5J-08

深層学習を用いたWeb API仕様書のモデル化方法の提案

永井　利幸†　加納　辰真†　青山　幹雄†
南山大学 理工学部 ソフトウェア工学科†

1. はじめに

Web API の利用が急増しているが，そ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[6]．一方，Web API 提供者は保守や管理に労力を要している．

2. 研究課題

本稿では以下の 2 点を研究課題とする．

(1) 自然言語で記述された Web API 仕様文書から仕様記述モデルを生成する．

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

3. 関連研究

(1) 深層学習

深層学習は多層 NN(Neural Network)を用いた機械学習である．深層学習プラットフォームとして，Chainer[3]などがある．

(2) Word2Vec[2][5]

入力文書から，単語を特徴量ベクトルとして出力する NN である．

(3) CNN (Convolutional Neural Network)[1]

畳み込み演算を行う NN である．

4. アプローチ

自然言語で記述された大量の Web API 仕様記述を CNN でモデル化する．その仕様記述モデルは複数のカテゴリで階層構造化されたものである．その仕様記述モデルを用いることで利用者が Web API の理解や比較を容易にすることを目的とする(図 1)．

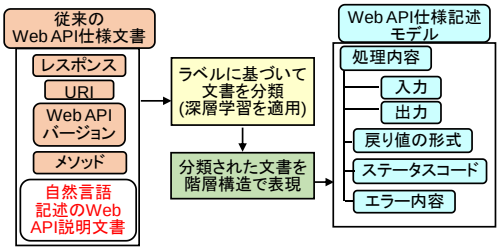

図 1 アプローチ

5. CNN を用いたモデル化システムプロトタイプ

CNN を用いたモデル化システムプロトタイプのアーキテクチャを図 2 に示す．Web 上に公開されている Web API 仕様文書をスクレイピングして，文章ごとに区切る．その文書を Word2Vec で特徴量ベクトル化し，CNN プラットフォームの Chainer を用いて文章を分類する．分類結果を階層構造で表現する．

A Generation Method for Web API Specifications Using the Deep Learning and its Evalution
†Toshiyuki Nagai, Tatsuma Kano,Mikio Aoyama, Department of Software Engineering, Nanzan University.

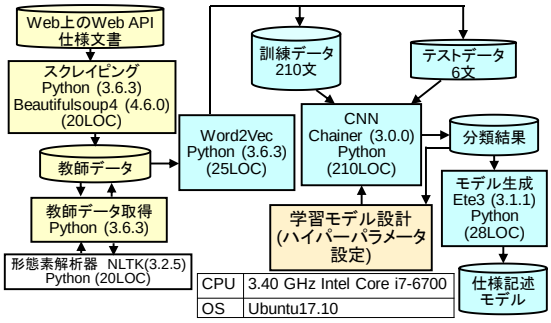

図 2 プロトタイプのアーキテクチャ

6. プロトタイプを用いたモデル化プロセス

プロトタイプを用いたモデル化プロセスを図 3 に示す．

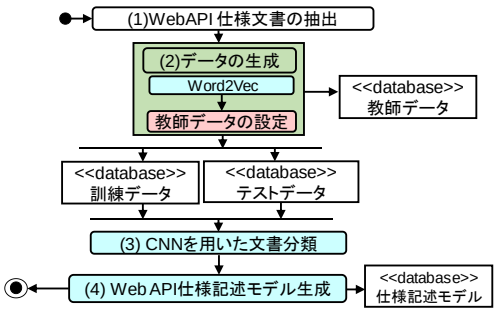

図 3 モデル化プロセス

モデル化プロセスの詳細は以下の通りである．

(1) Web API 仕様文書のスクレイピングによる抽出

Web API 仕様文書から自然言語で記述された文章をスクレイピングする．

(2) コーパスの生成によるモデルの階層構造生成

(1)で得た文章にラベルを付与し，Word2Vec で単語をベクトル化する．ラベルを Web API 仕様記述モデルの階層構造と対応付ける．

(3) CNN を用いた Web API 仕様記述文章の分類

(2)で得た特徴量ベクトルを訓練データとテストデータに分割する．訓練データを用いて CNN の学習モデルを生成する．学習モデルにテストデータを適用し，学習モデルを評価し，学習モデルパラメータを再設定する．

(4) Web API 仕様記述モデルの生成

(3)の分類結果を(2)で生成した階層構造で表現し，仕様記述モデルを生成する．

7. Web 上の Web API 仕様文書への適用

(1) 実際の Web API 仕様文書の収集

ProgrammableWeb[4]上に公開されている Web API 仕様文書データの”Mapping”に関する記述ページ 100 件に含まれる文章 210 個と”Social”

に関する記述ページ 100 件に含まれる文章 324 個を収集した．抽出文章から HTML タグの削除などの整形をした．整形済文章ごとにラベル付けを行う．Word2Vec で単語をベクトル化し，訓練データとテストデータに分割した．

(2) Chainer を用いた Web API 仕様記述文章の分類

(1)で得た訓練データとテストデータを対し，Chainer を用いて文章分類し，ラベル付けた．表 1 の想定と結果は人手によるラベル付けと Chainer によるラベル付けの一部を示す．

表 1 入力データと出力データの例

	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

(3) Web API 仕様記述モデルの生成

(2)の分類結果を(1)の定義に基づく階層構造に割り当て，階層化した Web API 仕様記述モデルを生成した(図 4).

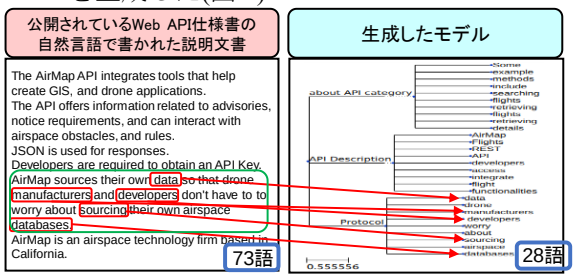

図 4 生成したモデル

8. 評価

(1) 文章分類結果に基づく CNN 学習モデルの評価

訓練データとテストデータに対して 100 エポック学習，テストした．学習モデルを評価するために正解率 A と誤差率 L を次の式で定義する．

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

誤差率はランプ関数(ReLU)で定義する．

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

正解率と誤差率の推移を図 5 と図 6 に示す．

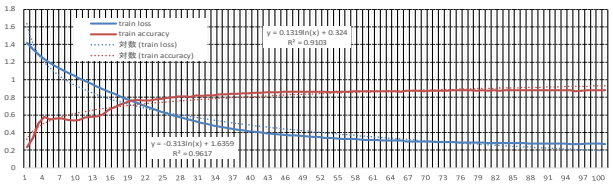

図 5 訓練データの正解率と誤差率

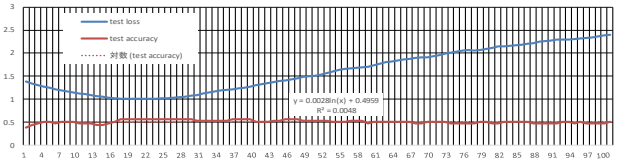

図 6 テストデータの正解率と誤差率

(2) 従来の Web API 仕様文書と生成した仕様記述モデルを比較し評価した．以下の効果を確認した．

1) 読解労力の軽減

構造化によって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

構造化によって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

構造化によって Web API 仕様記述の項目をカテゴリ毎で比較できるので，複数の Web API 仕様の比較が容易となった．

(3) CNN の適用効果

RNN と CNN を比較し，文章分類の学習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. モデルの考察

CNN による文章分類で下記の 3 つのカテゴリを持つことを発見した．さらに，Web API 仕様文書がその中で指定されたカテゴリによらず，CNN を用いて 3 つの共通のカテゴリに適切に分類できた．

(1) “About API Category”カテゴリ

Web API 仕様文書がどのカテゴリに当てはまるかを説明した文章のキーワードが属する．

(2) “API description”カテゴリ

Web API 仕様文書がどんなものかを説明した文章のキーワードが属する．

(3) “Protocol”カテゴリ

Web API 仕様文書の Protocol を説明した文章のキーワードが属する．

10. 今後の課題

今後，深層学習モデルの正答率と誤差率の向上，他の深層学習モデルとの比較を行う．

11. まとめ

大量の Web API 仕様文書に対して CNN を用いた文章分類し，構造化された Web API 仕様記述モデルを生成する方法を提案した．提案方法を支援する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．

参考文献

[1] Y. Kim, Convolution Neural Networks for Sentence Classification, Proc, of EMNLP 2014, Oct. 2014, pp. 1746-1751.
[2] T. Mikolov, et al., Efficient Estimation of Word Representations in Vector Space, arXiv: 1301.3781v3 [cs.CL], Sep. 2013, pp. 1-12.
[3] Preferred Networks, Chainer, https://chainer.org/.
[4] ProgrammableWeb, https://www.programmableweb.com/.
[5] X. Rong, Word2vec Parameter Learning Explained, arXiv:1411.2738v4 [cs.CL], Jun. 2016, pp. 1-21.
[6] S. Sohan, et al., SpyREST: Automated RESTful API Documentation Using HTTP Proxy Server, Proc. of ICASE 2015, IEEE/ACM, Aug. 2015, pp. 271-276.