

履修支援のための Doc2Vec を用いた科目推薦システム

竹森 汰智¹ 亀井 清華^{2,a)}

受付日 2019年3月1日, 採録日 2019年6月19日

概要: 大学において学生が時間割を作成する際には、シラバスや学生便覧を参照する。しかし、これらに掲載されているたくさんの科目情報の中から、自分が履修したいと思うような科目を見つけることは容易ではない。本研究では、学部新入生を対象に、教養科目を推薦することで履修支援を行うシステムを構築する。提案システムでは、ユーザとのインタラクションによってユーザの学びの志向を取得し、推薦対象の科目集合を絞り込んでいくことを考える。Doc2Vecによって、各科目のシラバスに掲載されている科目情報の分散表現を獲得し、得られた分散表現に基づいてクラスタリングを行い、各クラスターに属する科目群の特徴を可視化する。それらを選択肢としてユーザに提示し、興味のあるものを選択してもらい、必要に応じてキーワードを入力してもらう。これにより、ユーザの学びの志向とマッチする科目を選択しやすいように支援する。本論文では、提案システムのプロトタイプシステムを構築し、アンケートによって評価した。評価結果から、提案システムが推薦精度の向上とユーザの負荷の軽減に有効であることを確認した。

キーワード: 推薦システム, Doc2Vec, ワードクラウド, 階層クラスタリング

Course Recommendation System Using Doc2Vec

DAICHI TAKEMORI¹ SAYAKA KAMEI^{2,a)}

Received: March 1, 2019, Accepted: June 19, 2019

Abstract: At universities, when a student registers for classes, he/she has to refer to the syllabus and student handbook distributed by his/her university, but it is not easy to find classes that he/she would like to learn from the large amount of information about classes. For this reason, in this research, we will construct a system that recommends liberal arts classes for undergraduate freshmen. The proposed system gains information about what the user wishes to learn through interactions with the user, and narrows down the set of classes for recommendation. By using Doc2Vec, the system acquires a distributed representation for the information about each class in the syllabus, and performs clustering based on the obtained distributed representations. In addition, it visualizes the features of each cluster and presents them to the user as options. After that, the system prompts the user to select some clusters of interest from among them and asks the user to enter some keywords in which he/she is interested as necessary. By doing this, it helps to make it easier for users to select the classes that match their learning preferences. In this paper, we evaluated the proposed system by means of a questionnaire survey. From the evaluation results, we confirmed that the proposed system is effective for improving the accuracy of recommendations and reducing the burden on users.

Keywords: recommendation system, Doc2Vec, word cloud, hierarchical clustering

1. はじめに

近年、Web 上には多くの情報が溢れており、ユーザが必要とする情報を取捨選択するコストも高くなっている。それにともない、対象とするユーザの好みに合わせて商品やサービスを推薦する手法についての研究 [2], [3], [4], [5] がさかんに行われている。推薦システムが対象とするアイテ

¹ 株式会社両備システムイノベーションズ
Ryobi System Innovations Co., Ltd., Okayama 700-0826
Japan

² 広島大学大学院工学研究科
Graduate School of Engineering, Hiroshima University,
Higashi-hiroshima, Hiroshima 739-8527 Japan

a) s-kamei@se.hiroshima-u.ac.jp

ムは、家電や映画、本、音楽、動画、レストラン、ニュースなど多様であり、我々の日常生活においても推薦システムの恩恵を受ける場面が多々見られる。たとえば、Amazon [6] では、ユーザが閲覧している商品に関連している商品を推薦したり、そのユーザと嗜好が似ている他のユーザの購入履歴を考慮した商品を推薦している。しかし、このような推薦方法では、ユーザが閲覧している商品が本当に興味のある商品か分からないため、適切な推薦を行うことは難しい。また、新規ユーザについては嗜好に関する情報がないために推薦を行うことができない場合もある。このように、ユーザの嗜好を取得する方法や、アイテムを推薦する手法は様々提案されているが、課題も多く残されている。

一方で、我々が推薦システムを必要と感じる場面は多く、その1つとして大学の履修登録作業があげられる。大学では、学生自身が時間割を作成しなければならず、学生自身が所属する学部や学科で定められた履修条件や時間制約を満たせば、興味のある学問分野の講義を自由に履修することができる。学生が時間割を作成する際には、大学で配布されるシラバスや学生便覧を参照するが、これらに掲載されている科目情報は膨大である。そのため、自分が興味のある学問分野の科目を探し、履修したいと思うようなものを見つけることは容易ではない。特に学部新入生にとって、それは負荷の大きい作業であるといえる。

履修科目を推薦するシステムは、e-learning システムを対象にさかんに研究されており、これらは過去の履修者の履歴や成績などを基に推薦している [7], [8]。これらのシステムにおいては、ユーザは学びながら自身の興味に従って、次に何を学びたいのかを選択する。その際、これまで学んだことを掘り下げたり、あるいは関連するトピックから視野を広げたりを繰り返すという自由がある。しかしながら大学では、学生は限られた履修登録期間のうちに、その開講時期に履修したいすべての科目をあらかじめ登録する必要があり、受講しながら追加履修することは認められていない。加えて、以下の理由により、大学の履修登録作業のために既存の e-learning システムのための技術を適用することは困難であるといえる。

- 対象となる科目の講義内容は、演習やグループワークなどを含むことも多く、シラバスのような文書データ以外の形で講義内容をデータ化することが難しい。
- 対象が学部新入生である場合、対象としている科目間には（多少の例外はあるが）推奨される履修順序関係などはほとんど存在しない。
- 対象となる学生数自体が e-learning システムに比べると非常に少なく、対象となる科目集合も毎年更新されるため、科目推薦に有用な履歴（受講履歴や成績、アクセス履歴など）を確保したり、統計などを使った手法を適用したりすることが難しい。

さらに、既存の e-learning システムと大学の履修登録作業

との共通の課題として、新規ユーザが最初に学ぶための科目推薦については、これらの既存手法では解決されていない。

そこで、本研究では、学部新入生を対象に教養科目を推薦するシステムの構築を行う。学部新入生は新規ユーザであると考えられるため、提案システムでは、システムとユーザとのインタラクションによってユーザの学びの志向を取得し、推薦対象の科目（以下、候補科目と呼ぶ）の集合を絞り込んでいくことを考える。提案システムでは、doc2vec [1] によって、各科目のシラバスに掲載されている科目情報の分散表現を獲得する。そして、得られた分散表現に基づいて候補科目のクラスタリングを行い、各クラスターに属する科目集合の特徴を可視化する。それらを選択肢としてユーザに提示し、興味のあるものを選択してもらい、必要に応じてキーワードを入力してもらい、また、高校科目に対するユーザの好みも入力してもらい、各教養科目との類似度を用いて候補科目のランキングを行う。これにより、ユーザの学びの志向とマッチする科目を選択しやすいうように支援する。本研究では、アンケート形式の実験によって、提案手法による推薦精度の向上とユーザの負荷が軽減されていることを示す。

本論文の構成は以下のとおりである。2 章では関連研究について述べ、3 章では提案システムの詳細について述べる。4 章では提案システムについての実験評価を行い、結果について考察する。最後に、5 章でまとめと今後の展望について述べる。

2. 関連研究

近年、Massive Open Online Courses (MOOC) などの e-learning システムを対象とした研究がさかんに行われている。その MOOC に対して、Estivill-Castro ら [9] は、メタデータがない状況でもビデオコンテンツから講義内容や大学名、教員名、カテゴリなどの属性を抽出し、それらの関係を整理したデータセットを作成する方法を議論している。しかし、我々が対象としている大学での（オフラインの）講義の場合には、講義自体をデータ化することは困難である。

また、MOOC では、新たな科目の履修によって、ユーザがすでに学んだことを深く掘り下げたり、関連する他のトピックへと視野を広げたりすることを支援する研究が数多くなされている。たとえば、O'Mahony ら [10] は、科目コンテンツを小さなテーマごとに区分けした小単位で推薦することにより、様々なテーマの履修を可能としている。Zhao ら [11] は、ユーザが現在学んでいる科目のビデオコンテンツの内容を元に、それに関連するビデオコンテンツを推薦し、それらの関係を可視化することによってユーザをさらなる学びへと導こうとしている。Jing ら [12] は、ユーザによるインターネットのアクセス履歴や科目の内容、統計的な情報や科目間の関係など多くの情報を元に科目コンテンツの推薦を行っている。いずれの場合も、現在の履修科目

あるいは過去の学習履歴などを元に、ユーザが今後学ぶべき科目を推薦するものとなっている。しかしながら、我々の対象は新入生であり、履歴を入手することは困難である。また、前章で議論したように、学生は限られた履修登録期間のうちに履修したいすべての科目をあらかじめ登録する必要があるため、これらの手法を適用することは難しい。

大学の講義を対象とした既存研究には文献 [13] がある。しかし、これは高校生に対してオントロジーを用いて大学や専攻を決めるための支援を行うものであり、各大学内での講義科目を推薦できるものにはなっていない。

次に、大学における履修支援を目的としてシラバスを用いた先行研究について述べる。Yu ら [14] は、Web 上にあるシラバスを抽出することを目的として、他のページとのリンクの関係を用いて自動的に分類する手法を提案した。しかし、これはシラバスの内容については考慮されていない。Gardner ら [15] は、シラバスの各特徴を 12 段階評価で人手で表現し、その評価値を用いて学生の成績を予測するモデルを提案した。しかしこれは、シラバスの一部の項目について条件を満たすか否かに着目しており、シラバスの文章を用いたものではない。本研究ではシラバスの文章から各科目の特徴を得ることで、科目の内容を考慮したものとなっている。

堀ら [16] は、学生の考えた時間割の特徴をレーダチャートで表示するシステムを提案した。具体的には、学生が入力した時間割がカリキュラム全体でどのような特徴を持つのかをレーダチャートで表示する。また、レーダチャートから値の低い成分を補う科目や、高い成分の関連科目を推薦する。本研究では、教養科目と高校科目との類似度を表現するために、レーダチャートを用いる。

西森ら [17] は、科目間の類似度、および学生の履修履歴を用いた科目推薦システムを提案した。具体的には、TF-IDF 法 (3.2.1 項参照) によるベクトルを用いて科目どうしのコサイン類似度を算出し、科目の成績推定を行い、得意な科目と不得意な科目を推薦する。しかし、この手法では履修履歴を持たない学部新入生には適用できない。本研究では学部新入生に対する科目推薦を想定しているため、明示的に興味を取得することで、ユーザプロフィールを獲得する。

3. 提案システム

本章では、学部新入生を対象に教養科目を推薦するシステムを提案する。図 1 に、提案システムの構成の概要を示す。提案システムでは、ユーザとのインタラクションによって、候補科目を絞り込んでいくことを考える。

まず、各科目の特徴を表現するために、doc2vec [1] を用いて科目ごとのシラバスを分散表現し、得られたベクトルを科目ベクトル (3.1 節参照) と呼ぶことにする。doc2vec は word2vec [18] の応用の 1 つで、文書の分散表現を計算する手法である。得られた科目ベクトルに対して階層クラ

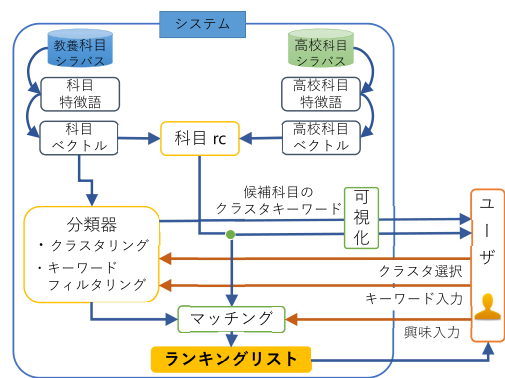


図 1 システム構成

Fig. 1 System configuration.

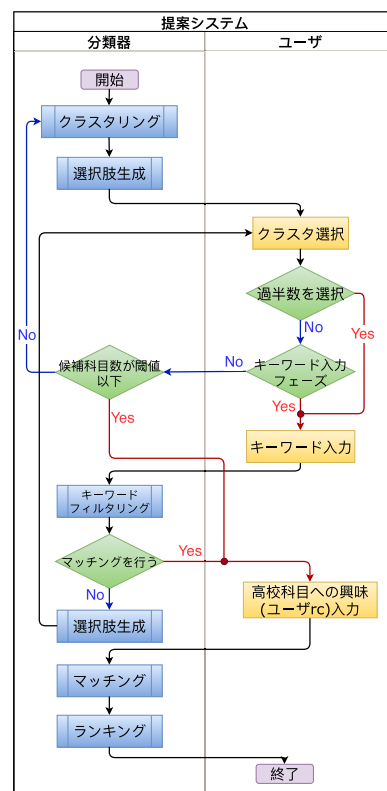


図 2 提案システムのフローチャート

Fig. 2 Flow chart of the proposed system.

スタリングを行い、クラスタごとにキーワード集合をワードクラウドで表現することで科目集合の性質を可視化する。また、高校のシラバスも科目ごとにベクトル化し、各科目ベクトルについて高校科目との類似度をレーダチャートで表現する。各教養科目についてのレーダチャートを、科目 rc と呼ぶことにする。さらに、得られた科目 rc をクラスタごとに平均し、クラスタの特徴をレーダチャートでも表現する。ユーザには、これらのワードクラウドやレーダチャートをクラスタごとに選択肢クラスタとして提示する。科目集合の特徴の可視化の方法については、3.2 節で詳しく述べる。

図 2 に、提案システムがユーザに科目を推薦するまでの流れを示す。ユーザには、システムが提示した選択肢の

中から興味のある選択肢を選んでもらう。ここで、ユーザが選択を行ってからシステムが候補科目を絞り込み、新たに選択肢をユーザに提示するまでの一連の流れを1フェーズと定義する。システムは、ユーザに選ばれた科目集合に対して、クラスタリングまたはキーワードによるフィルタリングを行うことによって候補科目数を絞っていく(3.4節参照)。このとき、ユーザが過半数の選択肢クラスタを選択したか、あるいはキーワード入力フェーズである場合に、システムは、ユーザが選んだワードクラウドの中から興味のある単語を入力するようユーザに促す。そして、ユーザによって入力されたキーワードに基づいて科目のフィルタリングを行う。その後新たに、クラスタごとに可視化して選択肢を生成し、ユーザに提示する。これを繰り返し行い、候補科目を絞り込む。最後に、高校科目に対する興味をユーザに入力してもらう。ここで入力された興味値のベクトルをユーザ rc と呼ぶことにする。システムは各候補科目の科目 rc とユーザ rc との類似度を計算する。そして、類似度が高い科目の上位5科目のランキングリストをユーザに提示する(3.3節参照)。

以下では、各段階について詳細を述べる。

3.1 科目の分散表現

まず、シラバスの科目情報を分散表現する方法について説明する。科目情報は、シラバス中の科目内容の特徴を表すと思われる項目のみに限定する。例としては、授業科目名、授業目標、授業計画などがあげられる。各科目情報から、形態素解析エンジンである MeCab [19] を用いて名詞のみを抽出し、以下の単語を除外する。

- 数
- 助数詞
- 記号
- ひらがな, カタカナ, アルファベット 1 文字

品詞によるフィルタリングを行った後に、残った単語を科目特徴語と呼ぶことにする。各科目の科目特徴語をもとに、doc2vec を用いてその科目の科目ベクトルを獲得する。doc2vec の学習には、推薦対象である教養科目のシラバスに加えて、

- Wikipedia の日本語記事 [20]
- 高校科目のシラバス
- 全学部専門科目のシラバス

を用いる。

本手法では、Wikipedia の日本語記事を利用することで、文書間の類似度の精度の向上を期待する。doc2vec の元となる word2vec は、コーパスの量が大きいほど、より良い精度になることが報告されている [21]。また、数の揃わない日本語テキストデータに対しては、オーバサンプリングなど文書データを適切にかさましする手法は確立されていない [22]。よって、精度を向上させるためには、適切な

コーパスを追加する必要がある。教養教育科目のシラバスは、学問科目の紹介文でありながら、対象が専門外の学生、特に新入生たちであることから、内容が多岐にわたっている。また、各分野の科目数や各文書の長さにもばらつきがある。そのため、特定の分野のコーパスを持ち込むよりも、Wikipedia のように多くの様々なトピックを網羅するような大規模なコーパスを導入することが望ましいと考えられる。

高校科目のシラバスを用いるのは科目 rc を獲得するためである。専門科目のシラバスを用いるのは、学問体系を網羅した空間における教養科目の科目ベクトルを得るためである。大学の講義科目について、教養科目をさらに専門に特化させた科目が各学部の専門科目である。一方で、学部生(主に1年生)が視野を広げるために、専門科目を広く浅く基礎的な内容にしたものが教養科目である。これらの関係性を考慮するために、学部の専門科目のシラバスを用いる。

3.2 科目クラスタリングとその可視化

次に、科目クラスタリングと選択肢クラスタの提示方法について説明する。得られた科目ベクトル集合に対して、ウォード法 [23] を用いてクラスタリングを行う。そして、デンドログラムと呼ばれる樹形図の根の子から順に、 k 個のクラスタを選択肢として可視化してユーザに提示する。以下では、クラスタの特徴を可視化する2つの方法について述べる。

3.2.1 ワードクラウド

各選択肢クラスタに属するすべての教養科目の特徴語を、そのクラスタの特徴語とする。各選択肢クラスタの特徴語の集合をワードクラウドとして表す。例を図3に示す。ワードクラウド [24], [25] とは、テキスト情報を可視化する手法の1つであり、文章中で出現頻度が高い単語を複数選出し、その頻度に応じた大きさで並べて図示するものである。また、文字の色や向きに変化を付けることで、文章の内容を一目で印象付けることが期待される。

ワードクラウドの作成のために、各選択肢クラスタの各特徴語に対して、TF-IDF 法で重みづけを行う。TF-IDF とは、TF (Term Frequency) と IDF (Inverse Document Frequency) の2つの指標に基づいて算出される。TF は、



図3 ワードクラウド
Fig. 3 Word cloud.

ある文書 d における、各単語 t の出現頻度を表しており、1つの文書内で多く使われる単語ほど、その文書の特徴的な単語になりうる。一方で、IDF は文書集合全体の単語 t が出現する文書数の割合の逆数を表しており、文書集合全体の中で出現する回数が少ない単語ほど、その文書の特徴的な単語になりうる。IDF は対数をとることで、文書数の規模による値の変化を小さくしている。これらは以下のように定式化される。単語 t について、 $n_{t,d}$ は文書 d における出現回数を表し、 $df(t)$ は少なくとも 1 回は出現した文書数を表す。 $\sum_{s \in d} n_{s,d}$ は文書 d 中に出現する全単語の出現回数を表し、 D は全文書数を表す。

$$tf-idf(t, d) = tf(t, d) \times idf(t) \\ = \frac{n_{t,d}}{\sum_{s \in d} n_{s,d}} \times \left(\log \left(\frac{D}{df(t)} \right) + 1 \right)$$

各選択肢クラスタの特徴語の集合をそのクラスタを表す文書であるとし、各特徴語の TF-IDF 値を計算すると、TF-IDF 値が大きい単語ほど、そのクラスタの特徴を表している単語であるといえる。ワードクラウドでは、TF-IDF 値が大きい特徴語を大きく、小さい特徴語を小さく表示する。また、過半数 ($k/2$ 以上) の選択肢クラスタで出現する単語を除去するために、IDF 値が 0.50 以下の単語を除外する。提案システムでは、選択肢クラスタの特徴語の TF-IDF 値が上位の 100 語を用いてワードクラウドを生成する。

3.2.2 レーダチャート

ここで、各教養科目の科目 rc の生成方法について説明する。高校科目は、高校の全学年の科目の中から、現代文、古典、数学、英語、物理、化学、生物、地理、世界史、日本史、公民の 11 項目に分類できるものを対象とする。このとき、たとえば、古文や漢文は古典に該当する。以下の手順で科目 rc を生成する。

- (1) 分類項目ごとに、高校科目のベクトルの平均を算出する。
- (2) 全教養科目の科目ベクトルに対し、(1) の各ベクトルとのコサイン類似度を算出する。
- (3) 分類項目ごとに、全教養科目の (2) の最小値を 1、最大値を 5 とし、正規化を行う。

こうして得られた科目 rc を用いて、ユーザに提示する選択肢クラスタのレーダチャートを生成する。分類項目ごとに、各選択肢クラスタに属する教養科目が持つ科目 rc の値の平均を算出し、レーダチャートのその項目の値とする。例を図 4 に示す。

3.3 rc マッチングによるランキング

次に、次節以降で述べる方法で候補科目を絞り込んだ後に、科目のランキングリストを生成する方法について説明する。提案システムでは、ユーザに、高校科目の 11 項目



図 4 レーダチャート

Fig. 4 Radar chart.

のそれぞれに対する興味 (ユーザ rc) を 5 段階評価で入力してもらおう。各項目に対する興味の値は、

興味がない 1--2--3--4--5 興味がある

と定義する。ユーザ rc は、そのユーザがそれまでに選択した選択肢クラスタのレーダチャートの値の平均を算出し、その値を補正してもらった形で、ユーザに入力してもらおう。そして、各候補科目が持つ科目 rc と、ユーザ rc をそれぞれ 11 次元のベクトルとし、コサイン類似度を算出する。これを rc マッチングと呼ぶことにする。そして、候補科目の中から類似度が最も高い 5 科目のランキングリストをユーザに提示する。

3.4 科目の絞り込み

本節では、ユーザが選択した選択肢クラスタに対して、候補科目を絞り込むための 2 つの方法について説明し、それらの方法の組合せパターンについて 3.4.3 項で述べる。

3.4.1 デンドログラム探索

まず、クラスタリングを用いて候補科目を絞り込む方法について説明する。提案システムでは、以下の一連の操作をデンドログラム探索と呼ぶことにする。

- (1) 現在のフェーズで対象となる候補科目集合に対してクラスタリングを行う。
- (2) 得られた樹形図を葉に向けてたどることで k 個の選択肢クラスタを得る。
- (3) 得られた k 個の選択肢クラスタを可視化し、ユーザに提示する。
- (4) ユーザが選んだ選択肢クラスタに属する科目総数に応じて、次の操作を決定する。

- 5 科目の場合： rc マッチングを行い、ランキングを決める。
- 5 科目未満の場合：選択されなかった選択肢クラスタに属する科目も含めて、 rc マッチングを行い、ランキングを決める。ただし、選択された選択肢クラスタに属する科目をランキングの上位に確定し、不足分を選択されなかった選択肢クラスタに属する科目から下位として補う。
- 6 科目以上の場合：選択された選択肢クラスタに属する科目を対象に次のフェーズの絞り込み操作へ進む。

3.4.2 キーワードフィルタリング

次に、キーワードを用いて候補科目を絞り込む方法について説明する。

システムはそのフェーズの候補科目の選択肢クラスタを可視化し、ユーザに選択させる。そして、ユーザ自身が選択した各選択肢クラスタのワードクラウドの中から興味のあるキーワードを入力させる。以下では、ユーザが選択した選択肢クラスタを**選択クラスタ**と呼び、各選択クラスタについて入力されたキーワードをそのクラスタの**選択キーワード**と呼ぶ。

各選択クラスタについて、それに属する各科目の特徴語に、その選択クラスタの選択キーワードが含まれるかどうかで、候補科目をフィルタリングする。以下では、選択キーワードが科目特徴語に含まれることを、その科目が**hit**するといひ、hitした科目を**hit 科目**と呼ぶことにする。

また、選択クラスタ数によって、フィルタリング後の処理を決める。以下では、1回目の選択キーワード入力時に、選択クラスタの数が選択肢の過半数の場合に行う処理を**分類 A**と呼び、選択クラスタの数が選択肢の半数以下の場合に行う処理を**分類 B**と呼ぶことにする。図 5 に、キーワードフィルタリングによって候補科目を絞り込むまでの流れを示す。

分類 A と分類 B とでは、候補科目の次フェーズへの活かし方が異なるため、どちらの分類を行うかは1回目の選択

キーワード入力時の選択クラスタ数のみで決まり、2回目の選択キーワード入力後は同じタイプの処理を行うものとする。また、キーワードフィルタリングを3フェーズ以上行くと、ユーザの入力の負担が増える割には、2回目の入力以降の選択肢にあまり変化が見られなかったため、キーワードフィルタリングは最大2フェーズまで行うものとする。さらに、提案システムでは、キーワードフィルタリングを行って次のフェーズに移った場合には、再度キーワードフィルタリングによって候補科目を絞り込むことにする。これは、キーワードフィルタリングを行った次のフェーズにおいて、デンドログラム探索を行ったとしても、選択肢の変化がほとんど見られなかったからである。デンドログラム探索を行うより、キーワードフィルタリングを行う方がより詳細に候補科目を絞り込むためと考えられる。

■選択クラスタ数が過半数の場合（分類 A）

● 選択キーワード入力 1 回目

全選択クラスタの hit 科目の合計数に応じて、以下の処理を行う。

– 5 科目の場合：rc マッチングを行い、ランキングを決める。

– 5 科目未満の場合：hit 科目はランキングの上位に確定し、不足分を以下から補って下位とする。

- (1) 選択クラスタに属しており、hit しなかった科目
- (2) (1) で不足する場合は、hit しなかったすべての（現在のフェーズの）候補科目

ただし、hit 科目間のランキングおよび補足分のランキングはそれぞれ rc マッチングで決める。

– 6 科目以上の場合：各選択クラスタの hit 科目のみを次のフェーズでの候補科目とする。各選択クラスタから hit しなかった科目を除いたものを選択肢として可視化する。

● 選択キーワード入力 2 回目

新たな選択クラスタのワードクラウドに、ユーザが1回目で入力した選択キーワードが含まれていた場合、その選択キーワードを入力欄に表示する。表示した選択キーワードの他に追加入力力がなければ、その選択クラスタに属するすべての科目は hit したものと扱う。新たに追加入力された選択キーワードがあれば、その選択クラスタに属する科目について、その選択キーワードを用いてフィルタリングを行う。その後、全選択クラスタの hit 科目の合計数に応じて以下の処理を行う。

– 5 科目以上の場合：rc マッチングを行い、ランキングを決める。

– 5 科目未満の場合：1回目と同様の処理を行う。

■選択クラスタ数が半数以下の場合（分類 B）

分類 A との大きな違いは、hit しなかった科目も次フェーズに用いることである。そのため、2回目の選択キーワード入力後に、1回目の選択キーワード入力でも hit した科目

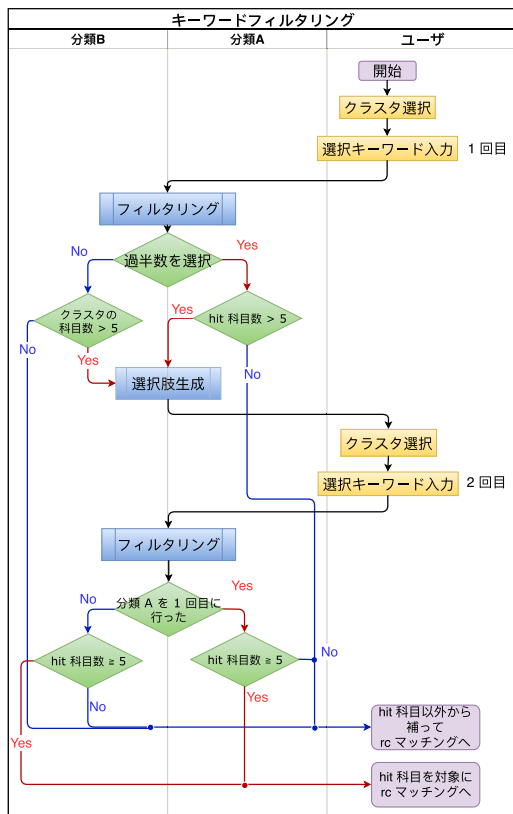


図 5 キーワードフィルタリングのフローチャート
Fig. 5 Flow chart for filtering by keywords.

に対してアドバンテージを持たせる。分類 B では、以下の処理を行う。

● 選択キーワード入力 1 回目

選択クラスタに属する科目総数に応じて、以下の処理を行う。

- 5 科目の場合：hit 科目はランキングの上位に確定し、残りを下位としてランキングを決める。ただし、hit 科目間のランキングおよび残り分のランキングはそれぞれ rc マッチングで決める。
- 5 科目未満の場合：hit 科目はランキングの上位に確定し、不足数を hit しなかったすべての候補科目から補って下位とする。ただし、hit 科目間のランキングおよび補足分のランキングはそれぞれ rc マッチングで決める。
- 6 科目以上の場合：選択クラスタに属するすべての科目を次のフェーズの候補科目とする。ただし、各選択クラスタを hit 科目とそれ以外の 2 つに分類し、それらを次フェーズの 2 つの選択肢として可視化する。このとき、ある選択クラスタに属するすべての科目が hit していた場合、その選択クラスタに対してクラスタ数 $k = 2$ として、3.4.1 項で述べたデンドログラム探索を行い、得られた 2 つのクラスタを選択肢として可視化する。こうすることで次フェーズの選択肢は 2~4 つとなる。

● 選択キーワード入力 2 回目

新たな選択クラスタが 1 回目の hit 科目集合でなかった場合、必ず選択キーワードを入力してもらい、その選択キーワードを用いて、その選択クラスタ内の科目についてフィルタリングを行う。選択クラスタが 1 回目の hit 科目集合であった場合、そのワードクラウドに、ユーザが 1 回目で入力した選択キーワードが含まれていれば、その選択キーワードを入力欄に表示する。表示した選択キーワードの他に追加入力が必要ならば、その選択クラスタに属するすべての科目は hit したものと扱ふ。追加入力があれば、その選択キーワードを用いて、その選択クラスタ内の科目についてフィルタリングを行う。

その後、全選択クラスタの hit 科目の合計数に応じて以下の処理を行う。

- 5 科目以上の場合：rc マッチングを行い、ランキングを決める。
- 5 科目未満の場合：hit 科目はランキングの上位に確定し、不足数を以下から補って下位とする。
 - (1) 1 回目の選択キーワード入力時に hit した科目 (アドバンテージ)
 - (2) (1) で不足する場合は、hit しなかったすべての (現在のフェーズの) 候補科目
 ただし、hit 科目間のランキングおよび補足分のランキングはそれぞれ rc マッチングで決める。

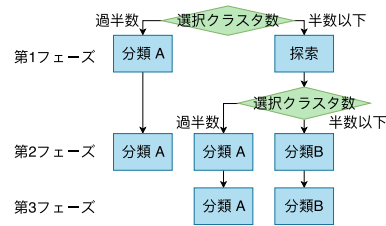


図 6 候補科目の絞り込みパターン
Fig. 6 Candidate course refinement pattern.

3.4.3 候補科目の絞り込みパターン

本項では、3.4.1 項、3.4.2 項で述べた方法を、提案システムの中で用いるタイミングについて説明する。

提案システムでは、ユーザが選択を行うフェーズの数の上限を 3 とする。これは、フェーズ数がこれ以上増えてしまうとユーザの負担が大きくなるためである。これにより、候補科目数が閾値に到達していない場合でも、絞り込みを終了して、rc マッチングに移ることができる。

今まで述べてきた制約は以下のとおりである。

- キーワードフィルタリング後の処理の種類は一貫させる。
- キーワードフィルタリングは最大 2 回までとする。
- キーワードフィルタリングを行った次のフェーズには、キーワードフィルタリングを行う。

これらに加えて、第 1 フェーズでは、選択肢クラスタの過半数が選ばれた場合にはキーワードフィルタリングおよび分類 A を、半数以下が選ばれた場合にはデンドログラム探索を行うものとした。これは、過半数の選択肢からデンドログラム探索を行ったのでは、候補科目の数がほとんど減らないためである。さらに、第 1 フェーズでデンドログラム探索を行った場合には、第 2 フェーズでは、キーワードフィルタリングを行うものとした。これは、キーワードフィルタリングを行う方がより詳細に候補科目を絞り込むからである。フェーズの上限数を 3 としたうえでこれらの制約を考慮すると、フェーズを追った選択肢の絞り込み方は図 6 の 3 パターンが考えられる。

4. 実験評価

本章では、提案システムの評価実験について述べる。以下では、プロトタイプシステムについて説明し、各実験方法や評価基準について述べる。

4.1 プロトタイプシステム

まず、構築したプロトタイプシステムについて説明する。図 7 と図 8 に、プロトタイプシステムのインタフェースを示す。図 7 は候補科目の絞り込み過程におけるインタフェースであり、図 8 はユーザ rc 入力時のインタフェースである。

科目クラスタリングのための階層クラスタリングには、

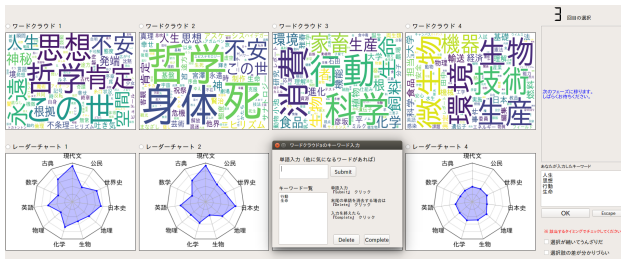


図 7 プロトタイプシステムのインターフェース
Fig. 7 Interface of the prototype system.

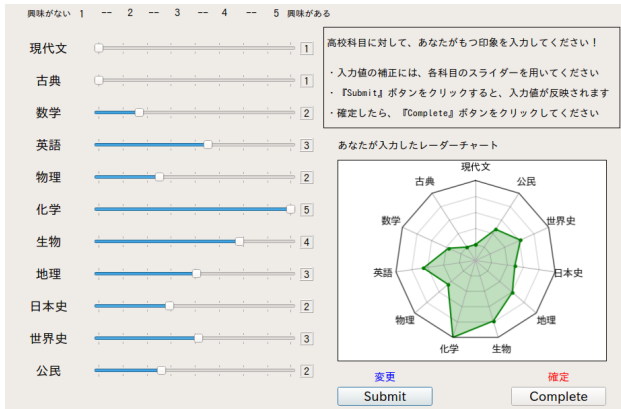


図 8 ユーザ rc の入力インターフェース
Fig. 8 Input interface for user-rc.

Python のライブラリの SciPy [26] を用いた。科目ベクトル間の距離尺度にはユークリッド距離を用い、クラスタ間の距離尺度にはウォード法を用いた。

選択肢としての各ワードクラウドを構成する単語数は、最大 100 語とした。選択肢を提示する際にはユーザに対して、同じ選択肢クラスタのワードクラウドとレーダチャートを上下に配置し、ワードクラウドは「類似した科目の集合ごとに、その科目の特徴を表すと思われる単語から構成されています」と説明し、レーダチャートは「上のワードクラウドと同じ科目の集合の特徴を、高校で学ぶ科目との類似度によるレーダチャートで表現しています」と説明した。

4.1.1 データセット

シラバスとしては、広島大学シラバス [27] と高校教科書シラバス案 [28] を利用した。

広島大学シラバスの科目情報としては、以下の項目を利用した。

- 科目名
- 授業のキーワード
- 教養教育での授業の位置づけ
- 学習の成果
- 授業の目標・概要など
- 授業の計画

4.3 節の実験では、2017 年度の広島大学シラバスから以下の科目情報を用いた。

- 教養科目 162 科目（集中講義および選択の自由がない

表 1 使用した 2018 年度科目の学問分野分類の内訳
Table 1 Breakdown of academic field classifications for 2018 courses.

分類	全科目数	教養科目	専門科目
社会人基礎	14	7	7
人文学	268	85	183
社会科学	568	53	515
理工学	530	41	489
生物・生命科学	192	21	171
語学教育	27	0	27
健康科学	17	0	17
合計	1,616	207	1,409

表 2 レーダチャートにおける高校科目の分類
Table 2 Classification of high school subjects on radar charts.

分類	該当科目
現代文	現代文
古典	古文, 漢文
数学	数学 I, A, II, B
英語	コミュニケーション英語 I, II, III, 英語表現 I, II
物理	物理
化学	化学
生物	生物
地理	地理, 地学
日本史	日本史
世界史	世界史
公民	現代社会, 倫理, 政治・経済

科目を除く)。

- 総合科学部, 文学部, 法学部, 経済学部, 理学部, 工学部, 生物生産学部の講義形式の専門科目 1,391 科目。

4.2 節, 4.4 節の実験では、2018 年度の広島大学シラバスから以下の科目情報を用いた。

- 教養科目 207 科目（集中講義および選択の自由がない科目を除く)。

- 総合科学部, 文学部, 法学部, 経済学部, 理学部, 工学部, 生物生産学部の講義形式の専門科目 1,409 科目。

これについて、広島大学シラバスの項目の 1 つである学問分野の分類の内訳を表 1 に示す。これにより、分野ごとの科目数のばらつきが非常に大きいことが分かる。また、推薦対象である教養科目について、シラバスの記述内容がほぼ同一である科目の組が多く見られた。そこで、科目間の類似度が 0.8 を超える科目の組のうちの 1 科目のみを残し、他の類似科目は除外した。そして、2017 年度のシラバスでは 128 科目を候補科目として用いた。さらに、2018 年度のシラバスでは英語で記述されたシラバスを除いたうえで、残った 166 科目を候補科目として用いた。

高校科目は、表 2 のように、22 科目を 11 項目に分類した。doc2vec の学習用の Wikipedia 日本語記事としては、2017 年 8 月時点の 1,070,383 記事を利用した。

doc2vec のモデル構築には、Python ライブラリの gensim [29] を使い、学習アルゴリズムとしては PV-DM (Paragraph Vector with Distributed Memory) モデルを用いた。学習時のパラメータは以下のとおりである。

- 次元数：300
- コンテキストの周辺単語の数：5
- 単語の低出現頻度：1
- エポック数：20

4.2 科目ベクトルの精度

まず、シラバスをベクトル化する方法として doc2vec が適切であることを示すために、最も一般的に用いられてきた TF-IDF 法によるベクトル化 (以下、TF-IDF ベクトルと呼ぶ) との比較を行う。

TF-IDF 法と doc2vec のそれぞれの方法で各科目のシラバスをベクトル化し、科目間の距離によって階層クラスタリングを行った。そして、クラスタリングの結果をデンドログラムとして可視化した。図 9 に、TF-IDF ベクトルの結果を、図 10 に、doc2vec の科目ベクトルの結果を示す。デンドログラムの横軸の数字は、科目間、クラスタ間のユークリッド距離を表す。デンドログラムの色分けは、各図の最大ユークリッド距離の 60% で設定している。

まず、クラスタ間の距離について考察する。図 9 を見

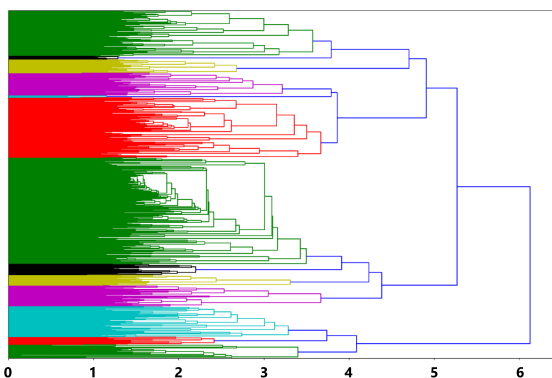


図 9 TF-IDF ベクトルを用いたデンドログラム
Fig. 9 Dendrogram using TF-IDF vectors.

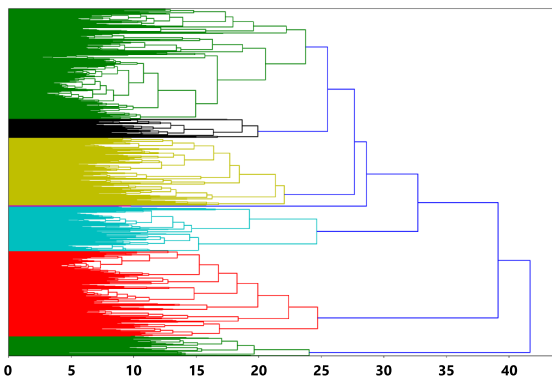


図 10 doc2vec の科目ベクトルを用いたデンドログラム
Fig. 10 Dendrogram using course vectors by doc2vec.

ると、ユークリッド距離が最大 6.0 であり、全体的に科目やクラスタ間の距離が近いことが分かる。また、形成されたクラスタに対して、近い距離にある科目が順に統合されて新たなクラスタを形成していく数珠つなぎのような形状が、特に真ん中の緑色のクラスタで多く見られる。一方で、図 10 を見ると、ユークリッド距離が最大 40 となっており、科目やクラスタ間の距離が広がっていることが分かる。このように、適度な距離感でクラスタが形成されることで、科目のまとまりや違いが把握しやすくなったといえる。したがって、doc2vec を用いて獲得した科目ベクトルに対してクラスタリングを行う方が、デンドログラム探索するシステムに適しているといえる。

4.3 可視化方法の有効性

4.3.1 ワードクラウド

3.2.1 項で述べたワードクラウドによって、科目および科目集合の特徴を表現できているかを確認するために、情報工学を専攻する学部生・大学院生 21 名に対して、アンケートを実施した。

ここでは、クラスタ数を 5 とし、システムでの第 1 フェーズの選択肢に該当する 5 つのクラスタのワードクラウドを作成した。次に、教養科目 15 科目を各参加者に対してランダムに選出した。被験者には、選出した科目のシラバスとシステムが提示するワードクラウドを見てもらい、各科目がどのクラスタに属するか回答してもらった。全被験者から得られた回答の平均正解率は 70% であった。したがって、ワードクラウドによって科目の特徴をおおむね表現できているといえる。

4.3.2 レーダチャート

3.2.2 項で述べたレーダチャートによって、科目および科目集合の特徴を表現できているかを確認するために、レーダチャートを出力し、考察をする。

まず、教養科目について、科目 rc を用いてレーダチャートを出力し、観察する。図 11 は、各分類項目の最大値を持つ科目 rc のレーダチャートである。教養科目「日本の文学」では古典のほかにも、現代文や日本史の項目が大きな数値になっていることが分かる。また、「アジアの近現代」では、世界史のほかにも、日本史や公民などの社会科に該当するような項目が大きな数値になっており、理系に該当する項目の数値は小さくなっていることが分かる。一方で、「物理の視点」では、物理のほかにも、数学や化学が大きな数値になっていることが分かる。また、「生物生産と自然との関わり」では、生物のほかにも、物理や化学など理科に該当するような項目が相対的に大きな数値になっていることが分かる。このように、各科目における科目 rc をレーダチャートとして出力することで、文系/理系のバランスがおおむね表現できていることが確認できた。

さらに、提案システムにおいてレーダチャートを提示し

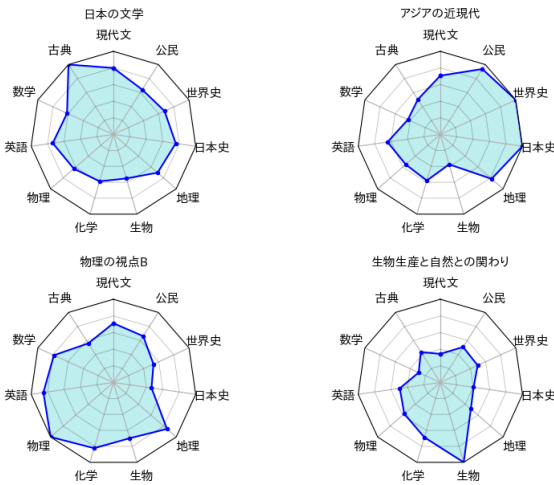


図 11 各分類項目の最大値を持つ科目 rc

Fig. 11 Course-rc with maximum value for each category.

た場合の効果についてアンケート調査を行った。4.3.1 項と同様の学生を被験者とした。

まず、システムがユーザに提示する選択肢として、ワードクラウドのみとレーダチャートのみの2種類を考える。そして、ユーザの立場から興味のあるものを選択する場合、どちらが選択しやすいか回答してもらった。被験者の81%が、レーダチャートよりもワードクラウドの方が選択しやすいと回答した。

次に、ユーザに提示する選択肢として、ワードクラウドとレーダチャートを並べる場合を考える。図7のように、ワードクラウドにレーダチャートを付加することで、クラスタに属する科目集合のイメージが分かりやすくなったかどうか回答してもらった。被験者の90%が、レーダチャートがあった方がイメージしやすいと回答した。したがって、提案システムがユーザに提示する選択肢として、ワードクラウドにレーダチャートを付加することは有効であるといえる。

4.4 提案システムの評価

最後に、提案システムの有効性の検証を行った。方法としては、プロトタイプシステムと比較システムを利用してもらい、アンケートによる比較実験を行った。これにより、システムによる推薦精度とユーザの負荷の増減について評価する。加えて、システムの使用感に関するアンケート調査も行った。プロトタイプシステムにおけるクラスタ数(最初の選択肢の数)は4とする。

本実験では、情報工学を専攻する学部4年生および大学院生19名を被験者とした。これは、新生には大学の教養科目についての知識が期待できず、また履修経験の有無により回答の確かさが異なるためである。本実験では科目内容自体についての事前知識は必要ないものの、教養科目がどのようなものなのかということについては知っておく必



図 12 比較手法 A のインターフェース

Fig. 12 Interface for comparison method A.

要がある。また、シラバスの記述の中から興味を持てる科目かどうかを判断する際には、履修経験がある被験者の方がより正確な判断が可能であると考えられる。このため、推薦精度を評価するうえでは、被験者として上級生を採用することとした。ただし、ユーザの負荷に関する評価は本来のユーザである新生に対して評価する必要があり、これについては今後の課題として残っている。

4.4.1 比較手法

比較手法を以下の4つとする。

- 比較手法 A: キーワードフィルタリングのみで候補科目を絞り込み、rc マッチングを行う
- 比較手法 B: デンドログラム探索のみで候補科目を絞り込み、rc マッチングを行う
- 比較手法 C: rc マッチングのみを行う
- 比較手法 D: ランダムに科目を選出する

以下に A~C の各手法の詳細を説明する。

まず比較手法 A は、プロトタイプシステムで最初の4つの選択肢として用いるワードクラウドを2x2で配置し、1つの大きなワードクラウドとしてユーザに提示する。このワードクラウドを構成する総単語数は400語である。ユーザには、提示したワードクラウドを眺めてもらい、気になる単語や興味がある単語をできる限り入力してもらい、そして、ユーザに入力してもらった単語を含む候補科目に対して、rc マッチングを行う。図12に、比較手法 A のインターフェースを示す。

次に、比較手法 B は、クラスタ数を4として科目クラスタリングを行い、選択肢としてのワードクラウドとレーダチャートをクラスタごとに並べて提示する。ユーザには、興味のある選択肢を選んでもらう。ただし、選択可能なクラスタの数は3つまでとする。選ばれたクラスタのみに対して、クラスタ数を4として、再度科目クラスタリングを行う。そして、クラスタごとに選択肢を提示し、ユーザに興味のある選択肢を選んでもらう。これを繰り返し行い、候補科目数が5科目以下になるまで絞り込んでいき、3.4.1 項と同様にランキングリストを選出する。図13に、

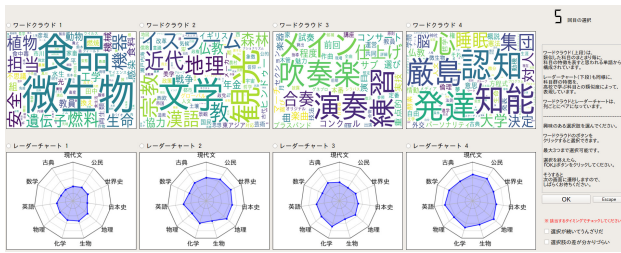


図 13 比較手法 B のインタフェース

Fig. 13 Interface for comparison method B.

比較手法 B のインタフェースを示す。

最後に、比較手法 C では rc マッチングのみを行う。候補科目を絞り込むことなく、すべての候補科目を対象として rc マッチングを行い、上位 5 科目のランキングリストを選出する。

4.4.2 実験手順

まず、被験者には以下の順でシステムを利用して、クラスタ選択操作とキーワード入力操作のみを行ってもらう。

- 比較手法 A
- プロトタイプシステム
- 比較手法 C (ユーザ rc の入力)
- 比較手法 B

このとき、被験者の興味を一貫させようとしてシステムを利用してもらうために、ユーザ rc は比較手法 C の操作として入力してもらい、それをすべての手法で共通して用いることにする。また、プロトタイプシステムの最初の選択肢の各ワードクラウドに対して、比較手法 A で入力されたキーワードを含む場合は、そのクラスタを選択クラスタとして扱い、そのキーワードを選択キーワードとして扱う。また、比較手法 B についても、そのようなクラスタの数が 3 以下の場合、それらが選択されたと見なすことにする。

そして、すべての選択操作を行ってもらった後、各手法で 5 科目ずつの推薦リストを作成し、得られた推薦リストを重複を除いてランダムにマージしたうえで被験者に提示する。被験者には、提示したリスト中の各科目のシラバスを読んでもらい、履修したいかどうかを回答してもらう。被験者に推薦リスト中の各科目のシラバスを読んでもらう際には、以下のことを依頼した。

- 科目名が同じだとしても異なる科目であるため、リスト中のすべての科目に目を通すこと。
- シラバス中の科目内容を確認してもらうために、以下の項目を読むこと。
 - 授業のキーワード
 - 教養教育でのこの授業の位置づけ
 - 学習の成果
 - 授業の目標・概要など
 - 授業計画

4.4.3 評価指標

■推薦精度

本実験では、推薦精度の評価のために以下の評価指標を用いる。ここで、 $hits_u$ を被験者 u が好んだアイテム (適合アイテム) の集合、 U を全被験者の集合と定義する。

- $Precision@5$: 各被験者に対する推薦リストにおける適合アイテム数の割合の、全被験者の平均値。

$$Precision@5 = \frac{1}{|U|} \sum_{u \in U} \frac{|hits_u|}{5}$$

- $hitrate$: 適合アイテムが少なくとも 1 つは推薦された被験者の割合。

$$hitrate = \frac{1}{|U|} \sum_{u \in U} hitrate(u)$$

$$hitrate(u) = \begin{cases} 1 & (\text{if } |hits_u| > 0) \\ 0 & (\text{otherwise}) \end{cases}$$

- $Mean\ Average\ Precision\ (MAP)$: ある被験者に対する推薦リストにおいて、適合アイテムが出現した時点それぞれ閾値として、各閾値における $Precision$ を算出し、平均を求める。その値の全被験者の平均値。

$$MAP = \frac{1}{|U|} \sum_{u \in U} Avep(u)$$

$$Avep(u) = \frac{\sum_{k=1}^5 Precision@k \times rel(k)}{\sum_{i=1}^k rel(k)}$$

$$rel(k) = \begin{cases} 1 & (\text{if } item_k \in hits_u) \\ 0 & (\text{otherwise}) \end{cases}$$

■ユーザの負荷とシステムの効率

さらに、最終的な科目推薦リストを得るまでのユーザの負荷に関する評価のために以下を考える。

- フェーズ数
- 選択キーワード数
- 選択クラスタ数

提案システムおよび比較手法のシステムでは、ユーザはクラスタの選択とキーワードの選択を繰り返すことになる。このとき、フェーズ数は表示される選択肢 (ワードクラウドとレーダチャートの組) が変更された回数であり、この値が大きいほどユーザはたくさんの選択肢を見せられることになり、その負荷が大きくなる。一方、選択キーワード数と選択クラスタ数は、選択の入力の手間こそあるが、その手間はインタフェースによるところが大きい。インタフェースについては改良の余地はあるが、それを議論することは本論文の範囲を超えている。また、提案システムにおいては、選択キーワード数と選択クラスタ数が大きいほど候補科目が絞られるということである。よって、少ないフェーズ数で選択キーワードおよび選択クラスタの数が多い方が、より効率が良いといえる。

4.4.4 結果

5つの各手法による推薦リストに対して、4.4.3項で述べた指標値を用いて評価を行った。

■推薦精度

表3より、プロトタイプシステムの推薦精度が、すべての指標において最も良いことが分かった。なお、プロトタイプシステムとの差についてt検定を行ったところ、優位水準5%で有意な差が見られた値に*を付けている。比較手法Dに対しては、どの指標でも有意差が認められた。また、比較手法A~Cも比較手法Dより精度が良いことから、ユーザとのインタラクションを通して候補科目を絞り込み、科目のランキングリストを生成することは、推薦精度の向上に有効であるといえる。比較手法BおよびCに関しては、Precision@5で有意差が認められ、提案システムの性質上、推薦リスト内の順位には重きを置かないことに鑑みれば、これらの比較手法よりは良い推薦精度が実現できているといえる。一方で、比較手法Aは推薦精度の上ではプロトタイプと同程度の精度であることになる。

■ユーザの負荷とシステムの効率

次に、フェーズ数やキーワード数、選択クラスタ数に着目して、プロトタイプシステムと比較手法A、Bを比較した。表4に結果を示す。まず、比較手法Bではキーワードを入力しないため、キーワード数は0となっている。また、比較手法Aでは1フェーズですべてのキーワードを入力させるため、フェーズ数は1となっている。さらに、比較手法Aの選択クラスタ数は、ユーザが入力したキーワードの属したクラスタであって、ユーザが明にそれらのクラスタを選択したわけではないことから()付としている。

プロトタイプシステムではフェーズ数が2~3となつて

表3 比較実験の評価結果

Table 3 Evaluation results of comparative experiments.

指標値	hitrate	Precision@5	MAP
プロトタイプ	1.00	0.64	0.81
手法A	0.95	0.58	0.69
手法B	1.00	0.46 *	0.71
手法C	0.90	0.48 *	0.58 *
手法D	0.68 *	0.28 *	0.42 *

表4 プロトタイプシステムと比較手法A、Bの比較

Table 4 Comparison of the prototype system and comparison methods A and B.

手法	プロトタイプ	手法A	手法B
フェーズ数(平均値)	2.3	1	4.8
フェーズ数(最大値)	3	1	9
キーワード数(平均値)	7.9	4.6	0
キーワード数(最大値)	16	10	0
選択クラスタ数(平均値)	5.4	(2.5)	10.9
選択クラスタ数(最大値)	8	(4)	23

いる一方で、比較手法Aと比べると、入力キーワード数がおおよそ1.6倍、選択クラスタ数がおおよそ2倍となっている。つまり、2フェーズ目以降によって線形に増えるようなことは起こっていないが、1フェーズ目で終わってしまうよりも多くの入力を得ることができている。推薦精度については比較手法Aとは有意な差は見られなかったが、プロトタイプシステムは、大きく負荷を増加させることなく、より詳細にユーザの志向を引き出しているといえる。また、比較手法Bと比べると、プロトタイプシステムは、デンドログラム探索のみでなくキーワードフィルタリングを組み合わせることで、ユーザはフェーズ数を抑えたうえで、より精度の高い推薦リストを得られていることが分かった。

4.4.5 使用感に関するアンケート

実験で扱ったシステムの使用感に関するアンケート調査を実施した*1。この使用感の調査では、推薦結果は明かさず、操作性のみを考慮してもらった。被験者には以下の作業をしてもらった。

- 4つのシステムを使いやすかった順に並べる。
- 4つのシステムを、講義科目の推薦システムとして実際に使いたいと思える順に並べる。

このとき、どのシステムが提案システムであるかも明かさないうちで行った。

結果を表5および表6に示す。それぞれ、各手法を各順位とした被験者の割合を示す。さらに、1位または2位に選んだ被験者の割合についても調査した。

表5より、クラスタ選択のみを行う比較手法Bが最も

表5 使いやすかったシステムの回答順位の割合(%)

Table 5 Percentage of system rankings that were easy to use.

手法	プロトタイプ	手法A	手法B	手法C
1位	16	21	37	26
2位	32	11	32	26
3位	32	37	16	16
4位	21	32	16	32
上位	24	16	34	26

表6 使いたいと思えるシステムの回答順位の割合(%)

Table 6 Percentage of system rankings that you would like to use.

手法	プロトタイプ	手法A	手法B	手法C
1位	42	21	26	11
2位	16	21	53	11
3位	26	37	21	16
4位	16	21	0	63
上位	29	21	39	11

*1 本実験では、システムの制約上、システムの使用順序は4.4.2項で述べたとおりであり、本アンケートにおいては順序効果による影響がある可能性があることに注意されたい。

使いやすいという結果となった。また、比較手法 A が上位に選ばれなかった原因は、1つの大きなワードクラウドから選択するよりも、プロトタイプシステムのようにクラスタに分かれたワードクラウドの方がキーワードを選択しやすいということが原因と考えられる。

表 6 より、1位とした被験者数ではプロトタイプシステムが最も多く、使いたいと思ってもらえるものとなった。また、単語入力や数値入力が1回のみと比較手法 A や比較手法 C よりも、選択や入力を複数回行って候補科目を絞り込む比較手法 B やプロトタイプシステムの方が良い結果となった。これは、1回の入力で結果を出されるよりも、何度かインタラク션을重ねて推薦する方がユーザにとって納得しやすいものと考えられる。上位に選んだ人数の割合で、プロトタイプシステムよりも比較手法 B が高くなっている原因としては、単語入力よりも、選択のみを繰り返す方がユーザの負担が小さいことが結果に影響したと思われる。ユーザの負担を軽減するために、テキスト入力をさせるのではなく、ワードクラウド中の単語を直接クリックするなどして選択できるようにインタフェースを改善することが今後の課題として考えられる。

5. おわりに

本研究では、ユーザとのインタラク션을通してユーザの興味を取得することにより、教養科目を推薦するシステムを提案した。提案システムの評価実験を行った結果、提案手法の推薦精度が最も良く、ユーザとのインタラク션을考慮することは推薦精度の向上に有効であることを示した。

今後の課題としては、システムのインタフェースの改善およびさらなるユーザの負荷の軽減があげられる。また、ユーザが選択したクラスタのみでなく、選択しなかったクラスタに属する科目も次のフェーズに活用することにより、セレンディピティを考慮することができるとされる。また、本来の推薦対象ユーザである学部新入生を対象とした実証評価を行うことも今後の課題である。

参考文献

[1] Le, Q. and Mikolov, T.: Distributed representations of sentences and documents, *Proc. International Conference on Machine Learning*, Vol.14, pp.1188-1196 (2014).

[2] Jannach, D., Zanker, M., Felfernig, A. and Friedrich, G.: *Recommender Systems: An Introduction*, Cambridge University Press (2011).

[3] 神島敏弘: 推薦システムのアルゴリズム (2016), 入手先 (<http://www.kamishima.net/>).

[4] Chen, M. and Liu, P.: Performance Evaluation of Recommender Systems, *International Journal of Performance Engineering*, Vol.13, pp.1246-1256 (2017).

[5] Pazzani, M. and Billsus, D.: Content-based recommendation systems, *The Adaptive Web*, LNCS, Vol.4321,

pp.325-341 (2007).

[6] Amazon.com, available from (<https://www.amazon.com/>).

[7] Dwivedi, S. and Roshni, K.: Recommender System for Big Data in Education, *Proc. 5th National Conference on E-Learning & E-Learning Technologies* (2017).

[8] Yanhui, D., Dequan, W., Yongxin, Z. and Lin, L.: A Group Recommender System for Online Course Study, *Proc. 7th International Conference on Information Technology in Medicine and Education* (2015).

[9] Estivill-Castro, V., Limongelli, C., Lombardi, M. and Marani, A.: DAJEE: A dataset of joint educational entities for information retrieval in technology enhanced learning, *Proc. 39th International ACM SIGIR Conference on Research and Development in Information*, pp.681-684 (2016).

[10] O'Mahony, M.P. and Smyth, B.: A recommender system for on-line course enrolment: An initial study, *Proc. ACM Conference on Recommender Systems*, pp.133-136 (2007).

[11] Zhao, J., Bhatt, C., Cooper, M. and Shamma, D.A.: Flexible Learning with Semantic Visual Exploration and Sequence-Based Recommendation of MOOC Videos, *Proc. CHI Conference on Human Factors in Computing Systems*, pp.329:1-329:13 (2018).

[12] Jing, X. and Tang, J.: Guess You Like: Course Recommendation in MOOCs, *Proc. International Conference on Web Intelligence*, pp.783-789 (2017).

[13] Obeid, C., Lahoud, I., Khoury, H. and Champin, P.-A.: Ontology-based Recommender System in Higher Education, *Companion Proc. Web Conference 2018*, pp.1031-1034 (2018).

[14] Yu, X., Tungare, M., Fan, W., Perez-Quinones, M., Fox, E., Cameron, W., Teng, G. and Cassel, L.: Automatic Syllabus Classification, *Proc. 7th ACM/IEEE-CS Joint Conference on Digital Libraries* (2007).

[15] Gardner, J., Onuoha, O. and Brooks, C.: Integrating Syllabus Data into Student Success Models, *Proc. 7th International Learning Analytics & Knowledge Conference*, pp.586-587 (2017).

[16] 堀 幸雄, 中山 堯, 今井慈郎: カリキュラムの特徴抽出と時間割の要約生成, 情報知識学会誌, Vol.20, No.2 (2010).

[17] 西森友省, 堀 幸雄, 今井慈郎: 履修履歴を用いた科目推薦システム, 情報処理学会第 75 回全国大会 (2013).

[18] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and their Compositionality, *Proc. 26th International Conference on Neural Information Processing Systems*, Vol.2, pp.3111-3119 (2013).

[19] Mecab, available from (<http://taku910.github.io/mecab/>).

[20] ウィキペディア, 入手先 (<https://ja.wikipedia.org/>).

[21] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, *Proc. Workshop at ICLR* (2013).

[22] 澤崎夏希, 遠藤聡志, 當間愛晃, 山田孝治, 赤嶺有平: 量的不均衡データに対する学習精度改善のための文書かさ増し手法, Web インテリジェンスとインタラクシオン研究会 (2017).

[23] Ward, J.H., Jr.: Hierarchical Grouping to Optimize an Objective Function, *Journal of the American Statistical Association*, Vo.58, pp.236-244 (1963).

[24] Heimerl, F., Lohmann, S., Lange, S. and Ertl, T.: Word Cloud Explorer: Text Analytics Based on Word Clouds,

Proc. 47th Hawaii International Conference on System Sciences, pp.1833-1842 (2014).

- [25] Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. and Qu, H.: Context preserving dynamic word cloud visualization, *IEEE Computer Graphics and Applications*, Vol.30, pp.42-53 (2010).
- [26] SciPy, available from (<https://www.scipy.org/>).
- [27] 広島大学シラバス (2017, 2018), 入手先 (<https://momiji.hiroshima-u.ac.jp/syllabusHtml>).
- [28] 第一学習社教科書 (2018), 入手先 (<http://www.daiichi-g.co.jp/kyoka>).
- [29] gensim, available from (<https://radimrehurek.com/gensim/models/doc2vec.html>).



竹森 汰智

1995年生。2017年広島大学工学部第二類情報工学専攻卒業。2019年同大学大学院修士課程修了。同年株式会社両備システムイノベーションズ入社。



亀井 清華 (正会員)

1978年生。2001年広島大学工学部第二類情報工学専攻卒業。2003年同大学大学院修士課程修了。2006年同博士課程修了。博士(工学)。2006年鳥取環境大学環境情報学部助手。2007年同助教。2008年広島大学大学院工学研究科情報工学専攻助教。2012年同准教授。分散アルゴリズムおよび推薦システムの研究に従事。電子情報通信学会, IEEE, ACM 各会員。

(担当編集委員 湯本 高行)