DEIM Forum 2021 F11-5

技術ブログにおける単語出現の順序構造を用いた全容把握型検索結果の生成

波木井　征†　　北山　大輔†

† 工学院大学情報学部システム数理学科　〒 163–8677 東京都新宿区西新宿 1 丁目 24-2
E-mail: †j317224@ns.kogakuin.ac.jp, ††kitayama@cc.kogakuin.ac.jp

あらまし　ある検索トピックにおいて，事前知識のないユーザが検索結果に含まれるトピックの全容を把握することは困難である．また，あるトピックに対して，他のトピックについて知りたい場合やそのトピックについて深く知りたい場合に，既存の検索エンジンでは，周辺トピックがわからず，適切な検索クエリを入力することができない．そこでトピックに対する全容を把握するための検索結果の提示手法を提案する．具体的には，検索結果集合から，前提知識関係にある単語で木構造を構築し，その構造に基づき検索結果を抽出し，ユーザに提示する．前提知識関係を利用した検索結果の提示により，ユーザはそのトピックについて深く知りたい場合に木構造をたどる．これにより，ユーザはトピックに対する全容を把握することができる．本研究では，トピックに対する全容把握をするための検索結果の提示手法を検討する．評価実験を行った結果，詳細関係 (木の深さ方向) の適合率は 70.5%であり，順序関係 (木の幅方向) の適合率は 65.8%であったが，詳細関係の正解に限定したときの順序の適合率は 86.0%であった．

キーワード　全容把握，木構造

1　はじめに

近年，検索技術の向上によりユーザが求める情報が容易に取得できるようになっている．しかし，ユーザが検索したいトピックに対して全く知識が無い場合には，検索が非常に難しいものになったり，新しい領域を学ぶ際にそのことについて検索する場合，どのようなキーワードを入力すればよいのか分からなくなる問題がある．また，複数ページに渡って，各ページ内から重要だと考えられるキーワードを見つけ出すことは容易ではない．例えば，全くの初心者が MySQL について調べたい場合，通常の Web 検索結果で単純に MySQL をキーワード検索すると，MySQL についての概要や機能の一部について説明されるものが生成され，どのような順番で閲覧すればよいか分からず，なおかつ全体を把握するにはさらにキーワードを付け足して検索し続けなければならない．

我々は，容易にトピックの全容を把握することができる検索結果を提示することで，この問題が解決可能であると考えた．出力されるイメージとしては,「MySQL」という検索キーワードが入力された場合には「MySQL 環境構築」というタイトルがまず表示され，その子要素として「データベース操作の基本」や「MySQL のダウンロード&インストール方法」というタイトルが表示される．このように構造化することで，本の目次のように，同階層の見出しによりそのサブトピックを網羅的に把握でき，興味のあるトピックについては，子要素をたどることで詳細に知ることが可能となることを期待する．

一方で，近年，技術ブログの記事が充実してきている．技術ブログとは Qiita[1]や，Developers.IO[2]に代表される，プログラミング等の知識に関するブログである．このような技術ブログにはさまざまな用語や知識について記事単位では体系的に書かれていることが多い．このような技術ブログの構造を利用することで，例えば「MySQL」の下に「データ」や「SELECT 文」が来るような親子関係を得ることが可能であると考える．これを利用することで，我々は，検索結果集合から，前提知識関係にある単語で木構造を構築し，その構造に基づき検索結果を抽出し，ユーザにトピックの全容を提示する手法を提案する．具体的には，技術ブログの見出し単語の大見出し，小見出しのような従属関係から用語の親子関係を抽出し，その単語の親子関係に従って技術ブログ中で該当するタイトルを抽出する．このことにより，章タイトル，節タイトルのような構造を作り，検索結果を構造化して提示する．

想定する出力としては図 1 となる．図のように，同一階層の出力結果を下へ進んでいくと，検索キーワードについての知識が増えていき，下の階層へ進むとそのトピックについてのより深い知識を得ることのできるようなものである．

この研究の貢献は，以下のとおりである．

- 技術ブログから，用語の親子関係を自動的に構造化し，ある種のオントロジを構築する
- 検索結果の構造化のアプローチとして，用語の親子関係に基づく構造化の効果と性質を示す
- 閲覧するべき順序も含めた構造化に挑戦する

以下に本論文の構成について記す．2 節では関連研究について述べる．3 節では提案手法について述べる．4 節では実験につい

1：https://qiita.com
2：https://dev.classmethod.jp

図 1　想定する出力

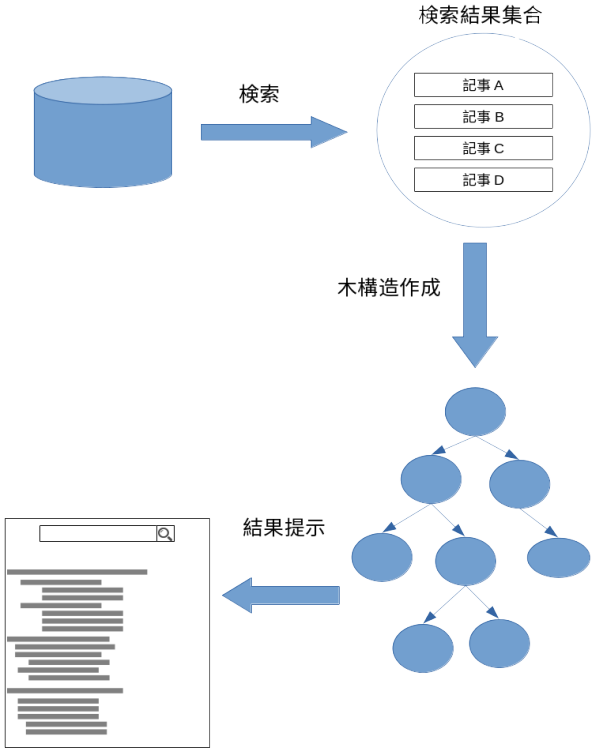

図 2　システム概要図

て述べる．最後に，5 節でまとめと今後の課題について述べる．

2　関 連 研 究

湯本ら [1] は，知りたい情報について知識がない状態で検索を行う場合，ユーザは検索結果を閲覧しても，必要なすべての情報を得られたのかどうかを判断することができない．また，現在のページごとの検索では知りたい事柄について 1 ページで十分な情報を持ったページが存在するとは限らない．そのため適切なページが 1 ページでないという考えから全容検索を提案している．全容検索は，通常のページごとの検索結果から，あるキーワードについて話題の広さと深さが両立したページ集合を生成し，それをランキングするものである．

池田ら [2] は Twitter の反応を利用しニュースの全体像の理解支援を行うための可視化手法を提案している．Twitter で投稿されたニュースに対する反応としてリプライ，引用リツイートを用い，ニュース自体の特徴語と反応の特徴語を抽出し，抽出した特徴語を利用してニュースや反応特徴および他のニュースとの関連性を可視化している．

間瀬ら [3] は，Web ページ集合からトピックマップを半自動で抽出する手法を提案している．従来のネットワーク構造に着目したクラスタリング手法に，Web ページのコンテンツによる類似度，Web サイトのディレクトリ構造と Web ページ間のリンクを利用した重み付けを導入し，Web ページ間のリンクに内包されるトピックの関係を考慮しページ間のリンクの意味を推定し，トピックだけでなくトピックの関連も併せて抽出する．

福田ら [4] は，LDA (Latent Dirichlet Allocation) によるトピック分析を用いることで，検索クエリに関連する論文を網羅的に発見するためのランキング手法を提案している．これは，トピック分析結果を介して，抄録および検索クエリ内の各語をトピックに置き換えたトピックレベルでのブーリアン検索に基づいている．そして LDA に与えるパラメータの設定が異なる複数のトピック分析を行い，それぞれのトピック分析結果を用いてブーリアン検索を実行し，結果を統合し論文をランク付けしている．

村山ら [5] は，Web ページをクエリとしたキーワードレスの研究情報検索を提案している．研究活動を始めたばかりの初心者にとって研究情報を適切に探し出すことは簡単ではなく，情報検索のための適切なキーワードを用意することができないことから，ブラウジング中の Web ページに基づき関連する研究情報を検索するブラウザ拡張アプリケーションを開発している．Web ページ中のテキストを利用し単語分散表現の足し合せにより，Web ページや論文，研究者のすべてをベクトルで表現しベクトルの類似度により順位付けすることで実現している．

南川ら [6] [7]Wikipedia から人手で作成したルールに基づき，技術とその技術によって実現できる技術・サービスのペアを抽出している．抽出したペアにはノイズが多い為，機械学習を用いて各項が技術，サービス名かどうかのフィルタリングを行っている．

阪田ら [8] は，検索キーワードの出現する文章を起点とした周辺文章となる位置関係に基づき文章を抽出し，検索キーワードに対するユーザの知識レベルに応じた記事要約手法を提案している．検索キーワードを含む文章とその文章の前後に出現する文章との位置関係は，検索キーワードを説明する内容の詳細と関連するという仮定に基づき，文書間の距離に基づいて要約を複数生成する．

倉門ら [9] は，Wikipedia に基づいたリンク情報やカテゴリ構造を解析することで，検索クエリの関連語を抽出し，検索結果の適切なリランキング手法を提案している．Wikipedia から利用できる素性として，inlink，outlink，リンク共起，カテゴリの 4 つがあると考え，それぞれを利用しリランキングを行っ

ている．

梅本ら [10] は，推薦クエリのみ閲覧情報を可視化インターフェースを提案している．探索型の検索から発展して，網羅性指向のタスクに対するアプローチを提案し，未閲覧情報量を重要度，適合性，新規性の観点からスコア化している．

隅田ら [11] は，Wikipedia の記事構造を知識源として量の上位下位関係を自動獲得する手法を提案している．Wikipedia の記事構造に含まれる節や箇条書きの見出しから，大量の上位下位関係候補を抽出し，機械学習を用いてフィルタリングすることで高精度の上位下位関係を獲得する手法を提案し，2007 年 3 月の日本語版 Wikipedia から精度 90%の精度を実現した

これらの関連研究は，検索キーワードに対して，基本的に全容を把握することを目的としているが，どのような順番でユーザが結果を閲覧すれば良いかを考慮していない．本研究では，全容を把握するという目的は共通しているが，どのような順序で閲覧すればユーザが全容を把握しやすいかという点で異なる．

3 提案手法

3.1 概　　要

全容把握型検索に必要な条件としては，まずはじめに検索キーワードについての概要や最初に取り組むべき内容などが閲覧でき，その後さらに詳しい内容を得たい場合に次にどこをみればよいのかが分かることである．提案手法の流れとしては大きく分けて以下となる．

（1） データの取得
（2） 検索結果記事集合の取得
（3） 検索結果記事集合から単語の木構造の作成
（4） 木構造に基づいてタイトルを抽出

図 2 がシステム概要図となる．まず，ユーザは全容を知りたい検索キーワードをシステムに入力する．システムはその結果，検索結果集合を得る．その検索結果集合から，従属関係のある単語の木構造を作成し，その木構造の構造を用いて結果を提示する．

3.2 データの取得

まず，研究に使用する技術ブログのデータを取得する．データセットとしては Qiita[3]の記事を使う．Qiita とはプログラマの技術情報共有サービスであり，プログラミング等のノウハウやメモを記録したり，公開することができるサービスである．データは Qiita が提供している API の QiitaAPIv2 を用いて，2014/6/7 から 2020/8/3 までの記事 (約 52 万件) を取得した．取得した内容は，タイトル，記事の本文，URL，タグである．記事はマークダウンによって書かれ，見出しは h1，h2 などのタグで表現される．このとき，これらのタグ間には明示的なタグの親子関係は存在しないが，h1 タグのあとの h2 タグは，意味的に従属するといった関係が存在することが多い．これを利用して単語の親子関係を抽出できると考える．記事本文は作業や時系列に沿って書かれることが多く，単語の出現順序として，先に出現する単語は，後に出現する単語の前提作業や前提知識となっていることが多いと考える．

表 1　単語ペアの例

前単語	後単語	ペア出現回数
MySQL	環境	675
MySQL	設定	627
MySQL	参考	596
MySQL	インストール	508
MySQL	作成	497
MySQL	確認	461
MySQL	テーブル	324
MySQL	起動	304
MySQL	データベース	273
MySQL	手順	272
MySQL	エラー	234
インストール	MySQL	236
MySQL	前提	214
MySQL	コマンド	158
環境	インストール	173
構築	環境	189
構築	MySQL	182
接続	MySQL	157
docker	環境	130
環境	構築	120

3.3 前提知識関係のある木構造の作成

まず，ユーザが入力した検索キーワードに対し，検索キーワードをタイトルまたはタグに含む記事を得る．記事本文中から，タイトルおよび見出しを抽出し，形態素解析器 MeCab [12] を用いて，名詞のみを抽出する．本稿では，辞書は ipadic-neologd[4] を用いた．

3.3.1 従属関係の単語ペアの作成

検索結果集合から，従属関係のある単語ペアを作成する．記事中の見出し間には，従属関係があると考える．具体的には，タイトル単語と h1 タグの単語，h1 タグの単語と h2 タグの単語のような関係である．この関係性を利用して，上位に概要的な単語が位置し，ノードをたどるにつれてより詳細な単語がになるような木構造を作成する．タイトル文章を形態素解析器を用いて名詞を抽出し，従属関係にある名詞でペアを作成する．このとき，URL や記号などをストップワードとして削除する．次に，ノイズの除去をする． ペアの出現回数が上位 N 件であるものは対象から除外し，それ以外のペアのうち，単語の出現回数が閾値以下である単語を含むペアを除去する．表 1 は，検索キーワードを “MySQL”として単語ペアの作成例である．

3.3.2 木構造の作成

作成した単語ペアを使って木構造を作成する．前に出現した単語をそのペアの親とし，ペア数をカウントする．この値をそのペアの親単語の支持度とする．このとき，構成単語は同じで順序だけが異なる単語ペアが出現するので，支持度が小さい単語ペアを除去する．このまま木構造を作成すると，ノードを辿

3：https://qiita.com

4：https://github.com/neologd/mecab-ipadic-neologd

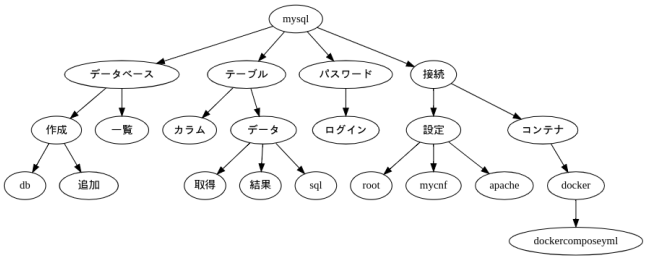

図 3　木構造の作成例

ると前後関係として不適切な単語が出現する可能性がある．そこで，出現単語をクラスタリングする．そして，クラスタ中の単語でのみ構成される単語ペアで木構造を作成する．木のルートは，単語ペアの親単語に検索キーワードがあればその単語，なければ親として出現する回数が最も多い単語とする．その後，クラスタ中の単語のみで構成した木構造を合成する．木構造のルートは検索キーワードとし，その子ノードに先に作成した木構造を結合する．木構造の作成方法は，ルート単語を親とする単語ペアを抽出し，その後，ペアとなっている単語のみを親子関係にし，一度出現した単語は再度使用しない，ルートノード直下のノードで深さが 1 の場合は削除する，というルールで木構造を作成する．

この時，木構造のノードに表示優先度の属性を付与する．木構造の兄弟ノードの表示に関して，どれを先に表示するかを制御するため，検索結果集合から，単語の表示優先度を作成する．ある単語に関して，記事中の同一サイズの見出し中の，より前半部分に出現する単語は，結果表示時により上位に提示すべきと考える．このことから，対象単語が出現する見出しと同一サイズの見出し数 M とし，対象単語が出現する見出しの出現位置（同一サイズの見出しにおける出現順）を i とした場合，式 (1) の計算を行う．その後，それぞれの記事で計算したあとに平均する．

$$\frac{M-i+1}{M} \tag{1}$$

各単語について，表示優先度を与えた後，この値が大きい単語をより上位に表示することとする．図 2 は検索キーワードを “MySQL” として木構造の作成例である．

作成した木構造は，兄弟間で似た単語や表記揺れのある単語が出現する可能性がある．具体的には,「ユーザ」と「ユーザー」といった単語である．そこで，全記事の本文とタイトルの文章で学習した単語ベクトルを使い，類似度が閾値以上となる単語のノードをマージする．マージされたノードの単語としては，表示優先度が高い単語を採用する．

3.4　木構造に基づいたタイトルの抽出と結果の表示

作成した木構造の構造に基づいてタイトル文を抽出する．作成した木構造から，対象のノードの単語を含み，親ノードの単語が含まれるタイトルを抽出する．祖先ノードを使用しない理由は，ノードの深さが大きくなるにつれて使用する単語が多くなってしまい，簡潔なタイトルが抽出できないと考えたためである．子ノードの単語が含まれるタイトル文や，子ノードに関

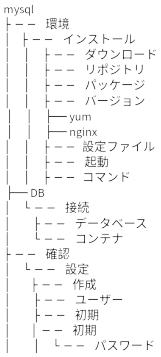

図 4　MySQL の検索結果から作成した単語の木構造

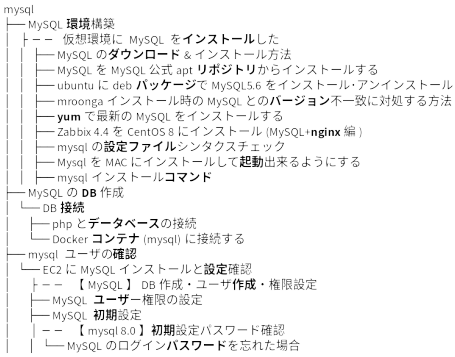

図 5　MySQL の検索結果から作成した構造化した検索結果

係ない単語が含まれるタイトル文は好ましくないことから，抽出するタイトル文は短く簡潔であり子ノードに出現する単語がなるべく含まれていないものを優先して選ぶ．決定方法は，親ノードのテキストと自身のノードのテキストを含むタイトル文書集合のから，もっとも文字列の長さが短いものとした．

抽出したタイトル文を検索結果として提示する．提示の仕方は，本の目次のように提示する．子ノードのタイトル文は親ノードのタイトル文の 1 段落下に位置する．これにより，より深い結果を得たい場合その段落を追うことで必要な情報を得ることが可能となる．同じ階層での表示の順番は，その階層の表示優先度の値が高い順に並べる．

3.5　実　行　例

ここでは，3 節で述べた提案手法の実行例を示す．図 4 に単語で並べた実行例，図 5 にタイトル文で並べた実行例となる．両方とも，検索キーワードは “MySQL”，最小ペア数が 25，クラスタリング手法が Kmeans，クラスタ数が 10 となっている．

出力例を見ると，MySQL の環境構築，DB 作成と続き，DB 作成の階層を辿るとデータベースの接続などについて出力されている．

4　実　　験

提案手法で作成した検索結果が，詳細関係かつ前後関係として妥当であるかを評価する実験を行った．

表 2 実験結果

種類	適合率 (%)
詳細関係	70.5
出現順序	65.8
詳細関係の正解に限定したときの出現順序	86.0

表 3 詳細関係実験結果

親タイトル	子タイトル	適合
MySQL 環境構築	仮想環境に MySQL をインストールした	正
MySQL 環境構築	docker で Laravel 環境構築	負
MySQL 環境構築	vagrant 環境構築	負

表 4 順序関係実験結果

親タイトル	子タイトル (システム出力)	子タイトル (正解データ)	適合率 (%)
【MySQL】DB 作成・ユーザ作成・権限設定	MYSQL ユーザー作成めも，フォーム作成のために作った 5 つのファイル	MYSQL ユーザー作成めも，フォーム作成のために作った 5 つのファイル	50

4.1 実験方法

実験は 2 種類行う．1 つ目は詳細関係について行う．出力の階層が深くなればなるほどより詳しい，または親のノードの情報を前提とした関係となっているかについて調査する．被験者には，あるノードのタイトル文を提示後，その子ノードのタイトル文を提示し，そのタイトル文で詳細関係にあると考えられるものを選択してもらう．その後，選択された割合を正解率として算出する．2 つ目はタイトル文の出現順序について行う．出力結果で同一階層での文章集合中で，より前半に並んでいるべき，またはより前提知識が必要と無い文章が前半に出現しているかについて調査する．被験者には，あるノードのタイトル文を提示し，その文に続けて読むと効果的であると考えられる文をその子ノードのタイトル文から選択してもらう．その後，選択されているものの上位のものと実際に出力されたものの上位を比較し，重複している割合を正解率として算出する．

4.2 実験結果と考察

両方の実験における正解率を表 2 に示す．正解率はそれぞれ 6 割程度となり，改善の余地がある結果となった．詳細関係についての結果の一部を表 3，出現順序についての一部を結果を表 4 に示す．

適していない出力を抽出すると，詳細関係については「MySQL 環境構築」 →「*docker* で *Laravel* 環境構築」や ,「$MySQL$ 環境構築」→「*vagrant* 環境構築」，出現順序については,「【$MySQL$】DB 作成・ユーザ作成・権限設定」→「フォーム作成のために作った 5 つのファイル」となっており，実験で良いと判断されているものと比べてタイトル文が相応しくないことから，タイトル文の抽出方法に課題があると考える．これは，タイトル文抽出時に，タグに検索キーワードを含むものを参考に抽出したが，タグは複数つけることができるので，検索キーワードがメインのトピックでない記事が抽出されてしまったからだと考察する．また詳細関係の正解に限定したときの順序の適合率が 86.0%と大幅に上がったことから，詳細関係であると判断されているものは出現順序にも影響していると考えられ，詳細関係が改善されると，出現順序も改善されると考える．

5 まとめと今後の課題

ある検索トピックにおいて，事前知識のないユーザが検索結果に含まれるトピックの全容を把握することは困難であるという考えから，技術ブログの記事の見出し間に用語の親子関係があると考え，そこから単語の木構造を作成し，それを元に全容を把握することのできる検索結果生成の提案を行った．結果としては，木構造で用語の親子関係を抽出することはできたが，タイトル文の抽出方法に課題が残る結果となった．今後の課題としては，タイトル文抽出方法の改善と単語ペア数の閾値の検討がある．

謝　　辞

本研究の一部は，2020 年度科研費基盤研究 (C)(課題番号：18K11551) によるものです．ここに記して謝意を表すものとします．

文　　献

[1] 湯本高行, 田中克己. Web ページ集合を解とする全容検索. 情報処理学会論文誌データベース（TOD）, Vol. 48, No. SIG11(TOD34), pp. 83–92, jun 2007.
[2] 池田将, 牛尼剛聡. Twitter の反応を用いたニュース全体像の理解支援のための可視化手法. 研究報告情報基礎とアクセス技術（IFAT）, No. 5, pp. 1–6, sep 2019.
[3] 間瀬心博, 山田誠二, 新田克己. Web ページ集合からの web ページのコンテンツと構造を用いたクラスタリングによるトピックマップの抽出. 情報処理学会研究報告知能と複雑系, Vol. 2007, No. 67(2007-ICS-148), pp. 53–60, jul 2007.
[4] 福田悟志, 冨浦洋一. 網羅性を重視した学術論文に対する検索手法. 研究報告情報基礎とアクセス技術（IFAT）, Vol. 2020, No. 2, pp. 1–6, jul 2020.
[5] 村山貴志, 河野翔太, 近藤佑亮, 中林雄一, 野々村一歩, 入江英嗣, 坂井修一. Web ページをクエリとしたキーワードレスの研究情報検索. 情報処理学会研究報告 (Web), Vol. 2020, No. 7, pp. 1–6, aug 2020.
[6] 南川大樹, 杉本徹. Wikipedia からの技術やサービス間の関係抽出. 第 80 回全国大会講演論文集, 第 2018 巻, pp. 299–300, mar 2018.
[7] 南川大樹, 杉本徹. Wikipedia からの技術やサービス間の関係抽出に用いるフィルタリングの改良. 第 82 回全国大会講演論文集, 第 2020 巻, pp. 493–494, feb 2020.
[8] 阪田晴香, Siriaraya Panote, 王元元, 河合由起子. 文章の相対位置関係に基づくユーザの知識レベルに応じた記事要約の提案. 第 81 回全国大会講演論文集, 第 2019 巻, pp. 437–438, feb 2019.
[9] 倉門浩二, 大石哲也, 長谷川隆三, 藤田博, 越村三幸. Wikipedia のリンク共起とカテゴリに基づくリランキング手法. 研究報告情報基礎とアクセス技術, No. 12, pp. 1–8, jul 2010.
[10] 梅本和俊, 山本岳洋, 田中克己. 網羅性指向タスクにおける未閲覧情報量の提示. 人工知能学会論文誌, Vol. 32, No. 1, pp. 1–12, 2017.
[11] 隅田飛鳥, 吉永直樹, 鳥澤健太郎. Wikipedia の記事構造からの上位下位関係抽出. 自然言語処理, Vol. 16, No. 3, pp. 3–24, 2009.

[12] 工藤拓, 山本薫, 松本裕治. Conditional random fields を用いた日本語形態素解析. 情報処理学会研究報告. NL, 自然言語処理研究会報告, Vol. 161, pp. 89–96, may 2004.