深層学習を用いたプログラム自動翻訳アプリの開発

学籍番号：184223　氏名：園田航一　　指導教官：工藤達郎

概要　本研究は、プログラミング初学者のハードルとなるプログラム読解力の問題に着目し、プログラムもある種の言語であることから翻訳することで学習を支援しようという試みである。本研究では、Google翻訳に使われている深層学習モデル「Transformer」を元にC#プログラムのソースコードを日本語に翻訳するアプリの開発を目指す。すでにQiitaApiを用いて学習データを収集するWebスクレイピングの方法を確立しており、Transformerも構築済みである。

キーワード：機械翻訳、プログラム読解力、深層学習、Transformer、Webスクレイピング

１．研究の背景

後輩にプログラミングの指導を行う際、一度各自で調べさせているが、どうしてもプログラムの意味を読み解けず行き詰ってしまう学生が多い。つまりプログラミング初学者にとっての壁とは、調べてもプログラムが読めないことである。また、ソースコードに対しコメント文を自動生成する翻訳機は実現されている[1]が、深層学習が用いられた事例はない。

２．目的

以上のような背景から本研究では、プログラミング初学者のプログラム読解を支援し学習効率の向上を目的とし、深層学習を用いてプログラムを日本語に翻訳するアプリの開発を進めていく。具体的には、C#のソースコードを入力すると、そのプログラムがどういう挙動をするかのテキスト提示を行うアプリである。また、実証実験の際にはWebページのみの学習と本アプリを併用した場合の学習効果についても検証を行う。

３．研究方法

研究の流れは以下のように行う。

1) Webスクレイパーの構築・データ収集
2) Transformerの構築
3) 収集したデータから説明文を抽出
4) 本研究に沿ったモデルに再構築
5) 学習
6) 学習精度の評価と再構築・再学習
7) 翻訳アプリの制作
8) 実証実験

なお卒業研究Ⅰでは、2)までのプロセスを行った。

４．制作

アプリ開発に用いた環境が以下の表である

A:Webスクレイパー	Windows Form Application
B:深層学習	CUDA Toolkit cuDNN Anaconda3 Jupyter Notebook Tensorflow-gpu2.6.0
開発言語	A: C# , B: Python

１）Webスクレイパーの制作

本研究で構築したWebスクレイパーはURLスクレイパーとページスクレイパーの２ステップに分けられる。まずURLスクレイパーがQiitaApiを使いC#タグのついた記事のURLを抽出しCSVファイルとして書

き出す。その CSV ファイルをページスクレイパー（図 1）に渡すことで記事の中からソースコードと本文を抽出しそれぞれ CSV ファイルとして出力する。

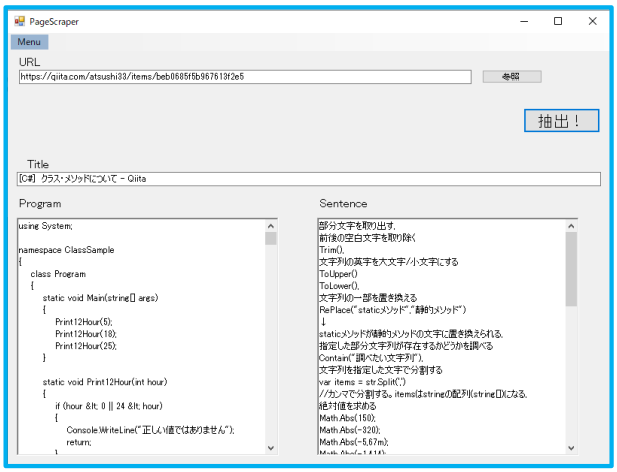

図 1　ページスクレイパー

2）Transformer の構築

Transformer は Google 翻訳にも使われている深層学習モデルである。以前までの Seq2Seq モデルより翻訳精度が高く Attention のみを用いているため構造がシンプルで、多言語間の翻訳にも対応している。Transformer の構築は参考文献[2]のページをもとに進めた。

5．結果と考察

スクレイピングにより、12,100 対のデータを取得したが、中には C#のソースコードを含まないものもあるため、学習に用いるデータを選別する必要がある。また、抽出した本文のデータは、ソースコードの説明文以外の文章も含まれるため、説明文のみを抽出するような機構を取り入れていかなければならない。

Transformer の学習データには、今回はテストとして夏目漱石の「明治座の所感を虚子君に問れて」を用い、同小説内の文章を入力すると次の文章を予測するようにした。106,067 行のデータを 2 日間学習し、268,500 ステップ学習することができた。

図 2 はその正解率をグラフ化したものである。

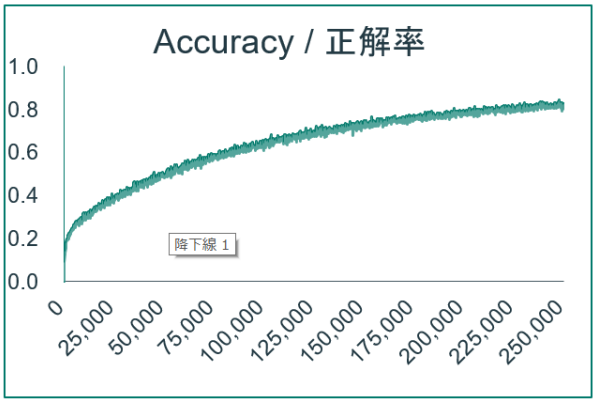

図 2　学習結果

今回は文章予測用の学習データを用意したが、機械翻訳の学習データは原文と翻訳文が 1 組になっている必要があるため、別々に CSV ファイルに出力するのではなく 1 つにまとめる必要があることが分かった。

6．卒業研究 II に向けて

Web スクレイパーには以下の改善を加える。

・ソースコードを含まない記事を省く機構

・参考文献[3]をもとに説明文のみを抽出する機構

・データセットを書き出す機構

上記の改善が終わり次第、自作スクレイパーで取得したデータセットで Transformer の学習と再構築を行い、学習精度の向上を目指す。学習が終わったらアプリ開発に着手し、本アプリの学習効果について実証実験を行う。

参考文献

[1] 機械翻訳を用いた擬似コード生成による学習者支援
https://ahcweb01.naist.jp/papers/conference/2015/201509_JSiSE_Fudaba_1/201509_JSiSE_Fudaba_1.paper.pdf
[2] 作って理解する Transformer / Attention
https://qiita.com/halhorn/items/c91497522be27bde17ce
[3] RNN を用いたテキスト二値分類による用語説明文抽出方法の提案
https://ipsj.ixsq.nii.ac.jp/ej/?action=repository_uri&item_id=196282&file_id=1&file_no=1