2015年度博士前期課程（ソフトウェア情報学）論文

プログラミング学習者の習熟度に応じたレコメンドを行う情報推薦システムの構築と評価

Construction and evaluation of a information recommendation system based on the proficiency level of the programmer.

岩手県立大学大学院
ソフトウェア情報学研究科

2016年3月

2312014030
藤田 拓

論文指導審査教授
澤本　潤

佐藤　裕幸

土井　章男

指導教員
杉野栄二

要旨

現代におけるプログラミングの学習環境はとても充実している．数年前までは，高度で専門な技術であったが，近年この認識は変わりつつあるからだ．コンピュータの小型化と一般化に伴って，プログラミングを行う環境を構築する事が非常に容易になった．いまや，PCさえあればプログラミングができる時代である．また，そのプログラミングによる成果物を公開する事が容易になった事も一因であると考えられる．アプリケーション配信プラットフォームであるGoogle Playは，その一つである．この登場によりアマチュアプログラマーが作成したアプリケーションを公開する事が容易になった．こうした状況で，プログラミングに対する認識が変わり，その学習環境も充実してきた．

プログラミングの学習環境の充実に伴って，プログラミングを学ぼうとする人々も増加傾向にある．あるプログラミング学習用の動画を提供するWebサービスの会員数は，2015年に15万名を超え，3ヶ月で1万名ペースで会員数を伸ばしている．こういった所からも学習者の増加と，その注目度の高さが伺える．Webサービス以外にも，プログラミング学習塾なども増えている．

しかし，こうした状況は，より多くの情報が存在するという事である．学習者には，多くの情報を得る事ができるというメリットと共に，多すぎる情報から適切な資料を選択する事が難しいというデメリットもある．本研究が行った調査によれば，プログラミングを学ぶ大学生のうち86%の人が，適切な資料を見つける事ができない場合があったと答えている．さらに，57%の人は，適切な資料を見つける事が出来なかったために学習自体を断念した事があると答えている．

そこで，本研究では，プログラミング学習者が良い資料を効率よく見つけるための支援を行う仕組み作りを目的とし研究を行った．良い資料を素早く見つけ，実際にプログラミングを行う時間を多く確保する事で学習効率の向上を目指す．その支援を行うために，学習者のレベルに応じたレコメンドを行う情報推薦システムの提案と構築を行い，その評価を行った．

Abstract

Programming of the learning environment is very fulfilling. Until a few years ago, it was highly specialized technology. However, this perception has changed. Environment for programming has become easy. By downsizing and the generalization of the computer. If there is even the PC, it can be programmed. In addition, it became easier to publish the artifacts. Google Play is an application delivery platform, is one of them.It has become easy to publish an application that amateur programmers have created.

In this situation, awareness of the programming has changed.And, also it has been enhanced the learning environment. People to learn programming tends to increase.

By downsizing and the generalization of the computer.Members of the service that provides video for learning, beyond the 150,000 people in 2015. (It has increased by 10,000 people in three months). A growing number of Web services and programming school.

However, this situation, a lot of information exists. Learner can get more information. But, the learner can not select the appropriate documentation from too much information. According to the survey of college students to learn programming, 86% of people can not find the appropriate documentation. In addition, 57 percent of people gave up the learning, because not be able to find the appropriate documentation. Therefore, the present study is to build a mechanism that the learner is to provide assistance in finding the appropriate documentation. Quickly find the appropriate documentation. And, increasing the programming of the time.Finally, it aims to improve the learning efficiency.

We have proposed and construction of a recommendation system base on proficiency of novice programmer for this purpose. And it was evaluated.

目次

第１章 序論----------8
１.１ 背景----------8
１.２ プログラミング学習に関する調査----------10
１.３ 本研究の目的----------12
１.４ 論文の構成----------12
１.５ 本章のまとめ----------12

第２章 プログラミング学習の支援----------13
２.１ 学習者についての仮説----------13
２.２ 現状の問題点----------14
２.３ 関連研究と関連技術----------15
２.３.１ 複数ドメインを利用した情報推薦手法----------15
２.３.２ 学習における情報推薦手法----------16
２.４ プログラミング学習サービス----------17
２.５ 本研究の位置付け----------17
２.６ 本章のまとめ----------18

第３章 情報推薦システム----------19
３.１ 概要----------19
３.２ 利用例----------19
３.３ システムの構成----------22
３.３.１ 学習コンテンツの収集----------23
３.３.２ 習熟度と難易度の推定----------24
３.３.３ 学習コンテンツ難易度の推定----------24
３.３.４ 学習者の習熟度の推定----------28
３.３.５ 情報推薦----------30
３.４ 本章のまとめ----------30

第４章 情報推薦システムの評価----------31
４.１ 評価実験概要----------31
４.２ 評価実験内容----------31
４.２.１ 実装までを含めた評価実験----------31
４.２.２ 実装を行わない評価実験----------32
４.３ 評価実験結果----------32
４.３.１ 実装までを含めた評価実験の結果----------32

4.3.2 実装を行わない評価実験の結果 ----------34
4.4 評価実験考察 ----------36
4.4.1 実装までを含めた評価実験の考察 ----------36
4.4.2 実装を行わない評価実験の考察 ----------36
4.4.2 評価実験全体の考察 ----------36
4.5 本章のまとめ ----------37
第5章 まとめ----------38
5.1 本研究のまとめ ----------38
5.2 結論 ----------38
5.3 本章のまとめ ----------38
参考文献----------39

図目次

図1.1 Google Trendによるプログラミング学習への注目度の動向----------8
図1.2 学習前の準備について----------11
図1.3 準備段階で適切な資料を見つける事ができたか ----------11
図3.1 提案システムイメージ----------20
図3.2 提案システム検索機能イメージ----------21
図3.3 ログインイメージ ----------21
図3.4 学習者とコンテンツのマッチングイメージ ----------22
図3.5 提案システム概念図----------22
図3.6 Webクローラー処理フロー ----------23
図3.7 資料の分類・難易度の推定 ----------25
図3.8 分類辞書の構築----------26
図3.9 具体的な機械学習の流れ----------27
図3.10 判定スコアの算出----------28
図3.11 Github上で確認できる情報の例----------29
図3.12 情報推薦の概要図----------30
図4.1 開発時間と調査時間の比較 ----------33
図4.2 点数の比較----------33
図4.3 実装を行う評価実験での発見率の違い ----------34
図4.4 情報推薦を行う提案システムを利用するグループの平均調査時間----------34
図4.5 情報推薦を行わない提案システムを利用するグループの平均調査時間----------35
図4.6 実装を行わない評価実験での発見率の違い ----------35

表目次

表２.１ 推定アルゴリズムの関連研究との位置付け ----------18
表２.２ サービス横断推薦手法の関連研究との位置付け ----------18
表４.１ 評価実験概要----------31

第１章 序論

本章では，本研究における序論について述べる．第１.１節では，本研究の背景について述べる．第１.２節では，プログラミング学習に関して行った調査について述べる．第１.３節では，本研究の目的について述べる．第１.４節では，論文の構成について述べる．第１.５節では，本章のまとめについて述べる．

１.１ 背景

2010年代までは，プログラミングは高度で専門的な技術とされており，一般の人々には一部の専門家の専門技術という認識だった．プログラミングを行うための環境を，一般の人々に用意することが難しい状況だったことも，その一因だと考えられる．しかし2010年代以降，この認識は変わりつつある．コンピューターの小型化や，コンピューターの家庭への普及に伴って，プログラミングを行う環境を準備する事が，以前よりも簡単になった．さらに，Android用アプリを配信するGoogle Playや，Amazon Android app store，iOS用アプリを配信するApp Storeなどのアプリケーション配信プラットフォームの登場によって，個人でもアプリ開発を行い世界中の人々に向けて，アプリケーションを配信することが可能になった．こうした状況の中で，プログラミングに対する認識は変わり，一部の専門家だけの技術では無くなりつつある

この流れの中で，現在，プログラミングを学ぶことに人々の注目が集まっている．Google Trendによるプログラミング学習に関する注目度指標は，図1.1のようになっている．

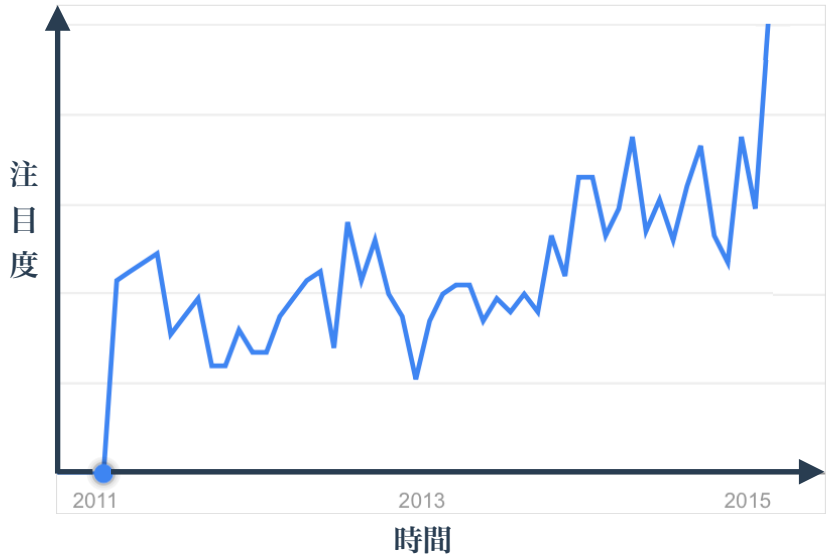

図１.１ Google Trendによるプログラミング学習への注目度の動向

この指標は，プログラミング学習に関する検索数や，関連するニュースの掲載数などによって算出されている．Google Trendによるプログラミング学習への注目度指標では，揺れ幅はあるものの注目度は上昇傾向にあることが分かる．

このような注目度の高まりの中で，2013年1月にアメリカでは，情報学の学習支援を目的としたcode.org[1]という非営利団体が設立された．この団体は，バラク・オバマ米大統領やマイクロソフト，アップル，フェイスブックといったIT企業最大手から，支援を受け大きな注目を集めた．さらに，マーク・ザッカーバーグ，ビル・ゲイツ，ジャック・ドーシーなどの起業家や各界の著名人がプログラミングを学ぶことの重要性について語るビデオを公開し，大きな反響を呼んだ．

日本では平成25年6月の産業競争力会議の成長戦略の素案[2]にて，プログラミング科目を学校教育に追加する事を検討している．プログラミングを学ぶことは成長戦略の重要な一部であると位置付けている．さらに政府レベルだけでなく民間レベルでも，プログラミングを学ぶ事ができるWebサービスや，従来のパソコン教室とは違ったプログラミングスクールが増加傾向にある．注目すべきは，このプログラミングスクールが，大人向けだけではなく，小学生などの子供向けスクールも同じくらい増加傾向にあることである．これは，前述の通り，プログラミングが義務教育となる可能性もあり，プログラミングを子供頃から学んで欲しいと考える親が増えているからだと考えられる．

一つのサービスのレベルには満たないが，Webに注目してみると，アプリケーションの作成で得た知見を公開したり，どのようにアプリケーションやシステムを作成するかについての知識を公開している個人や，企業が多く存在することが分かる．アプリケーション配信プラットフォームの登場と同様，そういった知識や知見を公開する場も増えていることが関係していると考えられる．QiitaやTeraTailなどの，プログラミングに関する知識を記録・共有するためのサービスや，プログラミングに特化したQ&Aサイトなどが，その最たる例ではないだろうか。他にも，プログラミング言語ごとのフォーラムサイトや，アプリケーションエンジンなどのコミュニティなども，その一つだろう．

こうした点を考えても，プログラミングに対する認識や，プログラミングを学ぶ環境は，以前と比較して整いつつある．例えば，学習者が学習を行いたい時に，参考資料が無くて困ることはほぼ無いと言えるのではないだろうか．もちろん，ある程度の人気のあるプログラミング言語や，その技術に限られるがあまり人気の無い言語を学びたい人は，少ないと考えられるので，ほぼ無いと言うことができると考えられる．

しかし，こうした状況は，より多くの情報が存在すると言う事になる．学習者側からしてみれば，数多くの情報の中から自分に適切な情報を選択する事が難しいと言える．本来であれば，実際に手を動かして学習を行うような演習が，プログラミングの学習において核となるにも関わらず，学習前の準備段階で，多くの時間を割かなけれ

ばならない．こうした準備の負担を減らすために，学習のための情報の取得や選択を手助けするような仕組みが，今後，必要になると考えられる．

また，情報を発信する側にとっては，発信する情報が，どの学習者に適切な情報なのかを明確にして発信しなければならない．例えば，プログラミング上級者向けの情報があるとする．しかし，初級者にしか閲覧されず，この情報は役に立たないと言われてしまう．役に立たなかったということで，上級者も閲覧しなくなってしまう．本来の対象者の上級者には，良い情報にも関わらず．このことからも，適切に情報を発信することは，学習者だけではなく，発信する側にも重要なポイントとなると言える．

1.2 プログラミング学習に関する調査

大学１年生から大学院２年生までのプログラミングを学ぶ人々，8人に対して調査を行った．その調査結果が図1.2と図1.3である．

図1.2より，学習前の準備や，学習中に何らかの調査を行う必要がある時に，その準備や調査について86%の人が時間を掛けたく無いと考えている．

図1.3より，学習前の準備や，学習中に何らかの調査を行う必要がある時に86%の人が準備段階で適切な資料を見つける事ができない場合があったことが分かる．また，57%は適切な資料を見つける事が出来なかったために学習自体を断念している．

この結果から，学習前の準備や，学習中に何らかの調査を行う必要がある時には，多くの人が手早く終わらせたいと考えていながら，資料を探す事が負担になっている事がわかる．さらに，学習自体を断念するなど，学習を始めるにあたっての障害になっているとも考えられる．

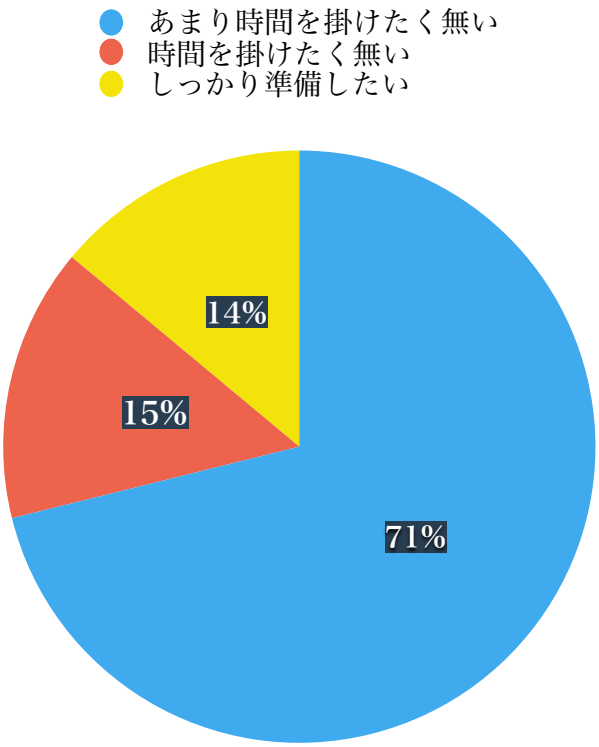

図１.２ 学習前の準備について

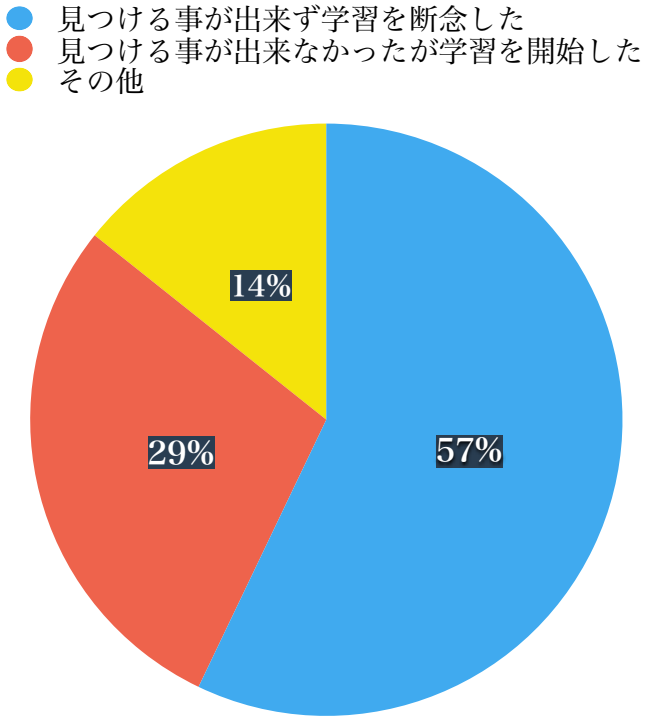

図１.３ 準備段階で適切な資料を見つける事ができたか

1.3 本研究の目的

本研究の目的は，プログラミング学習者が良い資料を効率よく見つけるための支援を行う仕組みを作ることである．良い資料を素早く見つけ，実際にプログラミングを行う時間を多く確保する事で学習効率の向上を目指す．また，良い資料を見つけやすくすることで，学習を断念することを防ぐこともできる．本研究における，良い資料とは，学習者の習熟度に適した資料や，学習者が欲しい情報を得る事ができる資料のことである．

本研究では，学習者のレベルに応じたレコメンドを行う情報推薦システムの提案と構築，その評価を行った．

1.4 論文の構成

本稿は，５章で構成される．

この第１章においては，研究背景や研究目的について述べた．次章からは，以下のように構成される．

第２章では，プログラミングを学習する上での詳細な問題点とプログラミングの学習を支援する目的で情報推薦技術を適用する事の問題点，プログラミングの学習支援の現状について述べる．

第３章では，推薦システムの提案と構築内容について述べる．

第４章では，第３章において提案，構築したシステムの評価について述べる．

最後に第５章では，本研究における結論を述べる．

また，評価実験により，利用者のレベルに応じて推薦する情報を最適化させることの実用性について示す．

1.5 本章のまとめ

本章では，序論として，背景，プログラミング学習に関して行った調査，本研究の目的と論文の構成について述べた．プログラミング学習者が増加している一方で，同様に学習のための情報も増加傾向にある．その結果として学習者にとっては情報の取捨選択が難しい状況になっている．そのため本研究では，プログラミング学習者が良い資料を効率よく見つけるための支援を行うことを目的としてシステムの提案と構築，その評価を行う．

第２章 プログラミング学習の支援

本章では，プログラミング学習ついて述べる．第２.１節では，プログラミングを学ぶ学習者について述べる．第２.２節では，プログラミング学習に関する現状の問題点について述べる．第２.３節では，本研究において関連のある２つの研究とその周辺技術について述べる．第２.４節では，一般的なプログラミングサービスについて述べる．第２.５節では，第２.３節で述べる関連研究と，本研究の関連とその位置づけについて述べる．第２.６節では，本章のまとめついて述べる．

２.１ 学習者についての仮説

第１.２節における，プログラミング学習に関する調査の調査結果から，学習におけるプログラマー(エンジニア)の傾向について仮説を立てた．その仮説をまとめた図が図２.１である．

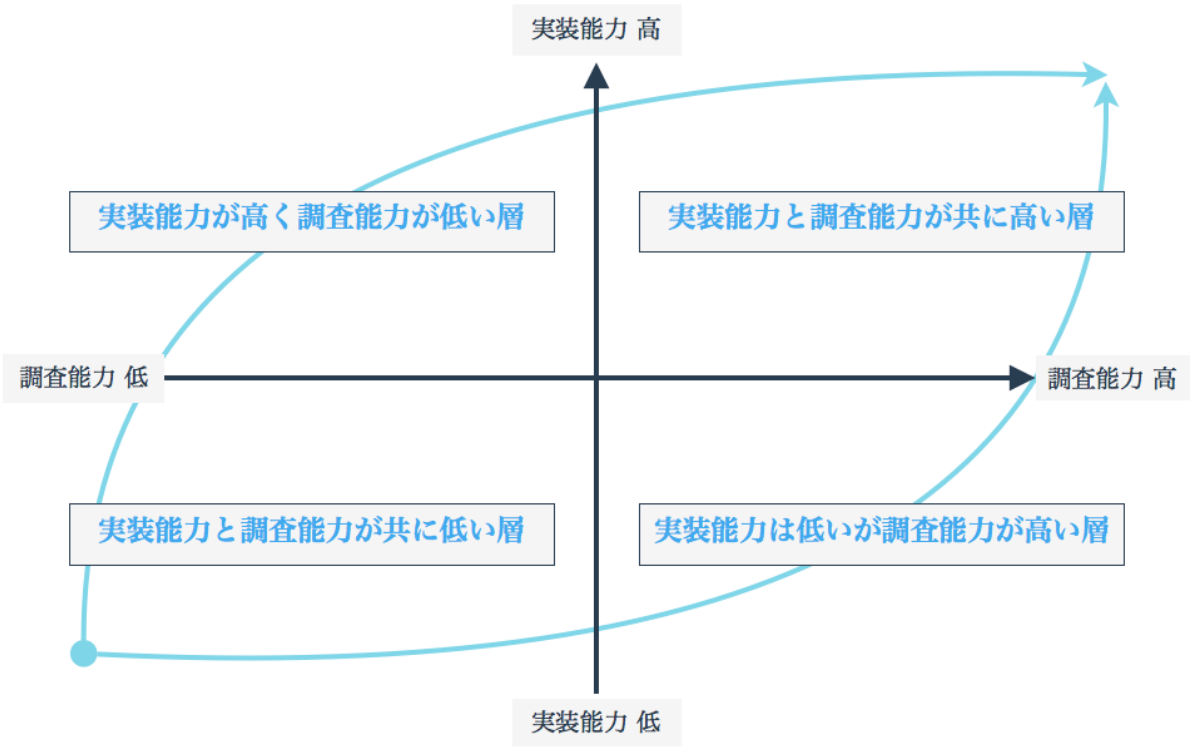

図２.１ 学習者のタイプ

図２.１では，プログラミングを学ぶ学習者について，学習前の準備や学習中に何らかの調査を行う必要がある時に，的確に情報を発見することができるかという調査能力と，実際にプログラミングを行う時の実装レベルでの能力としての実装能力に着目

した仮説を立てた．さらに，その仮説から学習者を4つに分け仮説を立てた．また，これらの学習者について，調査能力と実装能力が共に最も低い学習者を初心者とした時に，この初心者の成長パターンとして2つのパターンがあるという仮説を立てた．

これらの4つの学習者は，実装能力(プログラミング能力)と調査能力(学習前の資料を探す能力)が共に高い層は，プログラミング能力が高く何を調べればどのような情報を得られるかを理解している．そのため自分の得意分野だけで無く触れた事のない分野においても実装能力が高い層である．実装能力は低いが調査能力が高い層は，実務経験が少ないので実装能力は低い．しかし，調査能力は高くどう調べれば何を得られるか理解している層のことである．実装能力が高く調査能力が低い層は，実装能力が高いため自分の得意分野に関しては十分な力を発揮する．しかし，触れた事のない分野だと何をどう調べれば良いか理解していないので同様のレベルを維持する事が難しいという様な層である．最後に，実装能力と調査能力が共に低い層は，学習を始めたばかりで実装能力が低くプログラミング能力も低い．さらに，調査能力も低く何をどう調べれば良いか分からない．これから学習を始める初心者が属する層である．このように，それぞれの学習者を説明することができる．

このうち，本研究において対象とする層は，実装能力と調査能力が共に低い層と実装能力が高く調査能力が低い層である．調査能力が低い層は，学習前の準備段階に多くの時間を割かねばならないと推測される．そこで，この層を主な対象として支援を行うことで，最適な支援ができるのではないかという考えから対象として選択した．本研究では，　　図２.１の成長パターンにおいて，これらの学習者が，上方への成長を支援するというよりも，より右側へと移行できるような支援を行いたいと考えている．

２.２ 現状の問題点

本研究の対象層が抱える問題を噛み砕いて考えると，迷子問題とミスマッチ問題の２つの問題が考えられる．

迷子問題とは，学習分野に関する知識の不足による問題である．どう調べれば良いか，また，どう探せば良いか分からないので，適切な資料の選択や取得が困難であるという問題である．例えば，プログラミングにおいて何らかの要素を並べ替えるという様な課題に直面した時，ある程度のプログラミングの知識を有していれば，並び替え -> ソートの様な連想ができるはずである．しかし，学習分野に関する知識がないため，この連想ができないという状態のことである．プログラミングの分野は，関連する項目が非常に多い．プログラミング言語に限定した場合でも，ある言語は，他の言語の拡張言語であるとか，プログラミングで実現したい目的により向き不向きがある等々，様々なことが関連している．ある分野において上級者レベルでも，また別な

分野においては初心者レベルになる可能性もある．そう考えると迷子問題は，プログラミング初心者だけの問題ではないとも言える

もう一方のミスマッチ問題は，前述の迷子問題から派生する問題である．迷子問題で述べた通り，学習者は知識が不足している状態である．適切な資料を選択できず学習を開始した場合に，学習者と資料とのミスマッチ問題が発生する．例えば，初心者レベルの学習者が，高難度の資料を利用して学習を行っても理解する事ができず，学習の意味をなさないという状況である．知識が豊富な学習者が基本レベルの資料を読んでも何も得るものがなく意味がないという状況も考えられる．これは理解難易度の上昇や，学習効率の低下を引き起こす問題である．ミスマッチ問題に関しては，知識量の不足だけでなく，学習者が自身の客観的なレベルを把握できないことにも起因すると推測される．

これらの問題によって，学習者は学習開始の準備に多大な労力をかけなければならず，学習を始めるにあたり大きな障害になっている．また，これらの問題は学習中にも同様に発生する問題である．これらの問題に学習者が対処するには，自身の客観的なレベルを把握し，学習分野に対して多少なりの知識を持って，学習のための情報を選択する必要がある．しかし，実際には自身の客観的なレベルを把握することは難しい．これから学習をはじめようとする分野の知識を身につけることも，学習のための学習を行うということになるので，学習者への負担は大きくなると考えられる．

２.３ 関連研究と関連技術

本研究における目的の，良い資料を効率よく取得する支援を行う上でポイントとなる技術は，機械学習処理技術，複数のドメイン(サービス)を利用した情報推薦手法，学習における情報推薦手法である．これらに関する研究は多く行われてきた．本節では，本研究と特に関連が深いと考えられる，これらの技術に関連する研究と技術について述べる．

２.３.１ 複数ドメインを利用した情報推薦手法

Web上のあらゆるデータをリンクさせ再利用を試みるLinked Open Dataへの関心の高まりから，情報推薦においても複数のドメインのデータを利用した情報推薦手法に関する研究が注目されている[3]．深澤ら[4]は，複数のドメインによる情報推薦手法を“目的のサービスとは異なるサービスで蓄積された履歴を用いて，目的のサービス上でレコメンドを行う手法”と定義し，複数ドメインの情報推薦に関するサーベイを行っている．この定義は，一般的な定義である言える．本稿でも，これに従う．

Liら[5][6]の研究では，異なるドメインのデータを利用する事で対象ドメインの推薦システムの高度化を目的としている研究のため，書籍，映画等の一つ一つのジャンル

を，ドメイン(サービス)としている．また，Berkovskyら[7]は，Liらと目的は同じであるが，映画という特定分野の中においてのジャンルをドメインとしている

一方で，堤田と中辻ら[8][9][10]は，特定のドメインのユーザーの利用履歴を利用し，他のドメインでユーザーの嗜好を予測する事を目的としている研究である．ユーザーがすでに利用しているドメインの情報に基づいて，他のドメインのアイテムを推薦する．堤田ら[9]によれば，複数のドメインに渡る情報推薦手法では，ユーザーが未利用のドメインでの推薦が難しい事，複数ドメイン環境においてドメイン毎のデータ量の不均衡によって，アイテムとユーザーのログがより疎になる推薦精度の低下の問題がある．

しかし，これらが問題にならない環境下であれば，複数ドメイン環境による情報推薦技術は，ユーザーにとってメリットが大きい推薦手法だと考えられる．本研究では，ソースコード管理ツールのデータと提案システム上で収集したデータを利用して推薦を行いたいと考えているため，複数ドメインによる情報推薦手法を利用することとなる．

本研究においても，複数ドメインによる情報推薦を行うにあたり，これらの問題が依然として存在している．しかし，本研究の提案システムでは，ソースコード管理ツールの利用を必須としている．さらに，必須とする事によって，データ量の不均衡も制御することが可能である．よって，これらの問題点は，本研究においては問題にならない．

2.3.2 学習における情報推薦手法

学習コンテンツにおける情報推薦に関しては，これまでにいくつか研究がなされている．濱田と三好ら[1][11]の，どのレベルの誰が，どのレベルの何を読んだかを表す2部ネットワークを構築，分析し，学習者の習熟度と学習コンテンツ難易度の推定する手法についての研究もその一つである．濱田と三好らは，学習者の習熟度に適した難易度の学習コンテンツを推薦するという課題について，新たな提案と分析を行った

濱田と三好らは，ある分野のアイテムを多く読んでいるユーザーは，その分野について詳しい．詳しいユーザーに読まれているコンテンツほど難しく，詳しくないユーザーに読まれているコンテンツほど易しい．難しいコンテンツを読んでいるユーザーは詳しく，易しいコンテンツを読んでいるユーザは詳しく無いという3つの仮説と，ユーザーの学習コンテンツを読んだ順序に基づいて，ユーザー習熟度と学習コンテンツの難易度を相互に求めるアルゴリズムについて提案を行った．

濱田と三好らの手法では，学習者の習熟度を学習者が対象の学習分野に関して持つ知識量とし，学習コンテンツの難易度を学習コンテンツを理解するために必要な学習者の知識量としている．そして，習熟度と難易度が等しい時に最も多く知識を得られ

ると定義している．本研究における習熟度と難易度の定義は，濱田と三好らによる定義と同義である．

2.4 プログラミング学習サービス

前節，2.2節では，プログラミング学習に関するアカデミックな関連研究と関連技術について述べたが，近年では，プログラミング学習を英語などの学習と同様に，サービスとして提供する企業や団体も増えている．第1章で述べたcode.orgもその一つである．code.orgは，Web上で動画によって情報学を学ぶことができる．日本では，ドットインストール，Schooなどが，プログラミング学習コンテンツをメインコンテンツとして提供している．

これらの学習サービスは，サービスを提供しているWebページによると，3ヶ月で1万人ペースで，ユーザー数が増加しており，人々の注目度の高さが垣間みえる．

2.5 本研究の位置付け

関連研究と本研究との関連は以下の表2.1と表2.2の様になる．

濱田らの研究では，習熟度と難易度の推定を，誰が何を読んだかを表す読書ネットワークを構築し分析する事で行っている．また着目点も，どのレベルのユーザがどのレベルの本を読んでいるかである．対して，本研究では，推定手法には，機械学習処理(ベイジアンフィルタ)による分類による手法を用いる．着目点もプログラミング学習という点を考え，実際にプログラミングを行ったコード量とその質に着目する．

堤田らの研究では，複数ドメインでの情報推薦について他のサービスと自サービスの位置付けを，自サービスの情報を利用して他サービスの推薦に利用するとしている．情報量の多寡への対処方法は，共通情報(ユーザ,アイテム)を利用する事で対処している．本研究では，他サービスの位置付けを，他サービスの情報を利用して　自サービスの推薦に利用するとして，情報量の多寡への対処は，他サービスの同時利用を必須にすることで対処している．

表２.１ 推定アルゴリズムの関連研究との位置付け

習熟度と難易度の推定アルゴリズムについて		
	本研究	濱田ら
推定手法	機械学習処理(ベイジアンフィルタ)による分類と推定	誰が何を読んだかを表す読書ネットワークの分析による推定
着目点	プログラミング量とその質	どのレベルのユーザがどのレベルの本を読んでいるか

表２.２ サービス横断推薦手法の関連研究との位置付け

サービス横断推薦手法について		
	本研究	堤田ら
他サービスの位置付け	他サービスの情報を利用して自サービスの推薦に利用する	自サービスの情報を利用して他サービスの推薦に利用する
情報量の 多寡への対処	他サービスの同時利用を必須にする	共通情報(ユーザ,アイテム)を利用し対処

２.６ 本章のまとめ

本章では，プログラミング学習ついて述べた．前章の調査に基づき，学習者について仮説をたてた結果，もっとも支援が必要となるのは，調査能力が低い層の学習者である．この層の学習者は，学習を行う上で大きな問題を抱えている．その問題を解決するために，本研究との関連が深い研究から，本研究においては，問題に対してどのようにアプローチするのか，関連研究との位置付けと関連を交えながら述べた．

第３章 情報推薦システム

本章では，本研究における提案システムについて述べる．第３.１節では，提案システムの概要について述べる．第３.２節では，提案システムのイメージと共に利用例について述べる．第３.３節では，システムの構成について，学習コンテンツの収集，習熟度と難易度の推定，学習コンテンツ難易度の推定，学習者の習熟度の推定と情報推薦の部分に分け，それぞれについて述べる．第３.４節でまとめについて述べる．

３.１ 概要

本研究で構築する情報推薦システムは，ここまでに述べてきた様な問題を解決するためのシステムである．Web上に数多く存在する学習に用いるコンテンツを収集し，機械学習処理により難易度で分類する．そして，ユーザーとなる学習者の習熟度(レベル)も共に分類する．ソースコード管理ツールによって管理されるソースコードと，その更新差分を機械学習処理によって習熟度を分類する．二つをそれぞれのレベルに分類する事ができたら，そのレベルに基づきコンテンツと学習者をマッチングさせる事でコンテンツの推薦を行う．こうすることで，前述の目的であった選択支援を行う．

本提案システムが，対象とする学習者は主に初心者から中級者までの学習前，もしくは，学習中の資料を探す作業を負担に感じる学習者である．副次的な対象としては，収集するコンテンツや分類手法を調整する事で，上級者も対象とすることが可能であると考えている．

3.2 利用例

本提案システムのメインページのスクリーンショットを図3.1に示す．

基本的に，ユーザーとなる学習者は，このページにアクセスする事で情報を取得する．ページにアクセスしただけの状態では，システムが収集した記事をカテゴリ(プログラミング言語)毎に，新着順で表示する．図3.1は，phpカテゴリを表示している様子である．また，カテゴリを任意に選択する以外に，上部に設置された検索バーより任意の値を入力する事で検索する事ができる．その様子を図3.2に示す．この例では，検索バーに「javascript」と入力する事で，JavaScriptに関する記事を表示している．

提案システムで，収集したコンテンツを難易度毎に分類することは，システムがコンテンツを収集する毎に自動で行うので，ユーザーである学習者が気にする必要はない．しかし，学習者を習熟度(レベル)毎に分類する機能に関しては，ユーザーのソースコード情報が必要となるためログインが必要である．提案システムのログインに関

する部分のスクリーンショットを図3.3に示す．提案システム外部のソースコード管理ツールのoauth認証を利用する事でログインを行う．

ログイン認証を行い学習者の分類処理が完了すると，図3.4のように学習者とコンテンツがマッチングされ，学習者に最適化コンテンツが表示される．さらに表示されているメインコンテンツを読むために必要になると考えられるコンテンツもあわせて表示される．

ユーザーとなる学習者は，このようにログイン機能や検索機能などを使う事で，情報を得る事ができる．

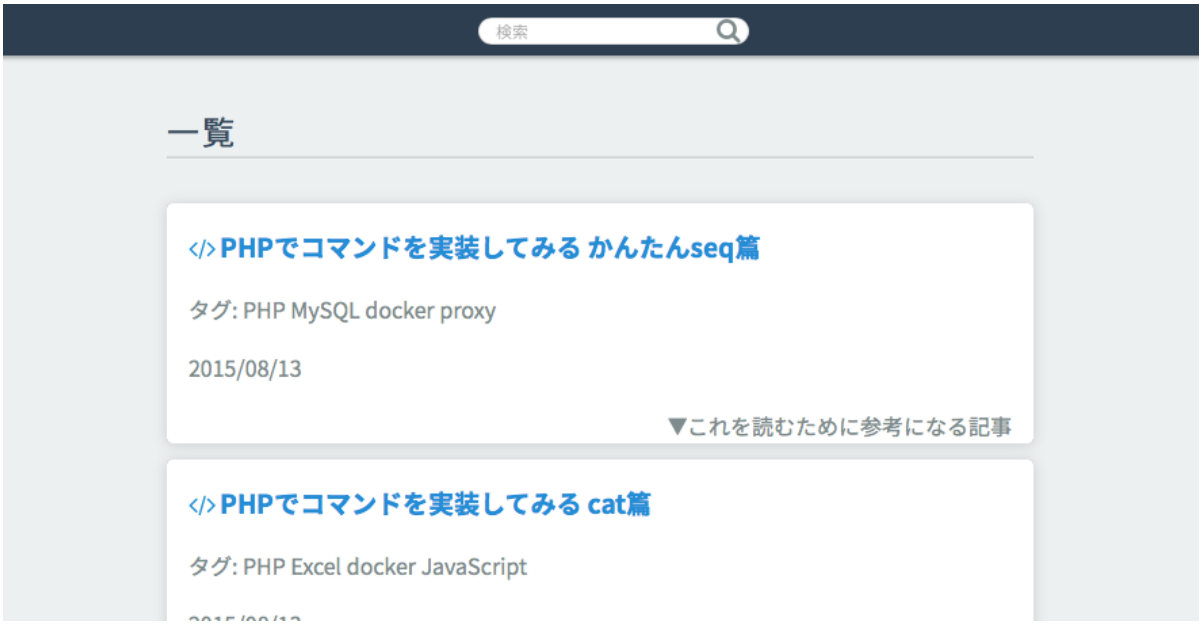

図３.１ 提案システムイメージ

図３.２ 提案システム検索機能イメージ

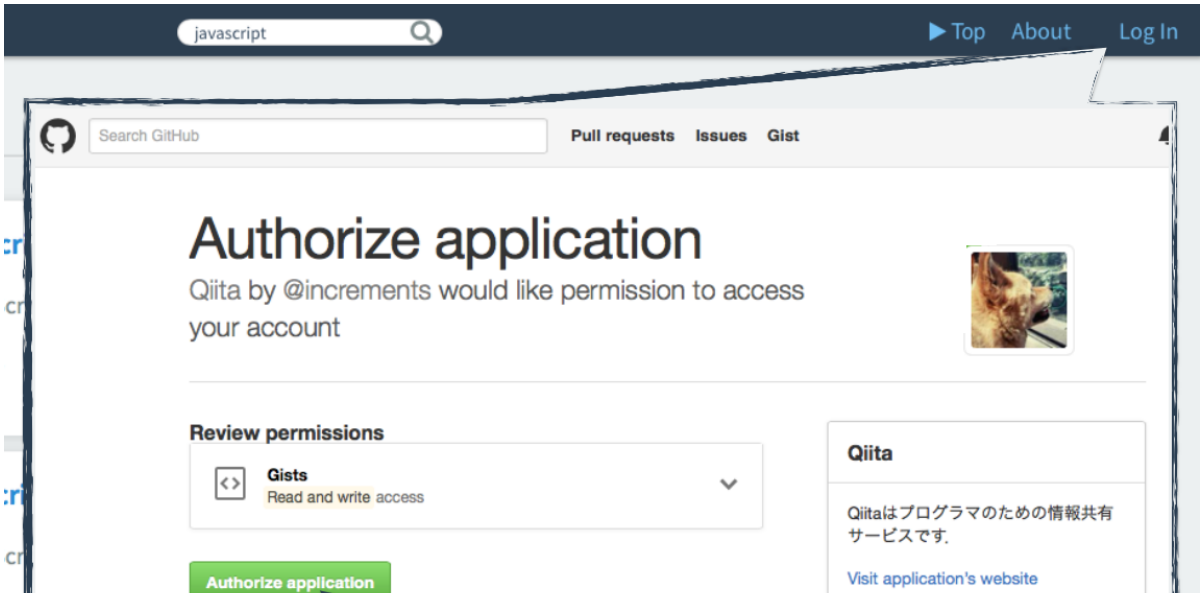

図３.３ ログインイメージ

図３.４ 学習者とコンテンツのマッチングイメージ

３.３ システムの構成

本節では，システムの構成について述べる．図3.5に提案システムの概念図を示す．学習コンテンツの収集，習熟度と難易度の推定，学習コンテンツ難易度の推定，学習者の習熟度の推定と情報推薦の部分に分けて述べる．

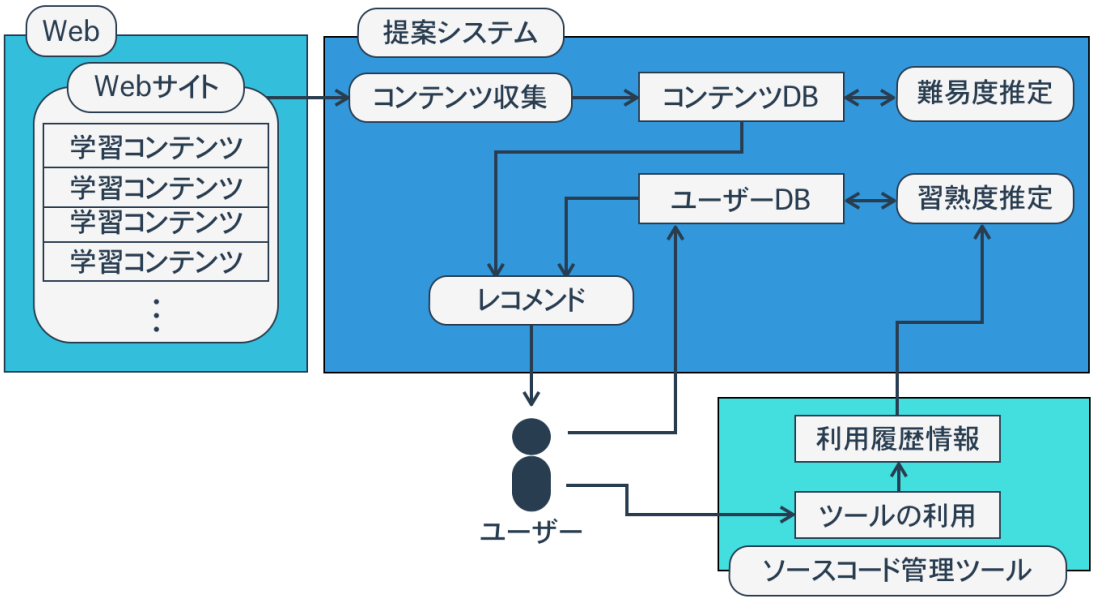

図３.５ 提案システム概念図

3.3.1 学習コンテンツの収集

本システムにおいて，情報推薦を行うコンテンツをWeb上から収集するWebクローラー部分について述べる．Webクローラーの動作フローを図3.6に示す．

まず，クローリング対象Webサイトのコンテンツ一覧ページにアクセスし，そのページのhtml文を全て取得する．そして，html文のパースを行い必要なデータをスクレイピングする．最後に，取得したデータから不要または不正なデータを削除し正規化を行い，記事データベースに追加する．これを対象Webサイトで一定期間毎に行い最新情報の取得を行う．

スクレイピング対象は，タイトル，コンテンツURL，記事概要，日付，コンテンツに付与されているタグ情報，お気に入り数である．記事概要はWebサイト側で付与されたものがあれば取得する．日付は，そのコンテンツが公開された日付である．コンテンツに付与されているタグ情報は，コンテンツにタグやカテゴリ等の情報が付与されていれば，それを取得し，無い場合はタイトルや記事概要から切り出して付与する．お気に入り数は，ストック数，ブックマーク数等，対象Webサイトにより呼称は異なるが本提案システムにでは，お気に入り数として取得しておく．

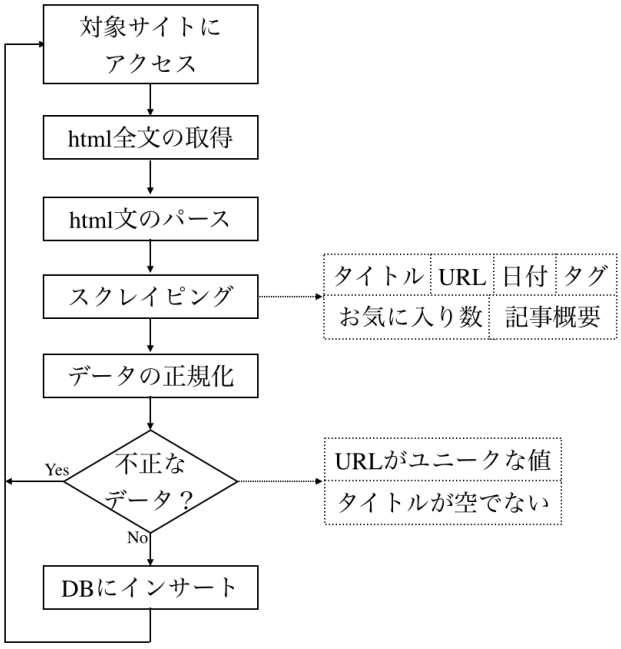

図3.6 Webクローラー処理フロー

データの正規化ルールは，タイトルが空で無い事，コンテンツURLが重複しないユニークなURLである事，日付が古すぎない事の3つである．対象がWebコンテンツで

あるため，タイトルで重複を管理するよりも，URLで管理する方が適当であると言える．日付情報に関してはプログラミング等の分野は，そのライフサイクルの変化が早い分野であるため，2015年8月を基準に3年前の2012年8月から現在まの最新のコンテンツを収集する事とした．

なお，このWebクローラーは推薦する学習コンテンツをWeb上から収集する事のみ行う．収集した学習コンテンツの難易度推定は行わない．また，その難易度推定においてテキストマイニングを行うためにコンテンツの本文が必要となる．しかし，コンテンツのURLを利用すれば本文の取得は容易であるため，このWebクローラーでは本文を取得せずデータベースへの保存も行わない．なおコンテンツが保存されるデータベースのテーブル構成は，ID，タイトル，URL，記事概要，日付，タグ情報，お気に入り数，難易度となっている．この段階では，難易度には何も保存されていない．後述の難易度の分類・推定時に難易度の保存を行う．

3.3.2 習熟度と難易度の推定

習熟度と難易度の定義は，濱田と三好らによる定義に倣う．推定手法に関しては，それぞれに機械学習処理を用いて推定を行う．この推定手法は，ある程度の量のデータが蓄積されていた方が効果を発揮できる．システムがコンテンツを収集する毎，学習者がソースコードを更新，追加するたびに，機械学習処理を行えるような設計と構成にし精度の向上を試みる．次から，難易度と習熟度について述べる．

3.3.3 学習コンテンツ難易度の推定

濱田と三好らの定義は，対象が読書である．そのためコンテンツの難易度は，そのコンテンツを読むために必要な知識量としている．本研究では，この必要な知識量について，あるプログラミング言語を学習する時に，そのプログラミング言語の知識だけでは理解できず他分野の知識が必要となる時(例:ある言語においてデータベースアクセスを行う時，その言語と使用するデータベースの知識が必要となる)により多くの知識量が必要であるという仮説に基づいた定義とする．

学習コンテンツの難易度の推定，分類を行う流れは，図3.7のようになる．

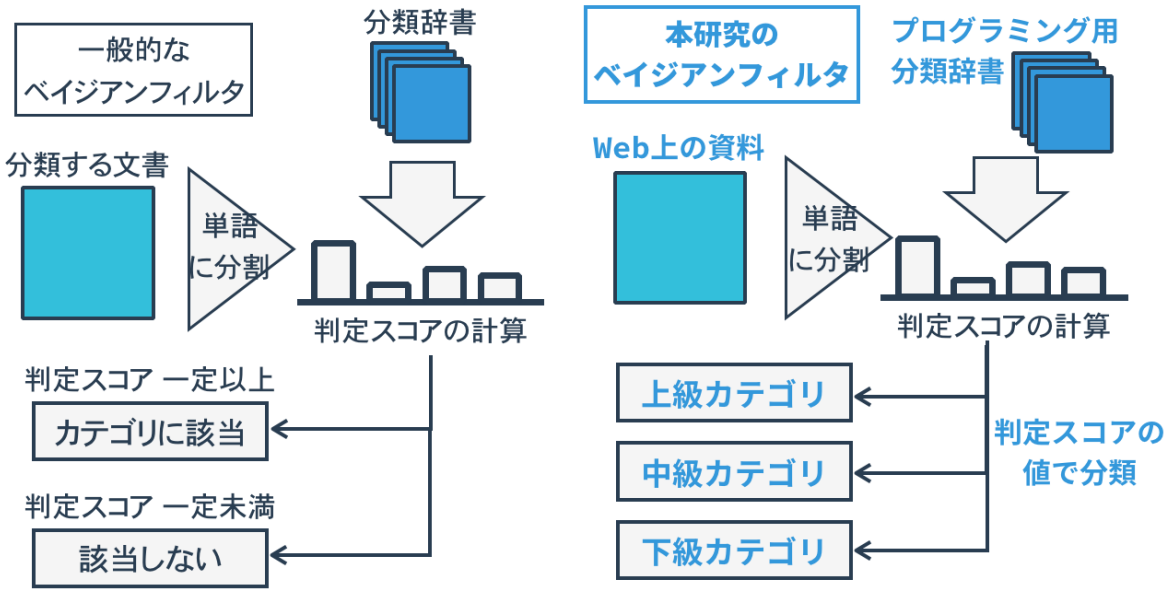

図３.７ 資料の分類・難易度の推定

本研究で用いる機械学習アルゴリズムは，ベイジアンフィルタを用いる．このアルゴリズムは，主に迷惑メールの分類などに用いられるアルゴリズムである．機械学習を行うことによって，ある文書があるカテゴリに属する確信を深めていくような機械学習アルゴリズムである．一般的なベイジアンフィルタ処理の流れは，まず，形態素解析処理によってコンテンツの文章を単語に分割する．その単語を分類辞書と呼ばれる単語とスコアの対応表から，単語とスコアの対応をとり単語を数値スコアに変換する(分類辞書については後述する)．そして単語の数値スコアを計算処理し，判定スコアを算出する．その判定スコアが一定以上の値かどうかによって，そのコンテンツがそのカテゴリに属するかどうか判定するというような流れである．本研究で用いるベイジアンフィルタは，基本的な流れは踏襲しつつ，プログラミング用の分類辞書や難易度をカテゴリーとする事で，コンテンツの難易度の推定と分類に対応している．

システムの分類辞書の構築について図3.8に示す.

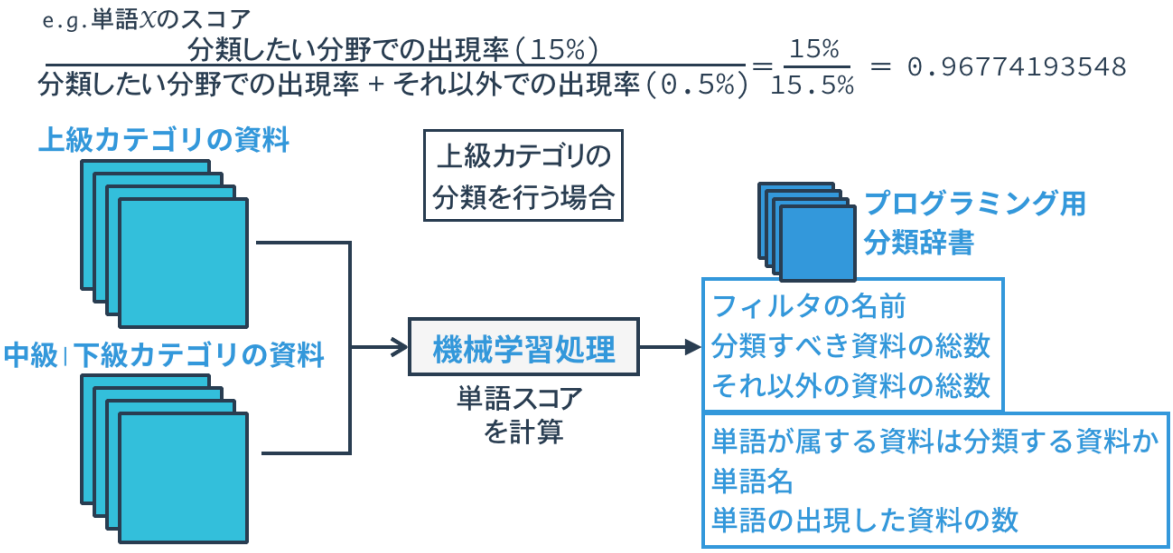

図３.８ 分類辞書の構築

分類辞書の作成は，初期学習用に予めカテゴリごとに分類された文書を用いて行う． 図3.8は，上級カテゴリの分類辞書を作成する例である．まず，上級カテゴリーに属する文書を形態素解析処理によって単語に分割し，それぞれの単語の出現回数をカウントしていく．これをそのカテゴリに分類されている資料全てで行う．その資料の総数等は，全て記録しておく．上級カテゴリの処理が完了したら，それ以外のカテゴリおいて同様の処理を行う．今回の例であれば，中級カテゴリと下級カテゴリになる．そして，全ての処理が完了したら，単語の出現回数や総数などから単語スコアを計算する．

これは，分類したい分野での出現率 / (分類したい分野での出現率 + それ以外での出現率)によって算出される．それぞれの単語の出現回数などを保存しているのは，学習用の文書の増加などに対応するためである．

具体的な流れを図３．９に示す．

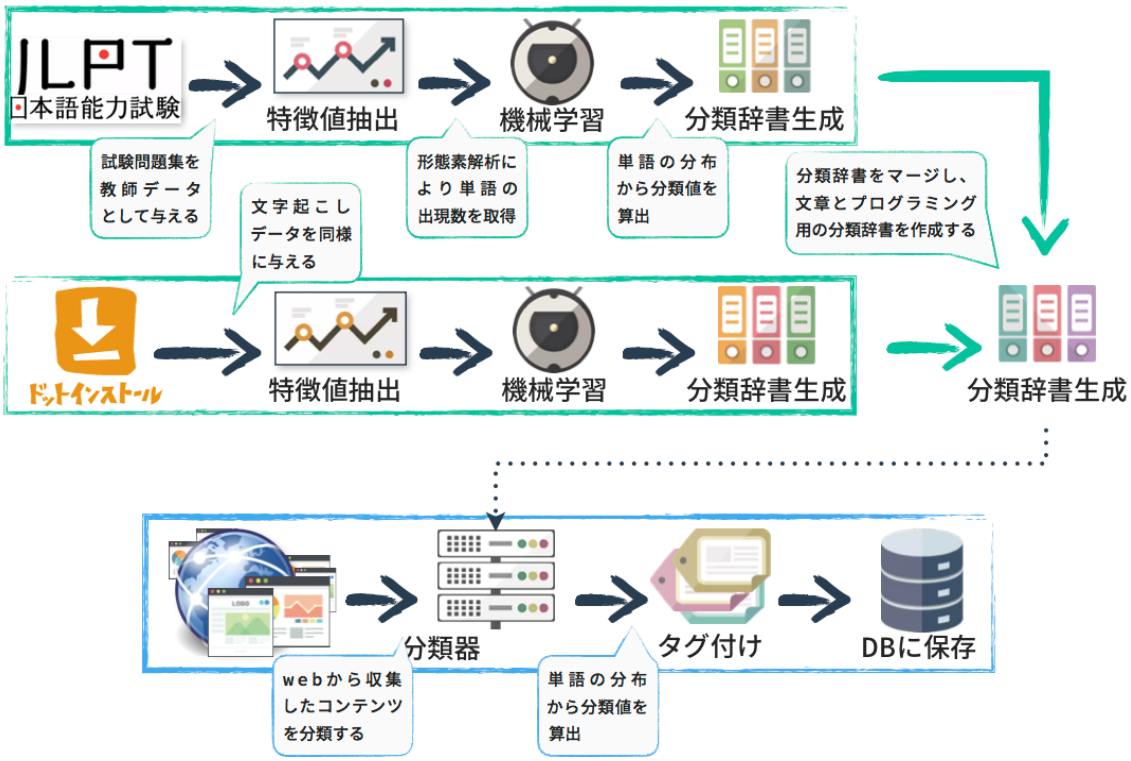

図３．９ 具体的な機械学習の流れ

具体的な機械学習のながれとしては，まず，日本語としての文章の難易度を求めたいので，日本語能力試験の問題集を教師データとして，機械学習処理を行い分類辞書を作成する．これで文章の難易度を特徴値として分類する分類辞書が一つできあがる．次に，本研究においては，プログラミング学習に関する情報を分類したいので，ドットインストールと呼ばれる，3分でプログラミングを学べる動画サービスの文字起こしデータを同様に教師データとして与える．これによりプログラミングに関わる情報の難易度を特徴値として分類する分類辞書が完成する．ここまでで，二つの分類辞書を作成した．次は，この分類辞書をマージすることで，日本語としての難易度の特徴値とプログラミングに関する情報の難易度を特徴値として分類する分類辞書を作成する．この分類辞書を用いて，Web上から収集してきた情報を分類する．

図３.１０に，判定スコアの算出手法に関する図を示す.

図３.１０ 判定スコアの算出

判定スコアとは，前述の図3.7で述べたような，その資料がどの難易度に属するかを表す確率である．図３.１０は一例である．この例では，判定スコアが0.9855となっており，このスコアが算出できたら，後は，このスコアが，分類したいカテゴリに属するための基準を満たしているかをチェックし，満たしていれば分類する．これを，それぞれのカテゴリに対して行う.

３.３.４ 学習者の習熟度の推定

難易度の定義は，濱田と三好らに倣った．習熟度に関しては，本研究は，実際にプログラミングを行った量に着目し，コード量とその質をもって習熟度とする.

具体的な学習者の分類，習熟度の推定の流れと手法は，ほぼ同じである．与えるデータと分類辞書に用いるデータが学習コンテンツでは無く，ソースコード管理ツールによって管理されている学習者個人のソースコードを使用する点でのみ異なる.

本研究で，利用するソースコード管理ツールは，Github社により運営されるWebツールのGitHub(https://github.com)である．Githubでは，Githubアカウントの情報を取得する事ができるGithub APIを提供している．通常はアカウントの公開情報しか取得する事ができないが，APIで認証を行う事で，非公開情報を取得する事もできる．このAPIを経由し学習者のプログラミング量の情報等を取得する．取得できる情報としては，どの言語でプログラミングを行っているかや，GitHubへのプログラムのアップロード数等のContributions情報，前回アップロードからのプログラムの差分情報のCommits情報を取得する事ができる．図３.１１に，GitHub上で確認する事のできるContributions情報とCommits情報を示す.

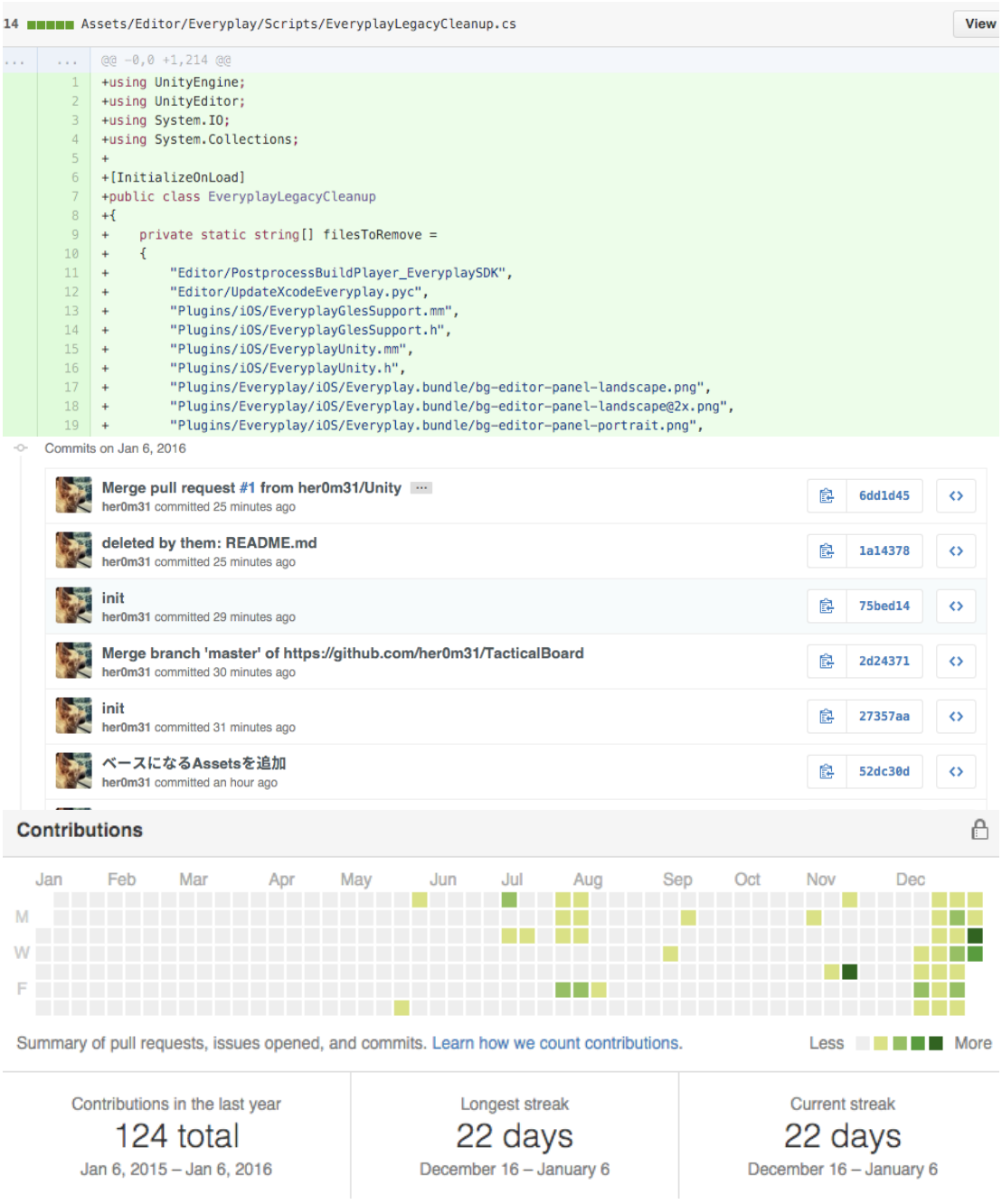

図３.１１ Github上で確認できる情報の例

3.3.5 情報推薦

それぞれ，コンテンツと学習者の分類と推定を行った後で，情報推薦を行う．本提案システムにおける情報推薦手法は，非常に単純で，それぞれ同じレベルのカテゴリから推薦を行う．それ以外に，学習者の習熟度がそれぞれのレベルカテゴリの境界に近い場合は，他のカテゴリーからの推薦も行う．

情報推薦の概要図を図３．１２に示す．

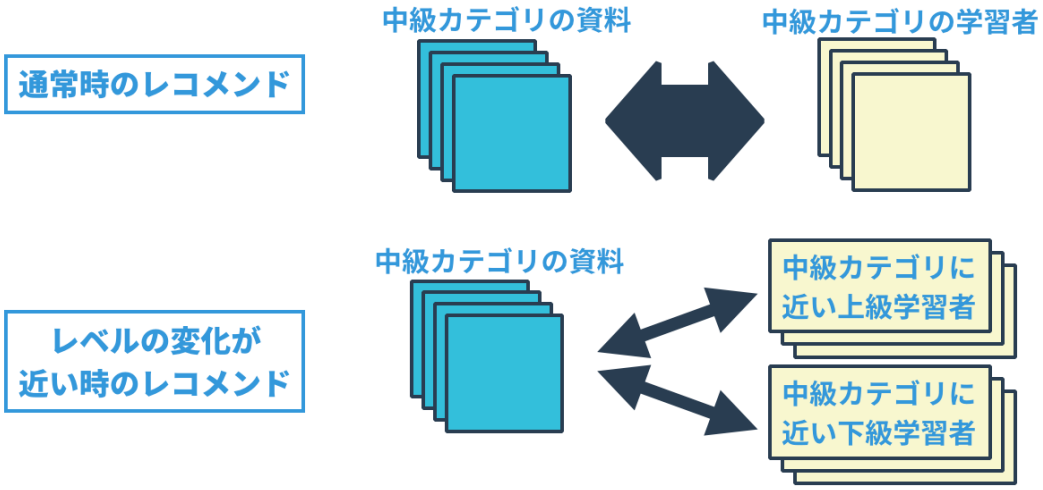

図３．１２ 情報推薦の概要図

3.4 本章のまとめ

本章では，本研究における提案システムについて述べた．本研究における目的を達成するために，学習コンテンツの収集，習熟度と難易度の推定，学習コンテンツ難易度の推定，学習者の習熟度の推定と情報推薦を柱として提案システムの構築を行った．

第４章 情報推薦システムの評価

本章では，本研究における提案システムの評価について述べる．第４.１節では，提案システムの評価実験概要について述べる．第４.２節では，評価実験の内容について述べる．第４.３節では，評価実験の評価基準について述べる．第４.４節では，評価実験結果について述べる．第４.５節では，評価実験結果の考察ついて述べる．第４.６節で，本章のまとめについて述べる．

４.１ 評価実験概要

被験者に課題を提示し，課題を行うグループをシステム利用グループと通常グループに分け，その過程と結果について評価を行った．この実験は，学習者のレベルに応じて推薦する情報を最適化する事の効果と，初級者と中級者が上級者と比較して経験や知識量に差がある事に関して，その差をどの程度システムによって埋める事ができるかを評価，調査するための実験である．

４.２ 評価実験内容

本節では，評価実験の内容について述べる．評価実験は，目的別に２回行った．一つは，実際にプログラミングを行った場合の実装までを含めた評価実験である．もう一つは，より調査時間にフォーカスした実装を行わない評価実験である．

４.２.１ 実装までを含めた評価実験

被験者に提示した課題内容は，90分間でCRUDを最低要件とする掲示板アプリの作成である．ここでのCRUDとは，投稿，編集，削除，一覧表示の事である．表4.1に環境を示す．そして，課題中に閲覧した記事や，閲覧するために使用したキーワードなどを全て記録してもらう．さらに調べるために使用した時間と，開発にかけた時間を記録して提出してもらう．なお被験者数は，上級者2名，中級者２名，初心者２名の内訳となっている．

表４.１ 評価実験概要

開発言語	PHP
開発環境	cloud9(Cloud IDE)

評価手法については，被験者に提出してもらった課題と課題中に記録されたデータを分析し評価を行う．課題の採点手法に関しては，CRUDの一つの機能を25点とする．CRUD全てを実装することができて100点となる．そして，実装レベルではソースコードを見て加点・減点を行う．

4.2.2 実装を行わない評価実験

被験者に提示した課題内容は，CRUDを最低要件とする掲示板アプリの作成すると仮定した場合に必要となりそうな資料を可能な限り収集するという課題である．想定する開発環境は，表4.1の環境と同様である．そして，課題中に閲覧した記事や，閲覧するために使用したキーワードなども同様に全て記録する．制限時間は30分としており，被験者が十分と判断した場合には時間内でも終了しても良いこととした．なお被験者数は，上級者2名，中級者６名，初心者６名の内訳となっている．上級者に関しては，この評価実験では，　参考程度のため２名としている．

評価手法については，被験者に記録してもらった課題中に記録された閲覧データと情報を収集した結果，被験者が必要と判断した資料を分析して評価を行う．

4.3 評価実験結果

本節では，２つの評価実験の結果について図表を用いながら述べる．

4.3.1 実装までを含めた評価実験の結果

図４.１，図４.２，図４.３に，実装までを含めた評価実験の結果を示す．それぞれの図表では，実験結果の平均値を示している．

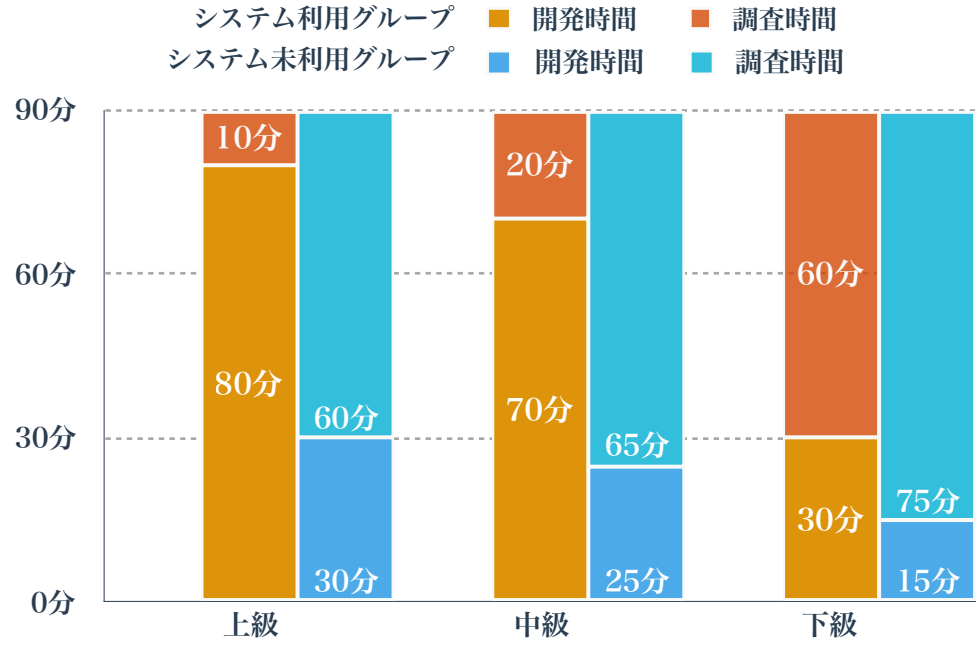

図４.１ 開発時間と調査時間の比較

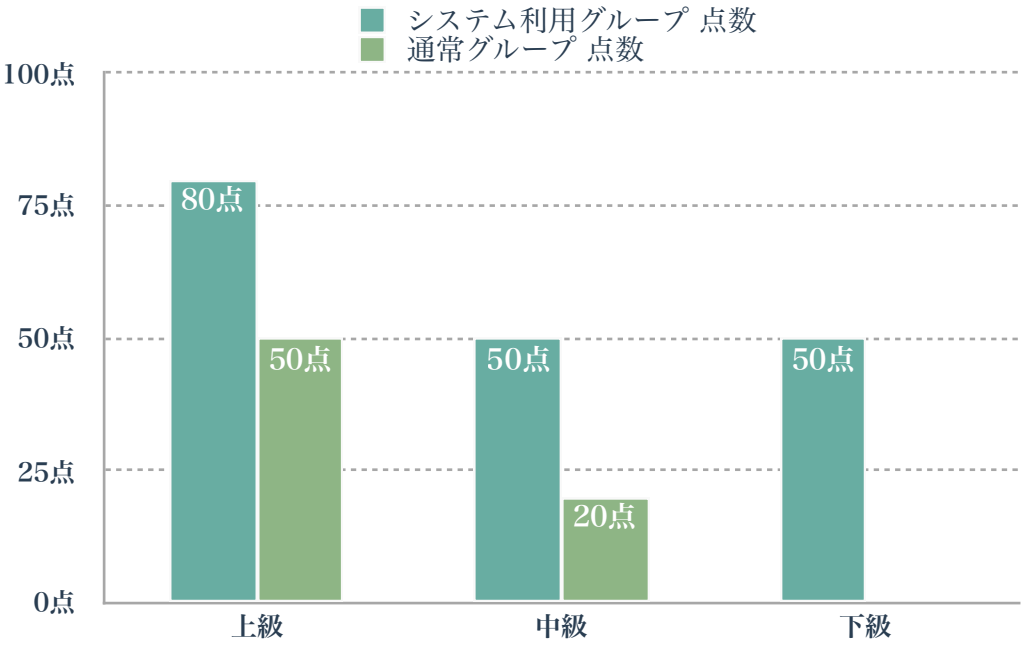

図４.２ 点数の比較

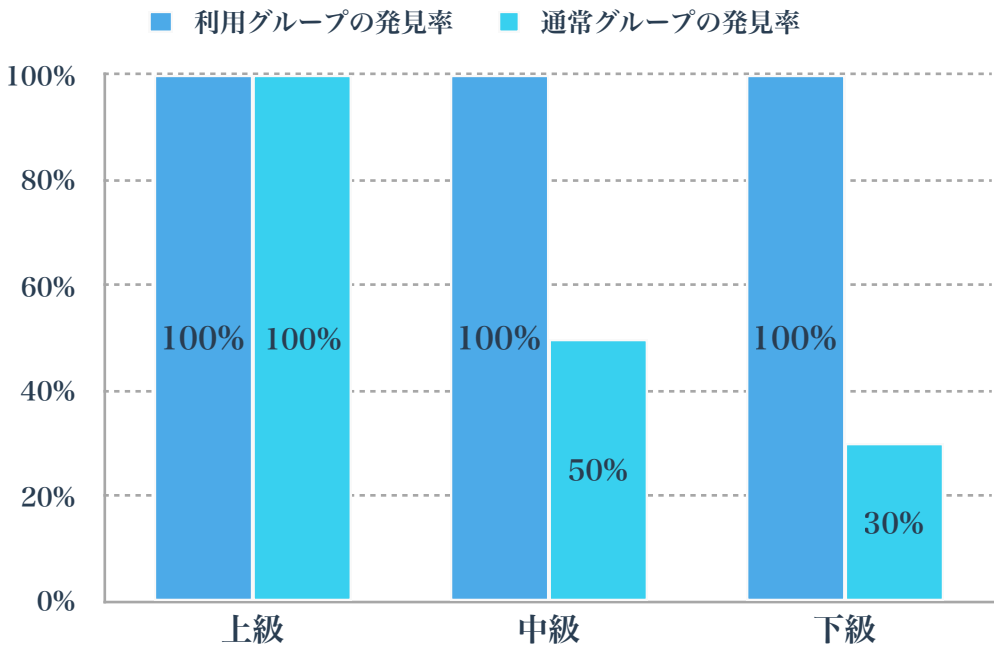

図４.３ 実装を行う評価実験での発見率の違い

４.３.２ 実装を行わない評価実験の結果

図４.４，図４.５，図４.６に，実装を行わない評価実験の結果を示す．それぞれの図表では，実験結果の平均値を示している．

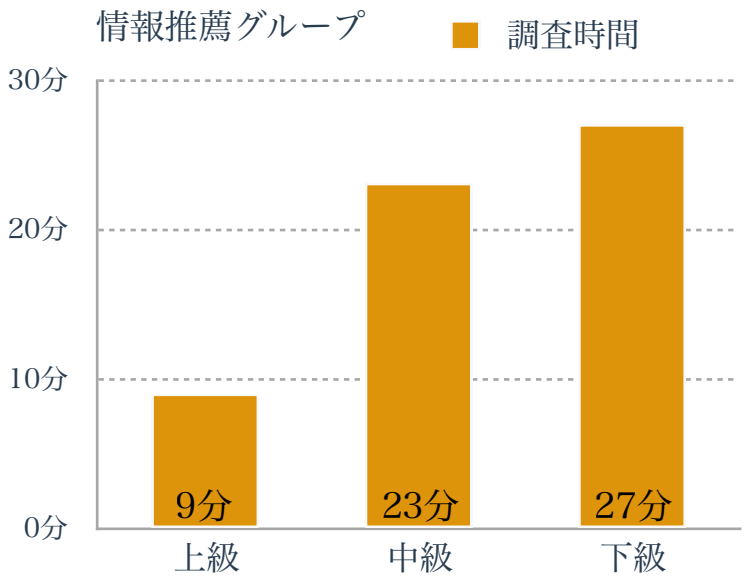

図４.４ 情報推薦を行う提案システムを利用するグループの平均調査時間

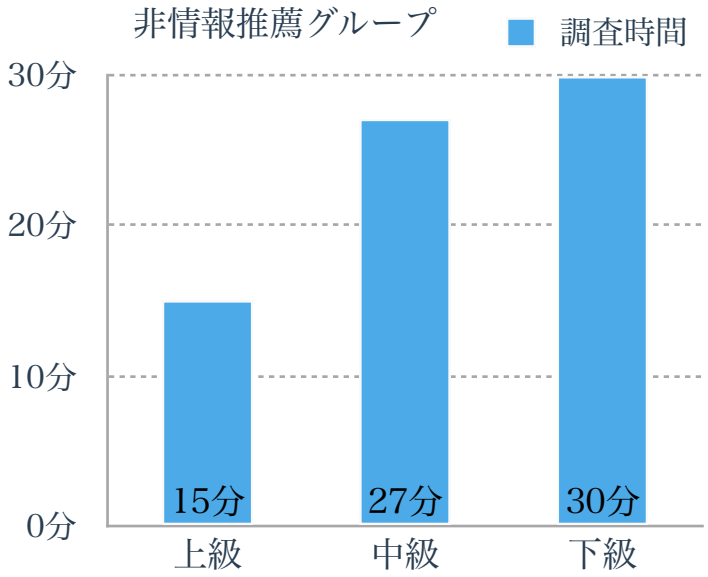

図４.５ 情報推薦を行わない提案システムを利用するグループの平均調査時間

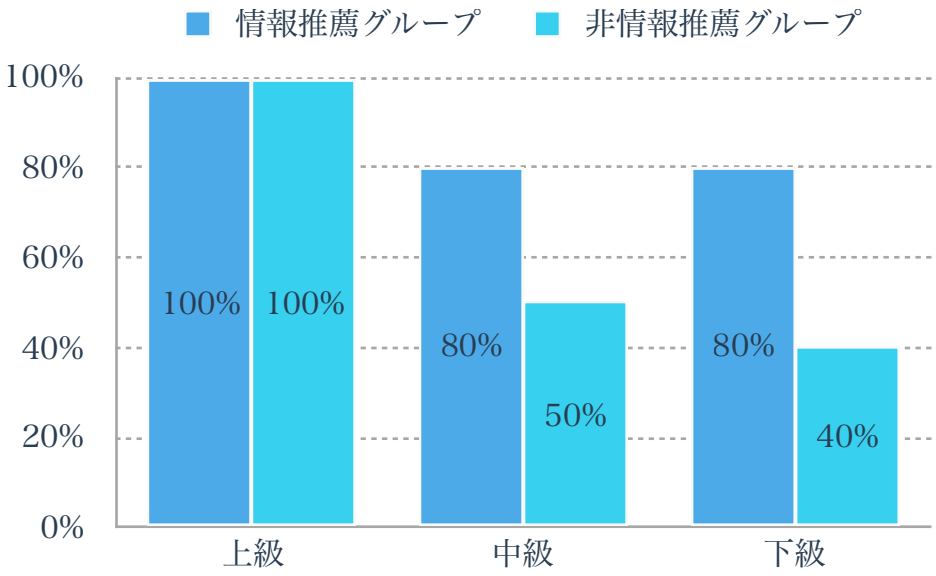

図４.６ 実装を行わない評価実験での発見率の違い

４.４ 評価実験考察

本節では，評価実験の考察を述べる．行った評価実験の考察をそれぞれ述べ，評価実験全体の考察を述べる．

４.４.１ 実装までを含めた評価実験の考察

図4.1のグラフについては，いづれの被験者についても，調査時間を短縮し開発時間を多く確保する事ができている．図4.2に関しては，いづれも30点以上の差がついており効率良く質の高い資料を見つける事ができている．図4.1と図4.2の結果から，レベルに応じて推薦する情報を最適化する事は一定の効果があると言える．また，図4.3は，一度の検索で自分の問題解決に役立つ資料を発見できるかの結果である．また，発見率は，通常は経験や知識量など実力による所が多い．図4.3の結果を見えると，提案システムは，その経験や知識量による差を補う事が出来ていると言える．

４.４.２ 実装を行わない評価実験の考察

前述の通り，実装までを含めた評価実験では，実装にフォーカスした評価実験なので，被験者数が少ない状況だった．そこで，実装を行わない評価実験では，本研究における主な対象層である中級者，初心者の被験者数を大幅に増加して実験を行った．図４.４，図４.５は，調査時間の比較である．評価実験の時間が短縮されてしまったために，それぞれの差は小さくなっている．しかし，実装までを含めた評価実験の結果と同様に，それぞれのレベルの被験者に対して，調査時間を短縮することができている．図4.6の発見率についても同様である．それぞれの差は，小さくなっているが，提案システムのサポートによって，発見率を向上させることができていると言える．

４.４.２ 評価実験全体の考察

今回，行った二つの評価実験を統合して考察する．いづれの実験結果についても，調査時間の短縮に成功している．これは提案システムのサポートによって，学習者の負担となっていた，学習の準備に関わる時間を短縮できるということを意味している．これにより本研究での目的であった資料を効率良く見つけるサポートができていると考えられる．さらに，一度の検索で自分の問題解決に役立つ資料を発見できるかの指標である発見率についても，いづれの実験結果においても，発見率の向上に成功している．これは見つけた資料が良い資料かどうかを表す指標であるとも考えられる．この二つの結果を統合して考えると，本研究で目的としていた，プログラミング学習者が良い資料を効率よく見つけるための支援を行うという目的は達成できていると言える．

４.５ 本章のまとめ

本章では，本研究における提案システムの評価実験ついて述べた．本研究における目的を達成できているか評価を行うために，評価実験の設定，評価実験結果の分析を行った．さらに，その結果についての考察を述べた．

第５章 まとめ

本章では，本稿のまとめについて述べる．第５.１節では，本研究のまとめについて述べる．第５.２節では，本研究の評価実験から得られた結果を踏まえ結論を述べる．第５.３節では，本章のまとめについて述べる．

５.１ 本研究のまとめ

本研究では，学習者の習熟度と学習コンテンツの難易度とのミスマッチ問題を解決し，効果的な学習を支援する事を目的とした，機械学習処理を主とした情報推薦システムの提案と構築を行った．また，構築した提案システムは，２回の評価実験を行った．その評価実験によって提案システムの効果を測定し，一定の効果があることが分かった．

５.２ 結論

本研究の目的を達成するために，情報推薦システムの提案と構築を行った．提案システムでは，機械学習アルゴリズムを利用して，ユーザとコンテンツを分類することで，情報推薦機能を実現した．構築したシステムは，評価実験により，良い資料と見つける事と効率良く資料を探す事が両立できている事が分かった．これにより本研究の，質の高い資料を効率良く見つけだす支援という目的は達成できたと言える．また，今回、ユーザとコンテンツを分類して，情報推薦を行った事で，学習者のレベルに応じて推薦する資料を最適化させるという点において，一定の効果がある事も分かった．本研究によって，実証された効果は，必ずしも多くはないが、若干なりともプログラミング学習の効率化に寄与できたと考えられる．

なお，本稿で構築したプロトタイプのソースコードは，CCライセンスで公開中である．

５.３ 本章のまとめ

本章では，本稿と本研究のまとめについて述べた．

参考文献

(1) Code.org, “Anybody can learn | Code.org”, https://code.org/ .

(2) 産業競争力会議, “成長戦略(案)”, http://www.kantei.go.jp/jp/singi/keizaisaisei/skkkaigi/dai11/siryou1-1.pdf, June.2011.

(3) 濱田一伸, 三好康夫, 鈴木一弘, 塩田研一, “2部ネットワーク分析によるユーザ習熟度とアイテム難易度の推定アルゴリズムの提案, ”教育システム情報学会第37回全国大会講演論文集, no.A6-1, pp.384-385, Aug.2012.

(4) 大田昌幸, 江田周平, “協調フィルタリングを利用した推薦システム築, ”Ventures Salonvol.6 http://www.slides hare.net/masayuki1986/recommendation-ml, Apr. 2015.(2015/8/3 参照)

(5) 大河原渉, 牧山宅矢, 大野祐, 森田武史, 杉山岳弘, 山口高平, “Linked Dataの構造分析に基づく高速道路立ち寄り推薦サービス, ”人工知能学会全国大会論文集, 28巻1G4-OS-19a-7, no.1-4, 2014.

(6) 深澤佑介, 太田順, “クロスドメインの情報推薦に関するサーベイ, ”第1回Webインテリジェンスとインタラクション研究会, ARG W12, no.25, Dec.2012.

(7) B. Li, Q. Yang and X. Xue, “Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction, “IJCAI, pp.2052-2057, Jul.2009.

(8) S. Berkovsky, T. Kuflik and F. Ricci, “Cross-Domain Mediation in Collaborative Filtering, ”User Modeling 2007, pp.355-359, July.2007.

(9) 堤田恭太, 中辻真, 内山俊郎, 藤村考, “ソーシャルメディア上のアイテム共起を用いたサービス横断推薦, ”第3回データ工学と情報マネジメントに関するフォーラム 論文集, F7-3, Feb.2011.

(10)堤田恭太, 中辻真, 内山俊郎, 戸田浩之, 内山匡, “アクセスログを用いたクロスドメイン環境における情報推薦, ”情報処理学会研究報告, Vol.2012-DBS-154,No.4 Vol.2012-IFAT-107,No.4, Aug.2012.

(11)中辻真, 藤原靖宏, 内山 俊郎, “ユーザグラフ上のランダムウォークに基づくクロスドメイン推薦, ”人工知能学会論文誌, 27巻, 5号, no.296-307, Mar.2012.

(12)三好康夫, 濱田一伸, 鈴木一弘, 塩田研一, 岡本竜, 金西計英, “学習コンテンツ推薦に向けた 2 部ネットワーク分析に基づく習熟度と難易度の推定手法の提案, ”第1回Webインテリジェンスとインタラクション研究会, ARG WI2 No.1, Dec.2012.

(13)Yao, Y. Y., “Measuring retrieval effectiveness based on user preference of documents, ”Journal of the American Society for Information Science, Vol.46, No.2, pp.133-145, 1995.

(14)大田昌幸, 江田周平, 安藤祐貴, 清水磨, “推薦システムを構築する手順書 with Azure Machine Learning, ”Geek Salon, http://www.slideshare.net/masayuki1986/azure-machine-learning-49034325, Jul.2015.(2015/8/3 参照)