日本語資料の全文テキストデータ分析ツール NDL Ngram Viewer の開発について

青池 亨（国立国会図書館）

概要：国立国会図書館（NDL）は 2021 年度に実施した OCR テキスト化事業の成果である OCR テキストデータを活用して，Google Books Ngram Viewer から着想を得た日本語版 ngram viewer を開発し，NDL Ngram Viewer と称して実験サービスとして 2022 年 5 月 31 日に一般公開した（https://lab.ndl.go.jp/ngramviewer/）．本実験サービスは，日本語のフレーズが検索可能である点と，正規表現をサポートした検索を行える点の 2 点において，先行する類似サービスにない特色を備えており，高い新規性を有する．一般的な性能の計算資源のみで各種機能を実現するために，大規模テキストデータの処理方法に多くの工夫を講じた．本論文では，検討の過程で実施した調査，採用した実装の詳細及び NDL Ngram Viewer を実際に利用した分析例について報告する．

キーワード：OCR，全文テキストデータ，頻度分析，可視化，システムアーキテクチャ

Development of NDL Ngram Viewer, a tool for full-text data analysis of Japanese materials

Toru Aoike (National Diet Library, Japan)

Abstract: The National Diet Library (NDL) developed a Japanese version of ngram viewer called NDL Ngram Viewer inspired by Google Books Ngram Viewer using OCR text data from the result of the OCR text conversion project conducted in FY2021. It was released to the public on May 31, 2022 (https://lab.ndl.go.jp/ngramviewer/). This service is highly innovative in two respects: it can search Japanese phrases, and it supports regular expressions. In order to realize this service using only general-performance computing resources, many innovations were made to the method of processing large-scale text data. In this paper, we report on the research conducted in the course of the study, details of the implementation adopted, and examples of actual analysis using the NDL Ngram Viewer.
Keywords: OCR, Full Text Data, Frequency Analysis, Visualization, System Architecture

１．まえがき

国立国会図書館（NDL）は，2021 年度の事業の一環として，2020 年末時点で国立国会図書館デジタルコレクション（デジコレ）から提供していたデジタル化済み資料のほぼすべて（約 247 万点，2.2 億画像）について，NDL 所蔵の図書館資料に対して高い性能でテキスト化できるよう最適化した OCR（光学文字認識）を用いて全文テキスト化する作業を行った．この中にはデジタル化済みの図書資料約 97 万点や雑誌資料約 132 万点等が含まれており，2022 年 12 月以降，デジコレの全文検索機能や視覚障害者等用データ送信サービスにおいて利用するテキストデータとして提供予定である．現時点では，著作権保護期間が満了した図書資料（約 28 万点）を次世代デジタルライブラリー（https://lab.ndl.go.jp/dl/）に投入し，先行的に実験サービスとして提供している．次世代デジタルライブラリーでは，全文検索機能や OCR テキストデータのダウンロード機能を提供しており，利用者からおおむね好評を博している．

他方，検索時にヒットする情報が大幅に増加したことで調べたい対象と関係が薄いノイズとなる情報も増え，利用者が必要とする資料が埋もれてしまうという新たな課題も浮かび上がってきた．例えば，キーワード検索において数千件ヒットした検索結果から必要な資料を人の目で確認して選抜することは多大な労力を要し，現実的ではない．また，逆に表記ゆれへの考慮が不十分なために本文検索にヒットせずに必要な資料を見落としてしまう場合もありえる．

利用者の労力を抑えて大量の情報から効率的に目的の資料に辿りつく手段として，全文検索サービスとは別に，検索結果の可視化ときめ細かな検索クエリのチューニングをインタラクティブに繰り返せるようなサービスを用意し，膨大な情報の中から必要な検索条件に「当たり」をつけることのできる手段に関する調査研究として，日本語版 ngram viewer の開発を行った．

２．イントロダクション

“ngram viewer”とは，書籍の全文テキストデータを利用して，特定の単語やフレーズの頻度を出

版年代に沿って可視化することができるサービスを指し，2010 年に公開された“Google Books Ngram Viewer”[1][2]がその端緒である．Google Books Ngram Viewer は，Google が“Google Books”プロジェクトにおいて進めてきたデジタル化資料の OCR 処理により蓄積された全文テキストデータの活用方法の一つとして考案されたサービスで，日本語には対応していないものの，英語やフランス語，中国語等の資料について，出版年代ごとの単語やフレーズの利用頻度の推移を可視化することができる．

これまで，Google Books Ngram Viewer と類似の可視化サービスが世界中で開発されてきた．HathiTrust が運用する Bookworm[3]は Google Books Ngram Viewer と同様の仕組みで検索を行えるサービスであり，一部のコレクションについては日本語の単語での検索可視化も可能である．しかし，1 単語での検索のみをサポートしており，フレーズやセンテンスでの検索を行うことはできない．2021 年に公開された Gallicagram[4]は，フランス国立国会図書館（BnF）の運用する Gallica の API を活用して，Google Books Ngram Viewer を意識したサービスが作られた事例である．BnF 外部の大学院生 2 名が Gallica の API を利用して開発した実験サービスであり，Gallica 収載の全文テキストデータを対象コーパスとして，その中に含まれる単語の出現頻度を時系列により可視化できる．また機能の観点では，Google Books Ngram Viewer が 2012 年にバージョンアップを行い，ワイルドカード（*）検索に対応した[5]が，正規表現をサポートしているサービスは 2022 年 5 月時点で存在しなかった．

日本における ngram を利用した研究は古くから存在し，長尾ら[6]は 1993 年の時点で大規模日本語テキストから 1 文字単位の ngram 統計を作成する方法を検討し，その性質を分析している．また，分析者が手元の日本語テキストデータを対象に ngram を用いた分析を行えるツールとしては KH Coder が存在する．しかし，大規模な日本語 ngram を対象としたウェブアプリケーションとしての分析ツールが開発された事例はこれまでなかった．

NDL が ngram viewer の実験サービスを開始するにあたっては，後発サービスの視点から，先行する類似サービスがまだ提供していないが，漢字の異体字や送り仮名等のために表記ゆれが多い日本語の全文テキストデータを検索する上で特に有用な機能と考えられる正規表現検索の実現に取り組んだ．また，事業の公益性の観点から，これまでの先行サービスが必ずしもオープンにしてこなかった大規模テキストデータの集計方法等サービスを実現するための手順の詳細についても広く共有し，研究者や技術者が自らのもつテキストデータ資源を活用した類似サービスを検討できるよう試みた．

３．概要

3.1.材料

本論文では次に挙げる 3 種類の対象について報告する．

①NDL が 2020 年 12 月時点でデジコレから提供している著作権保護期間が満了したデジタル化図書資料全件約 28 万点（約 3,000 万画像）から作成した OCR テキストデータ

②NDL が 2020 年 12 月時点でデジコレから提供しているデジタル化雑誌資料全件（個人送信資料や館内限定公開資料を含む）約 132 万点（約 7,200 万画像）から作成した OCR テキストデータ

③NDL が 2020 年 12 月時点でデジコレから提供しているデジタル化図書資料全件（個人送信資料・NDL 館内限定公開資料を含む）約 97 万万点（約 1.4 億画像）から作成した OCR テキストデータ

なお，①の成果物は 2022 年 5 月に公開した NDL Ngram Viewer に搭載されているデータ断面である。②及び③のデータ断面については，今後の NDL Ngram Viewer への追加搭載に向けて開発を継続中である．

3.2. 分析及び検討

NDL Ngram Viewer の開発にあたって検討した内容を，「テキストデータに含まれる文字列の分割方法」「分割した文字列の集計方法」「集計文字列を検索するシステムの構築」の 3 段階に分けて報告する．

3.2.1. テキストデータに含まれる文字列の分割方法

日本語は英語等の言語とは異なり，単語間の切れ目が明確ではない．仮に 1 文字単位で ngram を計算すると意味上の切れ目と乖離したキーワード・フレーズの割合が多くを占めてしまう．従って，意味上の区切りとある程度対応する箇所で文字列が切れるよう，形態素解析による分かち書き処理を OCR テキストデータに行った後，分割された形態素 gram について 1gram から 5gram までを集計することとした．これにより，例えば「大正デモクラシー」や「同潤会アパートメント」といったある程度長い単語についても，個々の単語の文字列長と関係なく集計に含めることが可能となった．以降，本論文では，集計対象とした 1〜5gram を“ngram”と表記する．

形態素 gram での分割には，Elasticsearch[7](ver 7.7)の analysis-kuromoji プラグイン（MeCab-ipadic 辞書）を採用し，Elasticsearch に処理させた．採用の理由としては，分割対象のテキストデータに

は旧字や異体字が相当な割合含まれることから，形態素分割の前段階として字体の丸め等の処理を効率的に行う必要があったためである．

具体的な方法としては，全文テキストデータを資料毎に 1 つに連結し，異体字マッピングテーブルを導入した Elasticsearch の analyze-API に POST し，返戻された分かち書き結果に対して，1 から 5 までの窓サイズで走査を行い，各 ngram の頻度を資料単位で合計した．

形態素解析結果を分割の根拠に用いる手法の懸念点として，得られた分割位置の一貫性の欠如や，辞書を更新した際の分割根拠の変化が挙げられる．特に，分割位置の一貫性の問題は，今回の対象が誤りを一定確率で含む OCR テキストデータであるという特性から，本来の文に数文字程度のノイズの入った文字列を形態素解析しなくてはならないケースが多く存在するため，ノイズの影響を受けて同じ文脈であっても分割位置が変化する事象が高頻度に起こり得ると予想された．この問題の影響を小さく抑えるため，今回のアプローチにおいては，形態素の分割結果を集計対象の上限である 5gram がどこからどこまでの範囲であるかを確定させる目的にのみ利用し，頻度集計時には分割位置の情報や分割数の情報を捨てて再度連結することとした．具体的な例を挙げて説明すると，「こんにちは」という文字列に対して形態素分割を行うとき，「こんにち-は」という 2gram と，「こんにちは」という 1gram の 2 種類の分割結果が生じ得るが，集計時には分割結果を再度連結し，同一の文字列としてひとまとめに扱っている．

この手法は，分割位置の変動に対して頑健な集計結果が得られる反面，集計対象の形態素分割位置の情報が失われることが短所であり，今後，Google Ngram Viewer のように品詞情報を区別した検索を行えるように機能追加を行う場合には，分割位置の統計情報を別途情報として保持して集計をやり直すなどの工夫が必要になると考える．

3.2.2 分割した文字列の集計方法

ngram viewer を実現するためには，ngram の出現頻度を，対象とする全ての資料を横断して集計する必要がある．対象資料数が少量に限定される条件下での理想的なアプローチは，全文テキストデータに出現する全ての ngram の頻度をメモリ上に保持しながら計算していくことであるが，NDL のデジタル化資料数百万点規模の資料群を対象に同様の集計を行う方法を考えるとき，利用可能なサーバのメモリリソースには物理的な制約があることから，現実的に適用することは不可能である．くわえて，今回の集計対象の特性として，OCR 処理時の読み取りミスに起因するただ 1 度しか出現しない多様な文字列パターンが膨大に存在し，これらがメモリ容量を圧迫することが予想された．この課題を解決するために，集計時にキャッシュアルゴリズムを導入することで，ほとんど出現しない ngram を逐次メモリ上から破棄し，必要なメモリリソースを抑えて集計する方法を考案した．

Least Frequently Used（LFU）[8]は，コンピュータ上のメモリの利用を効率的に管理するために考案されたキャッシュアルゴリズムの一種で，オブジェクトがメモリ内で参照される頻度を監視し，事前に設定したメモリサイズを超過しそうになった際に，最も参照頻度の少ないオブジェクトから順に破棄する手法である．今回は集計方法に LFU を組み入れることで，常に一定量のメモリを利用しながら集計を行えるようにした．

次に実際の集計方法を述べる．メモリサイズの上限を 250GB として LFU を有効化した KeyDB[9]（マルチスレッド対応した Redis 互換のキーバリューストア）コンテナを用意し，ハッシュキーとして ngram の文字列を持たせ，ハッシュフィールドとして当該 ngram の出版年代を用い，各ハッシュフィールドの値として当該出版年度の出現情報を持つように設計した．例えば 1990 年に出版されたとある資料が，「人文学」という ngram を 4 つ含んでいたとすると，KeyDB 内のハッシュキー「人文学」のハッシュフィールド「1990」の値を+4 加算する．加算時に親となるハッシュキーが参照されるため，当該ハッシュは使用頻度が高いオブジェクトとして削除の優先度が低下する．

この集計方法を採用する際の懸念点として，ngram の頻度分布によっては，集計対象としたい ngram とノイズ扱いとして破棄して構わない ngram の間でうまく足切りを行うことができず，頻度分析上重要な意味を持つ ngram を誤って破棄してしまう危険性が想定される．このような危険を回避して集計システムが正しく動作するかを検証するため，①のデータに対して ngram の頻度分析を行った．具体的な方法としては，上記の集計方法で集計した結果から，ngram を 1,000 万種類ランダムサンプリングし，サンプルに含まれた ngram について総出現回数と該当する ngram の数量の分布を確認した．図 1 に結果のグラフを示した．

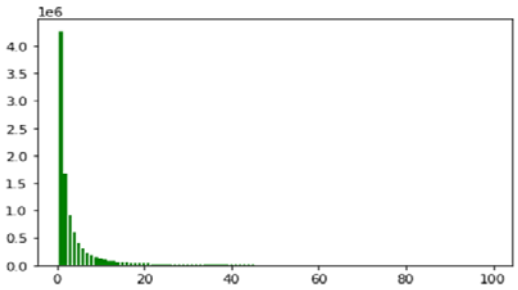

図 1 ①の集計結果からランダムサンプルした 1,000 万種類の ngram について，総出現頻度（横軸）と当該総出現頻度を持つ ngram の個数（縦軸）
※頻度 100 以上の ngram については省略している．

この分析から，①については LFU アルゴリズムによって ngram の破棄が行われた後においても，低頻度の ngram が大きな割合存在することが読み取れる．これら低頻度の ngram は集計途中に削除されている可能性があるため信用できないが，出現頻度が増えるほど 該当する ngram の数量は減少していくため，一定頻度以上出現する ngram については集計開始から集計完了まで一貫して保持されると考えられる．

ここまでの議論から，正確な数量を反映していない低頻度な ngram と，集計対象全資料に対して正確な頻度情報を保持していると考えられる高頻度な ngram との間を，集計後の総出現頻度を利用して区別し，一定基準を下回った場合に当該 ngram を切り捨てることで，結果の信用性を高める工夫を検討した．

①から③まで一律に，総出現頻度が 4 未満のものについて切り捨てる場合，切り捨て操作前後の ngram の種類数は

①について 16 億 ngram→8.3 億 ngram
②について 12 億 ngram→8.9 億 ngram
③について 10 億 ngram→8.5 億 ngram

となった．

集計対象となったテキストデータの分量が増えるほど，KeyDB が管理する総種類数が減少する傾向がみられた．これは，複数回出現した種類数が増えていくことで，一定間隔で出現する ngram の情報についてメモリ上に占める割合が増加していくことから，必要な容量を空けるために 1 度出現して以降参照がない ngram が LFU によって短い期間で頻繁に破棄されることが原因と推察される．したがって，集計処理後半に初めて出現する ngram は集計序盤に初めて出現する ngram と比較して，間隔が短いうちに LFU によって破棄されると考えられるため，真の総出現頻度が切り捨て基準を上回っていても誤って破棄される可能性が高まっていく点に注意する必要がある．特に③については，①や②と比較して，対象とした画像数に対して切り捨て操作後の ngram の個数が少ないと考えられるため，今回設定した切り捨て基準である総出現頻度が 4 付近の ngram については，集計情報の正確性が低い懸念が示唆された．

なお，現在公開している①のデータを投入した NDL Ngram Viewer は次世代デジタルライブラリーの全文検索と同様の範囲を対象としており，NDL Ngram Viewer の検索結果から全文検索の検索クエリを生成できるため，集計した ngram の頻度が実際の全文出現頻度と一致するか検証を行うことが可能である．上記の切り捨て操作後のデータについて，全文に対するヒット箇所の数と，本手法で集計した ngram の出現回数に大きな乖離がみられるケースは現時点では確認されていないことから，少なくとも①の切り捨て基準については，総出現頻度 4 未満として支障がないと考えている．

②や③の結果の信頼性を高めるためには，切り捨て操作に用いる総出現頻度の基準を①よりも高くする必要があるが，切り捨てられた ngram は検索から除外されるため，検索結果の正確性と検索結果のヒット数はトレードオフの関係にある．①から③までのそれぞれの対象に対して，切り捨て操作基準となる適切な総出現頻度を設定することが今後の課題である．

3.2.3 集計文字列を検索するシステムの構築

3.2.2 で作成したデータを Elasticsearch に，文字の正規化のみをかけた keyword フィールドとしてインデックスすることで検索可能なサービスを構築したが，正規表現検索機能の実現について大きな課題があった．

具体的には，Elasticsearch の正規表現検索は，前方一致検索に特化して高速であるが，逆に後方一致検索は非常に応答が遅い特性を有する．①のデータに対するクエリの応答時間の事前調査時には，同程度の長さの前方一致検索クエリと比して後方一致検索クエリは 3,000 倍以上長い応答時間を要したケースが多く見られた．この問題に対処するため，ngram に対して反転した文字列を持つ反転文字列フィールドの検索インデックスを別途用意しておき，サービスに問い合わせてきた正規表現検索クエリが前方一致か後方一致かをサーバサイドで自動判定し，後方一致クエリが来た場合には反転文字列フィールドに対する前方一致検索となるようにクエリを組み替えて検索応答する工夫を行った．例えば，「.{2,4}図書館」という後方一致クエリは，内部で「館書図.{2,4}」という前方一致クエリに組み替えられ，反転文字列フィールドの検索インデックスに対して問い合わせを行っている．この工夫により，前方一致と後方一致検索をどちらも高速に行える機能を実現した．

参考のため，2022 年 5 月公開時点での NDL NgramViewer の Elasticsearch に投入したドキュメントのフィールド内容の例を図 2 に示した．


```
{
"id":"jzA4w7V6r7U4ObGX6vmvTwRRmmPn",
"version":1,
"ngramkeyword":"ぽんぽこぽん",
"reversedkeyword":"ぽんぽこぽん",
"ngramyearjson":"{¥"1930¥":1,¥"1941¥":3,¥"1940¥":3,
¥"1929¥":16,¥"1928¥":7,¥"1939¥":2,¥"1935¥":7,¥"1934
¥":4,¥"1944¥":1}",
"count":44,
"confident":true(※不使用)
}
```


図 2. インデックスされたドキュメントの例

ngramkeyword フィールドに ngram の文字列を格納し，キーワード検索や前方一致クエリを処理する際にはこのフィールドを検索する．reversedkeyword フィールドにも同じ文字列を格納するが，このフィールドには reverse filter を適用しており，インデックス時に反転文字列として処理されるため，後方一致クエリの処理時に検索する．ngrasmyearjson フィールドには 3.2.2 の集計処理後のハッシュフィールドを値とともにそのまま json 文字列として格納しており，この json 文字列をフロントアプリケーションが直接読み込んでグラフの描画を行っている．また，総出現頻度は count フィールドに別途格納しており，検索結果の頻度順ソートに利用している．

このように ngram viewer の機能実現に必要とする情報を，ドキュメント側に事前に用意しておき，Elasticsearch やバックエンドアプリケーション側で検索時に大きな計算コストを要する追加処理を行わずに済むように設計したことで，10 億ドキュメント近い検索対象に対して，高度な計算能力を持たないサービス提供環境においても高速な応答が可能となった．

3.3. 集計時の計算資源及びサービス提供環境情報

・ ngram の集計に利用した計算機サーバ環境
CPU:Intel（R） Xeon（R） Bronze 3206R CPU @ 1.90GHz
DRAM:384GB

・ サービス提供環境
データストア：
AWS Opensearch Service i3.xlarge インスタンス
アプリケーション：
Amazon EKS クラスタ上の pod（メモリ 2GB）

4. NDL Ngram Viewer の機能

開発した NDL Ngram Viewer は次に挙げる機能を有する．

- キーワードにヒットした ngram の頻度を折れ線グラフに可視化して，グラフのデータ点を全文検索へのリンクとする機能
- 「/（半角スラッシュ）」区切りで複数のキーワードを入力することで頻度の比較を行う機能
- 検索キーワードとして正規表現を入力することで，マッチした ngram を列挙して表示する機能
- 頻度情報をタブ区切りテキストとしてダウンロードする機能
- 利用者が選択した複数のキーワードを合算して集計する編集機能

5. NDL Ngram Viewer を利用した分析例

まず，出版年代に対する流行語の出現頻度の変遷を確認する例として，②の雑誌全件を集計した対象についての検索結果を示す．図 3 は出版年代を横軸にとった際の「みゆき族」と「竹の子族」の出現頻度の推移の比較を表している．Wikipedia によれば，「みゆき族」は 1964 年頃，「竹の子族」は 1980 年頃それぞれ流行語となった単語であるが，いずれも実際に流行した時期と同時期に単語が出現し，徐々に使われなくなっている様子が読み取れる．

図 3「みゆき族」（水色）と「竹の子族」（赤色）の推移

図 4 は，同じく②の雑誌全件を集計した対象について「真空管」と「トランジスタ」の語を比較したものである．トランジスタラジオの商品化が始まった 1960 年前後に，出現頻度が逆転する様子がうかがえる．

当該単語が使われだしたおおよその時期を把握するだけでなく，流行が終焉するまでの期間や減衰の程度についても可視化できることで，特定の年代の文化について資料をもとに調べたい研究者等に新たな発見をもたらす手段を提供することができると考えている．

図 4「真空管」（水色）と「トランジスタ」（赤色）の推移

次に，正規表現を利用した強力なあいまい検索の活用例として，デカルトの命題である「われおもうゆえにわれあり」について，表記ゆれを含めて検索する場合を示す．例えば表 1 及び表 2 は「(われ|我|吾)(思|おも).*(故|ゆえ)に.*り」という s 正規表現クエリで②のデータを対象に検索し，ヒットした ngram を出現頻度の多い順に並べた結果を表している．これら 2 つの表からは，全文検索においては細部の表記に違いがある検索語のバリエーションを把握しているかどうかで，発見できる資料数に大きな差が出ることや，資料種別

や刊行年代の違いによっても使われる表記は異なる様子が読み取れる.

表 1.①の著作権保護期間満了図書資料を対象に正規表現検索で取得した ngram と出現頻度

ngram	出現頻度
我思ふ故に我在り	41
我思ふ故に我あり	22
吾思ふ故に吾在り	21

表 2.②の雑誌資料を対象に正規表現検索で取得した ngram と出現頻度

ngram	出現頻度
我思う故に我あり	284
我思う故に我在り	173
われ思うゆえにわれあり	125
我思うゆえに我あり	92
われ思う故にわれあり	91
われ思うゆえにわれ在り	66
われ思う故にわれ在り	61
我思ふ故に我在り	37
我思う故に我有り	36
われ思う故にわれ有り	9

正規表現検索によって，意図的に表記ゆれに対して幅をもたせた検索を行い，その結果を確認しながら必要とする資料に到達する可能性の高いキーワードを探索的に発見する，といった利用を想定している.

また，もう一つの使い方としては，プログラミング言語による処理を書かずに，ウェブブラウザから正規表現の挙動を体験する使い方を想定している．例えば，正規表現を学ぶ初学者の演習用途や，正規表現を教える講師が実際に動かして説明するための例として利用されることを期待している.

６．今後の見通し

現在，本論文で報告した②及び③を対象に加えた NDL Ngram Viewer の拡張に向けた開発に取り組んでいる．また，NDL 以外の機関においても，所蔵するデジタル化資料に対して，NDLOCR[10]等を適用することでテキストデータを入手し，活用を行いたいケースは考えられ，既に実例も存在する[11]ため，こうした取組を支援するため集計処理やサービスのソースコードのオープンソース化を予定している．併せて，ngram と出版年代ごとの頻度の集計結果についてもデータセットとしての公開を予定している.

NDL Ngram Viewer に今後追加を検討している実験的機能としては，検索結果において，利用者がひらがなとカタカナを集約した表示を選択できる機能や，Google Books Ngram Viewer に存在する品詞情報を区別した用法の検索機能，抽出元資料の書誌情報に含まれる日本十進分類に基づく絞り込み機能等を検討している．実現の目途の立ったものから組み込んでいきたい.

謝辞

いつも有益なコメントを下さる NDL 次世代システム開発研究室の皆さんに感謝します．また，今回の NDL Ngram Viewer のシステムの実現にあたっては，人文情報学研究所主席研究員・NDL 委嘱研究員の永崎研宣先生から頂いたご助言が大きなヒントになりました．この場を借りて御礼申し上げます.

参考文献

[1] Google Ngram Viewer. https://books.google.com/ngrams, (参照 2022-10-24)
[2] Michel, J. B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Google Books Team,Aiden, E. L. “Quantitative analysis of culture using millions of digitized books”. science, 331(6014), 176-182, 2011.
[3] “HathiTrust Research Center. HathiTrust+Bookworm. Accessed July 11, 2022. https://bookworm.htrc.illinois.edu, (参照 2022-10-24)
[4] Gallicagram. https://shiny.ens-paris-saclay.fr/app/gallicagram, (参照 2022-10-24)
[5] Ngram Viewer 2.0. https://ai.googleblog.com/2012/10/ngram-viewer-20.html, (参照 2022-10-24)
[6] 長尾眞，森信介．大規模日本語テキストの nグラム統計の作り方と語句の自動抽出．情報処理学会研究報告自然言語処理 （NL）,1993-NL-096, 1-8, 1993
[7] Elasticsearch. https://www.elastic.co/jp/elasticsearch/, (参照 2022-10-24)
[8] Lee, Donghee, et al. “LRFU (least recently/frequently used) replacement policy: A spectrum of block replacement policies.” *IEEE Transactions on Computers* 50.12, 1996.
[9] KeyDB. https://github.com/Snapchat/KeyDB, (参照 2022-10-24)
[10] 2 令和 3 年度 OCR 処理プログラム研究開発（令和 3 年度 OCR 関連事業について．https://lab.ndl.go.jp/data_set/ocr/r3_morpho/, (参照 2022-10-24)
[11] 中村覚，劉冠偉，山田太造. NDLOCR を用いた東京大学史料編纂所史料集版面画像に対する検索システムの開発，研究報告人文科学とコンピュータ（CH）,2022-CH-130（5）, 1-8, 2022.