深層学習を用いた Web API 仕様書のモデル化方法の提案

2014SE042 加納 辰真　2014SE069 永井 利幸

指導教員 青山 幹雄

1 研究背景/課題

1.1 研究背景

スマートデバイスや，インターネットの普及により Web サービスの利用が増加している．それに伴い，Web API の利用も急増している．これらの Web API の仕様記述文書は統一されていないので，利用者が Web API 仕様の理解や比較が困難となっている．一方，Web API 提供者は保守や管理に労力を要している．

1.2 研究課題

本研究では，上記の背景を踏まえ，以下の2 点を研究課題とする．

(1) 自然言語で書かれた Web API 仕様文書から仕様記述文書のモデルを生成する．

(2) 実際の Web API 仕様文書に適用し，提案方法のモデルの有効性と妥当性を評価する．

2 関連研究

2.1 Web API [7]

Web API は REST 形式に従って，HTTP メソッドを用いて呼び出す API である．

2.2 深層学習 [1]

深層学習は機械学習の一種で独立した最小の計算単位ニューロンをつなげたネットワークを何層にも構成する多層の NN (Neural Network)である． 深層学習フレームワークとして，Chainer [4]や TensorFlow がある．

NN には以下の 2 つが含まれる．

(1) Word2Vec [3][6]

Word2Vec とは，2層の NN から成り，テキスト処理を行う NN である．テキストコーパスを入力すると，コーパスにある単語の特徴量ベクトルを出力する．

(2) CNN (Convolutional Neural Network) [2]

CNN は，順伝播型の NN の 1 つで画像認識等に利用されているモデル．CNN は最小限のデータ前処理しか必要としないように設計された多層パーセプトロンの一様である．

3 アプローチ

アプローチを図 1 に示す．

深層学習によって，人手で扱えない大量の Web API 仕様文書に書かれた自然言語の Web API 仕様記述をモデル化する．そのモデルは構造化された Web API 仕様記述文書である．その文書を用いることによって Web API 利用者が Web API を理解や比較を容易にすることを目指す．

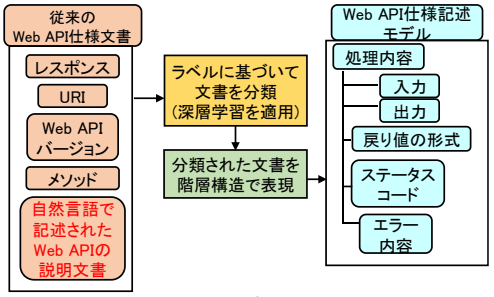

図1 アプローチ

4 深層学習を用いた Web API 仕様書モデル化方法

4.1 開発プロセス

本研究での全体の開発プロセスを図 2 に示す．

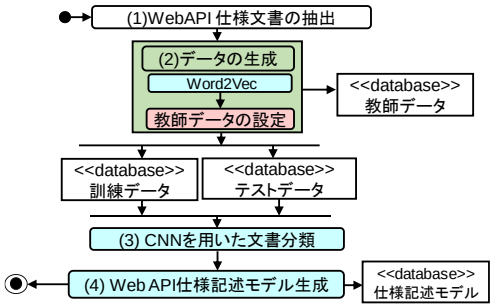

図2 開発プロセス

各プロセスは以下のようになっている．

(1) Web API 仕様文書の抽出

Web 上に公開されている Web API 仕様文書からスクレイピングを用いて，自然言語で書かれた仕様記述文書を抽出する．抽出した文書は自然言語処理を行いやすいようにデータの整形を行う．文書は文章ごとに改行を行う．

(2) コーパスの生成

(1)で得られた Web API 仕様文書を入力データとして，Word2Vec を用いて数値データに変換し，教師データを作成する．

(3) CNN を用いた文章分類

(2)のプロセスで生成された数値データの集まりであるコーパスを訓練データとテストデータに分割する．訓練データは CNN を用いて文章分類を行うことにより仕様に対し関連度の高い文章を抽出する．本研究では，API の名前，Protocol，各カテゴリーにおいて重要な言葉をクラスと定義している．それらに元づきラベルが対応しているので，多クラス分類を行う．

(4) Web API 仕様記述文書モデルの生成

(3)のプロセスで得られた数値データを元に仕様に対して関連度の高い文章を階層構造で表現する．本研究では，その仕様に対して関連度の高い文章を階層構造で表現されたものをモデルと定義する．

5 プロトタイプの実装と実行

5.1 プロトタイプの実装

本研究のプロトタイプのアーキテクチャを図 3 に示す.

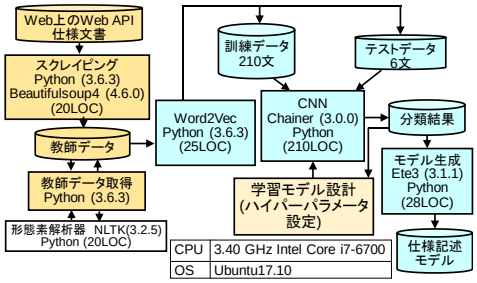

図3 プロトタイプのアーキテクチャ

5.2 プロトタイプの実行

図 2 に示すプロセスに従って次のように実行される.

(1) Web API 仕様記述文書を抽出

Python を使用して, 自然言語で書かれた Web API 仕様文書に対して, スクレイピングを行い, description タグ内の文書を抽出する(図 4).

抽出した文書は自然言語処理を行いやすい形にデータ整形を行う. 行った文書は文章ごとに改行する.

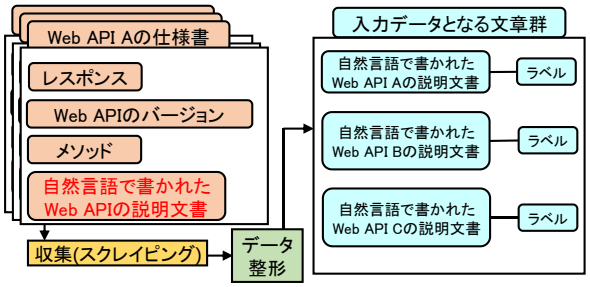

図 4 仕様文書から文章を抽出

(2) 訓練データとテストデータを生成

(1)のプロセスで得られた文章群は Word2Vec を用いることで文章データから数値データに変換する(図 5).

ラベル付けされた文章データを変換されたものが訓練データとなる. ラベルつけされていない文章データがテストデータとなる.

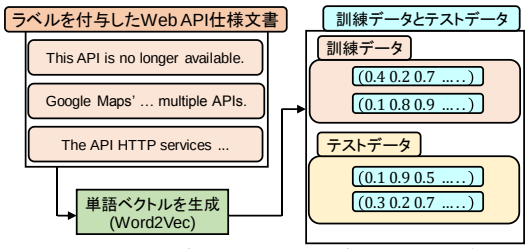

図 5 文章データを数値データに変換

(3) CNN を用いた文章分類

(2)で得られた訓練データをもとに CNN を用いた文章分類を行う. 訓練データを使用して深層学習モデルを学習させる. その深層学習モデルに対してテストデータを用いることで学習モデルの評価を行う.

得られた深層学習モデルに実データを用いてラベル付けを行う(図 6).

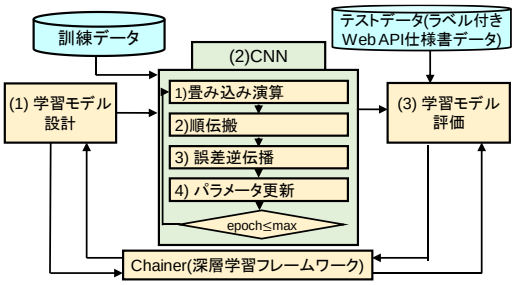

図 6 CNN を用いた文章分類の例

(4) モデルを生成

ラベル付けされた数値データをもとに Python のライブラリを用いて階層構造で表現された Web API 仕様記述文書モデルを生成する(図 7).

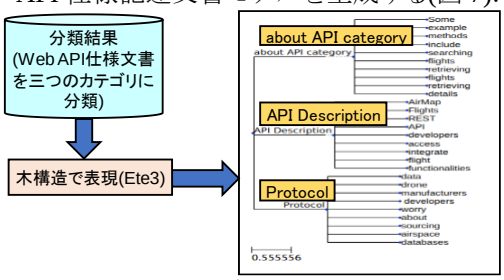

図 7 モデル生成の例

6 ProgrammableWeb 上の Web API 仕様文書への適用

6.1 適用の目的

プロトタイプを作成したのち, ProgrammableWeb に公開されている Web API 仕様文書に適用する.

プロトタイプを適用して, 提案方法の有効性と妥当性を評価する. さらに, 生成されたモデルを分析することで Web API 仕様文書の記述ルールの特性, 共通に使われている表現を発見する.

6.2 データの収集

ProgrammableWeb 上に公開されている Web API 仕様文書データの"Mapping"に関する Web API 仕様を記述したページ 100 件に含まれる文章 210 個と"Social"に関する Web API 仕様を記述したページ 100 件に含まれる文章 324 個を収集した.

その一例を表 1 に示す.

スクレイピング後にデータを自然言語処理しやすい形に整形する. その整形されたデータに対してラベル付けを行う. ラベルと文章が対応している. それらの群をコーパスといい, CNN の訓練データとテストデータとして扱う.

表1 文章データとそのラベルの一例

ラベル	収集データ
1	Some example API methods include searching for flights, retrieving flights, and retrieving flight details. AirMap sources their own data so that drone manufacturers and developers don't have to to worry about sourcing their own airspace databases
2	The AirMap Flights REST API allows developers to access and integrate the flight functionalities of AirMap with other applications
3	It is an open protocol to allow secure authorization in a simple and standard method from web, mobile, and desktop applications

6.3 分析

(1) 分類

Web API 仕様文書のデータを取得したのち，データの整形を行い，ラベル付けを行ったものが訓練データとテストデータとなる．現在のコーパスでは数値データではなく，文字列と数値の組み合わせであるため，これらをベクトル化し，数値データ化する．

CNN を用いた文章分類で，ラベル付けを行った例を以下の表 2，表 3，表 4 に示す．

表 2 入力データと出力データの例 1

	入力データ	想定	結果	差
1	It, is, an, open, protocol, to, allow, secure, authorization, in, a, simple, and, standard, method, from, web, mobile, and, desktop, applications	3	3	無
2	protocol, authorization, web, mobile, applications	3	3	無
3	protocol, web, mobile, applications	3	3	無

表 3 入力データと出力データの例 2

	入力データ	想定	結果	差
1	It, is, geospatial, big, data, made, accessible, using, a, cloud-based	1	3	有
2	geospatial, big, data, made, accessible, cloud-based	1	1	無
3	big, data, made, accessible, cloud-based	1	3	有

表 4 入力データと出力データの例 3

	入力データ	想定	結果	差
1	AirMap, is, an, airspace, technology, firm, based, in, California	2	2	無
2	airspace, technology, firm, based	2	2	無
3	technology, firm, based	2	2	無

本研究では，ストップワードは自然言語処理を行う際に，処理対象外とする単語と定義する．

表 2，表 3，表 4 はそれぞれ以下に対応している．

1) 元の文章そのものの単語群
2) ストップワードと固有名詞を 1)の単語群から除いた単語群
3) 各クラスに対して，コサイン類似度が最も高い単語を 2)の群から除いた単語群

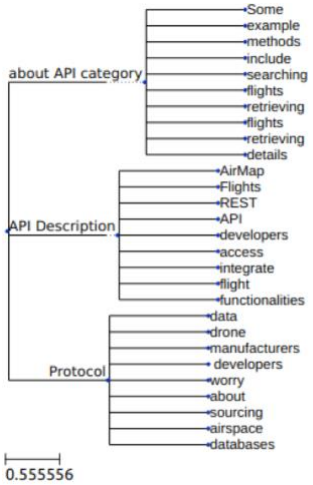

図8 モデルの例

(2) 分析(構造化)

6.2 で得られた訓練データを学習させたモデルに基づいてテストデータを適用させた．テストデータから得られた分類結果を元に生成されたモデルを図 8 に示す．

図 8 は，CNN を用いた文章分類を行ったテストデータを Python のライブラリである Ete3 で構造化したモデルである．

7 適用例の評価

7.1 CNN を用いた文章分類の評価方法

CNN を用いた文章分類の評価を行うために，A と L を評価する．正解率は以下の式 1 で定義する．

$$A = \frac{\text{正解文章の数}}{\text{テストデータの文章の総数}} \quad [1]$$

誤差率は以下の式 2 でランプ関数として定義する．

$$L = \max(0, x) \quad [2]$$

ランプ関数とは，非負整数の値を入力したときはそのままの値を，負の整数が入力されたときは，0 を返す関数である．

7.1 CNN を用いた文章分類の評価結果

本研究では，訓練データとテストデータに対してそれぞれ 100 エポック学習させ，テストを行った．以下の表 5 に 100 エポック時点での訓練データとテストデータの正解率と誤差率を示す．

表5 100 エポック時点での訓練データとテストデータの正解率と誤差率

	正解率	誤差率
訓練データ	0.88	0.26
テストデータ	0.50	2.39

さらに，対数関数を用いて学習のエポック数が増加した時の訓練データとテストデータそれぞれの正解率と誤差率の値の推移を図 9 と図 10 に示す．

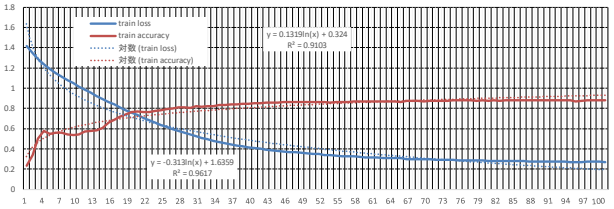

図 9 訓練データの正解率と誤差率

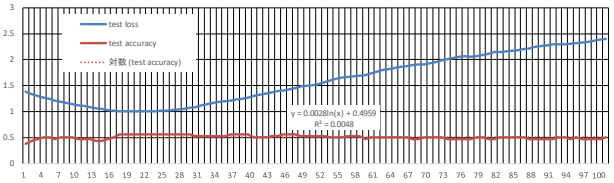

図 10 テストデータの正解率と誤差率

7.3 モデルの評価方法

CNN を用いた文章分類の出力データで得られたモデルを評価する．

従来の Web API 仕様書と比較して評価を行う．

7.4 モデルの評価結果

ProgrammableWeb 上に公開されている自然言語で書かれた Web API 仕様文書をプロトタイプに適用した. その生成されたモデルを図 11 に示す.

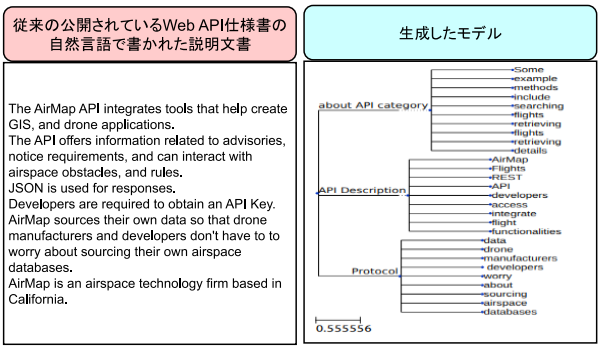

図11 従来との比較

従来と, Web API のカテゴリーについて, Web API の説明, プロトコルの 3 つにカテゴライズしたモデルと比較する. Web API 仕様記述文書で書かれている単語 73 語に対して, モデルでは, 28 語となった. これは説明文章に書かれている接続詞, 名詞などが削除されたからである. よって, 以下の点がモデルの効果と考えられる.

(1) 読解労力の軽減

Web API 仕様文書に含まれる単語数が 73 語から構造化されたモデルでは 28 語に減少したことで仕様文書を読解する労力が軽減された.

(2) 理解容易性の向上

自然言語で記述されている Web API 仕様記述文書を共通に構造化されたことで, 文書の読解をする必要がなくなり, 内容の理解が容易となった.

(3) 異なる Web API 仕様文書の比較容易性の向上

構造化されたモデルにより Web API 仕様記述の項目をカテゴリーごとで比較できるので, 複数の Web API 仕様の比較が容易となった.

8 考察

8.1 CNN を用いた文章分類

本研究で用いた深層学習モデルではテストデータの正解率が 0.50 であった. 訓練データ数とテストデータ数がラベルによって偏りがあり, ストップワードを除かないと想定した結果にならない文章も存在した.

訓練データの数とラベルごとに分けた時にそれぞれが同じ程度になるように調整する必要があると考えられる.

8.2 モデル

モデルの階層構造も以下の 3 つに分けられる. 複数のカテゴリーを検証したが, どのカテゴリーに属していても自然言語で書かれた Web API 仕様文書は 3 つに分けられる.

(1) “About API Category”のカテゴリー

About API Category は, その Web API がどのカテゴリーに当てはまるかを説明した文章もしくは, 単語が属する. ProgrammableWeb では大きなカテゴリーわけで Web API 毎にいくつかのカテゴライズされたラベルがついている. そのカテゴライズされたラベルには含まれないものも Web API 仕様文書にあることが Web API 仕様記述文書モデルを生成することで発見した.

(2) “API description”のカテゴリー

API description は, その Web API がどんなものかを説明した文章, もしくは単語が属する. Web API がどんな形式なのか, 使い方, レスポンスされる URI などがこの API description に当てはまる. 複数の Web API との比較要因となる項目であり, モデル化することで比較が容易になった.

(3) “Protocol”のカテゴリー

Protocol では, その Web API の Protocol を説明した文章, もしくは単語が属する. その Web API に使われているプロトコルや, 使用するプロトコルが当てはまる.

8.3 CNN の適用効果

RNN と CNN を比較し, 文書分類のため非順序データの分類に適する CNN を適用した. 誤差関数は分類が 3 種類以上となるためシグモイド関数は適さず, ランプ関数を用いた. この結果, 100 エポックでの訓練データとテストデータの正解率はそれぞれ 88.3%, 50.0%となった.

9 今後の課題

今後の課題は以下の 3 点である.

(1) 深層学習モデルの正答率, 誤差率の向上

(2) 他の深層学習モデルとの比較

(3) モデルの詳細な分析

10 まとめ

近年, Web API の利用が急増している. これらの Web API 仕様書は異なる記述言語で公開されているため, 利用者が Web API の仕様を比較できず, 選択や検索が困難となっている. 本研究では, 大量の Web API 仕様文書データに対して CNN を用いた文章分類により, 仕様に対し, 関連度の高い文章を分類し, Web API 仕様文書のモデルを生成する方法を提案した.

提案方法のプロトタイプを実装し, Web 上のレジストリに登録されている実際の Web API 仕様文書に適用することで, その有効性と妥当性を評価した.

11 参考文献

[1] 神嶌 敏弘 他, 深層学習, 人工知能学会, 2016.

[2] Y. Kim, Convolutional Neural Networks for Sentence Classification, Proc, of EMNLP 2014, Oct. 2014, pp. 1746-1751.

[3] T. Mikolov, et al., Efficient Estimation of Word Representations in Vector Space, arXiv: 1301.3781v3 [cs.CL], Sep. 2013, pp. 1-12.

[4] Preferred Networks, Chainer, https://chainer.org/.

[5] ProgrammableWeb, https://programmableweb.com/.

[6] X. Rong, Word2vec Parameter Learning Explained, arXiv:1411.2738v4 [cs.CL], Jun. 2016, pp. 1-21.

[7] S. Sohan, et al., SpyREST: Automated RESTful API Documentation Using HTTP Proxy Server, Proc. of ICASE 2015, IEEE/ACM, Aug. 2015, pp. 271-276.