アルケイア―記録・情報・歴史―
第19号　2024年11月　*35-46*頁
南山アーカイブズ

資料のデジタルデータ化
～テキストデータを中心に

吉 根 勝 美
南山大学経済学部准教授

Digitization of Materials - Focusing on Text Data

Department of Economics, Faculty of Economics,
Nanzan University
YOSHINE, Katsumi

Archeia: Documents, Information and History
No.19 November, 2024 pp.*35-46*
Nanzan Archives

1　はじめに
2　各種資料のデジタルデータ化
　2.1　資料のデジタル化
　2.2　テキストデータのデジタル化
　2.3　日本語で使う文字のJIS規格
　2.4　Unicode
3　表形式によるデータの整備
　3.1　表形式データ
　3.2　機械可読性
　3.3　データの横持ち・縦持ち
4　まとめ

資料のデジタルデータ化〜テキストデータを中心に

吉 根 勝 美

1　はじめに

アーカイブズという施設（機関，あるいは事業）の役割は，公共性が高く，のちに歴史的重要性をもち得る記録や資料を収集・調査し，整理・保管し，公開・活用することである。実際に所蔵する資料には絵画・彫刻・文書・写真・音声・映像などの種類があり，その現物を長期間にわたって保存することが最優先であろうが，特に，教育・研究機関に設置されるアーカイブズには，教育・研究目的に資料を公開し，活用してもらう役割も重要である。しかし，保存を優先すべき資料を無制限に公開することはできない。そこで，資料の現物をデジタルデータ化すれば，閲覧回数の制限がなくなり，デジタルデータをネットワークで公開すれば，閲覧時間・閲覧場所の制限もなくなる。

一方，公開されている資料を活用する利用者の立場からすれば，そもそもどのような資料が所蔵されているのか分からなければ，アーカイブズの活用のしようがないから，まずはアーカイブズの目録をブラウジングするところから始めるはずである。ある程度の目的意識を持った利用者であれば，目録を検索して目的とする資料を得ようとするはずである。こうしたアーカイブズ利用者に対して，所蔵資料の目録を適切に作成して公開するという役割もアーカイブズには求められる。

絵画・彫刻・写真・音声・映像や現物の文書のデジタル化については，ハードウェアの高性能化に伴い，より高精細なデジタル化が可能になっている。本稿では，ハードウェアに依存するデジタル化については扱わないこととし，文書に書かれているテキストのデジタルデータ化について言及したい。また，文字で書かれる目録は，結局テキストデータの範疇に入るので，目録を表形式データに整備するときの留意点にも言及したい。

２　各種資料のデジタルデータ化

２．１　資料のデジタル化

2022年度から必修となった高等学校の科目「情報Ⅰ」では，情報のデジタル化に関して以下のことを高校生は理解することになっている。

> 情報のデジタル化に関して標本化，量子化，符号化，二進法による表現などを理解するようにするとともに，標本化の精度や量子化のレベルによって，ファイルサイズや音質, 画質の変化が生じることを科学的に理解するようにする。また，情報をデジタル化することにより，情報の蓄積，編集，表現，圧縮，転送が容易にできたり，複数のメディアを組み合わせて統合したり，大量の情報を効率よく伝送したりできることなどについて理解するようにする。（高等学校学習指導要領（平成30年告示）解説【情報編】，p.28）

したがって，彫刻のような立体物のデジタル化を除けば，音声の標本化と音圧の量子化・符号化，絵画・写真・現物の文書の標本化と画素の量子化・符号化，デジタル化された資料の圧縮に代表されるような，資料のデジタル化の基本原理は高校段階で学ぶことであり，本稿では触れないこととする。

２．２　テキストデータのデジタル化

文書をデジタルデータ化するにあたり，現物の文書の写真としてのデジタル化とは別に，文書に書かれている内容であるテキストをデジタルデータ化することも重要である。なぜなら，写真から特定の文字列を抽出するには，近年進んでいる画像検索の技術をもっても100％抽出できるという保証はないが，テキストそのものをデジタルデータ化してあれば，文字列による検索が100%可能になるからである。また，目録公開の目的にかなうためには，目録自体が適切に検索されるように構成されたテキストデータである必要がある。

２．３　日本語で使う文字のJIS規格

テキストデータを構成する文字についても，高校生は以下のような学習活動を経験することになっている。

> 数値や文字については，文書を作成して保存する際に，文字数や全角・半角の違い，改行やスペースの入力によって，ファイルサイズが変化すること，Webブ

ラウザの設定を切り替えることで表示が変化したりすることから，シフトJISなど様々な文字体系があることなどを確認する学習活動が考えられる。（高等学校学習指導要領（平成30年告示）解説【情報編】，p.28）

そもそもデジタルとは，データを有限桁の整数で表現することであるから，文字をデジタルデータ化するためには，同じ文字には同じ番号，異なる文字には異なる番号が割り当てられた文字コード表が必要となる。文字列による検索が100%可能になるためには，同じ文字には同じ番号，異なる文字には異なる番号という原則は必須である。

コンピュータ黎明期には，8ビットすなわち8桁の二進数で表現できる256通りの整数の範囲内に，アルファベット大文字，数字，記号を割り当てたEBCDICコードや，それにアルファベット小文字とカタカナを追加した文字コードもあったが，後に主流となったのは，7ビットでアルファベット大文字・小文字，数字，記号を表現するASCIIコードである。これに1ビット追加してカタカナも扱えるようにした日本産業規格（旧日本工業規格）（JIS規格）JIS X 0201（旧JIS C 6220）「７ビット及び８ビットの情報交換用符号化文字集合」が1969年に制定されており，これが，いわゆる半角カタカナである。また，ASCIIコードのバックスラッシュ (\)が円記号(¥)に，チルダ(~)がオーバーライン(‾)に置き換わっている。

その後，日本語をコンピュータで扱うために，漢字・ひらがな・カタカナを16ビットで表す文字コードをメーカーが独自に作ったものも現れたが，JIS規格としては，JIS X 0208（旧JIS C 6226）が1978年に制定された。ひらがな，カタカナのほか，いわゆる第1水準・第2水準の漢字，その他記号，合わせて6,000字以上の文字ひとつひとつに番号が割り当てられたが，JIS X 0201で定義されて文字についても，いわゆる全角文字として，異なる番号に同じ文字が割り当てられた。異なるJIS規格とはいえ，同じ文字には同じ番号という原則から外れてしまっており，この不都合をカタカナについて回避するため，半角カナを一切用いないという手段が取られることが多い。

また，1983年と1990年の改訂では，同じ漢字に別の番号を割り当て直したり，同じ番号の漢字の字形・字体が一部変更されたりしている。現在からみれば検索可能性を棄損する改訂だが，当時は，とにかくコンピュータで漢字を表示すること自体が，工業規格としての最優先課題だったかもしれない。その後も，JIS X 0208は改訂されるとともに, JIS X 0212（いわゆる補助漢字）が1990年に, JIS X 0213（いわゆる第3水準・

第4水準の漢字）が2000年に制定され，番号が割り当てられた文字は，合計で11,000字を超えた。

２．４　Unicode

日本で使う文字だけではなく，世界で使われるありとあらゆる文字を利用できるように考えられたのがUnicodeである。現代の文字だけでなく古代の文字や歴史的な文字，さまざまな記号も取り入れようとしている。また，JIS規格で定められた漢字はUnicodeでも番号が割り当てられているが，中国や韓国で用いられている似た漢字と統合されており，JIS規格における漢字の並び順は，Unicodeにおけるそれとは一致しない。

現在のテキストデータのデジタル化は，Unicodeを用いるのが第1の選択肢だが，2000年以前にデジタル化されたテキストデータの蓄積も考慮すると，Unicode以外を用いることも検討事項である。なお，JIS規格やUnicodeは，あくまでも文字ひとつひとつに対する番号の割り当て方であり，その番号をコンピュータ上でどう表現するか，つまりどんな二進数で表現するかという符号化の方法は1通りでないので，デジタル化においてどの符号化方式を採用するかという検討事項もある。

ちなみに，JIS規格の符号化方式は，ISO-2022-JP（いわゆるJISコード），EUC-JP（EUC），Shift_JIS（シフトJIS）が代表的であるが，いずれの方式もすべてのJIS規格を完全に符号化するものではない。Unicodeの符号化方式にはUTF-16，UTF-32もあるが，最も利用される方式は，1 ～ 4バイトで符号化するUTF-8であり，ASCII文字は1バイト，日本語は3バイトで表現される。このため，JIS規格のテキストデータの約1.5倍のデータ量となる。

Unicodeでは，例えば「ガ」を1文字で表現する方法と，「カ」と「゛」を組み合わせた合成済み文字として表現する方法がある。したがって，同じ文字に2通りの番号が割り当てられることになり，単純な文字コードの比較だけによる検索では不十分である。

2章で使われた文字列「情報Ｉ」のうち「Ｉ」に代表されるローマ数字や，丸囲み数字（①②③…）などは，かつては機種依存文字と呼ばれ，機種によって文字化けが起きていた。これらの文字はJIS X 0208に含まれておらず，WindowsとMacがそれぞれ独自にJIS X 0208を拡張したことが原因であったが，Unicodeを使用すれば，この問

題は解消する。

ローマ数字をアルファベットで表記することで機種依存性を回避することも行われてきたが，ローマ数字で1 ～ 12を表す12文字（ⅠⅡⅢⅣⅤⅥⅦⅧⅨⅩⅪⅫ）には連続する番号が割り当てられているため，検索時には，アルファベット表記よりも分かりやすく検索条件を記述できるという利点がある。

ただし，大きい数字をローマ数字で表すときに用いる文字（L，C，D，M）と全角のアルファベット大文字（Ｌ，Ｃ，Ｄ，Ｍ）は，見た目は同じでも番号が異なるので，検索時には注意を要する。

3　表形式によるデータの整備

3．1　表形式データ

アーカイブズにおける目録を公開するときには，誰にも分かりやすい表形式による公開が望ましい。表計算ソフトウェアで管理できる程度の目録作成なら，初めから表形式データで目録を整備すればよい。例えば，名古屋市の市政資料館（公文書館）における行政資料（刊行物・パンフレット）の検索結果からは，図1のような表形式の目録が想定できる。

ID	文書名	形態区分・時代区分・出所区分	発行年度	整理番号1	発行年度（西暦）	発行年月	ページ数	大きさ	発行者
88762	名古屋市男女平等参画基本計画２０２５　推進状況報告書　令和４年度事業実績　男女平等参画白書（令和５年度版）	R 17	5	3	2024	R6年1月	76	A４標準	スポーツ市民局
88763	名古屋市公園配置図 令和５年度	R 12	5	5	2024	R6年1月	50	A４標準	緑政土木局
88764	名古屋市地域公共交通計画（案）・概要版（案） 令和６（２０２４）年１月	R 11	5	7	2024	R6年1月	82	A４標準	住宅都市局
88771	名古屋市における自動車環境対策の推進について（報告） 令和６年２月	R 06	5	12	2024	R6年2月	149	A４標準	環境局 名古屋市自動車公害対策推進協議会

図１　目録を表形式で表示する例

３．２　機械可読性

表形式で目録を整備する際には，あらかじめ目録に記載する項目名を定め，1件の資料を1行のデータとして表わす。このとき，不明な項目は空欄のままにすることが多い。

しかし，しばしば統計表においては，単に見た目を整えるために空欄を設けることがある。例えば，学校基本調査（令和６年度速報）の各種統計表は表計算ソフトで作られており，そのファイルが政府統計の総合窓口（e-Stat）のウェブサイトで公表されている。高等教育機関の学校調査の総括表をダウンロードし，それを印刷したものが図2である。見た目には美しく整えられた表ではあるが，実際の統計表には，空欄の行が挿入されていたり，セルが結合されていたりするため，統計表に入力されているデータを活用するには不都合である。

このような表形式データは，機械可読性が低いデータであり，実際に分析するときは，分析しやすいように，データを並べ替えるなどの下処理を施す必要がある。人工知能に学習させるには不向きな形式のデータでもある。2020年12月に総務省は，「各府省が政府統計の総合窓口（e-Stat）に掲載する統計表における機械判読可能なデータの表記方法の統一ルールを策定した」と公表した。簡素な目録作成なら，この統一ルールを適用することで，のちの目録活用の際にも有用であろう。

区分		学校数	学生数			教員数						職員数					
						本務者			兼務者			本務者			兼務者		
			計	男	女	計	男	女	計	男	女	計	男	女	計	男	女
大学	計	(663) 813	2,950,038	1,630,002	1,320,036	192,526	139,085	53,441	200,167	133,955	66,212	268,517	82,759	185,758	7,557	3,282	4,275
	国立	(86) 86	603,975	386,193	217,782	63,821	50,790	13,031	38,982	29,144	9,838	91,985	30,021	61,964	220	114	106
	公立	(91) 103	168,071	77,787	90,284	14,854	10,318	4,536	17,402	11,976	5,426	20,703	5,742	14,961	2,185	613	1,572
	私立	(486) 624	2,177,992	1,166,022	1,011,970	113,851	77,977	35,874	143,783	92,835	50,948	155,829	46,996	108,833	5,152	2,555	2,597
短期大学	計	297	78,295	11,050	67,245	6,237	2,874	3,363	13,627	6,346	7,281	3,391	1,267	2,124	3,005	1,383	1,622
	公立	15	4,921	982	3,939	385	250	135	781	496	285	165	76	89	150	62	88
	私立	282	73,374	10,068	63,306	5,852	2,624	3,228	12,846	5,850	6,996	3,226	1,191	2,035	2,855	1,321	1,534
高等専門学校	計	58	56,342	43,139	13,203	3,922	3,426	496	1,881	1,472	409	2,798	1,657	1,141	68	35	33
	国立	51	50,859	38,523	12,336	3,506	3,062	444	1,515	1,168	347	2,627	1,579	1,048	55	30	25
	公立	3	3,762	3,157	605	276	246	30	252	209	43	105	47	58	1	1	－
	私立	4	1,721	1,459	262	140	118	22	114	95	19	66	31	35	12	4	8
(別掲)																	
通信教育	大学（私立）	46	222,806	100,192	122,614	641	456	185	9,817	6,317	3,500	1,338	600	738	207	108	99
	大学院（私立）	26	6,804	3,864	2,940	134	95	39	965	723	242	42	16	26	113	44	69
	短期大学（私立）	11	20,691	4,740	15,951	188	63	125	2,744	1,191	1,553	62	25	37	65	29	36

図２　空欄を用いて見た目を整えた統計表の例

３．３　データの横持ち・縦持ち

表1は，2011 ～ 2024年度の学部別入学者数（秋学期入学者数を除く）である。この表の作成は，毎年刊行されている南山大学概要から，5月1日現在の学部別1年次在籍者数を集約する作業を，表計算ソフトを用いて行った。

表１　2011 ～ 2024年度の学部別入学者数

学部	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024
人文	373	333	364	400	394	363	350	309	334	313	366	377	356	332
外国語	409	431	369	433	411	407	390	409	344	379	399	380	359	438
経済	281	260	273	288	365	256	300	267	245	267	279	287	285	330
経営	230	249	297	291	268	253	301	284	253	263	271	279	291	300
法	273	315	251	314	290	325	305	276	242	268	292	285	305	271
総合政策	325	339	305	326	371	363	301	268	271	267	247	305	295	297
情報理工	261	238	218											
理工				240	250	253	260	239	189	257	273	254	295	247
国際教養							154	150	137	157	147	168	143	156
短期大学部	162	183	144	154	145	151								

表計算ソフトを用いてデータを整理するとき，表1のようなまとめ方をするのがごく自然であろう。データがないところは空欄のままでいいし，大学全体の入学者数を求めることも，横軸に年度を取った折れ線グラフを作成して，学部別入学者数の推移を視覚化することも容易にできるという利点は大きい。

しかし，年度を経るたびに表全体の構造が変わってしまうという不利な点もある。すなわち，図1のように，あらかじめ項目名が固定することができず，毎年項目数がひとつ増えることになる。縦横を入れ替えても同様の問題が発生する。項目名が学部名で固定されるように見えるが，学部が改編されたときに，表全体の構造を変えなければならない。

表1と同等のデータを，構造を変えないでする表形式のデータと作成する方法があり，実際に作り直すと表2が完成する。この表は，112行のデータからなる，かなり縦長の表となる。表1で空欄だったデータは，表2には現れない。

表２　学部別入学者数を別の形式で表わした表

学部	年度	入学者数
人文	2011	373
人文	2012	333
…	…	…
人文	2024	332
外国語	2011	409
…	…	…
総合政策	2024	297
情報理工	2011	261
情報理工	2012	238
情報理工	2013	218
理工	2014	240
理工	2015	250
…	…	…
理工	2024	247
国際教養	2017	154
国際教養	2018	150
…	…	…
国際教養	2024	156
短期大学部	2011	162
…	…	…
短期大学部	2016	151

表2の形式ならば，1年ごとにデータの行数が増えるだけであり，もし将来学部の改編があったとしても，表の構造を変える必要はない。つまり，学部，年度，入学者数の3項目が変わることはないという利点はある。また，ある統計解析プログラミング言語では，表2の形式のデータを読み込ませないと，折れ線グラフの作成ができない。

ただし，表としての見通しはあまりよくない。例えば，年度ごとの大学全体の入学者数を求めようとするだけでも，表計算ソフトに対する詳しい知識を要する。そもそも，表1を表2に変換する作業，あるいは表2を表1に変換する作業自体，表計算ソフトでは難しい部類の操作である。

表1，表2どちらの形式でも，表形式データの持ち方としては間違いなく，表1のようなデータの持ち方を横持ち，表2のようなデータの持ち方を縦持ちと呼ばれることが多いようである。ただし，e-Statで得られたデータをダウンロードするときは，横

持ちの形式をクロス集計表形式，縦持ちの形式を列指向形式と呼び，前述の統計解析プログラミング言語では，wide, longと区別している。

表2の形式で作られたデータは整然データ（Tidy data）と呼ばれ，以下の条件を満たすものとされている (Wickham 2014)。

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

データを横持ちにするか，縦持ちにするかは，データごとに選択すればよい。図1のように表現している目録データをあえて縦持ちに変換する必要性はあまりない。ただし，例えば，資料ごとに検索用のキーワードを付す場合，ひとつのセルに複数のキーワードを記入することを避けるのに，2つの方法がある。ひとつは，あらかじめキーワードの最大個数nを決めておき，項目にキーワード1，キーワード2，…，キーワードnを追加して横持ちの形式を維持する方法である。もうひとつは，キーワードの個数を事前に定めずに，項目がIDとキーワードの2つだけからなる別表を縦持ちで作成する方法である。前者の方が圧倒的に扱いやすいが，キーワードの最大個数をあらかじめ決める必要がないという点においては，後者の方が有利である。

4　まとめ

本稿では，アーカイブズにおける資料のデジタルデータ化を意識しつつ，テキストデータに特化して，デジタル化において考慮すべき2つのこと，文字コードとデータ形式について述べてきた。特に，テキストデータのデジタル化において考慮すべきこととして，検索可能性と機械可読性の2点を挙げた。

アーカイブズ資料をデジタル化して公開するにあたっては，著作権，個人情報保護，セキュリティについて，もちろん配慮する必要がある。ちなみに，こうしたことは，すでに中学生に指導されていることでもある。

> 情報のデジタル化の方法と情報の量，著作権を含めた知的財産権，発信した情報に対する責任，及び社会におけるサイバーセキュリティが重要であることについても扱うこと。（中学校学習指導要領（平成29年告示）解説【技術・家庭編】，p.50）

コンテンツに用いる各種メディアの基本的な特徴や，個人情報の保護の必要性

についても扱うこと。（同，p.53）

参考

矢野啓介．(2018)．プログラマのための文字コード技術入門，改訂新版，技術評論社．

総務省報道資料．(2020年12月18日)．統計表における機械判読可能なデータの表記方法の統一ルールの策定．

https://www.soumu.go.jp/menu_news/s-news/01toukatsu01_02000186.html

Wickham, H. . (2014). Tidy Data. Journal of Statistical Software, 59(10), 1–23.

https://doi.org/10.18637/jss.v059.i10