協調フィルタリングを用いたソフトウェア技術者向け開発技術推薦の試み

角田　雅照*，秋永　知宏*, (注 1)，引地　一将*, (注 2)，大杉　直樹**，柿元　　健***，
門田　暁人*，松本　健一*

A Development Technique Recommendation Method for Software Engineers Using Collaborative Filtering

Masateru TSUNODA*, Tomohiro AKINAGA*, (1), Kazumasa Hikichi*, (2), Naoki OHSUGI**, Takeshi KAKIMOTO***,Akito MONDEN*, Ken-ichi MATSUMOTO*

It is important for software engineers to improve their skills by self-learning existing software development techniques. However, it is not easy for engineers to find out which techniques should be learned because there are so many existing techniques and also new techniques are emerging continuously. To support software engineers' self-learning, we propose a development technique recommendation method using collaborative filtering (CF). In our method, a user (engineer) gives ratings to development techniques based on his/her interest or benefit, then development techniques are recommended based on similarity measure of CF. The similarity is computed based on co-occurrence of terms of development techniques on web pages. We conducted an experiment to evaluate the recommendation accuracy of the proposed method. As a result, the average NDPM was 0.41 for recommending development techniques that are unfamiliar to software engineers.

キーワード：学習支援，協調フィルタリング，クローリング，共起関係

1. はじめに

ソフトウェア開発において，技術者のスキルを高めることは非常に重要である．例えば，経済産業省が組込みソフトウェア産業の事業責任者を対象に行った調査[(1)]では，設計品質の向上，開発期間の短縮，生産性の向上のためには，技術者のスキル向上が最も有効であるとの回答が最多であった．また，日本情報システム・ユーザー協会が行った調査[(2)]によると，IT 要員数は足りているが能力が不足していると回答している企業の割合が 22%，要員数も能力も不足していると回答している企業の割合が 57%あった．すなわち，スキル向上は重要な課題であるにも関わらず，十分に解決されていないのが現状であるといえる．

技術者のスキルの向上には，自発的学習が大きな役割を果たす．ニッセイ基礎研究所の調査[(3)]によると，専門スキルの獲得方法として，自発的学習を挙げている労働者の割合は，企業の規模により異なるが 49%～63%であり，OJT（On the Job Training）に次ぐ 2 番目である．大企業（従業員が 301 人以上の企業）の場合，勤務先が実施する研修により専門スキルを獲得する労働者の割合は 36%であり，研修も比較的大きな役割を果たしているが[(4)]，中小企業の場合，同割合は 11～21%であり，自発的学習の重要性がより大きい．また，近年オープンソースソフトウェア（OSS）の開発が盛んであるが，OSS を開発する技術者（学生や研究者の場合もある）の多くは，業務と無関係に開発を行っており，自発的学習に頼らざるを得ない．

* 奈良先端科学技術大学院大学 情報科学研究科 (Graduate School of Infor-mation Science, Nara Institute of Science and Technology)
** 株式会社 NTT データ (NTT DATA Corporation)
*** 香川高等専門学校 電気情報工学科 (Department of Electrical and Computer Engineering, Kagawa National College of Technology)
(注 1) 現在，日立ソフトウェアエンジニアリング株式会社 (Presently with Hitachi Software Engineering Co., Ltd.)
(注 2) 現在，株式会社日立製作所 (Presently with Hitachi, Ltd.)

ただし，ソフトウェア開発に関する技術を，技術者自ら学習することは容易ではない．技術者がスキルアップするためには，情報処理技術者試験で扱われているような，ある程度普遍的な技術を学習することに加え，最新の開発技術を学習する必要がある．前者の技術は体系的に整理されており，技術者はある程度容易に学習すべき技術を特定することができる．後者の技術については，最新の技術が多数提案され続けており，その中から学習すべき技術を特定することは容易ではない．

そこで本論文では，技術者の学習支援を目的として，学習すべき技術を推薦する方式を提案する．有用な推薦は技術者によって大きく異なる．例えば，Web アプリケーションを開発するための技術を学習している技術者に対して，JSP（Java Server Pages）関連の技術を推薦することは有用であると考えられるが，ERP パッケージ関連の技術を推薦することはミスマッチである．提案方式では，協調フィルタリング[5][6][7]の考え方に基づき，現在学習している（興味を持っている）開発技術と関連の強い技術の推薦が有用であると考え，推薦を行う．協調フィルタリングとは，主に書籍や音楽などのアイテムを推薦するシステムで用いられてきた技術である．

提案方式では，開発技術の関連の強さを求めるために，インターネット上での情報を利用する．ソフトウェア開発技術の多くは，様々な Web ページで紹介，解説されている．このとき，関連の強い技術同士は，同じ Web ページ上に記述されている可能性が高いと考えられる．筆者らが調べた範囲では，Web ページでの開発技術の解説記事では，組み合わせて使う技術，及び類似した技術との対比が記述されることが多い．例えば，典型的な解説ページ[8]では，CGI（MySQL）と組み合わせて用いる Ruby が紹介されるとともに，Ruby と対比される Perl, Java, C++について記述されている（調査を行ったわけではないが，一般に技術に関する解説ページ（例えば統計学など）では，同様の傾向が見られると考えられる）．

そこでインターネット上の多数の Web ページにおける共起関係に基づいて単語間の関連の強さを求めるアルゴリズム[9]を適用し，開発技術間の関連を求める．従来の協調フィルタリングのアルゴリズムでは，推薦時に多数のユーザがそれぞれのアイテム（開発技術）について評価を行っている必要があるが，提案する方式の場合，ユーザ数の多寡，他ユーザの評価の有無に関わらず，推薦が可能となる．

以降，2 章において協調フィルタリングについて述べ，3 章で提案する推薦方式について説明する．4 章では評価実験について説明し，5 章において関連研究について述べる．最後に 6 章で結論と今後の課題について述べる．

2. 協調フィルタリング

協調フィルタリングは，ユーザにとって好ましい，または役立つと考えられるアイテム（書籍，音楽など）を推薦するための手法として用いられている[5][6][7]．「協調」とは，ユーザの知識を利用することを意味し，「フィルタリング」とは，大量のアイテムの中から，役立つアイテムだけを選び出して推薦することを意味する．一般的な協調フィルタリングで推薦を行う場合，各ユーザが各アイテムを（5 段階の数値などで）評価していることが前提となる（システムによっては，ユーザがそのアイテムを閲覧したかどうかを評価の代わりに用いることもある）．あるユーザが未評価のアイテムが，そのユーザにとって役立つと考えられる場合，そのアイテムを推薦する．

協調フィルタリングの主なアルゴリズムとして，ユーザベース手法とアイテムベース手法の 2 つがある．ユーザベース手法は，「アイテムの評価（好み）が似たユーザは，どのアイテムに対しても似た評価を行う」と仮定し，推薦を行う．具体的には，各ユーザの各アイテムに対する評価を要素とするベクトルを，ユーザごとに作成し，そのベクトルのなす角をユーザの類似度とする．そして推薦対象のユーザが未評価で，かつ類似したユーザの評価が高いアイテムを推薦する．ユーザベース手法を用いた推薦システムとして，Resnick ら[6]の GroupLens が挙げられる．GroupLens は，Usenet にある多数のニュース記事から，ユーザの好みに合うと予測される記事を選び出して推薦するシステムである．

アイテムベース手法は Sarwar[7]らによって提案されたアルゴリズムであり，アイテム間の類似度に基づいて推薦を行う．アイテムベース手法の場合も，各ユー

ザの各アイテムに対する評価を要素としてベクトルを作成するが，ユーザごとにベクトルを作成するのではなく，アイテムごとに作成し，類似度を計算する．すなわち，「あるグループのユーザに高評価されるアイテムは，類似の性質を持っている」と仮定し，推薦対象のユーザが高い評価を行っているアイテムと類似度の高いアイテムを推薦する．本論文では，利用者以外のユーザの登録，評価を不要なシステムとするため，アイテムベース手法に Web ページにおける単語の共起関係に基づく類似度計算法を組み合わせた方式を提案する．詳細については次章で述べる．

3. 提案方式

3.1 推薦アルゴリズム

提案方式では，協調フィルタリングのアイテムベース手法に基づいて推薦を行う．ただし，他ユーザの評価を利用する代わりに，Web ページにおける単語の共起関係に基づいて関連の強さを求めるアルゴリズム[9]を適用し，開発技術（を指す単語）の Web ページでの共起関係に基づいて類似度を計算する．すなわち，「1つの Web ページ内で同時に紹介，解説されることが多い開発技術同士は関連が強い」と仮定し，推薦対象のユーザが高い評価を行っている開発技術と類似度の高い（Web ページにおける共起関係が類似している）開発技術を推薦する．これにより，推薦対象のユーザ以外の評価なしで推薦が可能となる．

提案方式での類似度計算は，表 1 のような $m \times n$ のマトリックス形式のデータを想定している．$p_i \in \{p_1, p_2, ..., p_m\}$は i 番目の Web ページを表し，$t_j \in \{t_1, t_2, ..., t_n\}$は j 番目の開発技術（の単語）を表す．また，$r_{ij} \in \{r_{11}, r_{12}, ..., r_{mn}\}$は Web ページ p_i に開発技術 t_j が含まれているかどうかを表し，含まれている場合は 1，含まれていない場合は 0 となる．

ただし，膨大な数の Web ページそれぞれについて，多数存在する推薦候補の単語が含まれているかどうかを調べるのは膨大な時間を要する．そこで提案方式では，検索エンジンで検索を行った場合のヒット数を用いた類似度計算法を適用し[9]，表 1 のデータを使った場合と同等の計算結果を得て推薦を行う．この場合の類似度は，検索エンジンから得られるヒット数を，何らかの類似度計算法（Jaccard 係数や Overlap 係数など）に当てはめて求められるが[9]，本論文では協調フィルタリングで一般的なコサイン類似度[10]に当てはめて計算を行う．

表 1　提案方式で想定するデータ

	t_1	t_2	…	t_j	…	t_n
p_1	r_{11}	r_{12}	…	r_{1j}	…	r_{1n}
p_2	r_{21}	r_{22}	…	r_{2j}	…	r_{2n}
…	…	…	…	…	…	…
p_i	r_{i1}	r_{i2}	…	r_{ij}	…	r_{in}
…	…	…	…	…	…	…
p_m	r_{m1}	r_{m2}	…	r_{mj}	…	r_{mn}

表 1 のデータを使ってアイテムベースのコサイン類似度を計算する場合，開発技術 t_b と t_j の類似度 $sim(t_b, t_j)$ は以下の式により計算される．

$$sim(t_b, t_j) = \frac{\sum_{f=1}^{m} r_{fb} \times r_{fj}}{\sqrt{\sum_{f=1}^{m} r_{fb}^{\,2}} \sqrt{\sum_{f=1}^{m} r_{fj}^{\,2}}} \quad (1)$$

ここで，Web ページ p_f に開発技術 t_b が含まれる場合，r_{fb} は 1 となり，$r_{fb}^{\,2}$ も 1 となる．p_f に t_b が含まれない場合，$r_{fb}^{\,2}$ は 0 となる．すなわち，$\sum_{f=1}^{m} r_{fb}^{\,2}$ は t_b を検索エンジンで検索した場合のヒット数に等しくなる．また，$r_{fb} \times r_{fj}$ は p_f に t_b と t_j が含まれる場合のみ 1 となる．すなわち，$\sum_{f=1}^{m} r_{fb} \times r_{fj}$ は t_b と t_j を検索エンジンでアンド検索した場合のヒット数に等しくなる．よって以下の式により，式(1)を使った場合と同等の類似度が得られる．

$$sim(t_b, t_j) = \frac{h_{bj}}{\sqrt{h_b}\sqrt{h_j}} \quad (2)$$

ここで h_b，h_j はそれぞれ開発技術 t_b，t_j を検索エンジンで検索した場合のヒット数，h_{bj} は t_b と t_j を検索エンジンでアンド検索した場合のヒット数を表す．

類似度 $sim(t_b, t_j)$ に基づき，ユーザが未評価の開発技術 t_b に対する予測評価値 $\hat{e}_b$ を以下の式（類似度を重みとした加重平均）により求める．

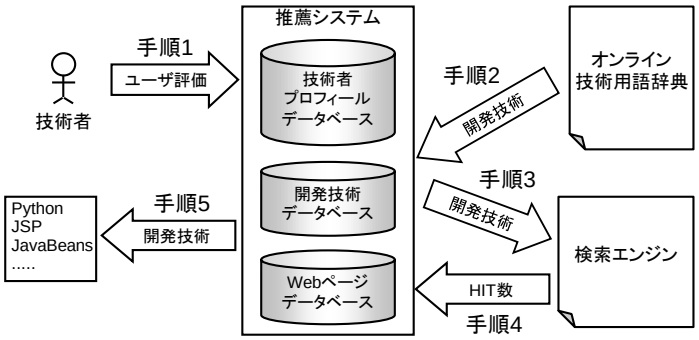

図 1　ソフトウェア技術者向け開発技術推薦システムの概要

$$\hat{e}_b = \frac{\sum_{j \in k-nearestTechs} e_j \times sim(t_b, t_j)}{\sum_{j \in k-nearestTechs} sim(t_b, t_j)} \quad (3)$$

ここで, e_jはユーザの開発技術t_jに対する評価値を表す. また，*k-nearestTechs* は，ユーザが評価を行っており，かつ t_b との類似度が高い k 個の開発技術の集合を表す. *k-nearestTechs* はあらかじめ決めておく必要がある. 式 3 の計算結果に基づき，予測評価値の高い順に開発技術をユーザに推薦する.

3.2 開発技術推薦システムの概要

提案方式に基づくシステムの概要を図 1 に示す．システムでは，最初にユーザ（技術者）が任意の開発技術に対して（興味があるかなどに基づき）評価を行う．システムはその評価に基づき，ユーザに適した（ユーザの評価が高い開発技術と類似した）開発技術を推薦する．推薦候補となる開発技術は，オンライン技術用語辞典[(11)(12)]から抽出する．開発技術の類似度は，開発技術を紹介，解説している Web ページにおける，開発技術の共起関係を Google[(13)]などの検索エンジンの検索結果に基づいて計算する．推薦の手順を以下に示す.

手順 1　ユーザは任意の開発技術に対して 4 段階で評価を行い，システムに入力する．得られた情報は**技術者プロフィールデータベース**と**開発技術データベース**に記録する.

手順 2　システムは, 推薦候補となる開発技術をオンライン技術用語辞典から抽出し, 開発技術データベースに追加する.

手順 3　システムは，開発技術データベースから，2 つの開発技術（ユーザが評価済みの技術と未評価の技術）を選択し，それらを検索エンジンでアンド検索する．得られたヒット数は **Web ページデータベース**に記録する.

手順 4　システムは，開発技術データベースと Web ページデータベースを用いて, 開発技術間の類似度を計算する.

手順 5　システムは，各データベースを用いて，ユーザが興味を持っている技術と類似しており, かつユーザが知らない（未評価の）開発技術を，興味が高いと推定される順番にリスト形式で表示する.

対象としたオンライン技術用語辞典には，情報処理技術者試験とオーバラップする部分もあるが，情報処理技術者試験に含まれない（新しい）技術用語も多く登録されており，システムで利用する対象としてふさわしいと考える．例えば e-words[(12)]には，クイックソート（情報処理技術者試験に含まれる技術用語）などが登録されているが，Ruby on Rails や PostgreSQL（情報処理技術者試験に含まれない技術用語）などの技術用語も数多く登録されている.

図 2 に各データベースのイメージを示す．ユーザが評価する開発技術（技術者プロフィールデータベースの開発技術）は，オンライン技術用語辞典（システムの推薦候補の開発技術）に含まれていなくてもよい．ただし，実際にユーザに推薦される開発技術は，オンライン技術用語辞典に含まれているもののみとなる．また，オンライン技術用語辞典以外でも，何らかのサイトから開発技術に関する単語を抽出し，それらを推薦候補とすることができる．さらに，人手で開発技術に関する単語を入力し，それらを推薦候補とすることが可能である．提案方式では，推薦候補となる開発技術の単語だけあればよく，単語間の関連はシステムに

技術者プロフィール
データベース

開発技術	評価
CGI	4
Perl	3
Java	2

Webページデータベース

検索語	ヒット数
CGI	97,700,000
Python	2,600,000
CGI, Python	789,000
・・・	・・・

開発技術データベース

開発技術	評価
CGI	済
Perl	済
Java	済
Python	未
Ruby	未
Java Server Pages	未
・・・	・・・

技術者プロフィール
データベースに含まれる
開発技術(オンライン技術
用語辞典に含まれていな
い場合がある)

推薦対象の開発技術
(オンライン技術用語
辞典に含まれている)

図 2　各データベースのイメージ

入力する必要がない.

ユーザは，開発技術データベースに含まれている技術全てを評価しなければならないわけではなく，1 つ以上の開発技術を評価すれば推薦可能となる．例えば，ユーザは「CGI」だけを評価しており，開発技術データベースにはユーザが評価していない「Python」，「Perl」，「Ruby」があったとする．この場合，「CGI」とそれぞれの開発技術の Web 上での共起関係と，「CGI」に対するユーザの評価に基づいて，各技術の推薦が行われる．

各手順について詳説する. 手順 1 では, ユーザは図 3 のような画面によって，任意の（知っている）開発技術 10～20 個程度について名称を入力するとともに，4 段階で評価を行い，システムに入力する.

手順 2 において，技術用語はオンライン技術用語辞典の索引ページから取得する．取得方法は用いる辞典により異なるが，本論文で用いる辞典の場合，用語が「ア」「イ」「ウ」などの頭文字ごとに別々のページに掲載されているため，まずこれらのページ（e-words の場合，トップページからリンクされており，かつ，URL

推薦技術1
・用語辞典による解説ページのサマリ　[詳細]
・関連の強い技術（ユーザが評価したもの）
・関連の強い技術とのAND検索で得られたWebページのサマリ　[検索結果一覧]

推薦技術2
・用語辞典による解説ページのサマリ　[詳細]
・関連の強い技術（ユーザが評価したもの）
・関連の強い技術とのAND検索で得られたWebページのサマリ　[検索結果一覧]

推薦技術3
・用語辞典による解説ページのサマリ　[詳細]
・関連の強い技術（ユーザが評価したもの）
・関連の強い技術とのAND検索で得られたWebページのサマリ　[検索結果一覧]

…

図 4　出力画面のイメージ

図 3　入力画面のイメージ

が「http://e-words.jp/p/*」となっているページ）を保存する．次に，保存したファイルから，定型句（e-words の場合，「 Python」など）により，技術用語を特定する.

手順 3 では，まず，検索エンジンでキーワード（開発技術データベースの開発技術）を検索するための URL （ Google の場合， http://www.google.co.jp/search?hl=ja&q=CGI+Perl など）を生成する．次に，生成した URL でアクセスし，検索結果の Web ページ（最初のページのみ）をファイルとして保存する. 最後に，保存したファイルから，定型句（Google の場合，「<title> CGI Python - Google 検索」や「 約 789,000 件」など）により，検索したキーワードとヒット数を特定する.

手順 4 は前節で詳説したので省略する. 手順 5 では，図 4 のような画面イメージで推薦結果が出力される. それぞれの推薦技術ごとに，用語辞典による解説ページのサマリ，関連の強い技術（ユーザが評価したもの），共起関係が見られた Web ページのサマリが出力される. サマリを出力する理由は，用語辞典の解説では，関連の強い開発技術と推薦技術をどのように組み合わせて使うかまでは記述されていない場合があるためである. さらに[詳細]のリンクをクリックすることにより，用語辞典による解説ページが表示され，[検索結果一覧]をクリックすることにより，検索エンジンによる共起関係抽出時の結果一覧が表示される.

4. 評価実験

4.1 概要

提案方式の有効性を確かめるために，開発技術の推薦精度を実験により確かめた．実験では，技術者の個別性に考慮しない，知名度の高い開発技術の推薦（単純推薦）の精度と，開発技術のランダムな推薦の精度を，提案方式による推薦の精度と比較した．単純推薦

では,「多くの Web ページに掲載されている開発技術は,知名度が高い」と仮定し，検索エンジン（Google[13]）を利用して開発技術を検索し，ヒット数の大きい順に推薦を行った.

4.2 利用したデータ

ソフトウェア開発技術者 29 人，情報科学専攻の大学院生 25 人,およびソフトウェア工学の研究者 11 人の計 65 人が開発技術の評価を行ったデータに対して，推薦を行った．評価では，まず各開発技術を知っているかどうか（開発技術の概要を説明できるかどうか）を回答してもらった．知っている場合，その開発技術に対する興味の度合いを 4 段階（1: 興味がない，2: 少し興味がある，3: 興味がある，4: 大変興味がある）で評価してもらった.

推薦候補とした開発技術を図 5 に示す．提案方式に基づくシステムでは推薦候補の開発技術はオンライン用語辞典から収集するが，実験では比較的知名度が高く, かつ多様な開発技術を著者らが選定した．これは，実験に用いる評価指標（4.3 節で後述する NDPM）では実際の評価値と予測された評価値を比較するが，知名度の低い開発技術では，実際の評価値が欠損値となり，推薦精度を評価することができないためである.

図 5 は開発技術に対する技術者の評価と知名度を示している．縦軸は開発技術の項目を示し，横軸はその開発技術を知っていた技術者の割合を示す．さらに，評価 3（興味がある）を閾値として，技術者を色分けしている．グラフを見ると，UML のように全員が知っており，さらに技術者の多くが「興味あり」と評価している開発技術や, Pascal のようにほとんどの技術者が知っているにもかかわらず，その多くが「興味なし」と評価している開発技術があった．また，全社的品質管理などのように，知らない技術者が多いが，知っている技術者の半分以上が「興味あり」と評価している開発技術，さらに，Visual Component Library のように，多くの技術者が知らず，知っている技術者でさえ,「興味なし」と評価している開発技術もあった．このように，開発技術によって，技術者の評価や知名度が大きく異なっていることが分かる.

4.3 評価指標

推薦精度を評価するための基準として, Yao が提案した NDPM （ Normalized Distance-based Performance Measure）[14]を用いた．NDPM はユーザがシステムに求める理想の推薦 O_a と, システムがユーザに行う推薦 O_a’との差異を表す値であり，推薦システムの精度評価に広く用いられている[15][16][17]．推薦システムの評価実験では，ユーザのアイテムに対する評価値が既知のデータを用い，評価値が未知と仮定して推薦を行う．理想の推薦とは，ユーザ評価値の実測値が大きい順にアイテムを推薦した状態を指す．これに対し，システムによる推薦では，ユーザ評価値の予測値が大きい順にアイテムを推薦する.

NDPM は[0,1]の範囲の実数値を取り，値が小さいほど O_a と O_a’の差異が小さい，すなわち，推薦の精度が高いことを示す．また，中間値である 0.5 はランダムに推薦を作成した場合（ランダム推薦）の NDPM の理論値である．ランダム推薦の NDPM は，未知でかつ有用と考えられる機能の候補を全て推薦する場合の NDPM と理論的に等価であり，推薦アルゴリズムが最低限備えるべき精度である（NDPM が 0 に近いほど推薦精度が高く, 0.5 に近いほど推薦精度が低い）. NDPM は式(4)で計算される.

$$NDPM(O_a, O_a') = \frac{\sum_{o_k \in O_a \cup O_a'} \sum_{o_l \in O_a \cup O_a'} dpm(o_{ak}, o_{al}, o_{ak}', o_{al}')}{\sum_{o_k \in O_a \cup O_a'} \sum_{o_l \in O_a \cup O_a'} dpmNormalizer(o_{ak}, o_{al})} \tag{4}$$

ここで, o_{ak} は理想の推薦 O_a における開発技術 t_k の推薦順位を表し，o_{ak}’はシステムがユーザに行う推薦 O_a’における開発技術 t_k の推薦順位を表す. 理想の推薦は，技術者の評価値が大きい順に推薦を行った場合を表す. 関数 *dpm* と関数 *dpmNormalizer* の定義を式(5)，(6)に示す.

$$dpm(r_{ak}, r_{al}, r_{ik}, r_{il}) = \begin{cases} 0 & \begin{pmatrix} (r_{ak} > r_{al}) \wedge (r_{ik} > r_{il}) \\ \vee (r_{ak} = r_{al}) \wedge (r_{ik} = r_{il}) \\ \vee (r_{ak} < r_{al}) \wedge (r_{ik} < r_{il}) \end{pmatrix} \\ 1 & \begin{pmatrix} (r_{ak} = r_{al}) \wedge (r_{ik} \neq r_{il}) \\ \vee (r_{ak} \neq r_{al}) \wedge (r_{ik} = r_{il}) \end{pmatrix} \\ 2 & \begin{pmatrix} (r_{ak} > r_{al}) \wedge (r_{ik} < r_{il}) \\ \vee (r_{ak} < r_{al}) \wedge (r_{ik} > r_{il}) \end{pmatrix} \end{cases} \tag{5}$$

$$dpmNormalizer(r_{ak}, r_{al}) = \begin{cases} 1 & (r_{ak} = r_{al}) \\ 2 & (r_{ak} \neq r_{al}) \end{cases} \tag{6}$$

なお，知名度の高い開発技術は，すでに技術者が知

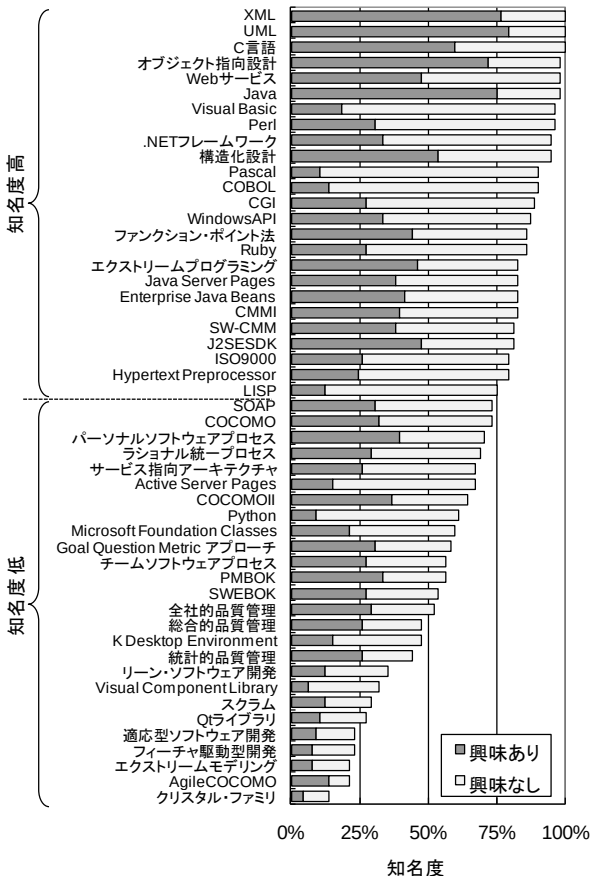

図 5　開発技術に対する技術者の評価と知名度

っていることが多いため，実際には推薦対象となることは少ないが，実験では知名度の高い開発技術も知らないとみなして推薦を行っている．そこで，実験の正確性を高めるため，全開発技術，知名度の高い（実際には推薦が不要な）開発技術，知名度の低い（実際に推薦が必要な）開発技術の 3 つの場合の推薦精度を評価した．開発技術が 2 等分されるように，知名度 74%以上の場合を知名度の高い開発技術，それ以外を知名度の低い開発技術とした．

4.4 手順

実験では，それぞれの開発技術が未評価であると仮定し，以下の手順により推薦を行った．

手順 1　i 人目の技術者 u_i を推薦対象とする．

手順 2　j 番目の開発技術 t_j の値を未評価（開発技術を知らない）とみなし，評価値の予測を行う．

手順 3　全ての開発技術に対して手順 2 を繰り返す．

手順 4　評価値の予測値が大きい順に開発技術を推薦したとみなして，NDPM を計算する．

手順 5　全ての技術者に対して手順 1～4 を繰り返す．

なお，技術者が知らなかった開発技術は，技術者による評価が不明である（実際に有用であるかどうかがわからない）ため，NDPM の評価対象外としている．これは，協調フィルタリングのアルゴリズムの精度評価では一般的な方式である[6][7][10]（推薦後のアンケートが不要となり，より多くのユーザによる実験が可能となるため）．

4.5 結果

図 6 に，知名度の低い開発技術を各手法により推薦した場合の NDPM の箱ひげ図を示す（NDPM が小さいほど精度が高い）．提案方式の推薦精度が最も高く，単純推薦の精度が最も低くなっていた．また，Wilcoxon の順位和検定で代表値の差の検定を行った結果，有意水準 5%で，提案方式の NDPM は単純推薦，ランダム推薦と差があるといえた（それぞれ p=0.0%）．

図 5 を見るとわかるように，知名度の低い開発技術では，興味を持つ技術者の割合が全体の 50%を超える開発技術は存在しない．興味を持つ技術者の絶対数が少なくなるため，単純推薦の精度が低くなったと考えられる．逆に提案方式では，「ある技術者が興味を持つ開発技術と関連の強い開発技術は，同様にその技術者に興味を持たれる」，「同じ Web ページで紹介，解説される開発技術は関連が強い」という仮定が，ある程度現実と合致していたため，推薦精度が高かったと考えられる．提案方式は，知名度の低い（実際に推薦が必要である場合が多い）開発技術の推薦に対して特に有用であると期待できる．

図 7 に，知名度の高い開発技術を各手法により推薦した場合の NDPM の箱ひげ図を示す．提案方式の推薦精度が最も高く，次に単純推薦の精度が高かった．ただし，Wilcoxon の順位和検定で差の検定を行った結果，有意水準 5%で提案方式の NDPM はランダム推薦と差があるといえたが（p=0.3%），単純推薦とは差があるとはいえなかった（p=9.4%）．図 5 より，知名度の高い（実際には推薦が不要な場合が多い）開発技術では，興味を持つ技術者の割合が全体の 50%を超える開発技術がいくつか存在し，興味を持つ技術者の絶対数が多くなるため，単純推薦と提案方式の精度の差が小さくなったと考えられる．

最後に図 8 に，全ての開発技術を各手法により推薦した場合の NDPM の箱ひげ図を示す．提案方式の推薦

精度が最も高く，単純推薦とランダム推薦の差はほとんどなかった．Wilcoxon の順位和検定で代表値の差の検定を行った結果，有意水準 5%で，提案方式の NDPM は単純推薦，ランダム推薦と差があるといえた（それぞれ p=0.3%，p=0.0%）．

4.6 考察

提案方式は，2 つの前提「関連の強い技術同士が同じ Web ページ上に記述されている可能性が高い」，「ある技術者が興味を持つ開発技術と関連の強い開発技術は，同様にその技術者に興味を持たれる」に基づいて推薦を行っている．前者については，実験において類似度の高かった開発技術のうちのいくつかは，用語辞典の関連用語として現れていたため，ある程度妥当であると考えられる．例えば e-words において，「Python」と最も関連が高かった「Perl」は，「Python」の関連用語に含まれていた．

さらに，ある技術者が興味を持つ開発技術と関連の強い開発技術を推薦した（上記の考察より，提案方式は，ある技術者が興味を持つ開発技術と関連の強い技術を推薦しているといえる）結果，ランダム推薦よりも推薦精度が高かったことから，後者についても，ある程度妥当であると考えられる．

推薦が高精度でなかった理由は，一部の開発技術のヒット件数に，開発技術以外の一般の単語のヒット件数が含まれてしまうためであると考えられる．例えば，Ruby の場合，宝石の Ruby．スクラムの場合，ラグビー用語のスクラムがヒット件数に含まれる．これらを除外し，さらに推薦精度を高めることは今後の課題のひとつである．

推薦において，オンライン技術用語辞典での分野分類や関連用語を用いて類似度を計算することも可能であり，今後比較すべきであると考えるが，十分な精度が期待できない．なぜならば，分野分類はあまり細分化されておらず，分類内の単語の関連はそれほど強くないからである．例えば，e-words では，「サーバ（分類「WWW」の下層）」という分類の中に，用語が 60 個以上含まれている．また，関連用語についても，（推薦のために利用することを考慮すると）十分整理されているとはいえない．例えば，CGI に Ruby が用いられることがしばしばあり，提案方式では CGI は Ruby と最も類似度が高くなっているのに対し，CGI，Ruby どちらの

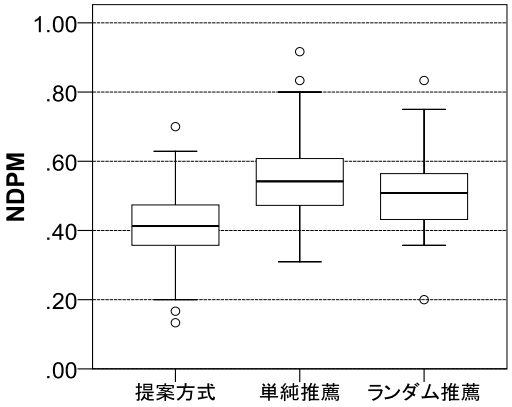

図 6　知名度の低い開発技術を推薦した場合の NDPM

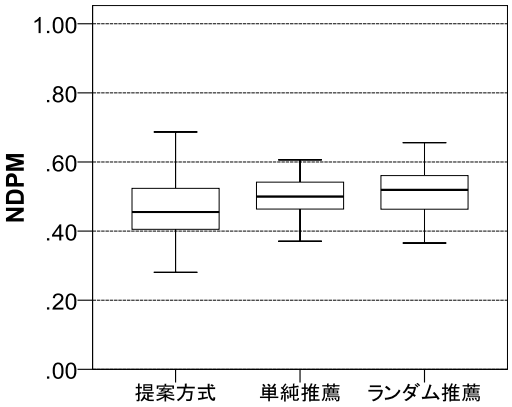

図 7　知名度の高い開発技術を推薦した場合の NDPM

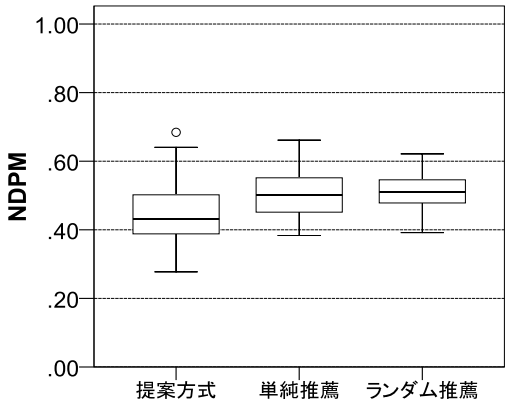

図 8　全ての開発技術を推薦した場合の NDPM

用語解説においても，関連用語として提示されない．

さらに，技術者が開発技術を自習するために，オンライン技術用語辞典を利用することを想定した場合，上記の理由により，学習すべき技術を容易に見つけることができる（学習のために十分に体系化されている）とはいえない．

5. 関連研究

学習支援のために，最適なアイテムを協調フィルタリングにより推薦するシステムいくつか提案されている．例えば岩下ら[18]は，学習者に英語リスニング教材を推薦するためのシステムを提案している．提案システムでは，学習者の教材に対する評価に基づき，ユーザベース手法により推薦を行っている．また，高橋ら[19]は e-ラーニングのコンテンツの閲覧履歴から，ユーザベース手法によりコンテンツの推薦を行っている．

これらの研究はユーザベース手法による推薦であり，推薦時に多数のユーザがそれぞれのアイテムについて評価している必要がある．ただし，ユーザベース法に基づいて推薦を行うためには，多数存在する開発技術をシステム利用者が評価している必要があり，その実現は容易ではない．そこで本論文では，Web ページでの開発技術の単語の共起関係に基づく類似度計算と，アイテムベース手法を組み合わせることにより，推薦対象のユーザ以外の評価を不要とした方式を提案しており，この点が従来研究と異なる．

Web ページでの単語の共起関係に着目して，検索エンジンのヒット件数から類似度を計算するアルゴリズムは，推薦システム以外の分野（例えば企業間の関係の抽出など[20]）には適用されているが，推薦システムの類似度計算に適用した事例は，筆者らの知る限り存在しない．奥ら[21]は，ユーザの地元にはなく，かつ旅先にしかないスポット（飲食店や観光場所）を推薦する方式を提案しており，検索エンジンのヒット件数から類似度を計算するアルゴリズムを方式の一部に利用している．ただし，推薦のための類似度計算ではなく，スポットと無関係な用語の除去のために利用しており，奥らの方式を情報技術の推薦に適用することはできない．

また，技術者に対する開発技術の推薦に，協調フィルタリングを適用した研究も存在しない．プログラムライブラリを推薦する方式はいくつか提案されており[22][23]，これらの方式ではユーザ数の多寡，他ユーザの評価の有無に関わらず推薦が可能である．ただし，プログラムライブラリは学習すべき開発技術ではなく，コーディング中のプログラムに直接適用するためのものであり，本論文で扱う開発技術の概念とは全く粒度が異なる（開発技術の一つであるプログラムライブラリに含まれる，各要素が推薦対象である）．そのため，推薦方式が異なる．具体的には，コーディング中のソースコードを解析した結果に基づいて推薦を行うため，開発技術の推薦に適用することはできない．

6. まとめ

本論文では，技術者の学習支援を目的として，協調フィルタリングに基づいて，多数存在する開発技術の中から，有用な開発技術を技術者に推薦する方式を提案した．提案システムは，Web ページにおける開発技術の共起関係に基づき開発技術を推薦する．評価実験の結果，特に知名度の低い開発技術の推薦に対して，提案システムが有効であることが確認された．

本論文の貢献は，技術者に対する開発技術の推薦を前提に，推薦システムのユーザ数の多寡，他ユーザの評価の有無に関わらず推薦を可能とするために，協調フィルタリングと Web ページでの単語の共起関係抽出を組み合わせた推薦方式を提案したことと，技術者に対する開発技術の推薦に，協調フィルタリングが有効に機能することを実験的に示したことである．今後の課題は，提案手法をシステムとして完成させ，技術者に広く利用してもらうことである．

謝辞

本研究の一部は，「次世代 IT 基盤のための研究開発」の委託に基づいて行われた．また，本研究の一部は，文部科学省科学研究補助費（若手 B：課題番号 22700034）による助成を受けた．

参 考 文 献

(1) 経済産業省 : “2009 年版組込みソフトウェア産業実態調査報告書”，経済産業省 (2009)

(2) 日本情報システム・ユーザー協会 : “企業 IT 動向調査 2009”，日本情報システム・ユーザー協会，東京 (2009)

(3) ニッセイ基礎研究所 : “働く人の就業実態・就業意識に関する調査”，ニッセイ基礎研究所，東京 (2004)

(4) 中小企業庁 : “2005 年版中小企業白書”， 中小企業庁 (2005)

(5) Goldberg, D., Nichols, D., Oki, B. M. and D. Terry: “Using Collaborative Filtering to Weave an Information Tapestry”, Communications of the ACM, Vol.35, No.12,

pp.61-70 (1992)

(6) Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J.: GroupLens: “An Open Architecture for Collaborative Filtering of Netnews”, Proc. of the ACM Conf. on Computer Supported Cooperative Work (CSCW’94), pp.175-186 (1994)

(7) Sarwar, B., Karypis, G., Konstan, J. and Riedl, J.: “Item-Based Collaborative Filtering Recommendation Algorithms”, Proc. of the 10th Intl. World Wide Web Conference (WWW10), pp. 285-295 (2001)

(8) 鶴長鎮一，渡辺恭弘：“Ruby で作るデータベース CGI”，快速 MySQL でデータベースアプリ！，第 7 回，アイティメディア，http://www.atmarkit.co.jp/flinux/rensai/mysql07/mysql07a.html

(9) Bollegala, D., Matsuo, Y. and Ishizuka, M.: “Measuring Semantic Similarity between Words Using Web Search Engines”, Proc. of the 16th International World Wide Web Conference (WWW 2007), pp.757-766 (2007)

(10) Breese, J. S., Heckerman, D. and Kadie, C.: “Empirical Analysis of Predictive Algorithms for Collaborative Filtering,” Proc. the 14th Conf. on Uncertainty in Artificial Intelligence, Wisconsin, pp.43-52 (1998)

(11) アイティメディア:“IT情報マネジメント用語事典”, http://www.atmarkit.co.jp/im/terminology/

(12) インセプト：“IT 用語辞典 e-Words”, http://e-words.jp/

(13) Google: “Google”, http://www.google.co.jp/

(14) Yao, Y.Y.: “Measuring Retrieval Effectiveness Based on User Preference of Documents”, Journal of the American Society for Information Sciences, Vol.46, No.2, pp.133-145 (1995)

(15) Balabanovic, M. and Shoham, Y.: “Fab: Content-based Collaborative Recommendation”, Communications of the ACM, Vol.40, No.3, pp.66-72 (1997)

(16) Balabanovic, M.: “An Adaptive Web Page Recommendation Service”, Proc. 1st Intl. Conf. on Autonomous Agents (Agents ’97), pp.378-385 (1997)

(17) Pretschner, A. and Gauch, S.: “Ontology Based Personalized Search,” Proc. the 11th IEEE Intl. Conf. on Tools with Artificial Intelligence, pp.391-398 (1999)

(18) 岩下文香，来住伸子：“協調フィルタリングを利用した英語教材推薦アルゴリズムの研究”，情報処理学会研究報告，Vol.2007，No.34，pp.53-60 (2007)

(19) 高橋泰樹，松澤俊典，山口未来，土肥紳一，和田雄次：“学習者に適した学習教材の推薦と配信”，情報処理学会研究報告，Vol.2007，No.12，pp.157-162 (2007)

(20) 金英子，松尾豊，石塚満：“Web 上の情報を用いた企業間関係の抽出”，人工知能学会論文誌，Vol.22, No.1, pp.48-57 (2007)

(21) 奥健太，服部文夫：“地域限定性を考慮した情報推薦方式に関する基礎検討”，Web とデータベースに関するフォーラム，Vol.2009，No.3，1A-1 (2009)

(22) 亀井靖高，角田雅照，柿元健，大杉直樹，門田暁人，松本健一：“ソフトウェアコンポーネント推薦における協調フィルタリングの効果”，情報処理学会論文誌，Vol.50，No.3，pp.1139-1143 (2009)

(23) McCarey, F., Cinnéide, M.O. and Kushmerick, N.: “Knowledge reuse for software reuse,” Web Intelligence and Agent Systems, Vol.6, No.1, pp.59-81 (2008)