

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

村山 貴志^{1,a)} 河野 翔太² 近藤 佑亮² 中林 雄一² 野々村 一步² 入江 英嗣¹ 坂井 修一¹

概要: ある分野の研究活動を始めたばかりの初心者にとって、論文や研究者といった研究情報を適切に探し出すことは簡単ではない。研究活動の初心者は分野に精通していないために、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。そこで我々はブラウジング中の 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た。Web ページのテキストや論文のテキストを利用し、単語分散表現の足し合わせにより Web ページ、論文、研究者のすべてをベクトルで表現する。ベクトルの類似度により順位付けすることで Web ページの内容に基づく検索を実現した。提案システムに対して速度評価と有用性評価を行い、実用的な時間内に実行可能なことを確認したが、検索の性能または評価方法に課題が残ることが明らかとなった。

1. はじめに

ある分野の研究活動を始めたばかりの初心者にとって、欲しい情報を探し出すことは簡単ではない。例えば、研究室やゼミに配属され初めて研究を開始する学部生は、分野全体に関する俯瞰的な知見にかけている場合が多い。そのため、自分が本当に興味のあるテーマに到達できず見落としてしまうことや、興味のある研究テーマを見つけることができてもそのテーマの先行研究を発見することができず、自らの適切なモデル構築に失敗してしまうことがある。

さらに、研究を始めたばかりの学生だけでなく、ある程度研究活動に従事している研究者であっても、過去に取り組んだ経験のあるテーマから大きく外れる分野については同様に詳しくないことが多い。これは昨今求められるような分野横断型研究・学際研究への進出に対する障害となり得る。

このような未経験分野の情報を探し出すうえでの難しさは、その分野におけるキーワードを熟知していない点に依るところが大きい。現代では情報システムの発達により多くの論文を中心とした学術情報が Web 上に蓄積され、様々な Web サイト上に分散して存在するそれらを横断的に検索する検索システムが整えられている。しかし、一般的な検索システムはキーワードをクエリとした検索を前提としている。検索エンジン大手の Google ^{*1} は通常検索サービスにおいて画像をクエリとする「Google 画像検

索」^{*2}を提供しているが、研究情報検索に特化した「Google Scholar」^{*3}ではクエリとして用いることができるのはテキストのみである。また、日本国内の論文約 2200 件を検索可能な「CiNii Articles」^{*4}もフリーワード検索を基本とするテキストをクエリとした検索が主となっている。このような情報検索環境においては、分野における、分野をあらゆるキーワードを熟知していない未経験者にとって、必要とする情報へ正確に到達することは困難である。

以上を踏まえ、我々は Web ページをクエリとした研究情報検索システム「HelloProfessor Recommendation」を提案し実装した。一般的な Web ページについて、そのページ内容と関連する論文および研究者の一覧を Web ブラウザ拡張機能を用いて表示する。検索は単語分散表現を拡張した Web サイト・論文・研究者のベクトル表現とそのコサイン類似度に基づいて行われる。さらに、このシステムの実用性を判断するために実行速度と検索結果の有用性についてそれぞれ評価を行った。

2. 関連研究

2.1 ベクトル空間モデル

情報検索におけるベクトル空間モデルとは、「クエリおよび文書を索引語の頻度などを要素としたベクトルでそれぞれ表現し、例えば両ベクトルの内積などにより文書の適合性を推定するという考え方」[1]である。Salton らはベクトル空間モデルに基づく文書検索システム「SMART」を作成し評価実験を行った [2]。また、大谷は TF-IDF と単

¹ 東京大学大学院情報理工学系研究科

² 東京大学工学部

^{a)} murayama@mtl.t.u-tokyo.ac.jp

^{*1} <https://www.google.com/>

^{*2} <https://www.google.com/imghp>

^{*3} <http://scholar.google.com/>

^{*4} <https://ci.nii.ac.jp/>

語共起に基づくベクトル空間モデル作成し、応用事例として構築した5つのシステムについて再現率と適合性の評価実験を行った [3].

2.2 単語分散表現

ベクトル空間モデルは文書のベクトル表現を必要とするが、その前提として文書の構成要素である単語のベクトル表現である単語分散表現について考える。

Mikolov らは意味的に強く関連する単語を近い距離に置くような単語分散表現を実現するフィードフォワードニューラルネットワークとして CBOW (Continuous Bag-Of-Words) と Skip-gram を提案し [4], その実装として Word2Vec を公開した. CBOW, Skip-gram 共に, 学習データにおいて同一の文章内で共起確率の高い単語同士が近い距離となるようなベクトルとして表現されるように学習が行われる. これにより, 単語分散表現の加減算により意味の加減算が可能であることが指摘されている [5]. 1つの単語は文脈に依らず必ず1つのベクトルが割りてられる.

2.3 文章のベクトル表現

Song and Roth は 単語分散表現を拡張し, 類似する単語同時の対応を取りつつ文章間の類似度を求める3つの手法を提案した [6]. これは類似度を適合性の尺度とするベクトル空間モデルと意味的には同等の検索を行うことができ, かつ単語の対応付けを行うことにより類似度計算の正確性を向上させることができる. しかし, 対応付けのためには文章内の単語の組み合わせすべての間で単語分散表現同士のコサイン類似度を計算する必要があるため計算時間が膨大となってしまう.

Le and Mikolov は文書のベクトル表現を学習するようなフィードフォワードニューラルネットワークとして PV-DM (Distributed Memory Model of Paragraph Vectors) と PV-DBOW (Distributed Bag of Words version of Paragraph Vector) を提案し, その実装として Doc2Vec を公開した. これは文書のベクトル表現を直接学習できるニューラルネットワークであるが, ベクトル間の類似度は学習データに含まれていた文書同士の間でしか計算できない. そのため, ベクトル空間モデルによる情報検索のためにはクエリも含め再度学習をしないといけない.

3. 提案手法: Web ページをクエリとする研究情報検索

3.1 HelloProfessor Recommendation

我々はブラウジング中の Web サイトからシームレスに研究情報を検索できる情報検索システム「HelloProfessor Recommendation」を提案する. フリー百科事典サイ

ト「Wikipedia」のトップページ*5で実行した際のアプリケーションの外観を図1に示す.



図1 HelloProfessor Recommendation 実行結果

HelloProfessor Recommendation はキーワードではなく Web ページをクエリとしてそれに関連する論文および研究者をそれぞれ関連度順に表示する. これにより「ある対象概念の学術情報に興味があるが, 学術的キーワードを理解していない」という研究活動初心者の陥る状況において, キーワードを用いずに必要な情報を検索することができる. 例えば, ある学術分野の入門記事の読解のより基礎的な理解を獲得した初学者が, その入門記事の中で見出した様々な疑問点に関連する最新の論文を検索しようとしたときや, 一般的な Web 記事で言及されている特定の社会課題に関して学術的なアプローチとしての論文を探し出したい研究初心者に対し, それら記事内容に関連した論文情報および研究者情報を提供することができる.

以下に続く 3.2 節で検索アルゴリズムの詳細について, 3.3 節で Web ブラウザ拡張のインターフェースについてそれぞれ説明する.

3.2 単語分散表現を拡張したベクトル空間モデルによる検索

検索は単語分散表現の足し合わせによるベクトル空間モデルのコサイン類似度判定に基づく.

Word2Vec の CBOW による単語分散表現 [4] の足し合わせとして論文ベクトルを, その足し合わせとして研究者ベクトルを表現する. まず, 論文のタイトルおよび概要の文章を分かち書きし, 各構成単語の Word2Vec[4]300次元単語分散表現を平均したものを論文のベクトルとした. 次に, 研究者については, 執筆した論文のベクトル表現を単純平均したものととしてベクトル化を行った. これにより, 単語・論文・研究者の3種類がそれぞれ同一の300次元空間にマッピングされ, ベクトル空間モデルを用いる前提が整った.

検索の実行は, 300次元ベクトル空間内にマッピングされた論文および研究者のベクトルと, クエリとなる Web ページのベクトルについて, それぞれ内積を計算し値の大きな順にリストを返すことで実現する. Web ページのベク

*5 <https://ja.wikipedia.org>

トルは、当該ページの HTML 文書から HTML タグを除去した文章について、分かち書きした各単語の Word2Vec による 300 次元単語分散表現の単純平均として計算される。クエリとなる Web ページの入力と検索結果の提示は、共に Web ブラウザ拡張アプリケーションとして実行および表示される。

3.3 検索インタフェースとしてのブラウザ拡張

インタフェースは Web ブラウザ拡張アプリケーションとして実装した。Web 上の情報検索システムとしての一般的なインタフェースは Web アプリケーションであるが、キーワードではなく Web ページを検索クエリとする本アプリケーションの特性上、Web ページの URL をユーザが入力することなく自動で取得できる Web ブラウザ拡張による実装の方がユーザの負担なく検索を行えると判断した。検索結果も Web ブラウザ拡張アプリケーション上に表示される。

4. 実装

4.1 全体構成

提案システムは、ベクトル空間モデルによる検索を実行するアプリケーションサーバと、Web ページの HTML 文書を検索クエリとして受け取り返ってきた検索結果を表示する Web ブラウザ拡張、および検索結果候補の論文および研究者情報を蓄積するデータベースサーバから構成される。

4.2 アプリケーションサーバ

AWS EC2^{*6} の t2.large インスタンス上に、Python 3.7.7 上で動作する Web アプリケーションフレームワーク Flask 1.1.1^{*7} を用いたアプリケーションサーバを構築した。アプリケーションサーバの単語分散表現は鈴木ら [7] の作成した日本語訓練済みモデル「日本語 Wikipedia エンティティベクトル」^{*8}のうち 2019 年 5 月 20 日版^{*9}を用いた。単語分散表現のための論文およびクエリ Web ページの分かち書きには MeCab 0.996[8] を用い、新語辞書 mecab-ipadic-NEologd[9][10][11] を追加した。

検索時に毎回発生する 300 次元ベクトルの内積計算は計算負荷が大きいので、高速に高次元ベクトルデータ検索が実行可能な Python ライブラリ NGT[12] の 1.8.4 を利用した。NGT ライブラリで用いるベクトル間距離の定義には、Word2Vec と距離の定義を一致させるためにコサイン類似度を設定した。

4.3 データベースサーバ

MySQL 5.7.31 上に検索結果として提示するための論文および研究者情報のデータベースを作成した。検索結果候補の研究者として東京大学大学院^{*10}の 16 の研究科^{*11}に所属する教員 1379 名の氏名を Web サイト上から手動で収集し、それらを著者とする日本語論文の情報について CiNii Articles API^{*12} を用いて合計 56783 件取得し検索結果に必要な情報をそれぞれデータベースに格納した。ただし、CiNii Articles API の仕様上同姓同名の人物を判別することができないため、同姓同名の人物が著した論文もその研究者の著作としてデータベース上は扱われる。

4.4 Web ブラウザ拡張

クライアントサイドは Web ブラウザ「Google Chrome」^{*13}の拡張機能として実装した。フレームワークには Vue.js 2.6.10^{*14}を用い、コンポーネントライブラリとして BootstrapVue 2.6.1^{*15}を用いた。ある Web ページで Chrome 拡張機能のボタンをクリックすると、その Web ページの URL をクエリパラメータとしてサーバーへ送信し、当該 Web ページに関連する論文及び研究者の検索結果を json 形式で受け取り、それぞれをボタンとして表示する機能を持つ。それらのボタンをクリックするとその論文及び研究者の Google 検索結果のページが表示される。論文及び研究者のボタンの下部にはそれぞれその論文の概要及び研究者の所属情報を、取得できた場合に限り表示する。

検索結果となる論文及び研究者は、既定ではそれぞれ最大 10 件表示されるが、設定画面にて最大表示件数をそれぞれ最大 100 件まで変更することができる。

5. 評価

5.1 評価項目と意義

「閲覧した Web ページをきっかけとして、キーワードのわからない研究分野に関する情報を検索する」という目的の達成を評価するために、速度評価と有用性評価を行う。速度評価では、実用的な実効速度で検索結果が応答するかどうかを確認する。有用性評価では、提案アプリケーションによる検索方法が既存のキーワード検索と比べどの程度関連する論文を見つけられるかを確認する。

^{*6} <https://aws.amazon.com/jp/ec2/>

^{*7} <https://flask.palletsprojects.com/en/1.1.x/>

^{*8} http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

^{*9} <https://github.com/singleton/WikiEntVec/releases/tag/20190520>

^{*10} <https://www.u-tokyo.ac.jp/ja/schools-orgs/index.html>

^{*11} 総合文化研究科と公共政策大学院を除く全研究科

^{*12} https://support.nii.ac.jp/ja/cia/api/a_opensearch

^{*13} https://www.google.com/intl/ja_jp/chrome/

^{*14} <https://jp.vuejs.org/index.html>

^{*15} <https://bootstrap-vue.org/>

5.2 速度評価

5.2.1 実験方法

日本語の巨大な Web ページでアプリケーションを実行し、レスポンスが返ってくるまでの時間を計測する。検索時にはアプリケーションサーバにおいて MeCab の分かち書き処理が発生することから、レスポンス速度はクエリとする Web ページのサイズが大きいくほど低下する。そこで、巨大な日本語 Web ページにおいて実用的な速度でレスポンスが返ってくるかどうかを計測する。クエリとして用いる巨大な日本語 Web ページとして、日本語 Wikipedia 記事の中で 2 番目に長く学術的な内容を含んでいる「反ユダヤ主義」*¹⁶のページを選択した。当該ページに含まれている語数は 174233 語であった。

計測は、キャッシュを無効化して毎回サーバへ通信するようにし、レスポンス時間を表示するように書き換えたクライアントサイドの Chrome 拡張を用いて行う。50 回の試行を行いレスポンス時間の平均と分散を求めた。

5.2.2 実験結果

表 1 速度評価実験結果

項目	論文	研究者
最小時間 [s]	10.867	8.160
最大時間 [s]	36.417	44.67
平均時間 [s]	19.03	13.23
標準偏差 [s]	5.41	7.26

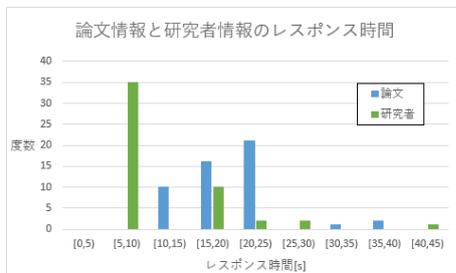


図 2 速度評価実験結果ヒストグラム

実験結果をまとめたものを表 5.2.2 に示す。また、レスポンス時間のヒストグラムを図 2 に示す。論文検索結果の平均レスポンス時間は 19.03 秒、標準偏差は 5.41 秒となった。研究者検索結果の平均レスポンス時間は 13.23 秒、標準偏差は 7.26 秒となった。

5.3 有用性評価

5.3.1 実験方法

日本語 Wikipedia の記事に関連した論文について、従来型のキーワード検索サービスである CiNii Article を用い

た検索と、提案手法である HelloProfessor Recommendation を用いた検索をそれぞれ行う。専門的な内容の日本語 Wikipedia 記事について、専門外の被験者がキーワード検索を行う場合と提案手法での検索を行う場合を比較し、記事と関連する論文が出現する順位を比較する。

被験者は 22 歳から 27 歳までの男性 5 人で、いずれも大学生または大学院生である。被験者は指定された記事について、CiNii Article を用いる方法と HelloProfessor Recommendation を用いる方法それぞれで関連する日本語論文を検索する。そして、各々の手法について関連すると考えられる検索結果が表示された順位を報告する。なお、キーワード検索に関しては初回検索キーワードで所望の結果が得られなかった際に別のキーワードで検索し直すことが許可されている。この場合順位はすべての検索キーワードに関する確認済みの検索結果数として報告する。

検索対象記事は HelloProfessor Recommendation データベースに存在する論文をランダムに 5 つ抽出し、それらと関連すると思われる Wikipedia 記事を選択した。これは、どちらの検索方法においても対応する論文が存在することを保証するためである。HelloProfessor Recommendation データベースに存在する論文の情報は CiNii Articles API により収集されているため、HelloProfessor Recommendation データベースに存在する論文は必ず CiNii Article で検索可能であることが保証されている。検索対象記事は、より具体的にはデータベースからランダムに選択された記事名と Google の Wikipedia サイト内検索により決定された。データベースから選択された記事名の先頭 2 または 3 単語を Google の Wikipedia サイト内検索にかけ、検索結果の最も上位に存在するものを実験の検索対象記事に選択した。データベースから選択された論文のタイトルと検索キーワード、最終的な Wikipedia 記事の対応は表 5.3.1 のとおりである。

各被験者ごとに報告した順位から平均逆数順位 [1] を求める。逆数順位とは報告された妥当な検索結果順位の逆数であり、その平均が平均逆数順位である。ただし、「あまり馴染みのない分野に対して研究情報の検索を行う」という趣旨であるので、被験者の専門分野に関する記事に関しては平均逆数順位の計算から除外する。

5.3.2 実験結果

実験結果を表 5.3.2 に示す。

6. 考察

6.1 実行速度

長いページをクエリとした時の平均レスポンス時間がそれぞれ 19.03 秒と 13.23 秒であったことから、現時点では実用的な速度でのレスポンスが実現できていることが確認できた。特にヒストグラムから、論文は 25 秒、研究者は 20 秒以内に大部分の場合で検索が完了していることが見て

*¹⁶ <https://ja.wikipedia.org/wiki/%E5%8F%8D%E3%83%A6%E3%83%80%E3%83%A4%E4%B8%BB%E7%BE%A9>

表 2 データベース内論文と検索対象記事の対応

データベース内論文	検索キーワード	Wikipedia 記事
BN 曲線上における Optimal Ate ベアリング向け演算 ハードウェアの最適化 (ディペンダブルコンピューティング) - (デザインガイア 2017 : VLSI 設計の新しい大地)	"BN" 曲線上	ロジスティック方程式
抗体の品質・物性評価に関する解説 (2) プロセス・製剤における評価	"抗体" の品質・物性評価	酵素
フランソワ一世とサラマンダー	フランソワ一世 サラマンダー	サラマンダー (妖精)
サイトメガロウイルス感染症 (特集 産婦人科感染症の最前線) - (母子に影響を与える感染症)	サイトメガロウイルス感染症	サイトメガロウイルス
浮体式と着床式洋上風力発電の現状と将来展望 : 世界で優位に立つため政策目標の設定と特区創設を (特集 検証 洋上風力発電)	浮体式 着床式	浮体式洋上風力発電

表 3 有用性評価実験結果

従来手法	提案手法
0.432	0.000
0.373	0.200
0.648	0.103
0.607	0.076
0.540	0.453

取れる。ただし、20 秒もしくは 25 秒のレスポンス時間は実用的な範囲内ではあるものの、一般的な検索 Web アプリケーションの基準からすればかなり長時間を要していることとなる。ブラウザ拡張として実装されているためユーザは検索中も表示されている Web ページを読むことができるためユーザエクスペリエンスの低下は限定的であると考えられるものの、可能であれば速度向上を目指したい。

また、論文検索と研究者検索でレスポンス時間に平均で 5.8 秒の差があることも考慮しなければならない。論文検索は 56783 件からの検索であり、研究者検索は 1379 件からの検索である。この件数の差が実行結果の差に現れた可能性があるが、その場合はデータベースの拡充と共にアプリケーションのレスポンス時間が著しく低下してしまうことを意味する。例えば、論文公開サービスである arXiv^{*17}には 2020 年 8 月 7 日現在 1744634 件の論文が収録されているが、レスポンス時間がデータベース件数に対し線形に増加していくのであればこの量の論文データベースに対しては実用的な時間内にサービスを提供することができなくなってしまう。現在はクエリ発行からレスポンスまでの時間のみを計測しているため、サーバーサイドのボトルネックがどの処理になっているのかは正確にはわからない。データベースの拡充等で高速化が必要になった際は各段階ごとの処理時間を計測し、問題となる部分を改善していく必要がある。

*17 <https://arxiv.org/>

6.2 有用性

有用性に関しては提案手法が従来手法であるキーワード検索より総じて低くなった。この用意としては性能上の要因と評価上の要因の二つが考えられる。

まず、性能上の要因としては、提案手法が Web ページと対応する適切な論文が提示できていないという理由である。

第一に、ベクトル空間モデルの構築アルゴリズムの素朴さが性能低下につながった可能性がある。単語分散表現を算出する際の辞書として Wikipedia 日本語コーパスで学習したものを利用したが、論文に登場する専門的な単語は、その論文を特徴づける重要な語であるにも関わらず辞書に含まれなくなってしまう可能性がある。さらに、算出したベクトルを単純平均することで文章である論文や Web ページのベクトルを計算しているが、単語同士の重要度を考慮していないため、文章を特徴づける重要単語のベクトルが他の一般的な単語のベクトルとの加算により、影響度が薄められてしまった可能性も考えられる。単語の重要度を加味したうえでの文章ベクトル作成を行うことで性能を向上させられることが予想される。

第二に、データベースのサイズについて、提案手法である HelloProfessor Recommendation の 56783 件に対し、比較対象の CiNii Articles は約 2200 万件^{*18}である。収録件数について大きく差がつけられていることから、アルゴリズムに問題はなくともクエリとなる Web ページベクトルの近くに適切な論文が存在しない場合が考えられる。

これら性能上の要因に加え、評価用記事にフリー百科事典サイト「Wikipedia」を用いたことが従来のキーワード検索に有利に働いた可能性が考えられる。ある単語について解説する百科事典記事は、その記事のキーワードがタイトルに現れることが多いため馴染みのない分野であっても関

*18 https://support.nii.ac.jp/ja/cia/cinii_db

連する論文を検索しやすい。これは本来のユースケースである「閲覧した Web ページをきっかけとして、キーワードのわからない研究分野に関する情報を検索する」という状況を正しく反映している評価方法であるとは言えない。このような本来のユースケースにおける有用性を定量的に評価できるような評価手法を今後開発していく必要がある。

7. まとめと今後の展望

研究活動に取り組み始めた初心者が研究分野に関するキーワードを知らず適切な研究情報にアクセスできないという問題を解決することを目的とし、キーワードを用いず、ブラウジング中の Web サイトからシームレスに研究情報を検索できる情報検索システム「HelloProfessor Recommendation」を提案した。閲覧した Web ページをきっかけとして、キーワードのわからない研究分野に関する情報を検索するといったユースケースを想定する。提案システムについて速度評価と有用性評価を行ったところ、速度評価では実用的な時間内に検索を完了できることを確認した一方で、有用性についてはシステム自体または評価手法に課題が残ることが明らかとなった。

今後の展望としては、データベースの拡充や文書のベクトル空間を作成する際のアルゴリズム改善を行うことで有用性の向上に努めると同時に、検索結果の有用性を評価するためのより適切な手法を開発していきたい。論文検索についてはよりユースケースにあった評価方法を考案すると同時に、定性的な評価と組み合わせることでシステムの改善すべき点を明らかにしていきたい。さらに、今回は研究者検索の有用性評価を行うことができなかった。これは論文検索と同様に「検索結果が記事と関連しているかどうか」を評価基準とすると、検索者にはそれが判断できないためである。この問題を踏まえた上で、研究者検索の結果に対する定量的評価手法も考案していきたい。

参考文献

- [1] 酒井哲也: 情報アクセス評価方法論—検索エンジンの進歩のために—, コロナ社 (2015).
- [2] Salton, G. and Lesk, M. E.: Computer evaluation of indexing and text processing, *Journal of the ACM (JACM)*, Vol. 15, No. 1, pp. 8–36 (1968).
- [3] 大谷紀子: 情報検索におけるベクトル空間モデルの応用, 武蔵工業大学環境情報学部紀要, Vol. 5, pp. 99–109 (2004).
- [4] Mikolov, T., Chen, K., Corrado, G. S. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *CoRR*, Vol. abs/1301.3781 (2013).
- [5] 西尾泰和: word2vec による自然言語処理, オライリー・ジャパン (2014).
- [6] Song, Y. and Roth, D.: Unsupervised sparse vector densification for short text similarity, *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics (ACL), pp. 1275–1280 (2015).
- [7] 鈴木正敏, 松田耕史, 関根 聡, 岡崎直観, 乾健太郎: Wikipedia 記事に対する拡張固有表現ラベルの多重付与, 言語処理学会第 22 回年次大会, pp. 797–800 (2016).
- [8] Kudo, T., Yamamoto, K. and Matsumoto, Y.: Applying conditional random fields to Japanese morphological analysis, *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 230–237 (2004).
- [9] Toshinori, S.: Neologism dictionary based on the language resources on the Web for Mecab (2015).
- [10] 佐藤敏紀, 橋本泰一, 奥村 学: 単語分かち書き用辞書生成システム NEologd の運用— 文書分類を例にして—, 自然言語処理研究会研究報告, 情報処理学会, pp. NL-229–15 (2016).
- [11] 佐藤敏紀, 橋本泰一, 奥村 学: 単語分かち書き辞書 mecab-ipadic-NEologd の実装と情報検索における効果的な使用方法の検討, 言語処理学会第 23 回年次大会 (NLP2017), 言語処理学会, pp. NLP2017–B6–1 (2017).
- [12] 岩崎雅二郎: 商品画像検索へのグラフ構造型インデックスの適用, 画像電子学会誌, Vol. 42, No. 5, pp. 633–641 (オンライン), DOI: 10.11371/iieej.42.633 (2013).