外国語教育メディア学会（LET）関西支部メソドロジー研究部会 報告論集 第16号
神原 一帆（pp. 1–33）

コーパス研究の思考法
—Sketch Engine を用いたデータの抽出・可視化—

神原　一帆
立命館大学（言語教育センター，R-GIRO）

概要

本稿はコーパスを利用した研究におけるデータの取得法，処理法，そして，その簡易な可視化の手順について論じるものである。本来これらの処理には高度なプログラミング技術が必須であるが，高性能なコーパスインターフェイスである Sketch Engine を利用することである程度のタスクを遂行することができる。本稿ではコーパス研究における基礎知識を概観した上で，複雑なデータ抽出を可能にする正規表現の利用法，そして Sketch Engine の活用法のそれぞれを，具体例と共に提示する。

Keywords: コーパス研究，正規表現，Sketch Engine，メソドロジー

1.　はじめに

理論言語学において，コーパスのデータを利用した実証研究の重要性は繰り返し論じられてきましたが (Fillmore 1990, Glynn & Fischer 2010, Glynn & Robinson 2014, Gries 2010)，これは研究者の内省判断にもとづく研究が主流であったことが背景にあります (Stefanowitsch 2020)[*1]。理論的な動機をもつものだけに限らず，応用志向の研究であっても，それを経験的に進めるためには適切な観察手法が必須となります。

本稿の目的は，(i) コーパスを言語研究などにもちいるための基礎知識と，(ii) 強力なコーパスインターフェイスである Sketch Engine の利用法を概観することです。本稿は次のように構成されます。2. 節ではコーパスを研究にもちいる際の基本事項を確認し，その使用の際の注意点を確認します。3. 節では Sketch Engine で実行可能な様々な機能の特徴について概観します。4. 節では本稿のまとめと，更に先に進むための文献紹介をおこ

ないます。

2.　コーパス言語学の基礎

本節ではコーパス言語学の基礎知識として，複雑なデータ抽出を可能にするアノテーションの有用性と，データ抽出をおこなう際のいくつかの留意点を確認していきます。**コーパス** (corpus) とは (i) 書き言葉や話し言葉などの現実の言語を，(ii) 大規模に，(iii) 基準に沿って網羅的・代表的に収集し， (iv) コンピューター上で処理できるデータとして保存し，(v) 言語研究に使用するものと特徴付けられます (石川 2021, 13)。近年ではコーパスを利用するための様々なサービスが整備されており，簡単にデータを取得することができるようになっています。しかし，このようなツールを適切に利用するためには，そのようなサービスが提供する機能の理解だけでなく，サービスの基本的な構造の理解も必要となります。本節では，2.1 節にてコーパスの構造について確認し，2.2 節にてコーパスを利用する際の留意点について導入します。

2.1　コーパスをつくる「技術」

本節では「コンピューター上で処理できるデータ」というものの形がどのようなものなのかを確認していきます。コンピューター上で利用可能な形式には様々なものがありますが，任意のデータを抽出するためには品詞などのメタ言語情報を付与したものが最も頻繁に利用されます。このようなメタ言語情報は一般的に**注釈** (annotation)（または原語をカタカナにした**アノテーション**）と呼ばれますが，本節ではこの情報の有用性とその利用法の二点について確認していきます。2.1.1 節ではデータ抽出の基礎として正規表現の基本を確認します。そして，2.1.2 節にてコーパスとコーパスインターフェイスの区別を説明します。

2.1.1　データ抽出の基礎としての正規表現

まずはコーパスの利用にメタ言語情報が有用になることを確認するための例として，ある作家による英語で書かれた小説のテキストを一つの text ファイルに集約した原始的なコーパスを考えてみましょう。このファイルは.txt という拡張子を持ち，手持ちのテキス

トエディターによって簡単に検索をすることができるものと仮定します。ここで現代英語の**定** (definiteness) に関心をもつ K さん（仮名）は不定冠詞 *a* または定冠詞 *the* の後ろにどのような名詞がくるのかを観察することを決めたとしましょう [*2]。

このタスクにおいて必須となる技術が**正規表現** (regular expression) です。例えば，あなたのコーパスで *woman* という名詞がどのように使われているのかを調べたいとしましょう。その時，“woman” という文字列をそのまま検索すると，（当たり前ですが）複数形の “women” を検索結果として得ることはできません。ここで求められるのは “woman” と “women” に含まれる a と e のどちらでもマッチするような検索をすることですが，これは正規表現を使って `wom[ae]n` のように実現することができます。正規表現は難しいものの，マスターすれば非常に便利な道具として活用することができます。正規表現の詳しい使い方を説明することは本稿の目的から逸れますが，代表的な表現を淺尾・李 (2013, 259) を参考にまとめたものを表 1 に掲載します。これらの詳細な使い方については適宜 Web 検索等で補ってください [*3]。

ここで本来のタスクの遂行に戻りましょう。今回のタスクの目的は不定冠詞 *a* または定冠詞 *the* の後ろにあらわれる名詞を当該のコーパスから抽出することです。まず，定冠詞の表現は簡単に `(a)|(the)` で検索することができます [*4]。英語であれば語と語の間に一文字分の半角スペースが入りますが，何かのミスや元ファイルのエラーなどで二文字以上の半角スペースが入ったら欲しいデータがとれなくなってしまうので，`+` を用いて `(a)|(the) +` と表現するようにしましょう。

これだけで問題は解決しません。K さんは自分のコーパスに含まれる**全ての**冠詞と名詞の組み合わせを観察することを目的としています。ここで K さんは元のコーパスにどんな名詞が含まれるのかを知らないものとすると，(1) のように名詞の単複を選言の `|` によってつなげていく必要がでてきます。ここで必要になるのは現代英語における名詞の包括的なリストですが，それを列挙することは到底不可能なだけでなく，テキストファイルに含まれる名詞を目視で書き留めていくのは現実的な作業とはいえません [*5]。

(1) a. `((a)|(the)) +(dogs?)`

b. `((a)|(the)) +((dogs?)|(wom[ae]n))`

c. `((a)|(the)) +((dogs?)|(wom[ae]n)|(m[ae]n))`

……

表 1　主な正規表現とその機能

正規表現	機能	例
`?`	直前の文字があってもなくてもよい	`dogs?` によって“dog”や“dogs”にマッチする
`.`	任意の一文字にマッチする	`d.g` によって“dog”や“dug”, “dig”などにマッチする
`+`	直前の文字の 1 回以上の繰り返しにマッチする	`no+` によって“no”や“noooo”などにマッチする
`*`	直前の文字の 0 回以上の繰り返しにマッチする	`dog*` によって“do”や“dog”, “dogg”などにマッチする
`[...]`	[ ] の中に列挙したどれか一文字にマッチする	`wom[ae]n` によって“woman”や“women”にマッチする
`[^...]`	否定	`[^s]` によって“s”以外の一文字に，`[^A-Za-z]` によってアルファベット以外の一文字にマッチする。
`...|...`	|の前後に生起する文字列の選言	`dog|cat` によって“dog”や“cat”などにマッチする。
`(...)`	文字のグループ化	`(dogs?)|(cats?)` によって“dog”, “dogs”, “cat”, “cats”などにマッチする。
`\w`	アルファベット，アンダーバー，半角数字（`[a-zA-Z_0-9]` という記述と同じ）	`\w+` によって“dog”, “kazy3024”, “whisky”などにマッチする。

ここで便利になるのがアノテーション付きのコーパスです。アノテーションとは何らかの基準によって分割された事例に対して何らかのメタ表現を付与したものを指します (cf. Pustejovsky & Stubbs 2013, Ch.1)。このようなアノテーションの代表例としては品詞情報が挙げられます。例えば,「名詞」のような品詞は{*dogs, hypothesis, noodle, ...*}のような表現を包摂するような一種の**クラス** (class) に対応します。言語学において，品詞とい

うものは**統辞的な関係** (syntagmatic relation) と**範列的な関係** (paradigmatic relation) にもとづいて同定されます (黒田 2003, 守田 2013)。近年では**自然言語処理** (Natural Language Processing; NLP) の発展によって品詞情報を自動で付与するプログラムが利用可能となっていて，様々なものを利用することができます。ここで一つ簡単な具体例を (2) に挙げます。(2a) は元の文, (2b) は品詞解析機 [*6] を適用した結果です。(2b) では (2a) に含まれるそれぞれの語に対して <...>...</...> という形で nnp, vbd などの品詞タグが付与されています。ここで重要なことは nnp, vbd などの品詞タグは元々のデータに含まれているものではなく，元のデータの特徴を記述するための表現として機能しているという点です。仮に K さんが自分のデータに対して同様の解析機を適用し，その結果に対して `<det>[a-zA-Z]+</det> +<nn[a-zA-Z]*>[a-zA-Z]+</nn[a-zA-Z]*>` のような正規表現を用いた検索をすれば，簡単に冠詞の直後に名詞が生起する事例を検索することができます。

(2) a. Alice thought that Bill said that Charlotte believed that David was a liar

b.
```
<nnp>Alice</nnp> <vbd>thought</vbd> <in>that</in> <nnp>Bill</
    nnp> <vbd>said</vbd> <in>that</in> <nnp>Charlotte</nnp> <
    vbd>believed</vbd> <in>that</in> <nnp>David</nnp> <vbz>was
    </vbz> <det>a</det> <nn>liar</nn>
```

ここでのアノテーションとはあくまで当該の表現に対して付与されるメタ言語情報であり，その情報は品詞のような文法的な情報だけに限らず，音韻的，統語的，意味的，語用論的なものを付与することも（理論的には）可能です。しかし，そのようなアノテーションをおこなうためには常に一貫した基準が必要で，特に意味が関わるようなアノテーションについては様々な議論があることには注意しましょう (cf. Fellbaum & Baker 2013)[*7]。

2.1.2 コーパスとコーパスインターフェイス

アノテーションを付与されたデータを利用すれば全ての問題が解決するわけではありません。(2b) では簡単な事例を挙げましたが，一般的にコーパスとして公開されているデータには非常に多くのメタ言語情報が含まれます。例えば，品詞 (e.g., *VERB*)，語彙素 (e.g., *break*)，屈折形 (e.g., “broke”)，そのデータの出典 (**使用域** (resister)) といった膨大なアノテーションが付与されたデータを人力で解釈することは到底不可能です。この理

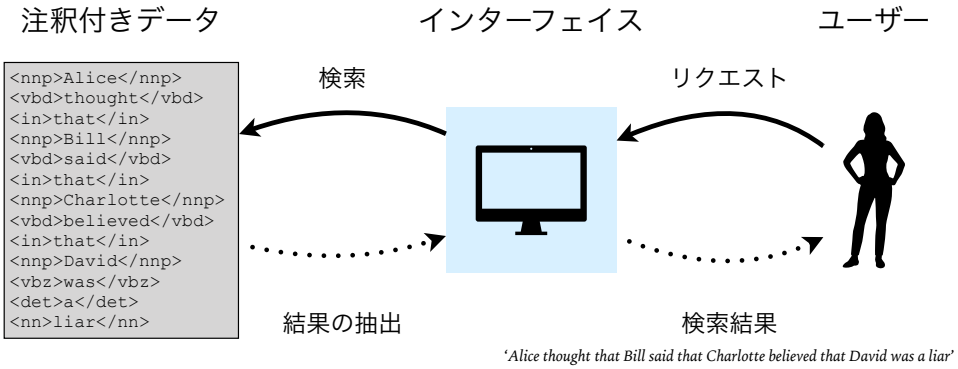


図 1　コーパス，コーパスインターフェイス，ユーザーの関係

由から，コーパスのデータを扱うためには必要に応じてコーパスのデータを処理するための何らかの機構が必要になります。このようなコンピュータによる処理を前提としたデータを扱うための機構のことは一般的に，**インターフェイス** (interface) と呼ばれます [*8]。この機構を簡単に図示すると図 1 のようになります。

コーパスインターフェイスとは機械処理が前提となっているコーパスのデータを人が扱いやすいように処理する機構を指します。図 1 では，ユーザーが (2a) の文をコーパスのデータから抽出するためのリクエストを送信し，インターフェイスが当該の条件に合致する事例である (2b) を検索・抽出し，それをユーザーが閲覧可能な (2a) のような形に成形する過程を示しています。

本来であれば，コーパスインターフェイスの構築にはプログラミング言語の習得が必要不可欠です。なぜならば生のアノテーション付きのコーパスデータを目視で確認することは現実的ではないからです。しかし，幸いなことにコーパスのデータにアクセスするためのサービスが数多く存在します。本稿で扱う Sketch Engine もその一つで，3. 節にてその使い方を簡単にみていきます。他にも言語学において頻繁に用いられるサービスとしては English-Corpora.org が挙げられますが，その詳細は長谷部 (2020) を参照してください。

ただし，Gries (2017, 4) が主張するように，有料・無料の是非を問わず，Web 上のコーパス検索アプリケーションは急な仕様変更や機能停止に直面する可能性が否めません。一定の検索結果が常に得られないという事態は，最悪の場合再現性が担保できなくなるという研究倫理上の大きな問題に直面する可能性が常に残ります。この理由から Gries はプログラミング言語の習得が全ての言語学者にとって必須の技能であるという立場をとります [*9]。

現実的には，（言語教育や理論研究を含む広義の）言語研究に携わる全ての研究者がある程度のプログラミングの技能を習得するまでにはまだ時間がかかるでしょうし，中にはその過程を苦痛に感じる研究者も多いでしょう [*10]。しかし，自分が普段利用しているアプリケーションに利用したいデータが含まれない場合，データの取得は自力でおこなう必要がでてくることは念頭に置くべきでしょう。また，使用するデータについてはどのような事例をどのような基準で分析し，どのような結果が得られたのかについて論文等で詳細に論じるだけでなく [*11]，実際に利用したデータをすぐに提示できるようにする工夫などが必要となります。

仮に，データの取得，分析，考察といった各サイクルの透明化の試みを極限まで発展させるのであれば，言語学における**オープンサイエンス** (open science) 化を促進することができるでしょう。特に既存のコーパスを用いた言語研究であれば（構築に関わらない限り）倫理的な問題に抵触することはほとんどありません [*12]。また，収集したデータに対して意味的なアノテーションを与えるような研究では再現性が特に問題になるため，データの公開は非常に重要な試みとして評価することができるでしょう [*13]

2.2　コーパスをつかう「技術」

前節にて詳しく述べたように，コーパスの元データには品詞などのメタ情報が付与されている場合が多く，Sketch Engine に限らず，多くのコーパス検索アプリケーションはこのメタ情報を用いた検索が可能になっています。本節ではメタ情報の利用は複雑な条件に基づく検索を可能にする反面で，検索条件の厳しさと検索結果の量は一般的にトレードオフの関係にあることを確認します。

コーパス検索アプリケーションは様々な条件にもとづく検索を可能にしますが，複雑な検索にはある程度の慣れと技術が必要になります。どのようなデータを抽出するのかは各々のタスクによって異なりますが，一般的に検索条件の厳しさと検索結果の量はトレードオフの関係にあります。つまり，厳しすぎる検索条件では十分なデータが取れませんし，簡単すぎる検索条件では必要以上のデータが取れてしまいます。この関係は集合論における外延の多さと内包の豊富さの関係と並行的なものとして理解することができます (i.e., 被覆率と精度の関係）[*14]。

この関係がどのようなものであるのかを実際の事例を用いて経験的に確認してみましょ

表 2　各語の頻度と条件の厳しさに応じた頻度

	kill	*read*	*seek*	*test*	*watch*
1	15,148	27,841	16,606	22,685	21,912
2	14,973	27,710	16,605	6,896	18,934
3	1,423	4,354	2,006	1,198	4,430
4	1,036	3,315	1,357	934	3,362

う。ここで恣意的ではあるものの，*kill, read, seek, test, watch* という五つの語を挙げ，それぞれの事例に対して 'British National Corpus (BNC), tagged by CLAWS' を用いて (3) に挙げた四つの条件で検索した際に頻度がどのように変動するのかを調査してみました。その結果，表 2 にあるような頻度が確認されました。この検索条件の厳しさと頻度を R Core Team (2022) をもちいて可視化したものが図 2 です。ここでは条件が厳しくなるほどその頻度が少なくなるという大まかな傾向が確認できます。

(3)　i. {*test, tests, tested*}のように，品詞を問わない様々な屈折形を含んだ語 *xxx* のあらゆる事例 (i.e., `[lemma="xxx"]`)

ii. {*test, tests, tested*}のような様々な屈折形の語のなかでも，動詞としてタグづけされた語 *xxx* のあらゆる事例 (i.e., `[lemma="xxx" & tag="V.*"]`)

iii. {*test, tests, tested*}のような様々な屈折形の語のなかでも，動詞としてタグづけされた *xxx* の直後に冠詞{a, the}が後続するあらゆる事例 (i.e., `[lemma="xxx" & tag="V.*"][tag="AT0"]`)

iv. {*test, tests, tested*}のような様々な屈折形の語のなかでも，動詞としてタグづけされた *xxx* の直後に冠詞{a, the}と名詞が後続するあらゆる事例 (i.e., `[lemma="xxx" & tag="V.*"][tag="AT0"][tag="N.*"]`)

研究者は仮説を検証，または形成するためにデータを取得，ないしは分析します。どのような目的の下で調査をおこなうにせよ，研究者は得られた結果から確認できる何らかの一般化を提示する必要があります。この一般化という過程を簡単に特徴付けると，当該のデータに含まれる共通性を見出すことに対応します [*15]。あるデータからどれだけ「興味深い」傾向を見出せるかは分析者の理論的立場（や経験年数，センス）などによって大きく異ります。もしある傾向がその分析者の理論から予測できないようなものであれば，そ

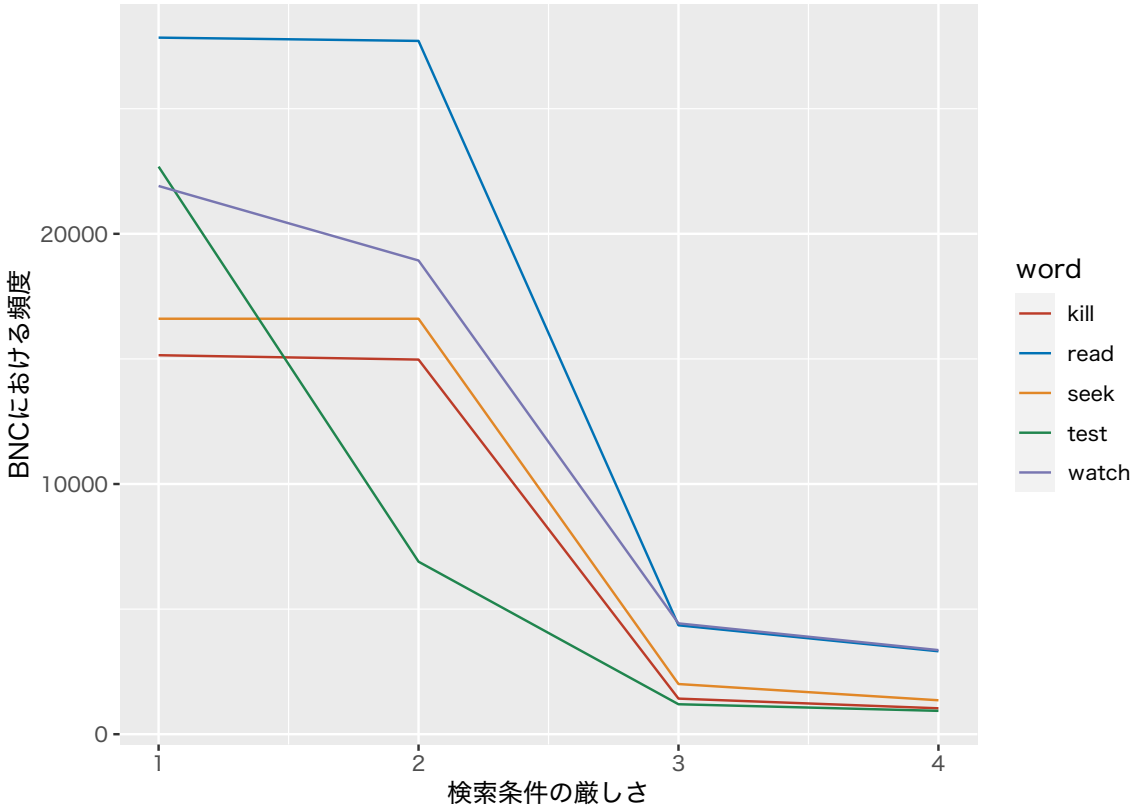


図 2　各語の頻度と条件の厳しさの関係

れは（典型的に）理論的な議論の余地がある「興味深い」ものとなりますが，そうでない場合は「興味深い」ものとしてみなされない可能性があります [*16]。しかし，一般的にそのデータの規模と一般化のしやすさには上でみたようなトレードオフの関係が成立することが容易に想像できます。つまり，データの規模が大きくなればなるほど，そこから共通性を見出すことが難しくなり，データの規模が小さくなればなるほど，そこから共通性を見出すことが簡単になりやすいのです。

ここで分析対象の量とその分析結果の一般化の難易度の関係をまとめると表 3 のようになる。一般化が困難であるというのは，その一般化がより複雑なものになりうるということに対応する。つまり，適切な規模のデータからは「P ならば Q である」のように前件と後件が一つの命題となっている単純な一般化を導くことができますが，必要以上に大規模なデータからは「P_1 かつ P_2，または P_3 であれば Q である」[*17] のように前件と後件が複合的な命題からなるような複雑な一般化を導かざるを得ません [*18]。

少ない事例から一般化を蓄積することの方が分析者の負担量は減る反面で，少ない事例

表 3　検索条件の厳しさと一般化の難易度

検索条件	事例の数	一般化
ゆるい	多い	困難
厳しい	少ない	容易

からの一般化は言明の**一般性** (generality) の欠如という問題に直面することになります。言語学は言語の科学である以上，特定の個人の言語知識がどうなっているのか，ということよりもある言語 L_i を使用する話し手全体がもつ傾向を明らかにしようとするものです。つまり，あまりに特殊な事例ばかりを扱っていても，L_i の全体の傾向を明らかにすることはできないということです。統計的には，調査を実施するまえに (i) 有意水準 (α)，(ii) 効果量，(iii) 検定量 ($1-\beta$)，(iv) サンプル・サイズのうちの，(i–iii) を決めてしまえば (iv) を決定することができる (水本・竹内 2010, 56–57)。一つの解決法としては，自分の決めた調査水準によって得られるデータの総数が統計的に得られたこのサンプル数と概ね一致するかどうかを検討することが挙げられるだろう。ただし，修辞的な表現などは出現頻度が低い場合も多々あるため，「統計的にも妥当な手法を用いて比喩表現を探しましたが，見つかりませんでした」のような事態に陥るのは馬鹿らしいので，分析対象の抽出には慎重になるべきでしょう [*19]。

このような検索条件の厳しさと検索結果の量の関係を踏まえると，コーパス基盤の研究には (4) に示す難しさが付きまといます。この問題をどのように克服するのかは各々が解決すべき問題で，非常に泥臭い作業になっていきます。これに関しては，それぞれが 3. 節で概観する Sketch Engine のようなツールを使って試行錯誤していければと思います。

(4)　適切な厳しさの検索条件によって，言語学的に有意義な一般化が可能になるような量のデータを抽出する必要性

3.　Sketch Engine

本節ではコーパス検索アプリケーションの一つである Sketch Engine の機能を概観していきます (Kilgarriff et al. 2004, 2014)。このアプリケーションは高度な検索が可能で，様々な目的をもった研究者にとって非常に有用なツールとなることは間違いないでしょう

(cf. 黒田 2017)。

2022 年 9 月現在の Sketch Engine では 94 カ国語の 742 個のコーパスが利用可能となっています。以降では ‘British National Corpus (BNC), tagged by CLAWS’ をコーパスとして利用しながら，以下に挙げる機能についてその概要と利用可能性とその注意点について簡単に論じていきます。このリストは完全なものではありませんが，コーパスをもちいた研究をおこなうためには事足りるでしょう。なお，利用可能性は神原個人の関心に多大な影響を受けているため，どのようなタスクに各機能が有用かどうかは各自で考えてもらえると嬉しいです。

3..1. Word Sketch

3..2. Word Sketch Difference

3..3. Thesaurus

3..4. Concordance

3..5. Wordlist

3..6. N-grams

3..7. Keywords

3.1 Word Sketch

■概要　Word Sketch とは，検索対象が当該のコーパスにおいてどのような振る舞いをみせるのかを要約したものを指します。図 3 に名詞 *dog* の Word Sketch の検索結果を記載します。この Word Sketch は検索対象とする語の頻度によってもその結果の豊富さが大きく変わることに注意してください。

■利用可能性とその注意点　ある表現の大まかな傾向を把握したいときに便利です。例えば，*replace* の主語や目的語には何が生起しやすいのかということを調べるのには有効な手段となる。しかし，共起語の傾向の確認は大雑把な理解しかもたらさないことには注意が必要です。

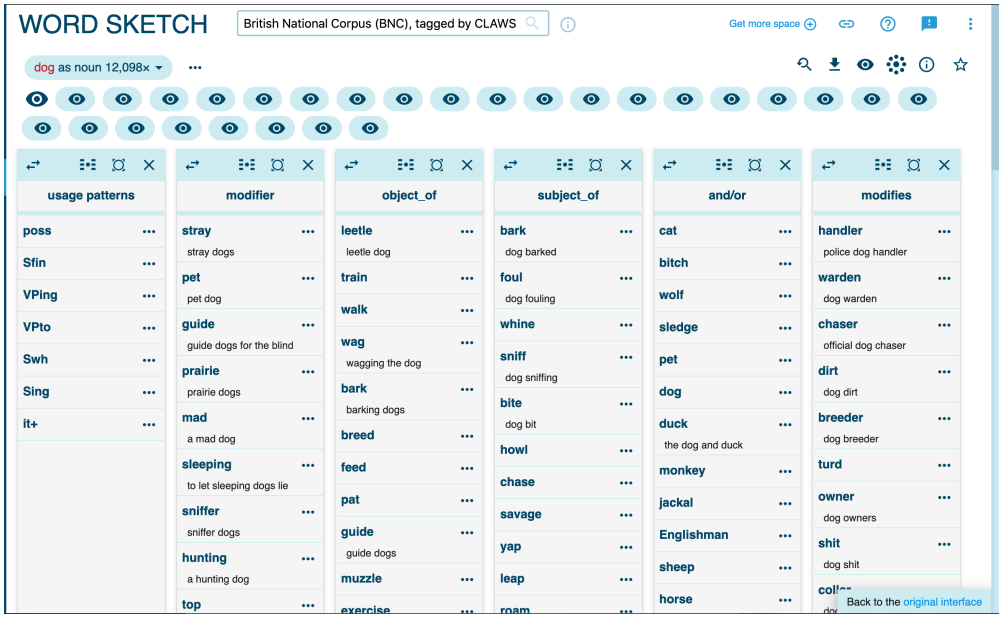


図 3　BNC における名詞 *dog* の振る舞い

3.2　Word Sketch Difference

■概要　Word Sketch Difference とは，二つの語彙同士のコーパス上での振る舞いを要約したものに該当します。図 4 に BNC 上での名詞 *dog* と *cat* の振る舞いを要約したものを記載します。緑色の着色は *dog* に特有なものを，赤色の着色は *cat* に特有なものを示しています。

■利用可能性とその注意点　Word Sketch Difference が最も有用になるのは語どうしに成立する**意味関係** (semantic relation) の分析でしょう (cf. Cruse 1986, 2011, Murphy 2003, 2010)。一般的に意味の類似性は共起語の類似性と相関があることが知られていますが，この詳細についてはまだ不明瞭な点が多いです。例えば，Murphy (2010, 109) は類義関係にある語のペアとして *sofa* と *couch* を挙げています。これを Word Sketch Difference で調べてみると，形容詞 *soft* は *sofa* としか，形容詞 *brown* は *couch* としか生起しないことが簡単にわかります。

ですが，意味の類似性と共起語の類似性の相関関係は完璧なものではないことには注意が必要です。例えば，先の例を用いて「*soft* となりうる椅子は *sofa* だけである」や「*brown* となりうる椅子は *couch* だけである」という結論を導くのは明らかに間違っています

図 4　BNC における名詞 *dog* と *cat* の振る舞い

(Kambara & Yamanaka 2023, 53–55)。コーパスをもちいた分析は研究者の直観が及ばないような事実を明らかにするものの，安直な結論に飛びつきやすくなることにも注意がいります。どのような頻度情報であればその表現の意味的な相違点を明らかにできるのかということはコーパスだけで明らかにすることはできません (cf. 神原 2021, 129–130)。

3.3　Thesaurus

■概要　Thesaurus は Sketch Engine によって自動的に生成された類義語のリストを指します。通常の単語リストとして出力することも可能ですが，図 5 のような可視化もできます。ここでは機械的に似た統語環境に発生しうる語を類義語としてまとめているだけなので，[*dog–doggy*] のような対が必ずしも得られるわけではないことに注意が必要でしょう。

■利用可能性とその注意点　Thesaurus の利用可能性としては，作例の支援などが挙げられます。黒田 (2011b) は，任意の数の変項を用意し，それぞれの変数に様々な表現をいれるという作例の作成方法を提案しています。(5) に動詞 *give* の例を挙げます。ここでは動詞 *give* に一つから三つの変項を用意し，それぞれの変項に様々な表現をいれています。

(5)　a.　Alice gave ____ a car.　($n = 1$)

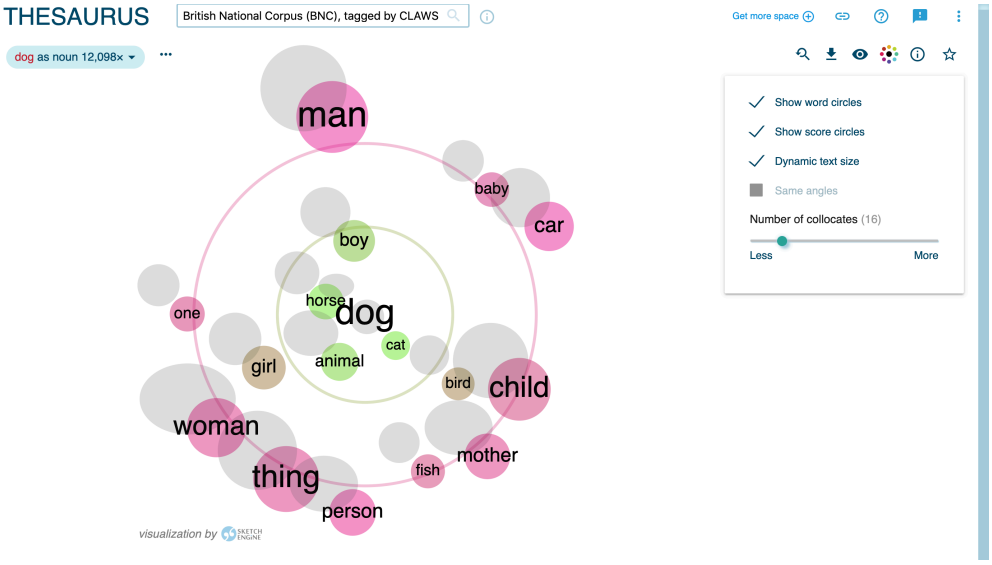


図 5　名詞 *dog* の類義語

i. Alice gave Bill a car.

ii. Alice gave Charlotte a car.

……

b. Alice gave ____ ____. ($n = 2$)

i. Alice gave Bill a bicycle.

ii. Alice gave Charlotte a book.

……

c. ____ gave ____ ____. ($n = 3$)

i. David gave Bill a bicycle.

ii. Elizabeth gave Charlotte a book.

……

ここでどのような変項と表現を用いるのかは例文の作成者に委ねられることになります。ここで Thesaurus の機能をもちいて作成した単語リストから様々な表現をいれることで，より客観的な作成をおこなうことができるでしょう，なお，当該の表現の形式的な類似性と容認度の変動にどのような対応関係がみられるのかを観察することは選択制限 (selective restriction) の分析に有用なものとなります。

3.4 Concordance

■**概要** Concordance は検索対象が含まれる事例を KWIC (*k*ey*w*ord *i*n *c*ontext) の形，ないしは文単位で抽出する機能を指します。この機能は通常の検索機能とほぼ同じですが，より高度な検索も可能になっています。主な検索オプションは (6) の通りです。

(6) a. simple: 語単位で自動的に事例を検索することができる。*dog* であれば，名詞判定をし，複数形も含めた事例も抽出される。

b. lemma: 語彙素単位で自動的に事例を検索することができる。名詞形の *test.n* ではなく，動詞形の *test* のみを検索する場合のみなどに使用することができる。

c. phrase: フレーズ単位で自動的に事例を検索することができる。句動詞の *take out* を含む事例を検索することができるが，*took out* や *taking out* といった事例は含まれないことに注意。

d. CQL: Corpus Query Language と呼ばれるデータベース検索用の言語を用いることで，複雑な条件の事例を検索することができる。

(6d) の CQL は正規表現を含む検索式によって複雑な検索を可能にする人工言語です。この言語では大括弧 [ ] が一つの単位をなし，その中に様々なタグを入力することができる [*20]。Sketch Engine では通常の品詞情報に基づく検索式だけでなく，3.1 で導入した Word Sketch の情報も利用することができます。(7) にその使用例を挙げます。

(7) 便利な CQL:

a. *'bring NP for NP'* という形をとる事例:

```
[lemma="bring"] [tag!="IN"]{0,2} [tag="N.*"] [lemma="for"]
[tag!="IN"]{0,2} [tag="N.*"]
```

b. *'NP is like a shark'* という形をとる事例:

```
[tag="N.*"] [lemma="be"] [lemma="like"] [tag="DT"] [lemma="shark"]
```

c. 目的語に *dog, elephant, animal* のどれかを取る動詞 *kill* の事例 [*21]:

```
[ws("dog-n|elephant-n|animal-n", ".*object.*","kill-v")]
```

この機能によって得られた結果には (8) に示すような様々な操作を適用することができます。

(8) a. DOWNLOAD: 得られた事例のダウンロードができる。形式としては.txt, .csv, .xlsx, .xml を選択することができる。現時点のブラウザの情報を pdf 形式で保存することも可能。

b. CONCORDANCE ANNOTATION MODE: 各事例に対して任意のアノテーションをおこなうことができる。各事例の分類などに用いることができるが，このアノテーションの結果をダウンロードできないことには注意すること [*22]。

c. VIEW OPTIONS: 各事例に含まれる様々な属性や構造を表示することができる。これは元のコーパスファイルに含まれるタグによって生成される。

d. GET A RANDOM SAMPLE: 最大で 10,000 件の任意の数のサンプルを抽出する。200 件のダウンロード結果と 201 件のダウンロード結果は異なるので注意すること。

e. SHUFFLE LINES: 条件に合致する事例をランダムにシャッフルする。このランダム化はシャッフルする回数に応じて同じ結果が得られるようになっている [*23]。

f. SORT: 検索対象を含む前語三語の中から，特定の条件に従った並び替えを実施する。

g. FILTER: 当該の結果から特定の条件に合致するもののみを抽出する。

h. GOOD DICTIONARY EXAMPLES: Kilgarriff et al. (2008) によって定義された GDEX (*g*ood *d*ictionary *ex*ample) 順にもとづく並び替えをする。

i. FREQUENCY: 当該の表現の前文脈，当該の表現，当該の表現の後文脈に生起する表現の頻度リストを (i) 語形，(ii) 品詞，(iii) タグ，(iv) レマの四つの基準によって生成する。

j. COLLOCATIONS: 当該の表現の前後五語までに生起する表現の頻度リストを取得する。

k. DISTRIBUTION OF HITS IN THE CORPUS: 当該の表現が使用しているコーパスに含まれるファイルのどの位置に生起するのかを可視化する。

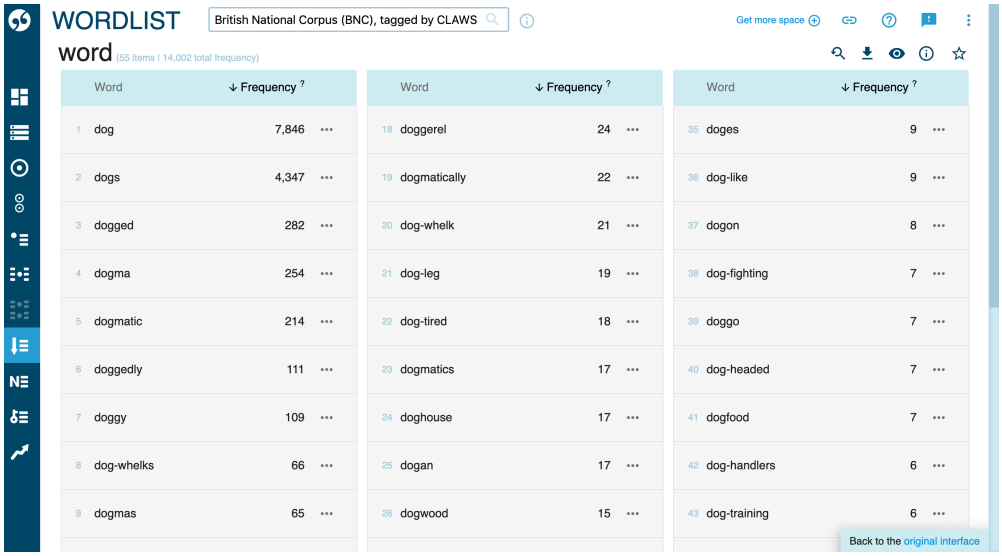


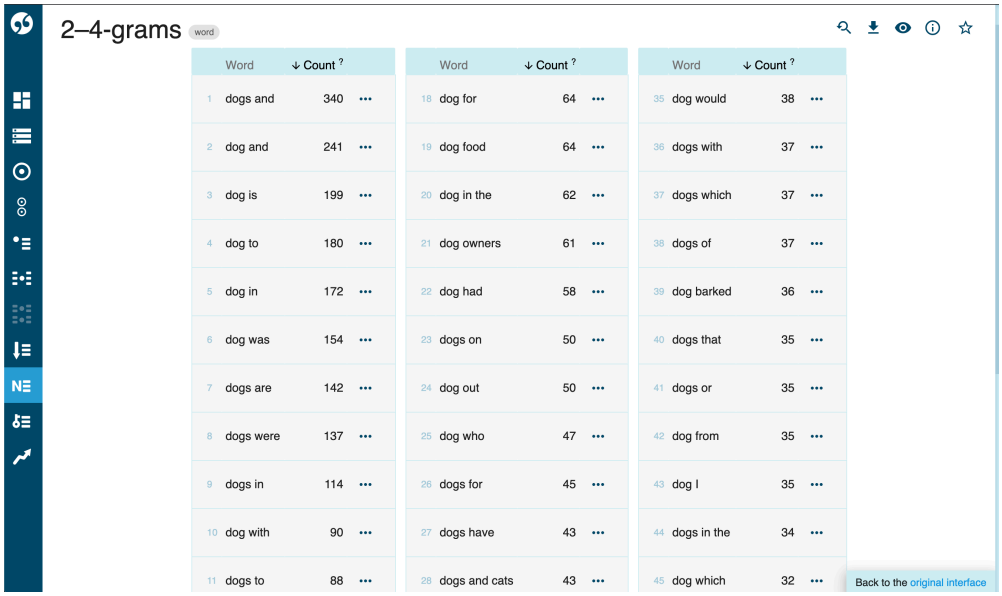
図 6　名詞 *dog* の wordlist [words, starting with]

■利用可能性とその注意点　個人的に一番よく使う機能ですが，(i) 言語研究のためのデータ抽出と，(ii) 教材の開発で用いることがです。(i) に関しては，当該の条件に合致する事例を抽出し，それに対して意味的な特徴を中心とするアノテーションを加えるための一次データとして用います。(ii) については (8h) の GDEX によって並び替えをした上で単語テストを作成することが多いです。いずれの場合も検索結果をダウンロードする際にはブラウザ上の表示が KWIC か Sentence なのかを確認してからでないと，予想通りのデータが得られない場合があることに注意が必要です。

3.5　Wordlist

■概要　Wordlist は任意の条件にもとづく頻度表を作成する機能を指します。頻度表の作成には{lemmas, nouns, verbs, adjectives, adverbs, pronouns, conjunctions, prepositions}という 8 つの条件 (A) と{all, starting with, ending with, containing}という 4 つの条件 (B) の全 32 通り組み合わせが利用できます。図 6 は (A) words，(B) starting with の条件で *dog* を検索した結果のサンプルです。

■利用可能性とその注意点　ある表現の使用傾向などを確認するためには便利なものになります。例えば，英語の形態素 re-から始まる表現の頻度と，それが付与される語幹の関



2–4-grams word

	Word	Count		Word	Count		Word	Count
1	dogs and	340	18	dog for	64	35	dog would	38
2	dog and	241	19	dog food	64	36	dogs with	37
3	dog is	199	20	dog in the	62	37	dogs which	37
4	dog to	180	21	dog owners	61	38	dogs of	37
5	dog in	172	22	dog had	58	39	dog barked	36
6	dog was	154	23	dogs on	50	40	dogs that	35
7	dogs are	142	24	dog out	50	41	dogs or	35
8	dogs were	137	25	dog who	47	42	dog from	35
9	dogs in	114	26	dogs for	45	43	dog I	35
10	dog with	90	27	dogs have	43	44	dogs in the	34
11	dogs to	88	28	dogs and cats	43	45	dog which	32

Back to the original interface

図 7　名詞 *dog* から始まる n グラム $(2 \leq n \leq 4)$

係を調査する際には有効な手段となると考えられるでしょう。しかし，このような形で得られた事例はあくまで形式的な共通性によって抽出されたものであるため，形式が類似しているからといって，意味的にも類似しているという保証がないことには注意すること。

3.6　N-grams

■概要　n グラムとは，機械的に抽出される n 組の形式のことを指します。この 1 組として計算される形式を仮に「語」とすれば，*'Alice likes a dog'* という文の n グラムは (9) のようなものになります。

(9)　a.　1（ユニ）グラム: {Alice}, {likes}, {a}, {dog}

b.　2（バイ）グラム: {Alice, likes}, {likes, a}, {a, dog}

c.　3（トライ）グラム: {Alice, likes, a}, {likes, a, dog}

…

Sketch Engine を用いることで，任意の n グラム $(2 \leq n \leq 5)$ を頻度順に取得することができる。図 7 は名詞 *dog* から始まる n グラム $(2 \leq n \leq 4)$ の頻度表です。これらの頻度表の取得には最低頻度の設定などの細かな設定が可能です。

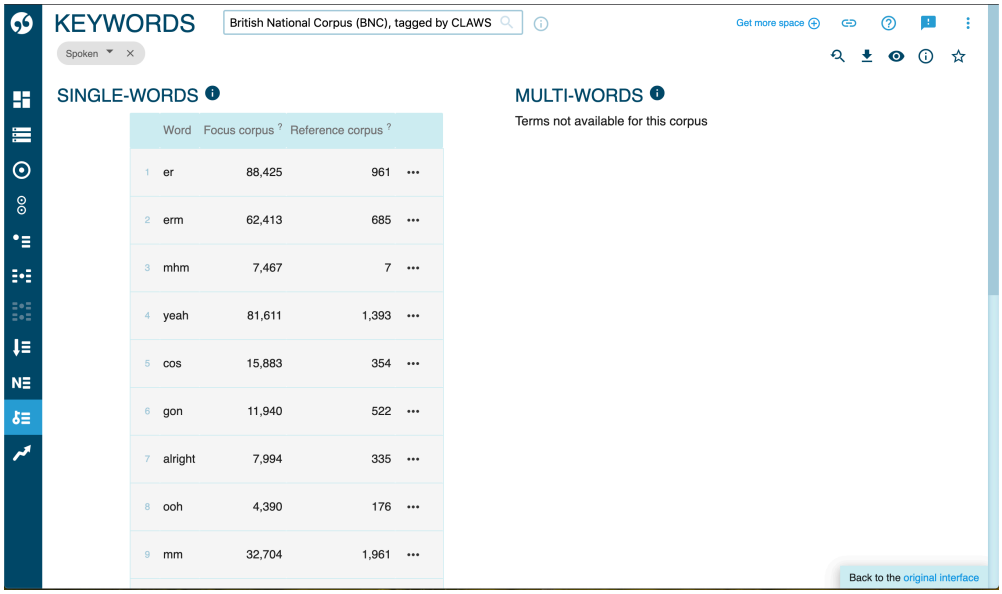


図 8　BNC コーパスでの「書き言葉」と比較した時の「話し言葉」のキーワード

■利用可能性とその注意点　n グラムは e_i という表現に後続する e_{i+n} の組を指すため，「e_i といえば，次は……」という情報に対応すると考えることもできるでしょう。頻度の高い n グラムはそれだけ経験する可能性が高い事例として扱うことができるため，反応速度が速くなる等の頻度効果が観察できるのかもしれません (cf. Baayen 2010)。このような観点から実験をおこなう際には刺激の作成にこの情報を利用することができるでしょう。

3.7　Keywords

■概要　Keywords では二つのコーパスにおいて，他方よりも一方のコーパスで頻出する語のリストを得ることができます。図 8 では，BNC のサブコーパスである「話し言葉」と「書き言葉」の間で，「話し言葉」に特有な語を挙げたものです。この比較するコーパスは様々なものを利用することができます。

■利用可能性とその注意点　コーパスを用いた研究に対するよくある批判として，「コーパスやレジスターが変われば分析結果が変わるだろう」というものがあります。Gries & Divjak (2010, 347–348) はこの批判への応答として，高頻度の表現の分布などはコーパスによって大きく変動するものの，態の交替（i.e., 能動態 vs. 受動態）のような文法的な構文の分布などはコーパス間に統計的な差が観察できない研究が複数あることを指摘してい

ます。このようにコーパスごとの共通性と相違点を模索する研究などではこの機能は有用になるでしょう。しかし，この keywords だけでは詳細な分析を展開することができないことには注意する必要があります。

4. おわりに

本 WS ではコーパスを利用する際に便利になるコーパス言語学の基礎として正規表現の基礎，アノテーションの必要性，コーパスとコーパスインターフェイスの区別について説明しました。その上で，Sketch Engine の代表的な七つの機能を概観しました。いずれの場合も内容としては必要最低限のことしか扱っていないため，研究や教育の場で利用するためには以下に挙げるような文献を参考にしてください。

■プログラミング関係　神原は Progate をオンライン教材としてプログラミングの勉強に使いました。どんな言語を用いるにせよ，基本的な利用法を学んでおくことは役立つでしょう。個人的には Ruby に一番慣れているのですが，Ruby を用いたテキスト分析に関しては田野村 (2012) が便利になるでしょう。言語学者向けの R へのイントロとしては Gries (2017) の二章が便利ですが，さらに先に進むためには松村ほか (2021) などが役立つでしょう。淺尾・李 (2013) は Python をもちいたテキスト処理に関して非常にわかりやすい解説になっています [*24]。

■コーパス言語学・統計関係:　McEnery & Hardie (2012), 石川 (2021) はコーパス言語学の教科書として代表的なもので，より近年のものであれば Stefanowitsch (2020) などが挙げられます。特に McEnery & Hardie (2012) は認知言語学をはじめとする機能主義的な言語学との関連を詳細に論じています [*25]。また，Stefanowitsch は Gries と共に**コロストラクション分析** (collostructional analysis) を提唱した研究者としてよく知られています (Stefanowitsch & Gries 2003)。

■統計・数学関係:　統計の教科書については（本当に）たくさんのものがあるので，ここではコーパス分析に関わるようなものを中心に挙げます [*26]。日本語であれば，石川ほか (2010) がよいですが，竹内・水本 (2023) も多くの分析法を具体例とともに解説しているため非常に分かりやすいです。英語であれば，Gries (2021) が非常に多くの事柄を扱っていて非常に勉強になりますが，R の操作を同時並行で学ぶという意味で難しい感じる方も

いらっしゃると思います。Levshina (2015) や Glynn & Fischer (2010) などが参考になるでしょう。また，近年では統計モデルを用いたコーパス分析が主流となっていますが，その感覚を掴むためには久保 (2012) や Winter (2019) などが大変参考になります。

謝辞

本稿は 2022 年 9 月 10 日に電子語学教材開発研究部会 第 38 回研究会「これから始めるコーパス分析: Sketch Engine 活用術」で配布した資料を加筆・修正したものです。企画をしていただいた木村修平氏（立命館大学），世話人の近藤雪絵氏（立命館大学），ならびにワークショップに参加して頂いた方々に感謝いたします。また，本稿の前身となった資料は京都大学の谷口研究室（京都大学 認知言語学系研究室）で開催された自主ゼミでも「コーパス基盤アプローチへの招待: Sketch Engine の活用をとおして」というタイトルで利用しました。そこでの議論に参加された方々にも感謝します。最後に本稿の投稿を勧めて下さった水本篤氏（関西大学）に感謝します。本稿に残る誤りは全て筆者によるものです。

注

[*1] 本稿は 2022 年 9 月 10 日に電子語学教材開発研究部会第 38 回研究会「これから始めるコーパス分析: Sketch Engine 活用術」での配布資料を加筆・修正したものである。Workshop での配布資料という特性上，「ですます」調による記述になっている。

[*2] これは仮定の研究方法であって，2.2 節でみる理由から，実際のコーパスで似たことをするととんでもない量のデータを扱うことになるのでお勧めしません。

[*3]Web 検索だけでなく，近年の Chat GPT などの生成 AI サービスを利用することで簡単に正規表現の式を取得することができます。単純なものであればすぐに正解を取得することができますが，複雑な検索式を生成するのは難しくなります。他のタスク同様に，AI を利用したコードは鵜呑みにせず，表 1 を参照しながら適宜修正をおこなってください。

[*4] 厳密には，このような検索式には二つの問題があります。一つは大文字の “A dog” のような事例を抽出できないという点で，もう一つは不定冠詞の異形である *an* を取得することができないという点です。ここでは議論を簡易にするため，そのような事情は考慮

しないことにします。

[*5](1) の正規表現の記述には冗長な部分が含まれてますが，ここでは度外視します。なお，(1a–c) の冒頭には半角スペースが含まれていることに注意してください。

[*6] ここで使用した品詞パーサーは EngTagger で，`https://github.com/yohasebe/engtagger` から入手可能です。

[*7] 生成 AI の発展によって，近年は事情が大きく変わってきていますが，理論言語学者が NLP に貢献できる一つの方向性としてはアノテーション方針の開発などが挙げられるでしょう。神原 (2021) は語の意味に関するアノテーション体系の開発・構築に関わる理論研究を事例研究とともに提示しています。なお，これに関する議論については黒田 (2012) なども参照してください。

[*8] 国立国語研究所が提供しているコーパスである現代日本語書き言葉均衡コーパス (Balanced Corpus of Contemporary Written Japanese; BCCWJ) と中納言に関していうならば，BCCWJ はコーパスデータで，「少納言」がそのデータにアクセスするためのインターフェイスとして機能していますが，神原 (2017) のような研究はこの区別がわかっていません。

[*9] 私自身は Ruby や R を使えばある程度のことができますが，それまでには数年の時間がかかりました。

[*10] とはいえ，生成 AI を上手に活用することで以前ほどの「プログラミング言語の流暢さ」が求められることはなくなるでしょう。しかし，出力されたコードのエラーを修正するためには基本的な知識が必要になるため，ある程度の学習は必要になることには注意しましょう。

[*11] この辺りの話は神原の出身大学院の後輩のために執筆した神原ほか (2019) にて詳しく論じました。興味のある方は読んでみてください。

[*12] 既存のコーパスを利用した分析に関して倫理的な問題が生じるとすれば，Web コーパスを利用した研究がその候補として挙げられるでしょう。Sketch Engine で利用可能な Web コーパスである TenTen コーパス (Jakubíček et al. 2013) などは大量のテキストデータを Web 上から取得し，コーパスとしてまとめたものです。多くの事例が利用可能となることは有り難い反面で，アダルトサイト等の一般的に好ましくないとされるソースの事例も多く含みます。そのような場で利用されることが多い表現の記述も重要であるとはいえ，その提示には細心の注意を払う必要があることには留意しましょう。

[*13] 例えば，Kambara et al. (2023) は Open Science Framework (OSF) に利用したアノテーションの結果だけでなく，その処理のためのコードを公開しています。このように公開されたデータは様々な研究者の資料として利用できるため，機会が許す限りにおいて積極的におこなうと良いでしょう。

[*14] 例として「生き物」という内包をもつ集合を考えてみましょう。この集合には，この世に存在する多くの存在物がその集合の外延として含まれることになります。ここで「生き物であり，かつ犬」という内包をもつ集合を考えると，「生き物」という内包をもつ集合よりも少ない外延を含む集合が得られます。さらに「生き物であり，かつ犬，かつ日本に生息している」という内包をもつ集合であれば更に少ない外延を含む集合が得られます。このように集合の内包の豊富さとその外延の数はトレードオフの関係にあり，検索条件の厳しさと検索結果の数の関係もおおむね同様の関係がみられる。

[*15] 最適な一般化の述べ方については，黒田 (2011a) を参照してください。

[*16] この「興味深さ」をどのように定式化するのか，という問題は非常に難しいため，本稿ではこれ以上の考察をおこないません。

[*17] ここで意図されている言明を記号論理学における記法を使うと $\{(P_1 \wedge P_2) \vee P_3\} \rightarrow Q$ となります。

[*18] 近年では，このような複雑な一般化を統計モデルをつかって得ることが多いです。統計的なモデリングの解説は難しいため，4. 節で紹介する文献を参照してください。

[*19] 分析の実行可能性，という観点の他にもコーパスの事例の提示には倫理的な配慮が必要になる場合があります。詳細は注釈*12 を参照してください。

[*20] これは (3) の分析でも用いたものです。

[*21] これは神原 (2021, Ch.4) で用いた CQL の一部に該当します。

[*22] 共同研究などでアノテーションをおこなう際には，その基準を議論しながら設定していくことが多いため，個人的には使わない機能です。

[*23] おそらく Sketch Engine 内では任意の規模の乱数の配列を管理することで，ランダム化によって同じ結果が得られるような工夫がなされていると思われます。

[*24] 個人的には Python は簡単な読み書きしかできませんが，心理言語学的な実験のプログラム作成や数値の処理にも便利なので一度学んでみるといいでしょう。

[*25] 類似する議論としては Gries (2023b) が参考になるでしょう。

[*26] 時折「数学や統計をどれだけ勉強する必要があるのか？」という質問を受けることが

あります。神原自身，数学がとてもできるという自負は全くありませんが，個人的な意見としては,「ある程度の勉強は常に必要」と答えています。しかし，コーパス研究に関して言えば，カテゴリー変数と呼ばれる変数の分析を学べばそれである程度のことはできるようになります。更に先に進みたいのであれば更に多くのことが必要になりますが，どの本を読むべきかということよりも，自分が分析する対象がどのような特性を持つのかの事前の理解がより重要になるでしょう。また，共同研究によって自分の不足分を補ってもらうということも大切になるでしょう。例えば，Kambara & Chika (2023) では Gries (2019, 2023a) の議論をもとに，複数の共起指標の統合を試みましたが，それを数学者の共同研究者を迎えることで Kambara et al. (2024) のような論文を書くことをおこないました。このように自分の限界を知り，誰かに助けてもらうことは恥ずかしいことではないので，悩んだら周りに（いなければ私でも構いません）相談する癖を付けることが良いでしょう。

References

Baayen, R. (2010). Demythologizing the word frequency effect: A discriminative learning perspective. *The Mental Lexicon*, *5*(3):436–461.

Cruse, A. D. (1986). *Lexical semantics.* Cambridge: Cambridge University Press.

Cruse, A. (2011). *Meaning in language: An introduction to semantics and pragmatics* (3rd edition). Oxford: Oxford University Press. （片岡宏仁（訳）(2012).『言語における意味: 意味論と語用論』東京: 東京電機大学出版局）

Fellbaum, C. & Baker, C. F. (2013). Comparing and harmonizing different verb classifications in light of a semantic annotation task. *Linguistics*, *51*(3), 707–727.

Fillmore, C. J. (1990). “Corpus linguistics” vs. “computer-aided armchair linguistics”. In Svartvik, J. (ed.), *Directions in corpus linguistics: Proceedings from a 1991 Nobel symposium on corpus linguistics* (pp. 35–66). Berlin: Mouton de Gruyter.

Glynn, D. & Fischer, K. (Eds.), (2010). *Quantitative methods in cognitive semantics: Corpus-driven approaches.* Berlin: Mouton de Gruyter.

Glynn, D. & Robinson, J. A. (Eds.), (2014). *Corpus methods for semantics: Quantitative studies in polysemy and synonymy.* Amsterdam: John Benjamins.

Gries, S. T. (2010). Corpus linguistics and theoretical linguistics: A love-hate relationship? Not necessarily... *International Journal of Corpus Linguistics*, *15*(3), 327–343.

Gries, S. T. (2017). *Quantitative corpus linguistics with R: A practical introduction* (2nd edition). London: Routledge.

Gries, S. T. (2019). 15 years of collostructions: Some long overdue additions/corrections (to/of actually all sorts of corpus-linguistics measures). *International Journal of Corpus Linguistics*, *24*(3), 385–412.

Gries, S. T. (2021). *Statistics for linguistics with R: A practical introduction* (3rd edition). Berlin: Mouton de Gruyter.

Gries, S. T. (2023). Overhauling collostructional analysis: Towards more descriptive simplicity and more explanatory adequacy. *Cognitive Semantics*, *9*(3), 351–386.

Gries, S. T. (2023). Quantitative corpus methods of cognitive semantics/linguistics. In Li, F. T. (ed.) *Handbook of cognitive semantics* (pp. 328–350). Leiden: Brill.

Gries, S. T. & Divjak, D. (2010). Quantitative approaches in usage-based cognitive semantics: Myths, erroneous assumptions, and a proposal. In Glynn, D. & Fischer, K., (Eds.), *Quantitative methods in cognitive semantics: Corpus-driven approaches* (pp. 333–353). Berlin: Mouton de Gruyter.

Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlý, P., & Suchomel, V. (2013). The TenTen Corpus Family. In *7th International Corpus Linguistics Conference (CL 2013)*, (pp. 125–127).

Kambara, K. & Chika, T. (2023). Toward a corpus-based identification of nominal relationality and uniqueness: A constructionist approach. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation* (pp. 47–73).

Kambara, K., Chika, T., & Takahashi, N. (2024). Conflating directional association measures: A case study on NP constructions. *Journal of Corpus-based Lexicology Studies*, *6*, 95–110.

Kambara, K., Nozawa, H., & Takahashi, T. (2023). Differentiating valence patterns: A quantitative analysis based on formal and semantic attributes. *Constructions*,

15(2), DOI: https://doi.org/10.24338/cons-571.

Kambara, K. & Yamanaka, T. (2023). Philosophy of data science for corpus linguistics: A pragmatistic point of view. *Annals of the Japan Association for Philosophy of Science*, *32*, 47–73.

Kilgarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovvář, V., Michelfeit, J., Rychlý, P., & Suchomel, V. (2014). The Sketch Engine: Ten years on. *Lexicography*, *1*(1), 7–36.

Kilgarriff, A., Husák, M., McAdam, K., Rundell, M., & Rychlỳ, P. (2008). GDEX: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX international congress* (pp. 425–432).

Kilgarriff, A., Rychlý, P., Smrž, P., & Tugwell, D. (2004). The Sketch Engine. In Williams, G. & Vessier, S. (Eds.), *Proceedings of XI EURALEX International Congress* (pp. 105–116).

Levshina, N. (2015). *How to do linguistics with R: Data exploration and statistical analysis.* Amsterdam: John Benjamins.

McEnery, T. & Hardie, A. (2012). *Corpus linguistics: Method, theory and practice.* Cambridge: Cambridge University Press. （石川慎一郎（訳）(2014).『概説コーパス言語学: 手法・理論・実践』東京: ひつじ書房）

McEnery, T. & Brezina, V. (2012). *Fundamental principles of corpus linguistics.* Cambridge: Cambridge University Press.

Murphy, M. L. (2003). *Semantic relations and the lexicon: Antonymy, synonymy, and other paradigm.* Cambridge: Cambridge University Press.

Murphy, M. L. (2010). *Lexical meaning.* Cambridge: Cambridge University Press.

Pustejovsky, J. & Stubbs, A. (2013). *Natural language annotation for machine learning.* California: O'Reilly Media, Inc.

R Core Team (2022). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria.

Stefanowitsch, A. (2020). *Corpus linguistics: A guide to the methodology.* Berlin: Language Science Press.

Stefanowitsch, A. & Gries, S. T. (2003). Collostructions: Investigating the interaction

of words and constructions. *International Journal of Corpus Linguistics*, *8*(2), 209–243.

Wickham, H. & Grolemund, G. (2016). *R for data science: Import, tidy, trans- form, visualize, and model data.* California: O'Reilly Media, Inc.

Winter, B. (2019). *Statistics for linguists: An introduction using R.* London: Routledge.

淺尾仁彦・李在鎬. (2013). **言語研究のためのプログラミング入門***: Python* **を活用したテキスト処理**. 東京: 開拓社.

石川慎一郎. (2021). **ベーシックコーパス言語学**（第二版）. 東京: ひつじ書房.

石川慎一郎・前田忠彦・山崎誠（編） (2010). **言語研究のための統計入門**. 東京: くろしお出版.

神原一帆. (2017). 日本語における「ノ構文」の振る舞い: 少納言コーパスを用いた属性パターンに基づく調査と「進撃の巨人」. **日本認知言語学会論文集**, *17*, 390–401.

神原一帆. (2021). **フレーム意味論にもとづく名詞の意味分析**. 博士論文, 京都大学大学院人間・環境学研究科, 京都.

神原一帆・春日悠生・田中悠介. (2019). 発表の構成について: 分かりやすい発表をするための留意点. （最終更新: 2022 年 8 月 14 日）.

神原一帆・野澤元・高橋武志. (2022). 事態認知の焦点化パターンに対するコーパス基盤アプローチ: 動詞 STAB を例とした意味役割の分析. *Journal of Corpus-based Lexicology Studies*, *4*, 14–28.

神原一帆・野澤元・高橋武志. (2024). 事態の焦点化と構文選択に関する量的分析: 動詞 *stab* の事例分析をとおして. **日本認知言語学会論文集**, *24*, xxx–xxx.

久保拓弥. (2012). **データ解析のための統計モデリング入門***:* **一般化線形モデル・階層ベイズモデル・***MCMC.* 東京: 岩波書店.

黒田航. (2003). 認知形態論. 吉村公宏（編） **認知音韻・形態論** (pp. 79–154). 東京: 大修館書店.

黒田航. (2011a). 一般化の述べ方について: いかに "過小" 般化と "過大" 般化を避けて最適な一般化を達成するか. （最終更新: 2011 年 9 月 14 日）.

黒田航. (2011b). 自作例を使った研究の基礎. 辻幸夫（監修）中本敬子・李在鎬（編）, **認知言語学の方法***:* **内省・コーパス・実験** (pp. 29–63). 東京: ひつじ書房.

黒田航. (2012). 言語学と言語処理の共生は可能か?: 統計基盤の言語処理の限界はどこにあるか? それは知識基盤の言語処理で克服できるか? **人工知能学会誌**, *27*(3), 326–332.

黒田航. (2017). Sketch Engine を使う; Regular Expressions を学ぶ. （スケッチエンジン・正規表現講習会　発表資料）.

竹内理・水本篤 (2023). **外国語教育研究ハンドブック*: 研究手法のより良い理解のために***（増補版）. 東京: 松柏社.

田野村忠温. (2012). ***Ruby* によるテキストデータ処理**. 東京: 明治書院.

長谷部陽一郎. (2020). English-corpora.org を用いた言語データの採取. （最終更新: 2020 年 5 月 29 日）.

福田純也・矢野雅貴・田村 祐 (編). (2023). **第二言語研究の思考法*: 認知システムの研究には何が必要か*** 東京: くろしお出版.

松村優哉・湯谷啓明・紀ノ定保礼・前田和寛. (2021). ***R* ユーザのための *RStudio* [実践] 入門*: Tidyverse* によるモダンな分析フローの世界** (第二版). 東京: 技術評論社.

水本篤・竹内理. (2010). 効果量と検定力分析入門: 統計的検定を正しく使うために. **より良い外国語教育研究のための方法** (pp. 47–73).

守田貴弘. (2013). 意味的分類の科学的妥当性. **言語研究**, *144*, 29–53.

付録 A　練習問題

“⋆” をつけたものは難易度が高いです。これらの問題については本稿で扱った内容以外のことを考慮する必要がありますので，現時点でできなくても落ち込まないでください。用意した問題は 40 問近くありますが，特に解答は用意していません。回答に際してサポートが必要な場合は `kazy0324@pep-rg.jp` までご連絡いただければ個別にフィードバックさせていただきます。

A.1　正規表現

A.1-1. 指定するファイルの中から，不定冠詞 *a* に後続する語の総数をテキストエディターの検索機能を使って調べなさい (e.g., “a dog”, “a beautiful”, ...)。

A.1-2. 指定するファイルの中から，不定冠詞 *a*，または定冠詞 *the* に後続する語の総数

をテキストエディターの検索機能を使って調べなさい (e.g., “a dog”, “the tall”, ...)。

A.1-3. ⋆`dogs.txt` の中から，単数形の dog で出現する回数をテキストエディターの検索機能を使って調べなさい。

♣ `dogs.txt` は無作為に *dog* と *dogs* という表現を 10,000 件ランダムに連ねたものです。複数形を検索し，その差分を引き算によって求めることでも同じ結果が得られますが，折角なので正規表現を使いましょう。

A.1-4. ⋆`hyphens.txt` の中から，ハイフン “-” が 6 回連続で用いられている回数をテキストエディターの検索機能を使って調べなさい。

♣ `hyphens.txt` は “-” が最大 50 回連続する行を 10,000 件連ねたものです。間違っても手動で数えようなどとは思わないこと。

A.1-5. ⋆ 指定するファイルの中から，五文字からなる英単語の総数をテキストエディターの検索機能を使って調べなさい (e.g., “whose”, “blond”)。

A.1-6. ⋆ 指定するファイルの中から，不定冠詞 *a*，または定冠詞 *the* に後続する語の総数五文字からなる英単語の総数をテキストエディターの検索機能を使って調べなさい (e.g., “whose”, “blond”)。

A.1-7. ⋆(1) で挙げた正規表現には，英語における名詞の全てを列挙する必要がある，ということに加えて決定的な問題がもう一つあります。K さんのタスクとの関係からこれがどのようなものなのかを論じなさい。

A.2 Sketch Engine

1.2.1 Word Sketch

1.2.1-1. BNC (tagged by CLAWS) のデータを利用し，*horse* の Word Sketch を取得し，その結果の一部を可視化しなさい。

1.2.1-2. BNC (tagged by CLAWS) のデータを利用し，*horse* と共起する語のリスト全てを csv としてダウンロードしなさい。

1.2.1-3. BNC (tagged by CLAWS) のデータを利用し，*horse* を目的語としてとる事例の全てを csv としてダウンロードしなさい。

1.2.1-4. ⋆BNC (tagged by CLAWS) のデータを利用し，動詞として用いられる *dog* の Word Sketch を取得し，その結果の一部を可視化しなさい。

1.2.1-5. ⋆BNC (tagged by CLAWS) のデータを利用し，動詞として用いられる *dog* と共起する語のリスト全てを csv としてダウンロードしなさい。

1.2.2 Word Sketch Difference

1.2.2-1. BNC (tagged by CLAWS) のデータを利用し，*begin* と *start* の Word Sketch を取得し，その結果の一部を可視化しなさい。

1.2.2-2. BNC (tagged by CLAWS) のデータを利用し，*end* と *finish* の Word Sketch を取得し，その結果の一部を可視化しなさい。

1.2.2-3. BNC (tagged by CLAWS) のデータを利用し, *stallion* と *mare* の Word Sketch を取得し，その結果の一部を可視化しなさい。

1.2.3 Thesaurus

1.2.3-1. BNC (tagged by CLAWS) のデータを利用し，*run* という動詞のレマの thesaurus のリストを.csv 形式でダウンロードしなさい。

1.2.3-2. BNC (tagged by CLAWS) のデータを利用し，*run* という名詞のレマの thesaurus のリストを.csv 形式でダウンロードしなさい。

1.2.3-3. BNC (tagged by CLAWS) のデータを利用し，*run* という動詞のレマの thesaurus のリストを可視化したものをダウンロードしなさい。

- ♣ Number of Collocates を 20 とし，Bubble Chart と Word Cloud の両方をダウンロードすること。

1.2.3-4. BNC (tagged by CLAWS) のデータを利用し，*run* という名詞のレマの thesaurus のリストを.csv 形式でダウンロードしなさい。

- ♣ Number of Collocates を 20 とし，Bubble Chart と Word Cloud の両方をダウンロードすること。

1.2.4 Concordance

1.2.4-1. BNC (tagged by CLAWS) のデータから CQL を利用し，レマ *read* のコンコーダンスを取得し，その結果を.csv 形式でダウンロードしなさい。

1.2.4-2. BNC (tagged by CLAWS) のデータから CQL を利用し，動詞のレマ *read* のコンコーダンスを取得し，その結果を.csv 形式でダウンロードしなさい。

1.2.4-3. ⋆BNC (tagged by CLAWS) から CQL を利用し，動詞 *read* の直後に定冠詞がくる事例のコンコーダンスを取得し，その結果を.csv 形式でダウンロードしなさい。

1.2.4-4. ⋆BNC (tagged by CLAWS) から CQL を利用し，定冠詞の直後に名詞が後続する事例のコンコーダンスを取得し，その結果を.csv 形式でダウンロードしなさい。

1.2.4-5. ⋆BNC (tagged by CLAWS) のデータを利用し，動詞 *read* の直後に定冠詞がくる事例のコンコーダンスを取得し，その結果を.csv 形式でダウンロードしなさい。

1.2.4-6. ⋆BNC (tagged by CLAWS) のデータを利用し，動詞 *stab* の目的語のコンコーダンスを取得し，その結果を csv 形式でダウンロードしなさい。

♣ 神原ほか (2022, 2024) では動詞 *stab* を分析していますが，動詞ではない事例を省くためにこの CQL をもちいてデータを抽出しました。

1.2.5 Wordlist

1.2.5-1. re から始まる動詞のリストを抽出し，その結果を.csv 形式でダウンロードし，この結果が *retake* のような事例ばかりでないことを確認しなさい。

1.2.5-2. tion でおわる名詞のリストを抽出し，その結果を.csv 形式でダウンロードしなさい。

1.2.5-3. ly でおわる形容詞のリストを抽出し，その結果を.csv 形式でダウンロードしなさい。

1.2.5-4. exam という形式を含むレマのリストを抽出し，その結果を.csv 形式でダウン

ロードしなさい。

1.2.5-5. re から始まる動詞のリストを抽出し，その結果を.csv 形式でダウンロードし，この結果が *retake* のような事例ばかりでないことを確認しなさい。

1.2.6 N-grams

1.2.6-1. BNC (tagged by CLAWS) のデータを利用し，word 単位で，最低頻度を 10 とした 3-gram のリストを.csv 形式でダウンロードしなさい。

1.2.6-2. BNC (tagged by CLAWS) のデータを利用し，lemma 単位で，最低頻度を 10 とした 3-gram のリストを.csv 形式でダウンロードしなさい。

1.2.6-3. ⋆BNC (tagged by CLAWS) のデータを利用し，word 単位で，{*I, you, we, they, it*}を除く，最低頻度を 10 とした 4-gram のリストを.csv 形式でダウンロードしなさい。

1.2.6-4. ⋆BNC (tagged by CLAWS) のデータを利用し，lemma 単位で，{*I, you, we, they, it*}を除く，最低頻度を 10 とした 4-gram のリストを.csv 形式でダウンロードしなさい。

1.2.7 Keywords

1.2.7-1. BNC (tagged by CLAWS) における話し言葉と書き言葉の Single-words を比較し，その絶対頻度と相対頻度を含む結果ををダウンロードしなさい。

1.2.7-2. BNC (tagged by CLAWS) における話し言葉と書き言葉の N-grams を比較し，その絶対頻度と相対頻度を含む結果をダウンロードしなさい。

1.2.8 応用編

ここでは実際の研究等で利用する可能性のあるタスクの一部を挙げています。ここでみるものは複数の機能を利用する必要があるため，一筋縄ではいかないと思われますが，是非挑戦してみてください。

1.2.8-1. ⋆CQL を利用し，BNC (tagged by CLAWS) のデータから *be made from* と *be made of* の両方を含むコンコーダンスを.csv 形式でダウンロードしなさい。

1.2.8-2. ⋆BNC (tagged by CLAWS) に含まれる名詞のレマ (e.g., *test*) のリストを入手し，その結果をもとに動詞形 (e.g., “tested”) のレマの (i) コンコーダンスと (ii) 頻度表の両方を.csv 形式でダウンロードしなさい。

1.2.8-3. ⋆BNC (tagged by CLAWS) に含まれる動詞のレマ (e.g., *test*) のリストを入手し，その結果をもとに “re” からはじまる動詞形 (e.g., “retested”) のレマの (i) コンコーダンスと (ii) 頻度表の両方を.csv 形式でダウンロードしなさい。

1.2.8-4. ⋆2.2 節で論じたような検索条件の厳しさに応じて頻度が影響を受けるさまを任意の名詞五語を例に確認し，その結果を別のアプリケーションを用いて可視化し，図 2 と類似の結果が得られることを確認しなさい。

♣ 神原はこの結果の可視化に R Core Team (2022) を利用しましたが，適宜 Excel 等の表計算アプリケーションを利用して問題ありません。