閲覧履歴と検索結果に対するWikipediaを用いた補間トピックの抽出と評価

波木井　征†　　北山　大輔†

† 工学院大学大学院工学研究科情報学専攻　〒 163–8677 東京都新宿区西新宿 1–24–2
E-mail: †em21018@ns.kogakuin.ac.jp, ††kitayama@cc.kogakuin.ac.jp

あらまし　未知のトピックについての情報を習得しようとする際，自身の知識が不足していると理解が困難になる．このような状況下では，ユーザは自身の知識外の未知のトピックについて理解するのに何が不足しているのかを把握することは困難である．そこで本研究では，一般的な知識グラフに基づき，段階的に知識を習得することを目標とし，既知のトピックと現在検索しているトピックを繋ぐトピックを抽出し，補間する手法を提案する．具体的にはWikipediaのリンク構造に基づき，トピックを繋ぐパス上にある補間候補ノードの重要度を，補間候補ノードの接続強度，補間候補ノードの集約性，未知ノードの関連性などの観点で算出することで補間ノードを決定する．補間ノードと未知ノードの語を組み合わせた検索クエリを作成し，その検索結果を閲覧することで未知トピックへの理解が容易になるかを評価する．

キーワード　Wikipedia，グラフ，補間トピック

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．未知の知識を習得する際，自分の知識が不足していると理解が困難になる場面がある．例えば，機械学習の基礎を知っている人が自然言語処理の BERT を学習しようとする場合，自然言語処理や機械学習のモデルなどの知識があればより理解が容易になると考えられる．ユーザの過去の Web 閲覧履歴から抽出したキーワードを既知のトピック，ユーザの今の検索キーワードを未知のトピックとすると，今の検索結果を理解するために必要となるキーワードは，その補間となるトピックといえる．そこで本研究では，一般的な知識グラフに基づき，段階的に知識を習得することを目的とし，既知のトピックとユーザが得たい未知のトピックを繋ぐ補間トピックを抽出し，検索クエリとして提示する手法を提案する．具体的には，一般的な知識グラフにはWikipediaのリンク構造を用い，グラフ上にて，トピックを示すノードを繋ぐパス上にある補間トピック候補のノードの重要度を算出する．我々は [1] にて，算出する際の観点として，補間ノード間の接続強度，補間ノードの集約性，既知ノードと未知ノードの対等性で算出した．しかし，対等性では関係のあるノードも候補から外れてしまうことがあることから，対等性を省き，未知ノードとの関連性の観点を加えて算出する．それに加えて，本稿では算出したスコアが高い補間ノードと未知ノードの語を組み合わせた検索クエリを作成し，その結果を閲覧することで未知トピックへの理解が容易になるかを評価する．

2　関連研究

2.1　全要把握

湯本ら [2] 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 [3] 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．

村山ら [4] 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．

梅本ら [5] 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

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ので，ユーザが現在何を把握していて，何を知るには何が必要などといった考えが考慮されていない．

2.2 単語の関係性把握

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．

倉門ら [8] 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．

隅田ら [9] 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．

大政ら [10] [11] は，Wikipedia のリンク構造を用いて，ある記事の関連概念の抽出と不適切なリンクの修正手法を提案している．関連概念に関しては，いくつかの関連概念としての指標を作成しスコアを算出し，不適切リンクに関しては，機械学習を用いてパターンを分析し，不適切なリンクの修正をユーザに提案している．

これらの関連研究は，あるトピックに対して単語間の関係性を明らかにすることを目的としている．本研究とは単語をトピックと見立て，単語間の関係を利用し得たい情報までを補間する情報を抽出するという点で異なる．

3 提 案 手 法

3.1 Wikipedia を用いた知識グラフの作成

各記事をある 1 つのトピックとみなし，記事タイトルをノード，リンク元から先への関係をエッジとした有向グラフ型データを作成する．Wikipedia の dump データから記事間のリンク構造を取り出す．dump データは 2022 年 7 月 1 日時点で最新のデータを使用し，parser には wikiextractor[1]を用いた．このデータを用いて，記事とその記事のリンクの参照先の記事タイトルの関係を作成する．データ作成の概要図が図 1 となる．大政ら [10] [11] が，Wikipedia のリンク構造を用いて，ある記事の関連概念の抽出と不適切なリンクの修正手法を提案している．そのなかの接続強度を利用する．エッジについては，その接続

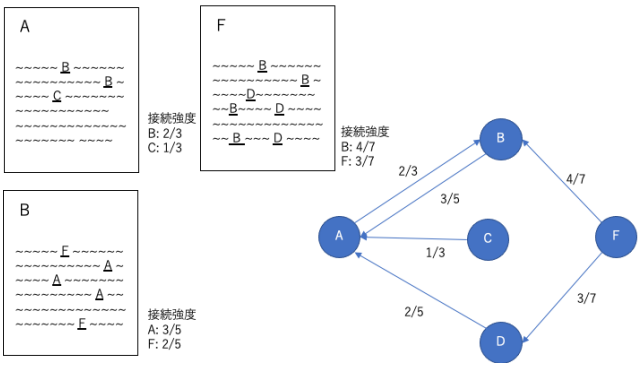

図 1 データ作成と接続強度の算出

強度を求める．リンクの参照元ノードを A，先ノードを B とすると，A の記事中に出現する B へのアンカーテキストならびに B の記事タイトルの出現回数を求める．これを A から他のノードへのリンク数の合計で除算したものを接続強度とする．また，各ノードには Wikipedia のそのノードがタイトルの記事にあるカテゴリの情報が付与されている．

3.2 既知ノードの抽出

ユーザの閲覧履歴から，既知であるトピックを抽出し，それを用いてグラフ上で既知ノードを決定する．既知ノードとして望ましいものとしては，ユーザの既知のトピックを示しており，なおかつ未知ノードと関連性があるものである．このとき，未知ノードは次に習得したいトピックについてのノードとなる．本稿では，入力された検索キーワードに一致する Wikipedia ページを未知ノードとする．既知トピックを示している語を TFIDF 法により抽出し，未知ノードとの関連性は未知ノードが属している Wikipedia のカテゴリを利用する．

まず，ユーザの閲覧履歴から，そのユーザの履歴の特徴を示している語の抽出を行う．使用する単語は，形態素解析器 MeCab [12] を用いて，人名，数，代名詞以外の名詞を原型に直し，なおかつ Wikipedia の記事タイトルに使われている語に限定した．すなわち，この単語は，対応する Wikipedia ページのカテゴリ等の情報も持つ．本稿では，辞書は ipadic-neologd[2]を用いた．特徴語の抽出には TFIDF 法 [13] を用いる．$tfidf$ 値に関しては式 1 で定義した．式 1 の第 1 項では閲覧した記事集合 D における単語 w の出現頻度を定義している．式 1 中の C は D における w の出現回数，L は記事集合 D における全単語の出現回数の和を示している．式 1 中の S は閲覧履歴でなく，Wikipedia 全記事数，$S(w)$ は単語 w を含む Wikipedia の記事数を示している．

$$TFIDF(D,w) = \frac{C(D,w)}{L(D)} \times log(\frac{S}{S(w)+1}) \qquad (1)$$

次に，算出した重要語について， 未知ノードと関連性のある度合いを算出する．図 2 が，未知ノードの関連性を度合いを算出する概要図である．算出方法としては，Wikipedia の各カテゴリの共起関係を利用し，カテゴリ間の関連性を用いる．具体

1：https://github.com/attardi/wikiextractor

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

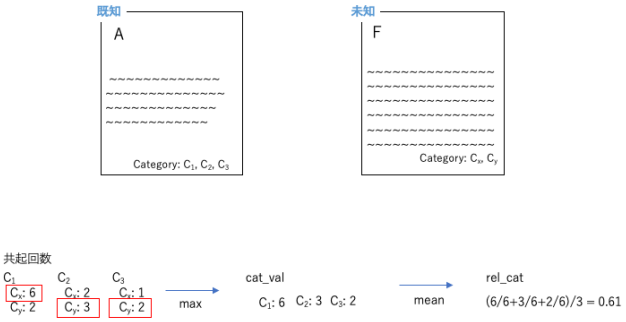

図 2　関連性の値の算出概要

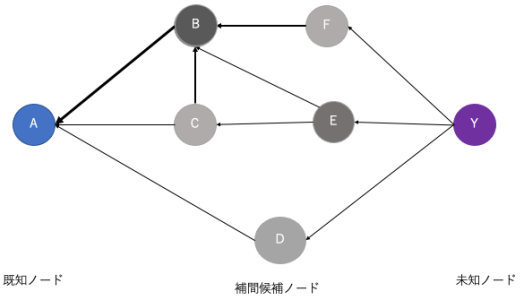

図 3　スコアの算出の概要図

的には，Wikipedia の各カテゴリが，同じ記事において同時に何回出現するかという共起回数を基に関連性の値 rel_cat を算出する．図 2 を例に，算出手順を説明する．既知の記事 A と未知の記事 F があり，それぞれ C_1，C_2，C_3 と C_x，C_y を持っている．C_1 が共起する C_x の回数は 6 回，C_y の回数は 2 回となる．これを既知の記事のカテゴリについてそれぞれ行う．その後，未知ノードが持つ各カテゴリと共起する最大共起回数を，その既知記事中の各カテゴリの共起回数とみなす．それらの各共起回数を既知記事のカテゴリの最大共起回数で割り，平均したものがその記事と未知の記事と関連性となる．この手順を式で表すと式 2 となる．未知ノードが持つ各カテゴリ c_u について，共起する既知のカテゴリ c_w の最大共起回数 cat_val を計算する．このとき，未知ノードのカテゴリと，既知ノードのカテゴリで同じものがあれば，その未知カテゴリと共起する他のカテゴリの最大値に 1 を足したものとする (式 4)．その後，その共起回数の平均値を関連度 rel_cat とする．式 2 と式 3 の w と u は記事なおかつノードであり，C_w と C_u は w と u がもつカテゴリについて表している．

$$rel\_cat(w,u) = mean_{c_w \in C_w}\left(\frac{cat\_val(c_w,u)}{cooc(c_w,c_w)}\right) \tag{2}$$

$$cat\_val(c_w,u) = max_{c_u \in C_u}(cooc(c_w,c_u)) \tag{3}$$

$$cooc(c_i,c_j) = \begin{cases} \#\text{ of cooc. } c_i \text{ and } c_j \text{ in Wikipedia} & c_i \neq c_j \\ max_{c_k \in C \setminus c_i}(cooc(c_i,c_k)) + 1 & c_i = c_j \end{cases} \tag{4}$$

式 1 と式 2 で算出した値を用いて既知ノードを決定する．既知ノードの決定に用いる値の算出方法は，まず閲覧履歴から抽出した語の $tfidf$ 値と，その語と同一のノードの各カテゴリの平均値を乗算したものとした．この値の上位 K 件を既知ノードとし，式 5 で定義する．抽出した既知ノードに未知ノードが含まれていたら，その既知ノードを除去する．

$$know(D,w,u) = tfidf(D,w) \times rel\_cat(w,u) \tag{5}$$

3.3　補間ノードの決定

Wikipedia のリンク構造に基づき，既知であるトピックと習得したいトピックを補間するトピックを抽出する．以降，既知であるトピックを既知ノード，習得したいトピックを未知ノード，補間するトピックの候補を補間ノード候補とする．3.1 節で得られたデータを用いて既知ノードと未知ノードを補間するノードを得るためのスコアを算出する．

Wikipedia の各記事に出現するリンクは，記事のタイトルの概念を構成するものと仮定する．補間ノードの候補は，未知ノードから既知ノードを有向グラフで n ホップ以内で繋ぐノードであり，以降の補間ノードを抽出するためのグラフは，これらのノードその間のエッジのみで構成された有向グラフとなる．本稿では $n = 3$ とする．

スコア付けの観点として提案手法中で使用する観点を下記に示す．

- 接続強度: 補間ノード候補は重要なノードであるか
- 集約性: 補間ノード候補は他の補間ノード候補と関連があるか
- 関連性: 補間ノード候補は未知ノードと関係性のあるノードか

各観点について説明する．

1 つ目の接続強度について，これは記事中に占めるリンクの割合が多ければ多いほど 2 つの記事は関連性が高く，重要だと考えられる．よって，3.1 節で記載したエッジの属性を用いてこの接続強度を求める．

2 つ目の集約性について，これはある補間ノード候補が他の補間ノード候補への関連が多ければそのノードは補間候補内で重要なものであるという考えに基づいたものである．

3 つ目の関連性について，これは接続強度や集約性が高くても，未知ノードとは無関係なノードが補間として良いと判定されることを防ぐ意図で用いる．その際，未知ノードからのホップ数と，式 2 で算出したカテゴリの関連性を用いる．具体的には，各ノードが持っている各カテゴリの値の平均値を，未知ノードからのホップ数で割ったものを用いる．

これらの観点の概要を図 3 に示す．ノードとノードのリンクの太さを接続強度，ノードへのリンクの多さを集約性，ノードの色の濃さで関連性を示している．A を既知ノード，Y を未知ノードとし，B,C,D,E,F を補間ノード候補としている．図から，AB 間の接続強度が最も強く，B の集約性が高く，D や E とは違い灰色が濃いことから，この図中で補間ノードとして最も相応しいのは B となる．

これらの観点より，多くのノードとつながっているノードは

表 1　機械学習について検索したユーザが BERT について知ろうとしているときの補間ノード

ランキング	補間ノード
1	Transformer (機械学習モデル)
2	能動学習
3	自然言語処理
4	教師なし学習
5	教師あり学習

表 2　Python 検索したユーザが Github について知ろうとしているときの補間ノード

ランキング	補間ノード
1	Erlang
2	Git
3	Ruby on Rails
4	リポジトリ
5	ソースコード

重要であり，また重要なノードと多くつながっているノードは重要であるということから，PageRank を用いてスコアを算出することを考える．エッジをパス，エッジの接続強度を遷移確率とみなすことで，PageRank アルゴリズムを適応ことができる．このことにより，補間ノード候補群から重要なノードを決定できると考える．その後，RageRank アルゴリズムで算出した各補間ノードのスコアに対して関連性の観点から算出した値を乗算し，未知ノードからのホップ数で割る (式 6)．h は補完ノードであり，hop は 2 ノード間のホップ数，$pagerank(h)$ は h の pagerank 値を算出する．この値の上位 M 件を補間ノードとする．

$$score(h,u) = pagerank(h) \times \frac{rel\_cat(h,u)}{hop(h,u)} \tag{6}$$

3.4　補間ノードの出力例

未知ノードから既知ノードに対して，有向グラフ上で 3 ホップでたどり着く全てのノードを補間ノード候補として扱った例について説明する．著者が既知のトピックと未知のトピック，既知のトピックについての閲覧履歴を仮想で収集し提案手法を適用した．既知ノードの数 K を 5，補間ノード数の M を 5 とする．想定するシチュエーションとしては，(1) 機械学習について知っているユーザが新たに BERT についての知識を得ようとしているものと，(2)Python でのプログラミングの基礎を知っているユーザがソースコードを保存するために Github について知ろうとしているものの 2 つとする．ユーザの閲覧履歴を，機械学習について検索したユーザの履歴，未知ノードを「BERT」として提案手法を適用した結果の一部を表 1 に示す．同様にユーザの閲覧履歴を，Python でのプログラミングの基礎を検索したユーザの履歴，未知ノードを「Github」として提案手法を適用した結果の一部を表 2 に示す．

表 1 と表 2 は情報学の学習関連のトピックである．表 1 について，機械学習について理解しているユーザが，機械学習系に関連しつつ，BERT のような自然言語処理に関連する Transformer や自然言語処理を補間としてだせているのは望ましい結果であると考える．表 2 について，Python について理解しているユーザが，プログラミング系に関連しつつ， Github について学ぶための補間候補として，2 位に Git が出現している．Github は Git を利用したサービスであり，ユーザも Git を学ぶ必要があることから，補間として適していると考える．

表 3　収集したトピックの一部

検索してもらった興味のあるトピック	次に興味を持ったトピック
春闘	給料
HTML	Curl
AWS	GCP
消費税	たばこ税

4　評 価 実 験

この提案手法について，評価実験を行なった．実験は 2 回に分けて行なった．1 回目の実験は，各被験者の閲覧履歴の収集と，その後どのようなトピックについて興味を持ったのかを調査した．2 回目の実験では，その閲覧履歴を用いて補間ノードを抽出し，その抽出した補完ノードを検索クエリとして用いた検索結果を被験者に評価してもらった．

4.1　閲覧履歴の収集

被験者に対し，閲覧履歴を収集することができる自作の Google 拡張機能をインストールしてもらい，数分間自分が興味を持っているトピックについて検索をしてもらった．これにより，閲覧した記事の URL を収集した．その後，アンケートにて検索したトピックについて代表的な語と，未知ノードの決定のため，今回検索した結果次にどのようなトピックに興味を持ったのかを収集した．未知ノードについて，アンケート中では自由記述で収集しているため，著者が Wikipedia で同様の記事を検索し，その記事がない場合はその被験者のデータを使わないものとする．収集結果としては表 3 となる．結果としては，計 8 人の被験者の履歴データを採用した．

4.2　補間ノードを用いた検索クエリによる実験

4.1 節で収集した各被験者の記事の URL から既知ノードを決定し，未知ノードをアンケート中の次に興味を持ったトピックとして，提案手法を適用して各既知ノードと未知ノードの補間ノードを抽出した．既知ノードの数 K を 5，補間ノード数の M を 3 とする．未知ノードは著者が Wikipedia で同様の記事を検索し，同様のタイトルや記事内容のタイトルを未知ノードとして扱う．提案手法にて出力された補間ノードのタイトルと，未知ノードのタイトルを空白でつなげた文字列を検索クエリとする．

提案手法を適用し閲覧履歴から既知の単語を抽出した結果を表 4 にまとめる．提案手法を適用し補間キーワードを出力した結果を表 5，Google による検索クエリを表 6 にまとめる．比較手法として，未知ノードのタイトルに対する Google によるクエリ推薦の上位 3 件の検索クエリとした．実験方法は以下の手順となる．

- 4.1 節にて収集した 1 人分の閲覧履歴を閲覧する
- 著者が選んだ，次に興味のあるトピックについて閲覧することで理解ができるであろう複数の記事 (未知のトピック) を閲覧してもらう
- 提案手法により出力されたクエリと，比較手法によるクエリの計 6 つのクエリによる検索結果を閲覧してもらう
- 各クエリに対して，未知のトピックを知るために必要な前提知識として重要だと思う度合いを 5 段階評価してもらう

クラウドワークスにて，各タスクにつき 50 件の計 350 件を公開し，130 人に対して実験を行った．結果を表 7 にまとめる．提案手法スコアと比較手法スコアは，各手法で出力したクエリの評価を平均した値をトピックごとに平均した値となる．

5 考　　察

提案手法を適用し補間キーワードを出力した結果 (表 5) を見ると,「春闘」,「スターバックス」,「行動経済学」の場合は補間として考えられる，未知トピックに対して足りていないトピックを出力されていると考える．

結果から提案手法でのスコアの値の方が比較手法よりも高いものが 8 つ中 6 つある．このことから，提案手法により補間クエリを出力できている可能性があると考える．著しく提案手法のスコアが低い，既知トピックが WBC の補間ノードのタイトルは「リリーフ」,「予告先発投手」,「先発投手」となっていて，2 つ目の補間ノードを組み合わせたクエリの値が特に低かった．収集した履歴中には，対戦成績や WBC の投手などが出現していた．このことから，提案手法では MLB と WBC の投手についての違いが抽出されたと考える．反対に，比較手法では MLB の一般的な成績やシーズンについてなどが抽出されていた．検索クエリを評価する被験者は投手の違いよりも，成績などが前提知識として重要であると考えたのだと考察する．履歴中に出現する成績などは表データであったため，既知ノード決定時の特徴の抽出時に，人の目で見て重要であると思える観点を考慮する必要がある．

また，最適な既知ノード (K) の数を決定するアルゴリズムが欠けていることから，実際に応用する際にはトピックのジャンルごとにこの数値を決定するアルゴリズムを組み込む必要があると考える．表 4 を見ると，既知トピックが WBC の閲覧履歴から取り出した単語は，WBC に関するものが少なかった．これも同様に提案手法のスコアに影響していると考える．よって既知の単語の抽出方法にも改善の余地がある．

6 ま と め

未知の知識を習得する際，自分の知識が不足していると理解が困難になる場面があるという考えから，段階的に知識を習得することを目的とし，既知のトピックとユーザが得たい未知のトピックを繋ぐ補間トピックを抽出した．Wikipedia 上において，トピックを示すノードを繋ぐパス上にある補間トピック候補のノードの重要度を，ノード間の接続強度，集約性，関連性などの観点で算出することで算出し，補間ノードを抽出した．ユーザが得たい知識の関連知識や他のユーザが同時に調べているようなものを提示するものではなく，既知の知識と未知の知識を結ぶことができる知識を抽出するものである．

通常の検索クエリ候補と比較した評価実験により，提案手法が優位であることを示した．これにより，Google による検索クエリ候補とは異なり，未知の知識に対する前提知識をユーザが習得することが可能であると考える．しかしながら，提案手法に適用する最適な既知の知識の量の選定方法や，出力される補間知識が既知の知識と類似していた場合の対処を検討する必要がある．また，一般的な知識構造からグラフを作成したが，ノード間のエッジの属性を考慮したアルゴリズムを考慮する必要もあると考える．

本研究により，ユーザの既知の知識を利用して未知の知識の習得の補助ができるようになると考える．今後としては，ユーザの既知の知識を起点に，習得という観点で相性の良い未知の知識の抽出があると考える．

謝　　辞

本研究の一部は，2022 年度科研費基盤研究 (C)(課題番号：21K12147) 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] 波木井征, 北山大輔. 閲覧履歴と検索結果に対する wikipedia を用いた補間キーワードの抽出手法. 電子情報通信学会技術研究報告, Vol. 122, No. 176, 2022.

[2] 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, 2007.

[3] 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, 2019.

[4] 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, 2020.

[5] 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.

[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, 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, 2020.

[8] 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, 2010.

[9] 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.

[10] 大政飛鳥, 井上潮. グラフ型データベースを用いた wikipedia からの関連概念抽出手法,. 第 11 回データ工学と情報マネジメントに関するフォーラム, 2019.

[11] 大政飛鳥, 井上潮. グラフ型データベースを用いた wikipedia における不適切リンクの修正手法,. 第 12 回データ工学と情報マネジメントに関するフォーラム, 2020.

[12] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empir-*

表 4　閲覧履歴中から抽出した単語

既知トピック	抽出したキーワード
春闘	春闘，賃金，ベースアップ，ステートメント，人事労務管理
HTML	Python，スクリプト言語，HTML，プログラミング言語，JavaScript
AWS	サービス，オンラインストレージ，サーバ，オンプレミス，Amazon Web Services
消費税	税率，決算，消費税，軽減税率，予算
WBC	ワールド，投手，ネクスト，サイト，コラム
カクテル	蒸留酒，ラム酒，焼酎，ソムリエ，リキュール
スターバックス	コーヒー，メルカリ，スターバックス，スターバックスコーヒージャパン，福袋
行動経済学	心理学，同値，説明，経済，アスタリスク

表 5　各トピックにて出力された補間ノードのタイトル

既知トピック	未知トピック	補間 1	補間 2	補間 3
春闘	給与	手取り給与	年棒	賃金
HTML	Curl (プログラミング言語)	Squirrel	Tcl/Tk	マークアップ言語
AWS	GCP	Google App Engine	クラウド コンピューティング	Infrastructure as a Service
消費税	たばこ税	租税	道府県たばこ税	地方たばこ税
WBC	MLB	リリーフ	予告先発投手	先発投手
カクテル	ザ・カクテルバー	アサヒ カクテル パートナー	発泡酒	チューハイ
スターバックス	コメダ	モーニング サービス	喫茶店	コッペパン
行動経済学	ベイズ確率	統計学	母集団	事前確立

表 6　各トピックにて Google による検索クエリ候補

既知トピック	未知トピック	推薦キーワード 1	推薦キーワード 2	推薦キーワード 3
春闘	給与	給与支払報告書	給与支払報告書総括表	給与明細
HTML	Curl (プログラミング言語)	コマンド	オプション一覧	”-f”
AWS	GCP	資格	console	ログイン
消費税	たばこ税	税率	増税	なくなると
WBC	MLB	個人成績	ポストシーズン	mvp
カクテル	ザ・カクテルバー	チューハイ	プロフェッショナル	シェーカー篇
スターバックス	コメダ	珈琲	福袋	メニュー
行動経済学	ベイズ確率	例題	わかりやすく	モンティホール問題

表 7　実 験 結 果

既知トピック	未知トピック	提案手法スコア	比較手法スコア
春闘	給料	**3.28**	2.55
HTML	Curl(プログラミング言語)	**3.21**	3.20
AWS	GCP	**3.43**	2.81
消費税	たばこ税	**3.80**	3.57
WBC	MLB	3.25	**3.45**
カクテル	ザ・カクテルバー	**3.54**	3.23
スターバックス	コメダ	3.69	**3.72**
行動経済学	ベイズ確率	**3.81**	3.50
全体平均		**3.50**	3.25

ical Methods in Natural Language Processing, pp. 230–237, 2004.

[13] Juan Ramos. Using tf-idf to determine word relevance in document queries, 1999.