作問学習における問題難易度の自動推定システム

Automatic Difficulty Estimation of Questions created by Students in Question-posing Learning

原山 一輝[*1], 山岸 芳夫[*2]
Kazuki HARAYAMA[*1], Yoshio YAMAGISHI[*2]
[*1] 金沢工業大学大学院工学研究科
[*1]Graduate School of Technology, Kanazwa Institute of Technology
[*2] 金沢工業大学 メディア情報学科
[*2]Department of Media-Informatics, Kanazawa Institute of Technology
Email: yamagisi@neptune.kanazawa-it.ac.jp

あらまし：本研究では、非常に教育効果が高いとされる作問学習において、学習者が作成した問題の難易度を自動的に推定するシステムの開発、検証を行った。本学で実施された「プログラミング基礎」科目において、過去三年間に作成された約 300 本の問題文をベクトル化し、一つ一つの問題に対し三段階の難易度を割り振り、これらの学習データをナイーブベイズ(NB)、サポートベクターマシン(SV)、ニューラルネットワーク(NN)の分類アルゴリズムを用いて学習させた。全データの内 80%を学習、20%をテストデータに用いて適合率と再現率を求めた結果、NB と NN はほぼ同じ結果だったが、SV は高難易度の問題を全く識別できなかったことが分かった。
キーワード：機械学習、文書分類、Python, scikit-learn

1. はじめに

作問学習は学習者が作成した問題を他の学習者に回答させることで、学習者同士で互いに教え合いを促す手法である。この学習手法を用いた科目は、既に創価大学などで導入されており、教育的に大きな成果を挙げている(1)。良い問題を作成するためには、その科目を理解していることが必要なため、問題の作成を行うだけでも教育効果があると考えられる。さらに、学習者が多ければ多い程問題の難易度がバラエティに富んだものになるため、アダプティブ・ラーニングを行う上でも大きな利点になる。

しかし、作問学習では大量に問題が作成されるため、問題の難易度を判定するにあたり、非常に労力と時間がかかることになる。もちろん学習者が作問同様自ら、もしくは互いに問題の難易度を評価するのであれば労力も分散できるが、やはり学習者による評価には妥当性に不安が残る。

そこで我々は、問題の難易度を機械学習によって自動的に推定することを考えた。これにより、学習者が作成した問題の難易度を、労力をかけることなく瞬時に判別することが可能になり、作問学習の有用性がさらに向上すると考えられる。

2. 先行事例

機械学習による学生の成果物からの成績推定の先行事例として、塚原と山岸は 2016 年に「ミニッツペーパーによる学生の成績予測」において、機械学習によって学生が授業終了後に 1〜2 分程度の時間で書けるミニッツペーパーから、対象の生徒最終成績を予測するシステムの構築を行っている(2)。この研究では最終的にミニッツペーパーによる成績の予測精度は 40%程度にとどまっており、期待していた予測精度に満たなかった。

このように低い予測精度になったのは学習用の教師データが不足していたことと、ミニッツペーパーの内容と成績の関連性が想定していたよりも低かったから、と考えられる。本研究で予測するのは問題とその難易度であり、ミニッツペーパーと成績の関係よりも関連性の深いものになっていると考えられ、精度の高い予測が期待できる。

3. 本研究の概要

3.1 実験環境

本研究は機械学習用ライブラリの scikit-learn 0.18.1 (3)、形態素解析ライブラリの Janome 0.3.5 (4)、文書をベクトルに変換するためのツールとして gensim 3.1.0 (5)を利用するため、Python を用いて開発を行っている。

3.2 実験方法

学習およびテスト用データとして 2017 年､16 年、15 年の過去三年間の内に本学で実施された「プログラミング基礎」において作成された問題合計 328 問を用いる。しかし、問題文のままでは学習用データとして利用できないため Janome と gensim で辞書を作成した後に、それを用いて文書をベクトル化し、教師用データとして用いた。この時、問題文中に含まれるソースコードが形態素解析で正常に分割できないことが予想されていたため、一部の C 言語の命令を名詞としてユーザー定義辞書に追加したものも一緒に用意した。ラベルには問題が実際に出題された際の平均点に応じて Easy、Normal、Difficult の三段階の難易度を割り振った。それぞれの問題数は 126,130,72 となる。

分類器アルゴリズムには SciKitLearn に搭載されているナイーブベイズ(NB)、サポートベクターマシン(SV)、ニューラルネットワーク(NN)の三種類のアルゴリズムを用いた。NN と SV のそれぞれのアルゴリズムでは、ハイパーパラメータを算出するためグリッドサーチを行った。これら一連のシステムのブロックダイヤグラムを図 1 に示す。

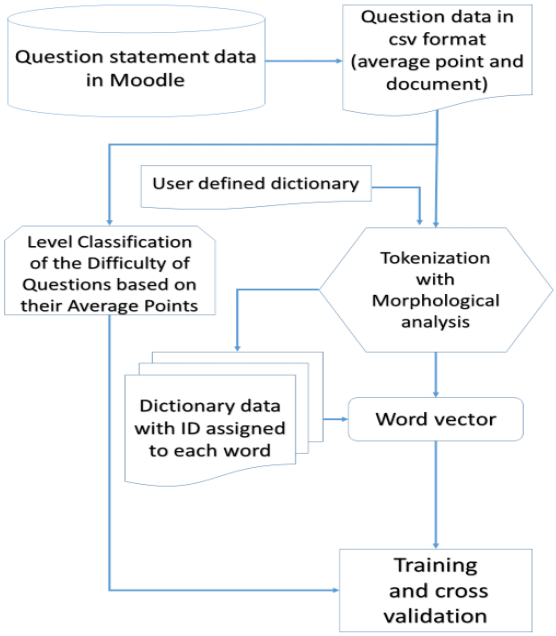

図 1 ブロックダイヤグラム

4. 結果、考察

4.1 実験結果

それぞれの分類器で全データの 80%を学習、20%をテストデータとして適合率（該当するラベルを識別した結果の内、正しくそのラベルを識別できたデータの割合。精度と同義）、再現率（該当するラベルの付いた全てのデータの内、正しくそのラベルを識別できたデータの割合）および F 値（適合率と再現率の調和平均）を求めた。結果を表 1～3 に示す。

表 1 ナイーブベイズ(NB)の結果

難易度	適合率	再現率	F 値
Easy	0.46	0.62	0.52
Normal	0.53	0.35	0.42
Difficult	0.36	0.36	0.36
平均/合計	0.46	0.45	0.45

表 2 サポートベクターマシン(SV)の結果

難易度	適合率	再現率	F 値
Easy	0.44	0.77	0.56
Normal	0.41	0.27	0.33
Difficult	0	0	0
平均/合計	0.34	0.41	0.35

表 3 ニューラルネット(NN)の結果

難易度	適合率	再現率	F 値
Easy	0.49	0.65	0.56
Normal	0.47	0.31	0.37
Difficult	0.36	0.36	0.36
平均/合計	0.45	0.45	0.44

いずれも適合率すなわち精度は 50%を超えない結果となった。最も高い精度だったアルゴリズムは NB だが、誤差の範囲と思われる。SV は”Easy”の再現率は高いが、”Difficult”の難易度の問題を識別できていないことがわかった。

4.2 考察

三種類の分類器とも、先行研究の結果はやや上回ったものの、決して精度は高いとは言えない結果となった。この結果になった要因としては、やはりまだデータが不足していることが考えられる。また、ハイパーパラメータやユーザー定義辞書の構成も最適化されているか再検討する必要があると思われる。

5. まとめ

今回利用した三種類の学習モデルは、そのどれもが実用には適さない予測精度となった。しかし、同等数のデータを用いた先行研究での精度を上回っているため、やはり機械学習ではミニッツペーパーより問題の方が特徴を抽出しやすいのではないかと考えられる。今後は他のクラスの同科目でも同様に作問学習を行うことで、学習およびテスト用のデータを増やすことを考えている。

また、本研究では「畳み込みニューラルネット(CNN)」や「再帰型ニューラルネット(RNN)」といった、いわゆる「深層学習」といったアルゴリズムを採用していない。このような新しい機械学習アルゴリズムを用いれば、精度が高まるのではないか、と期待できる。

参考文献

(1) 学生が協調的に作問可能な WBT システム http://wbt1.soka.ac.jp/index.php?id=1
(2) 塚原 大揮・山岸 芳夫、ミニッツペーパーによる学生の成績予測、平成 28 年度電気関係学会北陸支部連合大会講演論文集
(3) scikit-learn: machine learning in Python — scikit-learn 0.19.1 documentation http://scikit-learn.org/stable/
(4) Welcome to janome's documentation! (Japanese) — Janome v0.3 documentation (ja) http://mocobeta.github.io/janome/
(5) gensim: Topic modelling for humans https://radimrehurek.com/gensim/