

自己学習を振り返るための 共起グラフによる学習履歴の可視化

稲子 翔太^{1,a)} 中野 裕司^{2,b)} 杉谷 賢一^{2,c)} 久保田 真一郎^{2,d)}

概要: 我々はブラウザの拡張機能と記録サーバを組み合わせ、学習時に閲覧した Web ページから得た頻出語等を学習履歴として記録するシステムを開発してきた。しかし、集めた頻出単語等は、ダッシュボード上にキーワードとして羅列されるだけにとどまっており、自己学習の振り返りに効果的な可視化ではなかった。自己学習の振り返りにおける効果的な可視化として、知識の変化を自己認識できることが望ましいと考え、本研究ではブラウザの拡張機能により収集した Web ページ内の文章を分析し、Wikipedia の文章をもとにした共起グラフを表示し、キーワード同士の関連性や知識の変化を表現することで、学習者にとって効果的な学習履歴の可視化を目指した。

キーワード: 学習履歴, ラーニングアナリティクス, 共起グラフ, 振り返り

Visualization of Learning Records with Co-occurrence Graph to Reflect on Own Learnings

SHOTA INAGO^{1,a)} HIROSHI NAKANO^{2,b)} KENICHI SUGITANI^{2,c)} SHIN-ICHIRO KUBOTA^{2,d)}

Abstract: In our past research, we had developed a system to store learning records, consisting of frequent words extracted from the web pages, which a learner viewed in learning process. The system had been constructed from a browser extension, which sends data of the web pages to a record server, and a record server, which records the frequent words. However, it had not been effective for learners to visualize a list of the frequent words in order to reflect on their self-learning. It was considered to be important that learners recognize differences between past and current knowledge by themselves. This research showed our consideration to visualize their learning records for each learner effectively. With presenting co-occurrence graphs on the basis of data in the Wikipedia, We had developed a system to visualize relationships between keywords, and differences between past and current knowledge.

Keywords: Learning Records, Learning Analytics, Co-occurring Graph, Reflection

1. はじめに

e ラーニングの利用により、学習者は時間や場所を問

¹ 熊本大学工学部情報電気電子工学科
Department of Computer Science and Electrical Engineering, Kumamoto University

² 熊本大学総合情報統括センター
Center for Management of Information Technologies, Kumamoto University

a) c6713@st.cs.kumamoto-u.ac.jp

b) nakano@cc.kumamoto-u.ac.jp

c) sugitani@cc.kumamoto-u.ac.jp

d) kubota@cc.kumamoto-u.ac.jp

わずいつでも学習ができる環境となっている。また、LMS(Learning Management System) や e ポートフォリオシステムが利用され、それらのシステムでは学習者個人の学習履歴が大量のデータとして発生している。こうしたデータに対して、LA(Learning Analytics) のように、学習者に適した学習を提供する取り組みが行われている。LA を行うためにはできるだけ多くの学習活動の履歴を集めることが必要だが、LMS や e ポートフォリオシステムに残る学習履歴だけで必ずしも十分なデータを収集できるわけではない。また、学習に利用するツールが多様になっており、

個人の学習履歴が複数のシステムに分散して保存されているのが現状である。こうしたなかで、LMSの統合的な運用性を高めるために米国の標準化推進団体 ADL(Advanced Distributed Learning) から学習履歴に関する国際標準規格として 2013 年には xAPI[1] が、2015 年には IMSGlobal から IMS Caliper[2] が公開された。

我々は、ブラウザの拡張機能を利用し、学習者が閲覧した Web ページから得た頻出単語を、IMS Caliper の形式で学習履歴として蓄積するシステムを開発した [3]。しかし、その学習履歴の可視化は、取得した頻出単語を表示するだけで、学習者にとって意味のある表示方法とは言い難かった。本研究では、拡張機能を利用し収集した学習履歴を、自然言語処理や Wikipedia のリンク、カテゴリ構造を利用し、自己学習の振り返りや今後の学習方針を定める際の支援となる可視化を検討した。

2. 関連研究

関連研究としてまず、我々の研究室で開発した学習履歴収集 Web ブラウザアドオンについて説明する。

2.1 学習履歴収集 Web ブラウザアドオン

e ラーニングで学習を行う際は、LMS に限らず Web ブラウザで検索しながら学習する状況が考えられる。このことから、学習者が閲覧する Web ページの文章を取得し、頻出単語を学習履歴として収集するアドオンの開発を行った。スマートフォンやタブレットからも学習履歴を収集することを考慮して、モバイルアプリケーションでもアドオンが導入可能な Firefox をブラウザとして選択した。

開発されたアドオンは以下のような機能を有する。

- ブラウザのツールバーにボタンを追加
- CAS[4] のユーザー認証ページへの遷移
- 閲覧ページの URL、タイトル、HTML コードを記録サーバに送信する機能
- 学習履歴を収集するかを設定する機能

なお、現在は Firefox のみではなく Google Chrome や Microsoft Edge で動作するアドオンを開発した。

2.2 学習の振り返りの可視化

可視化による学習の振り返り支援として、山本らがマインドマップを利用した思考支援ツールを提案している [5]。マインドマップとは、Tony Buzan が提唱した思考の表現方法であり、中心にあるイメージやキーワードから放射状にキーワードやイメージを広げつなげていくものである。山本らは、知識がどのくらい身についたかを自己認識するために、学習の要所ごとにマインドマップを書くことで、知識整理の過程において知識を再構築し、学習者が次の学習に意欲的に取り組むことを確認している [6] 本研究では、閲覧した Web ページから取得した頻出単語を学習内容の

キーワードとして可視化することで、振り返りの支援を行う。また、学習を進めるほど学習履歴としてのキーワードが増えていき、可視化された学習履歴を学習前後で見比べることで、より振り返りとしての効果が期待される。

2.3 Wikipedia による学習コンテンツの発掘支援

インターネット百科事典である Wikipedia を利用し、知識のない学習領域における検索ワードのサジェストツールを横山らが提案している [7]。中島らの研究では、学習領域の知識がない学習者ほど、問題文内に出てきた単語のみを使って検索を行っている事が分かっている [8]。Web 検索を行う際に、領域知識の理解と検索キーワードの提案を兼ねて支援することで、学習者の自律的・能動的な学習態度へ変化することを目的としている。横山らの実験では、単位時間あたりの平均獲得キーワード数に有意差があることから、ツールの支援により学習の効率化に加え、学習者の好奇心や興味に影響を与え、自律的・能動的な学習を促進した可能性が示唆されている。

本研究では、学習履歴として収集した Web ページの頻出単語と Wikipedia を利用して導出した関連語を表示することで、振り返りだけではなく、今後の学習方針を決める際の支援や、学習分野に対する好奇心や興味に影響を与え、自律的・能動的な学習を促進する。

3. 学習履歴収集システム

学習履歴収集システムは図 1 のような全体構成となっている。以下に学習履歴の収集から可視化までの流れを説明する。

- (1) ユーザー認証
シングルサインオンである熊本大学統合認証システムを利用し、ユーザー認証を行う。
- (2) Web ページの閲覧
学習者はアドオンを導入した Web ブラウザで学習内容について調べる。学習履歴を残すかどうかは学習者で選択が可能となっている。
- (3) Web ページの情報を取得・送信
アドオンは、閲覧している Web ページの URL・タイトル・HTML コードを取得し、学習履歴を蓄積している可視化サーバへ送信する。
- (4) Caliper Event の作成・送信
アドオンから送信された情報を可視化サーバが受信し、HTML コードから頻出単語を抽出し、Caliper Sensor を利用して IMS Caliper 形式のデータ (Caliper Event) を OpenLRW[9] に送信する。
- (5) データベースに保存
送信された Caliper Event を受け取った OpenLRW は、データベースである MongoDB に保存する。
- (6) WebAPI 経由でデータ取得

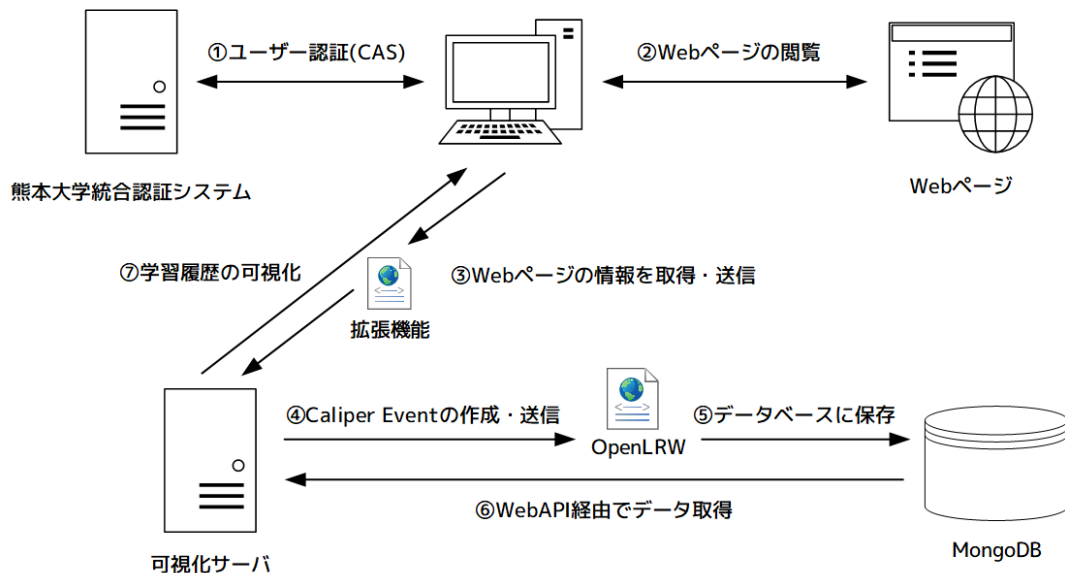


図 1 学習履歴収集システムの全体構成

可視化サーバは、データベースに保存された学習履歴を WebAPI 経由で取得する。

(7) 学習履歴の可視化

取得した学習履歴（頻出単語）に加え、Wikipedia の辞書情報をもとに抽出した関連語を同時に可視化する。

3.1 頻出単語の抽出

本システムでは、Web ページの頻出単語を学習内容のキーワードとして収集し、学習者に提示している。頻出単語の抽出方法は以下の通りである。

- (1) Web ブラウザに導入したアドオンは、HTML コードを取得し、HTML のタグ部分を取り除いた文字列を可視化サーバへ送信する
- (2) 可視化サーバで MeCab[10] を利用し、形態素解析を行い文章を単語に分ち書きする
- (3) 可視化サーバで単語の中から名詞のみを抜き出し、その他不要語を除去する
- (4) 可視化サーバでそれぞれの単語の出現回数をカウントし、上位 20 個をキーワードとして取得する

4. 学習履歴の可視化

本研究では学習履歴収集システムで収集した情報をもとに、3つの手法を用いて学習履歴の可視化を検討した。

4.1 自己組織化マップによる学習履歴の可視化

1つ目の手法として、自己組織化マップ (SOM)[11] による可視化を行った。SOM はニューラルネットワークの一種であり、大脳皮質の視覚野をモデル化したものである。教師なし学習によって入力ベクトルを任意の次元へ変換することができ、主に 1~3 次元への変換に用いられる。出力

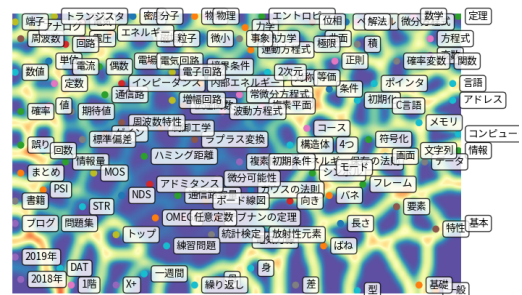


図 2 キーワードの自己組織化マップ (ラベルあり)

画像は、近いベクトルを持つノードがクラスターを形成するした画像となる。

可視化を行う学習履歴には執筆者が過去の同時期に受講していた 10 個の授業を仮想的に設定し、それらの授業に関する Web ページを閲覧して学習履歴収集システムに記録されたデータを利用した。今回使用した学習項目は、電磁気学、物理学、情報理論、電気回路、複素関数、微分方程式、制御工学、確率統計、アナログ回路、C 言語の 10 個である。

学習項目ごとに収集したキーワードを、単語をベクトル化する手法である Word2Vec[12] を用いて 100 次元ベクトル化し、SOM を用いて 2 次元画像に可視化したものが図 2 である。SOM は Python の somoclu[13] を使用した。

プロットされているノードの色は、学習項目ごとに分けられている。マップの色については、近傍ノードとの差が大きいほど赤色に近づくため、近傍ノードが異なる学習項目の場合は黄色や赤の境界線でクラスタリングされているのが理想である。しかし境界線が薄い左下の部分では異なる学習項目が混在し、逆に、境界線が多く見られる部分でも同じ学習項目が隣り合っている場合も見受けられる。これらの原因として以下のことが挙げられる。

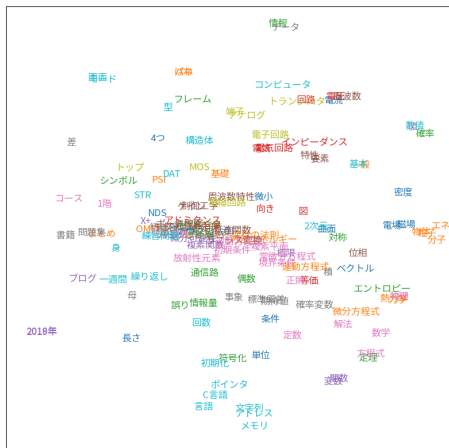


図 3 t-SNE での学習履歴の可視化 (perplexity=30)

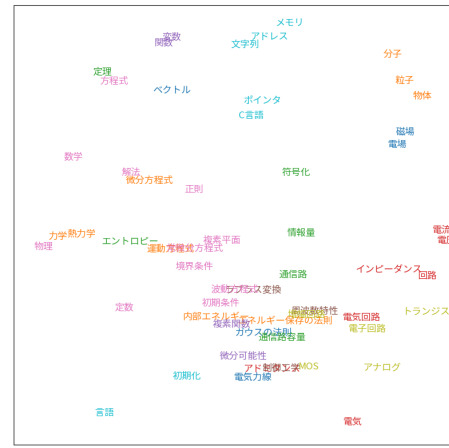


図 4 キーワード整理後の
t-SNE での学習履歴の可視化 (perplexity=30)

- キーワードが多い
- 同じ学習項目のキーワード全てが似たベクトルになることは少なく、別の学習項目のキーワードのベクトルに近いものが含まれている
- 100 次元の単語ベクトルでは単純な分類が難しい

また、今回の学習項目は実際に同時期に受けていた授業に沿ったものを用意しており、他の学生の場合でもこのような結果になる可能性は大きいと考えられる。

以上のことを踏まえると、SOM での可視化はあまり実用的ではないと考えた。

4.2 t 分布型確率的近傍埋め込み法 (t-SNE)

2 つ目の可視化手法として t 分布型確率的近傍埋め込み法 (t-SNE) を用いた可視化を検討した。t-SNE[14] とは可視化のための機械学習アルゴリズムであり、高次元データを可視化するために次元削除を行う非線形次元削除手法である。高次元を低次元空間へ配置する際に、高い確率で類似したデータが近傍に、異なるデータが遠方となるよう対応付けることでデータをクラスタリングすることができる。t-SNE のパラメーターには、コスト関数のパラメーターである perplexity などを与える。perplexity は利用するデータにもよるが、5 から 50 までの値を用いることが多く、与える値が小さいとデータ同士が密集しやすく、逆に値が大きくとデータ同士の間隔が大きくなる。

次に SOM と同じく、Word2Vec で 100 次元ベクトル化したキーワードを利用し、perplexity をおおよそ中間である 30 に設定した t-SNE で可視化した (図 3)。t-SNE は Python の Multicore-TSNE[15] を使用した。

図 3 を見ると、キーワードの量が多く、また学習履歴としてあまり意味のないキーワードも含まれていた。そのため、Word2Vec を用いて学習項目とキーワード同士の類似度を求め、類似度が 0.6 以上のキーワードのみを用いることでキーワードの選別を行った。図 4 がその結果である。

図 4 可視化画像の文字色はそれぞれ学習項目で分けられ

ている。図 4 を見ると、ある程度同じ学習項目のキーワードがまとまって表示されていることが分かる。また、情報理論のキーワードとして“エントロピー”が取得されたが、物理学などでも使われているため、物理学や微分方程式のキーワードが多い位置にプロットされている。同様に複素関数のキーワードである“関数”が C 言語のキーワードの近くにプロットされている。このように、同じ単語でもそれぞれの分野で異なる使われ方をしているキーワードに気づくことができる。

一方で、SOM や t-SNE は高次元データが必要であり、キーワードを利用するためには Word2Vec を用いて単語ベクトルを作成しなければならない。そのため、どちらの手法もキーワードのベクトルが共通となってしまう、複数の学習項目で取得されたキーワードも同じ位置にプロットされてしまい、画像では 1 つの学習項目のキーワードとして可視化される問題が起きる。また、既に学習項目でクラスタリングされたキーワードを用いるため、類似したキーワード同士がクラスタを形成する可視化の必要性はない。可視化による振り返りの支援においても、キーワードを表示することで学んだことの確認や整理は行えるが、振り返りには学んだことの確認だけでなく次の学びへ繋げる意味もある。しかし、学習してきたキーワードのみを表示しても次の学習へ繋げる効果はあまり望めない。

以上のことから、SOM や t-SNE を用いた手法は、今回の可視化にはあまり向いていないと考える。

4.3 Wikipedia を用いた共起グラフによる可視化

次元削除によるキーワードの可視化を検討した結果、学習履歴のキーワードのみを利用した可視化では、次の学びへ繋げる支援は難しいことが分かった。そこで、キーワードと同時に今後の学習で出てくる可能性がある単語を表示させることで、振り返りのみならず今後の学習を支援することができると考え、キーワードの関連語の利用を検討

表 1 Wikipedia の記事データベース

コレクション	フィールド
記事データ	記事タイトル
	所属カテゴリ
	転送先
	リンク記事 被リンク記事
カテゴリ	カテゴリ名 上位カテゴリ 下位カテゴリ

した。

4.3.1 関連語の導出

関連後の導出には、オントロジー構築や自然言語処理のデータとしても利用されることが多い Wikipedia を利用した。Wikipedia は、ウィキメディア財団が運営しているインターネット百科事典であり、2020年2月15日の時点で、全世界で309言語、52,088,393の記事が登録されている[16]。関連語の導出で Wikipedia を利用する利点として以下のことが挙げられる。

- 日本語版のみの場合でも2020年2月20日現在、1,190,829記事と、情報量が豊富[17]
- 記事のタイトルが概念となっており、そのままキーワードにできる
- カテゴリは階層構造になっており、関連語の分類が容易
- ハイパーリンクにより記事同士が繋がっているため、関連性が分かる
- DBpedia といった利用しやすいデータを提供するプロジェクト

本研究では DBpedia[18] から提供されているデータを利用し、各記事のタイトルや所属カテゴリ、リンク先の記事などを保存したデータベースを作成した。表1にデータベースの構成を示す。なお DBpedia から提供されているデータは2016年4月7日時点の Wikipedia 日本語版を対象としたものであり、データベースには MongoDB を利用している。

データベースに保存したリンク情報とカテゴリ情報を利用し、取得したキーワードや学習項目の関連語を導出する。以下に「電気回路」を学習項目として設定した場合を例に、関連語の導出方法を説明する。

(1) 学習項目に該当する記事の選択

学習項目である「電気回路」をタイトルとする記事を選択する。この学習項目は予め学習者が選択しているものとなる。

(2) 学習項目のカテゴリの取得

学習項目に該当する記事が所属するカテゴリに加え、それぞれのカテゴリの下位カテゴリを加えたものを学習項目のカテゴリとする。「電気回路」ならば、「電気

回路」「電気工学」「電気理論」が所属している3つカテゴリに加え、それぞれのカテゴリの下位カテゴリ「電気理論」「制御工学」など21個のカテゴリが加わったものが学習項目のカテゴリとなる。

(3) 学習項目の関連語取得

次に学習項目の記事と相互にリンクがある記事を取得し、さらに相互リンク先の記事と相互リンクする記事を取得する。相互リンク先の記事の取得はこの2回のみとなる。

(4) カテゴリによる関連語の削除

取得された関連語候補の記事から、学習項目のカテゴリに属していない記事を削除する。

こうして得られた関連語を共起グラフで描写する。

4.3.2 共起グラフによる可視化

共起グラフとは、同一文章内での単語の関係性を表現する際に利用されるグラフであり、共通で出現する単語同士を線で結び、単語の頻出度をノードの大きさで表す。学習履歴の可視化では、Wikipedia 内でのリンクの関係や、他の記事とのリンクの数で表現する。

学習履歴を含めない学習項目と関連語のみを表示した図が初期画像となる。

出力される共起グラフは以下のような設定となっている。

- ラベル：記事のタイトル
- 辺：記事同士のリンク
- 緑ノード：学習項目ノード
- ノード濃淡：他ノードとのリンク数

こうして得られた関連語に学習履歴として収集したキーワードを加えることで、学習の振り返りとともに今後の学習の支援を目的とする可視化を目指した。

学習履歴であるキーワードを共起グラフに追加する。キーワードは、そのままでは Wikipedia の記事と一致するものがない場合がある。これを避けるために Wikipedia の検索機能を用いてキーワードの正規化を行っている。

(1) キーワードで検索し、文字列が一致する記事があればそのままキーワードを使用

(2) 一致する記事がない場合、候補記事の上位10個を取得

(3) 文字列の一致割合が7割以上かつ、その記事が学習項目のカテゴリに属していればその記事を使用

なお、この処理には時間がかかったため学習履歴を保存する過程でキーワードを正規化することになっている。

また、関連語と学習履歴であるキーワードの差別化や、学習度合いを表現するためにノードに以下のような処理を行った。

- 学習履歴として得られたキーワードを赤色ノードで追加し、キーワードの関連語を取得。すでにノードがある場合は、ノードの色を赤に変更
- 学習度合いを表現するために、キーワードが出現するたびにノードが大きくなる

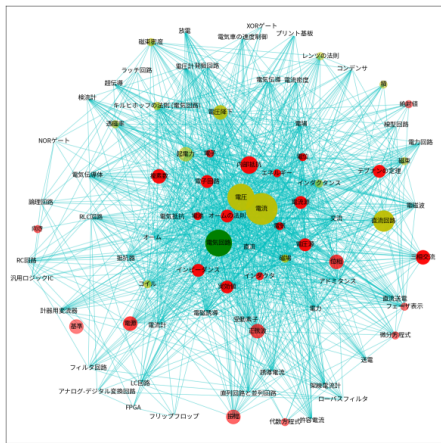


図 5 電気回路を学習項目とした時の学習履歴の可視化画像

- 最近学習したキーワードは黄色ノードに変更

また、キーワードはすべて利用するわけではなく、関連語と同じくカテゴリで選別を行っている。なお、キーワードのカテゴリ選別の際は、利用しているカテゴリは学習項目のカテゴリに加え、得られたキーワードに該当する記事が所属するカテゴリ全ての出現回数をカウントし、その上位 10% のカテゴリを採用している。これは、学習項目のカテゴリのみでは他分野のキーワード、特に基礎科目である数学などのキーワードを他の分野で拾えないことがあるためである。

なお、ノード数が多すぎると見づらくなるため、他の記事とのリンク数でソートし、上位 100 個のノードを利用する。なお、100 個目と同じリンク数を持つノードも利用しているため、実際のノード数は 100 個を少し超えた数となる。

上記の処理を施して学習履歴の可視化を行ったものが図 5 である。

SOM や t-SNE と比較すると、ノードの色や大きさに意味をもたせることで様々な表現が可能となり、画像に表現できている情報が豊富である。また、Wikipedia によって導出した関連語を同時に表示することで、学習していないキーワードなどが分かり、次の学習へ繋がられる可視化と考えられる。

5. 共起グラフによる可視化手法の検証

まず、Wikipedia の姉妹プロジェクトであり、オープンコンテンツである WikiBooks[19] を閲覧し、講義などの体系的な学習を行った際の可視化を検証した。今回学習履歴として、WikiBooks の「線型代数学」の各項目を順番に閲覧した際に取得した学習履歴を利用した。図 6 は学習前の関連語のみを可視化した初期画像である。

次に最初の項目である「序論」、中間にある項目「行列式」、最後の項目「行列の対角化」のそれぞれを閲覧した学習履歴の可視化画像を図 7~9 に示す。

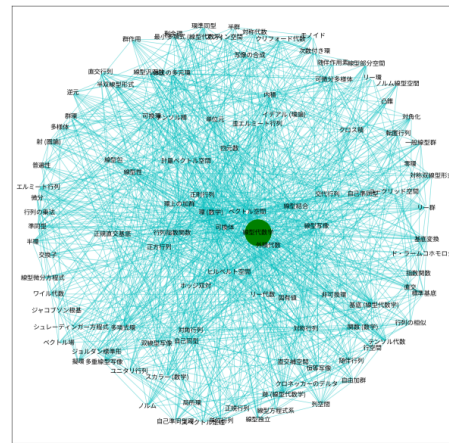


図 6 線型代数学を学習項目とした時の初期画像

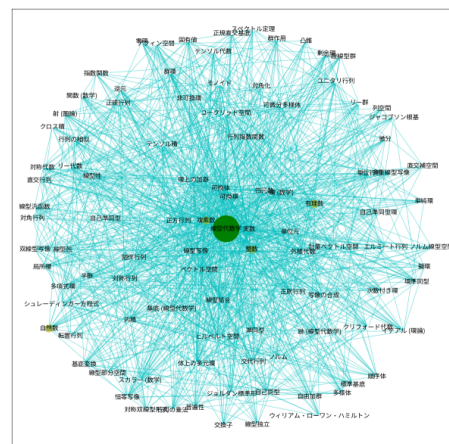


図 7 「序論」を閲覧した後の学習履歴の可視化画像

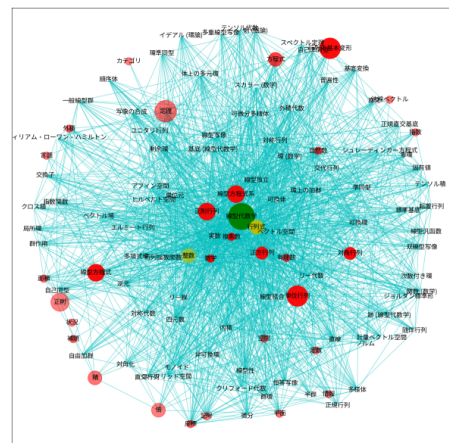


図 8 「行列式」を閲覧した後の学習履歴の可視化

次に、Web ページのブラウジングにおいて、気になったワードを調べていくことを繰り返すような体系的でない学習を想定し、学習項目に関する Web ページをランダムに閲覧し学習履歴の可視化を検証した。また、こうしたワードを調べていくような学習は、単語を覚えることが多い科目となっていることから、学習項目には高校地理で学習する「地形」を選択した。

閲覧したページは以下の通りとなっている。

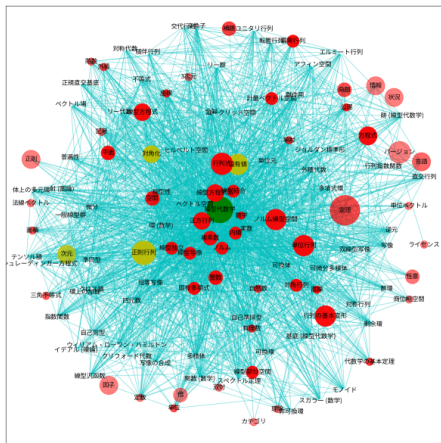


図 9 「行列の対角化」を閲覧した後の学習履歴の可視化

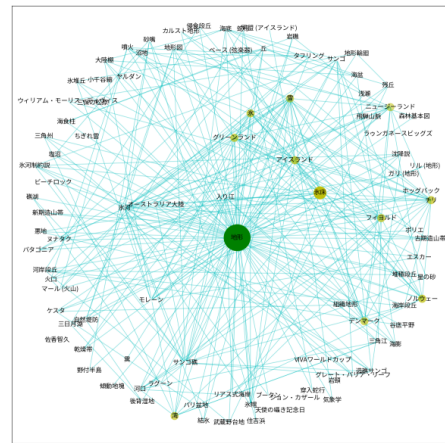


図 11 「フィヨルド - Wikipedia」を閲覧した後の学習履歴の可視化画像

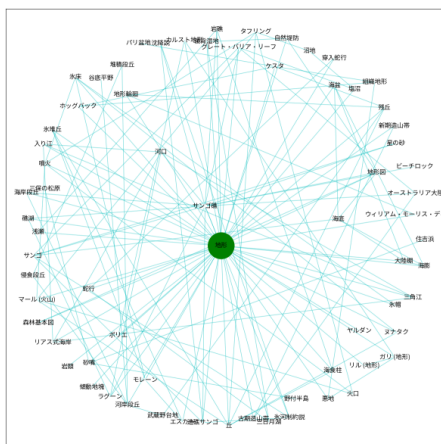


図 10 地形を学習項目とした時の初期画像

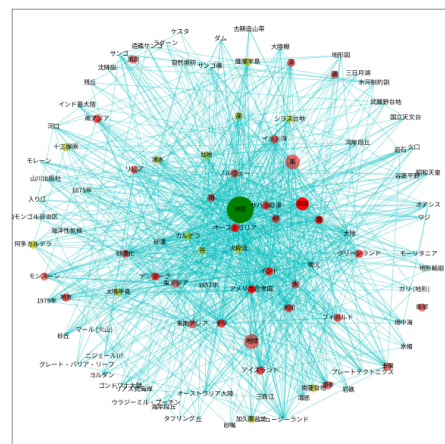


図 12 「シラス台地 - Wikipedia」を閲覧した後の学習履歴の可視化

- (1) フィヨルド - Wikipedia
- (2) 砂漠 - Wikipedia
- (3) モンスーン - Wikipedia
- (4) シラス台地 - Wikipedia
- (5) ワジ - Wikipedia
- (6) カルデラ - Wikipedia
- (7) 楯状火山 - Wikipedia

図 10 は学習前の初期画像である。

次に最初のページ「フィヨルド - Wikipedia」を図 11 に、中間の「シラス台地 - Wikipedia」を図 12 に、最後に閲覧した「楯状火山 - Wikipedia」を図 13 に示す。

6. 考察

「線型代数学」のような体系的な学習における可視化は、初期画像で表示されているキーワードの多くが徐々に赤ノードに変化している事がわかる。また、多くの赤ノードが次項目でもキーワードとして現れる事が多く、徐々に大きくなっているノードが確認できる。これより、体系的な学習における可視化は、学習前後のノードの色や大きさの変化を見ることで、学習の振り返りに有効的な可視化であると考えられる。また、学習が進むにつれ赤ノードが増

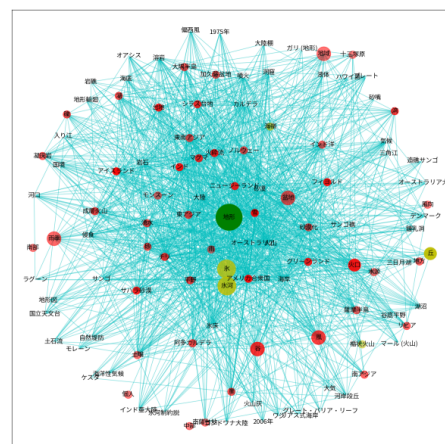


図 13 「楯状火山 - Wikipedia」を閲覧した後の学習履歴の可視化

え、大きくなっていくことから、学習者のモチベーションに影響を与えることも期待できる。

用語を覚える機会が多い地理の「地形」を学習項目とした場合、未学習である透明ノードのキーワードを確認すると、高等学校の地理で学習する単語などが多く見られる。地理のような暗記を必要とする科目においては、ノードのラベルとノード同士の繋がりを見ることで知識整理の支

援となる。このように、学習したキーワードと関連語を同時に表示することは、暗記を要する学習において特に有効的であると考えられる。また「線型代数学」と比べると、「地形」の赤ノードの大きさが比較的小さいことが分かる。これは、閲覧したページが「地形」に関することではあるが、適当にページを閲覧したことで、前後の学習とは関わりが少ない学習を行ったことで、キーワードがあまり被らなかったためと推測できる。

一方で、関連語の導出に Wikipedia を利用していることから、Wikipedia の情報が乏しい学習項目ではあまり効果のある学習履歴の可視化は行えないという欠点があり、これは今後改善すべき課題である。

7. まとめ

本研究では、Web ブラウザ拡張のアドオンで取得したキーワードと、Wikipedia を用いて導出した関連語を同時に表示させることで、学習の振り返りに効果的な可視化を検討した。結果として、学習前後の知識の変化をノードの色や大きさの変化で可視化し、学習で得た知識を再認識することが可能となり、振り返りを支援する有効な可視化となった。また、まだ学習をしていない関連語を同時に可視化することで、今後の学習の指針となったり、繋がっているキーワードや関連語同士の関係性を考察することで、知識の整理に役立つことが期待できる。しかし、ノードを繋ぐ辺が多くグラフが見にくいという欠点があり、ノードの繋がりを考察するのは難しいことが分かる。今後は辺で繋ぐノードの条件などを設定し、キーワード同士の関係性が分かりやすい可視化を目指す。

謝辞 本研究は JSPS 科研費 15H02795 の助成を受けたものです。

参考文献

- [1] Experience xapi — adl initiative. <https://adlnet.gov/projects/xapi/>. (Accessed on 02/20/2020).
- [2] Caliper analytics — ims global learning consortium. <https://www.imsglobal.org/activity/caliper>. (Accessed on 02/20/2020).
- [3] 横原竜之輔, 久保田真一郎, 杉谷賢一, 中野裕司. 自己調整学習の支援を目的としたダッシュボードの開発研究—国際標準規格 ims caliper に基づく実装—. Technical Report 22, 熊本大学大学院自然科学研究科, 熊本大学総合情報統括センター, 熊本大学総合情報統括センター, 熊本大学総合情報統括センター, mar 2018.
- [4] Cas — apereo. <https://www.apereo.org/projects/cas>. (Accessed on 02/20/2020).
- [5] 山本利一, 鈴木航平, 北畠謙太郎, 本郷健. 学習の振り返りを可視化する「思考支援ツール」の活用と評価. 教育情報研究, Vol. 33, No. 3, pp. 35–42, 2018.
- [6] 山本利一, 大関拓也, 五百井俊宏. マインドマップを活用した生徒の思考整理を支援する指導過程の提案. 教育情報研究, Vol. 24, No. 3, pp. 23–29, 2009.
- [7] 横山誠, 村上貴彦, 中村太戯留, 上林憲行. Wikipedia からキーワードを援用し初出の知識の学習と学習コンテンツの発掘を支援するサービスの研究. 第 77 回全国大会講演論文集, 第 2015 巻, pp. 701–702, mar 2015.
- [8] 中島悠, 土方嘉徳, 西田正吾. 検索経験と領域知識の www 情報検索行動に与える影響. ヒューマンインタフェース学会論文誌, Vol. 7, No. 2, pp. 309–319, may 2005.
- [9] Shuhari — apereo. <https://www.apereo.org/projects/shuhari-new>. (Accessed on 02/20/2020).
- [10] Mecab: Yet another part-of-speech and morphological analyzer. <http://taku910.github.io/mecab/>. (Accessed on 02/20/2020).
- [11] David J Willshaw and Christoph Von Der Malsburg. How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, Vol. 194, No. 1117, pp. 431–445, 1976.
- [12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [13] Github - peterwittek/somoclu. <https://github.com/peterwittek/somoclu>. (Accessed on 02/20/2020).
- [14] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, Vol. 9, pp. 2579–2605, 2008.
- [15] Github - dmitryulyanov/multicore-tsne. <https://github.com/DmitryUlyanov/Multicore-TSNE>. (Accessed on 02/20/2020).
- [16] Wikipedia:全言語版の統計 - wikipedia. <https://ja.wikipedia.org/wiki/Wikipedia:%E5%85%A8%E8%A8%80%E8%AA%9E%E7%89%88%E3%81%AE%E7%B5%B1%E8%A8%88>. (Accessed on 02/20/2020).
- [17] Wikipedia:日本語版の統計 - wikipedia. <https://ja.wikipedia.org/wiki/Wikipedia:%E6%97%A5%E6%9C%AC%E8%AA%9E%E7%89%88%E3%81%AE%E7%B5%B1%E8%A8%88>. (Accessed on 02/20/2020).
- [18] Dbpedia japanese. <http://ja.dbpedia.org/>. (Accessed on 02/20/2020).
- [19] Wikibooks. <https://ja.wikibooks.org/wiki/%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%9A%E3%83%BC%E3%82%B8>. (Accessed on 02/20/2020).