

# 深層学習を用いた Web API の仕様文書の生成方法の提案と評価

永井 利幸<sup>†</sup> 青山 幹雄<sup>†</sup>

南山大学 大学院 理工学研究科 ソフトウェア工学専攻<sup>†</sup>

## 1. 研究背景と研究課題

Web API の仕様を集約した様々なキュレーションサイトが公開されているが、仕様が自然言語の記述で、かつサイトごとに異なる形式のため、その理解や比較が困難である[3]. 本稿では機械学習を用いた Web API 仕様記述生成方法を提案する. 次の 3 点を研究課題とする.

- (1) Attention を用いた機械学習による Web API 記述から要約した仕様記述の生成方法
- (2) 要約仕様記述から OpenAPI 形式の仕様記述生成方法
- (3) 実際の Web API に提案方法を適用し, 有効性と妥当性の評価

## 2. 関連研究

### 2.1 Attention を用いた機械学習

Attention は入力データの一部のみに注目したベクトルをデコーダで使用する機械学習の方法である. 文書要約における Attention の適用方法が提案されている[1].

### 2.2 Web API 仕様文書の生成方法

Web API の説明文書から Web API の仕様文書を生成する方法が提案されている. しかし, 生成文書は要約されていないため記述量が増える[5].

### 2.3 OpenAPI

OpenAPI[3]とは, REST API の記述形式である. 記述された Web API の仕様はそれぞれオブジェクトとして定義される.

## 3. アプローチ

従来の方法では静的に定義しているため, Web API を公開した後に変更が発生した場合, 仕様書を変更して Web API を公開し直す必要がある. このことから, 仕様の変更を動的に仕様書に反映できるプロセスを提案する. ProgrammableWeb[2]上の説明文書を抽出し, OpenAPI の形式に変換する. ProgrammableWeb での Web API の挙動を説明した文書を要約し, OpenAPI で定義可能な仕様を抽出することで, 仕様書を生成する. 要約を行うことで, Web API の仕様に対する理解が促進される.

## 4. 深層学習による Web API 仕様文書生成方法

### 4.1 提案プロセス

提案プロセスを図 1 に示す.

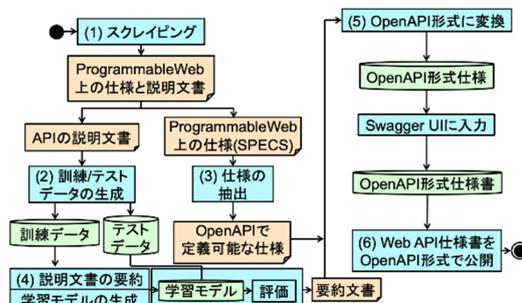


図 1 提案プロセス

### (1) Web API 記述文書のスクレイピング

ProgrammableWeb 上に登録されている Web API の説明文書と仕様を合わせたテキストデータを取得する. 説明文書と仕様は, それぞれ別のテキストファイルとして分割する.

### (2) 訓練データとテストデータの生成

モデルの学習と評価に用いる訓練データとテストデータを生成する. ProgrammableWeb 上でのカテゴリ内で Web API の登録数に偏りがあるため, カテゴリ毎で訓練データとテストデータの生成を行う.

### (3) Web API 仕様記述の抽出

OpenAPI で定義可能な仕様記述を取得する.

### (4) Web API 仕様記述の要約

取得した説明文書に対して要約する. 仕様記述上の重要文に対して Attention を付与し, 機械学習を用いて要約する.

### (5) Web API 仕様記述の OpenAPI 形式への変換

Web API 仕様記述を OpenAPI 形式に変換する.

### (6) Web API 仕様書を OpenAPI 形式で公開

OpenAPI 形式の Web API 仕様記述を SwaggerUI を用いて表示するために, YAML ファイルで公開する. Swagger UI を用いて Web API 仕様文書の内容を確認する.

### 4.2 Web API 説明文書の仕様生成

Web API 説明文書内の文章間の関係を CNN を用いてベクトル表現として抽出し, その関係を元に Attention を付与し, LSTM を用いて文書を要約する. Web API 仕様のカテゴリとして次の 5 つに分割する場合と分割しない場合の 2 パターンで学習する. Mapping, Weather, Mobile, Transportation, Social 教師データとして重要文に 1, 2, その他の文に 0 をラベルとして付与する. 重要文は tf-idf を計算し, スコアの高い上位 3 文章とする. tf-idf は文章中に多数出現する, かつ他文章中に出現しない重要語を特定できる. 重要語を多数含む文章を重要文と定義する. 生

A Method of Generating Specifications of Web APIs Using Deep Learning and its Evaluation

<sup>†</sup>Toshiyuki Nagai, Mikio Aoyama, Graduate School of Science and Engineering, Nanzan University

成したテストデータを用いて要約を行い、式(1)で表される ROUGE, 式(2)で表される単語の網羅率, 式(3)で表される圧縮率を用いて要約の適切さを評価する。

$$ROUGE - N(C, \mathcal{R}) = \left( \frac{Count_{match}(gram_N)(C, \mathcal{R})}{\# of N-grams \in \mathcal{R}} \right) \quad (1)$$

$$単語の網羅率 = \frac{説明文書と要約で一致した単語数}{説明文書の単語数} \quad (2)$$

$$圧縮率 = \frac{説明文書の単語数 - 要約の単語数}{説明文書の単語数} \quad (3)$$

## 5. プロトタイプの実装

プロトタイプのアーキテクチャを図2に示す。

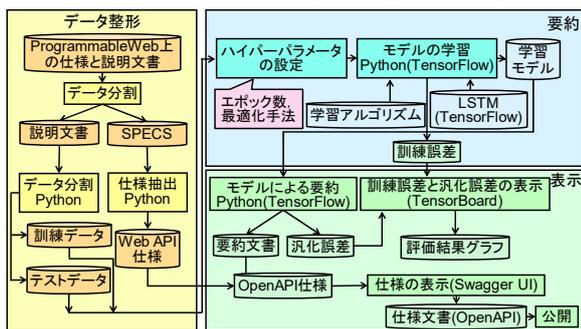


図2 プロトタイプのアーキテクチャ

## 6. 実データへの適用と評価

### 6.1 適用対象

ProgrammableWebのWeb APIに対して提案方法を適用した。文章数の違いによる結果への影響を最小限にするため、抽出するWeb APIの数は約300に統一した。表1に適用対象のデータ数を示す。なお、更新日時は全て2019年である。

表1 適用対象のデータ数

カテゴリ	Mapping	Weather	Mobile	Transportation	Social
更新日時	8/6	8/7	8/15	8/17	7/15
Web API 数	1,030	299	1,063	509	1,547
文章数	1,126	1,151	1,148	1,137	1,143
単語数	19,689	20,614	19,461	19,165	19,141

### 6.2 評価結果

ProgrammableWebのWeatherカテゴリに属するWeb APIに対して要約を行った結果を図3に示す。

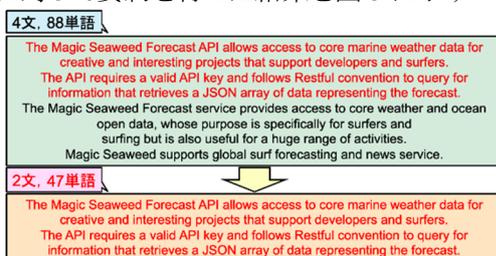


図3 要約の実行結果

4文(88単語)の文書が2文(47単語)に短縮された。削除された文章は同じ内容を繰り返している文や、Web APIではなくサービスに関する情報を含んでいた。損失の評価結果を図4, ROUGE-1, 単語の網羅率, 圧縮率の評価結果を図5に示す。全てのカテゴ

リを学習した場合の評価は、Weather カテゴリの Web APIを用いて行った。

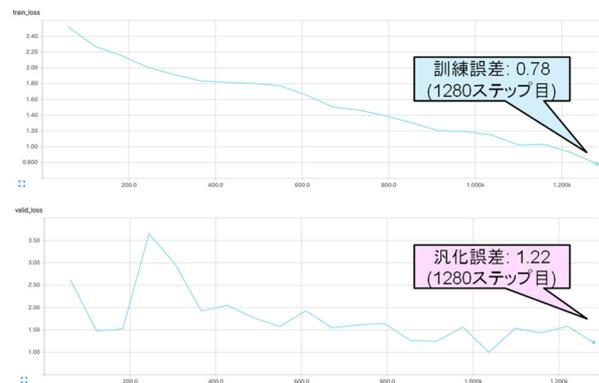


図4 損失の評価結果



図5 プロトタイプのアーキテクチャ

## 7. 考察

### (1) 適用結果の考察

図5に示す ROUGE-1の結果は生成したWeb API仕様記述が利用者にとって重要な情報を保持したまま要約できたことを示している。要約によって記述量が削減されたことからWeb APIの利用者にとって習得容易性[4]が向上したといえる。

### (2) 関連研究[5]との比較

関連研究[5]では機械学習を用いてURLなどを自動的に抽出しているが、説明文書の要約はされていない。提案方法は仕様記述の抽出と説明文書の要約が可能である。

## 8. まとめ

機械学習を用いて説明文書を要約し、Web API仕様記述を抽出することによるWeb API仕様文書の生成方法を提案した。重要文にAttentionを付与することで重要な情報を保持したまま要約でき、Web API利用者の仕様理解の促進が可能になる。提案方法は異なる仕様記述フォーマットの統一への支援として期待できる。

## 参考文献

[1] J. Cheng, et al., Neural Summarization by Extracting Sentences and Words, Proc of ACL '16, Jul. 2016, pp. 484-494.  
 [2] ProgrammableWeb, <https://www.programmableweb.com/>.  
 [3] Swagger, OpenAPI Specification, <https://swagger.io/specification/>.  
 [4] 山本 里枝子 他, Web APIの習得容易性と相互運用性, およびその定量評価方法の提案と適用評価, 情報処理学会論文誌, Vol. 60, No. 10, Oct. 2019, pp. 1896-1914.  
 [5] J. Yang, et al., Towards Extracting Web API Specifications from Documentation, Proc. of MSR '18, ACM, May. 2018, pp. 454-464.