技術ブログにおける既知フレーズとの共起性に基づく補完トピック抽出手法

波木井　征†　　北山　大輔†

† 工学院大学大学院情報学専攻　〒 163–8677 東京都新宿区西新宿 1 丁目 24-2
E-mail: †em21018@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可能な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，そのトピックについて深く知りたい場合や閲覧した記事の抜けている知識を知りたい場合に，既存の検索エンジンでは，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して既知フレーズとの共起性に基づいた補完トピックの抽出手法を提案する．具体的には，検索結果集合から，トピックを表すフレーズを作成し，フレーズの既知/未知の関係から既知を補完することのできる未知フレーズを抽出する．これにより，ユーザはトピックに対する閲覧した記事の補完トピックを知ることができる．

キーワード　補完トピック，LexRank，技術ブログ，検索支援

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，python のライブラリについて検索している場合，使用方法についての記事のみの閲覧では，知識が全くない場合パフォーマンス向上や設定方法などのユーザが未閲覧である補完トピックが存在することに気がつかない可能性がある．

我々は，検索トピックに対して，ユーザの閲覧トピックに関係のある未閲覧のトピックを抽出して提示することで，この問題が解決可能であると考えた．例えば，「python ライブラリ」について検索し，あるライブラリに関する記事を閲覧したとする．出力としては，そのライブラリの使用方法や作成方法などといったトピックを提示することを想定している．

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を利用することで，閲覧した記事に含まれるフレーズとの共起する未知のフレーズを見つけることができ，より共起性の高いフレーズであり，次に習得すべきフレーズを抽出する手法を提案する．具体的には，閲覧記事からフレーズを抽出し，そのフレーズと共起するフレーズを未閲覧記事から抽出する．その未閲覧記事から抽出したフレーズの重要度と共起性を考慮した値を算出し，その値が上位のフレーズに含まれるキーワード集合を次に習得すべき補完トピックとして抽出する．想定するシステムの UI を図 1 に示す．ユーザは検索キーワードを入力し，記事を閲覧した後にナレッジパネル上で閲覧した記事と，次の知識獲得のための参考情報を得ることができる．閲覧した記事の下には，閲覧トピックに関係のある未閲覧のトピックが含まれる記事のタイトルと，そのトピックの単語集合を載せている．

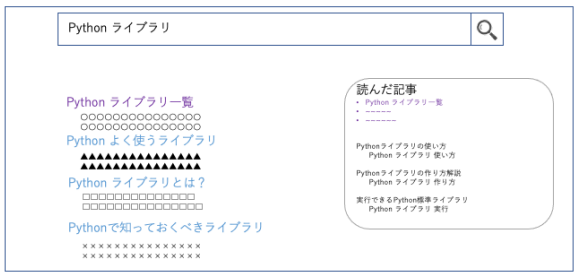

図 1: 想定するシステムの UI

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 節では評価実験とその考察について述べる．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

2.1　全要把握

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

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

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．

村山ら [3] 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．

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，ユーザにとっての習得順序を考慮していない．本研究では，どのようなトピックを習得すれば検索キーワードに対しての全容を得やすいかを考慮している点で異なる．

2.2 単語の関係性把握

南川ら [4] [5]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 [6] 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 [7] 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 [8] 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 [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．

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．本研究とはあるトピック内の単語間の関係性を明らかにする点は似ているが，習得推薦度を定義して習得の優先度合いを考慮している点で異なる．

2.3 検索支援

山本ら [10] は，信憑性指向のウェブ検索を支援するために，新しいタイプのクエリ推薦手法を提案している．提案システムはユーザからクエリが入力されると，クエリに関するセンテンスのうち，ウェブ上で反証されているセンテンスをリアルタイムに抽出・収集する．その後，収集された被反証センテンスのウェブ上における典型度，およびクエリとの関連度を計算する．最終的に，典型度および関連度が高い反証センテンスの上位 N 件を，検索中のユーザに提示する．

三好ら [11] は，ユーザが閲覧しているニュースにおいて，具体的に記載されていない内容を補完するニュースの推薦を目指す．初めにトピックモデルを作成する手法である LDA を用いてモデルの作成を行う．作成したモデルを適用し，ニュースのトピック分布を算出しユーザが閲覧しているニュースに関連するニュースをコサイン類似度を用いて検索する．次に LexRank を用いて，ユーザが閲覧しているニュースと関連ニュースに対して, 1 文ずつ重要度を算出する．最後に，重要度が高い文をニュースに詳しく記載されている内容，重要度が低い文をニュースに詳しく記載されていない内容とし，ユーザが閲覧しているニュースの重要度が低い文と関連ニュースの重要度が高い文の類似度を Doc2Vec を用いて算出する．算出した類似度が事前に定めた閾値を上回った場合，関連ニュースを閲覧中のニュースを補完するニュースとして推薦する．

灘本ら [12] は，見落とされた視点をコンテンツホールと呼び，SNS やブログにおけるコミュニティ内の議論の履歴からコンテンツホールを抽出しユーザに提示することを試みている．そのための第一歩となる Web 空間のあるテーマに対する視点抽出と視点間の比較によるコンテンツホールの抽出を行う．具体的には「名詞 A+が+形容詞+名詞 B」の構造に注目し，あるテーマ名詞 B に対しその視点構造を「名詞 A ＋形容詞」であると定義し，Web 空間とコミュニティ内との 2 つの視点構造を抽出する．それら視点構造を比較しその差分情報を取得することによりコンテンツホールを抽出する．

これらの関連研究は，検索トピックに対して，より理解や質の向上を目的とした支援を行っている．本研究では，どのようなトピックを習得すれば検索キーワードのトピックに対して理解しやすいかの度合いとして，習得推薦度を定義している点で異なる．

3 提案手法

3.1 概　　要

検索結果集合に対して，ユーザの過去の閲覧履歴から漏れていて，次に習得すべきトピックを含むフレーズを提示する．提案手法の手順としては大きく分けて以下となる．

（1） 検索クエリに基づく記事集合を取得
（2） 記事集合からトピックを表すフレーズを抽出
（3） 未知フレーズの習得推薦度を算出
（4） 未知フレーズの冗長性を排除しキーワードへ分解

(1) の記事集合の取得では，記事のタイトルもしくは本文に検索キーワードを含む記事集合を抽出する．検索キーワードが複数ある場合は，各キーワードで抽出できる記事集合の積集合を抽出する．

3.2 記事集合からトピックを表すフレーズを抽出

通常，文は複数のトピックを含んでいる．本研究では，1 文につき 1 つのトピックを含んでいる状態が望ましい．すなわち，簡潔でありトピックを示している文である．よって，文に対して係受け解析を行い，係受け先の最後の単語が名詞もしくは動詞もしくは文末になるまで結合したものを 1 フレーズする．また，習得すべきフレーズとして，係受け先の最後の節の中の単語が不適切な場合がある．例えば,「埋め込む」や「印象」という単語である．よって，係受け先の最後の節の中に表 1 の単語が含まれている場合，そのフレーズはトピックを表すフレーズとして採用しない．トピックを表すフレーズ例としては,「MySQL をインストール」,「データベースへの接続方法」,「Python との連携方法」などである．

なお，各フレーズを構成するベクトルは，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，各単語のベクトルに tfidf 値をかけたものを平均して作成した．tfidf 値に関しては式 3 で定義した．式 1 ではフレーズ 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 はフレーズ d における全単語の出現回数の和を示している．式 2 中の S は全記事数，$S(w)$ は単語 w を含む記事数を示している． ベクトルに使用する単語は，形態素解析器 MeCab [13] を用いて，名詞と動詞のみを使用した．名詞は，人名，数，代名詞は排除した．本稿では，辞書は ipadic-neologd[3]を用いた．

$$TF(d,w) = \frac{C(d,w)}{L(d)} \tag{1}$$

$$IDF(w) = log(\frac{S}{S(w)+1}) \tag{2}$$

$$TFIDF(d,w) = TF(d,w) \times IDF(w) \tag{3}$$

3.3 未知フレーズの習得推薦度を算出

未閲覧の記事集合中のフレーズに対して，次に習得すべき度合いである習得推薦度を算出する．以下に高い習得推薦度となる基準を示す．

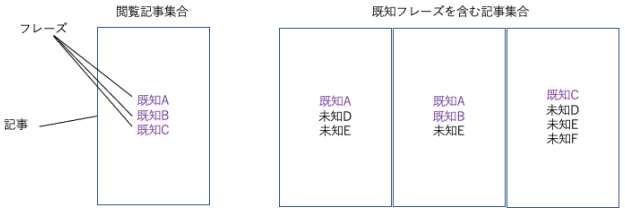

図 2: 既知フレーズを含む記事に出現しやすいフレーズ

（1） 検索結果記事中での重要度が高い
（2） 既知フレーズを含む記事における共起度が高い

これらは，重要なものは習得すべきであるという考えと，閲覧した記事に含まれるフレーズ (以降，既知フレーズと記す) が他の記事にも出現していたら，その記事に含まれるフレーズは既知の知識と関係があり，次に習得すべきであるという考えに基づく．

上記 2 つの考えから，習得推薦度の算出方法は式 4 と定義する．式中の F はフレーズの重要度，K は既知フレーズとの共起度，t は対象のフレーズ，Q は入力クエリによる検索結果の全記事となる．

$$rec(t,Q) = F(t) \times K(t,Q) \tag{4}$$

3.3.1 検索結果記事中での重要度

本研究では，重要度の算出に要約アルゴリズムである LexRank [14] を用いる．LexRank とは，Erkan らが提案した PageRank を応用したテキスト要約手法である．Lexrank は多くの文と類似する文は重要な文かつ，重要な文と類似する文は重要な文という考え方である．文をノード，文間の類似度をエッジとしたグラフを作成し，グラフから文の重要度を算出する．文をフレーズに置き換え，重要かつ典型的なフレーズの値が高くなる．表 2 に検索キーワードを「mac mecab」として LexRank を適用した結果の一部を示す．

3.3.2 既知フレーズを含む記事における共起度

2 つめの基準は，閲覧した記事に出現するフレーズを含む記事に出現しやすいフレーズであるかどうかである．各未知のフレーズにおいて，その未知フレーズが出現する未閲覧記事中に既知フレーズを含む記事数と定義する．ここで，全フレーズのうち既知フレーズを除いたものを未知フレーズと定義する．図 2 を用いて考え方を説明する．左が閲覧した記事であり，記事の中のフレーズ ABC を閲覧している場合，既知フレーズを含む記事集合中のフレーズ DEF に関して，D は 2 つの記事に出現，E は 3 つの記事に出現，F は 1 の記事に出現している．この場合，一番既知フレーズと共起する度合いが高いのは E となる．

実行手順は以下である．

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

表 1: ストップワード (一部)

実施, 話, 差し替え, 載, OK, 違い, 移動, 置, ならない, いかない, 優, 異なる, 思, 例, 消す, 困難, 指定, 打, 解説, 用意, 一緒, 割愛, 強化, 提供, 疑, 構, 分か, 推奨, 選ぶ, 完了, 説明, 不要, 整, 開く, 加工, 異なる, 統一, 補足, 問い合わせ, 石, 該当, 驚, いいかも, お話, 表される, 登録, 面倒, 汚, test, マッチ, 渡, 意味, 省, 送る, 困, 簡単, 疲れ, 打ち込む, 概要, 組む, 厚, 対応, 詰, 採用, 落ち, 隔離, 切り出, 手探, 到着, 混ぜる, 取り除, プロファイリング, お勧め, 用い, 検索, 認識, 直面, コピペ, 一つ, 走, 実現, 終了, 囲む, 区別, 注意, 命令, 行, 示す, 混在, おすすめ, 取り扱い, 進, 目的, 考え方, 出来, 表す, 拳, ついて, 確か, 導く, 課題, ある, 集計, 印象, 付加, 動かす, 一般, 埋め込む, 使, 一覧, 始ま, 絞, 用意, 表現, 書, 把握, 足, 考える

表 2: LexRank 適用結果

フレーズ	LexRank 値
Mecab の形態素結果と一部単語はアプリ	0.000663
WEB 便利ツールで単語は取得できるようになりましたが、文章	0.000663
手動で単語リスト作るのは微妙なので、自動作成したい	0.000599
正規化済みの日本語テキストを単語分割する	0.000598
文脈と感情情報の文章からの読み取りと会話ランク付けへの利用	0.000593
有用な情報抽出には使用しづらいのが現状です	0.000590
記事から MeCab で単語だけ切り出して変換	0.000590

（1） 全フレーズから，既知フレーズを判定

（2） 全フレーズから既知を抜いたものを未知フレーズとする

（3） 各未知フレーズに対し，その未知フレーズが出現する記事集合を抽出

（4） その記事集合中の既知フレーズが出現する記事数を算出

この時，既知フレーズ・未知フレーズに似ているフレーズは同じように既知フレーズ・未知フレーズとして扱う．

3.4 未知フレーズの冗長性の排除と単語へ分解

次に習得すべきフレーズから冗長性を排除するために，MMR [15] という文書要約手法を使う．MMR は式 5 で定義される．これは関連性を維持しながら，冗長性を排除することを目的として抽出する文を順に決定していく手法である．

本手法では，R はランク付けされたフレーズ集合，S が選択済みのフレーズ集合となり，rec が次に習得すべき度合いの値，sim がフレーズ間の類似度となる．このとき，rec は sim との値域を合わせるために正規化を行う．

冗長性を排除した後，フレーズから単語を抽出する．単語は，ベクトルを構成する名詞と動詞を抽出した．

$$\arg\max_{t_i \in R/S} \left[\lambda rec(t_i, Q) - (1-\lambda) \max_{t_j \in S} sim(t_i, t_j) \right] \qquad (5)$$

4 評価実験

4.1 評価実験

提案手法の有効性を示すために，評価実験を行った．式 5 の λ の値は 0.7 とした．比較手法としては，以下の 2 つを設定した．

- 次に習得すべきトピックを含むフレーズの度合いを重要度のみとし，式 4 の第 1 項のみ用いる手法 (以降重要度重視)

表 3: 「mac mecab」で閲覧した記事に含まれるフレーズ (一部)

Mac に mecab インストール，mecab の準備，mecab のインストール，mecab の辞書をインストール，mecab の辞書をパワーアップ，辞書が入っている，mecab-python3 をインストール

- 次に習得すべきトピックを含むフレーズの度合いを既知フレーズとの共起性のみとし，式 4 の第 2 項のみ用いる手法 (以降共起性重視)

この 2 つの手法にした狙いは，フレーズの重要度とユーザの知識に関係のある度合いを組み合わせた手法は，有用であるかを検証するためである．各手法で次に習得すべきトピックを含むフレーズを算出し，各フレーズから抽出された各単語集合についての正解率を算出する．

被験者は大学生 7 人で，ある検索キーワードで検索したと仮定してもらい，検索結果からある記事を閲覧した上で，各手法で抽出した 5 つの単語集合を提示し，次に各単語集合に検索キーワードとして参考になる単語が何個含まれているかを回答してもらった．検索キーワードは「mac mecab」と「python ランダムフォレスト」の 2 種類とした．「mac mecab」において被験者は mac に mecab をインストールするための記事を閲覧したと仮定する．閲覧した記事に含まれるフレーズは表 3 に記載する．提案手法を実行した結果が表 4 となる．「python ランダムフォレスト」において被験者は python を使ってランダムフォレストを実装する方法や，ランダムフォレストについて基礎の記事を閲覧したと仮定する．閲覧した記事に含まれるフレーズは表 5 に記載する． 提案手法を実行した結果が表 6 となる．

4.2 結果と考察

表 7 に「mac mecab」，表 8 に「python ランダムフォレスト」での実験結果を示す．正解率は，各被験者が回答した各単

表 4: 「mac mecab」で提案手法を適用した結果

次に習得すべきトピックを含むフレーズ	抽出された単語
指定したファイルを形態素解析し、計算します	指定, ファイル, 形態素解析, 計算
係り受け解析や、人名地名抽出、文章をベクトル化する	係り受け, 解析, 人名, 地名, 抽出, 文章, ベクトル
好きな文章を形態素解析します	好き, 文章, 形態素解析
形態素解析用のストップワードの定義	形態素解析, ストップワード, 定義
形態素解析結果のうち、抽出	形態素解析, 結果, 抽出

表 5: 「python ランダムフォレスト」で閲覧した記事に含まれるフレーズ (一部)

目的変数に属する，属する確率を算出する，複数の説明変数の組み合わせで算出する，算出する方法，イメージは以下で、算出する，Yes/No などの条件に属するかどうかで算出する，確率を算出する，ランダムフォレストは、アンサンブル学習法，構成される分類器，決定木を複数集めて使うので、フォレスト，木が集まってフォレスト，sklearn での決定木

表 6: 「python ランダムフォレスト」で提案手法を適用した結果

次に習得すべきトピックを含むフレーズ	抽出された単語
パフォーマンス予測の高速化の一番の課題は、点です	パフォーマンス, 予測, 高速化, 一番, 課題
分析に必要ないデータは削除	分析, 必要, データ, 削除
このバイアスを教師無しで補正するというのが研究です	バイアス, 教師, 無し, 補正, 研究
正解できる部分を抽出してゆけば，向上するはず	正解, 部分, 抽出, 向上
「未知のデータに対する性能」のことです	未知, データ, 性能

表 7: 「mac mecab」での実験結果

手法	平均正解率
提案手法	53.4%
重要度重視	39.4%
共起性重視	44.5%

表 8: 「python ランダムフォレスト」での実験結果

手法	平均正解率
提案手法	43.8%
重要度重視	38.7%
共起性重視	42.4%

語集合のうち参考になる単語が含まれている割合を平均することで算出する．表中の平均正解率は，各手法が算出した単語集合の正解率を平均したものである．

2 つの検索キーワードを用意した実験結果から，提案手法，共起性重視，重要度重視の順番で正解率が良いことがわかった．このことから，今回の提案手法は有用であり，重要度よりも共起性の方が正解率を上げることがわかる．また，比較手法において正解率が高く，提案手法で抽出できていないフレーズが存在する．表 9 に提案手法で抽出できていて，正解率が高いフレーズを示す．表 10 に提案手法で抽出できていないが，正解率が高いフレーズを示す．

この表 9 と表 10 から，共起性が高いが lexrank 値が低いために，抽出できていないものが出てきていることがわかる．このことから，重要度の算出方法に課題があることがわかる．重要度を lexrank 値にしたことにより，重要であるという観点が他のフレーズと似ているか否かというものになっていることが課題だと考える．

5 まとめと今後の課題

ある検索トピックにおいて，知識のないユーザが検索結果中の記事を閲覧したあとに，不足している知識を把握するのは困難であるという考えから，技術ブログの閲覧履歴と検索結果集合のトピックの差を抽出し，重要かつ既知の知識と関係があるトピックは習得すべきという考えから，次に習得すべき指標を作成し，それを元にトピックの抽出を行った．結果としては，評価実験により，提案手法の有用性を示すことができた．今後の課題としては，次に習得すべきトピックを含むフレーズの重要度の算出方法の改善がある．

謝　　辞

本研究の一部は，2021 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] 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] 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.

[4] 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.

表 9: 検索キーワード「python ランダムフォレスト」において，提案手法で抽出できたフレーズ

次に習得すべきトピックを含むフレーズ	lexrank 値	共起する度合い	正解率
正解できる部分を抽出してゆけば，向上するはず	0.000199787	103	64.3%
パフォーマンス予測の高速化の一番の課題は、点です	0.000234973	128	42.9%

表 10: 検索キーワード「python ランダムフォレスト」において，提案手法で抽出できなかったフレーズ

次に習得すべきトピックを含むフレーズ	lexrank 値	共起する度合い	正解率
作成中の検証用データには，含まない	0.000098	118	52.4%
任意の連続値データ，2 値データ	0.0000885	130	50.0%

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[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] 祐輔山本, 克己田中. 反証センテンスの提示による信憑性指向のウェブ検索支援. 情報処理学会論文誌データベース（TOD）, Vol. 6, No. 2, pp. 42–50, mar 2013.

[11] 良弥三好, 拓奥野. 閲覧中のニュース記事に不足している情報を補完するニュース記事推薦手法の提案. 研究報告データベースシステム（DBS）.

[12] 明代灘本, 武阿辺川, 英治荒牧, 陽平村上. コミュニティ型コンテンツのコンテンツホール抽出手法の提案. 研究報告データベースシステム（DBS）, 2007.

[13] 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.

[14] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

[15] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, p. 335–336, New York, NY, USA, 1998. Association for Computing Machinery.