修士論文

学生の注目箇所に基づく
科目を超えた関連ページの発見

2021年度

印部 太智

明星大学大学院
情報学研究科 情報学専攻
20MJ-003

修士論文

学生の注目箇所に基づく
科目を超えた関連ページの発見

2021 年度

印部 太智

2021 年度 明星大学大学院 博士前期課程
情報学研究科情報学専攻
丸山研究室 20MJ-003

概 要

大学のカリキュラムは 1，2 学年のときに基礎科目を学習し，3，4 学年では応用科目を学習する体系を取ることが多い．情報系の学部では，学生は AI やセキュリティなど現実的な課題に興味をもって入学してくると考えられるが，それらの詳細を学ぶのは応用科目に配当されている．そのため，基礎科目を学習中の学生は現在の内容が自分の興味分野に応用されるのかイメージできず，学習のモチベーションを維持できないと考えた．本研究では，自習中の学生がスライドを用いて学習中であることを前提とし，学習中の授業資料から関連する科目の内容を学生に提示するようなシステムを想定する．学生がスライド内のある 1 ページに注目している場合に，そのページと関連する学習済みの内容や今後学習する内容を説明した具体的なページを科目を超えて示す．これによって，学生が現在の学習は自分の興味分野に必要な知識であることを把握し，学生のモチベーションの維持につながると考えた．本研究の目的は，基礎科目を自習中の学生を対象に，注目しているページと関連する，過去や将来の学習内容を説明した具体的なページを発見するためのスコアリング手法を提案することである．本研究では TF-IDF を用いて，スライドのページごとに，スコア付けを行った．加えて，学生に提示する具体的なページを発見するために，授業スライドが持つ特徴を重視し，時系列，重み付け，正規化の手法を導入しスコアを補正する．このスコアを降順に並び替え，上位を学生に提示する予定である．Stanford University の Computer Science コースの科目の中で Web ページから入手可能な 11 科目を対象に，有効なスコアリングを評価する実験を行った．時系列のみ，時系列と正規化，時系列と正規化と重み付けを施した 3 つのスコアリング手法を評価するために，2 種類の実験を行った．1 つは筆者による実験で，筆者が思う最適なページを 3 枚を選び，各科目ごとにシステムが算出したページに出現した割合と，筆者が選んだページを 1 位から 3 位のランクをつけた正解のラベルとしシステムが該当するページを発見してきた場合には 1 位から順に得点を与える Q-measure の，2 種類の評価を行った．もう 1 つは，被験者による実験で，すべての科目の結果から得られたページと同様のページを用いて被験者にランキング作成してもらい，それぞれの被験者の結果に基づく Q-measure を求めた．これらの実験から，時系列と正規化と重み付けを行った補正が，最も有効な結果が得られていることが示された．スコアリングに対して，過去や未来という区分けを十分に検討できていないため，最適な手法を検討していく．

目 次

1　序論

1.1　背景

大学のカリキュラムは 1，2 学年のときに基礎科目を学習し，3，4 学年では応用科目を学習する体系を取ることが多い．明星大学の入学者向けに配布される「2019 年度の履修の手引き」[1] の情報学部の項では，1 学年目に「プログラミング演習」や「基礎情報数学」，「基礎解析学」といった情報学を学ぶ上で必要な基礎科目を，2 学年では「アルゴリズムとデータ構造」や「コンピュータアーキテクチャ」など，1 学年の科目を踏まえた情報科学の基礎的な科目を，3 学年以降は「コンピュータネットワーク」や「ウェブプログラミング」など応用的な科目を履修する．例えば学生が，AI やセキュリティに興味がある場合，それらに直接的に関わる科目として，3 学年の「人工知能」や「情報セキュリティ」などを履修することが想定される．一方，これらには 2 学年で履修する「確率統計学」や「情報数学」，「コンピュータアーキテクチャ」や「情報通信工学」などの科目の知識が必要となる．しかし，2 学年の学生にとっては，2 学年で履修する内容が 3 学年で「人工知能」や「情報セキュリティ」に発展すると実感を持って学習することは難しい．加えて，学生が意味のない学習だと思ってしまうと，その科目の学習に対するモチベーションの低下や大学からのドロップアウト[1]につながることも考えられる．本研究では，学生のモチベーション低下を抑えるために，学生に発展的な科目の内容を示すことが有効であると考えた．

学生が発展的な内容を支援する手法として，シラバスとカリキュラムマップを関連づける仕組みがある [2]．カリキュラムマップは学科内の科目を学年ごとに整理したものである．しかし，シラバスやカリキュラムマップから把握可能な情報と実際の授業の内容では，差異が生じる．例えば 1 学年後期に配当されている「プログラミング演習 2」は，シラバス上は#12 ので「参照」について扱うことになっていて，C 言語のポインタについての内容を扱っている．この科目で初めてポインタを学習した学生にとっては，これがどのように活用されるのか実感が湧かないことがある．さらに，ポインタは C 言語の中でも学習が難しいとされている部分であり，学習内容を整理できないまま学習を続けると学生の学習モチベーションが低下してしまうことが考えられる．ポインタの内容では配列や文字列などを合わせて扱うことになる．これは「プログラミング演習 1」や「プログラミング演習 2」の#11 のまでで学生が学習した内容である．ある科目からすでに学習した内容を示すことで，すでに学習して理解している内容と結びつけて学習ことができる．これを 2 学年で履修する「コンピュータアーキテクチャ」や「アルゴリズムとデータ構造１」および「アルゴリズムとデータ構造２」で扱うメモリの構造や stack，heap と合わせて理解することで，モチベーションの低下を抑えることにつながると考えている．また 3 学年で履修する「コ

[1] さまざまな理由により大学を途中退学すること

ンピュータネットワーク」や「ネットワークコンピューティング」でもネットワーク通信の実装を行う場合，ポインタの知識は必須である．このように，ある科目から発展する科目の内容を示すことは，学生とって理解が難しい内容の発展する部分を理解し，その内容を挫折せずに学習することにつながる．本研究では，基礎科目を学習中の学生に対して，その科目の内容と関連する授業科目の具体的な資料を示すことによって，学生のモチベーションを維持したり，現在の学習を整理することができると考える．

学生が学習した内容の復習や現在の内容を整理するために，ノートや授業で使用された資料などを用いることが考えられる．特に授業資料は，教員が作成し Learning Management System(LMS) に公開している．LMS 上で教員が公開したデータや学生の提出物，学生の資料の閲覧情報などはサーバに蓄積され，Learning Analytics(LA) に活用される．LMS 以外にも Open Course Ware(OCW) と呼ばれる，大学がオンライン上で公開しているフリーの学習資料や，Open Educational Resources(OER) と呼ばれる，Web 上で公開されている資料や動画なども分析に活用されている [3][4]．本研究では LMS や OCW などで公開されている授業資料のうち，スライドの資料を用いて学習支援を行っていく．スライドの資料を用いる理由は，教員が授業の進行のために作成することもあるため，スライド内のそれぞれのページで行っている具体的な説明を学生に提示できると考えたためである．また，LMS の内部では特定の学年の資料だけではなく，大学全体で公開された資料が蓄積されている．そのため，他の学年の資料も入手可能である．

本研究では，スライドの資料のうち特に教員が授業の進行に用いるものに限定する．1 回の授業ではその授業内の到達目標に対する 2，3 のトピックを扱うことになる．そのトピックはさらに細かい詳細な解説要素に分割できる．スライドの資料であれば，そうした細かい解説要素 1 つに対して数ページ扱うことができるため，1 つのページにおける説明の内容は簡潔になる．そのため，自習中の学生に関連するページとして提示する情報として適していると考えた．学生が注目した内容と合わせていくつかのページをピックアップすることで，学生は提示された内容の中身を整理することができると考えている．そこで本研究では図 1 のようなシステムを想定する．このシステムでは，学習中の内容に関連して，学習済みの科目のページやこれから学習する科目のページを示す．学生が現在学習中の内容は，配列を使って木構造を管理するアルゴリズムを説明している．この中から学生は **tree** という単語に注目していると仮定する．システムは学生が注目した **tree** に基づいて，科目を超えて関連するページを示す．図 1 では，示す内容は 1 ページとしているが，実際には数ページ同時に示す．学習済みの科目のページからは，二分木やディレクトリの構造を示している．これにより，学生は木構造そのものの内容を思い出すことができる．これから学習する科目のページからは，コンパイラや自然言語処理で用いるパースツリーの内容が示されている．コンパイラや自然言語処理に興味がある学生にとってこれらの情報を示す

ことは，興味がある分野で必要な知識を知ることにつながる．そうではない学生にとっても，今後発展的に学習する内容を示すことで，科目の対策につながる．

図 1 は，以下の 3 つのステップで構成することができる．ステップに関して図 2 を示す．

ステップ 1 学習中のページから単語 (名詞) をすべて取得する．本研究ではページ中に含まれる単語に注目する．これは，ページ内の重要な要素は文章で説明されていると考えているためである．その中の名詞を取得することで，そのページの特徴的な単語が取得できる．

ステップ 2 それぞれの単語に基づいて，その単語が含まれる全ての科目の全てのページに対してスコアリングを行う．図 2 では，**tree** に注目した場合である．

ステップ 3 ステップ 2 のスコアリングに基づいて，学生に対してページの提示を行う．図 2 では 1 ページの提示を行っているが，実際には数ページ分の提示が可能である．

1.2 目的

本研究の目的は，基礎科目を自習中の学生を対象に，注目しているページと関連する，過去や将来の学習内容を示した具体的なページを発見するために，すべての科目のページをスコアリングすることである．これは図 2 の中のステップ 2 に該当する．本研究では，スコアリングの手法として TF-IDF を採用する．TF-IDF に対して授業資料の特徴を考慮するために，時系列や重み付け，正規化を導入し，スコアの補正を行う．これらのスコアに基づいて，関連するページを学生に示す．学生は注目しているページに関連するページを示されることで，学習済みのページからこれまでの学習内容を振り返り現在の学習について整理することができる．これから学習するページを示すことで，現在学習中の内容がどのように応用されるのか知り，学習意欲の向上や興味のある分野の発見につながる．このように，大学のカリキュラムでの学習全体を俯瞰することができる．

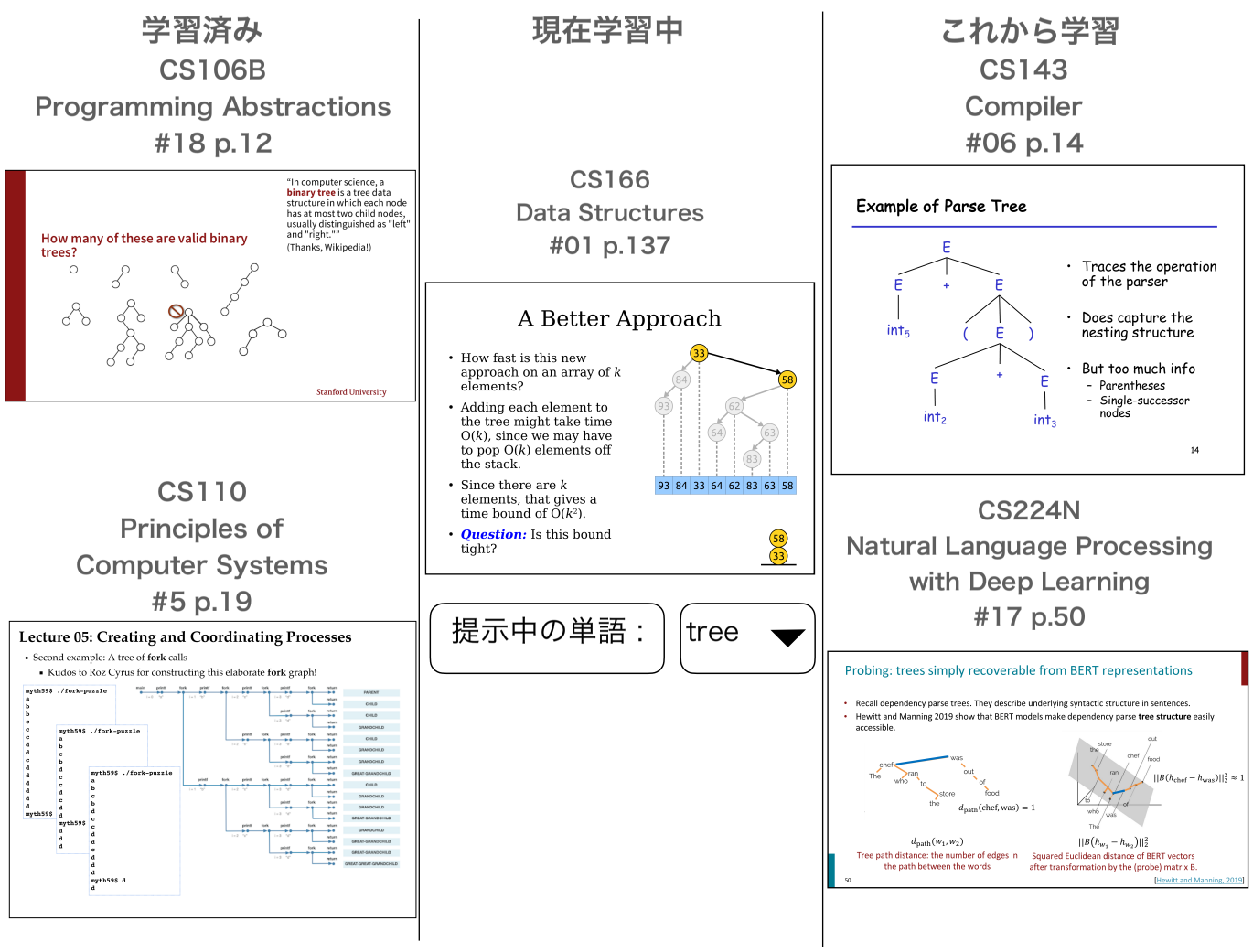


図 1: システムが提示する内容の例 (CS106B[5], CS110[6], CS149[7], CS143[8], CS224N[9] から引用)

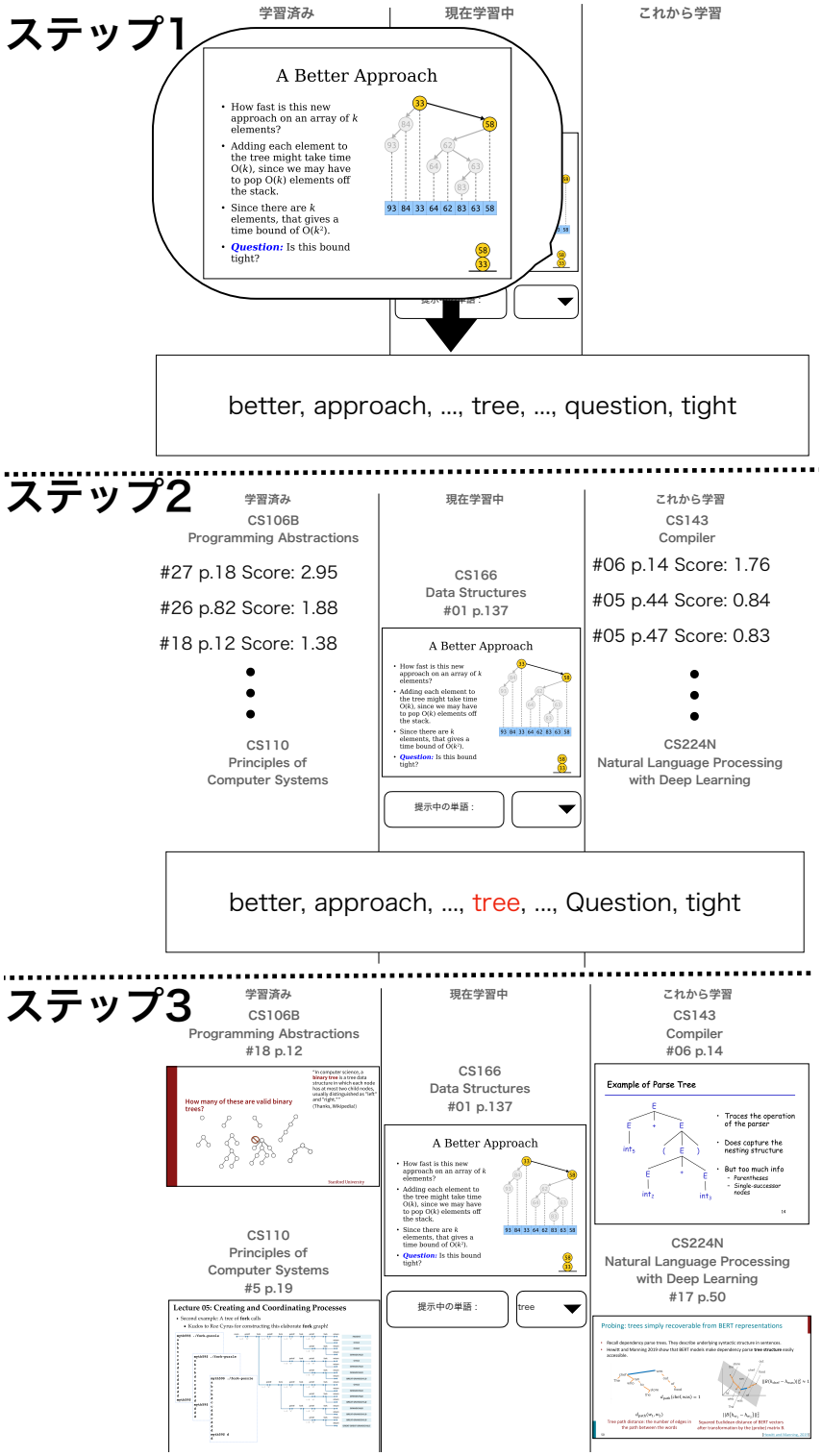


図 2: 図 1 のステップ (CS106B[5]，CS110[6]，CS149[7]，CS143[8]，CS224N[9] から引用)

2　関連研究

2.1　教育ビックデータ分析・分類

Educational Data Mining(EDM) や LA の分野では，学生の学習状況や提出物，資料など LMS や授業用の Web システムなどから取得できる情報を活用し，学生と教員に支援を行っている．Romero ら [10][11] は，2010 年と 2020 年に EDM 分野で行われている研究について調査している．この調査の中で学生に対する支援に焦点を当てたものも多くあり，学生個人の学習状況に基づいて関連する科目の情報や適切な学習スタイルの提案が可能であるとしている．本研究の目的としている内容も EDM や LA の分野の一部である．しかし，推薦システムとして関連する科目を提示する試みはあるものの，現在までに資料に基づいて過去や未来の関連した科目やその内容を示す試みはない．

Tavakoli ら [4] は，SkillCommons[2] と YouTube[3] の動画の中から教育向けのものについて，URL や説明文，再生時間などのメタデータを使って，メタデータの品質評価と予測を行った．この研究では，動画が持つメタデータ自体の分析を行っているが，ここから得られるデータを活用することで，授業で用いた資料を補うような動画の発見が可能になる．本研究では動画やそこから得られるメタデータについては対象としていないが，学習で用いるコンテンツを扱うという意味では類似している．

Goncharow ら [12] は，ACM と IEEE が公開しているカリキュラムガイドラインに対して授業資料のマッピングを行う手法を提案している．カリキュラムガイドライン [13] とは，ACM と IEEE が公開している Computer Science(CS) 系のカリキュラムに必要な情報を体系的にまとめたガイドラインである．この研究では，そのカリキュラムガイドラインと論文の著者らが用意した資料の関連付けを行っている．カリキュラムガイドラインは領域ごとに分けられているため，この仕組みを用いることでそれぞれの領域で扱われている資料を整理することができる．本研究では学生が学習する科目について学習順序を重視して資料を分析し，学生に提示する．この研究のように学習順序が体系的に整理されたものと併用することで，学生に対してより有効なフィードバックが可能になると考えられる．

山本ら [2] は，シラバスとカリキュラムマップに基づいてこれらを結びつける手法を提案している．この研究では TF-IDF と Word2vec を用いて，シラバスの文章とカリキュラムの文章の関連性を発見し，それらを結びつけている．図 3 にこの研究の中で提案されているシステムの動作を示す．カリキュラム全体が表示され，任意の科目を選択することで図の右側のようにその科目の概要や関連する科目名やキーワードなどが示される．これによって学生は自らのカリキュラムを作成するときに，ある科目がどういった科目に発展す

[2] `https://www.skillscommons.org/`(参照 2022/01/04)
[3] `https://www.youtube.com/`(参照 2022/01/04)

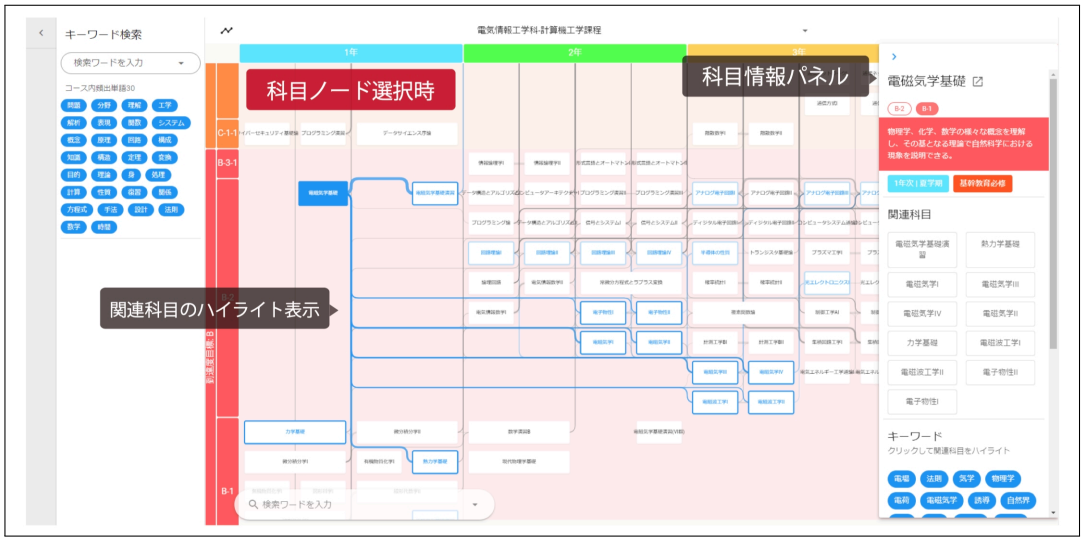


図 3: 科目を選択した時に提示される情報 [2]

るか発見できる．本研究とは，学生に発展する科目を示すという目的は類似するが，シラバスやカリキュラムマップを用いる点が異なる．特に本研究ではシラバスではなく授業資料を重視する．シラバスやカリキュラムマップはマクロな視点であり，授業資料はミクロな視点であるといえるため，より具体的な中身を学生に伝えることができる．また，学生が学習中にも異なる科目について知ることで，他の科目について意識した学習を行える．

Mima[14] は知識の構造化を目的として，MIMA(Mining Information for Management and Acquisition) Search システムを提案している．この研究では統計分析を用いて命名規則が明確ではない単語の統合とクラスタリングを行い，オントロジーを作成している．そのオントロジーを用いて文書同士の類似性を求め，異なる文書間の関係性を可視化するシステムを開発している．このシステムは東京大学のシラバス検索システムに応用されている[4]．こういった形で単語から他のコンテンツを抽出する試みは本研究と類似する．一方，オントロジーでは，授業資料の特徴は考慮できないと考える．本研究では，特に過去や未来といった順序関係を重視したいため，本研究ではこの手法は採用していない．

[4] https://catalog.he.u-tokyo.ac.jp/search/(参照 2022/01/13)

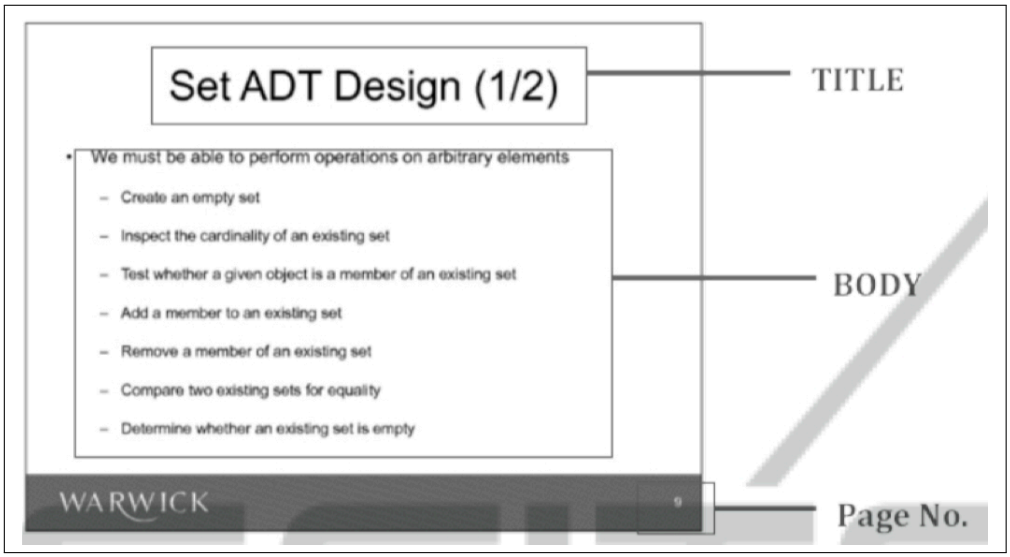


図 4: 講義スライドの構造 [15]

2.2 資料の提示

Sajjacholapunt ら [15] は，演習問題の文章とスライドのタイトル部分を重視する重み付けを行うことで，演習問題と関連するスライドを発見する手法を提案している．この研究では，スライドを図 4 に示すようなタイトルと説明部分とスライドのページ番号などの補足情報，演習問題の資料をコースの情報と日付と本文の情報でそれぞれ構成されていると仮定し，文章部分について TF-IDF の計算を行っている．加えて，タイトル部分の単語を重視するために，タイトル部分に対象となる単語が出現した時に，その出現頻度を重視する形で TF-IDF に重み付けしている．本研究で想定するスライドの構造は図 4 に類似する．また，本研究でも TF-IDF に対して複数の重み付けを行うが，その中でタイトル部分を重視する考えはこの研究と類似する．一方，本研究では TF-IDF に時系列の考えを加え、スライド内の文章は箇条書きによって整理されていると考え，インデントの高さを重視する．

椎野ら [16] は，演習問題と授業中の資料を関連付け学生に提示する研究を行っている．この研究では，まず，演習問題の問題文と授業中のスライド資料に対して TF-IDF を用いて重み付けを行う．それらを東北大学の乾・鈴木研究室が公開している日本語の Wikipedia のコーパスモデルと合わせて Word2vec モデルを用いて学習を行い，コサイン類似度を用いて類似ページの発見を行っている．図 5 に，フィードバック時の表示を示す．この図は，学生に対して実際にスライドを示すときである．学生にスライド本体と関連する授業用の

図 5: フィードバック時の例 [16]

Web ページ上のスライドを示すことで，学習を助けることができる．学生に対して授業資料を用いて学習を支援する点は本研究と類似するが，この研究では，演習問題の問題文に基づいてスライドを発見しようとしているため，本研究とは検索元の対象が異なる．坂本ら [3] は他大学の OCW から授業スライドを収集し，学生が自らの大学で学習中の科目と類似するスライドの対応づけを目的としたスライド結合手法を提案している．TF-IDF とコサイン類似度を用いて，OCW から収集したスライドの中で類似するページを一つのまとまりにし，別の資料同士に関連性を有するかの判定を行っている．関連する資料を学生に提示する点や同一の内容をまとめる試みは本研究と類似する．本研究では，関連するスライドを発見する上で，授業資料が持つ特徴を重視するため，この手法とは異なる結果が示される．

2.3 システムを用いた学習支援・教育支援

筆者は，卒業研究では授業中の教育支援を目的として，学生の注目箇所を教員にフィードバックする研究を行った [17]．この研究では教員と学生は PC やタブレットなどでスラ

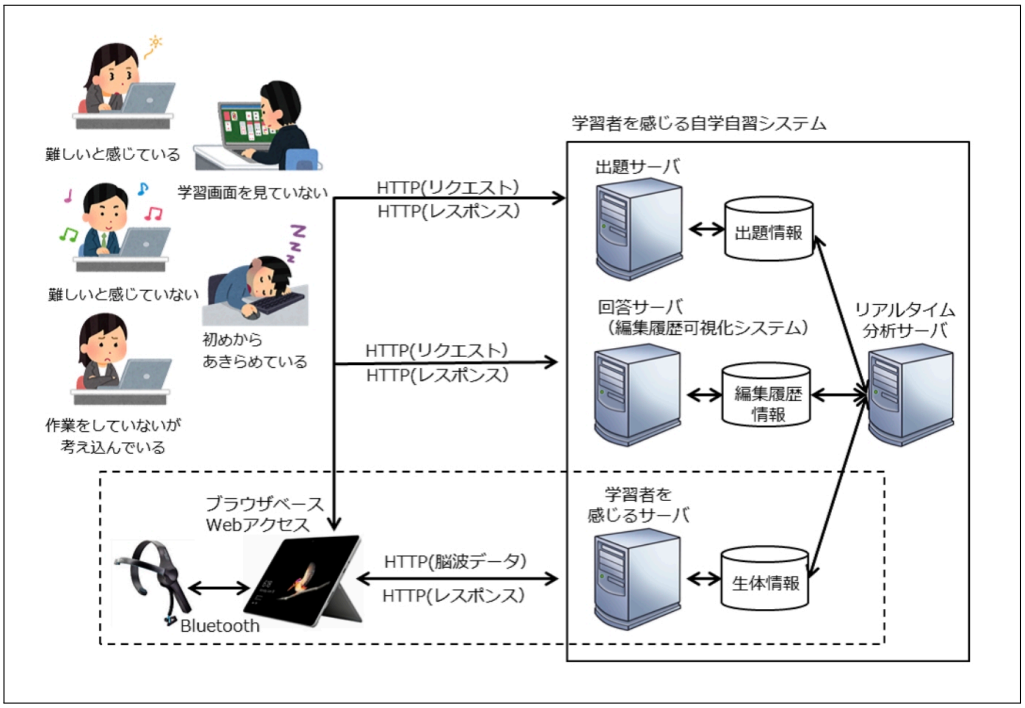


図 6: 自習システムの全体像 [18]

イドを閲覧しながら，授業を実施することを想定している．この研究では学生の注目箇所をスライドへの書き込みと仮定し，書き込んだ場所を教員にフィードバックしている．本研究とはスライドを対象にしている点は同様である．一方，この研究とは異なり自習中の支援を目的としている．またフィードバックの対象は教員ではなく自習中の学生である．

梅澤ら [18] は，自習支援を目的として，プログラミング教育と英語教育のためのケアレスミスフィードバックシステムを開発している．図 6 に，この研究で提案しているシステムの全体像を示す．この研究では，学生のプログラミングや英語学習中の編集履歴と脳波を取得している．編集履歴からそれぞれの学生のケアレスミスを発見し，リアルタイムにフィードバックとして問題を出題している．脳波から自習中の学生の思考状況の観察を行って，システムが示したフィードバックの問題や自習が難しいのか容易なのか判断し難易度の調整を行っている．本研究も自習中の学生を対象としている部分が類似する．この研究では，現在の学習の中でミスから発生するつまづきを解消するが，本研究では学生が学習の中で不明瞭になる将来性を示すことでモチベーションを維持する狙いがある．したがって本研究とは自習の中で解決する問題が異なる．

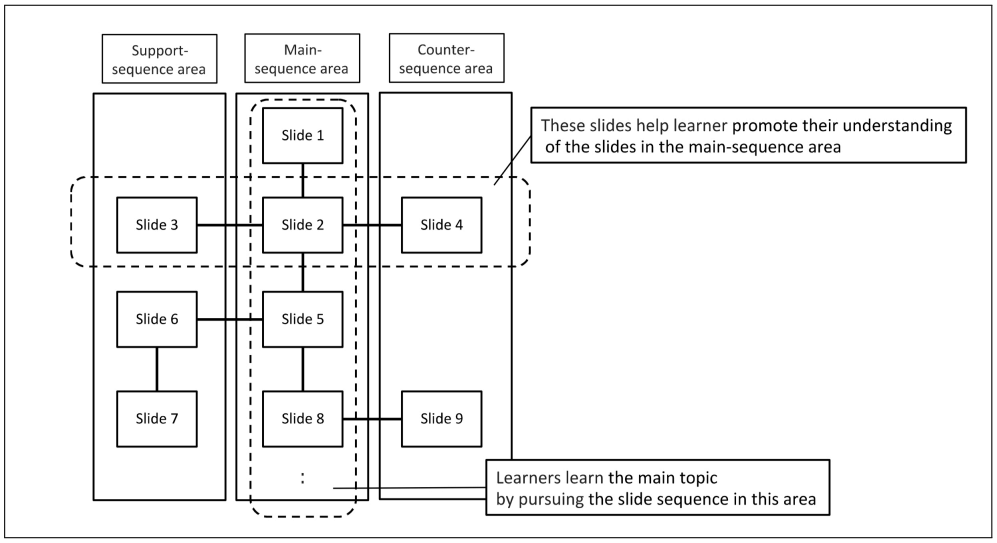


図 7: スライド同士の分類例 [19]

2.4 スライドの分析

Goto ら [19] は授業 1 回分のスライドを対象に，図 7 のように主軸のスライドとそのスライドを補足するスライドに分ける手法を提案している．この研究では接続詞に基づいてスライド同士の接続関係を考慮し，スライドの前後関係や関連性を整理する木構造の作成を行っている．この研究では石黒が記した「文章は接続詞で決まる」[5]で述べられた接続詞を考慮することで，キーワードが含まれないページについても関係性を発見している．生成した後は，提示されたユーザによる正誤判定に応じて適宜内容の修正している．スライド同士の関係性を発見する点は本研究と類似しているが，1 つのスライドではなくその学生が大学内で参照する全てのスライドを対象としている．本研究もスライドの出現順序や単語の出現時期を考慮するが，この研究で分類している 1 つのスライド内でのスライドの前後関係については考慮しない．これは本研究では 1 つのスライド内の関係よりも第 1 回から最終回まで資料同士の学習順序を重要としているためである．

羽山ら [20] は，スライドからの情報取得を容易にするためのインタフェースを提案している．この研究では，ユーザが検索時に指定した情報量に応じて，スライドのページタイトルのみや、ページと画像など，ユーザに示す情報の粒度を変更している．これにより，俯瞰的な情報をユーザは受け取ることができる．こういった情報を取得するために，スライドのページタイトルや本文，図や表などの情報をプレゼンテーションの xml から抽出し，

[5] https://www.kobunsha.com/shelf/book/isbn/9784334034733(参照 2022/01/10)

タグ付けして管理をしている．本研究もスライド内の構成やスライドタイトルなどの情報の一部を重視するが，図や表などの情報は重視する対象に含まない．一方，本研究では想定していないが，学生が実際に使用する場合の情報提示手法の一つとして参考になるものである．

桐原ら [21] はオンラインシステムで学習中の学生が注目している授業スライドの 1 スライドに対して関連する授業スライドの推薦を行う手法を検討している．この研究では，注目しているスライドに対して学生が質問を行うことで，その質問に基づいて関連する資料を示す．図 8 にこの研究における提示例を示す．左側は学生が学習中のスライドと質問，右側は質問に基づく候補スライドの一覧がランキング表示されている．右側は任意のスライドを学生が選ぶことで，拡大表示することができる．こういった提示を行うためにこの研究では，スライドが持つインデントの深さに応じて，そのスライド内の単語の重要度 (論文 [21] 中では寄与率と表現) を求めている．インデントの深さはスライドのページタイトルが 1.0 になるように重みを設定し，0.5，0.33，0.25 としている．そのため，ページタイトルに出現した単語は重要度が高くなりランキングの上位となる．本研究も学生の注目しているスライドを重視して関連するスライドの発見を行うが，その際にスライド内の単語を直接使用する．これによって，この研究とは異なり単語レベルでの関係性を示すことができる．また，本研究でも関連するスライドを発見するためにインデントを重視する．しかし，本研究ではページタイトルと本文中のインデントは別のものとして扱い，さらに文字に対する修飾も考慮する．

2.5 TF-IDF の改善

Sato ら [22] は TF-IDF に時系列の考えを導入し，災害発生時の Web ニュースで出現する単語の変化を分析している．この研究では，Chronological-incremental-TF-IDF(C-TF-IDF) と呼ばれる手法を用いて，一定期間ごとに IDF の値を更新しある単語が最初に出現した時期を最も重視する TF-IDF を提案している．Sato らはこの手法によって，災害発生時，最初期に出現し出現頻度が減少する単語，中期に出現する単語など時期ごとに特徴的な単語の発見している．本研究では，この手法を授業回とスライドごとの TF-IDF の値を計算するために採用する．本来は一定期間の資料をまとめて C-TF-IDF の計算を行っているが，本研究では 1 ページごとに計算するため C-TF-IDF による影響が強くなる．

Paik[23] は，対象となる資料内の単語数に基づいた TF-IDF の正規化と類似度計算手法の提案を行っている．単語数が多い場合は $BRITF$ という手法を，単語数が少ない場合は $BLRTF$ という手法を用いている．これらをバランスよく適用するために，類似度を計算したい資料同士の単語数に基づいて，重みを決めている．これによって，類似資料の計算

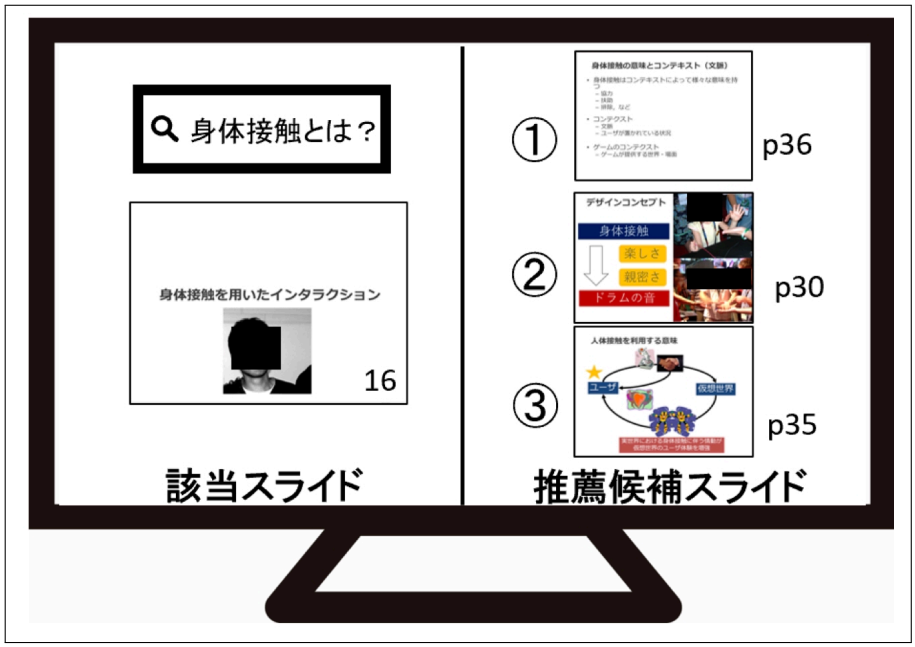


図 8: 質問に基づくスライドの提示例 [21]

タスクにおいて，従来の手法よりも有効な結果を得ている．本研究では，単語数の異なるスライドの TF-IDF のスコアを正規化する目的で本手法を導入する．本来は資料同士の計算を行うが，本研究ではこの研究の手法を用いて単語同士の計算を行う．

3　提案手法

本章では, スライドを発見するためのスコアリング手法について述べる．スライドはPDFにフォーマットされたものとし，1 ページあるいは数ページ程度のものを指す．まず，学生に関連する情報を示す際に有効な手法を整理し，他の手法との比較を行う．次に本研究で用いる TF-IDF と，他の考えられる手法との比較を行う．最後に学生に示すための最適なスライドを発見するために必要な，TF-IDF に対する重み付けの手法について述べる．

3.1　研究の全体像

本研究では，スライドを用いて自習中の学生に対し，科目を超えて過去や未来の関連するスライドを示すために，スコアリング手法の提案を行う．図 2 について，再度，以下に整理する．この中で，ステップ 2 が本研究の対象となる．

ステップ 1 自習中の学生が注目しているスライドに含まれる全ての単語 (名詞) を取得する．学生がそのスライドに辿り着いた時にスライド内の単語を取得することで達成可能である．

ステップ 2 全ての科目の中で，単語が含まれるスライドについてスコアリングを行う．

ステップ 3 学生が単語を選択した時, ステップ 2 のスコアに基づいて, 学生に対してスライドの提示を行う．ユーザインタフェースに該当する部分であるため，今後検討する．

学生に対し科目同士の関連付けを行う手法はいくつか考えられる．シラバスとカリキュラムマップから科目の関係性を整理することは可能である [2] が，具体的な授業中の内容を示すわけではないため，学生が内容をイメージしにくい．ほかにも，キーワードを用いる方法も考えられる．しかし，キーワードの設定へ負担やキーワードによって関連する科目の範囲が制限されると考えたため，採用を見送った．また，資料同士を結びつける場合，スライド単位ではなく，資料をそのまま結びつけることも考えられる．大抵の場合，1 回の授業の中で 2，3 のトピックを扱うため，その中から関連する内容を学生が発見する必要がある．スライド単位で提示を行うことで，一眼で関係性を俯瞰できると考えている．

本研究では，TF-IDF を用いてスコアリングを行う．この手法を用いることで，ある単語が特に重要視されているスライドを発見することができる．スコアリングやスライドの発見のために単語を用いる手法として，TF-IDF 以外に以下が考えられる．

BM25 BM25 は，検索システムで用いられている手法の一つである [24]．TF-IDF は単語数が多い場合を重視するが，BM25 は単語数が少ない場合に有効な手法の一つである．授業資料は単語数にばらつきが生じるため，どちらも有効に作用する．

表 1: 式 1 で用いる定義

t_i	$TF\text{-}IDF(t_i, d_j)$ で計算される単語.
d_j	t_i を含む特定の文書. 添字 j は文書数を N とし，$1 \leq j \leq N$ の範囲.
$\Sigma_{s \in d_j} f_{s,d_j}$	d_j に含まれる全ての単語 s の出現数.
N_j	文書 d_j が出現した時点での資料の総数.

オントロジー オントロジーは単語を知識として管理する手法である．オントロジーは既存の資料の中で単語同士のつながりを発見するために適しているが，新しい単語の更新にかかるコストが大きい．新しい単語が出現した場合のコストを考え，本研究では資料内の単語に基づいて計算が可能な TF-IDF を採用する．

Word2Vec Word2Vec では単語をベクトルとして扱う．この手法を用いることで，類似性が高いスライドを発見することができる．一方で，ある単語を別の意味で用いている場合に，そのスライドを発見することは難しい．例えば **heap** はアルゴリズムで用いるヒープソートと，コンピュータのメモリ内で用いるヒープ領域では，使い方が異なる．これらを同時に示すことで，学生が異なる視点を持つことにつながると考えている．シンプルに単語の重要度を測る TF-IDF を用いることで，そうしたスライドを発見可能であると考えている．

TF-IDF は式 (1) から式 (3) で定義される．

$$TF\text{-}IDF(t_i, d_j) = TF(t_i, d_j) \cdot IDF(t_i) \tag{1}$$

$$TF(t_i, d_j) = \frac{freq_{t_i d_j}}{\Sigma_{s \in d_j} freq_{s,d_j}} \tag{2}$$

$$IDF(t_i) = \log(\frac{N}{|d : d \in t_i|}) \tag{3}$$

ここで用いる定義を表 1 に示す．

式 (2) は，単語の出現頻度で，ある 1 つの単語 t_i と 1 つの文書 d を入力として受け取り，t が d 内で出現する頻度を求める．本研究では文書 d は 1 ページとする．式 (3) は，文書の逆頻度である単語 t_i が対象となる文書全体で，どの程度出現するか計算するものである．式 (2) と式 (3) の結果を掛け合わせることで式 (1) を求めている．TF-IDF は，多くの文書に出現する単語の値は小さくなり，特定の文書のみに出現する単語を重視することができる．

本研究では，TF-IDF の計算を行う段階で，分析のためにタイトルのみのスライドは除去する．タイトルのみのスライドはサムネイルのように，その後に続く内容を知る上で有

効な提示となる．しかし，本研究のようにスライドを示そうとした場合，タイトルのみでは学生に提示する情報としては不足していると考える．

一方，授業資料やスライドには TF-IDF では検討しきれない，以下のような特徴があると考える．

- 第 1 回から最終回まで連続になっていて，過去の内容に基づいて説明が展開されるなど，時系列が存在する．
- スライド自体の特徴としてページタイトルや箇条書き，文字の装飾があり，スライドの作成者が意図して強調することができる．
- 図形が主体のスライドと文字が主体のスライドがあり，スライドの単語数が異なる．

これらの特徴を重視するために，以降，第 3.2 節で時系列の導入について，第 3.3 節で単語数を考慮した TF-IDF のスコアを正規化について，第 3.4 節でスライドの構造に注目した単語レベルの重み付けについて述べる．

3.2 Chronological incremental TF-IDF

本研究では，TF-IDF に対して時系列を考慮した重み付けを行う．大学の授業は 1 学年から 4 学年の間で時間的に連続し，以前に扱った内容を前提に説明が行われる．また，1 つの授業も第 1 回から最終回までの間も同様に連続したものである．そのため，TF-IDF をそのまま計算するよりも，時系列を取り入れたほうが適切であると考えた．

TF-IDF に時系列の考え方を導入した研究として，Sato ら [22] が *Chronological incremental TF-IDF*(C-TF-IDF) を提案している．この研究は，地震発生から時間経過によって変化する Web ニュースの単語を分析するために用いられている．ある期間から特定の時期までの一定区間内の単語の変化を見ることに適しているため，本研究でも有効であると考えた．C-TF-IDF は式 (4) で定義される．

$$
\begin{aligned}
&Chronological\ Incremental\ TF\text{-}IDF(t_i, d_j) \\
&= TF(t_i, d_j) \cdot IDF(t_i, d_j) \qquad (4) \\
&IDF(t_i, d_j) = log(\frac{N_j}{DF(t_i, d_j)}) \qquad (5)
\end{aligned}
$$

C-TF-IDF の式中に用いる定義を表 2 である．文書 d_j は，事前実験では 1 回分の資料を 1 単位，提案手法ではスライド内の 1 ページを 1 単位として用いる．式 (1) との違いは，IDF の計算方法である．式 (3) では，単語 t_i のみを引数に受け取り，すべての文書を対象にして

表 2: 式 3.2 で用いる定義

t_i	$TF\text{-}IDF(t_i, d_j)$ で計算される単語. 添字 i は文書 d_j 内の単語数を n とし, $1 \leq i \leq n$ の範囲.
d_j	t_i を含む特定の文書. 添字 j は文書数を N とし, $1 \leq j \leq N$ の範囲.
$TF(t_i, d_j)$	文書 d_j 内の単語 t_i の出現回数. TF-IDF と同様.
$DF(t_i, d_j)$	文書 d_j 出現時の単語 t_i の出現文書数.

いた. 式 (5) は, 単語 t_i のほかに文書 d_j を受け取り, 文書 d_j までの総数 N_j の中で出現した文書の数を計算する. 文書数が増え, 単語 t_i が出現する文書が増えると IDF の値は次第に減少する. そのため, 最初に出現した文書を重視することができる. これにより, 授業内である単語 t_i を説明しているスライドを発見できると考えている. また, C-TF-IDF は単語が期間を空けて出現した場合にも, その文書を高く評価することが筆者の実験からわかっている. 図 9 は C-TF-IDF の評価のために行った事前実験の結果から得た C-TF-IDF の値の変化を示したものである. 横軸は, 単語「初期化」が出現したときの授業回とページ番号を示している. 縦軸は C-TF-IDF の値である. 定期的に値が上昇することが示されている. これは C-TF-IDF の特性として, 連続して単語が出現し続け場合には値を減少させ, 出現しない期間があると値を上昇させる. これにより, 単語が再び出現したスライドも C-TF-IDF を用いて正しく評価することができる.

事前実験として, 1 スライドを d_j の 1 単位として C-TF-IDF を用いて, 有効な結果が得られるか検証を行った [25]. この事前実験では, 明星大学情報学部のカリキュラム内で, 2017 年度に開講された「プログラミングⅡ」と 2018 年度に開講された「確率過程」を対象に単語検索を行った場合の, TF-IDF(表内では N-TF-IDF) と C-TF-IDF とで生成されるランキングの違いを評価した. どちらの手法についてもそれぞれの回の資料中のすべての単語を計算に用いている. C-TF-IDF については, 次の回に進むタイミングで IDF の更新を行っている. ここから, 表 3, 表 4 のような結果が得られた. 表 3 は, プログラミングⅡ内で「配列」という単語を対象にした場合である.「配列」という単語は第 2 回で初めて出現し, その後ほとんどの資料で出現した. この結果では C-TF-IDF では授業が進むごとに値が減少し, 初めて出現した第 2 回を重視することができている. このうち, 第 4 回は授業内でテストを行った回で他の資料よりもスライドのページ数が少ない. 対象としている単語も前回授業で課されていたレポートの解説でのみ出現していた. このようにページ数が少ない場合, 全体で出現する単語数が減少し, 単語数が多い他の資料と比較して一つ一つの単語を大きく評価している可能性が考えられる. これについては第 3.3 節の正規化を用いることで解決を図る. 表 4 では, 確率過程の中で,「分散」について調べた場合で

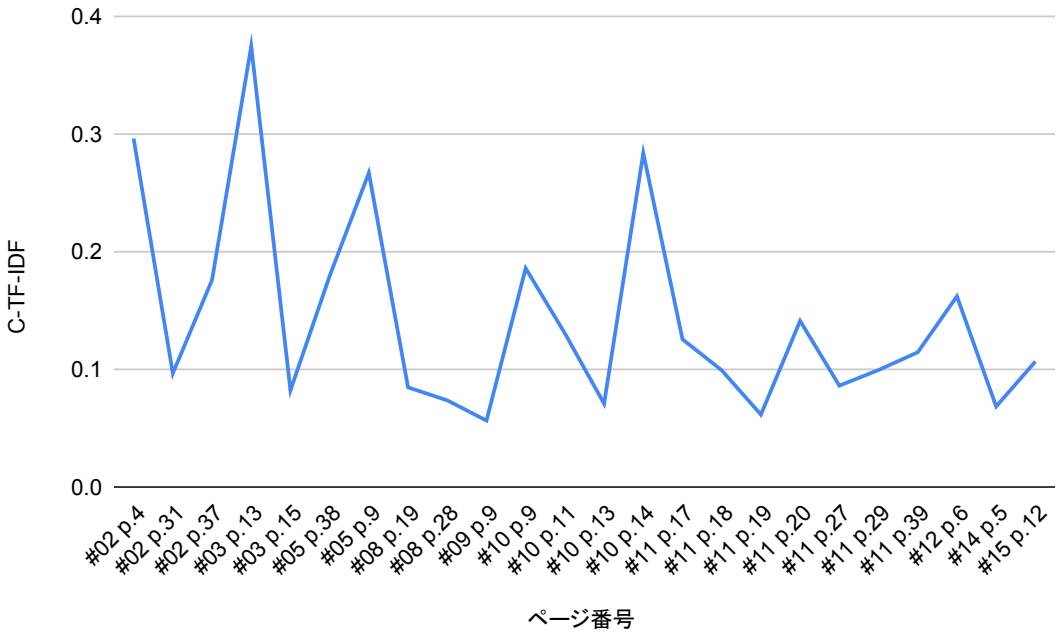


図 9: プログラミングⅡ内の単語「初期化」における C-TF-IDF の値の変化

表 3: プログラミングⅡを対象とした C-TF-IDF と TF-IDF(N-TF-IDF) の結果

word	C-TF-IDF		N-TF-IDF	
	Doc	Score	Doc	Score
配列	#02	0.0281	#05	0.0292
	#04	0.0279	#04	0.0269
	#05	0.0256	#02	0.0255

表 4: 確率過程を対象とした C-TF-IDF と TF-IDF(N-TF-IDF) の結果

word	C-TF-IDF		N-TF-IDF	
	Doc	Score	Doc	Score
分散	#06	0.0398	#06	0.0276
	#07	0.0198	#09	0.0179
	#09	0.0167	#07	0.0160

ある．「分散」は第 6 回で初めて出現し，その後第 7 回，第 8 回，第 9 回，第 11 回と続く．スコアを見ると第 6 回をより重視しているのは C-TF-IDF のほうである．そのため有効性はあるといえるが，ランキングの順序にはほとんど違いが見られなかった．

スライド単位での結果を踏まえ，スライド内の 1 ページ単位での結果についても事前実験で検証を行った．ページ単位で分析を考えた時，C-TF-IDF の計算方法は 2 通り考えられる．(1) スライド単位で IDF を更新する．(2) ページ単位で IDF を更新する．(2) については，スライドごとにこれまでの結果をリセットする方法も考えられるが，本研究では第一回の第一ページから最終回の最終ページまで連続して IDF を更新する．この理由は，授業は毎回蓄積されるものであるため，そこまでに使用された単語の出現状況について考慮するべきと考えたためである．表 5 と表 6 に検証の結果を示す．表 3 や表 4 と検索語は同一である．具体的な結果として，表 5 の結果からプログラミングⅡの#2 の p.26(図 10) と#5 の p.7(図 12) を，表 6 の結果から確率過程の#6 の p.2(図 13) と#9 の p.7(図 14) を示す．プログラミングⅡの結果である図 10 は配列の利点について説明をしているが，内容の振り返りや発展などで示すことを考えると，図 12 のような例の方が望ましい．図 10 の次のページである p.27 では，図 11 に示す実装時のルールに触れているため，p.26 よりも p.27 のほうが適切である．確率過程の結果である図 13 は，初めて分散という言葉が出てきた回であったため，振り返りの観点で有効である．図 14 については，定理の説明の中で分散の単語を扱っている．こうした形で数式的な関連性を示すことは発展を示す意味で有効であると考えられる．

表 5: プログラミングⅡを対象としたページ単位の C-TF-IDF と TF-IDF(N-TF-IDF) の結果．上位 3 つについて示す．

word	C-TF-IDF(1)		C-TF-IDF(2)		N-TF-IDF	
	Doc	Score	Doc	Score	Doc	Score
配列	#05 p.7	0.6422	#02 p.26	0.3804	#05 p.7	0.6566
	#02 p.26	0.5797	#05 p.7	0.2176	#02 p.26	0.3896
	#02 p.27	0.3027	#08 p.7	0.1978	#05 p.9	0.2814

表 6: 確率過程を対象としたページ単位の C-TF-IDF と TF-IDF(N-TF-IDF) の結果．上位 3 つについて示す．

word	C-TF-IDF(1)		C-TF-IDF(2)		N-TF-IDF	
	Doc	Score	Doc	Score	Doc	Score
分散	#06 p.2	0.1419	#06 p.2	0.0292	#06 p.2	0.0684
	#06 p.3	0.0670	#09 p.7	0.0226	#09 p.7	0.0658
	#09 p.7	0.0613	#11 p.3	0.0225	#09 p.4	0.0480

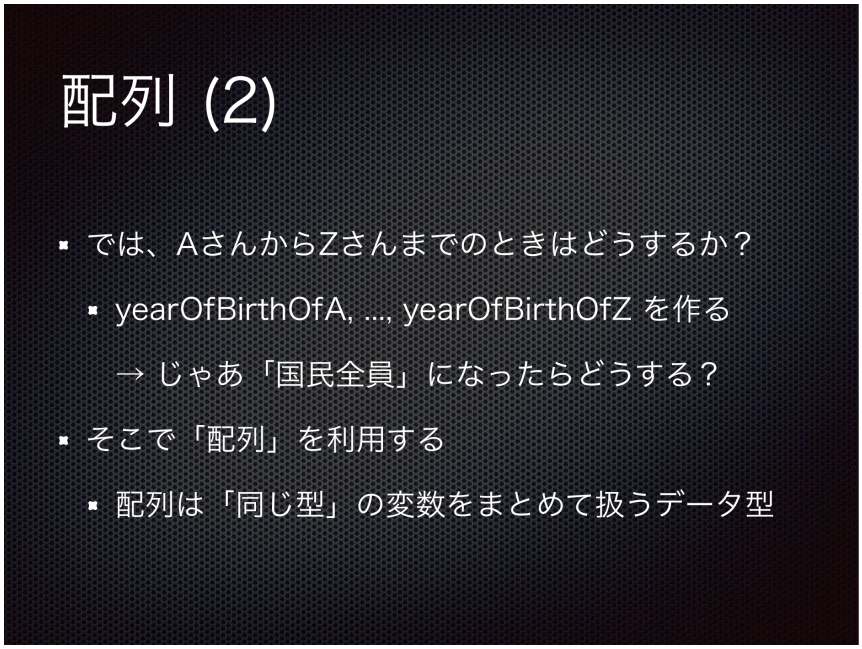


図 10: プログラミングⅡの#2 の p.26 の内容

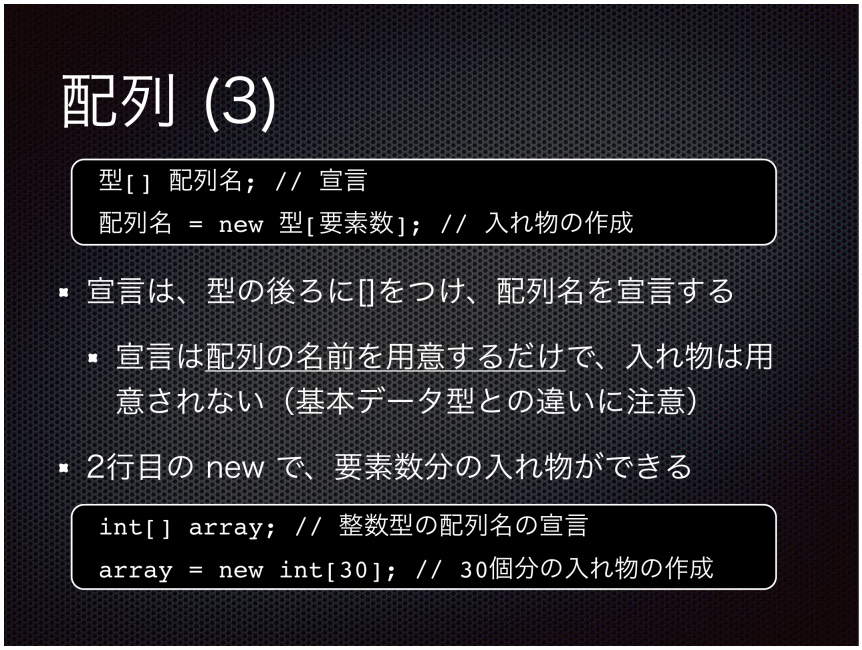


図 11: プログラミングⅡの#2 の p.27 の内容

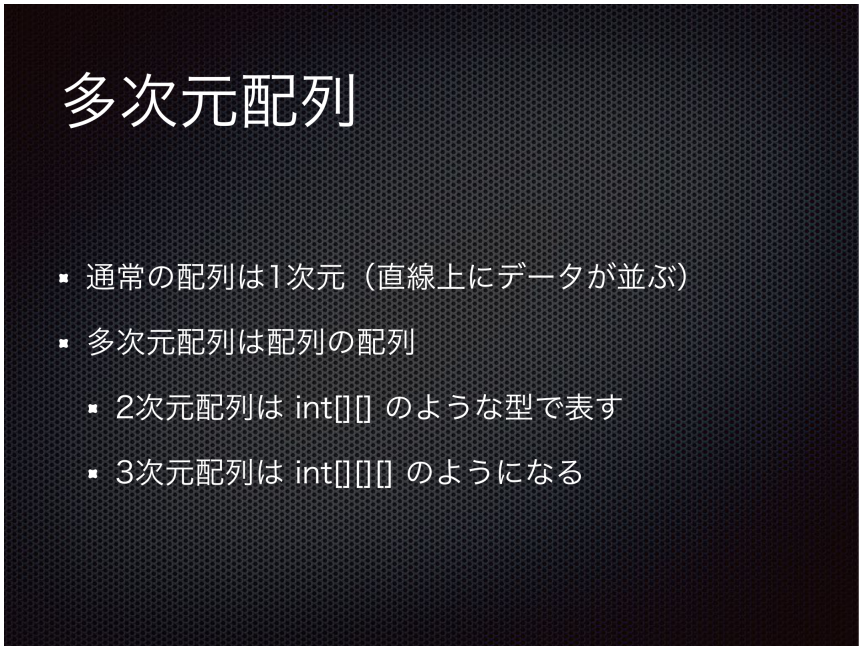


図 12: プログラミングⅡの#5 の p.7 の内容

期待値の性質（4）

期待値のかけ算の $E[XY] = E[X]E[Y]$ は成り立つか？

確率のかけ算$P(A \cap B) = P(A)P(B)$が成り立つとき、事象AとBは独立である。

「確率変数の独立」についても次のように定義できる。

確率変数X, Yが独立とは、$X = x,\ Y = y$ という2つの事象がすべてのx, yにつき独立であること、つまり、X, Yのとりうるすべての変数の値に対して

$$P(X = x, Y = y) = P(X = x)P(Y = y)$$

が成り立つことをいう。

期待値の性質（5）

前述の2つのサイコロの例なら、A,Bの出る目はそれぞれ独立と考えられるので、

$$P(X = 10, Y = 5) = P(X = 10)P(Y = 5)$$

などが成り立っている。したがって、

$$\begin{aligned} &E[XY] \\ &= (10 \cdot 10)P(X = 10, Y = 10) + (10 \cdot 5)P(X = 10, Y = 5) \\ &+ (5 \cdot 10)P(X = 5, Y = 10) + (5 \cdot 5)P(X = 5, Y = 5) \\ &(10 \cdot 10)P(X = 10)P(Y = 10) + (10 \cdot 5)P(X = 10)P(Y = 5) \\ &+ (5 \cdot 10)P(X = 5)P(Y = 10) + (5 \cdot 5)P(X = 5)P(Y = 5) \\ &= \{10P(X = 10) + 5P(X = 5)\} \times \{10P(Y = 10) + 5P(Y = 5)\} \\ &= E[X]E[Y] \end{aligned}$$

分散

- 分散（variance)とは確率分布の散らばりの指標である。
- 分散の正の平方根を**標準偏差**という。
- 変数Xの分散は母平均μからの偏差の2乗の期待値として得られる。

離散型の場合は、

$$V[X] \equiv E[(X - \mu)^2] = \sum_{i=1} (x_i - \mu)^2 f(x_i) = \sigma^2$$

連続型の場合は、

$$V[X] \equiv E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x_i - \mu)^2 f(x) dx = \sigma^2$$

また、$V[X] \equiv E[(X - \mu)^2] = E[X^2] - \mu^2$ と変形でき、X^2の期待値から母平均の2乗を引いて求めることができる。

分散の性質（1）

分散の定義

$$V[X] = E[(X - \mu)^2]$$
$$\mu = E[X]$$

$$\begin{aligned} V[X] &= E[X^2] - 2E[X]E[X] + E[X]^2 \\ &= E[X^2] - (E[X])^2 \end{aligned}$$

分散は「偏差の2乗の期待値」であるのでマイナスになることはないので、

$$E[X^2] \geq (E[X])^2$$

は常に成り立つ。

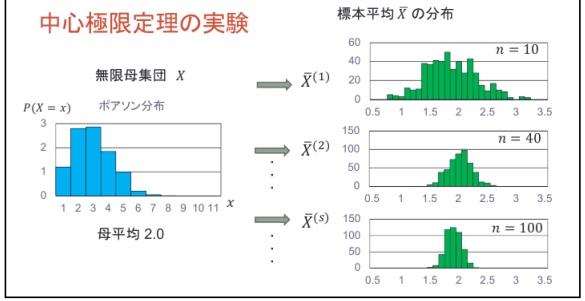
図 13: 確率過程の#6 の p.2 の内容

中心極限定理

- 同じような確率変数を加えると、その和の分布は正規分布に従ってくる。nが大きいときは
  - 期待値: $n \times \mu$(1個の期待値)
  - 分散: $n \times \sigma^2$(1個の分散)

  の正規分布に従う。
- 母集団がどのような分布であっても中心極限定理は成り立つ。
- **中心極限定理**は、nが大きいとき標本平均$\bar{X}$と母平均μの差$\bar{X} - \mu$が従う分布は平均0, 分散σ^2/nの正規分布に近づく。
- $X_1, X_2, X_3, \cdots, X_n$が互いに独立で平均$\mu$、分散$\sigma^2$の同一確率分布に従うとき、$Z = \frac{\bar{X}-\mu}{\sigma/\sqrt{n}}$の分布は、$n \to \infty$とすると、標準正規分布$N(0,1)$に近づく。

中心極限定理の実験



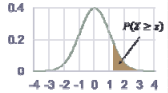
中心極限定理(例題)

ある企業における男性社員の体重の分布は平均が$\mu = 70$ kg 、分散が$\sigma^2 = 64$ kg^{2}であるとき、男性社員の中から無作為に50人選び体重を測定したとき、50人の標本平均$\bar{X}$が68 kg以下である確率は求めよ。

標本平均の正確な値は分からないが、無作為抽出であることから中心極限定理により、標本平均$\bar{X}$は平均70 kg 、分散が$\sigma^2 = 64/50$ kg^{2}の正規分布に従う。

$$P(\bar{X} \le 68) \cong P\left(\frac{\bar{X}-70}{8/\sqrt{50}} \le \frac{68-70}{8/\sqrt{50}}\right) = P(Z \le -1.77) = 0.0384$$

標準正規分布の上側確率



Z	.00	.01	.02	.03	.04	.05	.06	.07	.08	.09
0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
0.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
1	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
2	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
2.6	0.0047	0.0045	0.0044	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0019
2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
3	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010

図 14: 確率過程の#9 の p.7 の内容

実験の結果から，出現したタイミングを考慮する場合，(2) の手法が適切であると考え採用する．一方，C-TF-IDF だけ単体では，図 10 のような例やページに含まれる文字数によって，意図しない結果になる場合もある．それらを抑制および望ましい結果を得るために，(2) の C-TF-IDF のスコアに対して正規化とスライドの構造を加味した重み付けを行う．

3.3 TF-IDF の正規化

授業用のスライドには，説明のために，図を豊富に用いるページや文章を豊富に用いるページがある．ここで，スライドのいくつかの例を示す．図が説明の中心で文字数が少ない場合，図 15 のようなものがある．ソースコードがある場合，図 16 のようなものがある．図 15 と図 16 はどちらも同じ科目のスライドだが，図 15 は図が説明の中心で文字数が少なく，図 16 はソースコードを含んでおり，スライド内の文字数が多い．これらに TF-IDF を適用した場合，前者は 1 つ 1 つの単語を重視する．後者は出現数が少ない単語を重視する．後者の場合，変数名として用いている **stack** は，スライド内の出現回数が多く，そのスライドの **stack** における TF-IDF のスコアが下がってしまう．ソースコードは図として扱い，TF-IDF の計算を行わないという手法も考えられるが，ソースコード内で用いている単語を学生に提示することで，抽象的な説明があるスライドに対して具体的な提示が可能になると考えるため，本研究ではソースコードも単語として扱う．しかし，ソースコードを文字として TF-IDF の対象とする場合，単語数が多いために TF-IDF では単語数が少ない他のページを重視してしまう可能性がある．これでは，本来重視したい単語を重視しきれないと考えたため，C-TF-IDF で求めたスコアに対して，スライド内の単語数に基づく正規化を行う．

正規化には Paik[23] の手法を用いる．この手法は TF-IDF をベースとし，対象となる資料の単語数に応じて正規化を行う．正規化に用いる計算方法は式 (6) から式 (15) である．式 (6) から式 (11) までは，TF に対する正規化である．式 (13) と式 (14) は IDF に対する

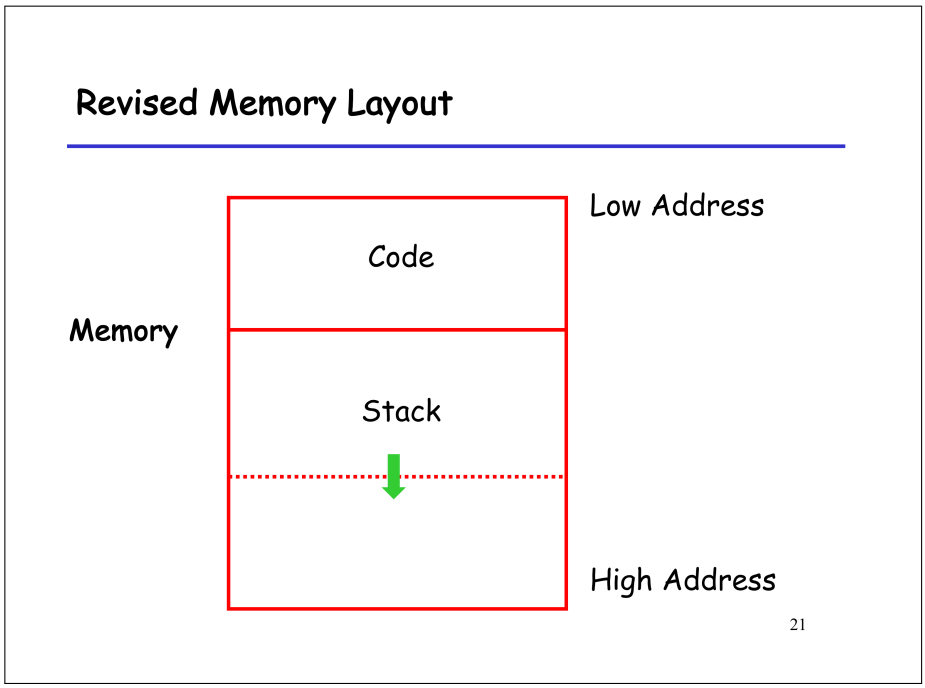


図 15: 図が中心のスライド (CS143[8] より引用)

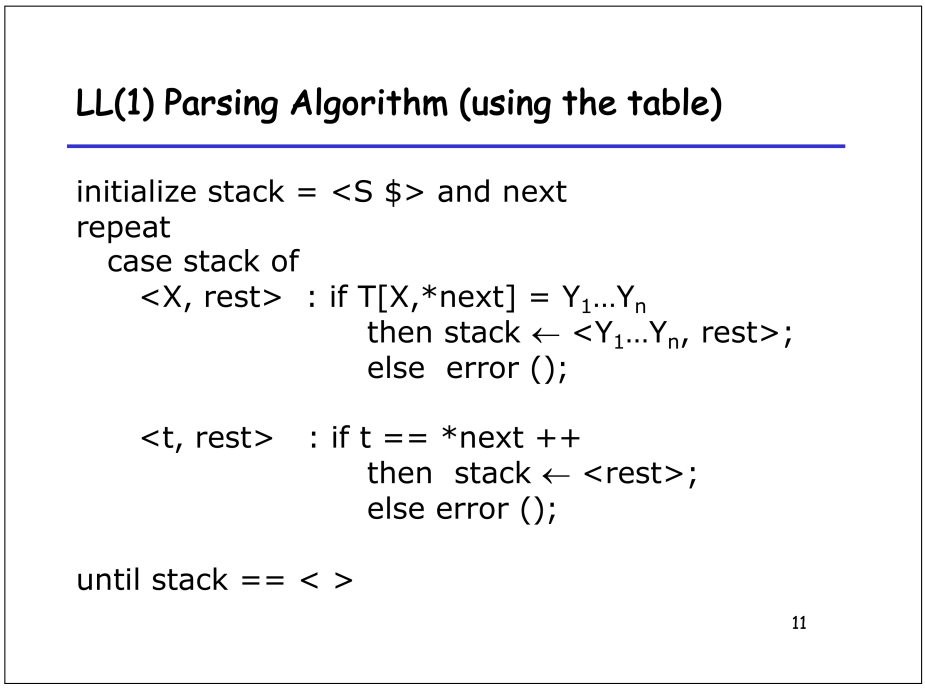


図 16: ソースコードが中心のスライド (CS143[8] より引用)

正規化である．最終的なスコアは式 (15) によって算出される．

$$RITF(t,D) = \frac{\log_2(1+TF(t,D))}{\log_2(1+Avg.TF(D))} \tag{6}$$

$$LRTF(t,D) = TF(t,D) \times \log_2(1+\frac{ADL(C)}{len(D)}) \tag{7}$$

$$BRITF(t,D) = \frac{RITF(t,D)}{1+RITF(t,D)} \tag{8}$$

$$BLRTF(t,D) = \frac{LRTF(t,D)}{1+LRTF(t,D)} \tag{9}$$

$$TFF(t,D) = w \times BRITF(t,D) + (1-w) \times BLRTF(t,D) \tag{10}$$

$$w = QLF(Q) = \frac{2}{1+log_2(1+|Q|)} \tag{11}$$

$$\tag{12}$$

式 (6) は，特に単語数が多い場合に有効な正規化である．一方，式 (7) は，単語数があまり多くない場合に有効である．ここで登場する $ADL(C)$ は全ての文書の単語数の平均である．式 (8) と式 (9) は，式 (6) と式 (7) を用いて，以下の 3 条件を満たすように調整したものである [23].

1. 0 のときに消失する．
2. $f'(x) > 0 \bigwedge f''(x) < 0$ となる．
3. 1 に上界する．

$$TDF(t,C) = IDF(t,C) \times \frac{AEF(t,C)}{1+AEF(t,C)} \tag{13}$$

$$AEF(t,C) = \frac{CTF(t,C)}{DF(t,C)} \tag{14}$$

$$Sim_{norm}(Q,D) = \frac{\sum_{i=1}^{|Q|} TFF(q_i,D) \times TDF(q_i,C)}{\sum_{i=1}^{|Q|} TDF(q_i,C)} \tag{15}$$

式 (10) は，重みによって式 (8) と式 (9) のバランスを取る．重みは式 (11) によって決定し，この式 (11) は検索時に使用する単語の集合や類似度を比較する際の単語の集合などの $|Q|$ によって決まる．本研究では，Paik[23] が比較した中でもっとも有効とされている手法を採用し，$|Q|$ のサイズについては 1 つの単語での計算を想定するため 1 とする．式 (13) は

平均出現頻度である式 (14) を用いて IDF の値を正規化する．式 (14) で登場する $CTF(t, C)$ は，すべての資料の中での単語 t の総出現数である．本研究では C-TF-IDF を採用するため，あるページまでに出現した単語 t の総出現数となる．式 (15) では，入力された単語の集合 $|Q|$ との類似度の計算を行うことで，正規化した値を算出する．そのため，$|Q|$ は複数の単語にも対応できる．しかし，本研究では $|Q|$ を 1 単語との類似度を求めるために扱う．今後，複数の単語の集合についても扱っていく必要がある．

3.4 スライドの構造を重視した重み付け

授業用のスライドの 1 ページには，いくつかの要素がある．図 17 にページの要素がまとめられた例を示す．図 17 から得られる要素は以下である．

- スライドの上部にあるページタイトル部分.
- 箇条書き.
- 太字や色付きなど文字の強調.
- ページ内の図や表.

本研究では，このうち，ページタイトル，箇条書き，文字の強調について重視する．図のうち，フォントが取れる情報は本研究の対象にする．図に代替テキストが含まれている場合，そのテキストを対象とする．表はその中の単語は対象とするが，表内の配置は対象としない．一方，図 16 は，図として線を読み取るとメモリの構造であることが明白である．しかし，本研究ではこうした線の部分は対象にしない．画像認識を用いたり，スライド作成ツールや PDF のフォーマットの情報からある程度図の情報取得は行えると考えている．一方，得られた画像 1 つ 1 つの認識結果の判断を行うことになるため，本研究ではその分野は対象としない．そのため，本研究ではテキスト形式の情報を基にした重み付けを行う．

特に本研究ではページタイトル，太字，箇条書きに注目し，重み付けを行う．文字の装飾に関しては色をつける場合もあるが，太字にするのか色をつけるのかは資料作成者によって異なる．そこで，本研究では太字にのみ注目し重み付けを行う．太字への重み付けで有効性が確認できた場合，色やイタリック体など他の文字への装飾に対して同一の方法で重み付け可能であると考えている．重み付けで用いるいくつかの定義を表 7 に示す．表 1 や表 2 と同様の定義は示していない．

ページタイトルと太字に関する重み付けはそれぞれ式 (16) と式 (17) である．本研究では，タイトルや太字について，説明する内容が変化したときに重要視したい．そのため，これらの重みは C-TF-IDF のアイディアを参考としている．C-TF-IDF は途中で単語が出現

Implementing search to emulate find

- The following **`main`** relies on **`listMatches`**, which we'll implement in a second. The full program, complete with error checks we don't present below, is right here.

```
1  int main(int argc, char *argv[]) {
2    const char *directory = argv[1];
3    struct stat st;
4    stat(directory, &st);
5    if (!S_ISDIR(st.st_mode)) return 0;
6    size_t length = strlen(directory);
7    const char *pattern = argv[2];
8    char path[kMaxPath + 1];
9    strcpy(path, directory);
10   // buffer overflow impossible, directory length <= kMaxPath else stat fails
11   listMatches(path, length, pattern);
12   return 0;
13 }
```

- This is our first example that calls **`stat`** and **`lstat`**, each of which extracts information about the named file and populates the **`struct stat`** supplied by address.
- You'll also note the use of the **`S_ISDIR`** macro, which examines the upper four bits of the **`st_mode`** field to determine whether the named file is a directory.
- **`S_ISDIR`** has a few cousins: **`S_ISREG`** decides whether a file is a regular file, and **`S_ISLNK`** decided whether the file is a link.
- Most of what's algorithmically interesting falls under the jurisdiction of this **`listMatches`** function, which performs a depth-first tree traversal of the filesystem to determine what filenames just happen to match the name of interest.

図 17: 単語の特徴について示されているスライド (CS110[6] より引用)

表 7: 式 3.2 で用いる定義

$T(d_j)$	ページ d_j でタイトルに含まれる単語の集合を求める．
$B(d_j)$	ページ d_j で太字になっている単語の集合を求める．
$indent$	箇条書きのインデントを指し，インデントは最大で 4 までの深さになる．

しない期間があり，その後に出現した場所を重要視する傾向にある．$Title\text{-}DF(t_i, d_j)$ は，単語 t_i が d_j までにタイトルとして出現した頻度を求める．$Bold\text{-}DF(t_i, d_j)$ は，単語 t_i が d_j までに太字として出現した頻度を求める．異なる内容を連続して扱った場合には対応できないが，回を跨いで出てきた単語や別の説明の中で関連した単語として扱った場合には有効であると考えられる．

$$T_w(t_i, d_j) = \begin{cases} \frac{N_{d_j}}{Title\text{-}DF(t_i, d_j)} & t_i \in T(d_j) \\ 1.0 & t_i \notin d_j \end{cases} \tag{16}$$

$$B_w(t_i, d_j) = \begin{cases} \frac{N_{d_j}}{Bold\text{-}DF(t_i, d_j)} & t_i \in B(d_j) \\ 1.0 & t_i \notin d_j \end{cases} \tag{17}$$

箇条書きは，文章を簡潔に整理するために使われている．本研究では，箇条書きの構造は，図 18 のように整理できると考える．これは，スライドの箇条書きを仮定したものである．そこで，スライド中で出てくる箇条書きはインデントが浅いものを重視する．箇条書きの重みは式 (18) で決定する．

$$I_w(t_i, d_j) = log(\Sigma_{indent=0}^{4} freq(indent, t_i, d_j) * w_{indent}) \tag{18}$$

ここで，w_{indent} は $indent$ が 0 から順に図 19 の値を割り当てる．$freq(indent, t_i, d_j)$ は，インデント $indent$ でページ d_j までの単語 t_i の出現回数を求める．このとき，箇条書きは，箇条書きを表す記号の有無に関わらず，最も左側を 3.5 とする．そこに，そのインデントでのその単語の出現回数を反映する．これにより，ページ d_j まででどの位置のインデントでその単語が出現したのかを反映することができる．単語が，多くのページでインデントの浅いに多く出現した場合の重みは大きくなる．1 つのページの中で単語が複数のインデントに現れる場合は，最もインデントが浅いものだけを重視する．箇条書きの値は，当初図 20 に示す桐原ら [21] のものを参考にしていた．このとき，単語の出現関数は重視していなかった．この重みでは，タイトルや太字に対する重みが大きく箇条書きを重要視することができなかった．次に，インデントの 1 段目の 1.5 とし，2 段目を 1.0，3 段目を 0.5，4 段目を 0.33 とした値も試したが，こちらも当初のものと同様，結果に変化がなかった．そして，インデントごとの出現回数を重視し，タイトルや太字に対する重みと同等に箇条書きを重視するために現在の値となった．

式 16 から式 18 を用いて，C-TF-IDF にさらに重み付けする．式 (19) はそれぞれの重みを足し合わせ，正規化した C-TF-IDF と掛け合わせている．重み同士を積算する方法も考

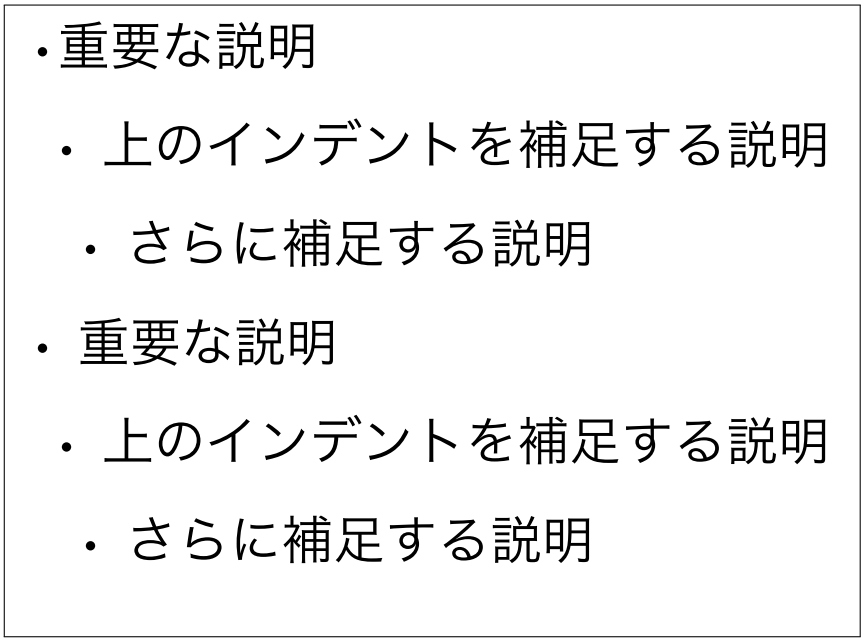


図 18: 本研究で用いる箇条書きの配置

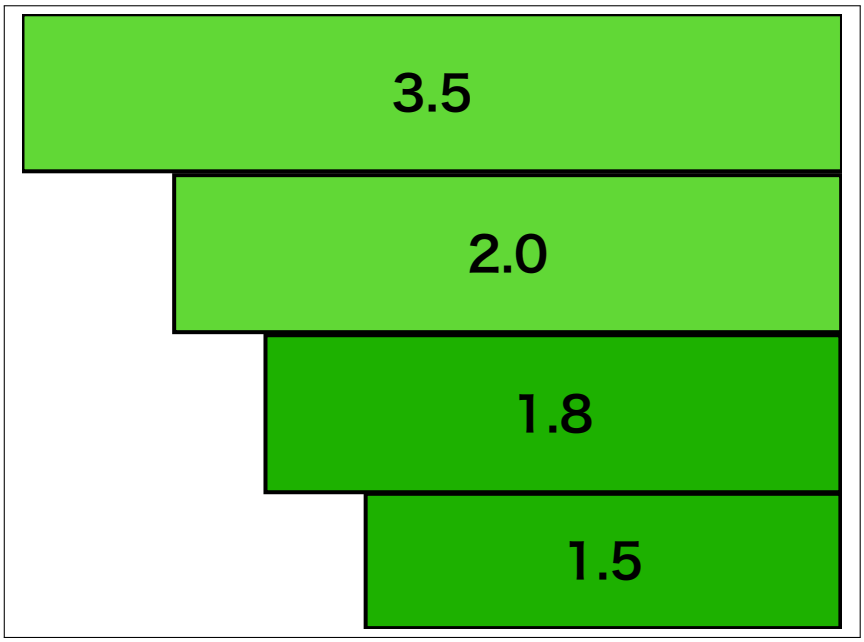


図 19: 本研究で用いる箇条書きの重み

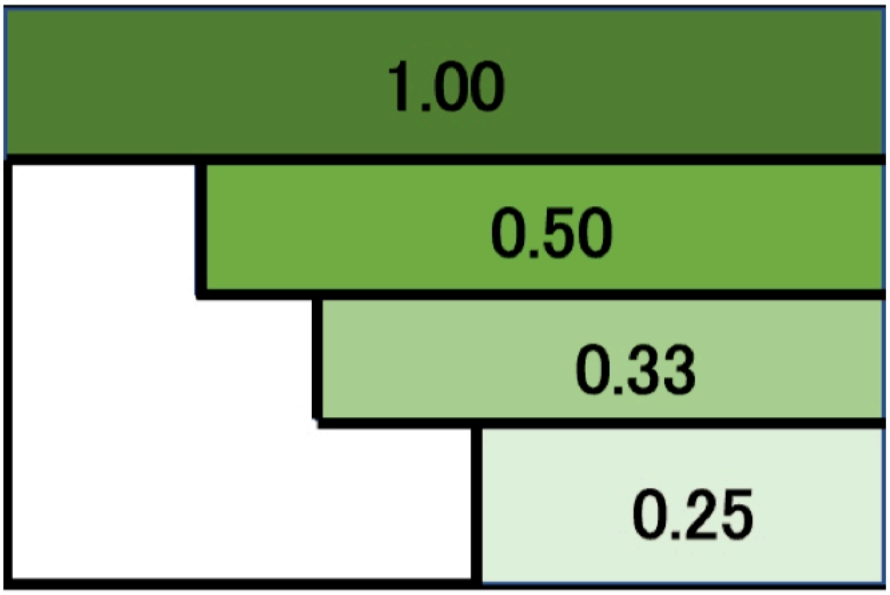


図 20: インデントによる重みの値 [21]

えられるが，重みの影響が大きくなりすぎると考え，本研究では足し算とした．

$$Weighted\text{-}C\text{-}TF\text{-}IDF = Sim_{norm}(Q, D) * (T_w + B_w + I_w) \tag{19}$$

4　実装

スコアリングを行うシステムの作成を行った．本研究では，フォントの情報が取得可能でPDF にフォーマットされたスライドに対象を限定する．これは，文字情報を画像情報からではなく PDF が持つ情報から取得しているためである．PDF のバージョンは 1.0.7 以上とし，構造の詳細は Adobe が提供する PDF format[26](PDF フォーマット) に従う．テキスト情報の抽出および TF-IDF の計算のプログラムについては Python3.9.2 をベースとする．PDF の解析のために pdfminer.six(version 20200517)[6](pdfminer) を使用した．pdfminer は，PDF フォーマットに基づいて実装された Python 用ライブラリの 1 つである．この API は，フォントの情報を用意することで複数の自然言語に対応可能であるが，今回実装したシステムが対応している自然言語は英語と日本語のみである．さらに文字だけの取得や画像のみの取得も可能である．英語側の品詞の矯正のために NLTK(version 3.6.2)[7]を用いる．NLTK は Python で自然言語処理のコーパスを処理できる大規模なライブラリである．単語と品詞情報を同時に取得でき，動詞句と名詞句の変換も内部で実現可能である．統計情報の管理のために pandas(version 1.2.3)[8]を使用する．pandas は，Python 内で DataFrame 型と呼ばれる Excel のような表を持ち，プログラム内でさまざまな統計情報や数値的な計算を可能とする．本研究では，ページごとに pandas の DataFrame 型を作成し，TF-IDF のスコアや重み付けに関する情報などは DataFrame 型で管理する．

4.1　重み付けのための特徴算出

4.1.1　単語の抽出

PDF フォーマットでは，一文字単位でフォントの情報や座標情報を保持する．そのため，抽出した文字を単語として整形する必要がある．あるいは，pdfminer の機能を使うことで，一行単位で文字列の取得が可能である．英語の場合は，単語同士は空白で区切ることができる．その上で，記号 (`()\/.,{}[]:;`) については除去する．ハイフネーション (ハイフン) については，パターンに応じて処理を変える．

ハイフンが行末の場合　この場合，この単語は 1 つの単語であると推測される．そのため，ハイフンを削除し次の行の先頭の文字と結合する．

ハイフンが文中の場合　この場合，2 つの単語を合わせた専門用語であると推測される．そのため，ハイフンはそのままとする．

[6] `https://pdfminersix.readthedocs.io/en/latest/`(参照 2022/01/11)
[7] `https://www.nltk.org/`(参照 2022/01/11)
[8] `https://pandas.pydata.org/`(参照 2022/01/11)

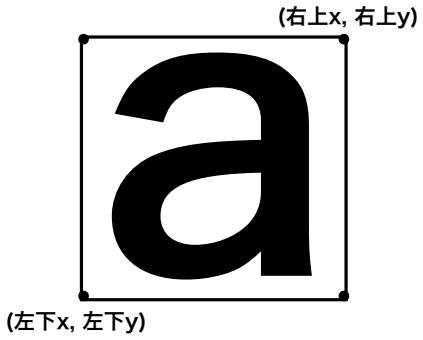


図 21: PDF 内の文字列座標の管理

最後に複数形や過去形などの品詞情報について，NLTK を用いて原型に戻す．日本語の場合は，形態素解析 (本研究では MeCab[9]を使用) を行う．これらの処理を終えた後，文字単位の情報と照らし合わせを行う．それぞれの単語の座標情報やフォントサイズ，フォント名を取得する．座標情報は，PDF フォーマットの中では文字列の情報として (左下の x, 左下の y, 右上の x, 右上の y) の座標が保存されている．図 21 に PDF 中の座標について示す．左下の座標と右上の座標からもフォントサイズの計算が可能である．単語を分割するとき，文字列の先頭座標からフォントサイズと文字数分の座標を動かし，単語の先頭座標を記録する．それらの取得した情報に基づいて，プログラム内部で pandas を用いて表を作成し，各種情報の取得や TF-IDF の計算に用いる．

4.1.2 ページの DataFrame 型

本研究ではページごとに pandas の DataFrame 型を作成する．DataFrame 型の列成分のヘッダーを以下に示す．

一行分のテキスト その行の単語を抽出できているか目視で確認するために使用．

一行の左下の x 座標 第 4.1.5 項で述べる箇条書きのインデントを求めるために使用．

単語 C-TF-IDF の計算や各種重み付けに使用．

フォント名 第 4.1.4 項で述べる太字の取得のために使用．

フォントサイズ 第 4.1.3 項で述べるページタイトルの取得に使用．

[9] `https://taku910.github.io/mecab/`(参照 2022/01/04)

単語の左下の x 座標 第 4.2.2 項で述べるタイトルのみのページの除去のために使用.

単語の左下の y 座標 第 4.2.2 項で述べるタイトルのみのページの除去のために使用.

タイトルの重み 算出したタイトルの重みを保存.

太字の重み 算出した太字の重みを保存.

箇条書きの重み 算出した箇条書きの重みを保存.

C-TF-IDF の値 算出した C-TF-IDF の結果を保存.

正規化した C-TF-IDF の値 正規化した C-TF-IDF の結果を保存.

重み付けした C-TF-IDF の値 重み付けした C-TF-IDF の結果を保存.

pandas では，ヘッダー情報を用いて，特定の列の情報を取得したり，統計情報を求めることができる．テキストや単語，それらの座標情報とフォント名やフォントサイズは，DataFrame 作成時に PDF から取得した情報に基づいて作成する．「一行分のテキスト」は第 4.1.5 節で述べる箇条書き記号の削除は行わず，pdfminer が取得してきた文字列をそのまま登録する．それぞれの重みと C-TF-IDF の値は初期値は 0 で保存する．それぞれの重みは，第 5 節の実験の際の確認のために保存しているが，現在のシステムでは記録のみで参照していない.

4.1.3 各ページタイトルの文字列の取得

タイトル部分の判断には，フォントサイズを用いる．今回対象となっている資料のほとんどは，タイトル部分は通常の文章よりもフォントサイズが大きい．PDF フォーマットの上では，*font-size* として保存されている．ページ内のすべてのフォントサイズを取得し，四分位数を計算する．第三四分位数以上の値がある場合，その部分をタイトルとしている．第三四分位数が確認できない場合，そのページはタイトルなしと判断する.

4.1.4 太字の取得

太字にはフォント名の情報を用いる．PDF フォーマットでは，*font-family* に分類される．英語で作成されたスライドの場合はフォント名に**-bold** が含まれる．日本語で作成されたスライドの場合，現状明確な規則を発見できてないため，科目ごとに太字のフォント名を設定することで対処する．これらのフォント名に基づいて単語の抽出を行う際に，単

語になるいずれかの文字に太字のフォント名が含まれる場合は，その単語を太字にするように設定する．例えば「**de**queue」というのように先頭 2 文字が太字で強調されている場合，本研究では **dequeue** として扱う．

4.1.5　箇条書き

本研究では，箇条書きには座標情報を用いる．DataFrame のヘッダーの内の「一行分の左下の x 座標」を用いる．箇条書きにはいくつかの種類がある．図 22 から図 24 は同じ科目内で用いられている箇条書きの例である．図 22 は，記号を用いた箇条書きの例で，現在取得している資料の中で最も用いられている箇条書きの書き方である．図 23 は，数字を用いた場合である．図 24 は，インデントの先頭には記号を用いていないが，2 段落目のものには記号を用いられている例である．記号には，画像になっている場合とフォントが取得できる場合がある．現在対象としている資料では，○や●はフォントが取得できることを確認している．画像になっているものは，テキストを取得する場合には，座標情報が取得されないため，無視する．フォントが取得できる場合，正規表現を用いることでこれらの文字の座標を取得できる．ページの DataFrame を作成する段階で，正規表現を用いて，○と●分の x 座標を移動させる．数字の場合は，そのまま数字から箇条書きの座標にする．こうすることで，表 8 のように，「一行の左下の x 座標」は一定の座標になる．表 8 の「一行の左下の x 座標」の情報を最も値が小さいものから順にグルーピングして箇条書きの値を決定する．グルーピングする段階で，○と●については取り除く．また，グルーピングする際に，表 8 における「Buffer Overflows」の部分はページタイトルとして事前に取り除去する．

4.2　想定される資料の特徴

4.2.1　科目の資料を読み込む順序

本研究の実装では，授業の順序はそのスライドが入っているディレクトリ名で管理する．例として，ディレクトリは図 25 のような構造となる．このディレクトリのうち，科目の下にあるナンバリングされたディレクリ名を取得し，昇順ソートする．これによって，初回から最終回まで順番に資料を読み込むことができる．初回については，図 25 では，1 としているが，0 から始まっても問題ない．現在の実装では，ナンバリングされていないディレクトリ名は保証していない．このディレクトリは学生が自分の履修状況によって自らで作成することを想定している．毎回の授業資料のディレクトリは 1 つのファイルが入るこ

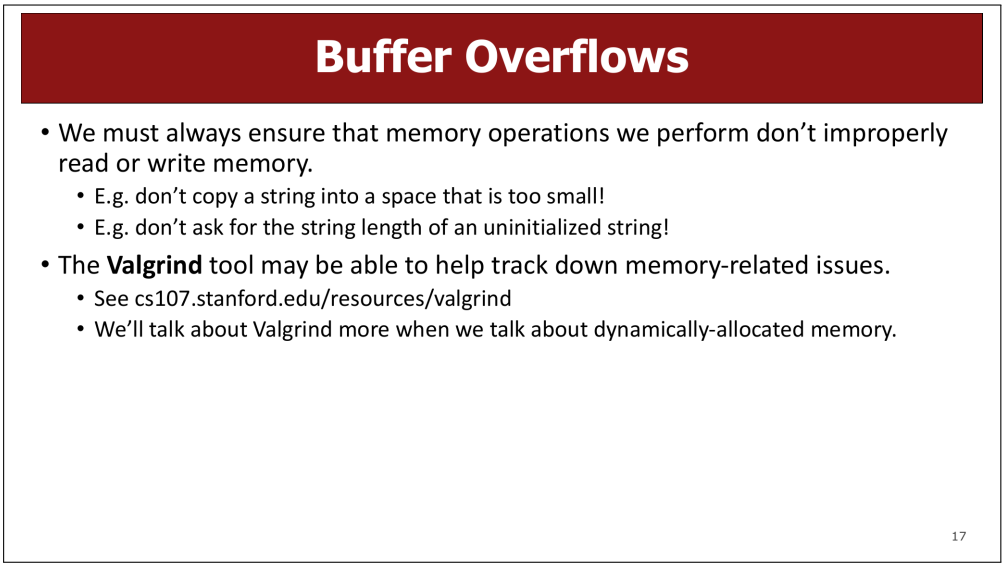


図 22: 最もよく見られる箇条書きの例で，・とインデントで分けられている (CS107[27] より引用)

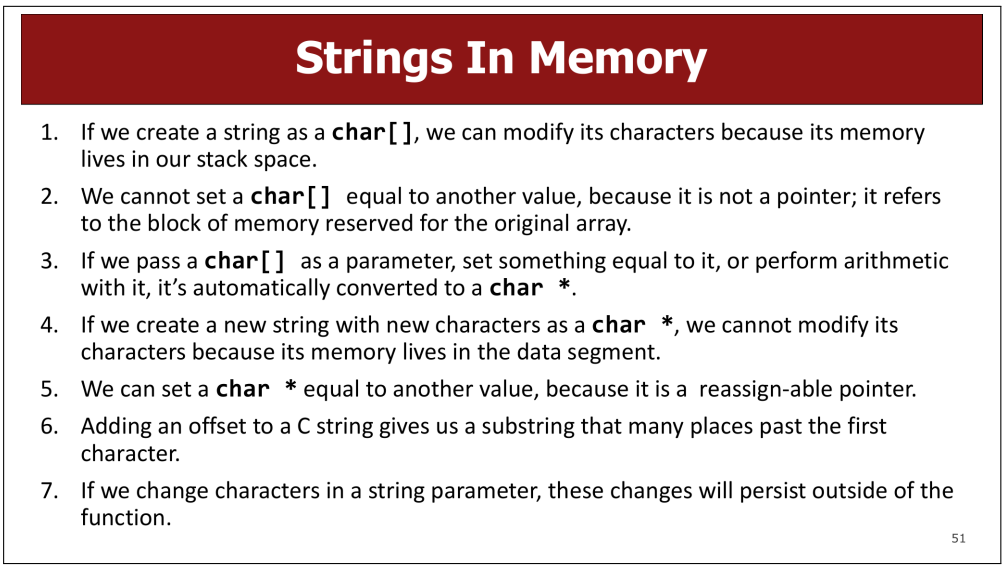


図 23: 数字を使った箇条書きの例 (CS107[27] より引用)

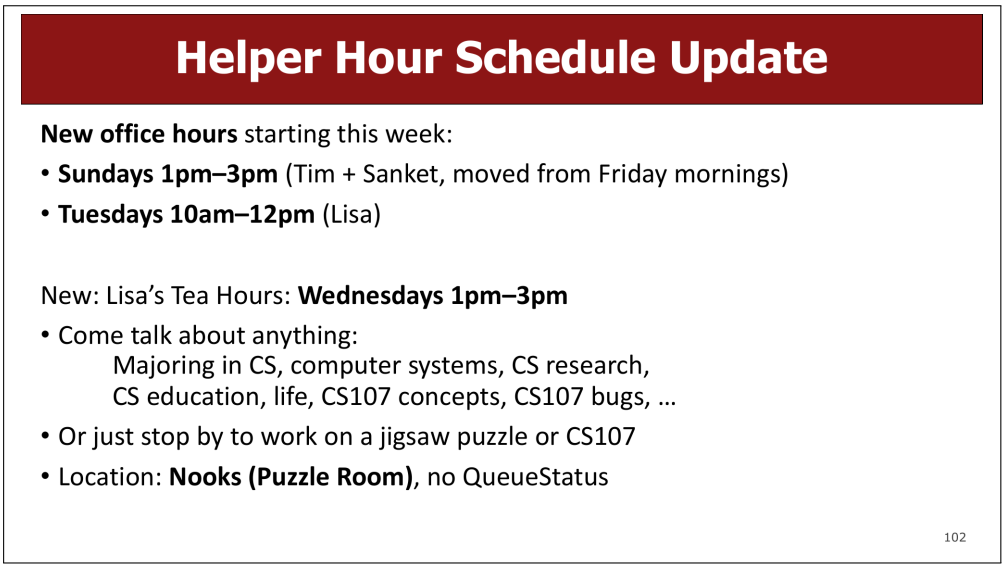


図 24: 途中に箇条書きのマークはあるが，そうではないものがトップのインデントにある例 (CS107[27] より引用)

表 8: 図 22 から生成したデータの一部．... は省略を表す

一行分のテキスト	一行の左下の x 座標	単語
Buffer Overflows	293.15	buffer
Buffer Overflows	293.15	overflow
• We Must always ...	19.2	•
• We Must always ...	19.2	we
• E.g. dont́ ...	55.2	•
• E.g. dont́ ...	55.2	e
• The Valgrind tool ...	19.2	•
• The Valgrind tool ...	19.2	valgrind
• See cs107 stanford ...	55.2	•
• See cs107 stanford ...	55.2	see
17	928.75	17

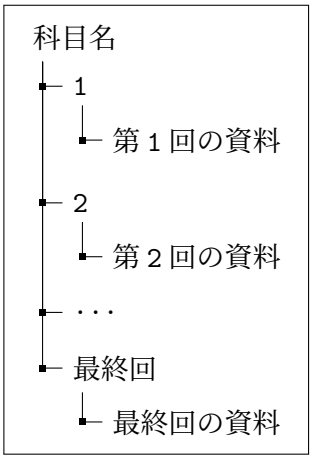


図 25: 資料のファイルを管理するディレクトリ構造の例

とを前提としている．複数のファイルが入っている場合，Python が取得してくるファイル名の順番でファイルを読み込む．

この順序で管理されたディレクトリに従って，pdfminer を用いて第 1 回の第 1 ページから順番に情報を読み込む．C-TF-IDF の計算もこのディレクトリの番号順に内部で 1(0 から始めた場合は 0) から順に辞書型として一対一の対応関係を設定する．これによってシステム内で常に時系列順を保つことができる．

4.2.2 タイトルのみのスライドの除去

本研究ではタイトルのみのスライドは除去する．タイトルのみのスライドは図 26 のようなものを想定し，文字の y 座標が中央に寄っているような特徴を持つものとする．現在の実装上は，最初のテキストの y 座標が，スライドサイズの 3/4 以下の場合をタイトルのみのスライドと仮定している．PDF フォーマットの中でスライドサイズは *MediaBox* に格納されている．この *MediaBox* も文字の座標と同様に左下が (0,0) となる．このとき，タイトルを取得して比較を行わないのは，タイトルだけだった場合その文字列がタイトルとして認識されないためである．しかし現在の手法では，図 27 のようなスライドはタイトルのみのスライドと判断されない．図 27 は，文字列「Next step for lookup-based structures...」が 3/4 より大きい座標に配置されているためである．この問題を解決する方法として，単語数が一定値以下の場合に除外するという方法である．しかし，第 3.3 節で取り上げたように，タイトルと図で構成されるスライドは少なからず存在する．そういったスライドに

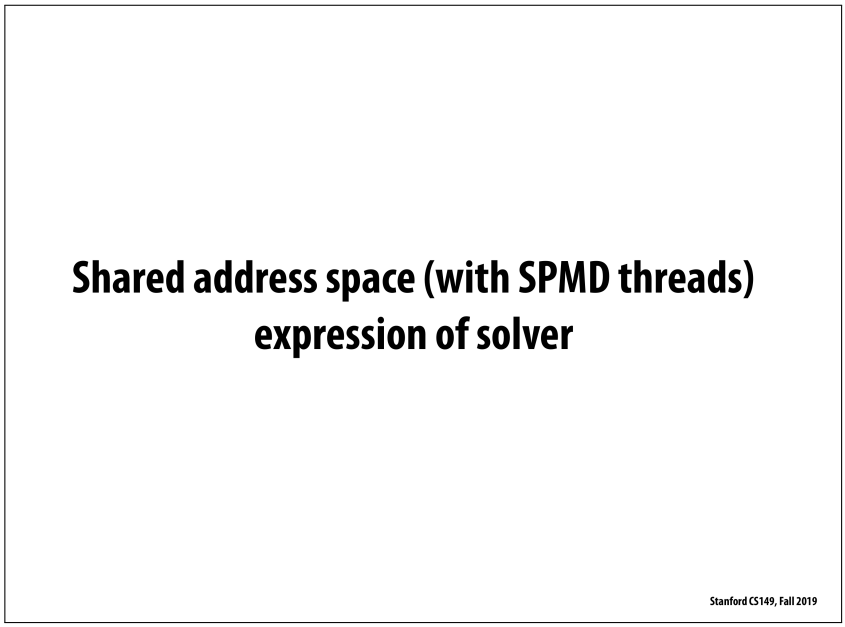


図 26: タイトルのみのスライド (CS149[28] より引用)

ついて除外しても問題がないかどうかは，今後実験で得られる知見に基づき判断する．あるいは，科目ごとにスライドの特徴を管理し，そのルールに従って除外するという方法も考えられる．資料作成者あるいは学生の負担が増えるため，実際に使用する場合にはあまり有効な手法ではない．そのため，本研究では一部タイトルのみと解釈できるスライドが含まれている．

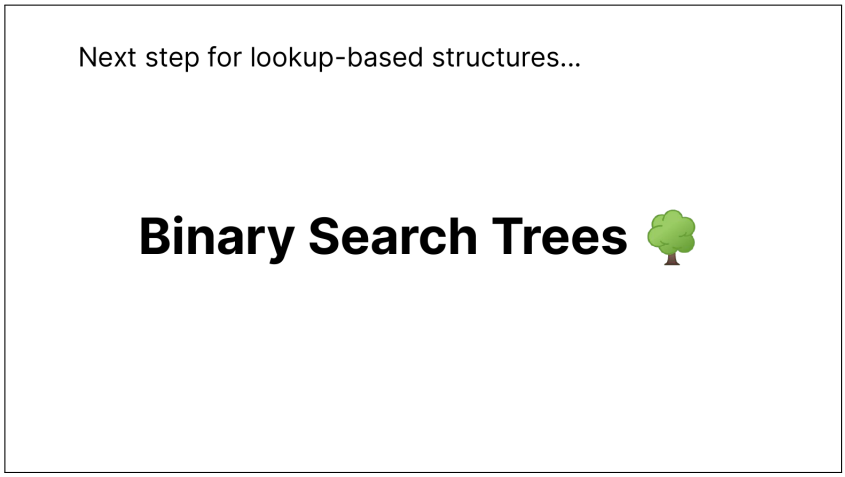


図 27: 上部に文字があるためタイトルのみと判断されないスライド (CS106B[5] より引用)

5 実験

提案手法の有効性を検証するための評価実験を行った．本研究では，Stanford University の CS コースの科目として公開されている Web ページ上のスライドを用いた．本研究で利用する科目の詳細は以下に示す．

CS106B [**5**] C++を用いてデータ構造の基礎を学習.

CS106L [**29**] C++でプログラムの記述方法を学習.

CS107 [**27**] C 言語を用いて，コンピュータの構成やアーキテクチャの部分を学習.

CS110 [**6**] CS107 の続きの科目．仮想メモリやネットワークなどについて学習する.

CS143 [**8**] コンパイラの基礎的な動作を学び，実際に独自言語を用いて設計.

CS149 [**28**] 並列コンピュータの基礎を学習.

CS161 [**30**] アルゴリズムとデータ構造について計算量の観点から学習.

CS166 [**7**] 応用的なデータ構造を学習.

CS205L [**31**] 機械学習分野の数学的観点を学習.

CS224N [**9**] 機械学習を用いた自然言語処理.

CS243 [**32**] コンパイラの視点からプログラムの最適化.

また，これらの科目のナンバリングは Stanford University の CS コースの Web ページ [33] で以下のように区別されている．上記の科目に必要な情報のみを抜粋して示す．実験に使用した科目は基礎科目を中心に応用科目を選んでいる.

100-199 other service courses, basic undergraduate

200-299 advanced undergraduate/beginning graduate.

00-09 Introductory, miscellaneous.

10-19 Hardware Systems.

20-29 Artificial Language.

40-49 Software Systems.

60-69 Analysis of Algorithms.

作成したシステムは，それぞれの科目について第 1 回の第 1 ページから最終回の最終ページまで読み込む．読み込んだページの情報を用いて「C-TF-IDF(通常の C-TF-IDF)」，「正規化した C-TF-IDF」，「正規化と重み付けをした C-TF-IDF(重み付けした C-TF-IDF)」の 3 つのスコアを求め，これらを比較する．図 28 から図 35 を学習中であると仮定し，**array**, **heap**, **stack**, **memory**, **queue**, **pointer**, **tree**, **hash** に学生が注目したと仮定し，システムがランキングのスコアを用いる．この結果に対して，以下の実験を行った．

筆者による実験 この実験は全てのスライドを網羅的に確認し，スコアリングの上位の妥当性を検証するものである．11 科目 8 単語の別々の結果それぞれに筆者が「将来のスライドとして出てきて欲しい」という基準でランキングを作成した．3 種類のスコアリング手法における上位 3 位と筆者が作成した上位 3 位以内の結果を評価の対象とする．

被験者による実験 スコアリング上位のスライドを対象とし，順位付けの妥当性を筆者以外が検証する実験である．11 科目の資料と 8 種類の単語の別々の結果を結合し，3 種類のスコアリング手法における上位 5 位を結果を用いて，被験者にランキングの作成を依頼した，8 種類の単語について基準となるスライドを用意し，「一緒に表示されて嬉しいと思う」順にランキングを作成してもらった．ランキング上位 5 位の結果を評価の対象とする．

このとき 3 種類の結果に対してランキング内で類似性が高いスライドは筆者が目視で判断し取り除いた．類似性が高いスライドは，パラパラ漫画のように一つの説明に複数のページを使う場合や同じ説明を複数回行った場合に現れる．これらのスライドはグルーピングして 1 つのものとして提示可能であると考えているが，現時点で実装が十分ではないため，類似したスライドは取り除くのみとした．ランキングが上位のスライドを元のスライドとし，そのスライドと類似すると思われるスライドで文章量で 2 行分の差があると判断できる場合は類似しないスライドとした．実験結果の一部には通常の C-TF-IDF では取り除いたスライドが重み付けした C-TF-IDF では出現する場合もある．これについては，筆者個人の実験では，データをまとめる段階でそのスライドの違いを筆者が手動で吸収することで解決する．被験者による実験では，別のスライドとして扱う．筆者のランキング作成と被験者による実験のランキング作成は実施方法が異なるため，詳細はそれぞれの節で述べる．学習中と仮定したスライドはすべて筆者が独断で選択し，被験者による実験用に作成したランキング内に重複した画像がないか確認した上で使用した．学習中と仮定したスライドの選定基準は，比較的基礎的な内容を扱っていると判断したものである．そのため，

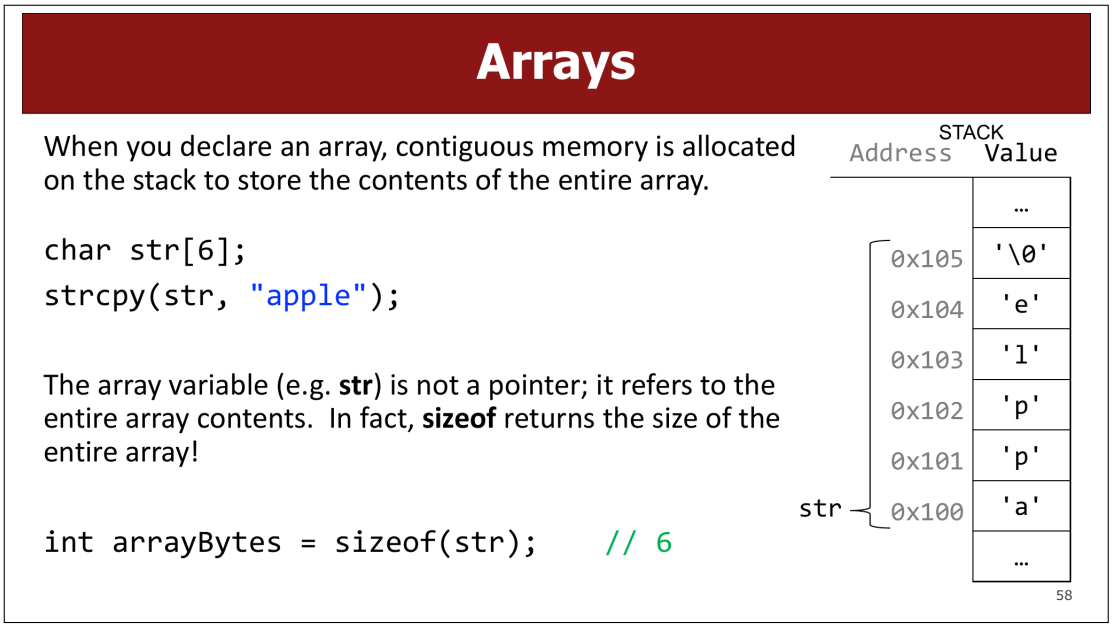


図 28: **array** に注目したと仮定したスライド (CS107[27] より引用)

CS106B や CS107 など学部を対象とし，CS コースの導入の科目としてナンバリングされている科目の中から選ばれたものが多い．これにより，これらのスライドを中心に発展する複数のコースを考慮することができる．

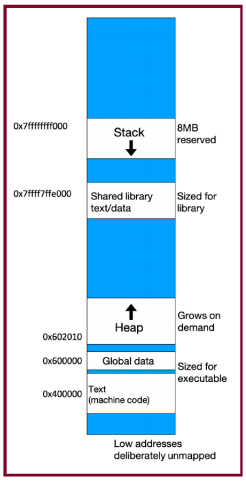
5.1 評価指標

本研究では評価の指標として Q-measure を用いる．Q-measure は Sakai[34][35] が提案している情報検索システムの評価に用いられる手法の 1 つである．図 36 を用いて，Q-measure について説明する．この手法では，「大正解」，「正解」，「おまけの正解」をシステムが検索してきた場合に事前に設定した得点を与える．ここでは，システムが「大正解」を提示した場合には 3 点，「正解」で 2 点，「おまけの正解」で 1 点，「不正解」の場合には 0 点とする．第 r 位までにシステムが検索してきた得点の合計を理想的な順序で得られた時の値で割ることで，ランキングの第 r 位のスコアを求めるのが基本的な考え方である．ブレンド比は，得点の値がどのような場合でも理想的な順序とシステムが出した順序の比を求めるために用いる．ブレンド比 $BR(r)$ は，以下の定義を用いて，式 (20) と定義できる．

The Heap

- The **heap** is a part of memory that you can manage yourself.
- The **heap** is a part of memory below the stack that you can manage yourself. Unlike the stack, the memory only goes away when you delete it yourself.
- Unlike the stack, the heap grows **upwards** as more memory is allocated.

The heap is **dynamic memory** – memory that can be allocated, resized, and freed during **program runtime**.



52

図 29: **heap** に注目したと仮定したスライド (CS107[27] より引用)

New ADT: Stack

```
#include "stack.h"

Stack<string> recentCalls;
recentCalls.push("Neel");

recentCalls.push("Julie");

recentCalls.push("Esteban");

recentCalls.push("Minh");

while (!recentCalls.isEmpty()) {
    cout << recentCalls.pop() << " ";
}
```

Source: http://www.flickr.com/photos/35237093334@N01/409465578/
Author: http://www.flickr.com/people/35237093334@N01 Peter Kazanjy]

Stanford University

図 30: **stack** の注目したと仮定したスライド (CS106B[5] より引用)

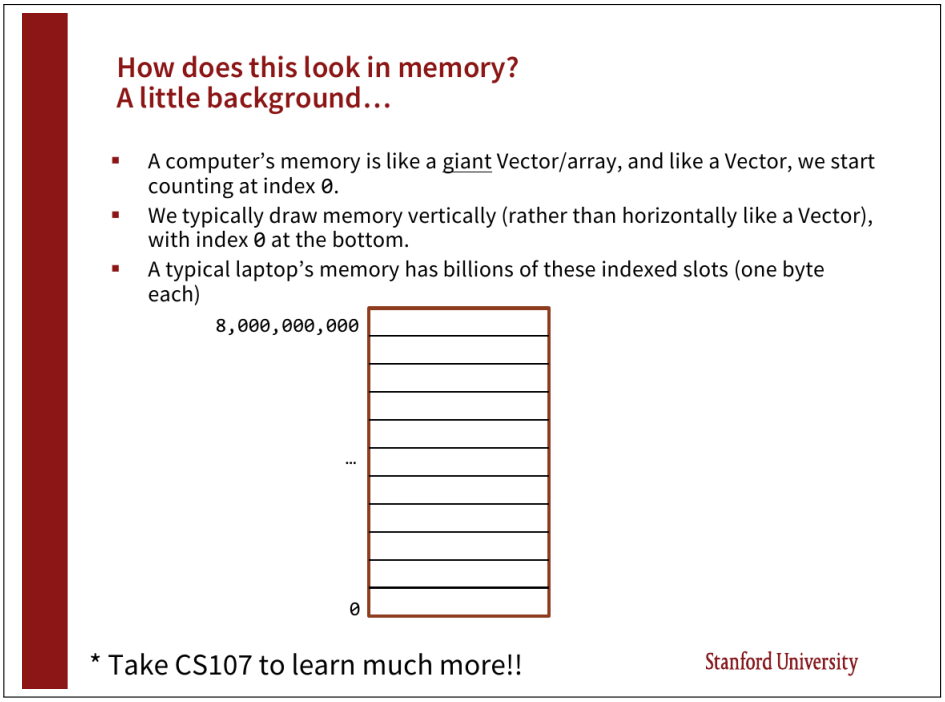


図 31: **memory** の注目したと仮定したスライド (CS106B[5] より引用)

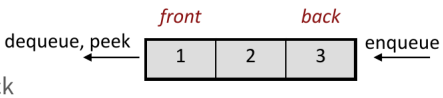
表 9: ブレンド比を求めるための定義

r	ランキング第 r 位.
$cg(r)$	第 r 位までにシステムが受け取った得点の合計.
$count(r)$	第 r 位までの得点を受け取った回数.
$cg_I(r)$	第 r 位までの理想的な得点の合計.

Queues

queue: First-In, First-Out ("FIFO")

- Elements stored in order they were added
- Can add only to the back, can only examine/remove frontmost element



queue operations

- enqueue: Add an element to the back
- dequeue: Remove the front element
- peek: Examine the front element

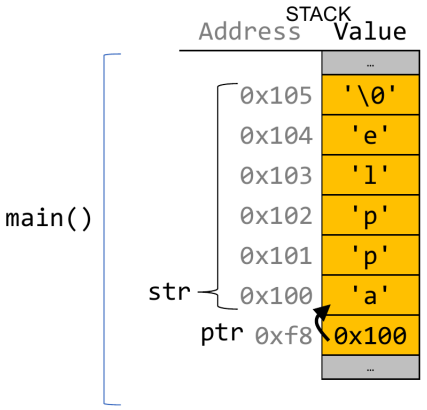
Stanford University

図 32: **queue** の注目したと仮定したスライド (CS106B[5] より引用)

Arrays and Pointers

We can also make a pointer equal to an array; it will point to the first element in that array.

```
int main(int argc, char *argv[]) {
      char str[6];
      strcpy(str, "apple");
      char *ptr = str;
      ...
}
```



75

図 33: **pointer** の注目したと仮定したスライド (CS107[27] より引用)

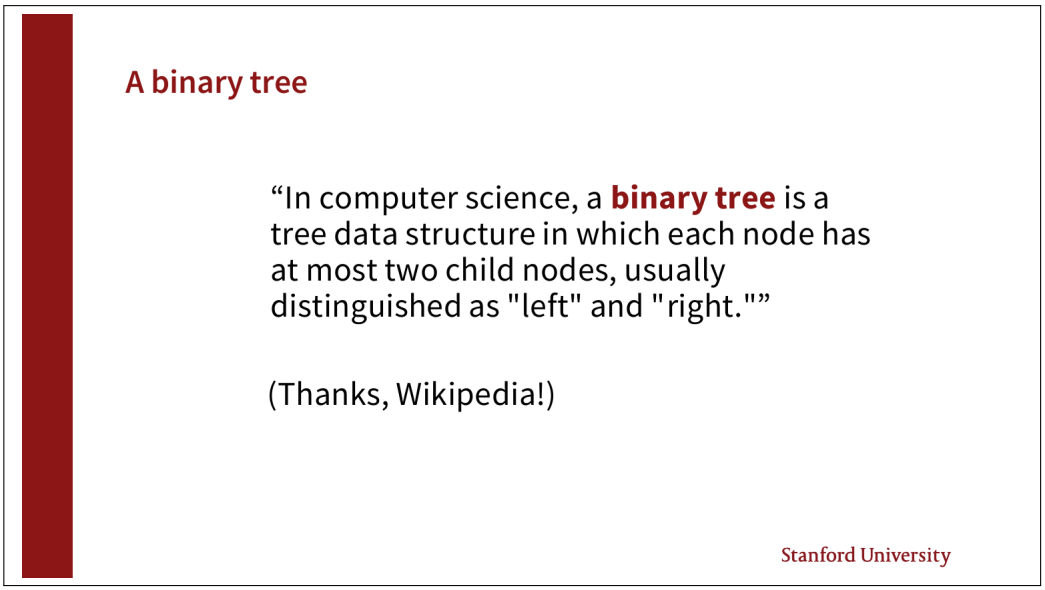


図 34: **tree** の注目したと仮定したスライド (CS106B[5] より引用)

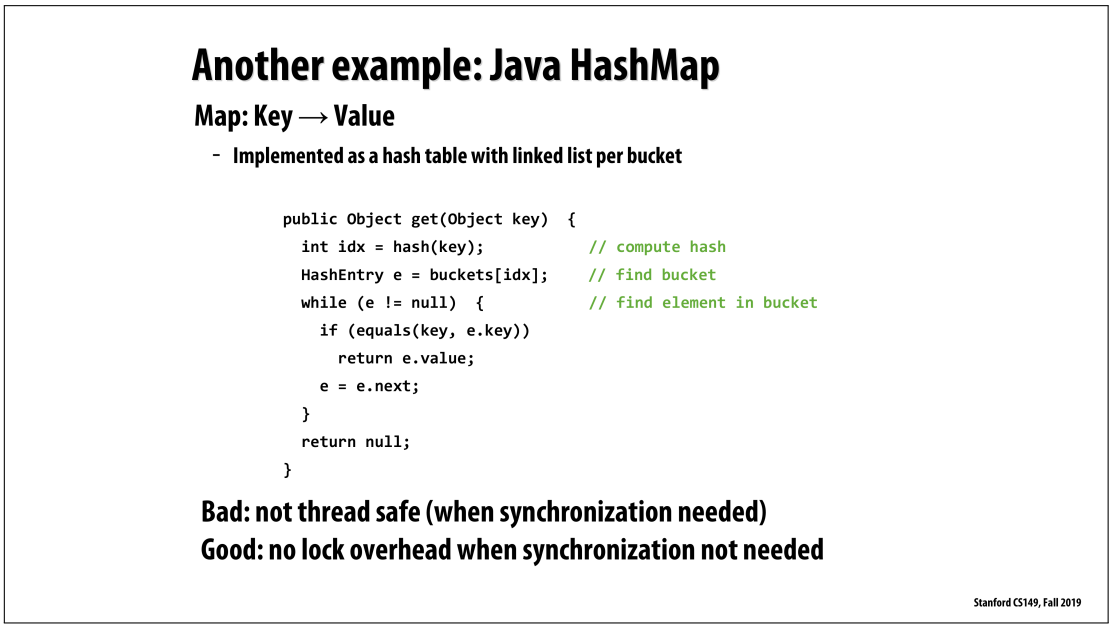


図 35: **hash** の注目したと仮定したスライド (CS149[28] より引用)

$$BR(r) = \frac{cg(r) + count(r)}{cg_I(r) + r} \tag{20}$$

Q-measure は,「各正解が検索された時点でのブレンド比を，全正解で平均したもの」[35] である．式 (21) は，Q-measure を求める式である．L は検索するデータのサイズ，R は r までの正解の数，I は得点の有無のフラグで 1 か 0 となる．

$$Q\text{-}measure = \frac{\sum_{r=1}^{L} I(r)BR(r)}{R} \tag{21}$$

Q-measure を用いて，図 36 から実際に値を求める．システムの第 1 位は,「不正解」であるため，0 となる．第 2 位は,「大正解」であるため，$cg(2) = 3$，$count(2) = 1$，$cg_I(2) = 5$, $r = 2$ であるため，$(3+1)/(5+2) = 4/7$ となる．第 3 位は,「おまけの正解」であるため，$cg(3) = 4$，$count(3) = 2$，$cg_I(3) = 6$, $r = 3$ であるため，$(4+2)/(6+3) = 6/9$ となる．第 2 位と第 3 位の合計と $R = 3$ で割った値 0.4126 が Q-measure の結果である．例えば，第 3 位が「正解」である場合，$cg(3) = 5$，$count(3) = 2$，$cg_I(3) = 6$, $r = 3$ であるため，$(5+2)/(6+3) = 7/9$ となり，Q-measure の値は 0.8306 となる．「大正解」や「おまけの正解」をシステムが検索してきた場合，Q-measure は高くなる．実験では，筆者が作成したランキングであれば上位 3 位以内，被験者が作成したランキングであれば上位 5 位以内の結果を関連するページとして最適なものと仮定する．本研究では，理想的な結果を筆者が作成したランキングは，得点を第 1 位から順に 3 点，2 点，1 点として Q-measure を求める．被験者が作成したランキングでは得点を第 1 位から順に第 5 位までを 5 点，4 点，3 点，2 点，1 点とした．

5.2 筆者による実験

筆者による実験では，科目ごとに評価を行う．これは，システムが科目ごとに C-TF-IDF の計算を行うためである．システムは提案手法に基づいて対象となる単語を含むスライドの C-TF-IDF のスコアを算出する．システムが算出した結果を降順に並び替え，ランキングとする．筆者はそれぞれの科目とそれぞれの単語について，単語を含むすべてのスライドの中から目視で上位 3 位を選びランキングを作成する．11 科目 8 単語で 88 件あるが，実際に対象となったのは 11 科目合計で 69 件，総ページ数は 5436 ページであった．筆者が作成したランキングとシステムが算出したランキングについて上位 3 位の Q-measure を評価する．合わせて，通常の C-TF-IDF と正規化した C-TF-IDF，重み付けした C-TF-IDF それぞれの値の違いについても示す．

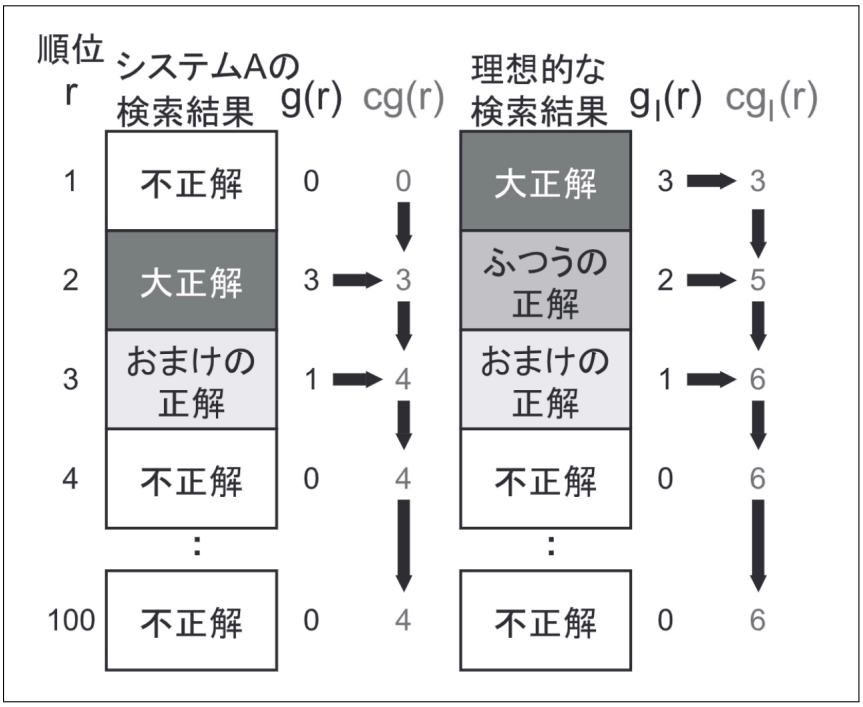


図 36: システムの検索結果に対して，大正解や正解で得点を与える評価手法 [35]

表 10: CS166 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成の ランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#16 p.77	0.4342	#16 p.77	0.3653	#04 p.105	25.9503	#17 p.33
2 位	#10 p.165	0.3578	#10 p.166	0.3417	#18 p.4	5.3742	#04 p.105
3 位	#04 p.106	0.3293	#17 p.33	0.2754	#16 p.77	5.2418	#10 p.118

表 11: 表 10 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.5039

5.3 実験結果と考察

すべての結果は付録に示し，ここでは特徴的ないくつかの結果についてのみ示す．

表 10 は，図 28 に基づいた，CS166[7] における，通常の C-TF-IDF，正規化した C-TF-IDF，重み付けした C-TF-IDF の結果である．Q-measure の値を表 11 に示す．#16 の p.77 を図 37 に，#4 の p.105 を図 38 に，#17 の p.33 の結果を図 39 に示す．図 37 は#16 の最後のページで次回実施する内容について述べている．図 38 は線形リストを動的に作成する場合の実装方法について取り上げている．図 39 は文字列探索のアルゴリズムである `Suffix Arrays` について取り上げていて，探索の対象の文字列が「banana」である場合の例である．筆者は図 28 の画像がメモリ上の配列の扱いであったことと，CS166 がアルゴリズムの内容であったから，配列を使ったアルゴリズムの特徴的なものを中心に選んだ．通常の C-TF-IDF で 1 位となった図 37 は，学生がこの科目を学習済みの場合に内容を思い出す手助けになると考えられる．しかし，将来の科目を示すという目的を考慮すると，1 つの内容に絞った具体的な例を示す方が有効であると考える．Q-measure の結果を見ると，重み付けした C-TF-IDF は，筆者が想定するスライドを選ぶことができているといえる．

表 12 は，図 30 に基づいた，CS107[27] におけるシステムが選んだスライドである．表 12 に対する Q-measure の値は表 13 となった．通常の C-TF-IDF および正規化した C-TF-IDF で最も値が大きい#13 のの p.34 は図 40 である．このスライドは，アセンブラの命令によって stack 領域が変化する様子を示したものである．重み付きの C-TF-IDF で最も値が高い #7 のの p.5 は図 41 である．このスライドは関数呼び出しによる stack 領域の変化をパラパ

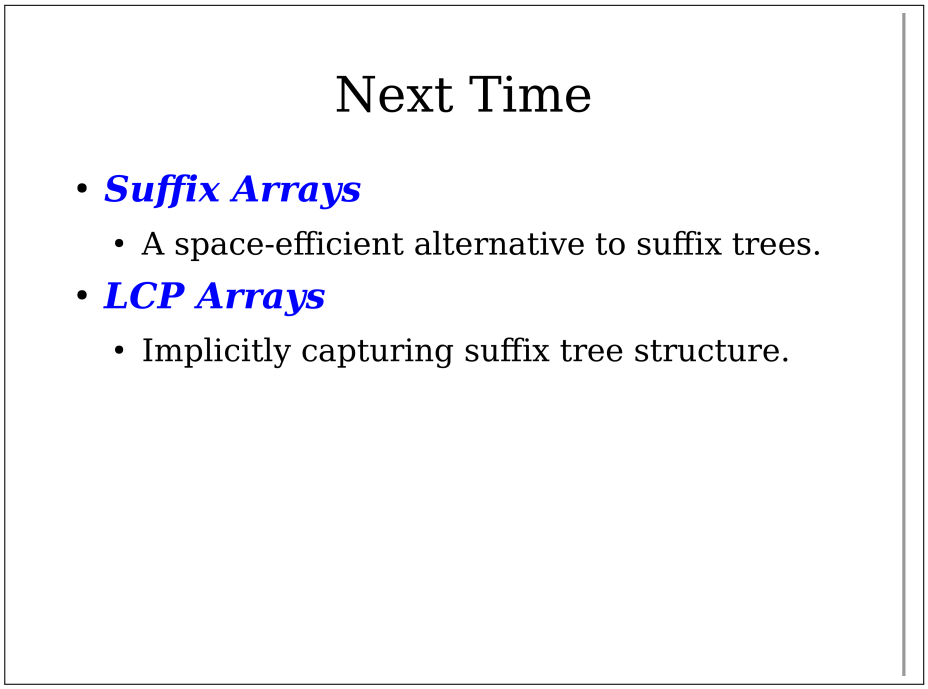


図 37: #16 の p.77(CS166[7] より引用)

Dynamic Arrays

- A ***dynamic array*** is the most common way to implement a list of values.
- Maintain an array slightly bigger than the one you need. When you run out of space, double the array size and copy the elements over.

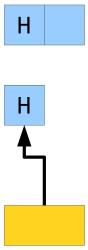


図 38: #4 の p.105(CS166[7] より引用)
[btp]

Suffix Arrays

- A ***suffix array*** for a string T is a sorted array of the suffixes of the string T\$.
- Suffix arrays distill out just the first component of suffix trees: they store suffixes in sorted order.

$
A$
ABANANABANDANA$
ABANDANA$
ANA$
ANABANDANA$
ANANABANDANA$
ANDANA$
BANANABANDANA$
BANDANA$
DANA$
NA$
NABANDANA$
NANABANDANA$
NDANA$

ABANANABANDANA$

図 39: #17 の p.33(CS166[7] より引用)

表 12: CS107 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成の ランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#13 p.34	0.3655	#13 p.34	0.5226	#07 p.5	13.1542	#08 p.99
2 位	#08 p.134	0.3276	#08 p.134	0.3645	#13 p.34	7.4636	#07 p.83
3 位	#08 p.99	0.2622	#08 p.99	0.3218	#08 p.134	4.8886	#13 p.23

ラ漫画のように述べている途中の p.1 である．通常の C-TF-IDF と正規化した C-TF-IDF では 3 位で，筆者も選んでいる#8 の p.99 は図 42 である．このスライドは，stack の動きを視覚的に示し，push と pop についても述べている．筆者は，stack の動きを視覚的に示したものと，そのスライドから別のスライドに発展する可能性があるスライドを選んだ．図 40 は，アセンブラやコンピュータアーキテクチャについて理解したあと改めて内容を整理する際に有効であるといえる．また，評価元となった図 30 は積み上がる様子を示しているため，学生にとってはコンピュータ内部での動作を整理することにつながる内容であるといえる．図 41 は，すでに stack を知っている筆者の視点から考えると理解しやすいものだといえるが，初めて学ぶときにこのスライドのみを示されても理解は容易ではないだろう．こういったスライドは類似度が高いスライドが連続するため，ショートアニメーションのようにスライドを連続して表示するやグルーピングして複数まとめて示すことで，学生にとってわかりやすい提示につながると考えられる．図 42 は push や pop，peek など技術的な単語を知るきっかけになる．学生がこのスライドからさらに別のスライドを知ることで，科目内で散り散りになっている知識を結びつける効果を期待できる．これらのことから，この結果では通常の C-TF-IDF や正規化した C-TF-IDF のほうが筆者の考えに近いスライドを示している．図 41 が第 1 位になった理由として，重み付けの効果が強く出たことが考えられる．#7 のの p.5 は本研究の実装では初めて stack がスライドタイトルとして出現したスライドであった．提案手法で重要視しようとしている初めて出現したスライドが必ずしも有効なスライドとは限らないということである．そのため重み付けの方法を変更することや，Goto ら [19] が行った受け取った学生自身による提示結果の判断などで，提示するスライドを調整していく必要がある．また図 41 はソースコードによって説明されたスライドである．このスライドについては，stack の動きを示した例の 1 つであるため，ソースコードを重要視したことは有効であると考えている．

表 14 は，図 32 に対する，CS107[27] の結果である．図 43 は，重み付けした C-TF-IDF と筆者が作成したランキング内で第 1 位となったスライドである．学習中と仮定したスラ

表 13: 表 12 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.0

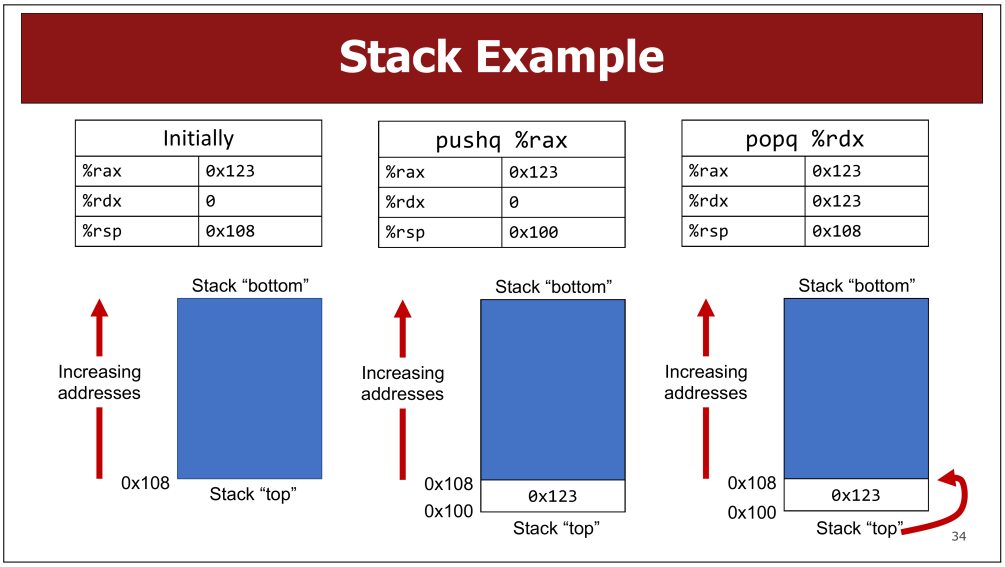


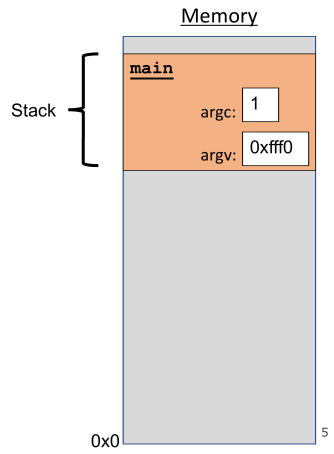
図 40: #13 の p.34(CS107[27] より引用)

The Stack

```
void func2() {
    int d = 0;
}

void func1() {
    int c = 99;
    func2();
}

int main(int argc, char *argv[]) {
    int a = 42;
    int b = 17;
    func1();
    printf("Done.");
    return 0;
}
```

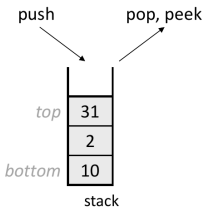


5

図 41: #7 の p.5(CS107[27] より引用)

Refresher: Stacks

- A **Stack** is a data structure representing a stack of things.
- Objects can be ***pushed*** on top of or ***popped*** from the top of the stack.
- Only the top of the stack can be accessed; no other objects in the stack are visible.
- Main operations:
  - **`push(value)`**: add an element to the top of the stack
  - **`pop()`**: remove and return the top element in the stack
  - **`peek()`**: return (but do not remove) the top element in the stack



99

図 42: #8 の p.99(CS107[27] より引用)

表 14: CS107 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成の ランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.93	0.2520	#11 p.10	0.3027	#05 p.21	12.2411	#05 p.21
2 位	#07 p.68	0.2291	#02 p.46	0.2984	#11 p.10	7.0997	#05 p.72
3 位	#02 p.46	0.1954	#07 p.52	0.2920	#07 p.68	2.9367	#02 p.46

表 15: 表 14 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0741
正規化した C-TF-IDF	0.1693
重み付けした C-TF-IDF	0.6720

イドでは，「CS107 でより深く説明される」とされており，実際に関連するスライドを示した例である．このようにして直接発展的な内容を示すことで，学生は **memory** に対する学習内容を整理することができる．また，図 44 は，メモリリークについて説明しているスライドである．このスライドは筆者がランキング作成時には考慮していなかったが，システムは発見したスライドである．こうしたメモリリークに関する内容を，初学者が意識することで，初めて実装する際には注意を払うことができる．加えて，実装に慣れてきて，意識せずにプログラムを書くことができるようなってきた頃にこのスライドを示すことで，自らの実装を見直すことにつながる．表 15 の Q-measure の値は重み付けした C-TF-IDF が最も高くなった．特に第 1 位が一致しているため，最も重要と考えられるスライドを発見できているといえる．

図 34 における，CS224N[9] の結果を表 16 に示す．表 16 に対する Q-measure の値は表 17 となった．これらに変化がなかった．これは CS224N では **tree** を検索語とした場合，対象となるスライドが少なかったことが原因と考えられる．その中で，学生に対して効果的なスライドを発見するためには重み付けが重要になると考えている．すべての結果で 1 位となった#17 の p.50 は第 3.1 節の木構造の例として図 45 で示している．図 45 は#13 の p.5 である．このスライドは，#13 ので利用する例文を示している．筆者が選んだ#4 の p.17 を図 46 に示す．図 46 は，文法を木構造で表現した場合の図である．システムが選んだスライドは順位を含めすべて同一であるが，重みづけによって#17 の p.50 が他よりも大きく

[6] CS107[27]

Memory

- Memory is a big array of bytes.
- Each byte has a unique numeric index that is commonly written in hexadecimal.
- A pointer stores one of these memory addresses.

Address	Value
	…
0x105	'\0'
0x104	'e'
0x103	'l'
0x102	'p'
0x101	'p'
0x100	'a'
	…

21

図 43: CS107[27] の#5 の p.21

Memory Leaks

- A memory leak is when you allocate memory on the heap, but do not free it.
- Your program should be responsible for cleaning up any memory it allocates but no longer needs.
- If you never free any memory and allocate an extremely large amount, you may run out of memory in the heap!

However, memory leaks rarely (if ever) cause crashes.

- We recommend not to worry about freeing memory until your program is written. Then, go back and free memory as appropriate.
- Valgrind is a very helpful tool for finding memory leaks!

68

図 44: CS107[27] の#5 の p.21

表 16: CS224N における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成の ランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#17 p.50	0.2384	#17 p.50	0.3005	#17 p.50	9.7129	#17 p.50
2 位	#13 p.5	0.2355	#13 p.5	0.1999	#13 p.5	0.7610	#04 p.17
3 位	#13 p.23	0.1571	#13 p.23	0.1941	#13 p.23	0.7391	#02 p.11

表 17: 表 16 の Q-measure

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.6719
重み付けした C-TF-IDF	0.6719

重視されている．図 45 は，学生がデータ構造や抽象的な木構造を学習した上で自然言語処理に応用されていることを知ることができる．本研究の狙いが最も反映された図の 1 つである．図 45 は単純な例文であり，これは示す必要がないものである．このスライドが上位となった理由は，C-TF-IDF に原因があると考えている．この科目の **tree** という単語は #4 の p.34 の次は#13 の p.5 まで出現しない．第 3.2 節で述べたが，C-TF-IDF では，初めて単語が出現したスライドや期間が空いた後のスライドの値が大きくなる．これはこの科目だけでなく，他の科目でも同様の結果が確認できている．これにより，出現期間が空いた後に説明しているスライドを有効に評価できている．今回実験に用いた単語では，学生に対して良い効果を生むスライドを発見できているが，そうではないスライドが出現する可能性もある．そういったスライドが持つ固有の特徴を改めて整理する必要がある．

A couple of years later, Vanaja met Akhila at the local park. Akhila's son Prajwal was just two months younger than her son Akash, and they went to the same school. For the pre-school play, Prajwal was chosen for the lead role of the naughty child Lord Krishna. Akash was to be a tree. She resigned herself to make Akash the best tree that anybody had ever seen. She bought him a brown T-shirt and brown trousers to represent the tree trunk. Then she made a large cardboard cutout of a tree's foliage, with a circular opening in the middle for Akash's face. She attached red balls to it to represent fruits. It truly was the nicest tree.

From The Star by Shruthi Rao, with some shortening.

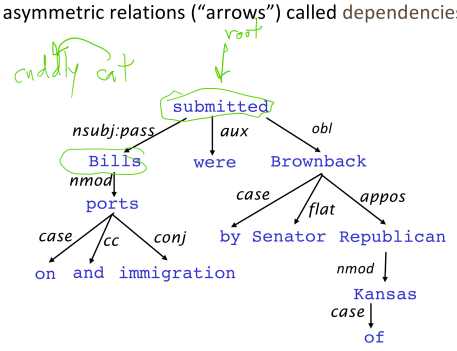
図 45: #13 の p.5(CS224N[9] より引用)

Dependency Grammar and Dependency Structure

Dependency syntax postulates that syntactic structure consists of relations between lexical items, normally binary asymmetric relations ("arrows") called dependencies

An arrow connects a head (governor, superior, regent) with a dependent (modifier, inferior, subordinate)

Usually, dependencies form a tree (a connected, acyclic, single-root graph)



21

図 46: #4 の p.17(CS224N[9] より引用)

表 18: アンケートの実施項目

この画像と同時に出て嬉しいものついて，以下の URL の画像からランキング作成してください．画像の URL https://drive.google.com/drive/folders/1GDOoUQ5RsF_vmg5I9ceOan1UKO_QJFqe?usp=sharing
あなたがそれの選択肢を選んだ理由についてお聞かせください．

5.4 被験者による実験

システムが算出した個別の結果を合算し，降順に並び替え，第 5 位までを整理した．そこから重複したスライドを排除し，学習中と仮定したスライドと一緒に出てきて嬉しいスライドのランキングの作成を行ってもらった．明星大学情報学部 3 年生 4 名，同学部 4 年生 2 名，大学院情報学専攻博士前期課程 1 年生 2 名，国士舘大学理工学部 2 年生 1 名，北陸先端科学技術大学院大学博士前期家庭 2 年 1 名 (国士舘大学および北陸先端科学技術大学院大学の学生は学部の学生であったため所属は収集していない) の計 10 名を対象にして，Google Form を使用したアンケート形式で実施した．学習前の内容と学習済みの内容を分けた場合，被験者の負担が大きいと考え今回の実験では実施していない．どの単語を用いたかは非公開とし，ランキングの結果から重複を排除した 8 枚から 13 枚のスライドを対象に，1 位から最大で 13 位までランキングを作成してもらった．合計で p.81 が実験の対象となった．また，それぞれの科目名やスライド番号などの情報も非公開とした．各設問の最後には数行程度の記述形式の設問も用意し，それぞれの設問のランキング作成理由も聞いた．アンケートの内容を表 18 に示す．このアンケートは図 28 から図 35 までの 8 項目分，同様のものが続く．ランキングは図 47 のように，フォーム内では学習中と仮定したスライドとランキングの選択肢を用意し，添付した URL から実際の選択肢の画像を閲覧しながら被験者が作成した．この実験では，被験者が作成したランキングとシステムから得られた結果を比較し，被験者のランキングごとに Q-measure を求める．Q-measure で用いる得点は，第 1 位から順に 5 点，4 点，3 点，2 点，1 点とした．

5.5 実験結果と考察

全ての結果は付録に示し，ここでは特徴的な結果のみを示す．

システムが算出した結果に基づく **queue** のランキングの結果を表 19 に示す．このうち，CS166 の#4 の p.13 と p.33 は類似したスライドであるが，完全に重複したスライドではないと考え被験者には両方を示した．被験者が作成したランキングに基づく Q-measure の値と平均を表 20 に示す．被験者全体で 8 名中 7 名が，第 5 位以内に選んだ，CS166 の#4 の

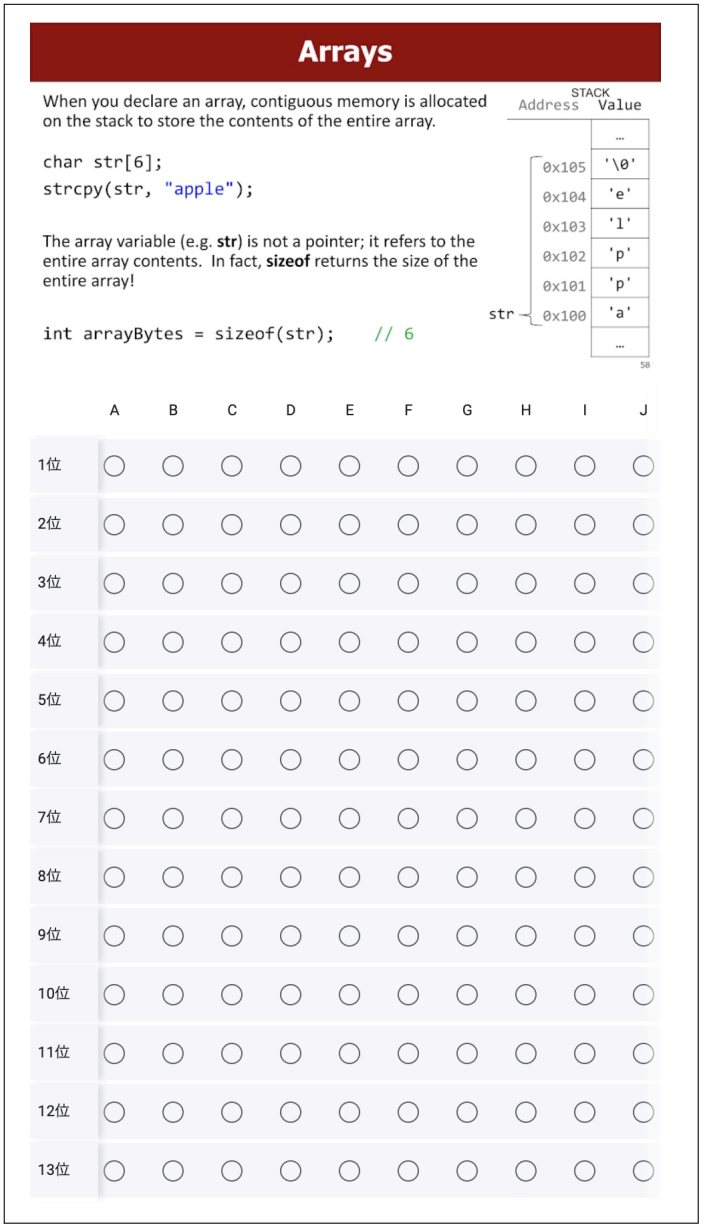


図 47: 実際のアンケート画面の 1 つ

表 19: **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS166 #04 p.13	1.2278	CS166 #04 p.13	0.7458	CS166 #04 p.33	7.8094
2 位	CS149 #05 p.18	0.4960	CS149 #05 p.37	0.6178	CS166 #04 p.103	5.3006
3 位	CS106L #04 p.05	0.4514	CS149 #05 p.18	0.5916	CS149 #13 p.34	3.0084
4 位	CS149 #05 p.11	0.4271	CS149 #05 p.35	0.4942	CS166 #04 p.193	2.9652
5 位	CS149 #05 p.37	0.3970	CS149 #05 p.11	0.4625	CS166 #05 p.16	2.9261

p.193 を図 48 に示す．このスライドは，*Two-Stack Queue* の計算量について述べているスライドである．一方，CS166 の#4 の p.33 は図 49 である．CS166 の#4 の p.13 や p.33 は，図 41 と同様連続したスライドの一部である．類似スライドを削除する前は通常の C-TF-IDF はこれらのスライドがランキングの上位を占めていた．正規化した場合，CS149 が上位に上がってきたことから，正規化は有効に作用していることを示している．その上で CS166 の#4 のの p.13 や p.33 が第 1 位になった理由は，CS166 は科目全体を通して対象となるスライド数が他の科目よりも多いためであると考える．科目間のスライド数についても正規化を行う必要があると考えられる．次に **queue** について被験者による実験で求めたランキングの作成に関するコメントを表 22 に示す．学習中と仮定した図 32 では，**queue** の概念的なイメージを示したため，より具体的な実装を行う面を重視したコメントが多い．また，Q-measure の値が最も高くなった，重み付けした C-TF-IDF は，被験者 H が望む「先入れ先出しの動き」という **queue** の動作に関わる部分を示すことに成功しているといえる．一方被験者 E に対しては通常の C-TF-IDF のほうが有効であった．おそらく被験者 E は，表 22 のコメントから，**queue** 自体がどういったものか知りたいと考え，そうしたスライドを上位に選んだと考えられる．これらから，重み付けした C-TF-IDF は動作について細かく説明したものを取得したい場合に有効で，通常の C-TF-IDF は単語自体の説明しているスライドを取得したい場合に有効である．

システムが算出した結果に基づく **pointer** のランキングの結果を表 23 に示す．被験者

表 20: 被験者ごとの **queue** の Q-measure の値

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.2105	0.180	0.3345
B	0.3160	0.1877	0.3752
C	0.040	0.2527	0.6476
D	0.0422	0.020	0.4989
E	0.5527	0.3878	0.0
F	0.3264	0.2866	0.2488
G	0.4197	0.4745	0.0422
H	0.2778	0.3631	0.7336
I	0.3438	0.4297	0.5157
J	0.2738	0.3675	0.6232
平均	0.2821	0.2818	0.3638

表 21: **queue** で被験者が作成した特徴的なランキング

	被験者 D	被験者 E	被験者 H
1 位	CS149 #13 p.34	CS106L #04 p.5	CS166 #04 p.193
2 位	CS166 #05 p.16	CS149 #05 p.11	CS166 #04 p.103
3 位	CS166 #04 p.193	CS149 #05 p.18	CS166 #04 p.33
4 位	CS166 #04 p.103	CS149 #05 p.37	CS166 #04 p.13
5 位	CS149 #05 p.11	CS149 #05 p.35	CS149 #05 p.37

表 22: **queue** で得られたコメントの一部

被験者	コメント
D	関連性がありそうなものとスライド 1 枚で内容が分かりやすそうなものを優先して選んだから
E	なんの説明をするかがほしいと思ったから
H	先入れ先出しの動きが知りたいから

Where We're Going

- The ***amortized*** cost of an enqueue or dequeue into a two-stack queue is O(1).
- Any sequence of n operations on a two-stack queue will take time

  $$n \cdot \mathrm{O}(1) = \mathrm{O}(n).$$

- However, each individual operation may take more than O(1) time to complete.

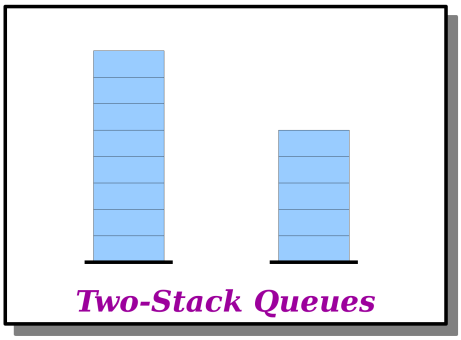


図 48: CS166[7] の#4 の p.193

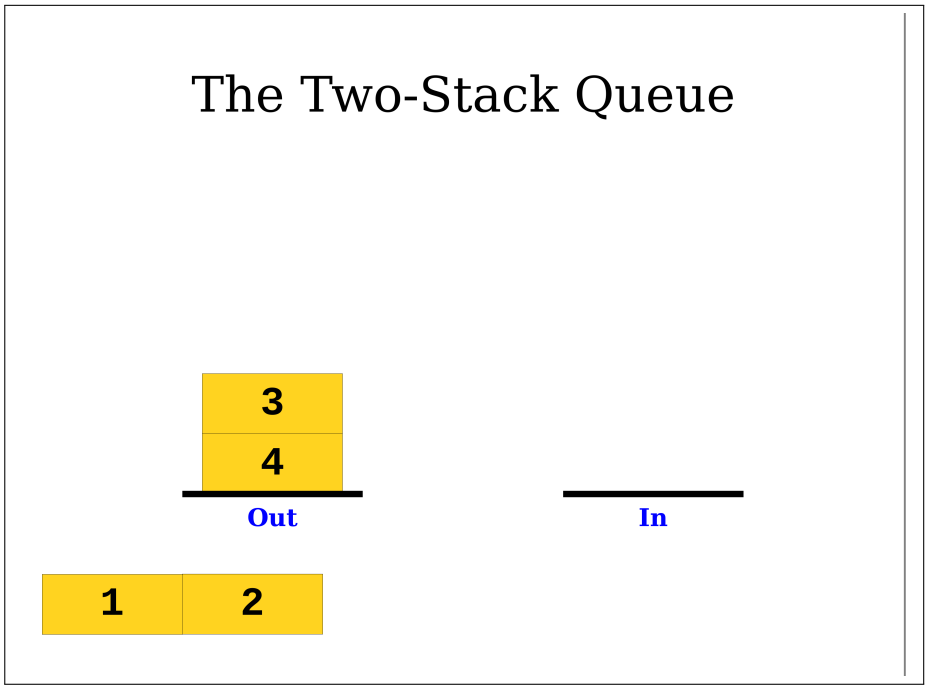


図 49: CS166[7] の#4 の p.33

が作成したランキングに基づく Q-measure の値と平均を表 24 に示す．被験者から得られたコメントの一部を表 25 に示す．通常の C-TF-IDF と正規化した C-TF-IDF で第 1 位となった，CS106L の#5 の p.2 を図 50 に示す．このスライドはその回の目次である．この回の具体的な内容は C-TF-IDF の第 3 位や第 4 位のスライドに繋がっている．また，重み付きの C-TF-IDF で第 1 位になった CS107 の#5 の p.20 を図 51 に示す．このスライドは，C 言語のポインタがどういったものなのか説明している．CS107 の#9 の p.98 を図 52 に示す．このスライドは実際は単体では意味がわからないものとなっている．この回ではバブルソートを取り上げ，関数ポインタや型に依存しない実装方法について取り上げている．表 25 のコメントから，**pointer** 自体の実装や解説が選ばれたことがわかる．特に図 51 を上位に選んでいる被験者が多くいた．そのため通常の C-TF-IDF や正規化した C-TF-IDF と比較して，Q-measure の値が高く，重み付けした C-TF-IDF が優位であるといえる．図 50 で示されている目次のスライドは，タイトルのみのスライドや次回の内容の予告と同様，あまり情報量がないスライドであると考えている．こういったスライドで現れるスライドは取り除いて検証することで，目次内で触れている具体的なスライドを取り扱うことができる．これは今回の実験では図 33 の前後を示しておらず，学生がどういった内容を把握しているのかわからなかったことが原因であると考えられる．また，図 52 については重み付けが適切に行われたため，重み付けした C-TF-IDF のランキングの上位に入ったと考えられる．現在の実装では，タイトルとして扱われたのはベン図内の *Function Pointers* と *Generics* であるが，タイトルと相違ない．加えて，箇条書きの重みも *Pointers* はこのスライドで最も高い値となる．こういったスライドはすでに学習済みの内容として学生に示された場合，図であることから学生の記憶に残っており記憶を想起できる可能性はある．しかし，まだこの科目を学習していない学生がこのスライドを見た場合，関数ポインタという名称のみを理解する可能性がある．こうしたスライドを上位に入らないようにする場合，図として重み付けを行うか，スライドの構造に基づいて C-TF-IDF のスコアを下げる方法が考えられる．前者は，スライドに図を含むか否か，あるいは図の座標から形を判断することになると考えられる．この場合，他の図を含むスライドにも影響を与える可能性がある．後者は，筆者が典型的なスライドの構造を用意したり，他のスライドのテキストの座標情報を使うことになる．型に沿ったスライドのみとなってしまうため，すでに示しているいくつかのスライドは対象から外れてしまうだろう．学生に対して有効と判断できるスライドもあるため，どういったスライドを除くべきか，スライドの構造について検討していく必要がある．別の手法として図 53 のようなインタフェースを考える．図 53 の右下は，学習中と仮定した図 33 である．このスライドからシステムによって得られた図 52 は右上である．右上から，任意の単語を選ぶことで，その単語をさらに詳細に説明しているスライドに飛ぶことができるというものである．こういったインタフェースを用いるこ

表 23: **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS106L #05 p.2	1.7987	CS106L #05 p.2	0.5995	CS107 #05 p.20	11.8060
2 位	CS166 #06 p.435	0.4397	CS106B #17 p.32	0.4828	CS107 #05 p.25	7.5299
3 位	CS106L #05 p.64	0.4331	CS166 #06 p.424	0.4372	CS143 #17 p.28	4.5536
4 位	CS106L #05 p.77	0.4136	CS106L #05 p.77	0.4306	CS107 #05 p.26	3.7230
5 位	CS143 #12 p.19	0.4035	CS166 #06 p.435	0.4302	CS107 #09 p.98	2.9638

とで，そのスライド単体では内容に乏しい場合であっても，関連した他の内容に発展させることができる．

hash に関してシステムが算出した結果に基づくランキングを表 26 に示す．被験者の作成したランキングに基づく Q-measure の値と平均を表 27 に示す．**hash** に関して得られた被験者からのコメントの一部を表 28 に示す．なお被験者 C は **hash** では有効なコメントを得られなかったため，掲載していない．図 54 に，CS106B の#27 の p.60 を示す．このスライドは，ハッシュの計算結果をそのままハッシュを格納するインデックスにする場合の問題点を出している．このようなスライドはシンプルなハッシュを学び始めた直後の学生にとって，有効なスライドである．また，ある程度ハッシュに関して理解をしたあとの学生にとっては問題点を再度示すという意味で価値があると考える．本研究では，こうしたスライドは想定していない重み付けを行っていたため，重み付けした C-TF-IDF ではこのスライドは上位にはならない．提示するスライドのうち，1 つはこうした問題提起があるスライドにすることで，学生に対して学習の中に考察する要素を加えられるのではないかと考えられる．評価元とした図 35 が HashMap に関した内容であったため，被験者が選ぶコメントについても HashMap のものが多い．学習中と仮定したスライドに対して，重み付けした C-TF-IDF が優れていることがわかる．一方，図 55 は CS166 の#7 の p.13 である．図 55 では，ハッシュの実装について説明している．HashMap の使い方について学習した上で，**hash** の内部の実装を示すことで，学生は同時に 2 つの側面を知ることになる．

表 24: 被験者ごとの **pointer** の Q-measure の値

	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.1322	0.0633	0.8318
B	0.1033	0.1052	0.7336
C	0.5510	0.4674	0.040
D	0.1455	0.0422	0.8318
E	0.5579	0.5579	0.2866
F	0.0	0.3157	0.4393
G	0.0	0.2267	0.5227
H	0.2105	0.4064	0.3357
I	0.1052	0.1578	0.2105
J	0.0688	0.1455	0.2011
平均	0.1874	0.2488	0.4433

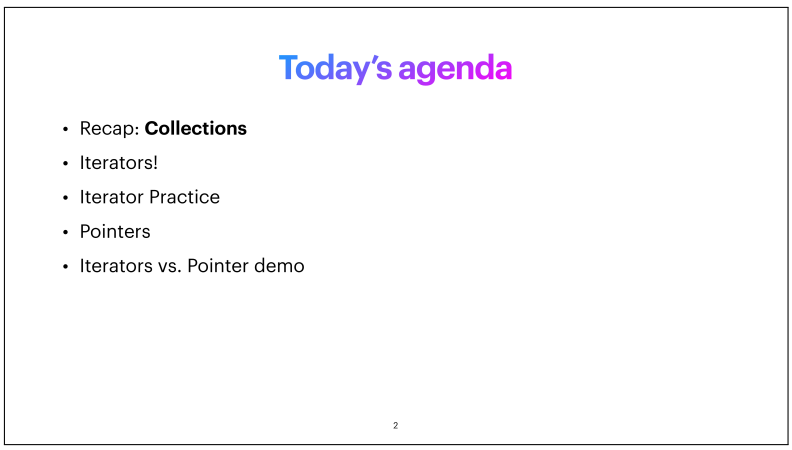


図 50: CS106L[29] の#5 の p.2

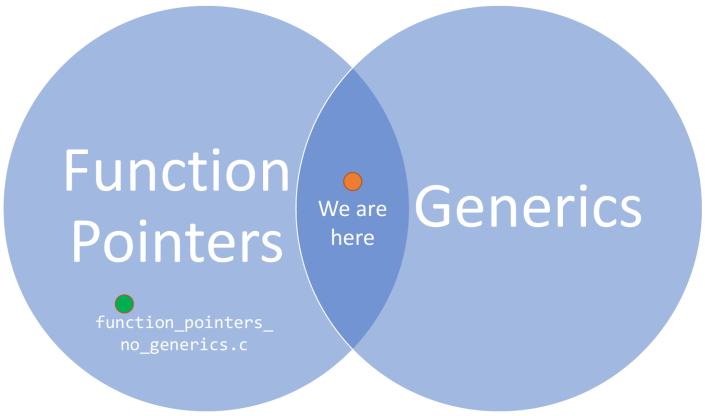
Pointers

- A *pointer* is a variable that stores a memory address.
- Because there is no pass-by-reference in C like in C++, pointers let us pass around the address of one instance of memory, instead of making many copies.
- One (8 byte) pointer can refer to any size memory location!
- Pointers are also essential for allocating memory on the heap, which we will cover later.
- Pointers also let us refer to memory generically, which we will cover later.

20

図 51: CS107[27] の#5 の p.20

Function Pointers and Generics



98

図 52: CS107[27] の#9 の p.98

表 25: **pointer** で得られたコメントの一部

被験者	コメント
A	配列とポインタについて学んでおり，ポインタについての復習ができるため．また，ポインタに似ているものや他の例を閲覧することで，覚え間違いの確認や，今後の応用例について学べるため．
B	表示されるポインタの値について載っているから
D	ポインタについての解説やそれに近そうなものを選んだから．

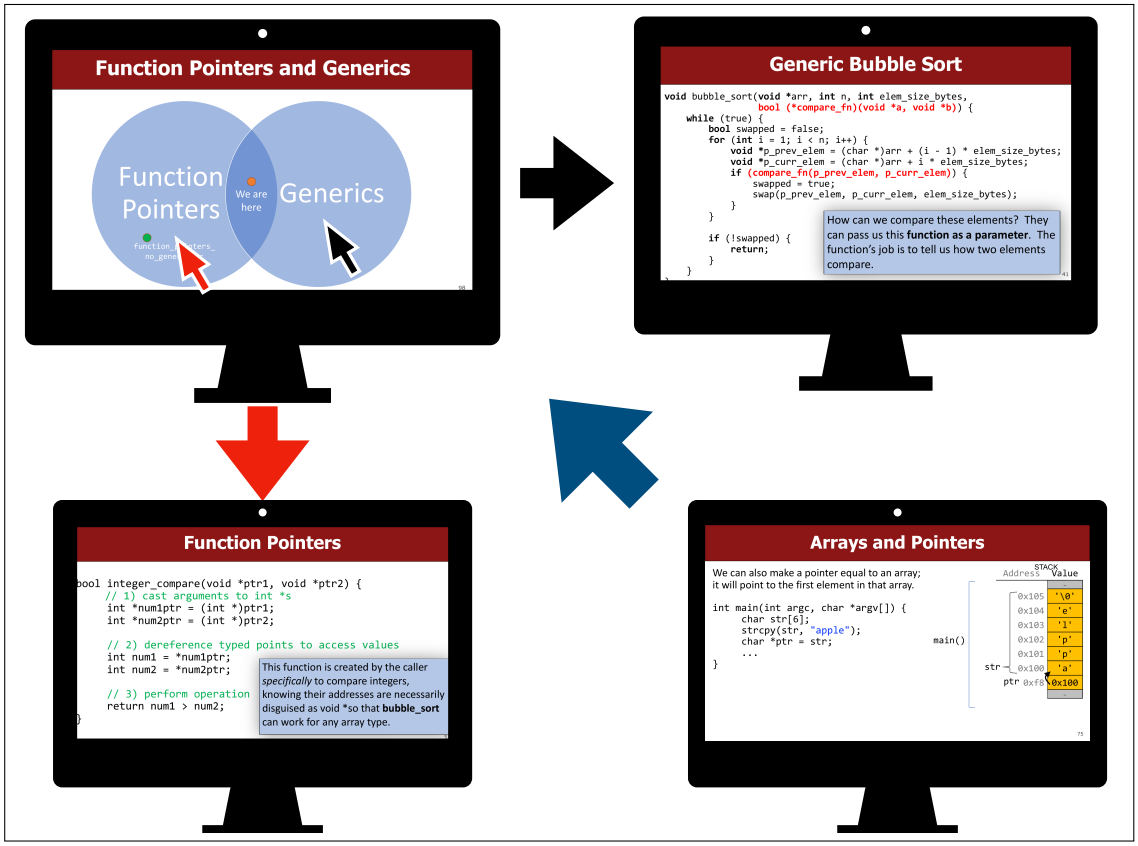


図 53: あるスライドから別のスライドへ移るインタフェースの例 (CS107[27] より引用)

表 26: **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS161 #07 p.121	1.6127	CS106B #27 p.2	0.8763	CS166 #13 p.18	10.5168
2 位	CS106B #27 p.2	0.9924	CS161 #07 p.121	0.7855	CS166 #08 p.7	8.7048
3 位	CS106B #27 p.60	0.7793	CS166 #07 p.4	0.6898	CS166 #07 p.13	8.3551
4 位	CS161 #08 p.3	0.7707	CS161 #08 p.3	0.6822	CS106B #27 p.112	6.4578
5 位	CS166 #07 p.4	0.7076	CS166 #07 p.6	0.6282	CS161 #08 p.93	5.5120

これは，学生は **hash** について理解する手助けになると考えている．Q-measure のスコアは重み付けした C-TF-IDF が最も高い．そのため，HashMap の使い方と同時に **hash** の内部について扱うことは有効であるといえる．

5.6 全体の考察

今回，結果の中に含められていないが，被験者実験の中で CS205L や CS243 についてはランキングに入ることはなかった．これらの科目では，今回対象とした単語があまり出現しなかった．単語が基礎科目の内容に寄っていたことが考えられるため，今後別の単語でも調査が必要である．

今回実装したシステムは，全科目の履修順序を考慮した C-TF-IDF の算出は行っていない．全ての科目を順番に読み込んだ場合，今回と大きく結果が異なると考えられる．しかし，履修順序までは十分に考慮できなかったため，今回は見送っている．その結果，学習済みの科目と今後学習する科目とに分けた実装や実験を行うことができていない．一方で，今回の実験から得られた結果は，重み付けした場合，筆者として将来出てきて欲しいスライドとなっているものが多い．また，学習済みの場合に有効と考えられるスライドは正規化した C-TF-IDF に多かった．これらの結果が筆者に対してのみ有効かどうか今後検証していく．

表 27: **hash** の被験者ごとの Q-measure の値

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.0	0.050	0.7114
B	0.03	0.1233	0.8469
C	0.0688	0.0	0.6942
D	0.020	0.0988	0.8136
E	0.0688	0.050	0.6750
F	0.020	0.0688	0.8136
G	0.1578	0.2578	0.6417
H	0.3157	0.5157	0.5145
I	0.1722	0.0	0.6275
J	0.0	0.020	0.8984
平均	0.085	0.1184	0.7237

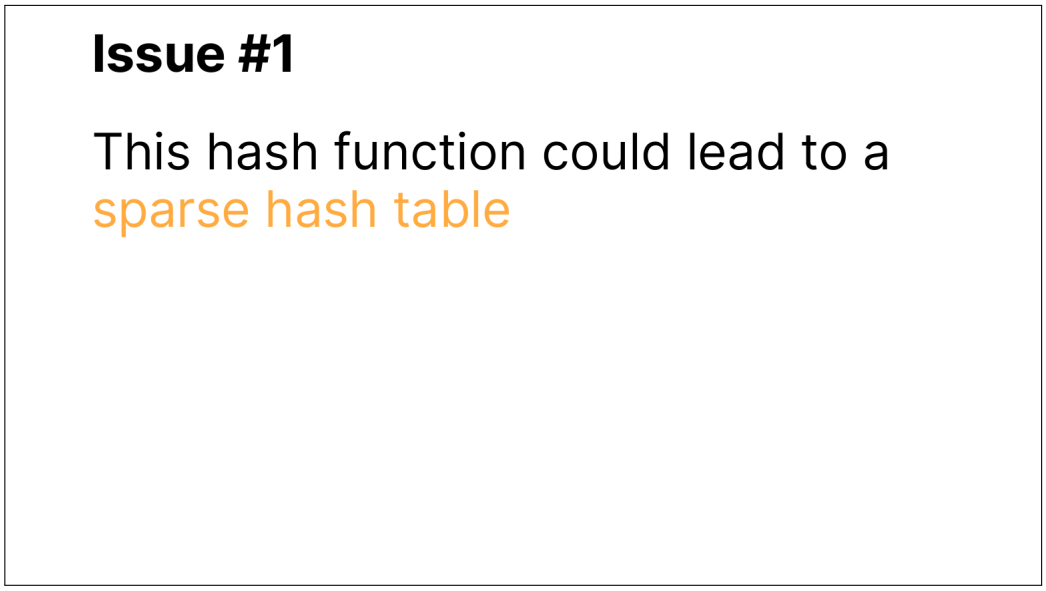


図 54: CS106B[5] の#27 の p.60

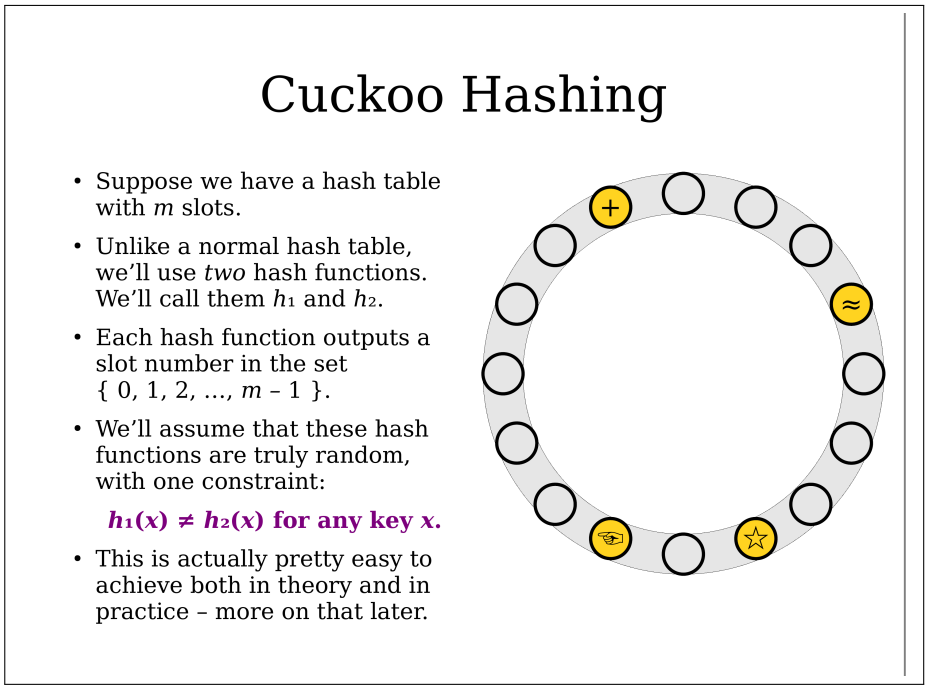


図 55: CS166[7] の#7 の p.13

表 28: **hash** で得られたコメントの一部

被験者	コメント
A	ハッシュマップを学んでいるときにハッシュについての資料を閲覧することで, ハッシュについての復習ができるため.
G	よくわからないものから決めていきました.

本研究では，単語の特徴を用いてスライドを提示しているため，学生が見ているスライドの特徴や学生個人の学習活動については考慮していない．これにより，単語に直接紐づいた内容を示すことができている．そのため，学生がある単語について説明しているスライドが欲しい場合，本手法は有効であるといえる．これは，桐原ら [21] が提案している質問に基づいて関連するスライドを示す手法で得られる結果と類似する．本研究やこの研究から学生に提案するスライドは，学生に対して学習の中で単語に対する疑問点の解消に役立つと考える．また，スライドの特徴を重視する場合，共起行列やコサイン類似度を用いてスライドの特徴同士を比較することが考えられる．すでに C-TF-IDF を用いて，スライド内の単語に対してスコアをつけることはできていることから今後先述した手法を検討していく．学生個人の学習活動については，Sacholapunt[15] や椎野ら [16] が対象としているように学生が履修している科目で出題された演習問題の内容から関連する情報を提示することが考えらえる．また，筆者が卒業研究で行ったような学生の書き込みを用いることが考えられる [17]．学生の書き込みは，その箇所を自身が重要だと思った箇所であるといえる．そのため，その箇所と関連するスライドを示すことで，その学生にとって重要な箇所に追加の情報を付与できる．さらに，書き込みを行っていない箇所からスライドを示すことで，学生が見落としていた内容を発見することにつながる．ほかにも，本研究では Mima[14] が扱ったようなオントロジーについては，考慮していなかい．本研究では同音異義語については同じものとして扱い，表記ブレがある単語が別のものとして扱った．同音異義語は別の側面を知るために有効であると考えている．しかし表記ブレに関しては，本来関連しなくてはいけないスライドやある単語の別の側面を知る機会を消失させてしまっている．例えば，**stack** や **queue** は，**Fist In First Out(FIFO)** や **Fist In Last Out(FILO)** と呼ばれることがある．分野や扱っている内容によって表記は異なるが，近しい意味を持つ単語はあると考えられる．こういった単語を発見するために有効な手法を検討していく必要がある．

本研究で対象とした単語から得られた結果は，学生の疑問解消や問題提起を行える可能性を示した．一方，モチベーションを維持したり，将来の学習する内容を発見できているとは言い難い．一つの原因として今回実験の対象とした科目には応用科目が少なかった．こうなってしまったのは，スライド形式の資料を入手できなかったためである．応用科目になると，レジュメや Web ページで説明を行うことが増えていた．現在の手法ではこれらに対応できなかったため，応用科目自体が少なくなってしまった．今後，レジュメや Web ページなどから時系列や特徴の抽出を実施する方法を検討する必要がある．

被験者による実験では，被験者のランキング結果に基づく上位 5 位の Q-measure を求めた．それぞれの結果の平均を見ると，重み付けした C-TF-IDF が高い傾向にあったため，提案手法で導入した重み付けは有効に作用したと考えられる．加えて，Q-measure の値も

重み付けした C-TF-IDF が高いことから，学生が求めるスライドを重み付けによって発見することができているといえる．しかしながら，被験者による実験はこちらが学習中と仮定したスライドを示し，関連するスライドについてランキングの作成を求めていた．被験者らは学習中と仮定したスライドの前後のスライドに関する情報が不明瞭なまま実験を行う結果となった．そのため，前後のスライドの有無によって，学習中と仮定したスライドの内容の理解に大きく差が生じていることが考えられる．特に学習をしている中で，いきなり特定のスライドのみを開くということは考えにくい．少なくとも指定したスライドの直前のスライドまでは被験者に示した場合，ランキングの結果にどこまで差が生じるのか検証が必要である．

図 56 は，被験者による実験の **queue** の中で通常の C-TF-IDF と正規化した C-TF-IDF の第 1 位である，CS166 の#14 の p.13 である．このスライドから *Two-Stack Queue* について説明が始まり，図 57 のような形で終わる．パラパラ漫画で進む形式で，アルゴリズムを動作を示す際によく用いられる手法である．図 56 は被験者による実験の際に提示したスライドに含んでいた．正規化を行い，文字数に応じた調整を行ったが，こうしたスライドを十分に除去できていないことが明らかになった．**hash** においても図 58 が 1 位となっている．図 56 については，**stack** の考察の際に述べたように，コサイン類似度や共起行列を用いてスライド同士の類似度を判断し，パラパラ漫画をショートアニメーションで表示したり，スライドの最初のみを C-TF-IDF の計算の対象とすることで改善できるといえる．この場合，パラパラ漫画の途中で動きの説明を文章で行うスライドを考慮することになる．一方，図 58 は避けようがない問題である．現在対象にした単語については，重み付けを行うことでこうしたスライドは出現しないが，すべての単語で同様の結果を得られるとは考えにくい．他の自然言語処理の手法を用いることで回避できるかどうか検証していく必要がある．

タイトルのみのスライドについて本研究では C-TF-IDF を計算する前に除外していた．これは，C-TF-IDF や正規化を行った時点でタイトルのみのスライドが上位になることが多かったためである．タイトルのみのスライドも C-TF-IDF の計算に用いる場合，C-TF-IDF の計算や重み付けに影響を与えると考えている．タイトルのみのスライドで一度単語が出現した後に対象となるスライドが続くため，期間が空いた後に多く値が上昇する C-TF-IDF の特性が抑制される．タイトルのみのスライドの有無が他のスライドに与える影響については検証する必要がある．

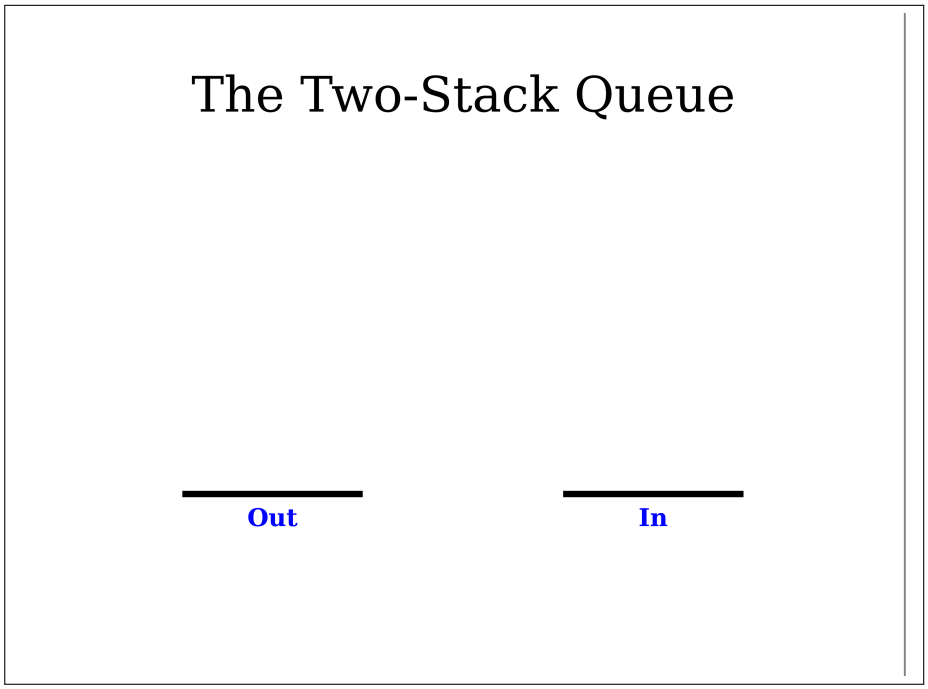


図 56: CS166[7] の#4 の p.13

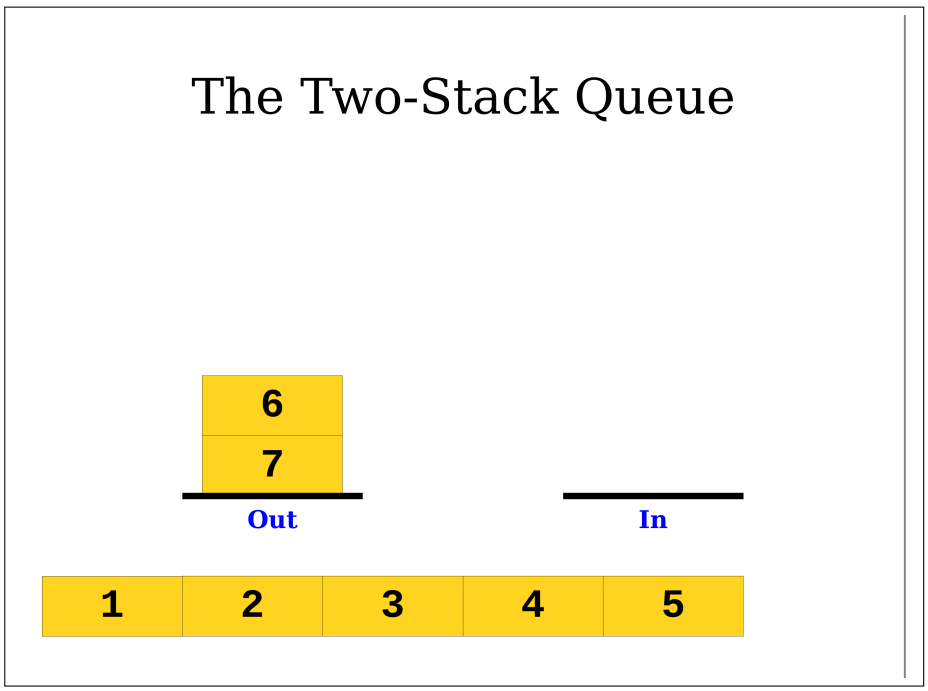


図 57: CS166[7] の#4 の p.51

図 58: CS161[30] の#7 の p.121

6 結論

6.1 結論

本研究では，自習中の学生のために，学習済みのスライドや今後学習するスライドを発見するためのスコアリング手法を提案した．スコアリングのために，TF-IDF を採用した．さらに，授業資料の特徴を生かすために，C-TF-IDF と重み付け，正規化を導入し，スコアの補正を行った．Python によるシステムの実装を行い，Stanford University の科目の中で Web サイトから入手可能な資料を使って 2 種類の実験を行った．両方に実験で C-TF-IDF，正規化を行った C-TF-IDF，正規化と重み付けした C-TF-IDF の 3 つの結果を比較した．1 つは筆者が対象とした科目それぞれに第 3 位までのランキングを作成し，システムが作成したランキングに対して Q-measure の値を比較した．その結果，正規化と重み付けした C-TF-IDF は C-TF-IDF よりも高いスコアを示すが，有効ではないスライドもあることが明らかになった．もう一つは，システムが算出しすべてを合算した結果から 3 つの結果の上位 5 つのスライドを対象に，被験者に対してランキングを作成してもらった．上位 5 位以内に入るスライドについて，Q-measure を求め，比較を行った．実験から，重み付け C-TF-IDF の Q-measure の値は重み付けしないものと比べて高い傾向となった．これらのことから，重み付けした C-TF-IDF は他の 2 つと比べて筆者と被験者に対して学習する上で価値のあるスライドを示すことができることが明らかになった．

6.2 今後の課題

本研究では，すべての科目を結合した状態で C-TF-IDF の計算をせず，科目ごとに計算を行っていた．実験でした使用した科目は，Stanford UniversityCS コースの Web ページ [33] の四半期ごとの科目リストから開講時期を参照可能である．加えて，これらの科目はナンバリングされているため，ある程度年度ごとに履修順序のコントロールも可能である．しかし，実際に学生が履修する順序を考慮しきれなかったため，本研究では導入を見送った．実際に学習する順序を考慮するならば，Web ページから得られる情報に基づき仮想的に履修スケジュールを作成するか，実際に筆者が履修した科目を用いた計算が必要である．この問題があったために，学習済みの科目と今後学習する科目とに分けて結果を算出するには至らなかった．今後，C-TF-IDF を計算する段階で，科目をすべて合わせた状態で算出し，その結果について考察していく．

本研究では学生への提示部分については検討していない．実験も筆者があらかじめ用意したスライドを対象としてたため，実際の自習中に本研究が有効であるかの判断は不十分である．また，提示を行う場合の適切なスライド数についても検討できていないため，今

後検討していく必要がある．そういった点を踏まえて，自習中の学生に適したユーザインタフェースを作成していく．

7　本研究の新規性・貢献

学生は本研究のシステムを活用することで，自分の目標を見据えた大学 4 年間の履修計画の作成につなげることができる．これは基礎科目を履修している段階で，将来学習する応用科目の断片を知ることで，自分の興味分野で必要な知識の整理や苦手な科目の克服が可能であると考えるためである．また，従来の学習では，整理に時間を要していた科目同士の繋がりを発見するヒントを与えることで，複数の科目を連結して学生個人の学習活動全体の改善に寄与する．

また，カリキュラム作成者やシラバス作成者である教員がこのシステムを活用することで，科目名やその科目のシラバスだけでは把握しきれない実際の授業内容の把握を支援できる．それにより，前提となる科目や発展となる科目を教員が把握できる．これは，従来前提として考えていた科目で説明が不足している内容を補ったり，逆に重複していた説明の時間を排除し新しい説明を追加することにつながる．そのため，学生にとって充実したカリキュラムの作成が可能になる．

参考文献

[1] 明星大学. 履修の手引 2019 年度. 2019.

[2] 山本雄介, 峰松翼, 長沼祥太郎, 谷口雄太, 大久保文哉, 島田敬士. 科目の関連性情報を付加したカリキュラム情報閲覧システムの開発. 情報処理学会研究報告教育学習支援情報システム研究会 (CLE), Vol. 2020-CLE-35, No. 9, pp. 1–8, 11 月 2021.

[3] 坂本祥之, 清水敏之, 吉川正俊. 講義プレゼンテーションスライド部分対応付けを用いた学習支援. 情報処理学会 第 77 回全国大会講演論文集, 第 2015 巻, pp. 615–616, 5 月 2015.

[4] Mohammadreza Tavakoli, Mirette Elias, Gábor Kismihók, and Sören Auer. Metadata analysis of open educational resources. LAK21, p. 626–631, New York, NY, USA, 2021. Association for Computing Machinery.

[5] Cynthia Baily Lee, Julie Zelensk, and Neel Sunil Kishnani. CS106B Programming Abstractions. `https://web.stanford.edu/class/archive/cs/cs106b/cs106b.1222/`, (Retrieved Dec. 22 2021).

[6] Jerry Cain. CS110 Principles of Computer Systems. `https://web.stanford.edu/class/archive/cs/cs110/cs110.1222/`, (Retrieved Dec. 22 2021).

[7] Keith Schwarz. CS166 Data Structures. `http://web.stanford.edu/class/cs166/`, (Retrieved Dec. 22 2021).

[8] Fredrik Kjolstad and Alex Aiken. CS143 Compilers. `https://web.stanford.edu/class/cs143/`, (Retrieved Dec. 22 2021).

[9] Chris Manning. CS224N Natural Language Processing with Deep Learning. `http://web.stanford.edu/class/cs224n/`, (Retrieved Dec. 22 2021).

[10] Cristobal Romero and Sebastian Ventura. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 10, No. 3, p. e1355, 2020.

[11] Cristóbal Romero and Sebastián Ventura. Educational data mining: A review of the state of the art. Vol. 40, pp. 601–618, Nov 2010.

[12] Alec Goncharow, Matthew Mcquaigue, Erik Saule, Kalpathi Subramanian, Jamie Payton, and Paula Goolkasian. Mapping materials to curriculum standards for design, alignment, audit, and search. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, SIGCSE '21, pp. 295–301, New York, NY, USA, 2021.

[13] Association for Computing Machinery (ACM) Joint Task Force on Computing Curricula and IEEE Computer Society. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Association for Computing Machinery, New York, NY, USA, 2013.

[14] Hideki Mima. Mima search: A structuring knowledge system towards innovation for engineering education. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, COLING-ACL '06, p. 21–24, USA, 2006. Association for Computational Linguistics.

[15] Petch Sajjacholapunt and Mike Joy. Analysing features of lecture slides and past exam paper materials. In *Proceedings of the 7th International Conference on Computer Supported Education-Volume 1*, pp. 169–176, 2015.

[16] 椎野徹也, 島田敬士, 峰松翼, 谷口倫一郎. 学習活動データに基づく個人適応型復習教材推薦システムの開発. 情報処理学会研究報告教育学習支援情報システム研究会 (CLE), Vol. 2020-CLE-32, No. 4, pp. 1–6, 3 月 2020.

[17] 印部太智, 丸山一貴. 注目箇所の把握を目的とした授業資料への書き込みの可視化. 情報処理学会 インタラクション 2020, pp. 1–5, 2020.

[18] 梅澤克之, 中澤真, 小林学, 石井雄隆, 中野美知子, 平澤茂一. 言語学習を対象とした自学自習システムの研究〜システム開発に関する研究成果〜. 情報教育シンポジウム論文集, 第 2021 巻, pp. 93–99, 8 月 2021.

[19] Mitsuhiro Goto and Akihiro Kashihara. Understanding presentation document with visualization of connections between presentation slides. *Procedia Computer Science*, Vol. 96, pp. 1285–1293, 2016.

[20] 羽山徹彩, 國藤進. プレゼンテーションスライド情報検索のためのスライドページからの要求関連情報抽出. 情報処理学会研究報告デジタルドキュメント研究会 (DD), Vol. 2010-DD-76, No. 2, pp. 1–7, 7 月 2010.

[21] 桐原牧紀, 王元元, 河合由起子, 角谷和俊. e-learning における講義コンテンツの階層構造に基づくスライド推薦方式の提案. 第 13 回データ工学と情報マネジメントに関するフォーラム (DEIM Forum 2021), pp. 1–6, 2021.

[22] Shosuke Sato, Haruo Hayashi, Norio Maki, and Munenari Inoguchi. Development of automatic keyword extraction system from digitally accumulated newspaper articles on disasters. *2nd International Conference on Urban Disaster Reduction Proceedings*, pp. 1–6, Nov., 2007.

[23] Jiaul H. Paik. A novel tf-idf weighting scheme for effective ranking. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, p. 343–352, New York, NY, USA, 2013.

[24] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, p. 232–241, Berlin, Heidelberg, 1994. Springer-Verlag.

[25] Taichi Imbe and Kazutaka Maruyama. *Finding the Noteworthy Learning Material about a Keyword through Courses Already Learned and to Be Learned*, p. 1268. Association for Computing Machinery, New York, NY, USA, 2021.

[26] Adobe Systems. *Document management – Portable document format – Part 1: PDF 1.7*. 2008.

[27] Nick Troccoll. CS107 Computer Organization & Systems. `https://web.stanford.edu/class/archive/cs/cs107/cs107.1216/`, (Retrieved Dec. 22 2021).

[28] Kayvon Fatahallan and Kunle Olukotun. CS149 Parallel Computing. `http://35.227.169.186/cs149/fall19/`, (Retrieved Dec. 22 2021).

[29] Frankie Cerkvenik and Sathya Edamadaka. CS106L Standard C++ Programming. `http://web.stanford.edu/class/cs106l/index.html`, (Retrieved Dec. 22 2021).

[30] Karey Shi. CS161 Design and Analysis of Algorithms. `https://web.stanford.edu/class/cs161/`, (Retrieved Dec. 22 2021).

[31] Ron Fedkiw. CS205L Continuous Mathematical Methods with an Emphasis on Machine Learning. `http://web.stanford.edu/class/cs205l/`, (Retrieved Dec. 22 2021).

[32] Monica S. Lam. CS243 Program Analysis and Optimization. https://suif.stanford.edu/~courses/cs243/, (Retrieved Dec. 22 2021).

[33] Stanford Univeristy. Stanford Engineering Computer Science Courses. https://cs.stanford.edu/academicz/courses, (Retrieved Jan. 18 2022).

[34] Tetsuya Sakai. New performance metrics based on multigrade relevance: Their application to question answering. In Noriko Kando and Haruko Ishikawa, editors, *Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization, NTCIR-4, National Center of Sciences, Tokyo, Japan, June 2-4, 2004*. National Institute of Informatics (NII), 2004.

[35] 酒井哲也. よりよい検索システム実現のために：正解の良し悪しを考慮した情報検索評価の動向. 情報処理, Vol. 47, No. 2, pp. 147–158, 2 月 2006.

謝辞

明星大学，丸山一貴先生には，本研究に関して多くのアイディアと指導をいただけましたこと，資料の使用を快諾していただいたことを感謝します．明星大学，長慎也先生，山中脩也先生には，本研究に関する貴重な知見をいただけたこと感謝いたします．明星大学，横山真男先生ならびに 2020 年度 2021 年度横山研究室の皆様，国士舘大学，中村嘉志先生ならびに 2020 年度 2021 年度中村研究室の皆様，東海大学，大西建輔先生ならびに 2020 年度 2021 年度大西研究室の皆様におかれましては，本研究について多くの議論を行えたことに感謝いたします．The 52nd ACM Technical Symposium on Computer Science Education(SIGCSE2021) のポスターセッションにて, 議論してくださった皆様にはここで感謝の意を示します．The 53nd ACM Technical Symposium on Computer Science Education(SIGCSE2022) の Student Research Competition にて，議論してくださった皆様にはここで感謝の意を示します．コンピュータと教育研究会第 164 回研究発表会において，議論してくださった皆様には感謝申し上げます．

明星大学，佐藤浩志先生，末田欣子先生，丹治昭夫先生，日本語の授業資料を快諾していただけたこと感謝いたします．明星大学，中村哲夫先生におかれましては資料をお借りしました．感謝いたします．Stanford University にて，CS110 を担当されております Jerry Cain 先生，CS166 を担当されております Keith Schwarz 先生，CS143 を担当されております Fredrik Breg Kjolstad 先生，CS224N を担当されております Christopher Manning 先生，CS106L を担当されております Sathya Edamadaka 先生と rankie Cerkvenik 先生，CS149 を担当されております Kunle Olukotun 先生，CS107 を担当されております Nick Troccoli 先生におかれましては学会発表の際に科目内のスライドの資料の使用を快諾していただけましたこと，この場をお借りして感謝いたします．また，Stanford University にて CS106B を担当されております Cynthia Baily Lee 先生，CS161 を担当されております Shi kareyshi 先生，CS205L を担当されております Ron Fedkiw 先生 CS243 を担当されております Monica S. Lam 先生には科目の資料をお借りしました．感謝いたします．

発表一覧

- 印部太智, 丸山一貴, 教員の授業進行補助を目的とした学習者の書き込み活動の可視化, 情報処理学会研究報告コンピュータと教育 (CE),2020-CE-156,4,pp.1-8,2020 (学生奨励賞)
- Taichi Imbe. and Kazutaka Maruyama., Finding the Noteworthy Learning Material about a Keyword through Courses already Learned and to be Learned., In Proceedings of the 52nd ACM Technical Symposium on Computer Science Education (SIGCSE '21), p.1268, 2021.
- Taichi Imbe, Finding the Most Relevant Pages of the Learning Materials on which a Student Just Focuses, In Proceedings of the 53rd ACM Technical Symposium on Computer Science Education (SIGCSE '22), (Accepted)
- 印部太智, 丸山一貴, 自習中の学生のための科目を超えた授業資料の提示, 情報処理学会研究報告コンピュータと教育 (CE),2022-CE-164,4,pp.1-8,2022

表 29: CS106B における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#27 p.6	0.5166	#21 p.21	0.5800	#18 p.21	2.6018	#08 p.33
2 位	#18 p.21	0.3693	#27 p.6	0.3964	#27 p.6	2.1749	#18 p.19
3 位	#27 p.11	0.3360	#16 p.8	0.2899	#27 p.11	1.4393	#16 p.25

表 30: 表 29 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

A　筆者による実験結果

A.1　CS106B

表 31: CS106B について **heap** でのランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#18 p.2	0.3500	#07 p.17	0.4290	#07 p.17	2.8843	#18 p.9
2 位	#07 p.17	0.2759	#18 p.2	0.2761	#18 p.2	2.0414	#07 p.15
3 位	#19 p.3	0.2643	#18 p.14	0.2041	#18 p.14	1.5828	#17 p.21

表 32: 表 31 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 33: CS106B について **stack** でのランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#16 p.4	0.2388	#16 p.2	0.2664	#16 p.4	1.0952	#16 p.4
2 位	#04 p.2	0.2174	#04 p.2	0.1723	#16 p.21	0.8287	# 16 p.15
3 位	#16 p.6	0.1408	#05 p.17	0.1711	#16 p.10	0.7572	#12 p.22

表 34: 表 33 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.6719
重み付けした C-TF-IDF	0.6719

表 35: CS106B における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#07 p.14	0.3140	#07 p.17	0.4518	#07 p.17	2.2186	#16 p.24
2 位	#08 p.17	0.2582	#07 p.14	0.4044	#07 p.14	1.2726	#07 p.15
3 位	#17 p.2	0.2500	#07 p.15	0.2931	#07 p.15	1.2139	#17 p.12

表 36: 表 31 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1111
正規化した C-TF-IDF	0.1111
重み付けした C-TF-IDF	0.1111

表 37: CS106B における **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#19 p.3	0.3406	#05 p.10	0.3762	#22 p.6	1.3640	#22 p.6
2 位	#19 p.4	0.2878	#22 p.6	0.2889	#19 p.3	1.2775	#25 p.24
3 位	#5 p.9	0.2199	#05 p.9	0.2852	#05 p.10	1.2442	#18 p.9

表 38: 表 37 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3386
正規化した C-TF-IDF	0.3386
重み付けした C-TF-IDF	0.6719

表 39: CS106B における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#17 p.32	0.3824	#17 p.32	0.4828	#17 p.32	2.2172	#16 p.8
2 位	#17 p.2	0.2739	#20 p.2	0.2544	#20 p.2	1.3263	#17 p.19
3 位	#20 p.2	0.2440	#20 p.9	0.2441	#20 p.9	1.1987	#15 p.25

表 40: 表 39 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 41: CS106B における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#27 p.18	0.5674	#27 p.18	0.4524	#27 p.18	2.9563	#23 p.36
2 位	#27 p.19	0.4517	#18 p.11	0.3840	#27 p.19	2.3891	#13 p.5
3 位	#18 p.11	0.4441	#27 p.19	0.3666	#27 p.20	2.3137	#27 p.18

表 42: 表 41 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3359
正規化した C-TF-IDF	0.3359
重み付けした C-TF-IDF	0.3359

表 43: CS106B における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#27 p.2	0.9924	#27 p.2	0.8763	#27 p.2	6.4003	#27 p.30
2 位	#27 p.60	0.7793	#27 p.60	0.5700	#27 p.112	4.8789	#28 p.9
3 位	#27 p.112	0.6500	#27 p.51	0.5445	#27 p.60	4.2850	#27 p.104

表 44: 表 43 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 45: CS106L における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.28	0.3960	#10 p.67	0.5139	#06 p.36	4.7466	#12 p.34
2 位	#10 p.67	0.3881	#10 p.68	0.4592	#06 p.37	3.9922	#06 p.37
3 位	#10 p.68	0.3300	#10 p.66	0.4225	#12 p.40	3.5615	#04 p.38

表 46: 表 45 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.2539

A.2 CS106L

表 47: CS106L における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#15 p.50	0.4417	#15 p.50	0.3586	#15 p.50	1.3151	#12 p.49
2 位	#15 p.25	0.2283	#15 p.25	0.1663	#15 p.25	0.4991	#15 p.49
3 位	#12 p.49	0.0890	#12 p.49	0.0918	#12 p.49	0.2617	#15 p.25

表 48: 表 31 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3174
正規化した C-TF-IDF	0.3174
重み付けした C-TF-IDF	0.3174

表 49: CS106L における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.4	0.1451	#05 p.4	0.1081	#05 p.12	2.0678	#04 p.52
2 位	#05 p.12	0.1028	#05 p.12	0.0795	#10 p.45	0.4381 # 05 p.12	
3 位	#05 p.6	0.1001	#05 p.6	0.0765	#05 p.6	0.2179	#05 p.40

表 50: 表 49Q-measure の値

	Q-measure
通常の C-TF-IDF	0.2539
正規化した C-TF-IDF	0.2539
重み付けした C-TF-IDF	0.5039

表 51: CS106L における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.64	0.4643	#05 p.64	0.2915	#12 p.44	2.2233	#10 p.52
2 位	#14 p.53	0.2149	#14 p.53	0.1550	#05 p.64	0.8746	#12 p.49
3 位	#15 p.43	0.2087	#15 p.43	0.1546	#15 p.43	0.8541	#15 p.45

表 52: 表 51 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 53: CS106L における **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.5	0.4458	#05 p.5	0.3129	#05 p.5	1.1915	#05 p.4
2 位	#04 p.59	0.3197	#04 p.59	0.1356	#10 p.40	0.9273	#04 p.57
3 位	#05 p.4	0.1474	#05 p.41	0.0993	#05 p.41	0.6645	#05 p.40

表 54: 表 53 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 55: CS106L における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.2	0.6493	#05 p.2	0.5995	#05 p.64	2.1757	#12 p.49
2 位	#05 p.64	0.4331	#05 p.77	0.4306	#15 p.43	2.1486	#05 p.81
3 位	#05 p.77	0.4136	#05 p.75	0.3967	#15 p.56	2.1191	#05 p.73

表 56: 表 55Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 57: CS106L における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.74	0.1762	#04 p.74	0.1990	#04 p.74	0.5145	#04 p.74
2 位	#04 p.77	0.1024	#04 p.77	0.1109	#04 p.77	0.3329	#04 p.77

表 58: 表 57 の Q-measure の値

Q-measure	
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 59: CS107 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#08 p.141	0.4249	#08 p.141	0.6814	#08 p.141	8.5134	#09 p.13
2 位	#05 p.55	0.3128	#08 p.140	0.3649	#05 p.11	5.7726	#04 p.12
3 位	#06 p.68	0.3005	#05 p.55	0.3374	#08 p.140	4.5987	#04 p.96

表 60: 表 59Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

A.3 CS107

表 61: CS107 における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#14 p.12	0.4384	#07 p.52	0.4528	#14 p.13	3.3769	#14 p.15
2 位	#16 p.5	0.3688	#14 p.119	0.3820	#14 p.119	2.8981	#07 p.65
3 位	#16 p.20	0.3655	#14 p.13	0.3472	#14 p.9	2.7719	#07 p.68

表 62: 表 61 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 63: CS107 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#13 p.34	0.3655	#13 p.34	0.5226	#07 p.5	13.1542	#08 p.99
2 位	#08 p.134	0.3276	#08 p.134	0.3645	#13 p.34	7.4636	#07 p.83
3 位	#08 p.99	0.2622	#08 p.99	0.3218	#08 p.134	4.8886	#13 p.23

表 64: 表 12 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.0

表 65: CS107 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.93	0.2520	#11 p.10	0.3027	#05 p.21	12.2411	#05 p.21
2 位	#07 p.68	0.2291	#02 p.46	0.2984	#11 p.10	7.0997	#05 p.72
3 位	#02 p.46	0.1954	#07 p.52	0.2920	#07 p.68	2.9367	#02 p.46

表 66: 表 65 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0740
正規化した C-TF-IDF	0.1693
重み付けした C-TF-IDF	0.6719

表 67: CS107 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#09 p.98	0.3711	#05 p.20	0.3143	#05 p20	11.7891	#06 p.5
2 位	#06 p.98	0.2994	#05 p.25	0.3111	#05 p.25	7.5189	#09 p.42
3 位	#05 p.25	0.2748	#09 p.98	0.3074	#05 p.26	3.7175	#05 p.42

表 68: 表 67Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 69: CS107 における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#14 p.116	0.4071	#14 p.116	0.1069	#14 p.116	0.1173	#14 p.166

表 70: 表 69 の Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 71: CS110 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#12 p.2	0.0826	#07 p.11	0.0994	#07 p.11	0.2984	#12 p.2
2 位	#07 p.11	0.0685	#04 p.11	0.0744	#12 p.2	0.2624	#07 p.9
3 位	#21 p.16	0.0601	#12 p.2	0.0689	#04 p.11	0.2233	#08 p.7

表 72: 表 71 と Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.3386

A.4 CS110

表 73: CS110 における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.4	0.0327	#05 p.6	0.0343	#05 p.4	0.1307	#05 p.4
2 位	#05 p.12	0.0256	#05 p.12	0.0272	#04 p.7	0.1005	#05 p.12
3 位	#04 p.7	0.0237	#04 p.7	0.0263	#05 p.12	0.0774	#04 p.7

表 74: 表 73Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	0.9523

表 75: CS110 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.6	0.1622	#05 p.6	0.2157	#06 p.6	0.8216	#06 p.6
2 位	#06 p.12	0.0221	#05 p.12	0.0303	#06 p.12	0.1365	#06 p.12
3 位	#05 p.12	0.0214	#06 p.12	0.0284	#05 p.12	0.1354	#05 p.12

表 76: 表 75Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	0.9523809524
重み付けした C-TF-IDF	1.0

表 77: CS110 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#12 p.5	0.3259	#12 p.5	0.2825	#12 p.2	5.8121 #02 p.21	
2 位	#12 p.11	0.2263	#12 p.11	0.1934	#12 p.5	3.4768 #12 p.11	
3 位	#12 p.2	0.2179	#12 p.2	0.1719	#12 p.7	1.7387	#11 p.4

表 78: 表 77 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.2539
正規化した C-TF-IDF	0.2539
重み付けした C-TF-IDF	0.0

表 79: CS110 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.9	0.1005	#04 p.9	0.1307	#04 p.9	0.3922	#05 p.7
2 位	#21 p.16	0.0665	#05 p.7	0.0832	#05 p.7	0.3168	#04 p.15
3 位	#05 p.7	0.0657	#21 p.16	0.0696	#21 p.16	0.2652	#21 p.16

表 80: 表 79 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3174
正規化した C-TF-IDF	0.4126
重み付けした C-TF-IDF	0.4126

表 81: CS110 における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.17	0.0549	#05 p.19	0.0806	#05 p.19	0.2418	#05 p.18
2 位	#05 p.19	0.0507	#05 p.17	0.0492	#05 p.17	0.1875	#05 p.19
3 位	#05 p.18	0.0254	#05 p.18	0.0327	#05 p.18	0.1248	#04 p.20

表 82: 表 81 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.4497
正規化した C-TF-IDF	0.7830
重み付けした C-TF-IDF	0.7830

表 83: CS143 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#18 p.5	0.5171	#18 p.14	0.4514	#18 p.9	4.8372	#18 p.9
2 位	#18 p.9	0.4046	#18 p.9	0.4221	#18 p.14	2.0133	#18 p.16
3 位	#18 p.14	0.3589	#18 p.5	0.4037	#18 p.5	1.8002	#12 p.54

表 84: 表 83 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3386
正規化した C-TF-IDF	0.3386
重み付けした C-TF-IDF	0.6719

A.5 CS143

表 85: CS143 における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#11 p.37	0.7890	#11 p.37	0.6611	#11 p.37	4.8409	#11 p.37
2 位	#11 p.36	0.3102	#17 p.21	0.3488	#17 p.21	1.6396	#11 p.35
3 位	#17 p.21	0.3071	#11 p.36	0.3184	#11 p.36	1.2125	#17 p.23

表 86: 表 85Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.6719
重み付けした C-TF-IDF	0.6719

表 87: CS143 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#07 p.65	0.5941	#07 p.12	0.8277	#11 p.50	5.6734	#07 p.65
2 位	#07 p.12	0.5569	#07 p.65	0.7815	#07 p.65	4.0405	#11 p.41
3 位	#11 p.21	0.3407	#11 p.4	0.3724	#09 p.22	3.7772	#07 p.10

表 88: 表 87 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.3386
重み付けした C-TF-IDF	0.3386

表 89: CS143 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#11 p.6	1.0248	#11 p.6	0.7575	#11 p.6	5.9146	#17 p.5
2 位	#11 p.21	0.9688	#11 p.21	0.7112	#11 p.21	5.1972	#02 p.25
3 位	#11 p.33	0.6956	#11 p.33	0.5987	#11 p.37	4.3106	#12 p.5

表 90: 表 89Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 91: CS143 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#12 p.19	0.4035	#12 p.19	0.4003	#17 p.28	7.4498	#17 p.8
2 位	#17 p.30	0.3017	#17 p.32	0.3742	#17 p.30	4.5536	#17 p.35
3 位	#17 p.32	0.2895	#17 p.36	0.3057	#12 p.19	1.1401	#17 p.37

表 92: 表 91 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 93: CS143 における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.44	0.3013	#05 p.44	0.2816	#06 p.14	1.7619	#05 p.44
2 位	#05 p.45	0.2504	#05 p.45	0.2620	#05 p.44	0.8448	#05 p.50
3 位	#05 p.8	0.2302	#06 p.12	0.2271	#05 p.47	0.8310	#06 p.34

表 94: 表 93 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.6719
重み付けした C-TF-IDF	0.3386

表 95: CS149 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#17 p.12	0.1522	#17 p.12	0.1313	#17 p.12	1.1329	#03 p.10
2 位	#18 p.60	0.1209	#04 p.18	0.1191	#18 p.60	0.9077	#03 p.57
3 位	#07 p.28	0.0846	#18 p.60	0.1071	#19 p.73	0.6853	#18 p.61

表 96: 表 95 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

A.6 CS149

表 97: CS149 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#13 p.41	0.2635	#13 p.41	0.3769	#13 p.37	2.1369	#05 p.34
2 位	#05 p.5	0.1701	#13 p.42	0.2732	#13 p.42	1.3791	#05 p.37
3 位	#13 p.42	0.1578	#05 p.34	0.2097	#18 p.79	0.8333	#13 p.39

表 98: 表 97 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 99: CS149 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.34	0.4320	#06 p.34	0.4896	#06 p.34	4.6582	#12 p.9
2 位	#18 p.59	0.3552	#03 p.35	0.4472	#18 p.59	4.1327	#03 p.29
3 位	#02 p.44	0.3516	#02 p.44	0.3703	#14 p.43	3.9618	#07 p.79

表 100: 表 99 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 101: CS149 における **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.18	0.4960	#05 p.37	0.6178	#13 p.34	3.0084	#05 p.18
2 位	#05 p.11	0.4271	#05 p.18	0.5916	#13 p.36	2.0272	#05 p.43
3 位	#05 p.37	0.3970	#05 p.35	0.4942	#05 p.18	1.7749	#05 p.40

表 102: 表 101 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.3386
重み付けした C-TF-IDF	0.1481

表 103: CS149 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#11 p.32	0.1847	#11 p.29	0.2591	#11 p.29	1.0367	#16 p.69
2 位	#11 p.31	0.1384	#11 p.32	0.2179	#11 p.32	0.5447	#11 p.30
3 位	#11 p.29	0.1374	#07 p.44	0.1752	#07 p.44	0.5257	#17 p.21

表 104: 表 103 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 105: CS149 における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#13 p.30	0.7422	#13 p.30	0.9147	#13 p.30	2.3646	#14 p.16
2 位	#11 p.91	0.3008	#11 p.91	0.3349	#11 p.91	1.2009	#13 p.30
3 位	#14 p.16	0.2176	#14 p.16	0.2267	#14 p.16	0.8128	#13 p.30

表 106: 表 105Q-measure の値

	Q-measure
通常の C-TF-IDF	0.7857
正規化した C-TF-IDF	0.7857
重み付けした C-TF-IDF	0.7857

表 107: CS149 における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#14 p.11	0.2535	#14 p.11	0.3125	#14 p.11	0.9642	#14 p.54
2 位	#14 p.54	0.2216	#14 p.54	0.3012	#09 p.38	0.7975	#19 p.36
3 位	#14 p.13	0.1557	#09 p.38	0.1993	#14 p.54	0.7788	#09 p.30

表 108: 表 107 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3386
正規化した C-TF-IDF	0.3386
重み付けした C-TF-IDF	0.1481

表 109: CS161 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.15	0.4191	#02 p.108	0.4225	#11 p.49	6.5712	#02 p.107
2 位	#02 p.105	0.3468	#06 p.15	0.3552	#11 p.50	3.5866	#011 p.52
3 位	#11 p.49	0.2970	#11 p.49	0.2985	#04 p.34	2.2065	#04 p.28

表 110: 表 109 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

A.7 CS161

表 111: CS161 における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#11 p.53	0.6694	#11 p.53	0.8616	#11 p.53	1.7233	#11 p.54
2 位	#11 p.54	0.2718	#11 p.54	0.5179	#11 p.54	1.0358	#16 p.106
3 位	#15 p.93	0.0729	#15 p.93	0.1119	#15 p.93	0.2894	#15 p.93

表 112: 表 111 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3386
正規化した C-TF-IDF	0.4126
重み付けした C-TF-IDF	0.4126

表 113: CS161 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#05 p.40	0.1251	#05 p.40	0.1091	#05 p.108	0.3664	#05 p.108
2 位	#01 p.49	0.0980	#01 p.49	0.0965	#05 p.40	0.3273	#01 p.49
3 位	#02 p.8	0.0911	#02 p.8	0.0725	#01 p.49	0.2895	#05 p.40

表 114: 表 113 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.5899
正規化した C-TF-IDF	0.5899
重み付けした C-TF-IDF	0.9523

表 115: CS161 における **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.76	0.1041	#06 p.76	0.0963	#06 p.76	0.3668	#09 p.30
2 位	#15 p.94	0.0738	#15 p.94	0.0684	#09 p.30	0.2787	#06 p.76
3 位	#09 p.30	0.0581	#09 p.30	0.0607	#16 p.67	0.2046	#15 p.95

表 116: 表 115 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6521
正規化した C-TF-IDF	0.6521
重み付けした C-TF-IDF	0.8425

表 117: CS161 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#07 p.57	0.0928	#07 p.57	0.1053	#07 p.17	0.3770	#07 p.18
2 位	#07 p.17	0.0807	#07 p.17	0.0990	#07 p.57	0.2724	#07 p.11
3 位	#07 p.11	0.0690	#07 p.11	0.0727	#07 p.11	0.2071	#07 p.57

表 118: 表 117 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.4470
正規化した C-TF-IDF	0.4470
重み付けした C-TF-IDF	0.2804

表 119: CS161 における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.125	0.9381	#16 p.11	0.6902	#16 p.11	6.2111	#07 p.100
2 位	#07 p.49	0.7716	#06 p.125	0.5853	#07 p.49	4.0249	#06 p.33
3 位	#16 p.11	0.5888	#07 p.49	0.5572	#06 p.125	3.6570	#09 p.53

表 120: 表 119 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 121: CS161 における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#07 p.121	1.6127	#07 p.121	0.7855	#08 p.93	5.5120	#08 p.8
2 位	#08 p.3	0.7707	#08 p.3	0.6822	#08 p.3	5.3267	#08 p.99
3 位	#08 p.93	0.6334	#08 p.93	0.6244	#08 p.94	5.2610	#08 p.94

表 122: 表 121 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0740

表 123: CS166 について **array** でのランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#16 p.77	0.4342	#16 p.77	0.3653	#04 p.105	25.9503	#17 p.33
2 位	#10 p.165	0.3578	#10 p.166	0.3417	#18 p.4	5.3742	#04 p.105
3 位	#04 p.106	0.3293	#17 p.33	0.2754	#16 p.77	5.2418	#10 p.118

表 124: 表 123 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.5039

A.8 CS166

表 125: CS166 における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.348	1.1141	#04 p.348	1.1138	#05 p.253	4.5004	#05 p.12
2 位	#05 p.2	0.5563	#05 p.2	0.6389	#06 p.435	3.9037	#06 p.435
3 位	#05 p.12	0.5025	#05 p.12	0.6325	#04 p.348	3.8986	#05 p.253

表 126: 表 125 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.5899

表 127: CS166 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.93	0.4431	#04 p.93	0.6201	#04 p.242	43.5064	#04 p.276
2 位	#01 p.102	0.4153	#01 p.102	0.6104	#04 p.93	27.3673	#01 p.102
3 位	#04 p.215	0.4089	#04 p.215	0.5157	#01 p.109	2.8830	#04 p.94

表 128: 表 127 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.2539
正規化した C-TF-IDF	0.2539
重み付けした C-TF-IDF	0.0

表 129: CS166 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#02 p.182	0.7087	#02 p.182	0.6296	#12 p.22	3.5863	#10 p.44
2 位	#12 p.22	0.4422	#12 p.22	0.5330	#12 p.23	3.2254	#13 p.10
3 位	#10 p.44	0.3412	#12 p.23	0.4727	#12 p.24	2.9508	#02 p.181

表 130: 表 129 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.1481
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 131: CS166 における **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#04 p.13	1.2278	#04 p.13	0.7458	#04 p.33	7.8094	#04 p.93
2 位	#05 p.65	0.3176	#04 p.2	0.2735	#04 p.103	5.3006	#05 p.4
3 位	#04 p.2	0.2763	#05 p.65	0.2500	#06 p.16	2.9261	#04 p.193

表 132: 表 131Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 133: CS166 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.435	0.4397	#06 p.424	0.4372	#06 p.112	2.5132	#06 p.112
2 位	#06 p.424	0.3175	#06 p.435	0.4302	#06 p.424	2.2245	#06 p.435
3 位	#06 p.420	0.3144	#06 p.420	0.3183	#06 p.435	2.1887	#04 p.149

表 134: 表 133 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.5039
正規化した C-TF-IDF	0.2539
重み付けした C-TF-IDF	0.7830

表 135: CS166 における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.285	1.1260	#01 p.60	0.7841	#06 p.285	5.6504	#04 p.122
2 位	#17 p.3	0.7664	#06 p.285	0.5850	#17 p.3	4.9316	#01 p.60
3 位	#02 p.70	0.5094	#17 p.3	0.4759	#05 p.84	4.2281	#14 p.110

表 136: 表 135 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.5039
重み付けした C-TF-IDF	0.0

表 137: CS166 における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#07 p.4	0.7076	#07 p.4	0.6898	#13 p.18	10.5168	#13 p.103
2 位	#07 p.3	0.5469	#07 p.6	0.6282	#08 p.7	8.7048	#13 p.86
3 位	#07 p.6	0.5209	#12 p.77	0.4561	#07 p.13	8.3551	#07 p.179

表 138: 表 137 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 139: CS205L における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#13 p.9	0.0714	#13 p.9	0.0869	#13 p.9	0.3310	#13 p.9
2 位	#19 p.9	0.0611	#19 p.9	0.0694	#19 p.9	0.2643	#19 p.9
3 位	#18 p.14	0.0354	#18 p.14	0.0555	#18 p.14	0.2114	#18 p.13

表 140: 表 139Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

A.9 CS205L

表 141: CS205L における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#18 p.13	0.1114	#18 p.13	0.1355	#18 p.13	0.3859	#18 p.13

表 142: 表 141 の Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 143: CS205L における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.15	0.2281	#06 p.15	0.2528	#06 p.15	0.5056	#06 p.15

表 144: 表 143 の Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 145: CS224N における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#09 p.12	0.0985	#09 p.12	0.1001	#06 p.56	0.5004	#06 p.56
2 位	#06 p.56	0.0747	#04 p.38	0.0747	#09 p.12	0.2851	#04 p.27
3 位	#04 p.38	0.0732	#06 p.56	0.0714	#04 p.38	0.2241	#09 p.18

表 146: 表 145Q-measure の値

	Q-measure
通常の C-TF-IDF	0.3386
正規化した C-TF-IDF	0.1481
重み付けした C-TF-IDF	0.6719

A.10 CS224N

表 147: CS224N における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.44	0.2710	#06 p.44	0.1845	#15 p.26	0.5676	#06 p.42
2 位	#15 p.38	0.1554	#15 p.38	0.1461	#06 p.44	0.4921	#15 p.38
3 位	#06 p.41	0.1416	#06 p.41	0.1260	#06 p.41	0.3781	#15 p.13

表 148: 表 147 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.2539
正規化した C-TF-IDF	0.2539
重み付けした C-TF-IDF	0.0

表 149: CS224N における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#08 p.9	0.0604	#08 p.9	0.0732	#08 p.9	0.2085	#08 p.9

表 150: 表 149 の Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 151: CS224N における **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#17 p.50	0.2384	#17 p.50	0.3005	#17 p.50	9.7129	#17 p.50
2 位	#13 p.5	0.2355	#13 p.5	0.1999	#13 p.5	0.7610	#04 p.17
3 位	#13 p.23	0.1571	#13 p.23	0.1941	#13 p.23	0.7391	#04 p.25

表 152: 表 151 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.6719
正規化した C-TF-IDF	0.6719
重み付けした C-TF-IDF	0.6719

表 153: CS224N における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#02 p.11	0.0593	#02 p.11	0.0555	#02 p.11	0.1583	

表 154: 表 153 の Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 155: CS243 における **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#09 p.20	0.2593	#09 p.20	0.2760	#11 p.27	1.0709	#15 p.19
2 位	#11 p.27	0.2365	#11 p.27	0.2071	#15 p.20	1.0031	#07 p.4
3 位	#11 p.14	0.2295	#11 p.14	0.2042	#11 p.14	1.0023	#09 p.10

表 156: 表 155 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

A.11 CS243

表 157: CS243 における **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#06 p.19	0.1968	#06 p.19	0.3009	#06 p.19	0.6019	#14 p.7
2 位	#14 p.4	0.0635	#14 p.4	0.0840	#14 p.4	0.2171	#14 p.4
3 位	#14 p.11	0.0609	#14 p.11	0.0736	#14 p.11	0.1903	#06 p.19

表 158: 表 157 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.5899
正規化した C-TF-IDF	0.5899
重み付けした C-TF-IDF	0.5899

表 159: CS243 における **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#12 p.27	0.0898	#12 p.27	0.1861	#12 p.27	0.5583	#14 p.11
2 位	#14 p.11	0.0685	#14 p.11	0.0781	#14 p.11	0.2224	#12 p.27

表 160: 表 159 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.875
正規化した C-TF-IDF	0.875
重み付けした C-TF-IDF	0.875

表 161: CS243 における **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#14 p.9	0.3703	#14 p.9	0.2629	#13 p.61	1.8389	#14 p.11
2 位	#09 p.11	0.1805	#07 p.15	0.1727	#11 p.11	0.8494	#10 p.2
3 位	#14 p.10	0.1679	#09 p.11	0.1670	#14 p.9	0.6796	#07 p.4

表 162: 表 161 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 163: CS243 における **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#12 p.21	0.2656	#12 p.21	0.2823	#12 p.11	2.1873	#14 p.5
2 位	#12 p.19	0.1978	#12 p.19	0.2155	#12 p.21	1.7765	#1 p.28
3 位	#13 p.6	0.1967	#12 p.11	0.2056	#13 p.6	1.4606	#12 p.25

表 164: 表 163 の Q-measure の値

	Q-measure
通常の C-TF-IDF	0.0
正規化した C-TF-IDF	0.0
重み付けした C-TF-IDF	0.0

表 165: CS243 における **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF		筆者作成のランキング
	Page	Score	Page	Score	Page	Score	Page
1 位	#13 p.55	0.2084	#13 p.55	0.6763	#13 p.55	0.2374	#13 p.55

表 166: 表 165 の Q-measure の値

	Q-measure
通常の C-TF-IDF	1.0
正規化した C-TF-IDF	1.0
重み付けした C-TF-IDF	1.0

表 167: **array** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS143 #18 p.5	0.5171	CS107 #08 p.141	0.6768	CS166 #04 p.105	25.9503
2 位	CS106B #27 p.6	0.5166	CS106B #18 p.21	0.5800	CS107 #08 p.141	8.6446
3 位	CS166 #16 p.77	0.4342	CS106L #10 p.67	0.5139	CS161 #11 p.49	6.5712
4 位	CS107 #08 p.141	0.4233	CS106L #10 p.68	0.4592	CS107 #05 p.11	5.7796
5 位	CS161 #06 p.15	0.4191	CS143 #18 p.14	0.4514	CS166 #18 p.4	5.3742

B　被験者による実験の結果

B.1　array

表 168: 被験者が作成した **array** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS166	CS107	CS143	CS166	CS107
	#04 p.105	#08 p.141	#18 p.14	#18 p.4	#05 p.11
B	CS107	CS107	CS166	CS143	CS161
	#08 p.141	#05 p.11	#04 p.105	#18 p.5	#11 p.49
C	CS166	CS106L	CS106L	CS106B	CS143
	#04 p.105	#10 p.67	#10 p.68	#27 p.6	#18 p.5
D	CS166	CS166	CS143	CS107	CS166
	#04 p.105	#18 p.4	#18 p.5	#05 p.11	#16 p.77
E	CS107	CS107	CS166	CS166	CS106B
	#05 p.11	#08 p.141	#04 p.105	#18 p.4	#18 p.21
F	CS106L	CS143	CS106B	CS166	CS107
	#10 p.67	#18 p.5	#18 p.21	#04 p.105	#05 p.11
G	CS106L	CS143	CS106L	CS107	CS161
	#10 p.67	#18 p.5	#10 p.68	#05 p.11	#06 p.15
H	CS107	CS166	CS166	CS107	CS166
	#05 p.11	#18 p.4	#04 p.105	#08 p.141	#16 p.77
I	CS166	CS107	CS166	CS166	CS143
	#04 p.105	#05 p.11	#16 p.77	#18 p.4	#18 p.14
J	CS107	CS166	CS143	CS143	CS106b
	#05 p.11	#04 p.105	#18 p.5	#18 p.14	#27 p.6

表 169: **array** の被験者ごとの Q-measure

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.1055	0.4697	0.8511
B	0.3845	0.5157	0.7651
C	0.3297	0.2566	0.5157
D	0.4127	0.0	0.6290
E	0.1055	0.5350	0.7636
F	0.4297	0.4171	0.3001
G	0.4297	0.2911	0.06333
H	0.1322	0.2578	0.6783
I	0.1377	0.02	0.6513
J	0.4490	0.03	0.5564
平均	0.2916	0.2793	0.5774

表 170: **array** で得られたコメント

被験者	コメント
A	arrays について学習しているため，それの復習や，他の言語での応用について同時に学ぶことができると嬉しいため.
B	上の画像は Array の概要が載っているいて、動作例と一緒に見れば理解が深まると思った。
C	情報量が増えると混乱するため配列自体の説明をしているスライドを上の順位にしました。
D	関係する資料の中で単体である程度理解できそうなものを選んだから
E	最初に配列の概要から入り、C の配列と Java の配列を比べ、最後に配列の応用を示してると思ったから
F	図がある方がイメージがしやすく、またプログラムが書いてある方が実際に入力することができる
G	図、コード、簡単な説明が載っているものから選んだ
H	配列とポインタについて知りたいため
I	関連性の高そうなものを選びました。
J	内容の関連度がより高く，かつ比較的基礎的な内容のものが表示されると分野を概観するのに役立つと考えたため.

表 171: **stack** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS143 #07 p.65	0.5941	CS143 #07 p.12	0.8277	CS166 #04 p.242	43.5064
2 位	CS143 #07 p.12	0.5569	CS143 #07 p.65	0.7815	CS107 #07 p.5	13.2791
3 位	CS166 #04 p.93	0.4431	CS166 #04 p.93	0.6201	CS107 #13 p.34	7.4804
4 位	CS166 #01 p.102	0.4153	CS166 #01 p102	0.6104	CS143 #11 p.50	5.6734
5 位	CS166 #04 p.215	0.4089	CS107 #13 p.34	0.5215	CS166 #04 p.93	5.5566

B.2 stack

表 172: 被験者が作成した **stack** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS107 #13 p.34	CS107 #07 p.5	CS166 #04 p.93	CS166 #04 p.242	CS143 #07 p.12
B	CS143 #07 p.65	CS143 #11 p.50	CS107 #13 p.34	CS107 #07 p.5	CS166 #04 p.242
C	CS166 #04 p.242	CS166 #04 p.93	CS143 #07 p.65	CS143 #11 p.50	CS107 #13 p.34
D	CS166 #01 p.102	CS143 #11 p.50	CS143 #07 p.65	CS107 #13 p.34	CS166 #04 p.93
E	CS107 #13 p.34	CS166 #04 p.242	CS107 #07 p.5	CS143 #07 p.12	CS166 #04 p.93
F	CS107 #13 p.34	CS166 #04 p.242	CS166 #04 p.93	CS166 #01 p102	CS107 #07 p.5
G	CS107 #13 p.34	CS166 #04 p.242	CS107 #07 p.5	CS166 #04 p.93	CS143 #07 p.65
H	CS166 #04 p.242	CS107 #13 p.34	CS107 #07 p.5	CS166 #04 p.93	CS143 #07 p.65
I	CS166 #04 p.93	CS166 #04 p.242	CS143 #07 p.65	CS107 #13 p.34	CS107 #07 p.6
J	CS107 #07 p.5	CS107 #13 p.34	CS166 #01 p.102	CS166 #04 p.93	CS166 #04 p.242

表 173: **stack** の被験者ごとの Q-measure

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.2430	0.3696	0.7676
B	0.5157	0.3557	0.5731
C	0.5160	0.4027	0.6979
D	0.5393	0.4360	0.2288
E	0.2267	0.3867	0.8669
F	0.2011	0.2611	0.7817
G	0.2752	0.2685	0.8769
H	0.2752	0.2585	0.9284
I	0.5505	0.4720	0.6843
J	0.1877	0.4315	0.5567
平均	0.3530	0.3626	0.6962

表 174: **stack** で得られたコメント

被験者	コメント
A	他と同様.stack について学んでいるため，他の資料で同じような説明をしているページ を確認できれば，理解が深まるため.
B	上の画像にはスタックの流れが書いてあるがスタックについての説明がなく、スタックについての説明が欲しかったから。
D	関連性が高いと思ったから。
E	先に図の説明からはいり、そのあとプログラムの説明でわかりやすいと思ったから
F	図があるものがやはりイメージがしやすいからです。その中でもレジスタではスタックがどうなっているのかを表しているものは優先度が高かったです。画像 A と画像 B の関連性が高そうだったので、そこを並べました。
G	stack に対してイメージがすぐ湧きやすいものから選択した
H	スタックの概念を知りたいから
I	関連数る内容であったためです。
J	図や表を用いた説明があると、学習内容をイメージするのに役立つと考えたため。

表 175: **heap** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS166 #04 p.348	1.1141	CS166 #04 p.348	1.1138	CS143 #11 p.37	4.8409
2 位	CS143 #11 p.37	0.7890	CS161 #11 p.53	0.8616	CS166 #05 p.253	4.5004
3 位	CS161 #11 p.53	0.6694	CS143 #11 p.37	0.6611	CS166 #06 p.435	3.9037
4 位	CS166 #05 p.2	0.5563	CS166 #05 p.2	0.6389	CS166 #04 p.348	3.8986
5 位	CS166 #05 p.12	0.5025	CS166 #05 p.12	0.6325	CS166 #06 p.4	3.4466

B.3 heap

表 176: 被験者が作成した **heap** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS166 #06 p.435	CS166 #05 p.12	CS166 #06 p.4	CS143 #11 p.37	CS166 #05 p.253
B	CS143 #11 p.37	CS166 #04 p.348	CS166 #05 p.12	CS166 #06 p.4	CS166 #05 p.253
C	CS143 #11 p.37	CS166 #05 p.12	CS166 #06 p.4	CS166 #05 p.12	CS166 #06 p.435
D	CS166 #05 p.12	CS166 #06 p.4	CS143 #11 p.37	CS166 #05 p.2	CS166 #04 p.348
E	CS143 #11 p.37	CS166 #06 p.435	CS166 #06 p.4	CS161 #11 p.53	CS166 #05 p.253
F	CS143 #11 p.37	CS166 #05 p.253	CS166 #06 p.4	CS166 #05 p.12	CS166 #05 p.2
G	CS143 #11 p.37	CS166 #06 p.4	CS166 #06 p.4	CS166 #05 p.12	CS166 #06 p.435
H	CS166 #06 p.4	CS166 #05 p.12	CS166 #06 p.435	CS166 #04 p.348	CS166 #05 p.253
I	CS166 #05 p.12	CS166 #04 p.348	CS166 #06 p.4	CS166 #06 p.435	CS161 #11 p.53
J	CS143 #11 p.37	CS166 #05 p.2	CS166 #04 p.348	CS166 #05 p.12	CS166 #06 p.435

表 177: **heap** の被験者ごとの Q-measure

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.2078	0.1533	0.6097
B	0.7855	0.6764	0.8598
C	0.4290	0.320	0.8514
D	0.5057	0.4330	0.7661
E	0.3157	0.2066	0.8332
F	0.3879	0.2788	0.9566
G	0.3457	0.2366	0.9484
H	0.3078	0.3078	0.4352
I	0.4897	0.4897	0.2833
J	0.7951	0.6880	0.9522
平均	0.4579	0.3788	0.7496

表 178: **heap** で得られたコメント

被験者	コメント
A	heep についての発展的な内容を確認することで，学習時のモチベーションにつながるから.
B	2 項目目に Heep のメモリーについて書いてあったから
C	混乱しないかつ分かりやすいため、ヒープそのものについて説明しており、図が載っているスライドを上位にしました。
D	関係がありそうで分かりやすそうなものを選んだから。
E	ヒープについて説明から順当に話してると思ったから
F	heap について図で書かれているため、E の優先度が高く、より内容が詳しく書かれているものが優先度が高くなりました。
G	図がわかりやすいもの、説明がわかりやすいものから選んだ
H	ツリー構造について知りたいため
I	具体的な例示があったためです。
J	学習中の内容に関連する項目をあらかじめ示されることで，全体像がイメージしやすくなると考えたため.

表 179: **memory** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS143 #11 p.6	1.0248	CS143 #11 p.6	0.7575	CS107 #05 p.21	12.2627
2 位	CS166 #02 p.182	0.7087	CS166 #02 p.182	0.6296	CS143 #11 p.6	5.9146
3 位	CS143 #11 p.37	0.6669	CS143 #11 p.37	0.5403	CS110 #12 p2	5.8121
4 位	CS106L #05 p.64	0.4643	CS166 #12 p.22	0.5330	CS143 #11 p.21	5.1972
5 位	CS166 #12 p.22	0.4422	CS149 #06 p.34	0.4896	CS149 #06 p.34	4.6582

B.4 memory

表 180: 被験者が作成した **memory** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS107 #05 p.21	CS149 #06 p.34	CS110 #12 p2	CS143 #11 p.37	CS143 #11 p.6
B	CS107 #05 p.21	CS143 #11 p.6	CS143 #11 p.21	CS106L #05 p.64	CS149 #06 p.34
C	CS110 #12 p2	CS107 #05 p.21	CS143 #11 p.21	CS143 #11 p.6	CS149 #06 p.34
D	CS107 #05 p.21	CS110 #12 p2	CS143 #11 p.21	CS143 #11 p.37	CS149 #06 p.34
E	CS143 #11 p.6	CS143 #11 p.21	CS143 #11 p.37	CS110 #12 p2	CS149 #06 p.34
F	CS143 #11 p.37	CS143 #11 p.21	CS143 #11 p.6	CS110 #12 p2	CS149 #06 p.34
G	CS143 #11 p.37	CS143 #11 p.21	CS143 #11 p.6	CS149 #06 p.34	CS110 #12 p2
H	CS107 #05 p.21	CS143 #11 p.6	CS166 #02 p.182	CS143 #11 p.37	CS149 #06 p.34
I	CS143 #11 p.37	CS107 #05 p.21	CS149 #06 p.34	CS110 #12 p.2	CS143 #11 p.21
J	CS107 #05 p.21	CS110 #12 p.2	CS143 #11 p.6	CS143 #11 p.21	CS143 #11 p.37

表 181: **memory** の被験者ごとの Q-measure

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.2752	0.3252	0.8087
B	0.4931	0.4497	0.8833
C	0.2578	0.2778	0.8987
D	0.1033	0.1233	0.7924
E	0.6535	0.6735	0.5446
F	0.5505	0.5705	0.4393
G	0.5505	0.5805	0.4149
H	0.7436	0.7636	0.7988
I	0.2066	0.2466	0.6153
J	0.4127	0.4127	0.9618
平均	0.4247	0.4705	0.7158

表 182: **memory** で得られたコメント

コメント	
A	メモリーについての過去の資料や，発展的な内容を確認することで，理解を深められるため.
B	メモリーの概要と一緒に見れば理解が深まると思った。
C	スライドに図が載っていたので、さらに図が載っているものではなく、説明が多めのスライドを上位に持ってきました。
D	関わりのありそうだと思ったもので分かりやすそうなものを選んだから。
E	図から説明してるのかなと思った
F	図と一緒に考える方がイメージができるため、図があるものが優先度が高く、説明が書かれているだけのものはそれだけではイメージがしにくいため優先度は低かったです。
G	図と説明が載っているやつから選びました。
H	メモリについて知りたいため
H	メモリのレイアウトがわかりやすく図示されているためです。
H	”Take CS107 to learn much more!!”というスライドの表示通り直接発展的な内容を示されることで、学習中の分野を概観し整理して理解することに役立つと考えたため。

表 183: **queue** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS166 #04 p.13	1.2278	CS166 #04 p.13	0.7458	CS166 #04 p.33	7.8094
2 位	CS149 #05 p.18	0.4960	CS149 #05 p.37	0.6178	CS166 #04 p.103	5.3006
3 位	CS106L #04 p.05	0.4514	CS149 #05 p.18	0.5916	CS149 #13 p.34	3.0084
4 位	CS149 #05 p.11	0.4271	CS149 #05 p.35	0.4942	CS166 #04 p.193	2.9652
5 位	CS149 #05 p.37	0.3970	CS149 #05 p.11	0.4625	CS166 #05 p.16	2.9261

B.5 queue

表 184: 被験者が作成した **queue** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS166 #04 p.193	CS166 #05 p.16	CS149 #05 p.18	CS166 #04 p.103	CS149 #05 p.35
B	CS149 #13 p.34	CS149 #05 p.11	CS149 #05 p.18	CS166 #04 p.193	CS166 #04 p.103
C	CS166 #04 p.193	CS166 #04 p.103	CS149 #05 p.37	CS166 #04 p.33	CS149 #05 p.11
D	CS149 #13 p.34	CS166 #05 p.16	CS166 #04 p.193	CS166 #04 p.103	CS149 #05 p.11
E	CS106L #04 p.05	CS149 #05 p.11	CS149 #05 p.18	CS149 #05 p.37	CS149 #05 p.35
F	CS149 #13 p.34	CS149 #05 p.18	CS149 #05 p.35	CS149 #05 p.11	CS166 #04 p.193
G	CS149 #05 p.11	CS149 #05 p.18	CS149 #05 p.35	CS149 #05 p.37	CS166 #04 p.193
H	CS166 #04 p.193	CS166 #04 p.103	CS166 #04 p.33	CS166 #04 p.13	CS149 #05 p.37
I	CS166 #04 p.103	CS149 #13 p.34	CS166 #04 p.13	CS166 #04 p.33	CS166 #04 p.193
J	CS166 #04 p.193	CS149 #05 p.35	CS149 #05 p.18	CS149 #13 p.11	CS149 #05 p.34

表 185: **queue** の順位ごとの Q-measure

	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.2105	0.180	0.3345
B	0.3160	0.1877	0.3752
C	0.040	0.2527	0.6476
D	0.04222	0.020	0.4989
E	0.5527	0.3878	0.0
F	0.3264	0.2866	0.2488
G	0.4197	0.4745	0.04222
H	0.2778	0.3631	0.7336
I	0.3438	0.4297	0.5157
J	0.2738	0.3675	0.6232
平均	0.2803	0.2950	0.4020

表 186: **queue** で得られたコメント

コメント	
A	学んでいるキューについての他の例や，発展的な内容を知ることで，深い理解を得られるため.
B	キューの概要が知れたので、一緒に実装例が見たかったから。
D	関連性がありそうなものとスライド 1 枚で内容が分かりやすそうなものを優先して選んだから
E	なんの説明をするかがほしいと思ったから
F	プログラムが書かれているものが実際に試すことができるので、優先度が高いです。図があるものは関連するものを並べました。
G	上位の順位を決めるのが難しかった。
H	先入れ先出しの動きが知りたいから
I	他の内容とがまとめられているため。
J	学習中のスライドに加えて特定の 1 枚のスライドが提示されるとしたら、ある程度内容が詰まったスライドのほうが学習に役立つかもしれないと考えたため。

表 187: **pointer** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS106L #05 p.2	1.7987	CS106L #05 p.2	0.5995	CS107 #05 p.20	11.8060
2 位	CS166 #06 p.435	0.4397	CS106B #17 p.32	0.4828	CS107 #05 p.25	7.5299
3 位	CS106L #05 p.64	0.4331	CS166 #06 p.424	0.4372	CS143 #17 p.28	4.5536
4 位	CS106L #05 p.77	0.4136	CS106L #05 p.77	0.4306	CS107 #05 p.26	3.7230
5 位	CS143 #12 p.19	0.4035	CS166 #06 p.435	0.4302	CS107 #09 p.98	2.9638

B.6 pointer

表 188: 被験者が作成した **pointer** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS107 #05 p.20	CS107 #05 p.26	CS107 #05 p.25	CS106L #05 p.77	CS106L #05 p.64
B	CS107 #05 p.26	CS107 #05 p.25	CS107 #05 p.20	CS106L #05 p.64	CS106B #17 p.32
C	CS166 #06 p.435	CS166 #06 p.424	CS107 #09 p.98	CS106L #05 p.77	CS106L #05 p.2
D	CS107 #05 p.20	CS107 #05 p.26	CS107 #05 p.25	CS106L #05 p.64	CS106L #05 p.77
E	CS106L #05 p.2	CS143 #17 p.28	CS107 #05 p.26	CS107 #09 p.98	CS106L #05 p.77
F	CS166 #06 p.435	CS107 #05 p.26	CS107 #05 p.25	CS143 #17 p.28	CS107 #09 p.98
G	CS143 #17 p.28	CS107 #05 p.26	CS107 #05 p.25	CS106B #17 p.32	CS166 #06 p.424
H	CS107 #05 p.25	CS106B #17 p.32	CS166 #06 p.435	CS166 #06 p.424	CS107 #09 p.98
I	CS107 #05 p.26	CS107 #05 p.25	CS107 #05 p.20	CS107 #09 p.98	CS166 #06 p.435
J	CS107 #05 p.20	CS107 #05 p.25	CS107 #05 p.26	CS106B #17 p.32	CS106L #05 p.64

表 189: **pointer** の被験者ごとの Q-measure

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.1322	0.0633	0.8318
B	0.1033	0.1052	0.7336
C	0.5510	0.4674	0.040
D	0.1455	0.0422	0.8318
E	0.5579	0.5579	0.2866
F	0.0	0.3157	0.4393
G	0.0	0.2267	0.5227
H	0.2105	0.4064	0.3357
I	0.1052	0.1578	0.2105
J	0.0688	0.1455	0.2011
平均	0.1874	0.2488	0.4433

表 190: **pointer** で得られたコメント

被験者	コメント
A	配列とポインタについて学んでおり，ポインタについての復習ができるため．また，ポインタに似ているものや他の例を閲覧することで，覚え間違いの確認や，今後の応用例について学べるため．
B	表示されるポインタの値について載っているから
D	ポインタについての解説やそれに近そうなものを選んだから。
E	ポインタと配列の関係性を説明してる文が最初はほしいと思ったから
F	ポインタのプログラムが書かれている方がプログラムを書いてみることで理解ができると思うので、プログラムが書かれているものの方が優先度が高いです。
G	よくわからない順から選択していきました
H	プログラムを実行してポインタを理解したいため
I	内容の関連性からです。
J	現在のスライドに対して、より関連度の高く理解に役立つようなスライドを表示してほしいと考えるため。

表 191: **tree** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS166 #06 p.285	1.1260	CS149 #13 p.30	0.9147	CS224N #17 p.50	9.7129
2 位	CS161 #06 p.125	0.9381	CS166 #01 p.60	0.7841	CS161 #16 p.11	6.2111
3 位	CS161 #07 p.49	0.7716	CS161 #16 p.11	0.6902	CS166 #06 p.285	5.6504
4 位	CS166 #17 p.3	0.7664	CS161 #06 p.125	0.5853	CS166 #17 p.3	4.9316
5 位	CS149 #13 p.30	0.7422	CS166 #06 p.285	0.5850	CS166 #05 p.84	4.2281

B.7 tree

表 192: 被験者が作成した **tree** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS166 #05 p.84	CS161 #16 p.11	CS166 #01 p.60	CS224N #17 p.50	CS149 #13 p.30
B	CS166 #05 p.84	CS166 #01 p.60	CS149 #13 p.30	CS166 #06 p.285	CS166 #17 p.3
C	CS166 #06 p.285	CS161 #06 p.125	CS166 #05 p.84	CS166 #17 p.3	CS166 #01 p.60
D	CS166 #01 p.60	CS166 #05 p.84	CS149 #13 p.30	CS224N #17 p.50	CS161 #16 p.11
E	CS166 #05 p.84	CS166 #01 p.60	CS224N #17 p.50	CS161 #16 p.11	CS166 #06 p.285
F	CS166 #05 p.84	CS166 #01 p.60	CS161 #16 p.11	CS224N #17 p.50	CS149 #13 p.30
G	CS166 #05 p.84	CS166 #01 p.60	CS224N #17 p.50	CS161 #16 p.11	CS149 #13 p.30
H	CS166 #01 p.60	CS166 #06 p.285	CS161 #06 p.125	CS166 #05 p.84	CS166 #17 p.3
I	CS166 #01 p.60	CS166 #05 p.84	CS161 #06 p.11	CS224N #17 p.50	CS161 #07 p.49
J	CS161 #07 p.49	CS166 #01 p.60	CS161 #16 p.11	CS224N #17 p.50	CS166 #05 p.84

表 193: **tree** の順位ごとの Q-measure の値

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.020	0.3824	0.3178
B	0.3401	0.6369	0.2055
C	0.5790	0.1652	0.310
D	0.040	0.6595	0.3078
E	0.1719	0.2831	0.4727
F	0.020	0.4350	0.3178
G	0.020	0.4350	0.4038
H	0.4720	0.3657	0.2444
I	0.0	0.0422	0.083
J	0.0	0.020	0.030
平均	0.1663	0.3425	0.2693

表 194: **tree** で得られたコメント

被験者	コメント
A	二分木について学んでおり，他の資料での二分木についての説明を同時に閲覧することで，理解を深められるから．二分木の他の例について学ぶことで，興味と理解を深められるから．
B	図と一緒に見ることでイメージしやすいから
D	関連性がありそう且つ中身のありそうなものを選んだから。
E	ツリーの図が欲しかったから
F	Binary Tree について知りたいのでプログラムよりも図で表されているものの方がイメージがしやすく、プログラムは図よりも優先度は低かった。
G	図と説明文があるやつを優先的に選んだ
H	二分木を図として知りたいから
I	説明の見やすさ等からです。
J	学習中のスライドに対して、より関連度が高くより易しいスライドが表示されると、理解の形成に役立ちそうだと考えたため。

表 195: **hash** のランキング

	通常の C-TF-IDF		正規化した C-TF-IDF		重み付けした C-TF-IDF	
	Page	Score	Page	Score	Page	Score
1 位	CS161 #07 p.121	1.6127	CS106B #27 p.2	0.8763	CS166 #13 p.18	10.5168
2 位	CS106B #27 p.2	0.9924	CS161 #07 p.121	0.7855	CS166 #08 p.7	8.7048
3 位	CS106B #27 p.60	0.7793	CS166 #07 p.4	0.6898	CS166 #07 p.13	8.3551
4 位	CS161 #08 p.3	0.7707	CS161 #08 p.3	0.6822	CS106B #27 p.112	6.4578
5 位	CS166 #07 p.4	0.7076	CS166 #07 p.6	0.6282	CS161 #08 p.93	5.5120

B.8 hash

表 196: 被験者が作成した **hash** のランキング

被験者	第 1 位	第 2 位	第 3 位	第 4 位	第 5 位
A	CS166 #08 p.7	CS166 #07 p.6	CS166 #08 p.7	CS166 #13 p.18	CS161 #08 p.93
B	CS166 #08 p.7	CS166 #13 p.18	CS166 #08 p.7	CS166 #07 p.4	CS166 #07 p.6
C	CS106B #27 p.112	CS166 #13 p.18	CS166 #07 p.13	CS161 #08 p.93	CS106B #27 p.60
D	CS166 #08 p.7	CS166 #08 p.7	CS166 #07 p.6	CS166 #07 p.6	CS106B #27 p.112
E	CS166 #08 p.7	CS166 #07 p.6	CS166 #08 p.7	CS166 #13 p.18	CS106B #27 p.60
F	CS166 #08 p.7	CS166 #08 p.7	CS166 #13 p.18	CS166 #07 p.4	CS166 #07 p.6
G	CS166 #08 p.7	CS166 #08 p.7	CS106B #27 p.2	CS166 #13 p.18	CS166 #07 p.4
H	CS106B #27 p.2	CS166 #07 p.13	CS106B #27 p.112	CS166 #13 p.18	CS161 #08 p.93
I	CS106B #27 p.112	CS106B #27 p.60	CS166 #07 p.13	CS166 #13 p.18	CS1616 #08 p.7
J	CS166 #13 p.18	CS166 #07 p.13	CS161 #08 p.7	CS161 #08 p.3	CS166 #07 p.6

表 197: **hash** の順位ごとの Q-measure の値

被験者	通常の C-TF-IDF	正規化した C-TF-IDF	重み付けした C-TF-IDF
A	0.0	0.050	0.7114
B	0.03	0.1233	0.8469
C	0.0688	0.0	0.6942
D	0.020	0.0988	0.8136
E	0.0688	0.050	0.6750
F	0.020	0.0688	0.8136
G	0.6124	0.6124	0.8210
H	0.3157	0.5157	0.5145
I	0.1722	0.0	0.6275
J	0.0	0.020	0.8984
平均	0.085	0.1184	0.7237

表 198: **hash** で得られたコメント

被験者	コメント
A	ハッシュマップを学んでいるときにハッシュについての資料を閲覧することで, ハッシュについての復習ができるため.
B	ハッシュテーブルの機能について載っているから。
D	関連性がありそうだと思ったから。
E	これについてはよくわからなかったです
F	関連するものを優先度を同じぐらいにした。やはり図がある方が優先度が高い感じがする。
G	よくわからないものから決めていきました。
H	ハッシュについて知りたいから
I	単純明快であったためです。
J	学習中のスライドに加えて特定の 1 枚のスライドが提示されるとしたら、ある程度内容が詰まったスライドのほうが学習に役立つかもしれないと考えたため。