3] The CJK Unified Ideographs refers to a set of Chinese characters used across Chinese, Japanese, and Korean languages to standardize and unify the use of characters.

[4] Released by Chinese company Baidu that rivals GPT-4.

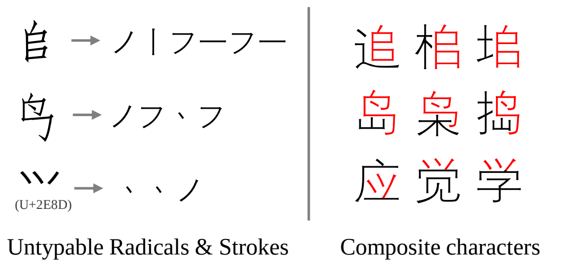

Figure 2: Example images of untypable radicals alongside example characters and their corresponding Unicode values, if available.

points when recognizing radicals on its own, but increases by 5.7 points when provided with correct radicals. For Name Entity Recognition (NER), we also observe an improvement in three out of six models. Analyzing the cases where incorporating radical degrades the model performance, we see that incorrect answers often occur when the model fails to identify unfamiliar words and bypasses the radical information process, indicating the decrease is likely due to long prompts. When evaluating only sentences where the model detects unfamiliar words, performance on NER generally improves. Our work demonstrates that models possess the ability to recognize and utilize radical information only to a certain limit, suggesting that deeper integration, such as additional training on radicals or improvement in the Chinese digital system to incorporate radicals, could unlock further potential.

2 Related Work

Chinese Character Decomposition in Computer Vision. The task of decomposing Chinese characters into constituent components has majorly been studied in the field of computer vision. Research within this domain, such as the studies by Ma et al. (2021), Xia (1994), and Liu et al. (2021), has explored analogous challenges. The work by Zhang et al. (2018) employs a methodical approach by categorizing characters into structured types and further decomposing sub-components according to their spatial arrangements—akin to the layered structural analysis that we adopt in this paper.

Chinese Decomposition Datasets. A comprehensive dataset (Kawabata et al., 2018) that offers decompositions for the CJK Unified Ideographs. Although this collection overlaps with our dataset, it does not cite any authoritative sources for its data. This omission leads to ambiguity due to multiple decomposition sequences for individual characters.

Our approach utilizes sources from authoritative Chinese dictionaries, such as the Kangxi Dictionary (康熙字典) and the Xinhua Dictionary (新华字典), ensuring a validated framework for visual information. Additionally, our dataset contains standard stroke orders for all 14,648 characters, which the aforementioned dataset lacks.

Glyphic Embedding Strategies in LMs. Few prior works have utilized the idea of adding additional input embedding with Chinese visual features. For instance, (Shi et al., 2015) attempted to add radical embedding in the pre-transformer era. (Sun et al., 2021) introduced font images into embedding, and (Si et al., 2021) experimented with stroke among other glyph-based embeddings such as Cangjie[5] (仓颉). Another interesting approach is PIXEL (Razzhigaev et al., 2022), which uses a pixel-based encoder to transform input into images. Our experiment results in §4 highlights the potential of pixel-based language models in CLP.

3 Chinese Character Dataset (CCD)

To evaluate contemporary LLMs and VLMs' proficiency with visual information in Chinese characters, we compile a dataset using characters from CJK Unified Ideographs with visual features collected from the digitized Kangxi Dictionary and Xinhua Dictionary. Our dataset includes 14,648 Chinese characters and details their corresponding radicals, strokes, and stroke count. A subset of 4,651 Simplified Chinese characters also contains structural composition information. The detailed statistics are provided in Table 1 with three tiers of character frequency based on the Table of General Standard Chinese Characters (通用规范汉字表).[6]

Structure of Chinese Characters. According to the digitized Kangxi Dictionary,[7] we categorize 4651 simplified Chinese characters into eight major structural arrangements, with examples of each structure illustrated in Figure 3: top-bottom, left-right, top-mid-bottom, left-mid-right, wrapping, inlay, triple-stack, as well as single structure, which refers to characters that cannot be further segmented. The structure of Chinese characters can be rather complex. For example, the character "花", shown in Figure 1, has a top-bottom structure, consisting of "艹" and "化". Furthermore, "化" exhibits a left-right structure which can be further decomposed into "亻" and "七." For the sake of clarity, we categorize all characters based on their top-level structure (i.e., top-bottom for "花").

Radicals of Chinese Characters. Based on the aforementioned structural composition, we collect radicals by decomposing each Chinese character into meaningful components that have corresponding Unicode representations. The radicals are then ordered according to specific rules: from top to bottom, left to right, outside to inside, and main parts before inlay parts, as illustrated in Figure 3. We avoid decomposition for simple Chinese characters that would reduce to meaningless strokes. For example, while the character "八" could be segmented as a left-right structure, we classify it as a single structure with only one radical to prevent it from being reduced to meaningless strokes.

The initial collection of radicals was performed using APISpace's Chinese character segmentation API,[8] which analyzes characters' stroke compositions and extracts radicals based on the optimal subsequences of strokes. After the automated annotation, we conducted a thorough two-round manual review to ensure accuracy. More than 1,000 characters required manual adjustments due to missing or incorrect radicals; in addition, another 500 adjustments were made to prevent unnecessary reduction of characters into strokes by one of the native Chinese-speaking authors. Four native Chinese speakers further reviewed the entire dataset and collectively corrected for about 2% annotations. Further details are provided in the Appendix B.

Stroke Composition of Chinese Characters. Stroke composition refers to the sequence of writing order of a Chinese character's strokes. Chinese dictionaries categorize all Chinese strokes into five basic stroke types: "一", "丨", "ノ", "丶", and "フ', which our dataset adopts. We first utilized the Xinhua Dictionary API to automatically annotate. For characters not found in the API, we attempted to concatenate the stroke composition of their radicals in order. We then manually reviewed all stroke compositions to ensure accuracy.

Stroke Count of Chinese Characters. The number of strokes required to write a Chinese character is also present in our dataset, which offers a measure of word complexity. Unlike the alpha-

[5] glyph-based Chinese character input method.
[6] https://zh.wikisource.org/wiki/通用规范汉字表
[7] https://www.kangxizidian.com/
[8] API documentation can be accessed at https://www.apispace.com/eolink/api/dfsdfsfsf/apiDocument

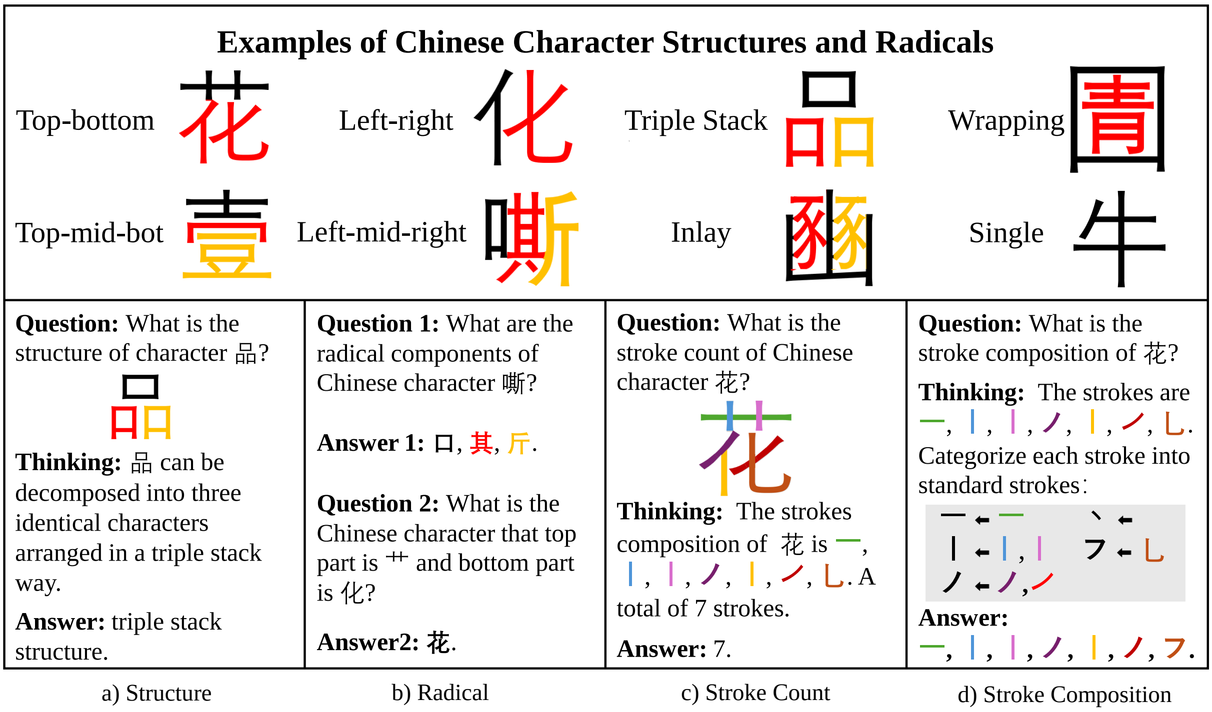

Figure 3: Examples of composition structures with radicals in order of **black**, **red**, **yellow** and four different tasks.

Character Level	**Statistic**
# of total Chinese characters	14,648
- Commonly used (tier 1):	3,500 (24.1%)
- Less commonly used (tier 2):	3,000 (20.6%)
- Terminology used (tier 3):	1,605 (11.0%)
- Hardly ever used (others):	5,543 (37.8%)
- w/ structure information:	4,651 (31.8%)
Radical Level	
# of unique radical	2,132
# of single-appearance radical	692
Stroke Level	
# of unique stroke composition	13,740
# of strokes per character (μ)	11.51
# of strokes per character (σ)	3.92
# of strokes per character (min)	1
# of strokes per character (max)	39

Table 1: Key statistics of our Chinese character dataset.

betic writing system, where word length can hint at complexity, Chinese characters occupy a uniform length of one, making stroke count a valuable indicator of intricacy. The statistics for strokes are provided in Table 1 with illustrations in Figure 3.

4 Evaluation on Recognizing Visual Information of Chinese Characters

To evaluate whether language models contain or can learn the visual information embedded in Chinese characters, we establish a benchmark by setting up a series of tasks (see examples of each task in Figure 3) derived from our dataset.

4.1 Chinese Character Tasks

Structure Recognition of Chinese Characters. We assess LLMs and VLMs' proficiency in identifying the correct structural arrangements of Chinese characters using a multiple-choice format. We present the character with eight structure types and evaluate the model's answer using the F1 score.

Radical Recognition of Chinese Characters. We evaluate LLMs and VLMs' ability to recognize radical information in two tasks: character-to-radical and radical-to-character. For the first task, models receive character and order guidelines and are prompted to identify their radicals in sequence. Performance is measured by the accuracy of each radical in order and the overall F1 score. For the second task, models are given radicals and their relative positions and asked to identify the correct characters, with accuracy as the metric.

Stroke Count Identification of Chinese Characters. We evaluate models' ability to identify the number of strokes required to write query characters with performance measured using Mean Absolute Error (MAE) and Mean Squared Error (MSE).

Stroke Decomposition of Chinese Characters. Similar to the radical-to-character task, we evaluate

Model	Structure		Radicals						Stroke Ct.		Stroke Composition				
	F1 ↑	H ↓	1st Acc	2nd Acc	3rd Acc	F1 ↑	H ↓	Acc ↑	MSE ↓	MAE ↓	1st Acc	2nd Acc	3rd Acc	F1 ↑	H ↓
Vision Language Models (VLMs)															
Ernie-4	54.30	-	41.03	34.21	12.50	41.67	-	71.79	12.54	1.78	53.85	35.90	47.37	30.90	-
Kimi-v1	45.60	-	36.73	19.15	0.00	32.93	-	42.86	15.32	2.68	30.61	26.53	16.67	20.70	-
Claude-3	23.70	0.54	8.80	0.61	0.00	2.44	1.09	57.30	5.93	1.22	15.40	19.60	26.80	19.62	1.22
Gemini-1.5	27.15	0.36	3.00	0.41	0.00	1.53	1.12	27.08	8.83	2.28	29.60	16.80	22.00	22.04	1.00
GPT-4	23.28	0.46	10.20	0.41	0.00	9.22	0.95	24.18	7.96	1.64	24.00	19.60	23.80	21.96	1.34
GPT-4o	26.66	-	6.00	0.40	0.00	6.62	0.74	67.39	10.35	1.72	46.40	31.40	34.80	30.85	0.70
Close-Sourced Models (LLMs)															
Ernie-Lite-8K	7.19	0.76	18.92	3.52	0.13	11.99	1.89	3.72	44.53	5.34	29.30	23.28	20.78	23.34	1.11
Kimi-v1	24.51	0.83	7.24	0.33	0.00	1.10	0.72	50.16	19.05	3.12	33.12	21.56	19.72	22.99	1.07
Aya-command	12.56	0.16	35.72	2.16	0.26	20.13	0.73	5.65	13.20	2.79	28.24	23.48	19.44	21.43	0.37
Claude-3	23.70	0.54	70.02	5.64	0.43	45.57	1.09	40.40	7.78	1.32	28.64	19.02	31.19	22.91	0.88
Gemini-1.5	23.04	0.56	4.20	0.04	0.38	1.37	1.16	11.26	13.23	2.76	26.66	24.52	15.14	20.24	0.81
Few-shot GPT-3.5	22.82	0.88	54.14	7.37	0.30	34.60	1.21	23.12	7.96	1.65	27.86	22.70	30.23	25.62	1.13
Zero-shot GPT-3.5	15.43	0.69	52.14	4.33	0.20	31.66	1.30	17.45	10.80	2.17	30.70	21.92	26.97	25.09	0.98
Fine-tune GPT-3.5	27.14	0.33	4.12	0.00	0.00	1.23	1.11	71.66	7.36	1.46	47.50	44.58	32.67	28.64	1.08
CoT GPT-3.5	38.08	1.25	5.24	0.16	0.11	1.63	1.05	24.41	8.93	1.92	31.06	22.22	26.85	25.60	0.83
Few-shot GPT-4	45.28	0.48	58.44	6.45	0.31	41.66	0.84	38.01	7.96	1.65	24.18	18.22	21.90	20.87	1.37
Zero-shot GPT-4	35.40	0.54	57.86	6.28	0.20	41.42	0.88	38.76	12.17	1.99	27.04	21.16	21.99	22.18	1.21
Open-Sourced Models (LLMs)															
Baichuan-13B	11.17	0.88	33.20	2.05	0.60	22.62	1.20	13.67	32.70	4.31	27.68	21.42	15.92	22.74	1.56
ChatGLM-6B	10.30	0.68	6.94	0.50	0.00	6.33	1.35	1.38	29.68	4.25	26.88	12.60	12.43	27.28	0.96
Chinese-LLaMA-7B	5.13	0.97	9.26	0.64	0.17	6.32	1.92	0.32	15.83	3.00	26.26	24.86	13.42	22.32	0.93
InternLM-7B	9.68	1.05	12.08	0.34	0.05	8.89	1.50	0.00	45.38	5.50	28.82	24.66	13.38	22.01	0.95
Yi-6B	8.86	0.70	14.18	1.05	0.21	12.14	1.40	0.32	29.49	4.24	28.56	22.40	7.76	24.17	0.85
Bloom-7B	9.81	0.96	3.48	0.54	0.04	4.15	1.70	0.00	46.76	4.05	27.92	24.96	14.47	23.19	0.87
Qwen-7B	5.25	1.16	17.30	0.85	0.23	12.41	1.50	1.59	34.16	4.62	25.02	20.20	21.92	23.30	1.30
Qwen-2-7B	6.76	1.50	15.42	0.68	0.22	10.70	1.75	0.42	44.48	5.39	23.16	18.50	21.54	22.68	1.40
Orion-14B	9.00	1.04	5.27	0.18	0.76	9.46	1.11	3.39	31.45	4.45	28.40	22.82	19.38	24.81	0.90
Fine-tune PIXEL	84.57	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Majority Baseline	52.86	-	5.61	0.97	7.55	0.00	-	0.00	15.72	3.12	31.83	30.34	41.61	0.00	-

Table 2: Model performance on Chinese character visuals on four different tasks (§4.1). **H**: Entropy, : Chinese-English bilingual models. The top scores for each type of models (VLM/close-sourced LLM/open-sourced LLM) and all models are highlighted in blue and green, respectively.

LLMs and VLMs' ability to identify the sequence of strokes required to write the query character. Performance is measured by the accuracy of each stroke in order and the overall F1 score.

4.2 Experimental Setup

We assess the visual information of Chinese characters using multilingual, bilingual, and open-source LLMs and VLMs. Multilingual LLMs include Aya-command (Üstün et al., 2024), Claude-3 (Anthropic, 2024), Gemini-1.5, GPT-3.5 Turbo (OpenAI, 2024a), and GPT-4 (OpenAI, 2023). Chinese-English bilingual LLMs include ERNIE-Lite (Baidu, 2024a), Kimi-v1 (MoonshotAI, 2024), and open-source LLMs such as Baichuan-13B (BaichuanInc, 2024), BLOOM-7B (BigScience, 2024), ChatGLM-6B (Zeng et al., 2023), Chinese-LLaMA-7B (HFL, 2024), InternLM-7B (InternLM, 2024), Orion-14B (Chen et al., 2024), Qwen-7B (Bai et al., 2023), Qwen-2-72B, and Yi-6B (01.AI et al., 2024). We also evaluate VLMs by providing images of query characters in the Microsoft YaHei[9] font to vision-capable models, including Claude-3, Gemini-1.5, GPT-4, GPT-4o (OpenAI, 2024b), and bilingual models Ernie-4 (Baidu, 2024b) and Kimi-v1. Additionally, we assess the pixel-based encoder model, PIXEL (Rust et al., 2023). Since PIXEL is limited to specific tasks such as span-based QA, it is evaluated only on the multiple-choice structure recognition task after fine-tuning. To explore the models' ability to learn Chinese visual information, we apply Chain-of-Thought (CoT) prompting

[9] YaHei is the default Chinese font in Microsoft Office.

and fine-tuning to GPT-3.5, and few-shot settings to both GPT-3.5 and GPT-4. The remaining models are evaluated in a zero-shot setting. We repeat the evaluation for each task five times to compute entropy, using it as an indicator of the models' confidence. Detailed prompting and fine-tuning procedures are provided in Appendix C.1.

4.3 Experimental Results

As shown in Table 2, the majority of models demonstrate only a vague understanding of Chinese character-visual tasks. Among the evaluated models, Chinese-English bilingual VLMs achieve the highest overall performance, effectively leveraging visual information from the images. Multilingual VLMs, however, perform similarly to their LLM counterparts, with both groups achieving better-than-random-guess results. The performance of vision-lacking LLMs suggests that they have likely encountered textual data discussing knowledge about radicals during pre-training as shown in Appendix C.4. In contrast, open-source LLMs often perform worse than random guesses.

Structure Recognition Task. Most models tend to struggle with the structural arrangement of Chinese characters, with F1 scores below 50%. A notable exception is PIXEL, which achieves an outstanding F1 score of 84.57. Despite being pre-trained solely on an English corpus (English Wikipedia and BookCorpus) and exposed to Chinese only during fine-tuning, PIXEL (Razzhigaev et al., 2022) demonstrates strong potential for Chinese language processing by naturally capturing visually embedded information. GPT-3.5 also saw a 75% performance increase from zero-shot to fine-tuning settings. To better understand the performance boost after applying learning methods, we further examine the impact of Chinese character encoding and potential mapping patterns between a character's structure and Unicode in Appendix D.

Radical Recognition Task. In the character-to-radical task, a clear trend emerges where model performance is the highest for the first component and sharply decreases for subsequent ones. For example, Claude-3 achieves an F1 score of 70.02 for the first component, but this drops to 5.64 for the second component and nearly zero for the third. This pattern suggests that models could possibly associate the meaning of the radical with the character, as the first radical often relates to the semantic attribute of the character, such as "艹" in "花."

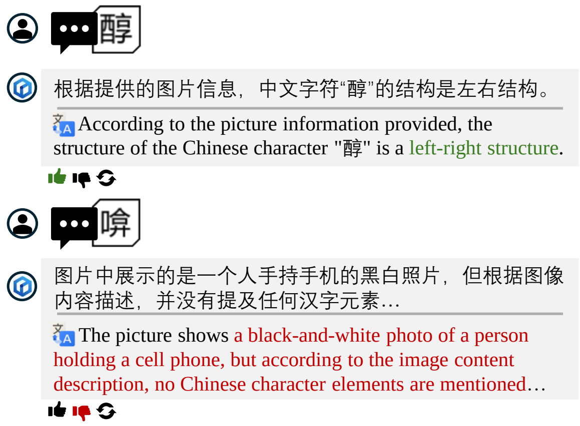

Figure 4: Example responses generated by Ernie-4 with vision, translated using Google Translate.

Interestingly, fine-tuning, CoT prompting, and the addition of vision in multilingual models drastically decreased performance of character-to-radical task to nearly zero. However, in the radical-to-character task, fine-tuning GPT-3.5 results in a significant improvement, achieving an F1 score of 71.66. The reason for this disparity, aside from the inherent difficulties between the two tasks, could be that fine-tuning and query characters for the radical-to-character task come from a subset of more common characters, for which we annotated structure information. In contrast, the character-to-radical task includes more complex and rarely used characters, potentially leading to catastrophic learning failure.

Stroke Decomposition and Count Tasks. Overall, most models struggle with identifying specific stroke compositions but demonstrate a general understanding of stroke count. For instance, Claude-3 achieves the lowest MSE among all LLMs at 7.78, significantly lower than the dataset's average stroke count of 11.51. Both tasks benefit from the applied learning methods. For stroke composition, fine-tuning yields the best results, while for stroke count, all methods show similar levels of improvement.

4.4 Error Analysis of Bilingual VLMs

To better understand the superior performance of bilingual VLMs, we conducted an error analysis on Ernie-4 and Kimi-v1 with vision. Both models exhibit common patterns of mistakes across several types of characters. First, complex and dense characters are often misrecognized as some other more frequently used characters that look similar. As the complexity of characters increases, the individual radicals become more compressed within

Models	Name Entity Recognition					
	People's Daily			Weibo		
	B	RP		B	RP	
Aya	52.00	54.61	(+2.6)	24.78	16.00	(-8.8)
Claude-3	68.54	70.48	(+1.9)	41.08	41.67	(+1.6)
ERNIE-Lite	7.55	21.05	(+14)	6.25	14.81	(+8.6)
GPT-3.5	55.74	55.96	(+0.2)	38.37	44.87	(+12)
GPT-4	65.23	65.96	(+0.7)	38.59	40.34	(+1.8)
QWen 72B	58.81	58.94	(+0.1)	29.39	33.17	(+3.8)

Table 3: Model performances for NER evaluated solely on samples where the model identifies unfamiliar words.

the available space, which can lead to misrecognition between similar characters, such as "黄" and "黃". Second, characters with only a single stroke difference are frequently mistaken for their more common variant. Third, when dealing with rare characters, Ernie-4 often fails to detect any character in the image, while Kimi-v1 may refuse to allow the user to send the prompt if it cannot extract the character. Additionally, both models sometimes mistake one radical for the entire character or confuse characters with black-and-white photos as shown in Figure 4. More examples of Kimi-v1 and Ernie-4 are provided in Appendix C.3.

5 Evaluation on Utilizing Radicals

We evaluated LLMs on downstream tasks, specifically examining performance differences when models are prompted vs. not prompted to use their knowledge of radicals to infer the meaning of unfamiliar words as illustrated in Figure 5.

5.1 Downstream NLP Tasks

Although LLMs may not match supervised LMs in traditional tasks, we chose these as key indicators of Chinese understanding to track improvements when models utilize radical information.

Part-of-Speech (POS) tagging. For the POS tagging task, we selected a 5-word window containing at most one punctuation mark and tasked the model with identifying the POS tag of the central word. The model's performance was evaluated using the F1 score. To cover a diverse range of sentences, we utilized three datasets: the GSD Simplified dataset (Qi and Yasuoka, 2023), the Parallel Universal Dependencies (PUD) dataset (McDonald et al., 2023), and a new dataset of 500 sentences from Tang Dynasty poems (written between 618 and 907 CE), processed using Classical Chinese RoBERTa (Yasuoka, 2023). We created and annotated this poetry

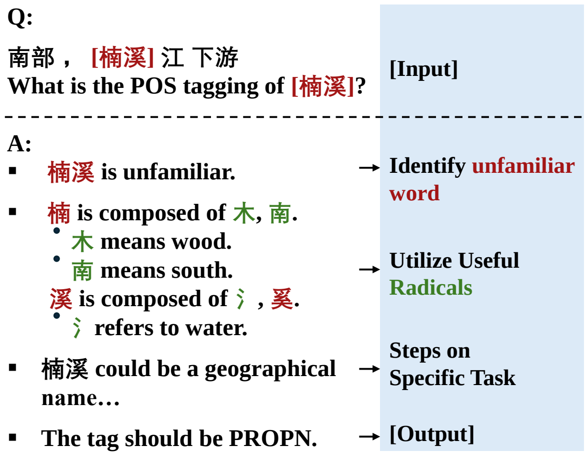

Figure 5: Example of model's answers for part-of-speech (POS) tagging with an unfamiliar Chinese word using radical prompting.

dataset to evaluate how well radicals perform in Classical Chinese (文言), which is characterized by compact and precise language where more information is carried by each character. Additionally, we conducted an ablation study with varied input window sizes, detailed in Appendix F.2.

Named Entity Recognition (NER). We tasked the model with identifying three types of entities — PER (person), LOC (location), and ORG (organization) — at the character level, using the BIO tagging standard. We excluded nominal entities provided in some datasets to streamline the analysis. The model's performance was evaluated using the F1 score. We use two distinct datasets for the NER task: the People's Daily dataset (Chen, 2023), which focuses on formal Chinese text, and the Weibo NER dataset (Peng and Dredze, 2015), which is oriented towards casual and online text.

Chinese Word Segmentation (CWS). CWS is a unique task in Chinese language processing. Distinguished from many other languages, Chinese does not use delimiters such as white spaces to separate words within sentences. Accurately segmenting text could be beneficial for many CLP applications. For this task, we provide sentences from the GSD and PUD and ask models to segment them into words. Answers are evaluated using the F1 score.

5.2 Experimental Setup

We select a series of LLMs for evaluation, including Aya-command, Claude-3, ERNIE-Lite-8K, GPT-3.5, GPT-4, and QWen-1.5 72B Chat. The models are instructed to return answers in JSON

Models	Part-Of-Speech Tagging														
	GSD					PUD					Poems				
	B	RP		RP (Oracle)		B	RP		RP (Oracle)		B	RP		RP (Oracle)	
Aya-command	68.86	68.91	(+0.1)	70.41	(+1.6)	73.87	77.21	(+3.3)	76.95	(+3.1)	64.71	64.72	(+0.0)	65.54	(+0.8)
Claude-3	69.37	70.68	(+1.3)	70.45	(+1.1)	69.37	70.45	(+1.1)	70.68	(+1.3)	65.53	66.20	(+0.7)	66.71	(+1.2)
ERNIE-Lite-8K	27.06	24.97	(-2.1)	32.73	(+5.7)	30.35	30.29	(-0.0)	41.29	(+10.9)	44.19	42.17	(-2.0)	49.07	(+4.9)
GPT-3.5	59.08	64.62	(+5.5)	67.56	(+8.5)	62.61	69.90	(+7.3)	73.46	(+10.9)	53.51	59.22	(+5.7)	61.39	(+7.9)
GPT-4	71.55	72.14	(+0.6)	72.95	(+1.4)	76.20	76.72	(+0.5)	77.35	(+1.2)	66.94	67.11	(+0.2)	67.57	(+0.6)
o1-mini	63.24	67.96	(+4.7)	64.31	(+1.1)	70.37	71.42	(+1.1)	75.49	(+4.1)	47.73	50.04	(+2.3)	49.00	(+1.3)
QWen-72B	62.20	65.38	(+3.2)	67.32	(+5.1)	60.09	64.70	(+4.6)	66.90	(+6.8)	55.63	57.78	(+2.2)	59.54	(+3.9)

Table 4: Model performances for POS tagging with baseline (B), radical prompting without golden components (RP), and radical prompting with oracle information (RP (Oracle)). Performance change relative to baseline is highlighted with green for increase and red for decrease.

Models	Name Entity Recognition						Chinese Word Segmentation					
	People's Daily			Weibo			GSD			PUD		
	B	RP		B	RP		B	RP		B	RP	
Aya-command	38.24	36.36	(-1.88)	37.88	30.83	(-7.05)	87.98	89.08	(+1.10)	88.68	91.05	(+2.37)
Claude-3	69.74	73.79	(+4.05)	45.64	46.86	(+1.22)	94.90	95.16	(+0.26)	94.12	94.96	(+0.84)
ERNIE-Lite-8K	12.10	12.99	(+0.89)	6.72	6.90	(+0.19)	88.04	88.70	(+0.66)	69.54	73.57	(+4.03)
GPT-3.5	56.89	55.97	(-0.92)	36.65	36.64	(-0.01)	95.68	94.87	(-0.81)	93.91	93.70	(-0.21)
GPT-4	66.04	68.05	(+2.01)	43.83	44.68	(+0.85)	94.21	94.88	(+0.67)	94.24	95.63	(+1.39)
o1-mini	84.21	91.67	(+7.46)	56.37	69.70	(+13.3)	97.21	100.0	(+2.79)	93.65	97.00	(+3.35)
QWen 72B	62.73	59.59	(-3.14)	31.78	35.83	(+4.05)	96.59	95.57	(-1.02)	89.79	91.94	(+2.15)

Table 5: Model performances for NER and CWS tasks with baseline (B) and radical prompting (RP).

format, with target sentences annotated in a manner similar to Blevins et al. (2023). Each task and dataset is evaluated using 2,000 sample sentences five times. Due to higher costs, Claude-3 and GPT-4 are evaluated with 1,000 samples. Additionally, we experiment with o1-mini (OpenAI, 2024c) using 100 samples, repeated three times. We experiment with three different prompting methods:

Baseline. Our baseline employs the CoT prompting framework with steps that guide the model to execute tasks. See prompts in Appendix F.4.

Radical Prompting. We incorporate the radical information into the input prompt as steps within the CoT framework. The process begins with the model identifying any unclear words within a given context. Then, the model is instructed to dissect these words into their constituent radicals and attempt to utilize useful radicals to aid the task. Steps are then provided to guide the model in executing specific tasks, identical to the baseline, with three examples. When using radical prompting, it is important to guide models to critically assess information from character components to avoid being misguided. Therefore, one of the three examples intentionally includes radical that can be misleading, helping the model learn to discern when to use radical information. Prompt lines of radical prompting are listed in Appendix F.3.

Radical Prompting (Oracle). Similar to the radical prompting method, instead of instructing the model to decompose characters, we directly provided the correct radicals in the input prompt. This method was applied only to the POS tagging task, as it required supplying the radical of just the central word. For the other tasks, it is impractical to provide radicals for all characters in the sentence.

5.3 Experimental Results

Our results suggest that radicals have strong potential if models can properly understand and utilize them. Results for POS tagging are shown in Table 4 with qualitative analysis in Appendix F.1. In the POS tagging task, models consistently show improvement across datasets, especially when the correct radicals are provided. Notably, in the PUD dataset, ERNIE-Lite-8K exhibits a slight decrease in performance without the correct radicals but shows an increase of approximately 11 F1 points when the correct radicals are included.

For the NER and CWS tasks, the initial results are mixed as shown in Table 5. However, our error analysis reveals that with the radical prompting method, incorrect answers often occur when the model bypasses the use of radicals and asserts that there are no ambiguous words in the sentence being examined. This suggests that the negative effect may be attributed to the longer prompts, as more robust models, such as Claude-3 and GPT-4, still demonstrate improvement in performance across datasets. When evaluating only the samples where the model identifies ambiguous words in the radical prompt setting, we find that the models genuinely perform better, as shown in Table 3. However, a notable exception is the Aya-command, whose performance drops significantly on the Weibo dataset. Upon closer examination, we find that Aya has a strong tendency to split words into individual characters rather than into radicals. Examples of such output are shown in Appendix F.5.

6 Conclusion

In this paper, we create a comprehensive benchmark on visual information embedded in Chinese characters. Our evaluation of the benchmark highlights the suboptimal performance of LLMs and VLMs in handling information below the character level. Despite this, our experiments with *radical prompting* demonstrate that these sub-character features can still be beneficial. The results show consistent improvements in POS tagging when correct radicals are provided, and promising results in NER on sentences containing unfamiliar words. Our work highlights the potential of radical knowledge for Chinese NLP applications and advocates attention to help models leverage it, including additional training on radicals or improving Chinese digital systems to more effectively integrate radical structures.

Limitations

While our study offers valuable insights into the integration of radical prompting in Chinese language models, it also highlights areas for further exploration. First, the dataset used in this research does not represent the full range of Chinese characters, as the majority are sourced from simplified Chinese.

Moreover, the study primarily evaluates radical prompting on a limited selection of models and tasks, which may not fully capture its potential across a wider range of models and language processing applications.

Lastly, an area for improvement in our methodology involves the exclusive use of English in our prompt lines. Incorporating Chinese in the prompting strategy could further enhance the relevance and effectiveness of the prompts, better aligning with the linguistic context of the target language.

Acknowledgements

We extend our gratitude to Dengchun Yuan for his inspiring discussions on this work and to Yao Dao for offering valuable advice on prompting design. We also thank Duong Minh Le for his helpful guidance on writing. Finally, we appreciate the efforts of Geyang Guo, Shu Zhu, and Wei Zhou for their assistance with the validation process.

References

01.AI, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open Foundation Models by 01.AI.

Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku. Accessed: 2024-06-09.

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. *arXiv preprint arXiv:2309.16609*.

BaichuanInc. 2024. Baichuan-13B-Base. Accessed: 2024-06-11.

Baidu. 2024a. Introducing ERNIE 3.5: Baidu's Knowledge-Enhanced Foundation Model Takes a Giant Leap Forward. Accessed: 2024-06-11.

Baidu. 2024b. Yiyan. Accessed: 2024-06-11.

BigScience. 2024. BLOOM-7B1. Accessed: 2024-06-11.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. 2023. Prompting language models for linguistic structure. In *Proceedings of the 61st Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 6649–6663, Toronto, Canada. Association for Computational Linguistics.

Du Chen, Yi Huang, Xiaopu Li, Yongqiang Li, Yongqiang Liu, Haihui Pan, Leichao Xu, Dacheng Zhang, Zhipeng Zhang, and Kun Han. 2024. Orion-14B: Open-source Multilingual Large Language Models. *arXiv preprint arXiv:2401.12246*.

Han Chen. 2023. People's Daily (RenMin Daily) Named Entity Recognition Dataset. http://paper.people.com.cn/. A comprehensive dataset from the People's Daily, covering news from 2021/01/01 to 2023/12/05, for Named Entity Recognition with news segments labeled for LOC, ORG, PER entities using BIO tagging strategy. License: CC0: Public Domain.

HFL. 2024. Chinese llama-2-7b. Accessed: 2024-06-11.

InternLM. 2024. Internlm-7b.

Kawabata Kawabata, Masaya Nakamura, and Huáng Jùnliàng. 2018. CJKVI-IDS: Ideographic Description Sequences for CJK Unified Ideographs. https://github.com/cjkvi/cjkvi-ids. Accessed: 2024-4-4.

Xiaodong Liu, David Wisniewski, L. Vermeylen, Ana F. Palenciano, Wenjie Liu, and M. Brysbaert. 2021. The Representations of Chinese Characters: Evidence from Sublexical Components. *Journal of Neuroscience*, 42(1):135.

Jiefeng Ma, Zirui Wang, and Jun Du. 2021. An Open-Source Library of 2D-GMM-HMM Based on Kaldi Toolkit and Its Application to Handwritten Chinese Character Recognition. *Lecture Notes in Computer Science*, 12888.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Tackstrom, Claudia Bedini, Nuria Bertomeu Castello, and Jungmee Lee. 2023. Parallel Universal Dependencies (PUD) Treebanks for Multilingual Parsing. Available for the CoNLL 2017 shared task on Multilingual Parsing from Raw Text to Universal Dependencies. Annotations provided by Google and converted to UD v2 guidelines by the UD community.

MoonshotAI. 2024. Kimi. Accessed: 2024-06-11.

OpenAI. 2023. GPT-4. Accessed: 2024-06-11.

OpenAI. 2024a. Gpt-3.5 turbo. Accessed: 2024-06-11.

OpenAI. 2024b. Gpt-4o system card. Accessed: 2024-10-16.

OpenAI. 2024c. Openai o1 mini: Advancing cost-efficient reasoning. Accessed: 2024-10-16.

Nanyun Peng and Mark Dredze. 2015. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In *Proceedings of the Human Language Technology Center of Excellence*, Baltimore, MD. Johns Hopkins University.

Peng Qi and Koichi Yasuoka. 2023. Simplified Chinese Universal Dependencies Version 2.13. Universal Dependencies (UD) Chinese GSDSimp treebank. Available from GitHub: UD_Chinese-GSDSimp.

Anton Razzhigaev, Anton Voronov, Andrey Kaznacheev, Andrey Kuznetsov, Denis Dimitrov, and Alexander Panchenko. 2022. Pixel-level BPE for auto-regressive image generation. In *Proceedings of the First Workshop on Performance and Interpretability Evaluations of Multimodal, Multipurpose, Massive-Scale Models*, pages 26–30, Virtual. International Conference on Computational Linguistics.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language Modelling with Pixels. In *The Eleventh International Conference on Learning Representations*.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical Embedding: Delving Deeper into Chinese Radicals. In *Proceedings of the Association for Computational Linguistics (ACL)*. Sogou Technology Inc., Beijing, China.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2021. Sub-Character Tokenization for Chinese Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 9:634–649.

Karl Stratos. 2017. A Sub-Character Architecture for Korean Language Processing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2420–2430, Copenhagen, Denmark. Association for Computational Linguistics.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information. In *Proceedings of the Association for Computational Linguistics (ACL)*. Shannon.AI; Zhejiang University; Key Lab of Intelligent Information Processing of Chinese Academy of Sciences.

Franck Xia. 1994. Knowledge-based sub-pattern segmentation: decompositions of Chinese characters. *Proceedings of the International Conference on Image Processing*.

Koichi Yasuoka. 2023. RoBERTa Model Pre-trained on Classical Chinese Texts. https://huggingface.co/KoichiYasuoka/roberta-classical-chinese-large-char. Derived from GuwenBERT-large with character-embeddings for traditional/simplified characters.

Suitable for tasks like sentence-segmentation, POS-tagging, dependency-parsing.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Zhiyuan Liu, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023. GLM-130B: An Open Bilingual Pre-trained Model. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Jianshu Zhang, Yixing Zhu, Jun Du, and Lirong Dai. 2018. Radical Analysis Network for Zero-Shot Learning in Printed Chinese Character Recognition. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Hefei, Anhui, P.R. China. IEEE.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya Model: An Instruction Finetuned Open-Access Multilingual Language Model. *arXiv preprint arXiv:2402.07827*.

A General Experiment Details

Model Snapshots The experiments incorporated different versions of widely recognized models to evaluate their performance in processing Chinese characters. The specific snapshots used for each model are as follows:

- **GPT-3.5** and **GPT-4** were used with the snapshot dated *2023-11-06*.
- **Claude** model's evaluation utilized the *2024-02-29* snapshot.
- **Ernie-Lite-8K** was tested using the *2023-09-22* snapshot.

Temperature Settings

- **Aya**, **Yi-6B**, **Qwen-7B-Chat**, **Baichuan-13B**, and **Mistral-7B** were set at a lower temperature of *0.3* as recommended.
- For **other models** not specifically mentioned, a temperature setting of *0.7* was used.

B Validation on Radical Annotation

To ensure the quality of our radical annotations, we conducted a validation process with a team of four annotators. This team included three volunteer native Chinese speakers and one native Chinese-speaking author. Prior to the formal annotation process, each annotator underwent a brief practice session to familiarize themselves with the task. Following this, the speed of checking 100 rows is around two and a half minutes per annotator.

For the validation task, we randomly assigned 3,500 rows to each annotator from our dataset, ensuring that the task could be completed within roughly two hours. Each annotator was instructed to review the assigned rows and flag any errors they identified in the radical annotations. The errors flagged were then collectively analyzed to compute the error rate and guide the necessary revisions.

C Details on Visual Info Evaluation

C.1 Detail Settings

For our evaluation, we use different sampling methods and settings based on the type of model. For LLMs, a random sample of 1,000 characters is selected for each task and model. Due to higher costs, the number of samples for VLMs is reduced to 500. ERNIE-V and Kimi-V, which lack API access, are tested manually with only 100 samples. We incorporate few-shot learning by providing models with three examples for each task, except for the structure recognition task, where one example per structure type is given. In the Chain-of-Thought (CoT) setting, models are prompted to break down their reasoning process step-by-step, with detailed prompts provided in the Appendix C.2. Models with fine-tuning are trained in OpenAI's platform using cross-entropy loss with a 7:3 split and tested using 1,000 samples randomly selected from the CCD dataset. To assess consistency and model entropy, each question is asked five times, and the best trial out of the five for each task is selected to calculate the overall results.

To adapt answers from models generating long responses conventionally, we first let models generate responses freely without a specific answer format. Then, we use GPT-3.5 Turbo to extract answers from various model responses. For open-source models and extraction-used GPT-3.5 Turbo, a temperature of 0.3 is applied. Closed-source models generally use a temperature of 0.7 unless otherwise recommended by model documentation.

C.2 CoT Prompting

We present the prompt lines used for visual info evaluation in Figure 9.

C.3 Chinese VLMs Behavior

Examples of VLMs misrecognizing images are shown in Figure 17, 18, 19, 20, and 21.

C.4 GPT-4o and GPT-o1 Behavior

When simply prompting GPT-4o and GPT-4.1 with the question "What does this character mean?" for unfamiliar characters, both models provide responses with radical information, but they do so incorrectly (as shown in Figures 22). This suggests that while the models have been exposed to radicals in their training corpus, they are still unable to correctly identify or interpret the appropriate radicals.

D Analysis on Chinese Character Encoding

D.1 Experiment on Encoding

To further investigate why models after fine-tuning perform exceptionally well on structure tasks but show decreased performance on other Chinese visual tasks, we conducted a side experiment on different encoding systems to determine if they learn some sort of implicit pattern from the encoding.

Setup. We fine-tuned GPT-3.5 by explicitly switching all Chinese characters in the training and testing documents to various encodings—namely, Unicode, stroke, Pinyin[10], Wubi, and Cangjie[11]—and evaluated them on the structure recognition task to assess the impact of these representations on the model's learning ability with visual knowledge of Chinese characters.

Results. The results shown in Table 7 indicate that Unicode encoding performs comparably to vision-rich stroke encoding and significantly outperforms Pinyin encoding, which is limited to phonetic information. Upon further investigation, we found that the order of Chinese characters in Unicode is closely related to the stroke count and structure of the characters: Unicode is ordered by the stroke count of their indexing radical and the stroke count of the remaining parts. However, the full potential of Unicode is diminished by numerous exceptions and a broad spectrum of extensions that complicate its utility in conveying visual knowledge, where similar structures are likely grouped together with stroke counts in incremental order, as detailed in Table 6.

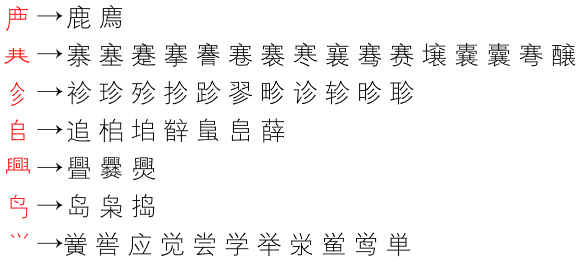

Figure 6: Examples of untypable radicals

D.2 Challenges on Radical Encoding

None of the encoding systems mentioned above can fully exploit the potential of radicals. Stroke-based systems over-decompose characters into individual strokes, losing meaningful structure, while glyph-based input methods like Wubi and Cangjie oversimplify and over-categorize characters to prioritize efficiency as input methods.

However, a significant challenge lies in developing a radical-based encoding system. While some radicals have corresponding Unicode representations, they cannot be typed using standard input methods. With around 100,000 CJK ideographs in Unicode, the task becomes even more difficult, as identifying the correct representation requires manual searching by sight, since there is no way to input the radical for automated searching, as shown in Figure 6. This limitation forces us to decompose some radicals further to maintain the integrity of character representation. Unfortunately, this results in a loss of meaning in certain cases, as the radicals become fragmented beyond their functional roles within characters.

E Discussion on Chinese Characters

To investigate the importance of Chinese radicals, we selected a sample of 100 Chinese characters from our dataset and annotated them to determine whether the radicals directly contribute to the meaning or pronunciation of the character, as shown in Figures 7. Although the majority of characters have clues derived from the radicals, we found that most characters contain a combination of only one meaningful radical with other radicals hinting at pronunciation. For example, in the character "花," we can infer that it is related to herbs from the radical "艹," while "化" only provides a pronunciation hint, resulting in only a vague idea of character's

[10]Pinyin is the Romanization of the Chinese characters based on their pronunciation. In Mandarin, it's the standard method for typing Chinese characters.

[11]Wubi and Cangjie are two glyph-based input methods that are uncommon to use.

Unicode	Character	Structure	Unicode	Character	Structure
U+4EBF	亿	Left-Right	U+4ED9	仙	Left-Right
U+4EC0	什	Left-Right	U+4EE3	代	Left-Right
U+4EC1	仁	Left-Right	U+4EEA	仪	Left-Right
U+4EC3	仃	Left-Right	U+4EEB	仫	Left-Right
U+4EC4	仄	Wrapping	U+4EF0	仰	Left-Right
U+4EC7	仇	Left-Right	U+4EF2	仲	Left-Right
U+4ECE	从	Left-Right	U+4EF5	仵	Left-Right
U+4ED1	仑	Top-Bottom	U+4EFB	任	Left-Right
U+4ED3	仓	Top-Bottom	U+4EFD	份	Left-Right
U+4ED5	仕	Left-Right	U+4F01	企	Top-Bottom
U+4ED6	他	Left-Right	U+4F0A	伊	Left-Right
U+4ED7	仗	Left-Right	U+4F0D	伍	Left-Right
U+4ED8	付	Left-Right	U+4F0E	伎	Left-Right

Table 6: This table showcases a randomly selected range of Unicode characters in the dataset along with their respective structures. This representation provides a snapshot of the structural information inherent in the Unicode.

Encoding	Structure Acc
Unicode	39.80
Stroke	43.80
PinYin	13.85
WuBi	11.81
CangJie	11.66

Table 7: GPT-3.5 fine-tuning' performance on different ways of encoding.

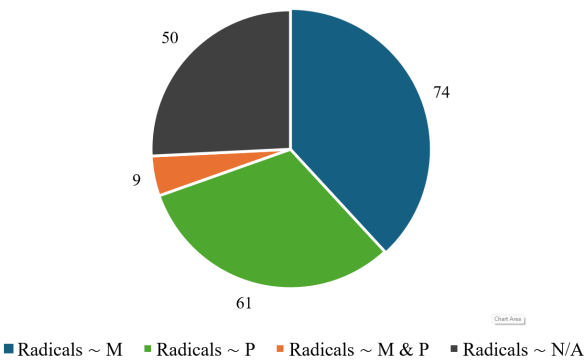

Figure 8: Sampled distribution of radicals with meaning (M) or Pronunciation (P) hint.

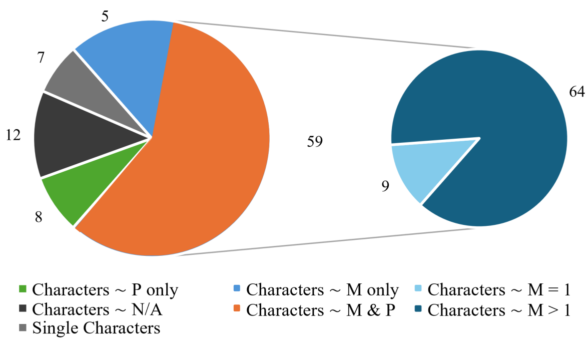

Figure 7: Distribution of Chinese characters with meaning (M) or pronunciation (P) hint from their radicals. The smaller circle on the right shows the distribution among all characters containing radicals with meaning (sum of Characters M only and Characters M & P).

meaning. In 12 out of the 100 characters, none of the radicals were helpful.

This is due to the evolution of the language, where historically, a single Chinese character often conveyed the meaning of a full word. However, more words are now composed of two or more characters, leading to individual characters losing their original meanings. For example, the Chinese character "况" is now commonly used to mean "situation" in words like "情况" or "状况". However, the original meaning of the character is "cold water" unexpectedly, which is closely related to the radical "冫", referring to cold water.

F Detailed Radical Prompting Result

F.1 Quantitative Analysis on POS tagging Accuracy

We provide a case analysis for POS tagging in Table 8.

Category	Baseline	RP (Oracle)
Correct& utilize Radical	-	81.2 (+81.2)
Correct without	608.6	611.2 (+2.6)
Incorrect & utilize Radical	-	41.8 (+41.8)
Incorrect without	391.4	265.8 (-125.6)

Table 8: Quantitative analysis of GPT-3.5-Turbo's POS tagging accuracy on the number of correct and incorrect predictions with and without the examination of components using radical prompting compared to the baseline. Improvement is shown in green.

F.2 Window size's impact on POS tagging

We evaluate the impact of different window sizes in POS tagging with GPT-3.5-Turbo in Table 9.

Window Size	Part-Of-Speech Tagging		
	GPT-3.5-Turbo with GSD		
	B	RP	RP (Oracle)
5	59.08	64.62 (+5.5)	67.56 (+8.5)
7	60.17	66.55 (+6.38)	66.73 (+6.56)
9	60.38	67.03 (+6.65)	67.23 (+6.85)

Table 9: Model performance for POS tagging with different word window sizes

F.3 Radical Prompting Prompts

We provide our radical prompting lines for POS tagging, NER, and CWS tasks in Figure 11, 12, and 13, respectively.

F.4 Base Prompting Prompts

We provide our base prompting lines for POS tagging, NER, and CWS tasks in Figure 14, 15, and 16, respectively.

F.5 Aya Model Behavior

Examples of Aya decomposing radicals incorrectly are shown in Figure 10.

G Responsible NLP Miscellanea

G.1 Intent usage

In response to potential inquiries regarding the scope and legitimacy of our experiments, it is important to clarify that all aspects of our research strictly adhere to the intended use cases of the Large Language Models (LLMs) and the NLP task datasets employed. Furthermore, our use of these models and datasets complies fully with the usage policies of the APIs for each model involved. We note that the use of rare Chinese words triggered some safety mechanisms in models such as Gemini-1.5. However, our intent complies fully with the ethical guidelines and usage policies provided by the API providers.

G.2 Computational Experiments Cost

In our research, we utilized vLLMs for evaluation on Yi 6B, Mistral 7B, Baichuan 13B, and Qwen 7B with a single a40 GPU. For other models, we accessed them through their respective APIs. The cost and running time for each model varied significantly. Specifically, the time required to run a single evaluation ranged from approximately 2 to 8 hours.

G.3 Avoid Data Leakage

For all NLP tasks assessed in this study, evaluations were exclusively conducted on the development sets of the respective datasets to prevent data leakage.

G.4 Personally Identifying Info

The dataset we created for evaluating the visual information of Chinese characters does not contain any offensive content or personally identifying information. However, we acknowledge the presence of individual names in the Weibo NER dataset that we use for evaluation.

G.5 Evaluation Tools and Methodologies

To evaluate our Named Entity Recognition (NER) tasks, we used a Perl script: conlleval.pl.

For other tasks, we calculated the F1 score using Scikit-learn.

G.6 AI Assistants

We acknowledge the use of GPT-4 for grammar checking and word polishing.

The structure of a Chinese character must be one of the following: 上下结构, 左右结构, 上中下结构, 左中右结构, 包围结构, 镶嵌结构, 单一结构, 品字结构. Let's think step by step. First identify the radical of the character. The radical is usually associate with the property of the character. Then, based on the relative position of the radical and remaining component of the character, identify the structure of the character. The structure of Chinese character {character} is	1) Recognize Basic Components: Break down the character into its basic components or radicals. This can help in counting the strokes more accurately. 2) Count Strokes in Each Component: For each component or radical, count the number of strokes. Use the general rules for stroke order to ensure no strokes are missed. 3) Sum the Strokes: Add the stroke counts of all components to get the total stroke count for the character. 4) Verify the Stroke Count: Cross-check the total stroke count with reliable sources or stroke count databases to ensure accuracy. Output the number of strokes required to write Chinese character {character}:
When decompose Chinese character into its constituent components, you should list its components in the following specific order based on its structure: For vertical structures: top to bottom, For horizontal structures left to right, For wrapping structures: from outside to inside, For inlays: main component first, followed by embedded components. Let's think step by step. First identify the radical of the character. The radical is usually associate with the property of the character. Then, based on the relative position of the radical and remaining component of the character, identify the structure of the character. The components of Chinese character {character} is:	In Chinese calligraphy, characters are composed of five standard strokes: 横 (一): This includes 横 and 提 竖 (丨): This includes 竖 and 竖钩 撇 (丿): This includes 撇 捺 (丶): This includes 捺 and 点 折 (フ): This includes all types of 折 such as 横折 and 横钩 1) Recognize the basic components. Break down the character into its basic components or radicals, as this can help in understanding the structure and stroke order. 2) Apply general stroke order rules. Recall the general rules for Chinese stroke order: Top to bottom Left to right Outside before inside Main before inlays 3) Determine the specific stroke order. Using the rules and components identified, determine the specific stroke order for the character. What are the strokes of the Chinese character {character} in order?"""

Figure 9: Prompt Line of Visual Tasks. Sub-windows are organized as follows: Structure (Top Left), Radical (Bottom Left), Stroke Count (Top Right), and Stroke Decomposition (Bottom Right). The red section indicates the Chain-of-Thought (CoT) prompting portion.

Aya Response:

Step 3: The name “田馥甄” can be understood by its components: “田” refers to a field. “馥” means fragrance, and “甄” means to select or distinguish. Together, “田 馥 甄” is a person‘s name. The song title “还是要幸福” doesn’t need further explanation. The term “声音” means voice or sound and doesn’t require additional context.

Step 3: Examine ambiguous entities:
“冷气思密达”: The meaning of this phrase is unclear. A further check on the components might provide some insight. The characters suggest something related to “cold” (“冷”) and “air” or “spirit” (“气”). (“思密达”) might be a transliteration of an English word or expression, but without context, it‘s challenging to determine its exact meaning.

“曾若彤”: This appears to be a person’s name. The characters suggest a female name with the components indicating “formerly” or “once” (“曾”), “like” or “as if” (“若”), and “red” or “pink” (“彤”), which is often used in names to imply beauty or a vibrant personality.

Figure 10: Example of Aya decomposes incorrectly.

Task
Perform name entity recognition (NER) task for the sentence below. The label should be chosen from the following set: {{'B-LOC' 'B-ORG' 'B-PER' 'I-LOC' 'I-ORG' 'I-PER' 'O'}}

Please note:
1. You should choose only from the label set provided above.
2. Strictly tag EACH character in the sentence with BIO standard.
3. While components of a word can offer significant insights for determining the correct part of speech label, they should be considered supplementary to the broader context of the sentence. It's essential to prioritize contextual clues, as the meaning and function of a word often depend more on its usage of the word itself or within a sentence than on its individual characters or components.

Read the provided sentence carefully and identify labels by strictly output steps below.

Step 1. Identify the meaning of the sentence as a whole.

Step 2. Recognize the all entities in the sentence.

Step 3: If an entity has an ambiguous meaning, examine the associated component information as it may offers additional insights.

Step 4: List entities' tags with reasons.

Step 5. Finalize the answer in a JSON format: there should be only one key "label" and the value must be the tag for each character and its corresponding tag combined by "|".

The Provided Sentence
Sentence to Analyze:
"{text}"

Thought:

Examples
*Example 1 Sentence to Analyze:
"在 陕 西 渭 南 市 ， 杜 雪 田 被 中 智 选 为 下 一 任 总 监。"
Thought:
Step 1. The sentence situates an event (meeting someone) in a specific location (Weinan City, Shanxi), and describes Du Xuetian being chosen as the next general manager by Zhong Zhi.
Step 2. The entities in this sentence includes: ""陕 西", "渭 南 市", "杜 雪 田", "中 智", and "总 监."
Step 3. The meaning of "杜 雪 田" might need a further check: The component of "杜 雪 田" is "{{木, 土; 雨, 彐; 田}}." "雨" suggests rain, which might be part of the given female's name to imply purity or natural beauty, while other component does not provide useful information, "杜 雪 田" is indeed a person name.
The meaning of "中 智" might need a further check: The component of "中 智" is "{{中; 知, 日}}." "中" suggest central, "知" suggest knowledge. "中 智" could suggest an organization that focus on central intelligence or wisdom.
Step 4. List of name entities in this sentence: "陕 西" should be tagged as LOC because it is a province in China. "渭 南 市" should be tagged as LOC because it is a city in ShaanXi. "杜 雪 田" should be tagged as PER because it is a person name. "中 智" should be tagged as ORG because it is a specific organization's name. "总 监" should be tagged as O because it is a general position.
Step 5. Final Answer:
{{
"label": ["在|O", "陕|B-LOC", "西|I-LOC", "渭|B-LOC", "南|I-LOC", "市|I-LOC", "，|O","杜|B-PER", "雪|I-PER", "田|I-PER", "被|O", "中|B-ORG", "智|I-ORG", "选|O", "为|O", "下|O", "一|O", "任|O", "总|O", "监|O", "。|O"]
}}

Example 2 Omitted…

*Example 3 Sentence to Analyze:
"如 今 ， 古 阳 关 烽 燧 还 在 。"
Thought:
Step 1. The sentence describes the beacon of Guyang Pass still exist.
Step 2. The entities in this sentence are "古 阳 关" and "烽 燧."
Step 3. The meaning of "烽 燧" may need additional look. The component of "烽 燧","{{火, 夆; 火, 遂}}," suggesting a relation to fire or signals. Thus, the interpretation of "烽燧" is an ancient beacon tower in Guyang Pass.
Step 4. List of name entities in this sentence: "古 阳 关" should be tagged as LOC because it is a historical location's name. "烽 燧" should be tagged as O because it is a general object.
Step 5. Final Answer:
{{
"label": ["如|O", "今|O", "，|O", "古|B-LOC", "阳|I-LOC", "关|I-LOC", "烽|O", "燧|O", "还|O", "在|O","。|O"]
}}

Figure 11: Radical Prompting Prompt Line of POS tagging.

Task
Analyze the sentence and perform Chinese word segmentation (CWS).

Please note:
1. The answer return must be separate by space.
2. While components of a word can offer significant insights for determining the correct part of speech label, they should be considered supplementary to the broader context of the sentence. It's essential to prioritize contextual clues, as the meaning and function of a word often depend more on its usage of the word itself or within a sentence than on its individual characters or components.

Please read the provided sentence carefully and segment it by strictly following the guide outlined below, step by step:
Step 1. Understand the sentence structure and general meaning

Step 2. Identify words that have unclear meanings.

Step 3. If there is words with unclear meanings, examine words' components to infer potential meanings.

Step 4. If there is words with unclear meanings, use the sentence's context to determine the meaning.

Step 5. Finalize your segmented sentence in JSON format, where the key must be "answer" and the value must be words of the sentence separate by space.

The Provided Sentence
Sentence to segment:
"{text}"

Thought:

Examples
*Example 1 Sentence to Analyze:
"上塘镇位于永嘉县南部，楠溪江下游。"
Thought:
Step 1. The general meaning of the sentence is: 上塘 Town is in the southern part of 永嘉 County and is downstream of the 楠溪 River.
Step 2. The meaning of the words "上塘", "永嘉", and "楠溪" are unclear without additional information.
Step 3. The component information of "上塘" is "上; 土, 唐." The first character, "上", means "above" or "up". The second component combines "土" (earth or soil) with elements of "唐." Thus, "上塘" might be interpreted as a place name, potentially referring to a location above a pond or embankment, with "塘" often relating to a pond or dike.
The component information of "永嘉" is "永; 士, 口, 丷, 一, 加." "永" means eternal or forever, suggesting a sense of timelessness. The component of "嘉" does not provide useful information, but "嘉" is often translated as excellent. "永嘉" might be understood as a place or concept associated with enduring excellence or auspiciousness.
The component information of "楠溪" is "木, 南; 氵, 奚." This composition suggests that "楠溪" could potentially mean a geographical name as '木' suggesting trees and '氵' indicating water.
Step 4. Final Answer:
{{"answer": "上塘 镇 位 于 永嘉 县 南部 ， 楠溪 江 下游 。"}}
*Example 2 Sentence to Analyze:
"南山截竹为筚篥。"
Thought:
Step 1. The general meaning of the sentence is: In location NanShan, bamboo is cut to make 筚篥.
Step 2. The meaning of "截竹" and "筚篥" is not clear.
Step 3. The component of "截竹" is "{{𢦏, 隹; 竹}}." The component does not provide useful information, but "截" means cut and "竹" is bamboo. Thus, two character should be separated.
The component of "筚篥" is "{{⺮, 毕; ⺮, 栗}}." "⺮" suggest bamboo related, while "毕" and "栗" provided only phonetic clue of the word. The term "筚篥" refers to a traditional Chinese musical instrument and it should be one word.
Step 4. Final Answer:
{{"answer": "南山 截 竹 为 筚篥 。"}}
*Example 3 Sentence to Analyze:
"如今，古阳关烽燧还在。"
Thought:
Step 1. The sentence describes the beacon of Guyang Pass still exist.
Step 2. The meaning of "烽 燧" may need additional look.
Step 3. The component of "烽 燧","{{火, 夆; 火, 遂}}," suggesting a relation to fire or signals. Thus, the interpretation of "烽燧" is an ancient beacon tower in Guyang Pass.
Step 4. Final Answer:
{{ "answer": "如今 ， 古阳关 烽燧 还在 。"}}

Figure 12: Radical Prompting Prompt Line of NER.

Task
Analyze the part of speech (POS) tag of the central word (enclosed in brackets [ ]) in a given section of a sentence with additional information on the component of the Chinese word. The label should be chosen from the following set: {{'ADJ', 'PUNCT', 'PRON', 'CCONJ', 'NUM', 'DET', 'X', 'PROPN', 'SCONJ', 'SYM', 'VERB', 'AUX', 'NOUN', 'ADP', 'PART', 'ADV'}}

Please note:
1. Label only the center word (the 3rd word) in the 5-word span provided.
2. You should choose only from the label set provided above.
3. Consider the broader spectrum of meanings and functions that a word can embody. For instance, the word "活动" at first glance may seem like a verb meaning "to move" or "to exercise." However, it can also function as a noun, referring to "an activity" or "an event."
4. The complexity of a character—determined by the number of components or the intricacy of each component—can influence its typical POS tag. Words with greater complexity tend to be nouns or pronouns, indicating specific entities or subjects. In contrast, words that are simpler or consist of a single component are more likely to be classified as particles (PART), coordinating conjunctions (CCONJ), or subordinating conjunctions (SCONJ). This pattern reflects the varying linguistic functions these words serve based on their structural complexity.
5. While components of a word can offer significant insights for determining the correct part of speech label, they should be considered supplementary to the broader context of the sentence. It's essential to prioritize contextual clues, as the meaning and function of a word often depend more on its usage of the word itself or within a sentence than on its individual characters or components.

Read the provided sentence carefully and identify the label.
Step 1. Identify the meaning of the central word without using component information. If the meaning is clear, ignore step 2 and go to step 3 without using component information.
Step 2. If the word's meaning is unclear, examine its components to infer potential meanings.
Step 3. Without looking at the context, consider all possible grammatical functions of the word, such as "活动" being both a verb and a noun.
Step 4. Use the sentence's context to determine the most suitable part of speech for the central word.
Step 5. Finalize your choice in JSON format, where the key must be "label" and the value must be the label you have chosen.

The Provided Sentence
Sentence to Analyze: "{text}" The Component of The Center Word: "{components}" “””

Examples
*Example 1 Sentence to Analyze:
"南部， [楠溪] 江 下游"
The Component of The Center Word:
"木, 南; 氵, 奚"
Thought:
1. The meaning of the central word "楠溪" is unclear without additional information.
2. The component information of "楠溪" is "木, 南; 氵, 奚." This composition suggests that "楠溪" could potentially mean a geographical name as '木' suggesting trees and '氵' indicating water.
3. Without consider the context, "楠溪" is most likely to be PROPN but there is a chance that it is NOUN.
4. Considering the sentence's context, "楠溪" is preceded by comma"，" which does not provide useful information. but it is followed by "江" , river. Thus, "楠溪" is most likely a proper noun here as the name of the river.
5. Therefore, the most suitable part of speech tag for the central word "楠溪" is PROPN.
Final Answer:
{{ "label": "PROPN"}}

*Example 2 Sentence to Analyze:
"他 希望 [能] 共享 办公"
The Component of The Center Word:
"厶, 月, 匕, 匕"
Thought:
1. The meaning of the central word "能" is clear without additional information. "能" means "be able to" in Chinese.
2. Without consider the context, The possible labels for "能" is VERB (when it means "to be able to" or "can") or AUX (when "能" is used to express capability, possibility, or permission).
3. Considering the sentence's context, The sentence structure and the presence of another verb "共享" (to share) immediately after "能" suggest that "能" is serving an auxiliary function rather than acting as a main verb on its own. The speaker's intent is to express a wish or hope, which is a modal use, supporting the use of "能" as an auxiliary (AUX) verb here.
4. Therefore, the most suitable part of speech tag for the central word "能" is AUX.
Final Answer:
{{ "label": "AUX"}}

*Example 3 Sentence to Analyze:
"技术 设计 [审查]， 随即"
The Component of The Center Word:
"宀, 申; 木, 旦"
Thought:
1. The meaning of the central word "审查" is clear without additional information. "审查" can refer to the process of examining or the action of reviewing documents to ensure they meet certain standards or criteria.
2. Without consider the sentence's context, "审查" can be VERB or NOUN. When "审查" means process of examining, it functions as a verb (VERB). When refer to the action of reviewing documents, it functions as a noun (NOUN).
3. Considering the sentence's context, the structure of the fragment suggests that "审查" is part of a nominal phrase "技术设计审查" (technical design review), indicating a process or event rather than an action being performed at the moment described. "审查" is followed by comma "，", which does not provide useful information.
4. Therefore, the most suitable part of speech tag for the central word "审查" is NOUN.
Final Answer:
{{ "label": "NOUN}}

Figure 13: Radical Prompting Prompt line for CWS.

Task
Analyze the part of speech (POS) tag of the central word (enclosed in brackets [ ]) in a given section of a sentence. The label should be chosen from the following set: {{'ADJ', 'PUNCT', 'PRON', 'CCONJ', 'NUM', 'DET', 'X', 'PROPN', 'SCONJ', 'SYM', 'VERB', 'AUX', 'NOUN', 'ADP', 'PART', 'ADV'}}

Examples
*Example 1:
Sentence to Analyze:
"南部， [楠溪] 江 下游"

Thought:
1. "楠溪" refers to a place name, specifically the lower reaches of the Nanxi River.

2. "楠溪" is most likely to be PROPN but there is a chance that it is NOUN.

3. Considering the sentence's context, "楠溪" is preceded by comma"，" which does not provide useful information. but it is followed by "江" , river. Thus, "楠溪" is most likely a proper noun here as the name of the river.

4. Therefore, the most suitable part of speech tag for the central word "楠溪" is PROPN.

Final Answer:
{{
"label": "PROPN"
}}

*Example 2:
Sentence to Analyze:
"他 希望 [能] 共享 办公"

Thought:
1. "能" means "be able to" in Chinese.

2. The possible labels for "能" is VERB (when it means "to be able to" or "can") or AUX (when "能" is used to express capability, possibility, or permission).

3. Considering the sentence's context, the sentence can translate into "He hopes [to be able] to share an office." "能" functions as a modal verb to express capability to share.

4. Therefore, the most suitable part of speech tag for the central word "能" is AUX.

Final Answer:
{{
"label": "AUX"
}}

*Example 3:
Sentence to Analyze:
"技术 设计 [审查]， 随即"

Thought:
1. "审查" can refer to the process of examining or the action of reviewing documents to ensure they meet certain standards or criteria.

2. Without consider the sentence's context, "审查" can be VERB or NOUN. When "审查" means process of examining, it functions as a verb (VERB). When refer to the action of reviewing documents, it functions as a noun (NOUN).

3. Considering the sentence's context, the structure of the fragment suggests that "审查" is part of a nominal phrase "技术设计审查" (technical design review), indicating a process or event rather than an action being performed at the moment described. "审查" is followed by comma "，", which does not provide useful information.

4. Therefore, the most suitable part of speech tag for the central word "审查" is NOUN.

Final Answer:
{{
"label": "NOUN"
}}
Please note:
1. Label only the center word (the 3rd word) in the 5-word span provided.
2. You should choose only from the label set provided above.
3. Consider the broader spectrum of meanings and functions that a word can embody. For instance, the word "活动" at first glance may seem like a verb meaning "to move" or "to exercise." However, it can also function as a noun, referring to "an activity" or "an event."

Read the provided sentence carefully and identify the label.
Step 1. Identify the meaning of the central word.

Step 2. Consider all possible grammatical functions of the word, such as "活动" being both a verb and a noun.

Step 3. Use the sentence's context to determine the most suitable part of speech for the central word.

Step 4. Finalize your choice in JSON format, where the key must be "label" and the value must be the label you have chosen.

The Provided Sentence
Sentence to Analyze:
"{text}"

Thought:

Figure 14: Base Prompt line for POS tagging.

```
## Task
   Perform name entity recognition (NER) task for the sentence below. The label should be chosen from the following set:
{{'B-LOC' 'B-ORG' 'B-PER' 'I-LOC' 'I-ORG' 'I-PER' 'O'}}

   ## Examples
   *Example 1:
   Sentence to Analyze:
   "在 陕 西 渭 南 市 ， 杜 雪 田 被 中 智 选 为 下 一 任 总 监。"

   Thought:
   Step 1. The sentence situates an event (meeting someone) in a specific location (Weinan City, Shanxi), and describes
Du Xuetian being chosen as the next general manager by Zhong Zhi.

   Step 2. The entities in this sentence includes: ""陕 西", "渭 南 市", "杜 雪 田", "中 智", and "总 监."

   Step 3. List of name entities in this sentence: "陕 西" should be tagged as LOC because it is a province in China. "渭
南 市" should be tagged as LOC because it is a city in ShaanXi. "杜 雪 田" should be tagged as PER because it is a
person name. "中 智" should be tagged as ORG because it is a specific organization's name. "总 监" should be tagged as
O because it is a general position.

   Step 4. Final Answer:
   {{
   "label": ["在|O", "陕|B-LOC", "西|I-LOC", "渭|B-LOC", "南|I-LOC", "市|I-LOC", "，|O","杜|B-PER", "雪|I-PER", "
田|I-PER", "被|O", "中|B-ORG", "智|I-ORG", "选|O", "为|O", "下|O", "一|O", "任|O", "总|O", "监|O", "。|O"]
   }}

 *Example 2:
   Sentence to Analyze:
   "这 个 网 络 连 接 着 周 绍 良 、 翁 万 弋 、 潘 吉 星 等 著 名 学 者 ， 连 接 着 王 益 、 王 仿 子 、 万 名 盈 等
出 版 界 耆 宿 。"

   Thought:
   Step 1. The sentence describes connections (network connections) to various individuals, categorizing them into two
groups: The first group includes 周绍良, 翁万弋, 潘吉星, who are referred to as famous scholars ("著名学者"). The
second group includes 王益, 王仿子, 万名盈, who are described as elder members of the publishing industry ("出版界耆
宿").

   Step 2. The entities in this sentence includes: "周 绍 良", "翁 万 弋", "潘 吉 星", "王 益", "王 仿 子", "万 名 盈", "学
者", "出 版 界", and "耆 宿."

   Step 3. List of name entities in this sentence: "周 绍 良", "翁 万 弋", "潘 吉 星", "王 益", "王 仿 子", "万 名 盈"
should be tagged as PER because they are individuals' names. "学 者" should be tagged as O because it is not a specify
name.  "出 版 界" should be tagged as O because it is not a specify organization but a general industry. "耆 宿" should be
tagged as O because it is a general term refers to respected elder.

 Step 4. Final Answer:
   {{
   "label": ["这|O", "个|O", "网|O", "络|O", "连|O", "接|O", "着|O", "周|B-PER", "绍|I-PER", "良|I-PER", "、|O", "翁|B-
PER", "万|I-PER", "弋|I-PER", "、|O", "潘|B-PER", "吉|I-PER", "星|I-PER", "等|O", "著|O", "名|O", "学|O", "者|O", "，
|O", "连|O", "接|O", "着|O", "王|B-PER", "益|I-PER", "、|O", "王|B-PER", "仿|I-PER", "子|I-PER", "、|O", "万|B-PER",
"名|I-PER", "盈|I-PER", "等|O", "出|O", "版|O", "界|O", "耆|O", "宿|O", "。|O"]
   }}
   *Example 3:
   Sentence to Analyze:
   "如 今 ， 古 阳 关 烽 燧 还 在 。"

   Thought:
   Step 1. The sentence describes the beacon of Guyang Pass still exist.

   Step 2. The entities in this sentence are "古 阳 关" and "烽 燧."

   Step 3. List of name entities in this sentence: "古 阳 关" should be tagged as LOC because it is a historical location's
name. "烽 燧" should be tagged as O because it is an general object.

   Step 4. Final Answer:
   {{
   "label": ["如|O", "今|O", "，|O", "古|B-LOC", "阳|I-LOC", "关|I-LOC", "烽|O", "燧|O", "还|O", "在|O","。|O"]
   }}


   Please note:
   1. You should choose only from the label set provided above.
   2. Strictly tag EACH character in the sentence with BIO standard.

   Read the provided sentence carefully and identify labels by strictly output steps below.

   Step 1. Identify the meaning of the sentence as a whole.

   Step 2. Recognize the all entities in the sentence.

   Step 3: List entities' tags with reasons.

   Step 4. Finalize the answer in a JSON format: there should be only one key "label" and the value must be the tag for
each character and its corresponding tag combined by "|".

   ## The Provided Sentence
   Sentence to Analyze:
   "{text}"

   Thought:
```

Figure 15: Base Prompt line for NER.

```
## Task
Analyze the sentence and perform Chinese word segementation (CWS).

## Examples
*Example 1:
Sentence to Analyze:
"上塘镇位于永嘉县南部，楠溪江下游。"

Thought:
Step 1. The general meaning of the sentence is: 上塘 Town is located in the southern part of 永嘉 County and is
downstream of the 楠溪 River.

Step 2. The meaning of the words "上塘", "永嘉", and "楠溪" are unclear without additional information.

Step 3. The component information of "上塘" is "上; 土, 唐." The first character, "上", means "above" or "up". The
second component combines "土" (earth or soil) with elements of "唐." Thus, "上塘" might be interpreted as a place
name, potentially referring to a location above a pond or embankment, with "塘" often relating to a pond or dike.
The component information of "永嘉" is "永; 士, 口, 丷, 一, 加."  "永" means eternal or forever, suggesting a sense of
timelessness. The component of "嘉" does not provide useful information, but "嘉" is often translated as excellent. "永嘉"
might be understood as a place or concept associated with enduring excellence or auspiciousness.
The component information of "楠溪" is "木, 南; 氵, 奚." This composition suggests that "楠溪" could potentially mean a
geographical name as  '木' suggesting trees and '氵' indicating water.

Step 4. Final Answer:
{{
 "answer": "上塘 镇 位 于 永嘉 县 南部 ，  楠溪 江 下游 。"
}}
*Example 2:
Sentence to Analyze:
"南山截竹为筚篥。"

Thought:
Step 1. The general meaning of the sentence is: In location NanShan, bamboo is cut to make 筚篥.

Step 2. The meaning of "截竹" and "筚篥" is not clear.

Step 3. The component of "截竹" is "{{戋, 隹; 竹}}." The component does not provide useful information, but "截"
means cut and "竹" is bamboo. Thus, two character should be seperated.
The component of "筚篥" is "{{⺮, 毕; ⺮, 栗}}." "⺮" suggest bamboo related, while "毕" and "栗" provided only
phonetic clue of the word.  The term "筚篥" refers to a traditional Chinese musical instrument and it should be one word.

Step 4. Final Answer:
{{
 "answer": "南山 截 竹 为 筚篥 。"
}}

*Example 3:
Sentence to Analyze:
"如今，古阳关烽燧还在。"

Thought:
Step 1. The sentence describes the beacon of Guyang Pass still exist.

Step 2. The meaning of "烽 燧" may need additional look.

Step 3. The component of "烽 燧","{{火, 夆; 火, 遂}},"  suggesting a relation to fire or signals. Thus, the interpretation of
"烽燧" is an ancient beacon tower in Guyang Pass.

Step 4. Final Answer:
{{
 "answer": "如今 ，  古阳关 烽燧 还在 。"
}}

Please note:
1. The answer return must be seperate by space.
2. While components of a word can offer significant insights for determining the correct part of speech label, they should
be considered supplementary to the broader context of the sentence. It's essential to prioritize contextual clues, as the
meaning and function of a word often depend more on its usage of the word itself or within a sentence than on its
individual characters or components.

Please read the provided sentence carefully and segment it by strictly following the guide outlined below, step by step:
Step 1. Understand the sentence structure and general meaning

Step 2. Identify words that have unclear meanings.

Step 3. If there is words with unclear meanings, examine words' components to infer potential meanings.

Step 4. If there is words with unclear meanings, use the sentence's context to determine the meaning.

Step 5. Finalize your segemented sentence in JSON format, where the key must be "answer" and the value must be words
of the sentence seperate by space.

## The Provided Sentence
Sentence to segement:
"{text}"

Thought:
```

Figure 16: Base Prompt line for CWS.

Input character: [illegible]

Image: 

Response by Ernie-4V:
图片中展示的是一个人手持手机的黑白照片，但根据图像内容描述，并没有提及任何汉字元素。因此，无法直接描述图片中的汉字由哪些部分组成以及它们的结构顺序。然而，图像专家模型答案提到了一个汉字"哼"，但这与图片内容描述不符，因为图片中并没有显示这个汉字。
综上所述，由于图片中并未包含汉字元素，所以无法根据图片内容描述来回答用户关于汉字组成部分和结构顺序的指令。如果确实需要了解某个汉字的组成和结构顺序，建议直接提供该汉字的信息，而不是依赖与汉字无关的图片内容，

Google Translate: The picture shows a black-and-white photo of a person holding a cell phone, but according to the image content description, no Chinese character elements are mentioned. Therefore, it is impossible to directly describe what parts the Chinese characters in the picture are composed of and their structural order. However, the image expert model answer mentioned a Chinese character "哼", but this does not match the picture content description because the Chinese character is not shown in the picture. In summary, since the picture does not contain Chinese character elements, it is impossible to answer the user's instructions about the components and structural order of Chinese characters based on the picture content description. If you really need to know the composition and structural order of a Chinese character, it is recommended to directly provide information about the Chinese character instead of relying on picture content that is not related to the Chinese character.

Figure 17: Example of Ernie-4 with vision response to rare character with English translation.

Input character: 拔

Image: 

Response by Ernie-4V:
经过仔细查看，图片中展示的中文字符是"拔"而非"拔"。这个字符"拔"的笔画顺序是:横、竖钩、提、撇、横横、竖。如您仍有疑问，建议查阅现代汉语词典了解更多字符的笔画顺序。

After careful inspection, the Chinese character shown in the picture is "拔" instead of "拔". The stroke order of this character "拔" is: 横、竖钩、提、撇、横横、竖. If you still have questions, it is recommended to consult the Modern Chinese Dictionary to learn more about the stroke order of characters.

Figure 18: Example of Ernie-4 with vision response to extremely similar character with English translation.

Input character: 吖

Image: 

Response by Ernie-4V:
图片中的汉字是**'丫'，它的结构是'单一结构'**。

The Chinese character in the picture is **'丫', and its structure is 'single structure'**.

Figure 19: Example of Ernie-4 with vision response to part of the character as an answer with English translation.

Input character: 岜

Image: 

Response by Ernie-4V:
经过深入分析图片中的中文字符，可以确定该字符是**"芭"，其结构为左右结构**。。

After in-depth analysis of the Chinese characters in the picture, it can be determined that the character is **"芭", and its structure is a left-right structure**.

Figure 20: Example of Ernie-4 with vision response a character with different component parts as an answer with English translation.

Figure 21: Example of Kimi-v1 with vision reject rarely used character with English translation.

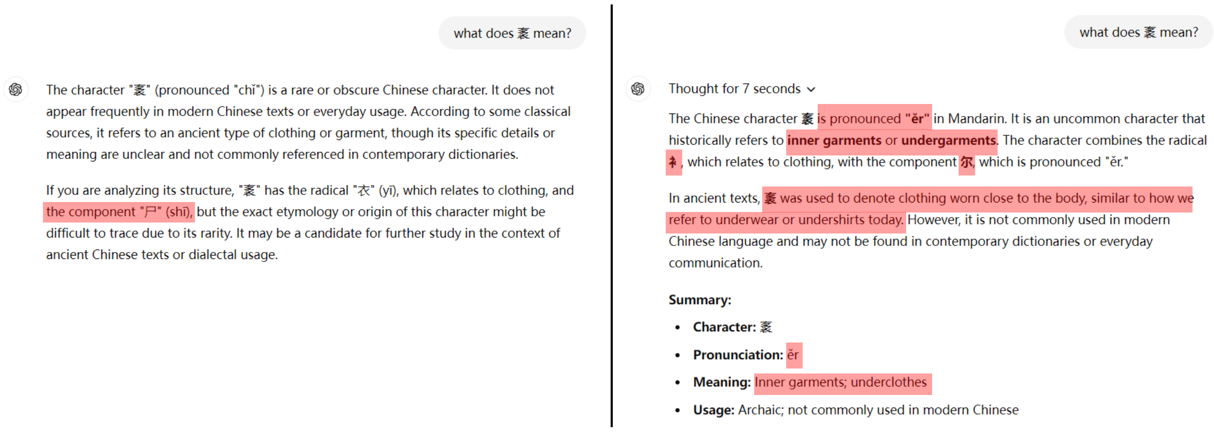

Figure 22: An example of GPT-4o and o1-preview's response, with incorrect statements highlighted in red.