

プログラミング教育に向けたオンライン レポジトリを活用した英語学習用コーパスの作成

沼田 哲史*1

*1 大阪電気通信大学 総合情報学部

Creating English Learner's Corpus using Online Repositories for Programming Education

Satoshi Numata*1

*1 Faculty of Information Science and Arts, Osaka Electro-Communication University

In programming education, it is important to mention the importance of English, because grammars of most programming languages are based on English. For example, a function name with a verb and its object will be helpful for programmers to guess how it behaves even without any documentation comments. In this study, I propose a method to create a corpus of English words that can suggest better English words for programming entities using programming repositories published online.

キーワード: プログラミング教育, 名前付け, 英語

1. はじめに

プログラミング教育において、将来活躍することが期待できる人材を輩出するために、アルゴリズムの設計や応用ができるように指導することと合わせて、可読性の高いプログラムコードが記述できるように学生を指導することは重要である。可読性の高さにつながる要素は様々であるが、文献(1)によれば、その要素は大きく3つに分類できるとされている。すなわち「1. 良い名前を付けること」「2. 良いコメントを書くこと」「3. コードの書式を整えること」である。2番目のコメントを書くことに関しては、変数名や関数名が内容をよく表すものになっていればコメントを省略することができるため、その半分は1番目の名前付けが重要であるということに帰着すると言える。

そこで本稿では、学生が英語を用いた良い名前付けを学ぶことをサポートするために、オンライン・レポジトリで公開されている多数の実用的なプロジェクトで使われている名前付けの実例を自動的に取得する手法を提案する。そしてそこから得られたデータをプログラミング教育に応用する手段についても論じる。

2. プログラミング教育と英語

2.1 プログラミングにおける英語の必要性

多くのプログラミング言語が英語の文法を前提として構成されているため、プログラミングにおいて、英語の知識は重要である。例として、ゲームのキャラクターがジャンプ中のみ行う処理を書くことを考える。ジャンプしているかどうかを判定する関数名を「Func()」「Tobu()」「IsJumping()」と変えてみると、条件判定を行うコードはそれぞれ次のようになる。

```
if (Func()) { /* ジャンプ中の処理 */ }  
if (Tobu()) { /* ジャンプ中の処理 */ }  
if (IsJumping()) { /* ジャンプ中の処理 */ }
```

このように比較すると、意味のない関数名や、日本語のローマ字表記で「飛ぶ」という事象を表した関数名に対して、英語で「ジャンプ中である」ことを示すbe-動詞+現在進行形で名前を付けたものは、明らかに表現力が高く、また「if」などの予約語と合わせて、通常の英語としてもある程度読み下すことができることが分かる。プログラミング教育において、英語の知

識とその応用能力は無視することができない。

2.2 カリキュラム編成の実例

筆者の所属する大学の学科*1 では、日常的な英会話などの技能を身につけるための英語の授業は総合科目としてまとめられている。工学的な文章の読み書きができるようにすることを目的とした専門科目が 1 科目用意されているが、可読性の高いプログラムのコードを書くことを目的とした英語の授業は用意されていない。そしてプログラミングの演習授業内では、アルゴリズムの理解やファイルの入出力といった事項を理解し、あるいはネットワーク・プログラミングやバージョン管理といった技術を学ぶ時間が必要であり、プログラミングに特化した英語の知識の授受を行う時間を確保することは難しい。

そのため、プログラミングにおいて活用できる英語の知識を身につけさせるためには、本稿で提案するようなツールの準備と利用が必要となってくる。

2.3 事例研究

プログラミングに活用できる名前付けのためのツールとして、文献(2)のオンラインツールが挙げられる。これは日本語の文章を入力すると、プログラミングの要素として使いやすい簡潔な英語の文章が生成されるというものである。例として「ファイルを生成する」と入力すると「generate_file」という翻訳案が提示される。しかしながら、内部的な処理としては機械翻訳が使われているらしく、例えば「ジャンプ中かどうか」と入力しても「is_during_jump」という翻訳案が提示され、時制の正確さに欠けたり、不要な単語が交じるなどの問題がある。正確さや翻訳の速度を考えれば、学生本人が英語を勉強することがやはり必要であり、それをサポートすることが主題となる。

2.4 プログラミング用コーパスの自動生成

上述のことから、筆者は文献(3)のシステムを提案している。このシステムでは、オンラインのソースコード管理システムである GitHub で公開されていて一定数以上の評価が付けられた C# のプロジェクトに対して、変数や関数といった要素に使われている英単語を自動的に取得してコーパスを生成する。

3. プログラミング用コーパスの利用

3.1 実際の命名規則の確認

前述のコーパスを活用することで、多くのユーザーによって現在進行形でアクティブに使われているプロジェクトの中で、どのような英単語が、どのような組み合わせで使われているかを確認することができる。変数の型や関数の型などに着目して単語を分類することにより、プログラミングの授業内でそれらの概念を説明するのと同時に、合わせて勉強しておくべき英語表現を実データとともに提示できる。

3.2 プログラミング用エディタの拡張機能

広く使われているプログラミング用のエディタには、拡張機能を追加できるものが多い。前述のコーパスから、変数や関数の型と、英単語間の関係などをあらかじめ計算して保持しておくことで、エディタ上で変数宣言や関数宣言の入力が行われる際に、それらのデータを元にして、変数名や関数名を予測入力する機能を実装することが考えられる。

4. おわりに

本稿では、プログラミング教育には英語の知識が必要であることを示し、オンライン上で公開されている膨大な量のコードからプログラミング用のコーパスを作成して、それをプログラミング教育に活用する方策を示した。今後、統計データのまとめやツールを完成させた後、学部 1 年生および 2 年生の授業で活用し、その結果を報告したいと考える。

参考文献

- (1) Boswell, D., Foucher, T.(著), 角 征典(訳): “リーダブルコード—より良いコードを書くためのシンプルで実践的なテクニック”, オライリージャパン (2012)
- (2) プログラマーのためのネーミング辞書 | codic <https://codic.jp> (2019 年 2 月 1 日確認)
- (3) Satoshi Numata: “Creating C# Programming Corpus using ANTLR4 for Non-Native English-Speaking Students”, International Journal of Education and Research, Vol.6, No.11, pp.115-126 (2018)