制限のない日本語文字表現手法の評価と漢和辞典への応用

代表研究者　Antoine BOSSARD　神奈川大学　理学部　准教授
共同研究者　金 子　敬 一　東京農工大学　工学府　教授

1 はじめに

本研究では,数年前に始めた研究を引き継ぎ,文字の体系化を進めて計算機システムによる活用を目指す.昨年度に提案した制限のない日本語文字表現手法の評価を行い,さらなる応用を提案して実用性を証明する.以下に，先行研究，関連研究およびそれらの限界について述べる.

日本語文字を含む漢字文化圏の文字全体の表現は難しい問題であるため，長年に亘って研究されている.一般に，その表現方法には，主に二つの考え方がある.

一つ目はローカル（局所的）エンコーディング（non-unifying）によって表現する方法である．ローカルエンコーディングの場合，一つの言語を中心にして文字エンコーディング，すなわち文字表現方法を定義する．したがって，ほとんどのローカルエンコーディングは，一か国の言語（文字体系）に対応する．これは伝統的な考え方であり，日本国内では日本産業規格（JIS）が提案した JIS コード（JIS X 0213（Japanese Industrial Standards Committee 2012）など）はローカルエンコーディングの例である．海外に関して，中国では，Big-5 はローカルエンコーディングの例であり，韓国には EUC-KR がある(Lunde 2009).

二つ目の表現方法の考え方は統合エンコーディング(unifying)である．この表現方法では，一つのエンコーディングで複数の言語（文字体系）に対応する．現在，一般的に使用されている統合エンコーディングは Unicode である(The Unicode Consortium 2020).

前者の表現方法では，表現可能な文字数が限られていて，現在でも，多数の日本語文字は表現できない状態である．なお，これは，異体字に限らず，人名用漢字，変体仮名などにも表現できないものがある．また，一般にローカルエンコーディングは局所的な対策であるため，複数の言語を含む文書の表現や処理は困難である.

後者の表現方法は，一般に多数の言語および文字体系に対応した対策であるため，複数言語からなる文書の表現や処理は可能である．しかし，文字エンコーディング自体は比較的使いにくく，専門家から不満の声が上がっている．例えば，日中韓の漢字すべてを統合しているため，コード内の文字検索は困難である．また，そもそも統合化自体の意味が問われている．詳細については，例えば Sekiguchi（2006）を参照.

2 準備

2-1 文字から得られる情報について

本節では，準備として漢字をはじめ本研究に関連する文字の特徴，すなわち，文字エンコーディングおよびシステム設計の際に利用できる関連情報を振り返る.

まず，以下で言及する「漢字」という用語は，一般的に漢字，国字とその異体字（本字，古字，俗字など）を意味する．本研究で対象とする漢字の特徴を以下に示す（Bossard 2018 を参照）:

- **部首**　各々の漢字は唯一の部首を持つ．ただし，特定の漢字については，その部首の定義が参考書によって異なることがある.
- **筆画**　漢字は筆画によって書かれる．一般に，筆画の数（画数）は定められている．しかし，部首と同じく，特定の漢字については，その画数は，参考書によって異なる場合がある．画数に加えて，漢字毎に筆順が定義されている．しかし，これも同様に参考書によって異なる場合がある.
- **異体字**　一つの漢字が複数の書き方（字体）をもつ場合がある．このとき，一般的に，ある字体を代表的な漢字として定め，それ以外の字体を持つ漢字を異体字と呼ぶ．本研究では，多くの場合，文化庁による基準（例えば，常用漢字表）に従って代表的な漢字を定義する．例として，新字体の「亜」が代表的な漢字であり，旧字体の「亞」は異体字になる.

なお，本研究では文字の構造（部首，筆画など）に焦点を当てており，文字の意味や発音（音訓）を直接使用することはない.

公益財団法人電気通信普及財団
研究調査助成報告書　第35号　2020年度

2-2 制限のない日本語文字表現手法（UCEJ）の基礎

本節では,我々が昨年度に提案し,今年度に改善した文字エンコーディングの基礎を振り返る(Bossard and Kaneko 2019/1)．日本語文字を対象にしたUCEJエンコーディングは，３軸に基づいて文字を分類する．したがって，文字一つ一つに唯一の３次元座標が定義される．３軸の意味を以下に示す：

- **X軸**　X座標は文字の部首を表現する．範囲は１から214まである．これに加えて，X座標０の文字はローマ字および仮名を表現するために確保されている．
- **Y軸**　Y座標は文字の画数（部首の画数を除く）を表現する．なお，同じ部首と画数を持つ文字をできる限りに区別するため，筆画の種類および筆順の情報を生かして，Y座標は実数の形で表現する．
- **Z軸**　Z座標は文字の異体字を表現する．なお，Z座標が０である文字は「代表」である（第２節を参照)．したがって，Z座標１以上を持つ文字は異体字である．

例として，代表文字「雪」と異体字「雪」の座標をTable 1およびTable 2に示す．参考までに，(172, 0, 0)の文字は「隹」であり，(174, 0, 0)の文字は「青」である．また，(173, 3.000003, 0)の文字は「雫」である．

Table 1　UCEJの３軸による文字の座標の例－「代表」文字の場合

Character	Properties	Coordinates
雪	radical: 雨	$x = 173$
	stroke number: 3	$y = 3.000002$
	representative	$z = 0$

Table 2　UCEJの３軸による文字の座標の例－異体字の場合

Character	Properties	Coordinates
雪	radical: 雨	$x = 173$
	stroke number: 3	$y = 3.000002$
	variant	$z = 1$

次に，各文字に座標を自動的に定義するには，既存の文字データベースを活用した．独立行政法人情報処理推進機構(IPA)が提供する「文字情報基盤データベース」および文化庁の「常用漢字表」を組み合わせて文字の座標を計算した．データベースの利用については，第３節を参照.

最後に,ソフトウェア工学の観点からUCEJの文字エンコーディングを実装するために必要となった構成を簡単に紹介する．主なオブジェクトはCharacterとEncodingであり，前者は文字一つを表現し，後者は座標計算をはじめ，エンコーディングの構成を動的に作成保持して，応用に向けて文字の読み込みなどを行う．オブジェクトの詳細はTable 3にある．文字数が数万に上るため，このオブジェクトの効率化（最適化）は非常に重要である．例えば，文字の検索や文字の画像データの管理など．また，他のエンコーディングからの変換などのためには，可能な限りUnicodeと対応した情報を保持する．

Table 3　UCEJシステムの主なオブジェクト

Character
internal x, y, z indices (for memory access)
UCEJ x, y, z coordinates
corresponding Unicode glyph (optional)
corresponding bitmap data

Encoding
3-dimensional data structure for characters
secondary structures for database loading and coordinate calculation
ASCII & kana generation routines

3 制限のない日本語文字表現手法(UCEJ)の改善

本節は，Bossard and Kaneko (2019/12)の注釈つき概要である．

3-1 データベースの作成について

最初に，提案システムとデータベースの関係について説明する．提案した文字エンコーディングは，前述のように日本語文字に３次元の座標を定義する．結果として，従来手法よりコードの構成は分かりやすく，文字は検索しやすくなり，結果として，文字エンコーディングはより使いやすくなった．先に説明したように，座標の計算（定義）は部首，画数などに依存する．そこで，既存の文字データベースを利用した．仮に，各文字の部首，画数などの情報を持つデータベースを使用しない場合，手作業でこの情報を集録する必要がある．しかしながら，このアプローチは，まったく実用的でないため，既存のデータベースを利用して，提案システム向けの特殊データベースを自動的に生成した．

この特殊データベースを生成するには，二つの既存データベースを利用した．はじめに，情報処理推進機構(IPA)が提供する「文字情報基盤データベース」の「MJ 文字情報一覧表」(005.02 版－2016 年）を利用した．IPA の MJ 文字情報一覧表には，総計 58,862 文字が集録されていて，文字の部首，画数などの情報がある．また，異体字について，記録されている Unicode の Ideographic variation sequence (IVS)を利用して異体字に対しての情報を推論することができた．

IPA のデータベースに加えて，文化庁の常用漢字表も利用した．このデータベースは 2,143 文字を集録している上に，文字によって舊字体も記録している場合があり，総計 2,529 文字について何等かの情報が得られた．具体的には，新字体を持つ漢字に対して，舊字体の情報も記録しているため，IPA のデータベースから推論した文字間の関係（代表文字と異体字）を増やすことができた．最終的に，生成された特殊データベースは総計 58,862 文字からなる．すなわち，IPA の MJ 文字情報一覧表の全文字の情報の利用に成功した．

この特殊データベースの作成には，約３時間かかる（Intel Core i5-7300U at 2.60Ghz, 8GB RAM, Windows 10 Pro 64-bit)．ただし，この処理は「前処理」であり，一旦，特殊データベースを作成したら，再度作成する必要はない．これ以降のデータベースの修正や訂正は，直接に特殊データベースに対して行えば良い．

3-2 実用性の証明：UCEJ Viewer

本研究で提案した文字エンコーディングの実用性を証明するために，UCEJ エンコーディングに対応するビューアおよびコンバータを実装した．

ビューアは UCEJ エンコーディングによって保存された文書の読み込みと可視化を行うプログラムである．なお，計算機上の文字の表現は Bossard and Kaneko (2019/12)の提案にしたがって実装した．具体的には，文字の表示は次のように行う：Unicode によるコードが割り当て済みの文字の場合，UTF-16 little-endian の該当コードを計算し，文字列に挿入して表示する．これ以外の文字に対しては，画像データを使用して表示する．特殊データベース内に，Unicode のコード未割り当ての文字は 21 文字が存在する．

異体字の表示について，以前に述べたように，Unicode の IVS 情報（あるいは，Unicode のコードを未割り当ての場合，画像データ）を利用する．ここで，文字の表示のために選択したフォント（書体ファイル）に注意する必要がある．IVS の基準（種類）が複数あるため（Adobe 社の Adobe-Japan1 など)，正しい異体字を表示するには，IVS コードが記録されたときと同じ基準でなければならない．本システムが使用する IVS コードは，IPA の MJ 文字情報一覧表に記録されているものであるため，IPA のデータベースの IVS 基準を採用する必要がある．これは，IPA 独自の Hanyo-Denshi の IVS 基準であり，OS 同梱のフォントと異なる（例えば，Microsoft Windows のフォント「游明朝」は Adobe-Japan1 の IVS 基準を実装する；なお，一部のみの対応となる可能性がある)．Hanyo-Denshi の IVS 基準を実装するフォントには IPA の「IPAmj 明朝」があり，第 005.01 版を利用した．

コンバータは，UCEJ による文書作成を支援するためのプログラムである．通常のテキストエディタを使って文書を作成し，Microsoft Windows における通常のエンコーディング形式である UTF-16 little-endian で保存する．次に，出力ファイルをコンバータに入力すると，当初の文書ファイルに対応する UCEJ ファイルが出力される．コンバータは，特に評価実験のために重要になる（以下の第 3-3 節を参照)．

3-3 実験による評価

（１）特殊データベースの改善の評価

本研究は漢字を主な対象とするため，この実験では，平仮名，片仮名およびローマ字は対象としない．はじめに，昨年度の研究成果と今年度の成果とを比較するために，部首による漢字の分布を基にして，「代表的」文字とそうでない文字（異体字など）を部首ごとに分け，それぞれの数を調べる．結果をFigure 1に示す：一つの部首において，黒色は「代表的」文字の数で，赤色は「代表的」でない文字の数である．この棒グラフの横軸は 18 世紀の「康熙字典」の 214 部首を表す．詳細をTable 4に示す．

Table 4　康熙字典の 214 部首

1	一	26	卩	51	干	76	欠	101	用	126	而	151	豆	176	面	201	黃
2	丨	27	厂	52	幺	77	止	102	田	127	耒	152	豕	177	革	202	黍
3	丶	28	厶	53	广	78	歹	103	疋	128	耳	153	豸	178	韋	203	黑
4	丿	29	又	54	廴	79	殳	104	疒	129	聿	154	貝	179	韭	204	黹
5	乙	30	口	55	廾	80	毋	105	癶	130	肉	155	赤	180	音	205	黽
6	亅	31	口	56	弋	81	比	106	白	131	臣	156	走	181	頁	206	鼎
7	二	32	土	57	弓	82	毛	107	皮	132	自	157	足	182	風	207	鼓
8	亠	33	士	58	彐	83	氏	108	皿	133	至	158	身	183	飛	208	鼠
9	人	34	夂	59	彡	84	气	109	目	134	臼	159	車	184	食	209	鼻
10	儿	35	夊	60	彳	85	水	110	矛	135	舌	160	辛	185	首	210	齊
11	入	36	夕	61	心	86	火	111	矢	136	舛	161	辰	186	香	211	齒
12	八	37	大	62	戈	87	爪	112	石	137	舟	162	辵	187	馬	212	龍
13	冂	38	女	63	戶	88	父	113	示	138	艮	163	邑	188	骨	213	龜
14	冖	39	子	64	手	89	爻	114	内	139	色	164	酉	189	高	214	龠
15	冫	40	宀	65	支	90	爿	115	禾	140	艸	165	釆	190	髟		
16	几	41	寸	66	攴	91	片	116	穴	141	虍	166	里	191	鬥		
17	凵	42	小	67	文	92	牙	117	立	142	虫	167	金	192	鬯		
18	刀	43	尢	68	斗	93	牛	118	竹	143	血	168	長	193	鬲		
19	力	44	尸	69	斤	94	犬	119	米	144	行	169	門	194	鬼		
20	勹	45	屮	70	方	95	玄	120	糸	145	衣	170	阜	195	魚		
21	匕	46	山	71	无	96	玉	121	缶	146	襾	171	隶	196	鳥		
22	匚	47	巛	72	日	97	瓜	122	网	147	見	172	隹	197	鹵		
23	匸	48	工	73	曰	98	瓦	123	羊	148	角	173	雨	198	鹿		
24	十	49	己	74	月	99	甘	124	羽	149	言	174	青	199	麥		
25	卜	50	巾	75	木	100	生	125	老	150	谷	175	非	200	麻		

次に，昨年度に生成した特殊データベースとの比較をFigure 2に示す．この図では，昨年度の特殊データベースとの代表的でない文字数の変化を部首ごとに示す．214 の値は，すべて＋1 以上変化しており，合計 578 文字（代表的でない文字）の増加を確認した．このうち，388 文字は常用漢字表のデータベースから得られた異体字である．

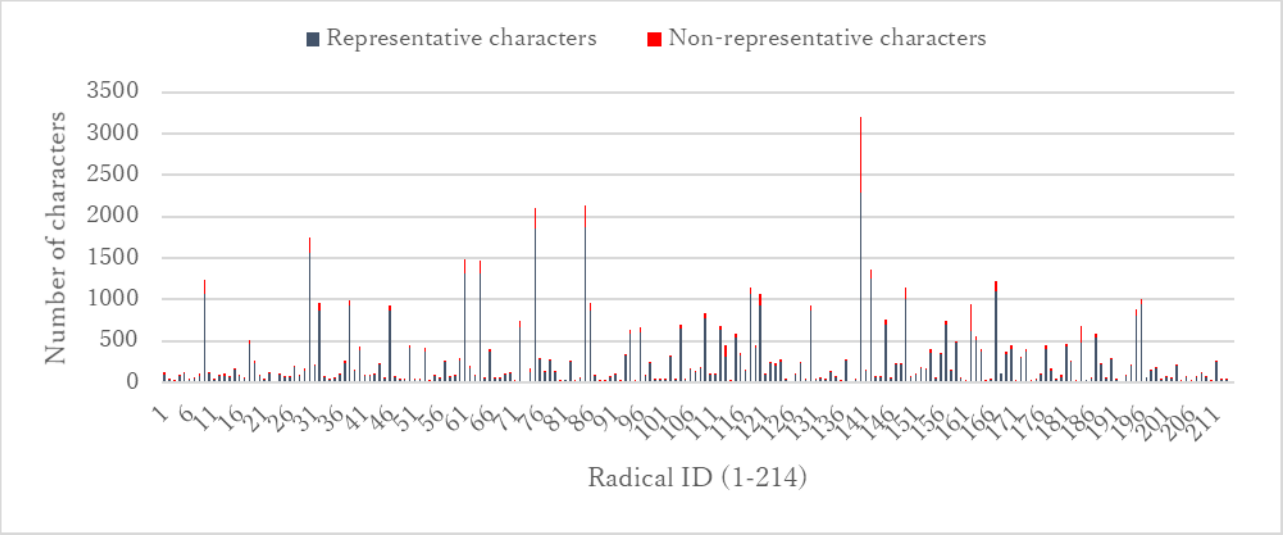

Figure 1　部首による代表的な文字とそうでない文字の割り当て．(Bossard and Kaneko 2019/12)

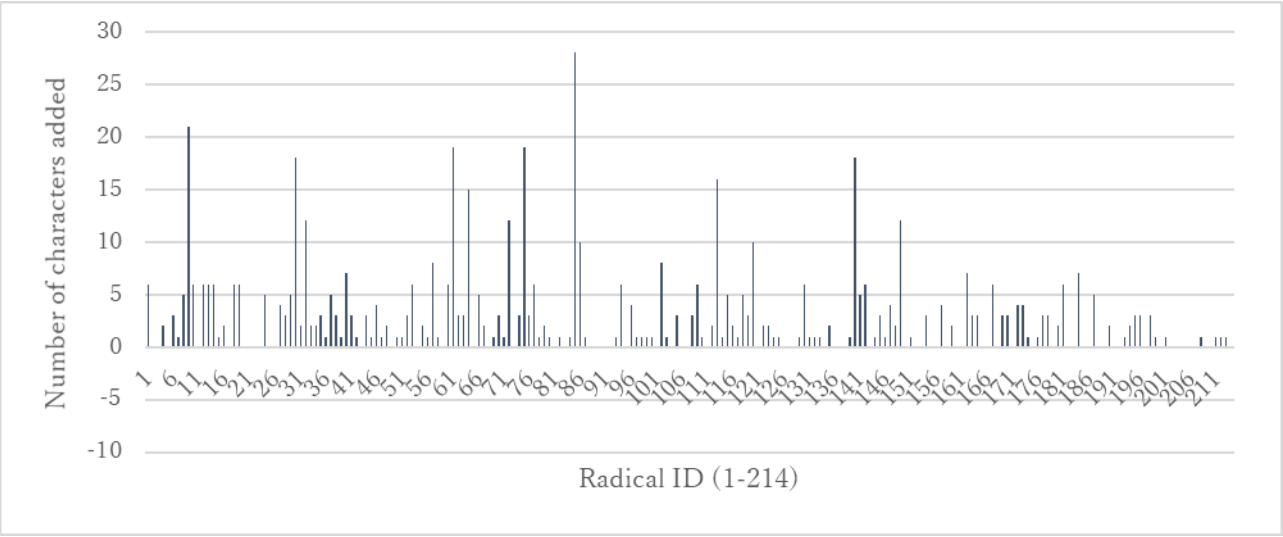

Figure 2　昨年度の研究から代表的でない文字の数の変化．全体で，578 文字の増加があった．(Bossard and Kaneko 2019/12)

（２）保存容量の比較

この実験の目的は，UCEJ 文字エンコーディングを採用したときの文書のメモリサイズ（ファイルの大きさ；単位はバイト）の測定と従来手法との比較である．このために，二つの UCEJ ファイルを準備する．一つ目は実際の日本語文書として，日本国憲法前文を電子政府「e-Gov」のポータルから取得して利用した．この文書のインデントと改行を削除して得られたファイルは 643 文字を含む．また，通常のテキストエディタを使用してこのファイルを作成した（上述のように UTF-16 little-endian 形式でエンコーディングした）．次に，このファイルを UCEJ ビューアのコンバータに入力して，UCEJ エンコーディングによる出力ファイルを取得する．最後に，この二つのファイルの大きさを比較する．なお，提案した UCEJ の計算機上の表現手法は０のビット列を多く含むため，圧縮アルゴリズムを適用したあとの比較も行う．この実験の結果を Table 5 に示す．

前述のファイルは実際の文書から得たため，異体字，特に Unicode のコード未割り当ての文字を含んでいない．そのため，これらの文字を表現可能にする UCEJ に対して，より公正な比較を行うために，二つ目の UCEJ ファイルを準備する．このファイルは UCEJ コードから 100,000 文字をランダムに選択して生成する．この実験の結果をTable 6に示す．

Table 5　提案した文字エンコーディングが要するメモリサイズ（バイト数）の測定と比較－実際の日本語文書の場合．(Bossard and Kaneko 2019/12)

		ZIP compression			
Encoding	Uncompressed	bzip2	deflate	lzma	ppmd
Unicode (UTF-16)	1288	1112	1003	1019	893
(compression ratio)		(14%)	(22%)	(21%)	(31%)
Unicode (UTF-8)	1844	1125	1108	1159	998
(compression ratio)		(39%)	(40%)	(37%)	(46%)
UCEJ encoding	6430	1801	1754	1674	1481
(overhead)	(+400%)	(+62%)	(+75%)	(+64%)	(+66%)
(compression ratio)		(72%)	(73%)	(74%)	(77%)

Table 6　提案した文字エンコーディングが要するメモリサイズ（バイト数）の測定と比較－UCEJ コード内でランダムに選出した 10 万文字からなる文書の場合．(Bossard and Kaneko 2019/12)

		ZIP compression			
Encoding	Uncompressed	bzip2	deflate	lzma	ppmd
Unicode (UTF-16)	365594	221147	258188	232044	208410
(compression ratio)		(40%)	(29%)	(37%)	(43%)
Unicode (UTF-8)	402132	207345	258541	242946	212784
(compression ratio)		(48%)	(36%)	(40%)	(47%)
UCEJ encoding	1000000	228993	312133	260838	209318
(overhead)	(+174%)	(+4%)	(+21%)	(+12%)	(+0%)
(compression ratio)		(77%)	(69%)	(74%)	(79%)

一つ目の実験では，UCEJ で保存した憲法前文の文書ファイルは，Unicode と比べて，400%と大きなオーバーヘッドを持つことを確認した．しかし，圧縮後のオーバーヘッドは，62%から 75%と小さくなった．一方，二つ目の実験では，ランダムに生成されたファイルの圧縮後のバイト数は Unicode と同程度であった．すなわち，ppmd 圧縮法の場合は+0%，bzip2 圧縮法の場合は+4%のオーバーヘッドであった．異体字の扱いが，結果に大きな影響を与えると考えられる．すなわち，Unicode では，IVS を指定して選択する文字（主に異体字）は二つの Unicode 番号を要する．また，二つ目の実験では，文字の数を増やしたことが，結果に影響を与えた可能性がある．

この実験ではメモリサイズの測定に限って UCEJ の文字エンコーディングを評価した．しかし，当然ながら，上述したように，UCEJ の文字エンコーディングには，Unicode などの既存エンコーディングに比べて，把握しやすい構成，広い表現可能な範囲など，いくつかの利点がある．

4 UCEJ の応用に向けて文書処理システム TeX による中日韓（CJK）の文書処理について

本節は Bossard (2019)の注釈つき概要である．

4-1 先行研究と既存システムおよび目的

伝統的に，TeX に基づく「pTeX」システム（Nakano 2018/9）は日本語文書処理（組版など）を可能にする．また，標準の TeX では，「CJK」パッケージ（Lemberg 2015）を利用すると中日韓の文字を扱うことができる．後者はある程度 Unicode に対応するものの，前者は対応しない．しかし，前者は，後者とは異なり，「縦書き」に完全に対応する．韓国語に関しては，「hlatex」パッケージ（Un 2005）もあり，すでに廃止されてしまったシステムである「Omega」（Plaice and Haralambous 1996）と組み合わせれば，Unicode に対応した．

現代の文書処理では，Unicode 対象の「XeTeX」および「LuaTeX」システムをよく利用する．日中文書に対しては、前者専用の「xeCJK」パッケージ（Liu and Lee 2018）があり，中国語に対応する．また，後者専用の「luatex-ja」パッケージがあり，日本語に対応する．最後に，前述の「pTeX」に基づく「upTeX」という，Unicode 対象のシステムもある（Nakano 2018/4）．さらに，前述の「CJK」パッケージに基づく「BXcjkjatype」

パッケージ（Yato 2013）はUnicodeの日本語ファイル（UTF-8形式）をpdfTeXシステムに対応させる．

上記のシステムやパッケージは，ある程度，日中韓の文書処理を可能にする．しかし，本研究では，UCEJ文字エンコーディングとの互換性を目指しながら，日中韓の文書処理における条件や課題を調べて，これらを根本的に解決できる機能を簡単に実装できることを証明した．

4-2 基礎的な実装

西洋文書と違って，日中韓の文書の一つの特性は言葉を区切るために空白を使用しない（ただし，現代の韓国語では使用する）．この影響で，TeXの改行アルゴリズムは無用になる．したがって，最初の課題として，段落の処理をまず検討した．空白を含まない文字列に改行を挿入するために，二文字毎に，文字の間に極めて狭い隙間を入れる手法がある（Veytsman 2006）．以下のTeXマクロ（関数）を使って，日中韓文書の段落に対応する．

```
¥def¥cjk@scan#1{% one single token as input
 ¥ifx#1¥cjk@stop% check if the current token is the stop signal
  ¥par% if so, the paragraph is completed
 ¥else
  #1% otherwise, the current character is displayed
  ¥hskip 0pt plus 1pt minus 1pt¥relax% insert some space after the current token
  ¥expandafter¥cjk@scan% recursively process the remaining text
 ¥fi
}
```

上記のマクロを適用するには，以下の定義と設定を利用する．

```
¥def¥cjk@scanstart#1¥par{% one paragraph as input
 ¥cjk@scan#1¥cjk@stop% append ¥cjk@stop at the end of the paragraph
}
¥everypar={¥cjk@scanstart} % apply to each paragraph of the document
```

また，ここでは詳細を省略するものの，ローマ字を含む日中文書や句点の処理を追加した．TeX（サブシステム「XeLaTeX」）のPDF出力の例をFigure 3に示す（文書は著者の中間報告書の一部である）．段落の改行を見やすくするために，ページ内の枠（天，側など）も出力させた．

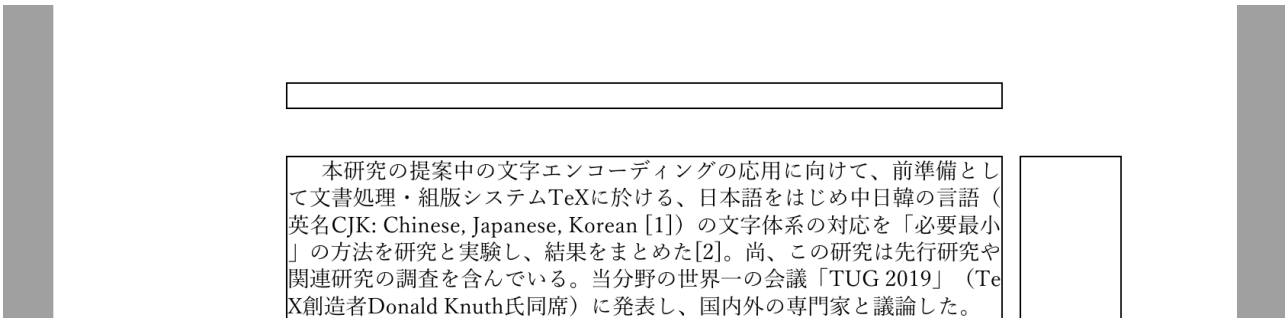

Figure 3　日本語文書の処理の出力の例：改行は対応済み

4-3 韓文書の対象

最後に，韓文書の処理について簡潔に述べる．古代韓文書の場合，すなわち空白を使用しない韓文書に対しては，上記の方法で処理が可能である．現代韓文書の場合，すなわち空白を使用する韓文書に対しては，文書をローマ字の文字列（Figure 3を参照）のように扱う．詳細は省略するものの，基本的に文書にある空白を保持する，いわゆる日中文書の場合と違って空白の無視はしない．

ただし，ラテン文字より，ハングル文字の幅は広いため，フォントの選択とその設定は課題である．以下の TeX マクロを使用して対応する．

```
¥def¥korean#1{% to be called for each Korean paragraph
 ¥latintrue% the Latin mode is activated (regular line-breaking)
 ¥leavevmode%
 {% font adjustments:
  ¥spaceskip=¥fontdimen2¥font plus
   3¥fontdimen3¥font minus
   3¥fontdimen4¥font% 3 times stretch & shrink
  ¥malgun #1% select the Korean font (here, Malgun)
 }
}
```

5 おわりに

日本語文字を含む漢字文化圏における文字の表現は，令和の時代となった今でも，情報処理技術の制約から逃れることができない．過去では，メモリ容量の制限のような計算機の物理的な限界が原因であった．しかし，現在の主な原因はソフトウェアの面にある．すなわち，文字エンコーディングの対応が不十分であることが理由である．本報告書に述べたように，計算機システムの文字エンコーディングには，多数の課題があり，現代の Unicode 基準が不可欠である．しかしながら，日本語だけをとっても，その表現への対応は未完成な状態であり，本研究では新たな解決方法を検討した．結果として，UCEJ 文字エンコーディングを提案した．評価実験などにより，この新しいエンコーディングの利点を証明することができた．また，UCEJ の採用に際して，利用者数の高い文書処理システム TeX をターゲットにして，さらなる実験を行った．その結果，日中韓の文書の基礎的な処理を十分簡単にできることを証明した．

今後の課題については，第一に，提案した文字エンコーディングを中国語に拡張することである．また，TeX などの文書処理システムへの対応も一つの課題であり，上述のように研究を展開中である．現在，平仮名と片仮名への対応はあるものの，変体仮名は今後の課題である．これは，平仮名や片仮名と同様， X 座標 0 に配属する予定である．

謝辞

公益財団法人電気通信普及財団より受けました本研究プロジェクトに対する助成について，心から感謝致します．

【参考文献】

Antoine Bossard, A glance at CJK support with XeTeX and LuaTeX, *Proceedings of the 40th Annual Meeting of the TeX Users Group (TUG)*, *TUGboat*, Vol. 40, No. 2, pp. 196–201, Palo Alto, CA, USA, August 2019.

Antoine Bossard, *Chinese Characters, Deciphered*, Kanagawa University Press, Yokohama, Japan, March 2018.

Antoine Bossard and Keiichi Kaneko, UCEJ database refinement and applicability proof, *Proceedings of the 21st IEEE International Symposium on Multimedia (ISM)*, pp. 64–71, San Diego, CA, USA, December 2019.

Antoine Bossard and Keiichi Kaneko, Unrestricted Character Encoding for Japanese, *Databases and Information Systems X (in Frontiers in Artificial Intelligence and Applications series)*, Vol. 315, pp. 161–175, IOS Press, Amsterdam, Netherlands, January 2019.

Japanese Industrial Standards Committee (JISC), JIS X 0213 (7ビット及び8ビットの2バイト情報交換用符号化拡張漢字集合, in Japanese), 2012.

Werner Lemberg, CJK, April 2015. Package documentation. https://ctan.org/pkg/cjk (last accessed August 2019).
Leo Liu and Qing Lee, xeCJK 宏包 (in Chinese), April 2018. Package documentation. https://ctan.org/pkg/xecjk (last accessed August 2019).
Ken Lunde, *CJKV Information Processing* (second edition). O'Reilly Media, Sebastopol, CA, USA, 2009.
Ken Nakano, Japanese TEX Development Community, and TTK, About pLATEX 2ε, September 2018. Package documentation. https://ctan.org/pkg/platex (last accessed August 2019).
Ken Nakano, Japanese TEX Development Community, and TTK, About upLATEX 2ε, April 2018. Package documentation. https://ctan.org/pkg/uplatex (last accessed August 2019).
John Plaice and Yannis Haralambous, The latest developments in Ω. *TUGboat*, Vol. 17, No. 2, pp. 181–183, June 1996. https://tug.org/TUGboat/tb17-2/tb51plaice.pdf
Masahiro Sekiguchi, 標準化教育プログラム - 第 12 章 文字コード標準 (in Japanese), Japanese Standards Association (JSA), 2006.
Koaunghi Un, 한글라텍 길참이 (in Korean), April 2005. Package documentation. https://ctan.org/pkg/hlatex (last accessed August 2019).
The Unicode Consortium, The Unicode Standard, Version 13.0, 2020.
Boris Veytsman, Splitting Long Sequences of Letters (DNA, RNA, Proteins, etc.), August 2006. Package documentation. https://ctan.org/pkg/seqsplit (last accessed August 2019).
Takayuki Yato, BXcjkjatype package, August 2013. Package documentation. https://ctan.org/pkg/bxcjkjatype (last accessed August 2019).

〈発 表 資 料〉

題　　名	掲載誌・学会名等	発表年月
UCEJ database refinement and applicability proof	Proceedings of the 21st IEEE International Symposium on Multimedia (ISM), pp. 64-71	2019 年 12 月
A glance at CJK support with XeTeX and LuaTeX	Proceedings of the 40th Annual Meeting of the TeX Users Group (TUG), TUGboat, Vol. 40, No. 2, pp. 196-201	2019 年 8 月