

視覚言語モデルを利用した漢字構造自動解析の試み

守岡 知彦*

1 はじめに

OCR (Optical character recognition; 光学文字認識) は画像中の文字を読み取って電子テキスト化する技術である。日本語や中国語などの漢字系の書記言語の場合、かつては現代の印刷文字しか実用的でなかったが、深層学習技術を用いたいわゆる AI OCR の登場以降、従来、での認識が非常に困難であった崩字・草書で書かれた日本の古典文献を実用的な精度で機械的読み取りが可能となった。これは古典的なパターン認識技術に基づく文字の識別は文字画像から求められた特徴ベクトルの分類問題として実現されていたため、漢字のように非常に自種数の多い文字体系の場合、原理的な困難さを抱えていたからといえる。これに対し、今日の大規模言語モデルを用いた OCR では自然言語としてのそれっぽさという別の軸を用いて文字を推定できるため、漢字の字種数の多さという原理的な困難さに対してブレイクスルーをもたらし、東アジアの古典籍を対象とした OCR を実用化したといえる。

このように、現代の一般的な OCR は基本的に対象となるテキスト画像のレイアウトを解析し、自然言語として意味の通るデータに直すという処理を行い、検索や要約、あるいは、翻字・翻訳等を行って人間用に可視化したり、編集用のデータとして出力するための前処理を目的としたものといえる。この観点では、画像中の漢字の字体を忠実に再現するよりも意味が通じるように字種を識別し現代の通行字体に直す方が望ましいし、処理的にもこちらの方が簡単であろう。

しかしながら、漢字字体史研究という観点ではどの字体がどういう文脈で使われたかという情報も重要であり、字体の情報が捨象されてしまうのは望ましくない。また、一般的な OCR においても、どちらでも意味が通る複数の候補があったときに、字形に着目してどういう部品が含まれどういう構造をしているかを識別しどういう字体として認定するかという処理が必要なケースも考えられる。現状、この部分は人間による校正に委ねられて

* 国文学研究資料館

いる場合が多いと考えられるが、できればこれも可能な限り自動化したいものである。また、大量の外字リストに対し、CHISE 漢字構造情報データベースのようなものを自動生成したい場合、字形画像から IDS 形式の漢字構造データを出力できるとうれしい。

そこで、本稿では Qwen 3.5 [2] 等の大規模視覚言語モデルを用いて、漢字の字形画像中の部品を自動抽出し、IDS 形式の漢字構造データの自動生成を試みた実験の概要について説明する。

2 課題

2.1 句構造文法と座標位置の対応関係

視覚言語モデルを使って漢字画像の構造解析をさせる場合、結果を IDS 形式で出力できれば便利なのであるが、現状、視覚言語モデルは IDS をうまく理解できないようである。Qwen 3.5 27B 等の漢籍のテキストと画像を十分に学習していると思われる視覚言語モデルを用いた場合、プロンプトを工夫すれば、与えられた漢字画像に対して IDS を出せるケースもあるが、結果が安定しないようである。期待する処理は、漢字画像中の部品を認識し、各部品の相対位置から構造を推定し、その2次元的情報を句構造文法に落とし込むことであるが、いくつかの例から観察するに、うまく IDS を出せているケースでは、文字認識が成功し、CHISE 漢字構造情報データベースなどの IDS 形式の漢字構造記述から認識した文字に対応する IDS を引っ張ってくるというある種の『カンニング』めいたことになっていて、異体字やレアな漢字など一般的な OCR が苦手とする文字では途端にうまくいかないのだと思われる。そもそも、“IDS” という略語には多義性があり、“Ideographic Description Sequence” や「漢字構成記述文字列」などと書いたとしても、Qwen 3.5 はそれが何なのか理解していないようである。ならば、その定義をプロンプトで説明することも考えられるが、現在の視覚言語モデルでは共通のトークン上限の枠内でトークンをやりくりする仕組みになっているため、プロンプトで複雑な要求をすると視覚処理に齟齬が起きて部品の認識精度が悪くなる場合も少なくない。そもそも原理的に、画像における領域間の関係と句構造文法の世界を対応づけるのはやや難易度が高いタスクといえ、仮にやるべきことをちゃんと理解させられたとしても、あまり良い成績を期待できなかったり処理時間がかかりすぎたりするかもしれない。

2.2 位置情報の取得

Qwen 3.5 をはじめとする現在の一般的な視覚言語モデルでは画像を千 × 千程度の荒いメッシュに区切って扱っているため、正確な座標位置を扱うのは苦手である。そもそも大

規模言語モデルは数を扱うのが苦手ということもあり、座標位置を出力させようとするときしばしばハルシネーションが起こる。

一方、画像中のパターンや複数のオブジェクト間の大まかな位置関係を把握したり、これらを言語的な意味と対応づけるのも得意である。

2.3 文字と字体と部品

Qwen 3.5 をはじめとする現在の一般的な視覚言語モデルでは、解析が追いつかないくらい細かい文字や図形が複雑に入り組んだ大きすぎる画像でない限り、そこそこ豊富な文脈情報を有する大きめの画像の方が解析しやすい。つまり、漢字画像全体の解析の方が部品画像の解析よりもうまくいきやすいという傾向性があると考えられる。

その一方で、文字全体の認識ではテキストの構成要素としての文字、すなわち、字種に引きずられて、字体を捨象してしまいがちである。UCS 統合漢字は同源 (cognate) な複数の似た字体を一つの符号位置に統合している箇所があり、かつ、その統合の粒度は一様でないため、プロンプトによって字体概念をやりとりするのは容易ではない。¹⁾

その一方で、視覚言語モデルは漢字を構成する部品の抽象形状をある程度理解しているようではある。ただ、UCS の漢字統合によって、ある符号位置に対応する典型的な画像パターン集合²⁾が言語依存性を持ち、プロンプトを日本語で書くか、簡体字を使った大陸の中国語で書くか、繁体字を使った台湾の中国語で書くか、あるいは英語で書いたとしても漢字を Chinese character と書くか Hanzi と書くか Kanji と書くかといった差異により、影響を受ける部分があり、結果が異なる場合がある。Qwen のような中国製のモデルの場合、英語とともに簡体字で書かれた中国語がネイティブ言語的に扱われる傾向があり、情報が乏しい際には字種が簡体字によって代表されてしまうケースがあると考えられる。

また、基本的に中国語や日本語などにおいて利用頻度が高い漢字の認識が得意なのに対し部品の認識そのものは得意ではない。部品が単一の漢字として用いられるものと同一もしくは似た形状で単一の漢字としての使用頻度が高ければ認識しやすいと考えられるが、部品としては使用頻度が高くても単一文字として使用頻度が低いものや単一の文字としては用いられないもの（あるいは、部品化の際に大きな変形が生じているケースなど）の場合、それを含む使用頻度が高い漢字に化してしまうケースがある。あるいは、内部的には部品を適切に認識していたとしても、それを外部に対して適切に表現できないケースもあるかもしれない。

1) 人間相手でも容易ではないかもだが。

2) あるいは、これを視覚言語モデルにとっての字体と言って良いかもしれない。

3 設計

2.1 節で述べたように、現状、Qwen 3.5 等の視覚言語モデルから直接 IDS を出力するのは難しいので、代わりに、表 1 のような、部品毎に矩形座標と相対位置を示すシンボル (left, right など) を付けたタブ区切り (TSV) 形式で出力することにする。この各欄は、

0	0	48	136	イ	left
48	0	136	136	弋	right

表 1 視覚言語モデルからの出力イメージ

左上 X 座標、左上 Y 座標、右下 X 座標、右下 Y 座標、部品 (文字)、相対位置を示すシンボルである。なお、Qwen3 は画像の縦横をそれぞれ 1000 に正規化とした座標系を用いているため、外部出力の際には、実際の文字画像の座標に変換して扱うこととする。相対位置を示すシンボルとしては、

- left (偏)
- right (旁)
- above (冠)
- below (脚)
- surround-from-upper-left (垂)
- surround-from-lower-left (繞)
- full-surround (国構など)
- surround-from-above (門構など)
- surround-from-left (匚, 匚など)
- surround-from-upper-right (勹, 气など)
- surround-from-below (凵など)
- upper-left (左上)
- upper-right (右上)
- lower-right (右下)
- lower-left (左下)
- enclosed (構えの中の部品)
- sandwiched-from-left-and-right (左右の間)
- sandwiched-from-above-and-below (上下の間)
- inserted-from-left (「亟」の「口」など)

- inserted-from-right (「医」の「矢」など)

を用いることにする。これにより、例えば、「壘」のように入れ子構造を持つ字であっても、表2のようにある程度表現可能である。なお、この方法では複雑な入れ子構造は表現

2	2	39	26	田	above
2	26	24	50	田	lower-left
24	26	47	50	田	lower-right
2	50	47	74	土	below

表2 入れ子構造の字の場合の出力イメージ (壘)

できないが、そもそも複雑な入れ子構造での微細な部品の組み合わせの認識は難しいため、今回は高々2階層程度までの構造を対象とする。

部品の矩形座標の値はしばしば不正確ないしは出鱈目なものとなるのに対し、相対位置を示すシンボルはそれよりはましな結果になりやすい。なお、部品の矩形座標の値が著しく小さかったり、大きかったり、各部品が全て同じ矩形座標を占めるなど異常な値になっていれば、視覚言語モデルが漢字構造をうまく捉えられずハルシネーションを起こしていることが推測できるので、結果を破棄するのに用いることができる。同様に、矩形座標と相対位置を示すシンボルが矛盾している場合、矩形座標がそれっぽいものであれば、相対位置を示すシンボルが不適切であると予想することができる。例えば、surround-*系の囲んだり食い込むタイプのシンボルはsurroundの部分はあるとしてもその後の位置関係の部分は間違いやすい。また、上部の部品が垂れるか垂れないか曖昧な場合があり、aboveとの間で揺れたり、あるいは勝手にupperというシンボルを作り出したりすることもある。相対位置を示すシンボルの形式的定義を渡す訳ではないので、その自然言語としての意味からふんわりと処理することになるため、結果を漢字構造として解釈しIDSに変換する際にこの種の曖昧性を受け止める必要がある。

4 実装

前節で述べた手法に基づき、漢字構造認識プログラムをPythonで記述した。視覚言語モデル用のフレームワークであるMLX-VLM³⁾を用い、実行環境として

- Mac mini 2024 (Apple M4 Pro (CPU 14コア (パフォーマンス 10、高効率 4)、GPU 20コア、メモリー 64GB))

3) <https://github.com/Blaizzy/mlx-vlm>

- MacBook Pro 2023 (Apple M3 Pro (CPU 11 コア (パフォーマンス 5、高効率 6)、GPU 14 コア、メモリー 36GB))

を用いた。また、パッケージマネージャーとして uv⁴⁾ を使い、Python 3.13 を用いた仮想環境を作成した実験を行った。なお、この漢字構造認識プログラム (ids-recognizer.py) は

- <https://gitlab.nijl.ac.jp/CHISE/ids-recognizer>
- <https://github.com/chise/ids-recognizer>

で公開している。視覚言語モデルとしては、

- mlx-community/Qwen3.5-72B-heretic-8bit
- froggeric/Qwen3.6-72B-Uncensored-Heretic-v2-MLX-6bit
- froggeric/Qwen3.6-72B-Uncensored-Heretic-v2-MLX-8bit

を用いた。なお、32~36GB のユニファイドメモリーの環境で動作可能なのはこのうち、Qwen3.6-72B-Uncensored-Heretic-v2-MLX-8bit だけである。

この ids-recognizer.py は、

1. 視覚言語モデルを駆動し、入力した漢字画像と文字認識を行うためのプロンプトから、画像に対応する文字認識結果を得る
2. 視覚言語モデルを駆動し、入力した漢字画像とプロンプトから、前節述べたタブ区切り (TSV) 形式の出力を得る
3. 視覚言語モデルが出した TSV 形式を評価し、結果が思わしくなければプロンプトを変えて再び 2 に戻る
4. 視覚言語モデルが出した TSV 形式を IDS に変換する。この際、問題があれば 2 に戻る
5. リトライの上限に達して、IDS の生成に失敗した場合、1 の結果を返す

という行程により、異なるプロンプトで何度か視覚言語モデルを処理を動かし結果を得る。

3 の評価と 4 の IDS への変換はルールベースで行っている。

4.1 出力データの評価

この処理において行うことは、視覚言語モデルの出力結果の査読である。すなわち、視覚言語モデルが漢字画像をうまく解析できたかどうかを判断し、

4) <https://github.com/astral-sh/uv>

- 採択
- 条件付き採択
- リジェクト

のどれにするかを決めるというものである。

1 の採択は結果をそのまま採用し、次の IDS への変換処理に回すというものである。但し、IDS への変換処理においてもチェックが行われ、ここでエラーが出た場合は結局リジェクトになる場合もある。

2 は情報の一部を採用し、より厳密なプロンプトによって再解析を求めるというものである。視覚言語モデルは未知の部品に遭遇するなどして推論がうまくいかない場合、座標や部品に出鱈目な答えを出したり、あるいは認識できなかった部品を飛ばしたりするが、その場合でも、漢字部品が組み上げ方向が縦か横かといった大まかな構造や顕著な部品は見つけられていることが少なくない。そこで、今回は、組み上げ方向が縦方向か横方向になっている場合、その方向に限定したプロンプトを使って再解析することにした。

3 は結果を完全に捨てるというものである。この場合も再解析を行うことになるが、この際、最初の解析時とプロンプトを変える必要がある。

結果を判定するための情報としては、

1. 部品の矩形座標
2. 相対位置を示すシンボル
3. 部品（文字）

がある。

まず、出力結果にある全ての部品の矩形座標から左上の座標の最低値と右上の座標の最大値を求めるとそれらの部品を組み合わせた文字の左上と右上の座標になるはずだが、それが文字画像のサイズに比べて著しく小さければこの座標値は信用できないと考えられる。

他にも、全ての部品の矩形位置が（ほぼ）同じ場所を占めていたとしたらこれも多分おかしい。

あるいは、部品の矩形座標から推定される相対位置と相対位置を示すシンボルが矛盾している場合もどちらか（あるいは両方）が間違っていると判断できる。

また、部品によって、配置される位置に傾向性があるものがあり、例えば、草冠が単独で偏や旁に来たとしたら疑わしいし、「攴」のように脚にくることが多い部品が単独で旁に来た場合も上の部品を読めてない可能性が疑われる。同様に、「攴」のように旁にくることが多い部品が単独で頭に配置される場合も疑わしい。

ところで、Qwen3.5/3.6 での解析結果には表 3 のように文字全体や中間部品、それらを

2	2	150	150	遊	surround
2	2	74	150	ゑ	surround
76	2	150	150	旂	enclosed
76	2	150	74	方	upper-right
76	76	150	150	子	lower-right

表 3 複数の単位で重複して出力される例

さらに分解した部品などが重なった状態で含まれることがある。部品の矩形座標や相対位置を示すシンボルから推測できるように、視覚言語モデルは入り組んだ部品の組み合わせ構造に混乱して、「旂」の矩形座標がおかしくなり、「𠂔」ないしは「㇀」をうまく認識できなかったかあるいは表現できず、その一方で、「方」がこれらに引きずられて「子」と上下に重なるものという誤った結果が出てしまっている。

もし、CHISE [3] などの IDS データ [1] にアクセスできるなら部品の包含関係等を利用して判断することも可能であるが、今回は結果に文字全体が含まれている場合、部品への分解をせずにその文字を返すということにした。

なお、こうした出力データの評価はおそらく深層学習的手法が適していると考えられるが、今回はルールベースで実装した。

4.2 IDS への変換

視覚言語モデルの出力を IDS に変換する処理も今回はルールベースで実装した。これは主に相対位置を示すシンボルを用いて次のような処理を行うものである。

4.2.1 部品数が 1 の場合

相対位置を示すシンボルが full-surround もしくは full であれば部品（文字）を文字として返す。

4.2.2 部品数が 2 の場合

1 番目の部品を C_1 , 2 番目の部品を C_2 , n 番目の部品の左上の X 座標を $P_{x,n}$, Y 座標を $P_{y,n}$, 右下の X 座標を $Q_{x,n}$, Y 座標を $Q_{y,n}$ としたとき、表 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 で示す IDS を返す。

C_1	C_2	IDS
left	right	$\boxed{\square} C_1 C_2$
left	surround-from-lower-right	$\boxed{\square} C_1 C_2$
left	full-surround	$\boxed{\square} C_1 C_2$
left	full	$\boxed{\square} C_1 C_2$
left	below	$\boxed{\square} C_1 C_2$
left	above	$\boxed{\square} C_1 C_2$

表 4 left + x の時の IDS 生成規則

C_1	C_2	IDS
upper-left(言/言/隔)	lower-right	$\boxed{\square} C_1 C_2$
upper-left(雨/々)	(任意)	$\boxed{\square} C_1 C_2$
upper-left	(儿/女/心/虫)	$\boxed{\square} C_1 C_2$
upper-left	(任意) 但し、 $P_{x,2} < Q_{x,1}$	$\boxed{\square} C_1 C_2$
upper-left	(任意) 但し、 $Q_{y,1} \leq P_{y,2}$	$\boxed{\square} C_1 C_2$
upper-left(夕/ナ)	(任意)	$\boxed{\square} C_1 C_2$
upper-left(上記以外)	(任意)	$\boxed{\square} C_1 C_2$
upper-left(ゝ)	lower-left	$\boxed{\square} C_1 C_2$
upper-left(上記以外)	lower-left	$\boxed{\square} C_1 C_2$
upper-left	upper-right 但し、 $Q_{y,1} \leq P_{y,1}$	$\boxed{\square} C_1 C_2$
upper-left	upper-right 但し、上記以外	$\boxed{\square} C_1 C_2$
right	left	$\boxed{\square} C_2 C_1$
upper-left	(full-surround enclosed)(儿)	$\boxed{\square} C_1 C_2$
upper-left	(full-surround enclosed)(上記以外)	$\boxed{\square} C_1 C_2$
upper-left	(口 ナ)	$\boxed{\square} C_1 C_2$
upper-left(麻)	(任意) 但し、 $Q_{y,1} > P_{y,2}$	$\boxed{\square} C_1 C_2$
upper-left(麻)	(任意) 但し、上記以外	$\boxed{\square} C_1 C_2$
upper-left(麻)	(任意) それら以外	$\boxed{\square} C_1 C_2$

表 5 upper-left + x の時の IDS 生成規則

C_1	C_2	IDS
lower-left(父/糸)	lower-right	$\square\square C_1C_2$
lower-left	lower-right 但し、 $ P_{y,2} - P_{y,1} < 5$ かつ $ Q_{y,2} - Q_{y,1} < 5$	$\square\square C_1C_2$
lower-left	lower-right 但し、それら以外	$\square\square C_1C_2$
lower-left	right	$\square\square C_1C_2$
lower-left	upper-right	$\square\square C_1C_2$
lower-left	full-surround	$\square\square C_1C_2$
lower-left	full	$\square\square C_1C_2$
lower-left	enclosed	$\square\square C_1C_2$

表 6 lower-left + x の時の IDS 生成規則

C_1	C_2	IDS
above	below	$\square\square C_1C_2$
above	full-surround	$\square\square C_1C_2$
above	full	$\square\square C_1C_2$
above	enclosed	$\square\square C_1C_2$
above	right 但し、 $P_{x,1} \leq Q_{x,2}$	$\square\square C_1C_2$
upper(气)	(below lower full-surround)	$\square\square C_1C_2$
upper(上記以外)	(below lower full-surround)	$\square\square C_1C_2$

表 7 above + x の時の IDS 生成規則

C_1	C_2	IDS
below	above	$\square\square C_2C_1$
lower	upper	$\square\square C_2C_1$
below	surround-from-above	$\square\square C_1C_2$
lower	surround-from-above	$\square\square C_1C_2$

表 8 below + x の時の IDS 生成規則

C_1	C_2	IDS
upper-right(戈)	lower-left	$\square\square C_1C_2$
upper-right(上記以外)	lower-left	$\square\square C_1C_2$
upper-right	enclosed	$\square\square C_1C_2$
upper-right	lower-right	$\square\square C_1C_2$
upper-right	below	$\square\square C_1C_2$

表 9 upper-right + x の時の IDS 生成規則

C_1	C_2	IDS
enclosed	surround-from-lower-left	 C_1C_2
surround(ゝ/ゝ)	middle	 C_1C_2
surround(門)	middle	 C_1C_2
surround(これら以外)	middle	 C_1C_2

表 10 surround + middle の時の IDS 生成規則

C_1	C_2	IDS
full-surround(口)	(middle/enclosed)	 C_1C_2
full-surround(几)	(middle/enclosed)	 C_1C_2
full-surround(广/麻/尸)	(middle/enclosed)	 C_1C_2
full-surround(ㄣ)	(middle/enclosed)	 C_1C_2
full-surround(門/冂)	(middle/enclosed)	 C_1C_2
full-surround(戈/咸)	(middle/enclosed)	 C_1C_2
full-surround(これら以外)	(middle/enclosed)	 C_1C_2

表 11 full-surround + x の時の IDS 生成規則

C_1	C_2	IDS
surround-from-above(虍/鹿/广/犮/尸)	(below/covered/middle/enclosed/full-surround)	 C_1C_2
surround-from-above(ㄣ/穴/ㄣ/雨)	(below/covered/middle/enclosed/full-surround)	 C_1C_2
surround-from-above(これら以外)	(below/covered/middle/enclosed/full-surround)	 C_1C_2

表 12 surround-from-above + x の時の IDS 生成規則

C_1	C_2	IDS
surround-from-below(凵)	(middle/enclosed)	 C_1C_2
surround-from-below(上記以外)	(middle/enclosed)	 C_1C_2

表 13 surround-from-below + x の時の IDS 生成規則

C_1	C_2	IDS
surround-from-left(犮)	(middle/full/enclosed/covered)	$\boxed{\square} C_1 C_2$
surround-from-left(門/口)	(middle/full/enclosed/covered)	$\boxed{\square} C_1 C_2$
surround-from-left(ゝ/ゑ)	(middle/full/enclosed/covered)	$\boxed{\square} C_1 C_2$
surround-from-left(食)	(middle/full/enclosed/covered)	$\boxed{\square} C_1 C_2$
surround-from-left(これら以外)	(middle/full/enclosed/covered)	$\boxed{\square} C_1 C_2$
surround-from-left	right	$\boxed{\square} C_1 C_2$
surround-from-left	surround-from-right 但し、 $Q_{x,1} \leq P_{x,2}$	$\boxed{\square} C_1 C_2$

表 14 surround-from-left + x の時の IDS 生成規則

C_1	C_2	IDS
surround-from-upper-left(匚)	middle/enclosed/lower-right/right	$\boxed{\square} C_1 C_2$
surround-from-upper-left(匚)	middle/enclosed/lower-right/right	$\boxed{\square} C_1 C_2$
surround-from-upper-left(几/門)	middle/enclosed/lower-right/right	$\boxed{\square} C_1 C_2$
surround-from-upper-left(丿)	middle/enclosed/lower-right/right	$\boxed{\square} C_1 C_2$
surround-from-upper-left(それら以外)	middle/enclosed/lower-right/right	$\boxed{\square} C_1 C_2$
surround-from-upper-left	below	$\boxed{\square} C_1 C_2$
surround-from-upper-left(孝)	(任意)	$\boxed{\square} C_1 C_2$

表 15 surround-from-upper-left + x の時の生成規則

C_1	C_2	IDS
surround-from-lower-left(几/門)	middle/enclosed/upper-right/full-surround	$\boxed{\square} C_1 C_2$
surround-from-lower-left(匚)	middle/enclosed/upper-right/full-surround	$\boxed{\square} C_1 C_2$
surround-from-lower-left(それら以外)	middle/enclosed/upper-right/full-surround	$\boxed{\square} C_1 C_2$

表 16 surround-from-lower-left + x の時の生成規則

C_1	C_2	IDS
surround-from-right	middle	$\boxed{\square} C_1 C_2$
surround-from-right	enclosed	$\boxed{\square} C_1 C_2$

表 17 surround-from-right + x の時の生成規則

4.2.3 部品数が3の場合

1 番目の部品を C_1 , 2 番目の部品を C_2 , 3 番目の部品を C_3 , n 番目の部品の左上の X 座標を $P_{x,n}$, Y 座標を $P_{y,n}$, 右下の X 座標を $Q_{x,n}$, Y 座標を $Q_{y,n}$ としたとき、表 20, 21, 22, 23 で示す IDS を返す。

4.2.4 部品数が4の場合

1 番目の部品を C_1 , 2 番目の部品を C_2 , 3 番目の部品を C_3 , 4 番目の部品を C_4 , n 番目の部品の左上の X 座標を $P_{x,n}$, Y 座標を $P_{y,n}$, 右下の X 座標を $Q_{x,n}$, Y 座標を $Q_{y,n}$ としたとき、表 24, 25 で示す IDS を返す。

4.2.5 プロンプト

最初、

```
<image> Locate every component of the Chinese character.  
Report each component with bbox coordinates as TSV format like:  
X0 Y0 X1 Y1 component position (left/right/above/below/surround-from-upper-left/...)
```

というようなプロンプトを用いて実験を行った。ここでは position の括弧内に書いた相対位置を示すシンボルの説明が何もないが、これでもある程度期待通りの結果を得ることができた。しかしながら、「傾」や「亭」のように縦横に3つ以上の部品が組み合わさっているようなケースでは外側だけを認識して中身を見捨てるケースがあった。また、文字としての使用頻度が低い部品はうまく認識できなかった。

そこで、

```
component_examples_of_surround_from_upper_left = '厂(厚), 厂(斥), 尸(后), 尸(危), ...'  
のように、相対位置を示すシンボルの例とそれを用いた例を列挙し、  
position_prompt_en = f'position (left(e.g.{component_examples_of_left})/...)'  
のようなプロンプトの言語毎の位置指定用変数に格納し、  
prompt_en = f'''<image> Locate every component of the Chinese character.  
Report each component with bbox coordinates as TSV format like:  
X0 Y0 X1 Y1 component {position_prompt_en}  
'''
```

のような言語毎のプロンプト変数を構成することにした。

これにより、これらの問題はある程度改善することができたものの、処理速度が激増した。また、やはりうまくいかないケースも残った。

そこで、処理速度と認識精度の兼ね合いを考慮し、最初は例示しないプロンプトを用い、リトライ時には例示するプロンプトを用いることにした。

C_1	C_2	IDS
surround-from-upper-right	middle	$\boxed{\square} C_1 C_2$
surround-from-upper-right	enclosed	$\boxed{\square} C_1 C_2$
surround-from-upper-right	lower-left	$\boxed{\square} C_1 C_2$
surround-from-upper-right	below	$\boxed{\square} C_1 C_2$

表 18 surround-from-upper-right + x の時の生成規則

C_1	C_2	IDS
surround-from-lower-right	middle	$C_1 C_2$
surround-from-lower-right	enclosed	$C_1 C_2$
surround-from-lower-right	upper-left	$C_1 C_2$

表 19 surround-from-lower-right + x の時の生成規則

C_1	C_2	C_3	IDS
upper	middle	lower/below(= C_2)	$\boxed{\square} C_1 \boxed{\square} C_2 C_2$
upper	middle	lower/below	$\boxed{\square} C_1 C_2 C_3$
upper	lower-left	lower-right	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
upper	surround-from-above($\square$)	middle	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
upper	surround-from-above(上記以外)	middle	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
above	middle	lower/below(= C_2)	$\boxed{\square} C_1 \boxed{\square} C_2 C_2$
above	middle	lower/below	$\boxed{\square} C_1 C_2 C_3$
above	below	below(= C_2)	$\boxed{\square} C_1 \boxed{\square} C_2 C_2$
above	below	below	$\boxed{\square} C_1 C_2 C_3$
above	enclosed	below	$\boxed{\square} C_1 C_2 C_3$
above	sandwiched-from-above-and-below	below	$\boxed{\square} C_1 C_2 C_3$
above	lower-left	lower-right	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
above	left	right 但し、 $Q_{x,1} < P_{x,3}$	$\boxed{\square} \boxed{\square} C_1 C_2 C_3$
above	left	right 但し、上記以外	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
above	surround-from-above($\square$)	middle/enclosed	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
above	surround-from-above(上記以外)	middle/enclosed	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$
above	surround-from-upper-right	middle/enclosed	$\boxed{\square} C_1 \boxed{\square} C_2 C_3$

表 20 above + x + y の時の生成規則

C_1	C_2	C_3	IDS
upper-left	upper-right	lower	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
upper-left	upper-right	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
upper-left	upper-right	lower-right 但し、 $P_{x,3} < Q_{x,1}$	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
upper-left	right	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
upper-left	middle	lower-right	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
upper-left	middle	lower-left	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
upper-left	full-surround(門)	enclosed	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$
upper-left	full-surround(上記以外)	enclosed	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$

表 21 upper-left + $x + y$ の時の生成規則

C_1	C_2	C_3	IDS
left	upper-right	lower-right	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$
left	right	lower-right	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$
left	above	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$
left	right	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$
left	full-surround(□/□)	enclosed	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$
left	full-surround(門/門/凡/...)	enclosed	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$
left	full-surround(これら以外)	enclosed	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3$

表 22 left + $x + y$ の時の生成規則

C_1	C_2	C_3	IDS
lower-left	upper-right	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 C_3$

表 23 lower-left + $x + y$ の時の生成規則

C_1	C_2	C_3	C_4	IDS
upper	lower-left(= C_1)	lower-right(= C_1)	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3 C_4$
upper	lower-left	lower-right	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3 C_4$
above	lower-left(= C_1)	lower-right(= C_1)	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3 C_4$
above	lower-left	lower-right	below	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 \begin{smallmatrix} \square & \square \end{smallmatrix} C_2 C_3 C_4$
above	below	lower-left	lower-right	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 \begin{smallmatrix} \square & \square \end{smallmatrix} C_3 C_4$
above	above	surround-from-above	enclosed	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 \begin{smallmatrix} \square & \square \end{smallmatrix} C_3 C_4$
above($\nwarrow$)	(□)	($\nearrow$)	(任意)	$\begin{smallmatrix} \square & \square \end{smallmatrix} C_1 C_2 \begin{smallmatrix} \square & \square \end{smallmatrix} C_3 C_4$

表 24 above + $x + y + x$ の時の生成規則

C_1	C_2	C_3	C_4	IDS
upper-left	upper-right	surround-from-above	enclosed	$\boxminus\boxminus C_1C_2C_3C_4$
upper-middle	lower-left	lower-right	below	$\boxminus\boxminus C_1\boxminus C_2C_3C_4$
surround-from-upper-left($\boxminus$)	upper-right	lower-left	lower-right	$\boxminus C_1\boxminus C_2\boxminus C_3C_4$
surround-from-upper-left(上記以外)	upper-right	lower-left	lower-right	$\boxminus C_1\boxminus C_2\boxminus C_3C_4$
surround($\boxminus$)	enclosed	enclosed	enclosed	$\boxminus C_1\boxminus C_2C_3C_4$
surround($\boxplus$)	enclosed	enclosed	enclosed	$\boxplus C_1\boxminus C_2C_3C_4$

表 25 他の4引数の時の生成規則

一方、プロンプトの言語による差も見受けられた。結論から言えば、プロンプトを英語で書いた場合、部品を簡体字化して出力するケースが多発した。繁体字の中国語にした場合も簡体字で出るケースがあり、一番簡体字が少なくなったのはプロンプトを日本語をした場合であった。そこで、プロンプトの言語としては基本的に日本語を用いることとした。

5 実験

漢字字体規範史データセットの開成石經論語（開成論語）⁵⁾ と今西本妙法蓮華經卷五（宮廷今西）⁶⁾ を用いて実験を行った。

正解の判定基準としては UCS 統合漢字における事実上の包摂規準といえる IRG Working Document Series (IWDS) 2: UCS Ideograph Unifiable Component Variations Summary List (UCV) を用い、部品に分解できず代わりに（この観点で）正しい文字を出したのも正解とみなした。また、部品が食い込むなどして形式的に IDC が異なってしまったものの、常識的に見て同じ文字とみなせるものや書き方によって揺れそうなものも正解とみなした。

また、同様に異体字も正解とみなした場合の指標も提示する。

この2つの基準に基づき、

- mlx-community/Qwen3.5-27B-heretic-8bit
- froggeric/Qwen3.6-27B-Uncensored-Heretic-v2-MLX-6bit
- froggeric/Qwen3.6-27B-Uncensored-Heretic-v2-MLX-8bit

を用いた場合の認識結果を測定した値を表 26 に示す。

5) https://gitlab.nijl.ac.jp/HNG/hng-cards_18_kar

6) https://gitlab.nijl.ac.jp/HNG/hng-cards_10_khi

モデル	対象	正解率 (%)	適合率 (%)	再現率 (%)	F 値 (%)
Qwen3.5-27B-8bit	開成論語	72.2 (75.7)	74.0 (76.2)	89.8 (93.4)	81.1 (83.9)
Qwen3.6-27B-6bit	開成論語	69.7 (70.6)	69.9 (70.4)	89.4 (90.5)	78.4 (79.2)
Qwen3.6-27B-8bit	開成論語	66.1 (66.7)	64.2 (64.7)	89.2 (90.0)	74.7 (75.3)
Qwen3.5-27B-8bit	宮廷今西	54.0 (63.9)	52.5 (61.1)	85.1 (94.4)	64.9 (74.2)
Qwen3.6-27B-6bit	宮廷今西	54.4 (64.0)	52.9 (60.6)	85.5 (96.0)	65.3 (74.3)

表 26 認識結果

6 おわりに

視覚（マルチモーダル）言語モデル Qwen 3.5 / 3.6 27B を用いて、漢字画像の漢字構造解析を試みた。視覚言語モデルから直接 IDS を出力するのは現状厳しいため、今回は、部品の矩形座標と文字における大まかな位置情報を示すシンボルを出力させ、そこからルールベースで IDS に変換する手法を採った。これにより、部品や漢字構造に関する追加学習なしの状態であっても、部品の認識や漢字構造の解析、そして IDS の出力をある程度の精度で実現することができた。しかしながら、視覚言語モデルは本質的に字体を捨象して字種にまとめることで OCR 処理を遂行していると考えられるため、今回の手法では本質的な意味で字体情報そのものを認識できているとはいえないと考えられる。しかしながら、部品情報を用いて誤認識をリカバーする可能性は認められ、また、外字集合等から IDS 経由で UCS とのマッピングをとるなどの目的においてはそれほど問題はないかもしれない。

なお、今回は視覚言語モデルからの出力を

X_1, Y_1, X_2, Y_2 , 部品 (文字), 相対位置シンボル

のような部品毎の矩形座標と相対位置情報としたが、もしかすると、

X_1, Y_1, X_2, Y_2 , 部品 (文字), 中間部品との位置関係, 中間部品 (文字)

のようにある部品を含む文字もしくは中間部品との関係を明示するような形式を採った方が良かったかも知れない。こうすれば、視覚言語モデルが部品の文字列表現を知らない場合に「■」などで出力するように指示した上で、例えば、「榮」の上の部分の「𦵏」の文字列表現を知らなくても、

1,1,238,190, ■,above, 榮

のように出力させることができ、別途 CHISE 文字オントロジー等から部品の情報を補完するという手法が取れるかも知れない。

また、視覚言語モデルからの出力の判定においても、CHISE の漢字構造情報が利用で

きた方がより精度が高く効率的な処理が可能であると考えられる。こうしたことから、将来的に（現在実装を進めている IPLD/DASL 版）CHISE の Python 用実装の実現を行うことを計画している。また、部品の追加学習に関しても今後の課題としたい。

参考文献

- [1]守岡 知彦. “通時的な多粒度漢字構造情報記述の試み”. In: **情報処理学会論文誌** 65.2 (Feb. 2024), pp. 340–347.
- [2]Qwen Team. *Qwen3.5: Accelerating Productivity with Native Multimodal Agents*. Feb. 2026. URL: <https://qwen.ai/blog?id=qwen3.5>.
- [3]守岡 知彦. “Common Lisp 版 CHISE 実装について”. In: **東洋学へのコンピューター利用 第 38 回研究セミナー**. July 2024, pp. 63–84.