情報システムの日本語検索機能に関する考察

A consideration on Japanese string searching for information systems

小山 裕司 [1*]
Hiroshi Koyama[1*]

[1] 東京都立産業技術大学院大学 Advanced Institute of Industrial Technology
*Corresponding author: Hiroshi Koyama, koyama@aiit.ac.jp

Abstract Recently, digital transformations have been aggressive, and many information systems have been designed and built. As the computing power increases, these information systems tend to grow in size, complexity, and functionality. However, the usability and simplicity of information systems have been left behind in the development of information systems. These thoughts reduce the user's workload, errors, and work time, and increase the efficiency and quality of operations. In particular, the string searching feature greatly affects usability. In Japanese, there are many characters and variants, so the same scheme used in the alphabet causes problems. In this report, we organize the history of coded character sets related to this issue and discuss usability improvements using fuzzy and approximate string searching in Japanese.

Keywords coded caracter set; character encoding scheme; variants; fuzzy/approximate string searching

1 はじめに

昨今、特に積極的にデジタル推進が行われ、官公関連でも多数の情報システム（アプリケーションソフトウェア）が設計、構築されている。ソフトウェアによる情報システムでは、正確に大量のデータを高速に処理（演算、記憶、伝達）することができる。情報処理装置の性能が上がるにしたがって、情報システムの規模が拡がり、複雑さが増し、機能が増加する傾向にあるが、情報システムの発展から取り残された事項として使い勝手があげられる。情報システムの黎明期には、利用者は専門家に限定され、多少の使い勝手の悪さによる負担は甘受されていた。しかし、情報システムの普及により、リテラシー及びスキルが低い、一般大衆の利用者が増加した現在、使い勝手の改善は喫急の課題である。使い勝手の改善は、利用者の負担軽減、ミス削減、作業時間の削減、業務効率及び業務品質の改善等に結び付く。

情報システムの使い勝手を改善する概念として、建築家 Ludwig Mies van der Rohe[1]が残した「Less is More」という言葉が引用されることがある。情報システム設計での「Less is More」は、利用者の要望を反映し、無尽蔵に増加した機能を単純（シンプル）に減らすこと、さらには効果及び効率を高めるため、機能を絞り、整理することを意味する。減らしたり絞ったりする対象は、機能から画面、遷移、操作等にも及ぶ。また、利用者の操作を補うための自動計算、バリデーションチェック、アラート表示、重複作業の削減、履歴機能、アンドゥ機能、あいまい検索等も、利用者の視認を補うための表現の統一、ダッシュボード表示、俯瞰表示、レイアウトの工夫等も、広義では「Less is More」の範疇である。

本稿では、これらの改善要素のうち、あいまい検索による使い勝手の改善を取り扱う。

2 文字列検索

情報システムで多用する基本機能に整列と検索がある。整列、データの集合を五十音順、年齢順等の一定の規則に従って並べる処理で、検索は、登録された人物 DB から特定の人物の情報を探したり、ファイルの内容に対し特定の文字列を探したりする等、データの集合の中から目的のデータを探し出す処理である。両者のアルゴリズム及び計算量は積極的に研究され、実装済みの汎用のアルゴリズムが多数存在する。検索は広義には、画像、音声等のメディアも対象であるが、今回はテキスト文字列の検索を扱う。

検索機能の実装で使われる、プログラミング言語標準の比較演算は厳密一致が基本であるが、最近のプログラミング言語、DBMS では標準で正規表現の機能が準備されていることもある。特に Perl から派生した PCRE（Perl Compatible Regular Expressions）[2]は高機能である。

文字の数が数十個であり、異体字の数が限定されているラテン語圏の文字（アルファベット）に対して概ね有効に機能する検索アルゴリズムは容易に実装することができる。以下にラテン語圏の文字で対処すべき事項を列挙する。これらの事項への対処を実装した検索を「あいまい検索」（Fuzzy/Approximate String Searching）と呼ばれる。

- アルファベットの大文字、小文字の無視
- 記号、空白類の無視
- 数値表現のゆれ（10cm、10.0cm、0.1m 等）
- 米英等でのスペルの差（disc、disk 等）、スペルミス
- 省略（I am、I'm 等）

しかし、日本語の検索では対処すべき事項が多数あり、検索アルゴリズムの実装は工夫を要する。次に、日本語の検索で対処すべき事項を列挙する。

- 平仮名、片仮名、漢字（いぬ、イヌ、犬、狗、戌等）
- 英語、日本語、ローマ字（Japan、日本、Nippon）
- 合字、組字等（令和、㍾等）
- 異体字
- 表現のゆれ（引っ越し、引越し、引越等）
  - ハイフン、長音、ダッシュ等
  - 拗音（ぁ、ぃ、ぅ、ぇ、ぉ等）、促音（っ）
  - 長音の有無（サーバ、サーバー等）
  - 踊り字（「々」「〻」「ゝ」「ゞ」「ヽ」「ヾ」「〃」「く」「ぐ」）
  - チ↔ジ、ヅ↔ズ、バ↔ヴァ、ハ↔ファ、セ↔シェ、ゼ↔ジェ、ヒュ↔フュ、ビュ↔ヴュ、ツィ↔ティ↔チ、ディ↔ジ等

Received: 2024-10-30; Published online: 2025-02-28

    - サ行の前のキ、ク
    - イ段、エ段に続くア、ヤ
- 同義語等（手帳、手帖等）
    - スマフォ、スマートフォン、スマホ等
    - スマートデバイス、iPhone 等
    - 会う、逢う、遭う、遇う等
    - 鮓、寿司、鮨、🍣（絵文字）等
    - バッグ、サコッシュ、リュック等

異体字は、同一の文字観念を有するが、グリフに違いがある文字のことで、これらは手書きに起因することが多い。アルファベットでは、異体字を同一の文字として扱い、グリフの差はフォントデザインの差とすることが原則である。しかし、漢字では、各種の経緯から異体字を別の文字として扱うこともあれば、アルファベット同様、フォントデザインの差として扱うこともある。文章で使われる際には異体字は相互に置換できることが多いが、人名の場合はこだわりもあって簡単に置換できることばかりでは無い。異体字の例を表 1 に示す。

表 1　日本語の漢字の異体字の例

基底字	異体字、旧字、俗字等
崎	﨑 嵜 埼 碕 碕 嵜
高	髙
沢	澤
島	嶋 嶌
浜	濱 濵
斉	齊
斎	齋
辺	邊 邉（100 以上）

情報システムの検索機能の実装が単純にアルファベット同様のアルゴリズムである場合、異体字に対処する仕組みが無ければ、「さいとう」を検索する操作を 4 回繰り返す必要がある。いくつかの情報システムではこの問題を意識し、基底字と異体字のゆれに対処する検索を組み込んでいるが、対処の程度は様々であり、また、多くの情報システムは対処が無い。

3　文字コード体系

文字情報の処理のため、文字に番号（数値、符号位置）を割り当てたものは文字コード体系と呼ばれ、特定の文字コード体系で扱う文字の集合は文字集合と呼ばれる。最初に、文字コード体系の変遷を整理する。

ASCII（American Standard Code for Information Interchange、現 INCITS 4）[3]は、1963 年 6 月に米国規格協会（American Standards Association: ASA、後の ANSI）で制定された情報処理のための文字コード体系である。

当時、米国での情報通信等で使われていた 128 個の文字、記号、制御文字に 00_{16}（0_{10}）から $7F_{16}$（127）の番号が割り当てられている。

- アルファベットの大文字及び小文字（52 個）
- 0 から 9 までの数字（10 個）
- 括弧、感嘆符等の約物、数学記号等の記号、空白文字（33 個）
- アルファベットの大文字、小文字
- 改行文字等の制御文字（33 個）

1967 年 6 月に定められた国際標準の 7-bit 符号の文字コード体系 ISO/IEC 646[4]は ASCII の文字集合と概ね同じであるが、18 個の記号（$22\text{-}24_{16}$、27_{16}、$2C_{16}$、$2D_{16}$、$2F_{16}$、40_{16}、$5B\text{-}60_{16}$、$7B\text{-}7E_{16}$ 番）を各国規格で変更できる。

日本では、ISO/IEC 646 に対し、$5B_{16}$（92）番のバックスラッシュ記号（'＼'）の代わりに円記号 （'¥'）を、$7E_{16}$（126）番にチルダ記号（'~'）の代わりにオーバーライン記号（'‾'）を割り当てた 7-bit 符号の文字コード体系と、8bit 符号に拡張し、$A1_{16}$（161）番から DF_{16}（223）番に日本語の片仮名と記号を割り当てた文字コード体系が JIS X 0201（JIS C 6220）[5]として 1969 年 6 月に定められている。8-bit 符号の JIS X 0201 はアルファベット（Alphabet）、数字（Numerical digit）、片仮名（Katakana）の頭文字から ANK コードとも呼ばれる。ANK コードの時代に構築された情報システムでは、異体字はもちろん、漢字の取り扱いは困難であったが、検索の実装は単純であった。のちに、漢字及び異体字を収録した文字コード体系を基盤に更改が行われると、漢字が表示できる代わりに使い勝手が下がってしまうことも発生した。

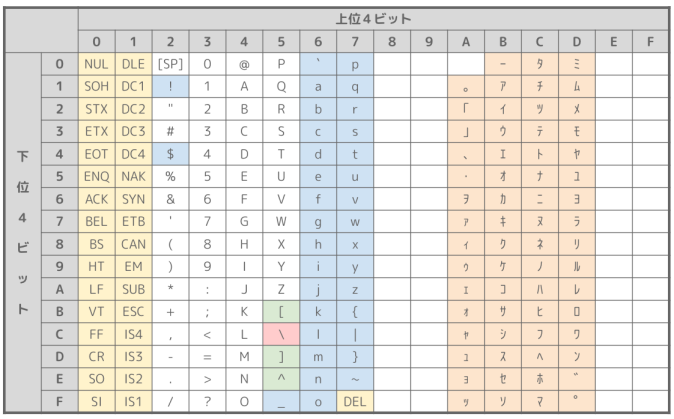

図 1　JIS X 0201 の文字集合

ASCII 及び ISO/IEC 646 は 7-bit 符号であるため、扱うことができる文字の自由度は低い。ISO/IEC 646 はもともと ASCII の流用であったため、ラテン文字の言語であっても英語以外では文字数が不足する。18 個の文字は各国規格で変更できるが、うち 8 個は制約があった。このため、8-bit 符号に拡張され、ASCII で未使用の 80_{16}（128）番以降に各国の独自の拡張ができる ISO/IEC 8859[6]が定められ、西欧諸語の ISO/IEC 8859-1（通称 Latin-1、1984 年 12 月）、中央欧諸語の 8859-2（1987 年）等が定められた。

ASCII 及び ISO/IEC 646 では、文字数の上限から数学記号の乗除算記号（×÷）が無いため、本来は約物の $2A_{16}$（42）番のアスタリスク記号（'*'）と、$2F_{16}$（47）番のスラッシュ記号（'/'）

が代用されているが、現在のISO/IEC 8859では乗算記号（'×'）が D7$_{16}$ 番に、除算記号（'÷'）が F7$_{16}$ 番に割り当てられている。

ASCII（ISO/IEC 646、ISO/IEC 8859）は広く普及しているが、IBM の大型機等の一部の環境では、ASCII が定められる前から蓄積されていた BCD（Binary-coded decimal）での情報資産との互換性を考慮し、BCD を拡張した EBCDIC（Extended Binary Coded Decimal Interchange Code）[7]が使われている。

日本では、16-bit（2-byte）符号の文字コード体系である JIS X 0208（JIS C 6226）[8]が 1978 年に定められた。JIS X 0208 には個（94 区×94 点）の符号位置があり、漢字 6,355 個、平仮名、片仮名、特殊文字、ラテン文字、罫線素片の非漢字 524 個の計 6,879 個が収録されている。JIS X 0208 は JIS 基本漢字等とも呼ばれる。JIS X 0208 の 6,355 個の漢字のうち、当用漢字字体表等の多数の漢字表で出現頻度が高い文字は第 1 水準 2,965 文字とし、残りを第 2 水準 3,390 文字とされた。

1990 年 10 月には、JIS X 0208 に漏れていた文字を集めた JIS X 0212（JIS 補助漢字）[9]が定められた。JIS X 0212 には、漢字 5,801 個、アルファベット 245 個、特殊文字 21 個の計 6,067 個が収録されている。

2000 年 1 月には、JIS X 0208 を拡張した JIS X 0213（JIS 拡張漢字）[10]が定められた。JIS X 0213 は、JIS X 0208 が規定する 6,879 個の文字集合に対して、非漢字 659 個、第三水準 1,259 個、第四水準 2,436 個、計 4,354 個を新規に、計 11,233 個が収録されている。JIS X 0213 は JIS X 0208 の上位集合ではあるが、JIS X 0212 とは互換性が無い。

JIS X 0208 の文字集合には、JIS X 0201 の文字集合のすべての文字が存在しているが、これらを区別するため、JIS X0201 を半角文字、JIS X 0208 を全角文字とも呼ばれる。

複数の文字コード体系の切替を行う技術として、ISO/IEC 2022（JIS X0202）[11]等がある。日本では、英数字等のための JIS X0201 と、漢字等の 16-bit 符号の JIS X 0208 が混在する場合は、ISO/IEC 2022 による SHIFT-IN（SI、0F$_{16}$）、SHIFT-OUT（SO、0E$_{16}$）で両者の切替を行う。しかし、これは処理が複雑であったため、処理効率を改善するため、両文字コード体系を ISO 2022 による切替無しで混在できるように巧みに設計された Shift_JIS が 1982 年に開発され、MS-DOS 及び CP/M-86 に実装され、広く普及した。

EUC-JP（Extended Unix Code Packed Format for Japanese、日本語 EUC）も両文字コード体系を ISO 2022 による切替無しで混在できるようにしたものであり、1986 年に定められ、Unix 系の OS で広く使われていた。

JIS X 0201 及び JIS X 0208 と、Shift_JIS と EUC-JP は概ね同じ文字集合を扱うが、各文字に割り当てられた番号が違う文字コード体系である。

黎明期の文字コード体系は文字集合の各文字に番号を単純に割り当てただけのものであったが、次第に複数の文字コード体系を一緒に扱ったり、番号割り当てに複数の変種（バリエーション）が存在したりすることが生じてきたため、CCS（Coded Caracter Set）と、CES（Character Encoding Scheme）の 2 段階で文字コード体系を示すこともある。前者の CCS は文字集合とも呼ばれ、特定の文字コード体系で扱う文字の定義である。後者の CES（Character Encoding Scheme）はエンコーディングとも呼ばれ、文字をどのように番号（バイト列）で表現するかという決まりである。

ASCII と EBCDIC、また JIS X0201 と EBCDIC 日本語仮名拡張（CCSID 5026）の CCS は概ね同じであるから、これらは CES だと解釈することもできる。同様に JIS X 0201 及び JIS X 0208 と、Shift_JIS と EUC-JP も概ね同じ CCS であり、これらは CES だと解釈することもできる。

ASCII、ISO/IEC 646、ISO/IEC 8859 の各国の独自の拡張による文字集合の差はデータ交換に問題を産み出した。複数の言語圏の文字が混在できるようにするために、複数の文字コード体系を ISO/IEC 2022 による切替も試されたが、あらゆる文字を単一のコード体系に収録する国際標準として、Unicode 1.1.0[12]及び ISO/IEC 10646（UCS: Universal Coded Character Set）[13]が 1993 年 6 月に定められた。

各国から文字を持ち寄り、取捨選択によって収録される文字が決められている。日本の文字は JIS X 0201、JIS X 0208、JIS X 0212、JIS X 0213 の内容が収録されている。

Unicode の文字集合の符号空間は 0$_{16}$ - 10FFFF$_{16}$ で 1,114,112 個（17 面×256 区×256 点）の符号位置がある。2024 年 9 月に定められた Unicode 16.0.0 では、154,998 個の文字が割り当てられているが、1/2 以上が CJKV（中国語、日本語、朝鮮語、越南語）文字である。

現在、Unicode には 3 種類の CES が存在する。UTF-8 は 1 文字を 8-bit（1-byte）から 32-bit（6-byte）の可変長符号で表現する。0 番から 127 番までは ASCII と同じであり、UTF-8 エンコーディングでは、ASCII の範囲（U+0000$_{16}$ - U+00FF$_{16}$）であれば、ASCII と同じデータであるが、大半の漢字は 24-bit 符号である。UTF-8 は、データ交換、ファイル保存の際に使われることが多い。UTF-16 は 1 文字を 16-bit（2-byte）あるいは 32-bit（4-byte）の可変長符号で表現する。漢字は 16-bit 符号である。UTF-32 は 1 文字を 32-bit（4-byte）の固定長符号で表現する。ASCII の範囲のデータの場合、UTF-8 の 4 倍の大きさを占める。UTF-32 はほかの CES よりも大きい領域を占めるが、固定長符号であるため、ソフトウェアの内部表現で使われることが多い。

Unicode では、CJKV 文字のグリフが似た文字が CJK 統合漢字としてまとめられてしまい、議論を呼んだ。また、日本で使われている漢字には異体字が多数存在する。日本の常用漢字[14-16]に収録されている漢字は 2,136 個であるが、これに対し、JIS X 0213 には 10,000 個以上のもの漢字が収録されている。また、日本の戸籍で使うことが許可されている戸籍統一文字[17]の漢字は 55,270 個にも及ぶ。

行政で使われる漢字を整備するため、戸籍統一文字等から 58,862 個の漢字が収録された文字情報基盤（MJ）[18]が定められ、文字フォント付きで公開された。MJ 収録の漢字は Unicode 16.0.0 に収録された。

しかし、日本の官公関連の情報システムの大半は古い文字集合を基盤にして構築されているため、戸籍で許されているすべての漢字を扱うには外字と呼ばれる、利用者定義文字を登録することで、この問題を回避している。外字は個々のシステム依存であり、同じ漢字に別々の番号が割り当てられることも頻発し、相互運用性の問題がある。また、JIS X0208 基盤の古いシステムでは、JIS X0212 の文字等も外字として登録する必要がある。このため、個々のシステムが独自に登録した結果、全国の自治体が戸籍で使っている文字の総数は 163 万個にも及ぶ。この外字を除去するため、ここから重複を削除し、現在戸籍で実際に使われている漢字を精査（同定）し、MJ に未収録であった 9,433 個を足した MJ+を定め、官公関連の情報システムに広めていく取り組みが行われている[19]。

各文字コード体系に収録されている漢字等の個数を表 2 に示す。

表 2　　各種の文字コード体系

文字集合	文字数	補足
当用漢字（1946 年）	1,850	
JIS X 0201（1967 年）	191	英数字、記号、片仮名等
JIS X 0208（1978 年）	6,879	第 1-2 水準漢字
常用漢字（1981 年）	2,136	
常用漢字＋人名用漢字	2,999	
JIS X 0208＋JIS X 0212	12,946	第 1-2 水準漢字＋補助漢字
JIS X 0213（2000 年）	11,233	第 1-4 水準漢字
戸籍統一文字	55,270	
MJ	58,862	文字情報基盤
Unicode 16.0.0（2024 年）	101,970	CJKV 文字
MJ+	68,295	

4　　あいまい検索の実現

プログラミング言語標準の比較演算の基本は厳密一致である。Python でのコード例を次に示す。

```
citizens = {"戸沢":"foo", "大﨑":"bar", "嶋津":"baz"}
v1 = citizens["島津"]
v2 = citizens["嶋津"]
```

最近のプログラミング言語は標準で正規表現の機能が準備されていることが多い。正規表現による検索の Python でのコード例を次に示す。

```
import re
citizens = {"戸沢":"foo", "大埼":"bar", "嶋津":"baz"}
r = re.compile("(戸沢|戸澤)")
v = [citizens[k] for k in citizens if r.search(k)]
```

アルファベットの大文字、小文字の区別の無視は `tolower` 関数等でも対処できるが、正規表現で大文字、小文字の区別を無視するオプション（`re.IGNORECASE`）でも対処できる。また、複数の文字コード体系が混在する場合（いわゆる全角文字、半角文字等）は、`unicodedata.normalize` 関数等で対処できる。このように、アルファベットに対しては、概ね有効に機能する検索アルゴリズムを容易に実装することができるが、日本語の検索で対処すべき事項の実装は工夫を要する。

次に日本語の異体字を意識した検索の設計手順を示す。

- 読み仮名
  JIS X0201 の時代に回帰する行為ではあるが、漢字の文字列に読み仮名を付与すれば異体字の問題は概ね解決できる。しかし、手間がかかるため、実際の運用に難がある。
- 異体字選択肢（Variation Selectors）
  異体字選択肢は 2002 年の Unicode 3.2 で異体字を扱うために準備された仕組みであり、基底字 1 文字に対し、複数の異体字を異体字選択肢として付加することができる。基底字及び異体字選択肢で付加された異体字は異体字列（Variation Sequence）と呼ばれる。異体字列には標準異体字列（SVS: Standardized Variation Sequence）と漢字異体字列（IVS: Ideographic Variation Sequence）がある。SVS は Unicode Consortium が直接管理しているが、IVS のグリフは漢字異体字 DB（IVD: Ideographic Variation Database）で定義され、定められた申請によって登録することができる。MJ の文字は IVS に登録されている。異体字選択肢を反映した実装を適切に行うと、異体字の問題は大枠では解決する。
- 異体字補助辞書（Variation Supplement Dictionary）
  Unicode の異体字選択肢でも、各種の経緯及び議論から、現状、異体字の登録は限定的である。「神」は「神」の異体字として登録されているが、「高」と「髙」、「鴎」と「鷗」等は別の文字として登録されている。このように異体字選択肢を単純に実装するだけでは、これらの異体字が検索から漏れてしまう恐れがある。このあたりを勘案し、異体字選択肢の仕組みを補うための補助辞書を整備し、これを参照する実装が必要である。

取り扱う文字（異体字）が Unicode の文字集合に無ければ、処理も表示も難しいので、MJ+の整備、Unicode での収録、普及が待たれる。

日本語でのあいまい検索では以下の処理が付加される。

- 部分文字列の抽出
  検索したい文字列を解析し、小さい部分文字列を抽出する。検索文字列が「データサイエンス特論」である場合は「データ」「サイエンス」「特論」を部分文字列として抽出する。日本語では、単語と単語の区切りが弱いので、MeCab[20]等の解析エンジンを使う。部分文字列の抽出することで、各部分文字列単位で評価し、広範に検索できる。
- 部分文字列の評価
  各部分文字列と、検索対象の文字列との間の類似性を評価値として算出する。評価値の算出には各種のアル

ゴリズムを使う。Web 等の検索エンジンであれば、評価値が高い順に表示すればよいが、情報システムの検索機能では、閾値を設定し、一致の判断を行う。

5 まとめ

情報システムの普及、発展から情報システムの使い勝手が取り残されている。情報システムの使い勝手の改善は、利用者の負担軽減、ミス削減、作業時間の削減、業務効率及び業務品質の改善等に結び付く。特に、検索機能は使い勝手を大きく左右するが、日本語は文字数も多く、異体字も多いため、アルファベットの検索と同じ手法では問題が残る。本稿では、この問題に関連する文字コードの変遷を整理し、あいまい検索による使い勝手の改善を試みた。

最後に、現状の課題及び今後の改善等を列挙する。

- 異体字補助辞書の整備
- 異体字同定の推進
- AI 概念検索を活用した部分文字列検索の拡張
- サンプル実装の存在

また、情報システムの設計及び実装にあたって、使い勝手、特に検索機能の改善が重要であることの啓蒙も必要だと感じた。

参考文献

1. Isabel Droge. Mies van der Rohe: The Architect Who Thought Less Was More. 2023. Available from: https://www.thecollector.com/mies-van-der-rohe-pioneer-modernist-architect/
2. PCRE: Perl Compatible Regular Expressions. Available from: https://www.thecollector.com/mies-van-der-rohe-pioneer-modernist-architect/
3. INCITS 4: 1986[R2022] Information Systems - Coded character Sets - 7-Bit Standard Code for Information Interchange. 2022. Available from: https://webstore.ansi.org/standards/incits/incits1986r2022
4. ISO/IEC 646:1991 Information technology - ISO 7-bit coded character set for information interchange. 1991. Available from: https://www.iso.org/standard/4777.html
5. JIS X 0201. 1997. Available from: https://www.jisc.go.jp/app/jis/general/GnrJISSearch.html
6. ISO/IEC 8859-1:1998 Information technology - 8-bit single-byte coded graphic character sets Part 1: Latin alphabet No. 1. Available from: https://www.iso.org/standard/28245.html
7. EBCDIC. Available from: https://en.wikipedia.org/wiki/EBCDIC
8. JIS X 0208. 2012. Available from: https://www.jisc.go.jp/app/jis/general/GnrJISSearch.html
9. JIS X 0212. 1990. Available from: https://www.jisc.go.jp/app/jis/general/GnrJISSearch.html
10. JIS X 0213. 2012. Available from: https://www.jisc.go.jp/app/jis/general/GnrJISSearch.html
11. ISO/IEC 2022:1994 Information technology - Character code structure and extension techniques. Available from: https://www.iso.org/standard/22747.html
12. Unicode. Available from: https://home.unicode.org/
13. ISO/IEC 10646:2020 Information technology - Universal coded character set (UCS). 2020. Available from: https://www.iso.org/standard/4777.html
14. 当用漢字. Available from: https://www.bunka.go.jp/kokugo_nihongo/sisaku/joho/joho/kakuki/syusen/tosin02/index.html
15. 常用漢字. Available from: https://www.bunka.go.jp/kokugo_nihongo/sisaku/joho/joho/kijun/naikaku/pdf/joyokanjihyo_20101130.pdf
16. 人名用漢字. Available from: https://www.moj.go.jp/content/001131003.pdf
17. 戸籍統一文字. Available from: https://houmukyoku.moj.go.jp/KOSEKIMOJIDB/M01.html
18. 文字情報基盤（MJ）. Available from: https://moji.or.jp/mojikiban/
19. 地方公共団体情報システムにおける文字要件の運用に関する検討会. デジタル庁. Available from: https://www.digital.go.jp/councils/local-governments-character-specification
20. MeCab. Available from: https://taku910.github.io/mecab/