

少数言語におけるコーパス利用に関する一考察

後藤 齊 (東北大学大学院文学研究科)

1. はじめに

本稿は、少数言語におけるコーパス研究の位置づけについて検討した上で、コーパスを活用するための一つの示唆を与えようとするものである。

コーパスは、英語をはじめとする多くの言語の研究において、広く使われるようになってきた。少数言語にとっても有効な手法であることは、論をまたない。しかし、英語コーパス研究の現在の状況はあまりにも大規模化していて、少数言語の研究にとって直接参考にできることは少ないようにも見える。

とはいえ、英語コーパス研究の流れの中から学ぶことはできる。まずは、基本的なツールと正規表現を使うことによって、コーパスの有用性を実感することである。さらに、コーパスの扱いに慣れるためには、身近な言語、特に日本語で試行錯誤を繰り返すことにも意味がある。

2. コーパスの大規模化

「コーパス」という用語を、現代的に、コンピュータで処理される大規模なテキストの集積と理解するならば、コーパスを用いた言語の研究は英語を中心に発達してきた。その理由を考察することは無駄ではない。

それは一つには、もちろん、コンピュータがもともとアメリカで開発されたという技術史上の理由による。しかし、そればかりではなく、言語研究史上の理由も存在する。1960年代というコンピュータ自体がまだ未発達であった時代に、コンピュータを言語の分析に使うことが可能だと考えついた人がいたという事実である。当初、コンピュータは文字通りに電子計算機として作られたのであり、言語の分析に使えるというアイデアは決して自明のものではない。

W.N.Francis と Henry Kučera によるこの画期的なアイデアからアメリカ英語を対象とした **Brown Corpus** が作られたことは、言語研究史における一つの出

来事である。Brown Corpus の先駆性は、その設計思想にも現れており、均衡コーパスという概念およびそれを実現するための方法を編み出した。これらの考え方は現在でもその意義を失っていない、Brown Corpus は現代にいたるまでのコーパスの古典的なモデルとなっている。

Brown Corpus 以後のコーパス研究の展開にも、技術史と言語研究史の要因がからんでいる。コーパスに基づく言語研究は 1970 年代から 80 年代半ばにかけてあまり顧られることがなかった。その理由は、一つにはコンピュータがまだ一般には普及しておらず、また、その能力が大きなテキストを扱うには未熟であったという技術的なことである。それに加えて、特にアメリカの言語研究の流れがデータより理論に関心を置く生成文法に大きく傾いたという言語学史上の要因も働いた。

1980 年代後半にコーパスは再び注目を浴びるようになり、その後は、イギリスを中心にしてコーパスに基づく英語研究が一つの手法として定着した。ここには、イギリスにおける経験主義的伝統という、より大きな文脈で語られるべき要因が働いたことも否定できない。このような研究の流れの中で、コーパスの作成や利用に関する理論と実践が蓄積されており、それは英語コーパス言語学と称されるに至っている。

1960 年代の英語のコーパスは 100 万語規模であった。コンピュータの処理能力の向上もあって、1980 年代以降のコーパスは大規模化している。1990 年代には 1 億語規模の英語コーパスが作られ、現在は数億語規模のコーパスも現れている。

このような大規模なコーパスの作成は、個人でできるものではなく、多数の人がチームとして参加し、多額の予算をかけて行われる大規模プロジェクトによって初めて可能な作業である。このような大規模プロジェクトの遂行が可能かどうかは、特定言語を対象とする研究者の人口およびそれを支える経済状態に大きく依存する。これは、言語学の多くの領域において個人による研究が可能であることと著しい対比をなしている。

コーパスの大規模化は、産業とのつながりと無縁ではない。1980 年代後半に

コーパスに基づく英語研究は注目を集めたが、それはイギリスのバーミンガム大学とコリンズ社が共同で作成したコーパスを基礎にして Collins COBUILD English Dictionary『コリンズコウビルド英語辞典』(1987)を刊行したことが大きな要因として働いた(カウイー 2003)。この辞書は英語学習辞典の在り方に大きなインパクトを与え、さらに、コーパスが持つ辞書編集、ひいては言語研究全般における大きな可能性を広く知らしめたのである。

90年代に作られた1億語規模の British National Corpus はイギリスの国家的なプロジェクトの産物であるが、それには大学や大英図書館のほか、出版社も参画したし、イギリス政府の通商産業省(Department of Trade and Industry)も補助金を出していた。現在最大規模のコーパス Bank of English は、コリンズ社による COBUILD プロジェクトの産物である。

出版社や通商産業省が言語研究と結びつきをもっていることは、英語教育が世界規模で産業として成り立っているという事実に基づいている。英語コーパスの成果は世界をマーケットとする商品となりうるし、それによってイギリスの出版業、教育産業が活性化するからである。

日本語でも、現在、国立国語研究所の KOTONOHA 計画(<http://www.kokken.go.jp/kotonoha/>)および科学研究費特定領域研究「代表性を有する大規模日本語書き言葉コーパスの構築」(<http://www.tokuteicorpus.jp/>)によって、現代日本語の1億語規模のコーパスが作成されつつある。その他の主要ヨーロッパ語や中国語、韓国語などにおいてもコーパスの構築が進められているが、いずれも、ある程度は、イギリスにおけるコーパス研究の進展に追随しようという性格を持っていることは否定できない。

3. 少数言語とコーパス

小言語がコンピュータ利用と無関係であったわけではない。例えば、1976年から学術的な電子化テキストのアーカイブとして機能していたオックスフォード大学の Oxford Text Archive(OAT)には、英語のテキストだけでなく、ギリシャ語やラテン語というヨーロッパの古典語のほか、フルフルデ語、ゲール語、

クルド語、ラトビア語といった小言語のテキストも寄託されていた。英語、フランス語、ドイツ語、スペイン語のテキストの中には、中世語のテキストも含まれていた。このことは、小言語の話者や研究者の中にも、早くから積極的にテキストの電子化とその共有に取り組むものがいたことを示している。

しかし、英語のコーパス研究は急速に進展していった。現在、新たに英語のコーパス研究を始めよう志す研究者は、規模や性質の異なる多くのコーパスのうちから、自分の研究目的に応じてできるだけ適切なものを選んで入手することができる。そして、このできあいのコーパスを対象として、語彙、文法などの言語の分析に取り組むことが比較的容易である。つまり、コーパスによる研究に関心を覚えた研究者が、実際にコーパスを扱うまでに必要な時間と手間はそれほど大きくない。

少数言語のコーパス研究においては、多くの場合、まず研究者自身でコーパスを構築するところから始める必要がある。もちろん少数言語には一般に先行研究があまり多くは存在せず、研究者が独力で研究を進めていくという事情は、コーパスを使用する場合に限られない。しかし、音韻や文法の分析といった言語研究一般にみられる状況とコーパスの状況とでは、その違いはあまりにも大きい。

それでは、少数言語の研究者にとって、英語コーパス研究から学ぶことはあまりないのであろうか。ここで考えたいのは、英語にも最初から大規模コーパスがあったわけではないという自明のことがらである。最初の段階では、現在から見ればプリミティブなコーパスとプリミティブなツールを使って研究がなされていた。その段階の研究によってコーパスの有用性が認識され、それがコーパスの改良とツールの改良へとつながって、さらに一段階進んだ分析を可能にすることにつながった。英語コーパス研究の進展は、このような循環が、プラスの方向に働いたことによる。

少数言語では、社会的な事情が異なるため研究成果と産業とのつながりが考えにくいとすれば、最終的に進むべき方向が英語コーパス言語学の場合のような大規模コーパスにあると考えることはできないであろう。それでも、初期の

英語コーパス研究の歴史に学ぶことは多い。その一つは、コーパス研究の初期においては、基本的なツールを使いこなして、その有用性を認識することが次のステップにつながるということである。

さらに、テキストファイルを扱う基本的なツールはコーパス作成過程でも有用になる。コーパスを作るということは、個々のテキスト（話し言葉データ）を電子的な形で収集し、それらを一つのコーパスとしてまとめるということである。したがって、コーパスの作成は以下の過程を経る。

1. 言語データの収集
2. 電子テキスト化
3. コーパスへの編成
4. 検索手段の確保

ここで考えるべきことは多いが、多数の電子テキストを一体として整合性のあるように編集しておくことは根本的に重要である。そうでなくては、コーパスとして一貫した検索と分析を行うのに大きな支障が生じてしまう。そのような編集は、来歴の異なる多数のテキストを扱う場合、それ自体手間がかかり、神経をつかうものになる可能性もある。できるだけ編集作業も自動化できることが望ましい。個々のテキストが一貫していれば、それらの間で一貫させることは、比較的容易に自動化することができるが、その自動化に際してテキストツールはここで大きな助力となるのである。

4. テキストツールと正規表現

4. 1 テキストツール

ここでテキストツールと呼ぶものは、テキストファイルに対して種々の文字列操作を行うことを主目的としているソフトウェアである。文字列の編集、置換、検索などの機能を備えた種々のソフトウェアが存在する。テキストエディターがその代表例であり、多くの場合複合的な機能を備えているが、単一の機能に特化したツールも少なくない。そのうち、検索機能の充実したものは、言語分析ツールとして使うこともできる。

なお、言語分析用のソフトウェアには、多くの一般的なソフトウェアと同様に企業によって開発され商品として流通しているものもある。しかし、言語研究者の数は社会全体に対して大きな割合を占めてはおらず、マーケットが限定される。そのため、多くの場合、開発元の企業にとって魅力ある分野ではないであろう。研究者自身が開発し、安価あるいは無料で、多くの場合ネットワークを通じて配布するもの（いわゆるシェアウェアやフリーウェア）の中に研究上有益なものが少なくないことには注意しておくべきである。

ここではごく少数のツールを紹介するにとどめざるをえない。むしろ、有用なツールを自分で探そうとする意志を促したい。そのためには、情報を集積しているサイトに頼ることが有用である。例えば、Alan Wood's Unicode Resources (<http://www.alanwood.net/unicode/>)は Unicode に関連した情報(特にフォントに関する情報)を集めているが、Unicode に対応したエディターなどのツール類の情報も充実している。

Windows に付属するエディター「メモ帳」は Unicode 対応については優れているが、エディターの基本である編集機能において極めて不十分である。正規表現による検索機能はエディターの必須要件と考えるべきであるが、その機能を備えていない。エディターとしては、正規表現検索の機能、さらに複数のファイルを同時に検索する `grep` の機能を備えているものを選ぶべきであって、簡易的な言語分析ツールとして使うこともできる。

日本語を扱うためのテキストエディターとしては、サクラエディタ (<http://sakura-editor.sourceforge.net/>)が比較的機能が充実している。一応 Unicode に対応するが、日本語文字の範囲に限られるので、十分な対応とはいえず、日本語以外の言語を扱うには必ずしも向いてはいない。しかし、それ以外の編集機能については優れており、特に検索の操作性はよい。エディターだけでどこまでできるかを示す一つの見本として捉えることができる。

多言語対応のテキストエディターも多く、選択は好みによるとも言えるが、UniRed (<http://www.esperanto.mv.ru/UniRed/ENG/index.html>)は特徴的である。ヨーロッパの多くの言語に対応したスペルチェック機能があり、文字コー

ド間のコンバートの機能も充実しており、文字コードの統一が必要な場合には役に立つ。ダイアクリティック付きの文字をキーボードからの入力するための機能にも工夫がなされている。

なお、正規表現検索には対応していないようであるが、BabelPad (<http://www.babelstone.co.uk/Software/BabelPad.html>)は、最新の Unicode に対応していて、Unicode の表から文字を選ぶことができるので、Unicode の多様な特殊文字を目で見確認しながら選ぶのに、また、文字コード間のコンバートの用途にとっては有用である。ここから文字選択の機能を独立させた BabelMap (<http://www.babelstone.co.uk/Software/BabelMap.html>)もあり、ワープロなどの他のソフトウェアで特殊文字を使いたい場合に役に立つ。

検索としては、エディターの `grep` 機能だけでも十分に有用であるが、言語分析に特化したツールは結果の表示のしかたなど、そのための高い機能を備えている。ラテン文字を使っており、分かち書きする言語では、英語用のツールをそのまま使える可能性がある。しかし、言語の形態論的特徴に応じて、その使い勝手は異なるので、少数言語を扱う際には、使用者自身が柔軟に対処する必要もあろう。

コンコーダンスとしては、Unicode には対応していないが、TXTANA (<http://www.vector.co.jp/soft/win95/util/se052447.html>)の性能が高い。開発者はこれを「情報吟味型」のコンコーダンスと位置づけているが、これは確かにあたっている。KWIC 検索は高速であるが、単にあらかじめ決めた条件で検索するだけでなく、様々な条件による並べ替え、絞込みなどができ、検索結果をさまざまな観点から見直すことができる。正規表現を登録しておいて、簡単に検索文字列に呼び出し、必要に応じて書き換えることができる。

もう一つのコンコーダンス `antconc` (http://www.antlab.sci.waseda.ac.jp/antconc_index.html)は、Unicode ほかの多くの文字コード、エンコーディングに対応している点で極めて特徴的である。日本語だけでも、ShinJIS、EUC、ISO-2022-JP ほかに対応していて、数多いとは言えない日本語対応コンコーダンスとして十分に実用になる。分かち書き

をする言語であれば、有用性はさらに高い。少数言語の分析において考慮に値する選択しである。

4. 2 正規表現

正規表現とは、いくつかの記号に特殊な意味を持たせて、複雑な検索対象をパターン化した式として表現するためのしくみであり、多くのツールやプログラミング言語に組み込まれている。もともと UNIX で用いられていたが、その影響を受けて正規表現を採用しているパソコン上のソフトウェアも少なくない。例えば、多くのエディターで検索・置換の際に正規表現を使うことができ、実際、上で紹介したツールの多くも正規表現に対応している。

正規表現そのものについては、参考書類も多く、ネットワークから得られる情報も充実しているので、参考文献に挙げた書籍類にゆずり、具体的な記述はここでは避けておく。正規表現は慣れていない者の目には一見とりつきにくいものに映るが、プログラミングだけでなく言語研究においても応用範囲が広いことを指摘しておきたい。

実際のところ、正規表現で使う記号（メタ文字）の種類はそれほど多くはない。しかも、すべてを覚える必要はない。使い勝手のよいものから、少しずつ覚えて、使えばよいのであり、使い始めるためのハードルは決して高くはないのである。

そもそも、プレーンテキストはコンピュータの内部では単なる文字の並びであって、通常の検索手段では単語など人間にとって意味のある概念を利用することができない。言語研究のためには語や形態素などの言語学的単位を検索できることが望ましいが、それをプレーンテキストに対して直接行うことは難しい。正規表現によっても言語学的な単位を直接扱えるわけではないが、その柔軟性をうまく使うと、擬似的にそれに近いことができるのである。

正規表現を使う前提として、文字コードに関する知識は必須である。例えば、[0-9]という正規表現によって、0 から 9 までのうちのどれか一つ、つまり任意の数字を表すことができる。これは 0~9 が文字コードの上で連続しているから

である。形は似ているが「0」や「9」のような全角文字はこの範囲には入らない。また、漢数字「一二三…」は文字コードの中で漢字のグループの中に散在しており、連続していないので、同様の表記で表すことはできない。

漢数字ゼロ「〇」は漢字のグループから外れたところに位置している。また、「一」は通常一緒に使われるカタカナとはコードの上では別のグループに属している。「々」も漢字のグループではない。これらをそれぞれカタカナや漢字と一緒に扱いたいのであれば、検索する側で意識的にそのように指定する必要がある。

また、テキスト・データそのものに文字コード上の誤りが含まれていることがある。例えば、ひらがなの「へ」とカタカナの「へ」など、形の似ている文字が混同されていることは少なくない。「一」(カタカナの音引き)、「一」(漢数字1)などは、「一」(ダッシュ)、「-」(全角ハイフン)、「一」(マイナス)など形の似た記号類とも相互に混用されていることがある。OCRで読み取ったテキストデータやさまざまな出所から集めたものは特に要注意である。

Unicodeの普及によりコンピュータでの多言語の取り扱いが容易になった。過去においては、日本で普通に販売されているパソコンで多言語を扱うことは不可能に近かったが、OSのレベルでUnicodeへの対応ができたため、現在ではそのようなことはない。しかし、Unicodeによって多言語に関わるすべての問題が解決されたと考えるのは間違いである。Unicodeには多くの文字が含まれているため、形の類似による混用の危険性はこれまで以上に高くなっている。

また、酒井(2005)が指摘するように、文字コードの数値に全面的に依存したソートは特定言語の利用者や研究者の想定する順序にならないことが多い。複雑な文字体系を持つ言語や、補助記号付きの文字を使う言語では、利用者の側で個別に解決することになる。そのための知識を仕入れることは、利用者の責任に属する。

正規表現は、有用であり、洗練されているとはいえ、所詮は表記に基づく検索であって、言語学的単位の検索ではない。検索結果に(言語研究の立場からは)ゴミであるものが含まれてしまうのはしかたがないことである。

英語ではすでに文法解析の加えられたコーパスが実用化されており、品詞を考慮した検索が可能になっている。日本語でも、形態素解析のソフトウェアが実用度を増してきている。現在進められている現代日本語書き言葉均衡コーパスでは、言語学的単位による検索が可能になる見込みである。これらの言語のコーパスの扱いにおいては文法情報を直接利用できるとすれば、正規表現の有用性は相対的に低くなる。

一方、少数言語においてこのようなコーパスへの文法情報の付与がなかなか期待できないとすれば、正規表現の有用性は高いというべきであろう。そのような場合、弱点を理解した上で使うのであれば、正規表現は十分にコスト・パフォーマンスはよい。

なお、正規表現には方言も多く、ソフトウェアによって、書式が微妙に違ったり、特定の書式が使えなかったりする。使っているツール類のマニュアルを参考にして確かめることが必要である。

正規表現の有用性を示すために、付録として、日本語分析における正規表現の応用例を挙げた。日本語を例としたのは、正規表現への習熟には慣れが必要であり、言語的直観の働く言語で試行錯誤を重ねることが有益だからである。

日本語の動詞や形容詞には活用があるから、基本形(終止形)だけの検索では足りず、あらゆる活用形を含めて検索する必要がある。単純な文字列の検索ではこれは不可能であるが、正規表現を使うと、あくまで擬似的にであるが、それを行うことができる。

また、日本語は、表記体系の複雑さにおいて、世界の言語のなかでも極めて特徴的である。この複雑さは、漢字、ひらがな、カタカナという複数の文字種を用いるというだけでなく、文字と音韻論的ないし形態論的単位との対応が複雑であること、正書法がゆるやかであること、正書法からの逸脱が実際にはかなり許容されていることなど、さまざまな要因に起因している。

日本語の検索においてはこのように複雑な状況に対処する必要があるが、まさにそのために、正規表現の有用性と限界を理解するために格好の実験台であると言える。

なお、大名(2008)は、言語研究者が言語分析のために正規表現を利用するためのノウハウを実践的にまとめたものである。対象言語は英語であるが、英語は分かち書きをするので、空白や句読点を手がかりにして比較的簡単に単語を同定することができる。これは、日本語において通常分かち書きをせず、単語の同定が容易ではない状況と大きく異なる。

正規表現の有用性は、言語の形態的特徴や表記上の特徴によって異なる。日本語や英語よりさらに複雑な形態変化を持つ言語においては、正規表現を使っても対処しきれない場合もあるであろう。残念ながら、それは正規表現の限界である。とはいえ、単純に文字列を検索するよりは、能率のよい検索方法を工夫することは可能なはずである。

5. むすび

少数言語の研究において、独力で対処しなければならない場面は多いであろう。コーパスを扱う際には特にそうであろう。しかし、基本的な知識を備えておくことによって、自力で対処できる範囲を広げることができる。テキストツールと正規表現に関する知識は、そのような基本的な知識に属する。

参考文献

本論中で直接参照したもののほか、正規表現を言語研究に利用するために有用な日本語文献を挙げた。ほか、Perl, ruby 等のスクリプト言語の参考書は正規表現の解説があるが、コンピュータのデータを扱う場合のことが主に扱われていて、言語分析用にはあまり役に立たないようであり、ここには含めなかった。また、英語原書からの翻訳書では、当然のことながら、英語以外の取り扱いに配慮しているものはあまりない。ここに挙げた文献の中には現在では入手困難なものも少なくないが、現在でも参照する価値のあるものである。

Dougherty, Dale & A. Robbins, 福崎俊博訳 1997 『sed & awk プログラミング』改訂版 オライリージャパン.

- Friedl, Jeffrey E.F., 長尾高弘訳 2008 『詳説 正規表現』第3版 オライリー
ジャパン.
- I D E A ・ C 2005 『正規表現の達人』第2版 ソフトバンク.
- SE 編集部編 1992 『MS-DOS テキストデータ料理学』翔泳社.
- Stubblebine, Tony, 歌代和正監訳 2004 『正規表現デスクトップリファレンス』
オライリージャパン.
- アスキー書籍編集部編著 1987-88 『MS-DOS を 256 倍使うための本 Vol. 1-3』
アスキー.
- 伊藤博康 1991 『入門 JGAWK』エーアイ出版.
- 伊藤博康 1992 『JGAWK スクリプト集』エーアイ出版.
- 伊藤博康 2001 『テキスト処理と CGI のための Perl プログラミング』エーア
イ出版.
- 岩谷宏 2002 『Java によるテキスト処理入門』ソフトバンク.
- 岩谷宏 2008 『入門 正規表現 検索・置換・テキスト処理に強くなる!』技術
評論社.
- 上田博人 1998 『パソコンによる外国語研究への招待』くろしお出版.
- 上田博人 1998 『パソコンによる外国語研究(I) 数値データの処理』くろしお
出版.
- 上田博人 1998 『パソコンによる外国語研究(II) 文字データの処理』くろしお
出版.
- エイホ, A.V.他 1989 『プログラミング言語 awk』トッパン.
- 大名力 2008『英語語法文法研究のための正規表現によるコーパス検索』私家版.
- 後藤斉 2004 「言語学 オン ザ WEB 第7回 テキスト・ツール」『月刊言語』
第33巻7月号, pp.76-77.
- 酒井純 2005 「古ロシア語文献のコンコーダンス化における技術的問題につい
て：特殊文字を処理できるコンコーダンスプログラムの試作を通じて」
『岡崎女子短期大学研究紀要』 Vol.38(20050325) pp. 95-103.
- 佐藤竜一 2005 『正規表現辞典』翔泳社.

- 佐野洋 2003 『Windows PC による日本語研究法』 共立出版.
- 島和郎他 2005 『正規表現ハンドブック』 ソフトバンクパブリッシング.
- 志村拓他 1993 『AWK を 256 倍使うための本』 アスキー出版局.
- 高橋良明 2005 『Web プログラマのための正規表現 実践のツボ』 九天社.
- 中尾浩他 2002 『コーパス言語学の技法 I. テキスト処理入門』 夏目書房.
- 中島靖 1997 『日本語 TEXT 加工実践ガイドブック』 情報管理.
- 中島靖 1997 『日本語 TEXT 加工実用レファレンス』 情報管理.
- 中島靖 1998 『Perl 使いへの旅立ち—日本語 TEXT 加工入門ガイドブック 改訂新版』 情報管理.
- 中野洋 1996 『パソコンによる日本語研究法入門 語彙と文字』 笠間書院.
- 西谷能英 2002 『出版のためのテキスト実践技法 編集編』 未来社.
- 西谷能英 2005 『編集者・執筆者のための秀丸エディタ超活用術』 翔泳社.
- ハーシー 2004 『正規表現ケーススタディブック』 毎日コミュニケーションズ.
- 平田豊 2007 『正規表現入門 改訂版』 工学社.
- 平山直之 1995 『テキストツールのテキスト』 メロン出版.
- 平山直之 1996 『Perl's パラダイス』 メロン出版.
- 藤岡和夫 2004 『実践実用 Perl』 毎日コミュニケーションズ.
- マークアップ編著 2002 『Web プログラミングのための図解でわかる正規表現』
ディー・アート.
- 宮前竜也 2006 『正規表現ポケットリファレンス』 技術評論社.
- 美吉明浩 1998 『Grep Sed Awk』 秀和システム.
- 目黒編集室 2004 『これだけで身につく Perl 入門 例題 80』 日経 BP ソフトプレス.
- 有限会社ジェイ・シー・エヌ著 2001 『正規表現ハンディリファレンス』 秀和システム.

付録 日本語分析における正規表現の例

サクラエディタ(bregonig.dll 使用時)で使えるもの。

.(ピリオド) 任意の1文字

可.化 「可能化、可視化」など

.っ.り 「ゆっくりに、ぴったり、どっきり」など

[] 文字クラス(択一)

語形のゆれ、表記のゆれ、異表記への対応

怒[濤濤]，中[曾曾]根

ウ[イィ]ルス，ウ[イィ]スキー

グルー[ブヴ]，メ[ーイ]ル

キ[ヤヤ]ノン

複数文字の選択

[白黒赤青]

[-] 範囲指定

[0-9] 数字

[A-Za-z] ラテンアルファベット

[あ-ん] ひらがな

[ア-ンヴー] カタカナ

[亜-腕] 第一水準漢字

[弍-熙] 第二水準漢字

[亜-熙] JIS 漢字

[亜-黒] Windows システム外字を含む漢字

活用 読[ま-むん] (五段活用動詞。音便に注意)

書[か-こい] (これも五段活用だがこれは不可)

書[かきくけこい]

^ 範囲指定の中での否定
[^。?!]+ 文末記号以外の文字の連続

? 0回または1回
送り仮名・音引きの有る無しへの対応

受け?付け?
コンピューター?・?ネットワーク

+ 1回またはそれ以上
[ァ-ヴー]+ カタカナ語

* 0回またはそれ以上
[ァ-ヴ][ァ-ヴー]* カタカナ語
[ァ-ヴ][ァ-ヴー・]* 中黒を含むカタカナ語

| 選択

() グループ化

表記のゆれ。かな表記と漢字表記、複数の語

争い(あ|合)う / 争い[あ合]う

(この|好)む

(バ|ヴァ)イオリン

(白|黒|赤|青|黄色|茶色)い

応用 (組み合わせ)

[読よ]み[出だ][さしすせそ]

[読よ]み[切き][ら-ろっ]

(投げ?|なげ)[込こ][ま-もん]

(関|かか)わ[ら-ろっ]

(暮ら?|くら)[さしすせそ]

(暮ら?|[くぐ]ら)し

(慶[応])(義塾)?大学|慶[応]?大)

エスケープ (メタ文字の特殊な機能をキャンセルする)

¥. ピリオドそのもの

¥? 疑問符そのもの

以下の正規表現はサポートしていないプログラムもある。

後方参照

() ¥1 グループ化したものと同じ文字列を後で¥1 で参照する。さらに、左から¥2, ¥3 など。

([あ-ん][あ-ん])¥1 「ゆらゆら」「ほどほど」など

([あ-ん][あ-ん])¥1¥1 「ゆらゆらゆら」など

([あ-ん])¥1([あ-ん])¥2 「じじばば」など

([亜-黒][亜-黒])¥1 「子供子供」など

([亜-黒])¥1([亜-黒])¥2 「明明白白」など

(+)は¥1[だで、。] トートロジー (「XはXだ」など)

(+)に¥1 を 「失敗に失敗を(重ねる)」「手に手を(取って)」など

回数指定

([あ-ん]{2})¥1 上記の例の書き換え

([あ-ん]{2,3})¥1 「ゆらゆら」「ちゃぷちゃぷ」など

([亜-黒]{2})¥1

先読み

おそらく(=[^。]+だろう)

「だろう」が後続する「おそらく」

否定先読み

東大(?!寺) 「寺」が直接後続しない「東大」
全然(?![^。]*([な無][いかくけ]|ません))
否定が後続しない「全然」

戻り読み

(?<=[^あ-ん])/(?=[^あ-ん])
ひらがな以外が先行し、かつ後続する「の」

否定戻り読み

(?<!天)文学 「天」が直接先行しない「文学」