訓点資料の加点情報計量のためのデータ構造

－国立国語研究所蔵「尚書（古活字版）」を対象として－

林　昌哉，田島　孝治（岐阜工業高等専門学校），
堤　智昭（東京電機大学），高田　智和（国立国語研究所），小助川貞次（富山大学）

本稿では，訓点資料の加点情報の計量と書き下し文の機械的な生成を目的として，資料に付与されたヲコト点などの加点情報を電子化する手法を検討した結果について述べる．具体的には「尚書(古活字版)」の 1 丁を提案した構造化方式で電子データとして記述し，統計処理を行った．1 丁分の結果ではあるが，文字の頂点部に「ニ」や「テ」を表すヲコト点が集中していることが分かった．それ以外にも，加点の総数や位置の分布など基礎的な計量が行えることが確認でき，提案したデータ構造の有用性と課題が明らかになった．

A Data Structure and Basic Statistics for Gloss on Chinese Text

—A Case Study of Syousyo —

Masaya Hayashi, Koji Tajima (NIT, Gifu College)
Tomoaki Tsutsumi (Tokyo Denki University)
Tomokazu Takada (National Institute for Japanese Language and Linguistics)
Teiji Kosukegwa (University of Toyama)

This paper describes a data structure for digitizing the gloss on the classical Chinese text. The data structure considered for the statistic for gloss and automatically vernacular reading text generation. Specifically, we made digital data of one page of "Syousyo (old type version)" and measured the number of wokototen and phonogram glosses. As the result, most of the wokototen appears the corner of a letter. It means "ni" and "te". Additionally, we measured the total number of glossing and distribution of the position. We clarified the availability and problems of the proposed data structure.

１．まえがき

訓点資料の分析は，記述内容の正確な理解を目指し，加点内容を理解することを中心に行われてきた．これまでに，ヲコト点図や釈文の利用により，資料に付与された加点などによる注釈を解釈する方法は確立されている．この結果，現在では訓点資料の現代語訳を容易に手に入れることができる．

一方で，現代語訳の作成に際しては，訓点資料から，書き下し文を一度作ってから解釈する作業が行われることが多い．書き下し文は，主に古典中国語で書かれた原文を日本語で読めるように，語順の調整や助詞・助動詞，読み方などを補って作られた文である．どのように原文を解釈するかは，ヲコト点や注釈により記されている．このため，原文から適切な書き下し文を作成するには，文献に関する知識に加え，ヲコト点と訓読方法に対する深い理解が必要である．

ヲコト点を解釈するためには，ヲコト点図だけでは不十分である．これは，代表的な加点内容に関しては，ヲコト点図に対応付けが記されているものの，文献固有の対応付けが存在することも多く，経験や知識などにより意味を解釈しなければならないためである．

本研究では，この書き下し文を機械的に生成するために，資料に付与されたヲコト点などの加点情報を電子化する手法を検討している．これまでに著者らは，漢文に付与された加点内容に注目し，統計的な処理の実現可能な構造化記述方式を検討してきた [1]．この手法により電子化を行えば，ヲコト点と文字の関係を統計的に処理する，加点内容どうしを比較するなどの基礎研究が実現できる．また，文献固有の対応付けや，ヲコト点図の記述内容に対して，統計的な裏付けが行えるようになると考えられる．この実現は，書き下し文の自動生成に向けた第一歩であり，特定の文献全体を電子化し，そのデータを解析することで，記述方式の有用性を検証する必要がある．

本稿では，「尚書(古活字版)」の 1 丁を提案した構造化方式で電子データとして記述し，統計処理を行った結果について述べる．統計処理を通じて，これまでに提案してきたデータ記述方法では

不適合な部分が判明したため，これに関して改良を行った．この不具合と改良部分についても，本稿にまとめる．

2．対象とした資料について

今回は国立国語研究所蔵「尚書(古活字版)」を対象として電子化を行う．尚書は書経とも呼ばれ，政治史・政教を記した中国最古の歴史書で，序文と 58 の通篇で構成される [2]．今回電子データを作成した資料は 1596[慶長元]-1615[慶長 20]年刊のものであり，巻 1 から巻 9 までの画像データが公開されている．冊子本であるため，画像データは 1 丁に対し表裏が存在し，半丁あたり 8 行構成である．今回の電子化は巻 1 の本文の冒頭部分を対象とした．具体的には漢文本文が連続する関係で巻 1 の第 1 丁と第 2 丁の 1 行目まで(計 17 行)を対象にした．

3．統計処理のためのデータ構造

3.1　想定する利用用途

訓点資料を電子化する場合，訓点を取り除いた漢文本文のみの電子テキストとするか，入力者が訓点を解釈し書き下し文とする方法が取られていた．漢文本文のみのテキストデータでは，訓点の情報が抜け落ちてしまい，訓点研究には利用できない．書き下された文章は，特定の時代・流派の読みに従って解釈された結果が記述される．よって，異なる時代・流派の読みに従って，再度別の解釈を試みる場合は，原文を確認しながら，付与された点の位置をもとに新たに書き下し文を作成する必要がある．しかし，原文は，その保存状況などから容易に再確認できるものではなく，写真などで電子化されたものでは，本文は読み取れるものの，ヲコト点などを読み取ることが難しい場合も多く，加点情報の確認が困難である [3]．

そこで本研究では，加点情報の定量的な分析と，加点情報と解釈情報と組み合わせ，書き下し文を自動的に生成できるデータ構造を提案する．訓点資料の加点情報の構造化を行い電子データとして記述することで，資料に含まれる加点の総数や文字，点ごとの傾向を定量的に分析することが可能となる．総数や傾向を定量的に集計することで，加点の流派や時代による加点傾向の違いに対してこれまで立てられてきた仮説を検証できる．また，加点という技術が広まった際に，同一資料への複数回加点された例を調べれば，資料の伝搬経路を奥付以外の要素から検証できる可能性がある．

訓点資料の加点数などの計測は，過去に例がなく，一つ一つ手作業での集計することは効率的ではない．データを構造化して記述することによって，計算機を使った高速な集計が実現できる上，集計したい情報を選択して抽出できるようになる．さらに，構造化した文書を利用して，新たなテーマの研究に，別の研究者も再利用可能になる．これらの統計処理を行うことで，加点内容の解釈に対するルールの記述が実現でき，最終的に書き下し文の自動生成という目的を達成できると考えられる．

表 1　電子化する加点情報とその解釈

点の名称	加点の解釈	区分
科段点	段落	加点レベル A
句点	文	
読点	文節などの文中の区切り	
合符	語のまとまりや解釈（人名を表す符号も含む）	
声点	語のアクセント（語の解釈の確定）	加点レベル B
ヲコト点	助詞・助動詞など（一部読みも含む）	
仮名点	ふりがな，送り仮名など	

しかし，加点情報を保持し，条件付きで参照できる構造化は前例がなく，理論的な設計だけでは，実用性があるかわからない．本稿では尚書を電子化し構造化して記述することで，実際に集計ができるほどの実用性があるか検証するとともに問題点を明確にする．今回は，データ構造の実用性の検証として，加点情報の総数や，各加点の数，形状や，加点位置ごとの個数，仮名の種類などの基本的な集計を試みる．

3.2　構造化記述を試みる加点情報

資料への加点は，複数回，理解度の違う人間によって行われているものがある．写本作業を行う際に，加点を理解せずそのまま写し，後からその解釈を別の記述方法で再度書き足してある資料などでは，これが顕著である．また，一人の読者が本文の理解のために読んでいく作業においても，初めは段落の区切りなど単純なものを加点し，その後，語の読みや意味などの詳しい情報を追加していったと考えらえる資料も存在する．このように，繰り返すことで加点内容は増え，より詳細な情報が付与されていく．

今回の電子化する加点内容を表 1 のように定めた．資料への加点は日本固有の現象ではない．韓国やベトナムでも，その国の言語に合わせた加点された漢文資料が発見されている．そこで，これらとの比較を考え，言語に依らない分割区分を加点レベル A，言語依存性の高い区分を加点レベ

ル B としグループ化した.

3.3 これまでに提案したデータ構造

加点レベル A の要素は RFC7159 に準拠した軽量化オブジェクトである JSON 形式で記述する. 加点レベル A に属する要素は，段落，文とそれを表す句読点，語であり，各要素が持つパラメータは大きく異なる. そこで，任意のキーに対して値を関連付ける key-value 型のデータ構造にすることで，要素の異なりに対応することにした. また，今後の統計処理は各種のプログラム言語を利用して行う為，多数のプログラミング言語で標準的に読み込みが可能な点もこの形式を選択した理由である. 特に単純な集計処理は Python や JavaScript などのインタプリタで実行可能な言語で処理するため，これらの言語において命令一つでデータ構造を含めて一括して読み込める JSON は利便性が高い.

加点レベル A の key の必須要素は，type と id である. type として指定する値は，sentence, period, comma, word, paragraph を想定しており，それぞれ文，句点，読点，語，段落を表すために利用する. これ以外の key には，他のオブジェクトへの関連付けである sequence と charcount，付与されている符号の形状を表す mark，付与されている符号の位置を表す x,y がある. 加点位置を表す座標系を図 1 に示す. 赤線を文字とし，中心を原点とした座標系で表記した [4].

加点レベル B の要素は XML 形式により記述する. このデータ構造では 1 文字または 1 語を親要素とし，そこに情報を付与していく. 親要素は Letter または Word とし，Letter の場合は任意の 1 文字を表し，加点レベル A で定義した文字の ID を属性に持たせる. Word は加点レベル A で定義された語であり，こちらも ID が付与してあるため，同様に処理できる. また，これらのタグには文字そのものである Character を必須の子要素とし，加点内容は Annotation として任意の数だけ子要素にする. Annotation には Mark，LocationX，LocationY が必須の子要素となっており，形状と位置を表している.また，Annotation タグにも複数個の区別のために属性として ID を付与できるようにする. この形状は一文字に多くの付加情報が付与されてもその文字に関連付けて表記可能な形式であり，XML の持つ木構造という特性も活かせる.このデータの処理は，加点レベル A と同様に Python などのインタプリタで実行可能な言語で行う.

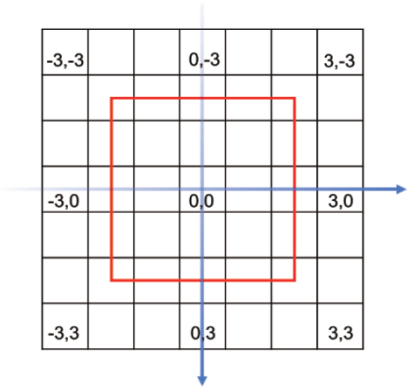

図 1 加点位置の座標系

3.4 尚書特有の要素に対する処理

尚書の電子化にあたり文字や加点には ID を設定する必要がある. 過去の ID は春秋経伝集解を対象とし，ページ数，行数，文字数の順で構成されていた. 本資料には巻数が存在し，常に丁数には表裏が存在するため，各要素を表す ID は「通篇-丁数(表裏)-行数-文字数」とした. さらに，本資料は一丁の表又は裏は全て 8 行，全て割注の小文字の最大 34 文字で構成されるため，3 桁存在した行数と文字数を 2 桁に変更した. 僅かではあるが，データ構造の適合性を調査するため，全ての情報を手入力で行われたデータ作成の負担を軽減する目的がある.

また，本資料は本文の構成が原文と割注を交互に記述されている. 割注は原文に比べ文字の大きさが小さいため，見た目の差は大きい. しかし，どちらも本文であり，同じように加点による Annotation が付与されているため電子データにおいては同等のものとして扱った.

本資料には本文，加点レベル A，B の他に多くの情報が存在する. 割注とは別に，筆で直接記入されたであろう本文横や上下の注釈，その注釈に付与されているヲコト点，目印として本文中に朱点で付与された校符などは今回のデータ構造化の対象から除外する. さらに本文上に点在する墨か虫食いによる汚れか判断しかねる点，裏移りか本資料自体に付着していると考えられる汚れもデータ構造には含まず，集計の対象とはしない.

3.5 加点レベル A のデータ構造

JSON 形式のデータ構造で記述した加点レベル A の電子データの一部を図 2 に示す. また資料のこれに対応する部分を図 3，4 に示す. 今回の電子化は段落要素 r002 には文要素 s009 と s010 が属するように設計したデータ構造であり，文字の集まりを文，文の集まりを段落，段落の集まりを篇として示せる. これまでに提案した加点レベル A の構成は最上位が段落による区切りだったが，尚書には全ての通篇に篇名が存在するため，各段落がどの通篇に属するかの篇要素を追加した. さらに，篇要素に篇名である title の key を作成し，将来的に篇名での検索や表示を達成できるように

```
{
    "type": "chapter",
    "id": "h001",
    "sequence": ["r001", "r002", "r003"],
    "title": "堯典"
}
```

(a)　篇要素

```
{
        "type": "paragraph",
        "id": "r002",
        "sequence": ["s009", "s010"],
        "mark": "●"
}
```

(b)　段落要素

```
{
        "type": "sentence",
        "id": "s001",
        "charcount": 4,
        "sequence": ["01-001A-03-01",
                     "01-001A-03-02",
                     "01-001A-03-03",
                     "01-001A-03-04"]
}
```

(c)　文要素

```
{
        "type": "period",
        "id": "p001",
        "char": "01-001A-03-04",
        "mark": "・",
        "x": 3,
        "y": 3
}
```

(d)　句点要素

```
{
        "type": "comma",
        "id": "c001",
        "char": "01-001A-03-06",
        "mark": "・",
        "x": 0,
        "y": 3
}
```

(e)　読点要素

```
{
        "type": "word",
        "id": "w001",
        "sequence": ["01-001A-03-01",
                     "01-001A-03-02"],
        "mark": "｜",
        "x": -3,
         "y": 3
}
```

(f)　語要素

図 2　加点レベル A の記述結果の一部

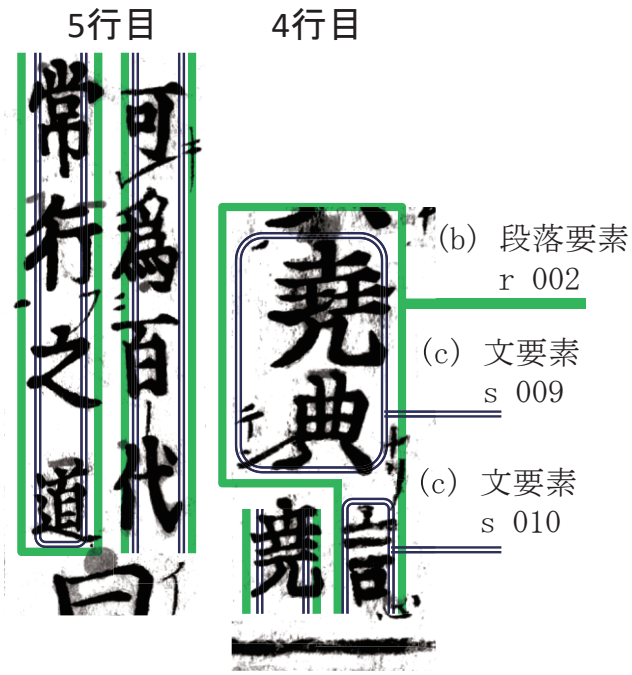

図 3　段落要素と文要素の対応箇所

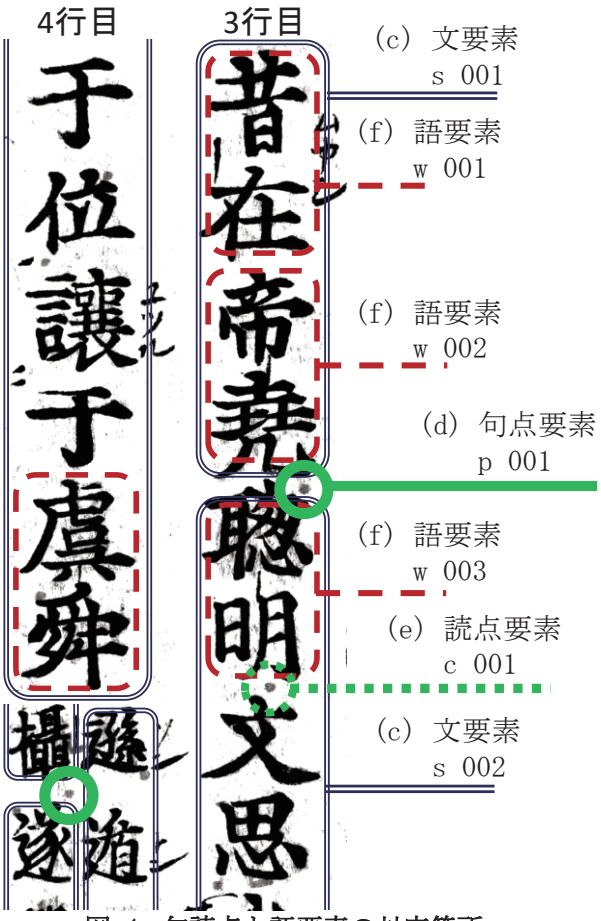

図 4　句読点と語要素の対応箇所

変更した． 句点，読点に関しては，星点などで表された要素を記述し，合符で表される語も独立した要素とした．これらは，ヲコト点とは分けて統計処理ができるようにするためである．

3.6 加点レベル B のデータ構造

尚書の第1丁に存在する加点レベルBの要素は主に仮名と記号で記述され，形状と機能で分類するとヲコト点，仮名点，声点の 3 種である．それぞれ形状や位置，読みがあり，ほぼ構成が同じで

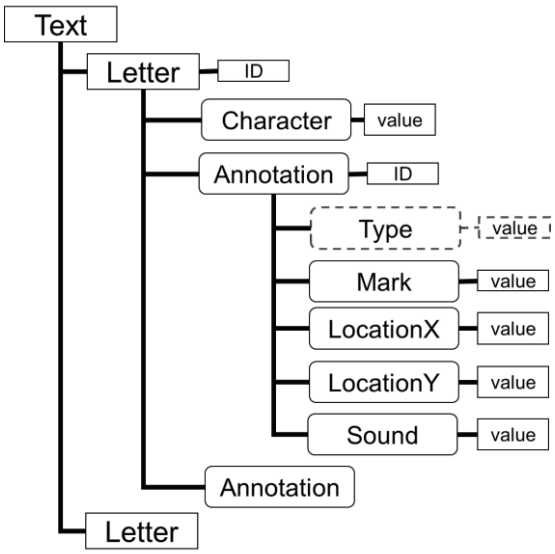

図 5 加点レベル B のデータ構造

```
<Letter id = "01-001A-03-10">
  <Character>宅</Character>
  <Annotation id = "0006">
    <Type>仮名点</Type>
    <Mark>シリ</Mark>
    <LocationX>3</LocationX>
    <LocationY>0</LocationY>
    <Sound>しり</Sound>
  </Annotation>
</Letter>

<Letter id = "01-001A-03-12">
  <Character>下</Character>
  <Annotation id = "0007">
    <Type>ヲコト点</Type>
    <Mark>・</Mark>
    <LocationX>-2</LocationX>
    <LocationY>-2</LocationY>
    <Sound>に</Sound>
  </Annotation>
</Letter>
```

図 6 加点レベル B の記述結果の一部

ある．よって構造を区別せず，Annotation の子要素として図 5 に示すようにもともと考えていたデータ構造に Type を追加し，そこにヲコト点や仮名点など加点の種類を入力した．このデータ構造により集計時に Type 別での計測も可能となるだけでなく，新たな Type が出現しても対応できる．本資料の XML データ構造化したところ，図 6 のような結果となった．

3.7 データ制作時に発生した問題

今回データの構造化記述に際しては，読みや区別の誤りを避けるために，漢文訓読に精通した研究者とともに，解釈を逐一確認しながらデータを入力していった．今回の範囲で最も複雑な解釈が必要であったヲコト点は，「言」の右下，(2,2)の位置に付与された星点である．このヲコト点をヲコト点図で調べると，読み方は「ハ」であるため，「言うは」と読みたくなる．しかし，今回の場合「言う心は」と発音するのが正解であり，初めて登場した「言」の文字には「心」という注釈が書かれている．このような解釈は漢文知識が浅い入力者にとっては難しく，正確なデータ作成の障害となりうる．

加点レベル A である文要素を作成する際にも問題が生じた．文は主に段落の区切りである科段点，本文と割注の境界，句点で区切ることができる．加えて，「〜である」の意味である「也」も意味としては文の終わりといえ，多くの場合句点が付与されている．しかし，本資料には「也」に付与された星点の位置が読点と解釈できるものや，句点と読点の中間の位置にあるものなどがあり，判別が難しい．よって，今回は「也」に対する点の位置に対する拘りが弱いと判断し，句点の有無に限らず「也」を意味的な文の終わりと決定した．

誤点と考えられる点も本資料には存在する．明らかに裏移りでも汚れでもなく，くっきりと句点の位置に付与されているが，文としてあり得ない構成になってしまう点がある．この点は誤点と判断し，データには加えなかったが，誤点であるかの判断は一度通しで文章を確認しない限り不可能である．さらに，加点者の書き間違えとも思われる，仮名点に似た読解不可能な点も存在する．同じ文字に正しい仮名点が付与されているため，読めない点は除外したが，これも漢文への知識なしでは判断の難しい点であると言える．

手作業によるデータの作成に膨大な時間がかかることも課題である．今回の範囲で入力した本文の文字数は 374 文字であるが，これに付与された情報の入力だけでも，作業時間は 4 日間の約 32 時間を要し，加点レベル B の XML だけでもコ

ードは 2831 行となった．手作業での入力であるため id の番号ずれが発生することもあり，大きく作業がロールバックすることも多々あった．

４．統計処理の手法と結果

4.1 各要素の個数

電子データの集計用プログラムを Python で実装し，制作したデータの統計処理を行った．プログラムはデータの総数を数えるだけでなく，これまでに集計したタグとの一致なども調べることができる．電子データは JSON または XML であるため，プログラミング言語が持つ標準の key-value 型のデータ構造を使えば容易に集計できる．今回，加点レベル B に関して(1)Annotation の Type 別総数，(2)位置別総数，(3)ヲコト点の形状別総数，(4)読み別総数を調べた．Annotation は Type で分別した後に，それぞれの Type ごとに細かく集計した．

基本データを調べたところ，今回電子化した範囲の Annotation が付与されている総文字数は 374 中 204 文字，段落数は 6，文の数は 45 だった．

(1) Annotation の Type 別総数を表 2 に示す．加点総数は 286 であり一文字に対し平均 0.76 個の Annotation が付与されていることが分かる．ヲコト点が一番多く，仮名点がそれに続く．声点は 1 丁目の表には存在せず，1 丁目の裏に集中して記入されていた．

4.2 ヲコト点に関する分析結果

(2)位置別総数に関して，ヲコト点のみをまとめた結果を図 7 に示す．特に文字の頂点部に集中し，文字の上下にも多く見受けられる．さらに，(-3,3)の点は返り点兼用の「テ」であり，明確に文字から離れているものが多くあった．(3,0)の点は音読みを表す線であり，訓読みを表す(-3,0)の線は一つもなかった．

(3)ヲコト点の形状別総数を表 3 に示す．星点が一番多く 152 個，続いて「└」が 10 個だった．星点にのみ注目して座標別の分布を作ったものを図 8 に示す．大きな傾向は図 7 と変わらないが，文字の上下に現れる点の数が少なくなっている．星点の配置は，文字の四隅と中央に偏っていることがより分かりやすい結果となった．

(4)読み別総数をヲコト点に注目すると「に」が 44 個，「て」が 28 個，「は」が 26 個，「を」が 19 個で，主要な助詞が占めていた．

4.3 仮名点に関する分析結果

仮名点は，ふりがな，送り仮名という二つの用途で使われているが，今回は区別せずに全てを仮名点として集計した．

表 2　Annotation の Type 別総数

加点の種類 (Type)	総数
ヲコト点	178
仮名点	105
声点	3
合計	286

表 3　ヲコト点の形状別分布

ヲコト点形状 (Mark)	総数
▪	152
└	10
—	8
|	6
┐	1
＼	1

Y＼X	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	**46**	0	**8**	0	**22**	0
-1	0	0	0	0	0	**1**	0
0	0	0	0	**11**	0	**1**	**5**
1	0	0	0	0	0	**5**	0
2	0	**17**	0	**19**	0	**30**	0
3	**15**	0	0	0	0	0	0

図 7　ヲコト点の座標分布

Y＼X	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	**44**	0	**2**	0	**20**	0
-1	0	0	0	0	0	**1**	0
0	0	0	0	**11**	0	0	0
1	0	0	0	0	0	**5**	0
2	0	**16**	0	**10**	0	**30**	0
3	**13**	0	0	0	0	0	0

図 8　星点に注目した座標別分布

表 4　仮名点の文字数に対する統計結果

文字数	延べ数	異なり数
1	34	15
2	36	30
3	14	14
4 以上	21	19

(2)位置別総数に関して仮名点に注目してまとめた結果を図 9 に示す．仮名点は 105 中 86 個が文字の右側に付与され，残り 19 個は全て左側に付与されていた．左側に書かれる場合は，割注の文字間隔の狭さなどから文字のあった位置によるものであった．ヲコト点と異なり文字の上下に現れることはない．また，文字の中央列に合わせて配置されることが多く，中央右に 60 個，中央左にも 12 個と多い．右下にも多く表れているが，そのほとんどは送り仮名として使われているものである．

(4)読み別総数を仮名点に注目すると，「シ」が一番多く 6 個，続いて「ル」「ク」が 5 個であった．仮名点の種類は多く，一番多い「シ」ですら 6 個であるため，個々の仮名について検討するには 1 丁分では個数が不足していると考えられる．一方，文字数ごとまとめてみると，表 4 のような結果が得られた．文字数は 1 文字，2 文字は均衡しているが，2 文字の仮名点はほとんど異なった種類であることが分かる．1 文字の仮名点も平均すると 2 回以上登場している計算になるが，実際の結果を細かく見てみると，1 文字で 1 回しか出てこない仮名点は 9 種類もあり，半数を超えている．

5．課題と考察

集計結果から，基本的な総数や位置の分布など基礎的な計量が行えることが確認できた．しかし，

Y＼X	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	**2**
-2	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	**1**
0	**12**	0	0	0	0	0	**60**
1	**1**	0	0	0	0	0	0
2	**2**	0	0	0	0	0	**3**
3	**4**	0	0	0	0	0	**20**

図 9　仮名点の座標別分布

今回の集計を行ったところ，現状の構造では必要な情報が一部不足していることが明らかになった．まず，仮名点の表現においては，送り仮名とふりがなに細分化する必要がある．単純に Type を二つに分けるだけでは，仮名点という表記的な属性がまとまっておらず望ましくない．仮名点の下に新しく属性を増やす必要がある．

さらに，加点レベル B の XML という構造は，文字を親とし，加点内容を Annotation として追加しているため，特定の Annotation を検索キーとして文字を収集するという用途に適していない．例えば文字の右上に星点が記されやすい漢字を統計的に調べたい場合，本文すべての文字について調べなければ現在のデータ構造では結果が得られない．

今回の集計プログラムにおいては，XML のタグを key，登場回数を value として集計することが多く，集計中も集計結果も key-value 型が入れ子となったデータ構造を使うことが多かった．わざわざ XML で入力したものを，木構造の上からたどり，key-value 型の構造にするのは二度手間であり，効率が良いとは言えない．

これまで加点レベルの違いと合符など 2 文字にわたる加点から JSON と XML に分けた構造を考えていたが，ID の設定と，JSON の柔軟性の高い記述方式を使えば，それぞれの Type ごとにフィルターをかけ，レベル別の訓点資料を作成することも可能になる．集計用のプログラムも分ける必要がなく，それによって効率化が望める．以上のことから加点レベル B は XML ではなく JSON 形式で記述したほうが適切であると言える．

6．あとがき

本稿では国立国語研究所蔵「尚書(古活字版)」を対象として，この加点資料に付与された情報を階層的に電子化し記述するための構造についてまとめ，冒頭 1 丁分を電子化して簡単な集計を行った．集計結果は母数が十分でないため，より広範囲を電子化し再度考察する必要がある．一方，これまでに検討してきたデータ構造では，基本的な集計は行えるものの，一部のデータ検索においては煩雑な処理が必要になることが分かった．

今後は，すべてのデータ構造を key-value 型に変更し JSON 形式で記述するように変更していく．この形式あれば，要素に好きなだけ属性を追加できるため，仮名点であり振り仮名である，などの記述も可能である．訓点資料の統計的な分析の有用性を示すためにも，すべての種類の点を完璧に入力していくのではなく，まずは特定のヲコト点だけに対象を絞って，本文全体についてデータを記述していくこと急ぎたい．この作成したデータ

も研究者らのホームページを通じて公開し，他の研究者の利活用に関しても意見を集めていく予定である．また，作成したデータから書き下し文の作成も試みる予定である．

謝辞

本研究は JSPS 科研費 17K1850606 の助成を受けたものである．また，人間文化研究機構広領域連携基幹研究プロジェクト「異分野融合による総合書物学」の国語研ユニット「表記情報と書誌形態情報を加えた日本語歴史コーパスの精緻化」による成果の一部である．

参考文献

1）田島孝治,堤智昭,高田智和,小助川貞次：訓点資料の加点情報に対する階層的なデータ化の試み—春秋経伝集解を事例として—，じんもんこん 2016 論文集，Vol.2016，pp.51-56(2016.12).
2）赤塚忠(翻訳)：中国古典文学大系(1)書経・易経(抄)，(1972.1)
3）小助川貞次：影印本環境における訓点研究の問題点，富山大学人文学部紀要 (64), 153-165, 2016
4）高田智和：ヲコト点の座標表現，国立歴史民俗博物館研究報告，Vol.192,pp.171-181(2014.12).