古典中国語 ModernBERT を用いた『古事類苑』引用書籍への返点付与

久保 旭 *

1　はじめに

『古事類苑』は明治時代に編纂された類書であり，国文学・歴史・法制・宗教・芸能など，広範な分野にわたる古典籍からの引用を体系的に分類・収録した資料である．明治政府の国学振興政策の一環として，宮内省図書寮が中心となって 1879 年から編纂が開始され，全 1000 巻の文献集として完成した．その構成は，部を頂点とした階層構造をもつ項目と，項目に紐づく解題及び複数の引用書籍からなる．

現在に至るまで『古事類苑』は様々な形態によってデジタル化及び公開が行われてきた．2025 年 7 月 1 日時点で広く一般公開されているデジタル化済み資料のうち主要なものを，次に示す．

1. 国立国会図書館デジタルコレクション（国立国会図書館）[*1]
2. 古事類苑データベース（国文学研究資料館）[*2]
3. 古事類苑ページ検索システム（国際日本文化研究センター）[*3]
4. 古事類苑全文データベース（国際日本文化研究センター）[*4]

この中で最も詳細なテキスト校正が行われているのは古事類苑データベースであり，校正段階においては，誌面と翻刻テキストとの照合作業が行われている [1]．その成果は古事類苑全文データベース（以下，古事類苑 DB という）に反映されている．古事類苑 DB には現在も新たな校正済みテキストが随時追加されており，2025 年 7 月 1 日時点で 30 部のうち 15 部が公開されている．

古事類苑 DB は 2025 年 4 月 2 日以降の公開分について，返点を入力しない方針への変更が行われた．この方針変更の背景については，過去にも 2021 年 4 月 13 日公開分から外字の作成を行わない方針への転換が見られることから，校正作業の負担軽減や公開スピードの向上を優先する方針が継続的に採られてきたことが示唆され，本変更もそうした方針の延長線上にあると考えられる．当初の古事類苑のデータベース化においては，原資料の情報を可能な限り忠実に再現することが重視されていたことが伺える [2]．しかし，全文公開の実現には膨大な作業量と時間を要することから，近年ではまず本文全体の公開を優先し，細部の再現は段階的に進めるという実務的な判断がなされている可能性がある．こうした方針変更や，校正作業に伴う人的負担を踏まえると，返点を含むテキストの再構築を自動化する技術の必要性が高まっているといえる．

返点付与を行うアプローチとしては，OCR を用いて高精度な返点認識を行うことが考えられる．返点認識に対応している OCR システムとしては NDL 古典籍 OCR[*5]が挙げられるが，古事類苑に対しては返点が欠落する場合があ

* 京都大学人文科学研究所

*1 https://dl.ndl.go.jp/pid/2609908（和装本），https://dl.ndl.go.jp/pid/1873261（洋装本）など，複数の版が存在する．

*2 システムの統廃合によって研究データが https://kokubunken.repo.nii.ac.jp/records/4735 に移行されている．

*3 https://lapis.nichibun.ac.jp/kojiruien/

*4 https://ys.nichibun.ac.jp/kojiruien/

*5 https://github.com/ndl-lab/ndlkotenocr_cli

る．天部天篇名稱條 [3] において引用されている日本書紀の OCR 結果を**図 1** に示す．この現象は認識モデルの学習に用いたデータと古事類苑との差異に起因しているものと考えられる．したがって，新たに OCR 用データセットを構築して再学習を行うことで古事類苑に特化した OCR の実現が期待できるが，誌面の各要素の領域，役割，テキストといった要素をアノテーションする必要があり，データセットの構築は容易ではない．

そこで，本研究では，引用書籍に本来含まれていたと推定される返点を白文から自動的に復元することを目的とする．具体的には，白文に返点を付与する処理を系列ラベリングタスクとして機械学習することで，既に公開されている古事類苑 DB の返点付きテキストをデータセットとして用いることを可能にする．また，古典中国語 ModernBERT を用いることによって漢文への最適化を行い，モデルの文字長制約に起因する問題を回避する．古典中国語 ModernBERT は最大 8192 トークンの入力幅に対応しており，長文構造を持つ古典文献に対する系列ラベリングに適していると考えられる．従来の BERT や RoBERTa は最大 512 トークンであることから，古事類苑に含まれる長文の引用書籍に対して有効に働くことが期待される．

2 関連研究

返点付与の研究としては，依存構造解析の結果とルールを組み合わせた手法が提案されている [4]．また，読み順の推定の研究としては，BERT 系の言語モデルを用いて文字単位のランクを学習し，それを用いて読み順を推定する手法が提案されている [5]．本研究の手法に近いものとしては，文字に加えて品詞などの素性を用い，条件付き確率場によって返点の推定を行う手法が提案されている [6]．本研究は，返点を白文に対する系列ラベリングタスクとして捉え，文のトークン列だけを用いて学習を行っている点が関連研究と異なる．

3 提案手法

白文の文字列を

$$X = (x_1, x_2, \ldots, x_n)$$

とする．ここで，x_i は漢字 1 文字を表す．

ここで，返点はその直前の文字に対して適用されているものであると見なすと，返点のラベル列は

$$Y = (y_1, y_2, \ldots, y_n)$$

と表すことができる．ラベル y_i は，x_i の直後に付与される漢文用記号に対応し，返点が存在しない場合は `O`（Outside）とする．返点が付与される場合は，BIO2 形式に従って `B-`**ラベル名**を用いる[*6]．

漢文用記号とラベル名[*7]の対応を**表 1** に示す．一レ点など，複数の漢文用記号によって構成される返点は，`One_Reverse` のようにラベル名をアンダースコアで連結し，新たに単独のラベル名を生成する．

入力 X に対し，最適な Y^* を出力する問題は系列ラベリングタスクと見なすことができ，スコアを返す関数 s を用いて次式のように表すことができる．

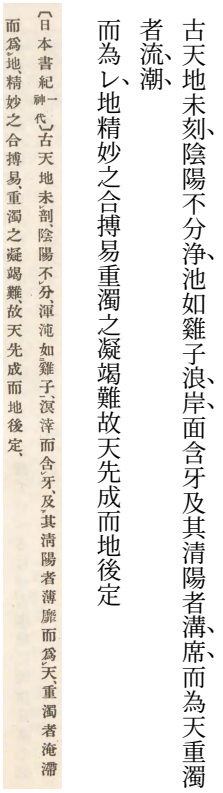
$$Y^* = \arg\max_{Y} s(X, Y)$$

スコア関数 $s(X, Y)$ の構築に当たっては，事前学習済みの言語モデルである古典中国語 ModernBERT[*8]のトークン分類モデルを用いる．このモデルは古典中国語に特化して事前学習されており，最大 8192 トークンの入力に対応可能であることから，古事類苑に含まれる長文の引用書籍に対しても有効に機能するものと考えられる．

[*6] このモデルでは単文字に対してラベルが付与されるので，`I`（Inside）は発生しない．

[*7] ラベル名は，Unicode における漢文用記号の文字名に由来している．

[*8] `https://huggingface.co/KoichiYasuoka/modernbert-{small,base,large}-classical-chinese`



古天地未刻、陰陽不分浄、池如雞子浪、岸面含牙及其清陽者溝、席而為天重濁者流、潮、而為レ地精妙之合搏易重濁之凝竭難故天先成而地後定

図 1　NDL 古典籍 OCR (ver.3) による日本書紀の OCR 結果. 字体の差異を除き，誤認識であると見なせる箇所を傍点で示す.

表 1　漢文用記号とラベル名の対応

記号	ラベル名
㇐	Link
レ	Reverse
一	One
二	Two
三	Three
四	Four
上	Top
中	Middle
下	Bottom
甲	First
乙	Second
丙	Third
丁	Fourth
天	Heaven
地	Earth
人	Man

```
古　O
天　O
地　O
未　B-Reverse
剖　O
陰　O
陽　O
不　B-Reverse
分　O
渾　O
沌　O
如　B-Two
雞　O
子　B-One
溟　O
涬　O
而　O
含　B-Reverse
牙　O
及　B-Bottom
```

図 2　CoNLL-2003 形式によるデータ例

具体的には，入力系列 X に対して ModernBERT によって各トークンの隠れ状態ベクトルを取得し，それらに対して線形層と softmax を適用することによって，各トークンに対するラベルの確率分布を得る．スコア $s(X, Y)$ は，各トークン x_i に対する予測確率 $p(y_i \mid x_i)$ の対数和として定義される．

$$s(X, Y) = \sum_{i=1}^{n} \log p(y_i \mid x_i)$$

なお，条件付き確率場（Conditional Random Fields）層は用いていない．これは，提案手法が文字単位でラベルを付与していることから，隣接するラベル間において BIO の整合性が問題となるケースが発生しないことによる．

モデルの入出力形式として CoNLL-2003 形式を採用する．これは，各文字に対して 1 行ずつラベルを付与し，文の区切りを空行で表す構造である．**図 1** 冒頭部分の CoNLL-2003 形式によるデータ例を**図 2** に示す．

4　実験

本節では，提案手法の有効性を検証するために行った実験について述べる．

4.1 実験設定

実験に用いるデータは，次の手順で作成する．

1. 引用書籍本文を古事類苑 DB から取得する[*9]．
2. ひらがな，カタカナが含まれている引用書籍本文を除外する．
3. 句読点，ルビ，傍点を削除する．
4. NFC 正規化を行う．
5. 1 文字単位で分割する．
6. 返点ラベルを BIO2 形式で付与する．
7. 外字，欠字を `[UNK]` に変換する．
8. 改行[*10]を `[SEP]` に変換する．

上記の処理の結果，48,050 件，4,168,415 トークンの漢文返点データセットが得られた．漢文返点データセットのラベル名とトークン数を**表 2** に示す．

ファインチューニングは Hugging Face Transformers の Trainer API を用いて行う．ファインチューニング時のパラメータを，次に示す．

- 事前学習済みモデル：`modernbert-small-classical-chinese`
- 最大エポック数：5
- 学習率：2e-5
- デバイス当たりのバッチサイズ[*11]：8, 16
- Warmup Ratio：0.2
- Weight Decay：0.01
- オプティマイザ：`adamw_bnb_8bit`
- 評価の間隔: 500 ステップ
- Early Stopping Patience：10

モデルの性能評価指標には，固有表現抽出などの系列ラベリングタスクで一般的に用いられる適合率（Precision），再現率（Recall），及び F_1 値（F1-score）を用い，各ラベル $l_i \in L$ に対しては次式で定義される．

$$P_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}$$

$$R_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}$$

$$F_1^{(i)} = 2 \cdot \frac{P_i \cdot R_i}{P_i + R_i}$$

ここで，TP_i（True Positive）はモデルが l_i と予測し，実際にも l_i であったトークン数，FP_i（False Positive）はモデルが l_i と予測したが，実際には異なるラベルであったトークン数，FN_i（False Negative）は実際にはラベル l_i であったが，モデルが異なるラベルと予測したトークン数を指す．

[*9] 本研究で用いる引用書籍本文は，古事類苑 DB の公開データに対してアノテーション修正や誤字修正を部分的に追加で行ったものである．したがって，公開データとは内容に差異が生じることがある．

[*10] `<br>`要素又は改行コード

[*11] 実験では，GPU を 2 デバイス使用して並列化を行っている．

表 2　漢文返点データセットのラベルとトークン数

ラベル名	トークン数
`Link`	0
`Reverse`	173,682
`One`	252,582
`Two`	256,168
`Three`	3,945
`Four`	67
`Top`	5,311
`Middle`	1,791
`Bottom`	6,020
`First`	123
`Second`	140
`Third`	90
`Fourth`	30
`Heaven`	1
`Earth`	1
`Man`	0
`One_Reverse`	66
`Two_Reverse`	2
`Three_Reverse`	2
`Top_Reverse`	14
`Bottom_Reverse`	1
`Top_Two`	1
`Bottom_Two`	1
`Top_Three`	1
`O`	3,468,376
合計	4,168,415

複数ラベルに対する評価指標である全体の F 値は，次式で定義されるマイクロ平均 F_1^{micro}，マクロ平均 F_1^{macro}，重み付き平均 F_1^{weighted} の 3 種類を用いる．

$$P^{\text{micro}} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FP}_i}$$

$$R^{\text{micro}} = \frac{\sum_i \text{TP}_i}{\sum_i \text{TP}_i + \sum_i \text{FN}_i}$$

$$F_1^{\text{micro}} = 2 \cdot \frac{P^{\text{micro}} \cdot R^{\text{micro}}}{P^{\text{micro}} + R^{\text{micro}}}$$

$$F_1^{\text{macro}} = \frac{1}{|L|} \sum_{i=1}^{|L|} F_1^{(i)}$$

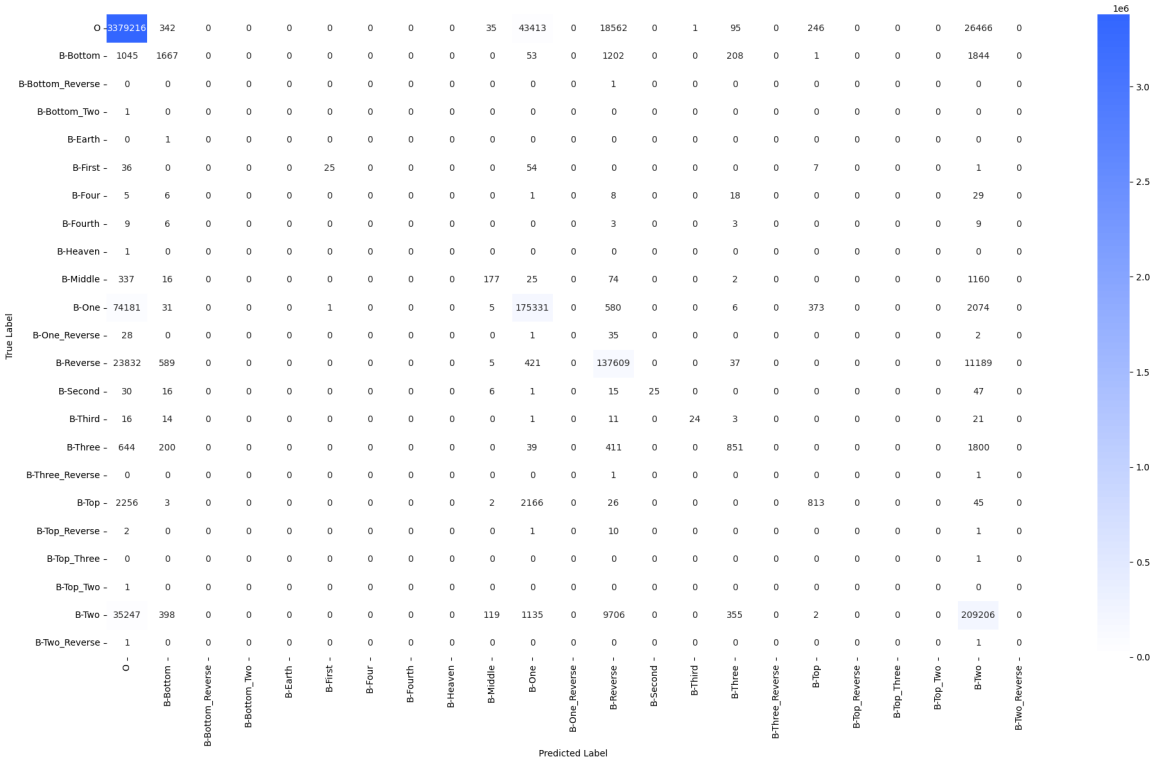


図 3　`small` モデル（バッチサイズ 8）のテストセットにおける混同行列

$$F_1^{\text{weighted}} = \frac{\sum_{i=1}^{|L|} \text{support}_i \cdot F_1^{(i)}}{\sum_{i=1}^{|L|} \text{support}_i}$$

ここで，$|L|$ はラベルの総数，support_i は l_i に対する実際の出現数を表す．

なお，適合率と F_1 値において分母がゼロになる場合は値を 0 とする．

学習時には 5-fold cross validation を行い，各 fold において最も高い F_1^{macro} を達成したチェックポイントを用いてテストセットに対する評価指標を計算する．

4.2　実験結果

全体の F_1 値を**表 3** に，ラベルごとの結果を**表 4** に示す．全体的にスコアが高いとは言えないが，傾向として，トークン数が多いラベルほど高い F_1 値となっていることが分かる．また，パラメータの選定については，`small` モデルにおいてはバッチサイズが小さい方が F_1 値が高いことが分かる．

また，各ラベルにどのような誤り傾向があるかを分析するために，バッチサイズ 8 のモデルを用いたテストセットの推定結果について，**図 3** に示す混同行列（Confusion Matrix）を作成した．対角線上の要素以外は誤って推定されたことを示しており，トークン数が多いラベルにおいて `O` と誤って推定したり，`O` をほかのラベルに誤って推定している事例が多いことが分かる．また，レ点と二点について，互いに誤って推定している事例が多いことが分かる．両者は相対的にトークン数が多く，F_1 値も比較的高いが，学習時に両者の文脈的な違いが捉えられていない可能性がある．

表 3　全体の F_1 値

モデル	バッチサイズ	micro			macro			weighted		
	/デバイス	適合率	再現率	F_1 値	適合率	再現率	F_1 値	適合率	再現率	F_1 値
small	8	0.8069	0.7510	0.7780	0.3394	0.1680	0.2042	0.8020	0.7510	0.7733
small	16	0.8026	0.7367	0.7682	0.3527	0.1582	0.1928	0.7979	0.7367	0.7623

表 4　ラベルごとの結果

ラベル	small (バッチサイズ 8)			small (バッチサイズ 16)			トークン数
	適合率	再現率	F_1 値	適合率	再現率	F_1 値	
Reverse	0.8179	0.7923	0.8049	0.8072	0.7887	0.7978	173,682
One	0.7875	0.6942	0.7379	0.7857	0.6703	0.7235	252,582
Two	0.8240	0.8167	0.8203	0.8188	0.8065	0.8126	256,168
Three	0.5393	0.2157	0.3082	0.5509	0.2195	0.3139	3,945
Four	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	67
Top	0.5635	0.1529	0.2405	0.6044	0.0823	0.1448	5,311
Middle	0.5072	0.0988	0.1654	0.6598	0.0357	0.0678	1,791
Bottom	0.5068	0.2769	0.3581	0.5315	0.2286	0.3197	6,020
First	0.9615	0.2033	0.3356	1.0000	0.2033	0.3378	123
Second	1.0000	0.1786	0.3030	1.0000	0.1786	0.3030	140
Third	0.9600	0.2667	0.4174	1.0000	0.2667	0.4211	90
Fourth	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	30
Heaven	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1
Earth	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1
One_Reverse	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	66
Two_Reverse	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	2
Three_Reverse	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	2
Top_Reverse	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	14
Bottom_Reverse	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1
Top_Two	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1
Bottom_Two	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1
Top_Three	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1

5　おわりに

本研究では，白文に返点を付与する処理を系列ラベリングタスクとして定式化し，古典中国語 ModernBERT を用いてファインチューニングを行い，その精度評価と誤り傾向の分析を行った．

今後の課題として，より大規模なモデルを用いた学習，依存構造の利用などによる精度向上が挙げられる．また，提案手法では複数の漢文用記号の合成は独立したラベルとして定義したが，マルチラベル分類として学習することも考えられる．実験では和文中に部分的な引用などの形で含まれる漢文については除外したが，そのような出現に対しても対応が必要である．実利用においては長文入力時に多くの GPU メモリを必要とする点も課題であり，GPU 資

源が制限された環境への対応についても検討していく必要がある．

また，本研究におけるデータセット構築の過程において，古事類苑 DB に残存している誤字のごく一部を修正したが，網羅的な確認や修正には至っていない．返点付与の精度向上だけでなくデータベースそのものの品質向上にも資することから，機械学習を用いた誤字検知や誤字訂正も重要な課題である．

参考文献

[1] 山田奨治, 早川聞多, 相田満. 古事類苑 (天部・地部) の全文入力と Wiki 版の試行-前近代の文化概念の情報資源化. 情報処理学会研究報告人文科学とコンピュータ (CH), Vol. 2006, No. 112 (2006-CH-072), pp. 39–46, 2006.

[2] 相田満. 和漢古典学のオントロジ. 勉誠出版, 2007.

[3] 神宮司庁. 天部一 天篇 名稱. 古事類苑, 第 2 巻, pp. 1–8. 古事類苑刊行会, 1928.

[4] 安岡孝一. 漢文の依存文法解析と返り点の関係について. 日本漢字学会第 1 回研究大会予稿集, pp. 33–48, 2018.

[5] 王昊, 清水博文, 河原大輔. 言語モデルを用いた漢詩文の返り点付与と書き下し文生成. 自然言語処理, Vol. 31, No. 1, pp. 134–154, 2024.

[6] 佐藤綾花. CRF を用いた漢文の返り点推定. 愛知県立大学情報科学部情報科学科 2013 年度卒業論文要旨, 2014.