2ZJ-06

プログラミング学習計画補助システムに向けた用語関係可視化の試み

北村利夫† 土肥紳一‡

東京電機大学システムデザイン工学研究科† 東京電機大学システムデザイン工学部‡

1. はじめに

学習プロセスを重視した学習計画は,学習分野の同定,目標の明瞭化，また以降の学習への再利用などの観点から学習をスムーズにし，教育の研究において注目すべき内容である．学習者の習熟度から学ぶべきプロセスをまとめることで，学習者は直近の課題から最終目標を明確にすることができる．そのため教育者が分かりやすい教材を作る場合のみならず，学習者が自身の学びを管理するためにも利用が見込める．本研究ではプログラミング教育の推進のため，このような学習計画の作成を容易にする学習計画補助システムを開発する．特に教科に触れていない学習者が独力で計画をすることは難しく，補助が必要な点である．今回多くの学習者が学んでいる内容として，大学の講義で受講者がどのような順序で学んでいるか自然言語処理を用いてシラバスから用語関係の可視化を試みた．その結果について述べる．

2. データ分析

2.1 対象データ

教育現場で学ばれている内容から学習者がどのような順序で学んでいるかを理解するため，本学の 2022 年に東京千住キャンパスで行われた授業のシラバスデータからプログラミング教育に関する 297 科目を対象として内容を抽出した．講義データのうち，講義内容に特に触れたデータとして，”授業名”,” 目的概要”,” 達成目標”,” テーマ・内容” を抽出した．

2.2 Word2Vec

Word2Vec は Tomas Mikolov らによって提案された文書表現手法である．講義の関係を学習する上でこの手法がふさわしいとされる理由は，単語ベクトル表現の違いにある．従来の単語ベクトル表現である one-hot 表現では，単語 1 つに次元を当てはめ表現する．一方 Word2Vec では分散表現を用いて，確率分布を介した重みの畳み込みによって単語をベクトル空間上に配置する．ある単語に表現される概念は他の単語との共通点や関連と結びつけられ表現するため，異なる概念を表したベクトルを計算によって表現することができる[1]．この表現方法によってより低次元に，正確な関係性を表現することができるため，シラバスに出現する単語から講義用語間の類似性を表現することができる. オープンソースライブラリである Gensim ライブラリから Word2Vec を利用した. 集めたデータの大きさから次元数を 150, 3 回以上登場する単語に設定し学習を行った．

2.3 データ補完

講義単語を学習する上で，多くの文書データが必要である．集められたシラバスは講義単語の学習に不十分であった．そこで単語学習に用いるデータには，講義を表す特徴的な単語を検索エンジンに通すことで汎化し補完することとした．特徴単語の抽出には単語の順位付け手法である OkapiBM25 を用いた．導出された特徴単語の周辺語取得には，検索クエリを用い関連した検索結果を取得する Google Custom Search API を利用した．

2.4 OkapiBM25

OkapiBM25 は文章中に見られる単語の重要度を評価する指標である TF-IDF 式に補正値をつけた式である．その文章を特徴づける単語に高い値をつける．TF-IDF は，ある単語の出現回数を示し(1)式で表される TF(Term-Frequency)と, ある単語の全体の文章中における出現頻度を示し，(2)式で表される IDF(Inverse-Document-Frequency)の積によって表される．ある文章にのみよく出現する単語が高い値になる．

$$tf(t_i, d_j) = \frac{文書\mathbf{d}_j内の単語\mathbf{t}_i の出現回数}{文書\mathbf{d}_j の全単語の出現回数の和} = \frac{f(t_i,d_j)}{\sum_{t_k \in d_j} f(t_k,d_j)} \quad (1)$$

$$idf(t_i) = \log \frac{総文書数}{単語\mathbf{t}_i が出現する文書数} = \log \frac{N}{df(t_i)} + 1 \quad (2)$$

TF-IDF の問題点として長い文章には大きな値に

An attempt at terminology relation visualization for a programming learning planning

† Rio Kitamura. Graduate School of System Design Engineering, Tokyo Denki University.

‡ Shinichi Dohi. School of System Design and Technology, Tokyo Denki University.

なりやすい．シラバスでは講義形式によって長さに違いがあることが想定されるため，この問題点を注視し OkapiBM25 を採用することにした．OkapiBM25 の値は(3)式で表される．

$$\mathrm{score}(t_i, d_j) = idf(t_i)\left(\frac{tf(t_i, d_j)(k+1)}{tf(t_i, d_j) + k_1\left(1-b+b\frac{dl(d_j)}{avgdl}\right)}\right) \quad (3)$$

avgdl は全文章の平均単語数, dl(d_j)は文章長を表し，この 2 つの値によって文書長に補正を与える．また定数 b が 0.75, k は 1.2 から 2 が最良の値とされ[2], k を 1.2 として特徴単語を抽出した．授業ごとに 2 つまでの特徴単語を抽出し，上位 150 単語を検索用単語として用いた．こうして選出された特徴語から検索エンジンを介したテキストを抽出し，データ補完を行った．これにより，学習結果は単語数が 3631 語から 5547 語まで増加した．

3. 結果

Wikipedia によって提供されている日本語 Wikipedia 全文データを含んだ Wikipedia コーパスと今回用意したシラバス+補完語コーパスを用いてプログラミングを表す単語の類似語を算出した．Wikipedia コーパスを表 1, シラバス+補完語コーパスを表 2 に示す．

表 1 Wikipedia の”プログラミング”類似語

類似語	近似値
コンピューター	1.364
アプリケーション	1.211
BASIC	1.146
ソフトウェア	1.140
コミュニケーション	1.076

表 2 シラバス+補完語の”プログラミング”類似語

類似語	近似値
オブジェクト指向	0.833
Java	0.785
アルゴリズム	0.778
データ構造	0.770
C 言語	0.743

導出された単語を比較すると，Wikipedia コーパスではプログラミングという言葉を説明する時に用いられるような単語や、エンジニア同士の会話ツールとしての意味を持つBASICやコミュニケーションといった単語が並んだ. 一方でシラバス+補完語コーパスはプログラミングに属する代表的な言語やプログラミングの書き方や考え方を表す概念など，該当した専門的な単語が並び，講義概念を表す単語の関係をより学習できていることが分かった．

4. まとめと今後の展望

Word2Vec を用いてシラバス+補完語コーパスから専門的な言語関係を学習，プログラミングに関連した特徴語の類似語を可視化した．150 次元で表された単語ベクトルを主成分分析によって 2 次元に変換してあり，座標空間上の点距離が関連性を表す．こうしてプロットされた内容から分野ごとの関連性が確認できた。図 1 では確認できた特徴語群の点外側を結ぶことで学習分野を示す。

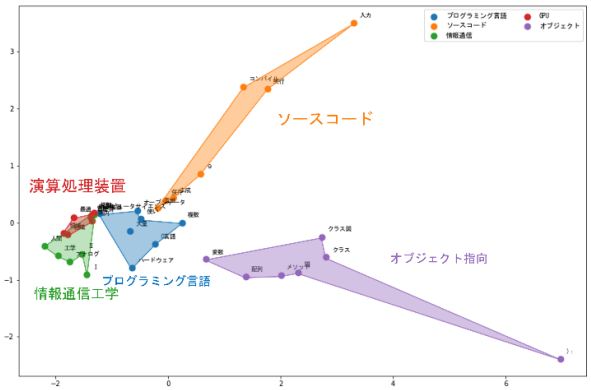

図 1 特徴語から識別される学習分野

情報通信工学分野に属する”情報通信”の類似語の分布範囲(図中緑)には演算処理装置の類似語が集中して分布している(図中赤). よって情報通信工学について学ぶとき，学習者は演算処理装置についての知識を保有していれば学びやすいと考えられる．現在の講義単語学習ではこうした単純な概念の順序関係を明瞭化することができるが，より複雑な順序関係を判断するためには講義ごとの順序や，単語の重要度，難易度を学習させる必要がある．

参照文献

[1] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, arXiv:1301.3781, 2013.

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, An Introduction to Information Retrieval, Cambridge University Press, pp.232-233, 2008.