

# 深層表現学習を用いた Web API エコシステム分析方法の提案と評価

加納 辰真† 青山 幹雄†

南山大学大学院 理工学研究科 ソフトウェア工学専攻†

## 1. はじめに

近年, Web API を通したネットワークで形成される Web API エコシステムが複雑化している. 本稿では, グラフモデルを用いて Web API エコシステムを Web API エコシステムグラフモデルとして定義し, 深層表現学習を用いて Web API エコシステムの特性を分析する方法を提案する.

## 2. 研究課題

本稿では以下の3点を研究課題とする.

- (1) Web API エコシステムグラフモデルの提案
- (2) 深層表現学習を用いて Web API エコシステムの特徴量を獲得, 分析する方法の提案
- (3) 実データを用いて提案方法の有効性を評価

## 3. 関連研究

- (1) Web API エコシステムの構造化  
Web API やマッシュアップサービスでグラフモデルを定義し, Web API エコシステムを可視化する[4][5].
- (2) 代替性を持つ Web API の推論  
深層学習やグラフ DB(グラフデータベース)を用いて類似性を持つ Web API を推論する[1][5].
- (3) GCN(Graph Convolutional Networks)  
グラフ構造で畳込み演算を行う NN(Neural Networks)[2].

## 4. アプローチ

Web API エコシステムの動的な構造変化に着目して分析を行う. Web API エコシステムの特徴量を深層表現学習を用いて獲得し, 進化特性を評価することを可能にする(図 1).

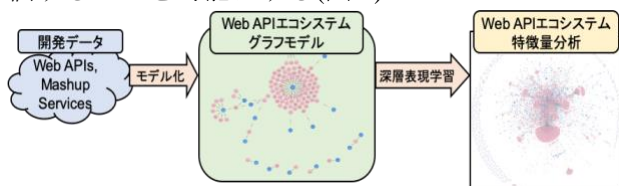


図 1 アプローチ

## 5. 提案方法

本稿で提案する分析方法のプロセスを示す.

- (1) Web API エコシステムグラフモデルを定義  
Web API エコシステムグラフモデルでは, ノードを

Web API とマッシュアップサービスとする. エッジはマッシュアップサービスノードに使用する Web API ノードへとリンクとする.

- (2) 分析項目の設定
- (3) グラフ DB の作成
- (4) 深層表現学習によるグラフの特徴量分析  
作成された Web API エコシステムグラフに対して GCN を用いてリンク予測を行い, Web API とマッシュアップサービスの内在関係を分析する. その関係を期間ごとに分析することで Web API エコシステムの進化特性を分析, 評価する.
- (5) 仮説検証  
必要に応じて, 一連のプロセスを繰り返す.

## 6. プロトタイプを用いた例題への適用

本稿の提案方法で生態系に見られる特性が Web API エコシステムでも明らかとなるか評価するためにプロトタイプを用いる(図 2). 生態系で見られる特性は以下の項目である.

- (1) 個体間の変化の連鎖がもたらすフィードバック機構  
フィードバックには, ポジティブフィードバックとネガティブフィードバックがあり, ポジティブフィードバックは変化を促進し, ネガティブフィードバックは抑制する.
- (2) 自己組織化  
個体群は個体間の局所的結合, あるいは適合によって組織を構成する.
- (3) 複雑系循環 (Complex System Cycle)  
生態系の変化は, 漸進的変化と突然変異を組み合わせた複雑系循環で構成する.
- (4) 共通化 (Coevolution) と共適応 (Coadaptation)  
共通化は Web API エコシステムとビジネスシステムが共に進化することである.  
共適応は相互に適応することである. 個体間や個体群間で発現する.

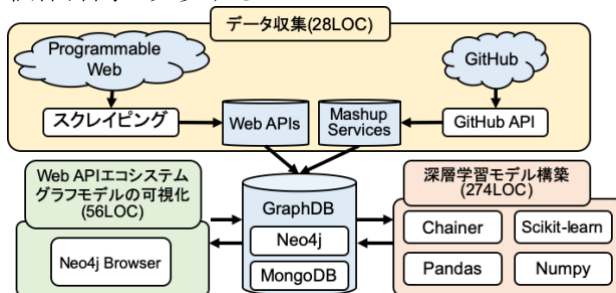


図 2 プロトタイプの構成

An Analysis Method of Web API Ecosystem Using Deep Representational Learning and its Evaluation)

†Tatsuma Kano, Mikio Aoyama, Graduate School of Science and Engineering, Nanzan University.

## 7. 実データへの適用による評価

### (1) 適用の目的と評価方法

Web API エコシステムの特徴を分析するために本稿で提案している分析方法を適用する。適用するデータを訓練データとテストデータに分割してそれぞれの正解率，誤差率を算出し，深層表現学習モデルの評価を行う。Web API エコシステムを一定期間ごとで可視化することで特徴を獲得する。獲得した特徴を生態系の特徴と比較することで本稿で提案している分析方法の有効性を評価する。

### (2) Web API エコシステムデータ

Web API エコシステムデータとして，期間中に ProgrammableWeb と GitHub 上で発表された Web API とマッシュアップサービスを用いた(表 1)。

表 1 分析対象データ

分析対象期間	Web API	マッシュアップサービス
期間	2005/09/01~2019/08/31	2005/09/01~2019/08/31
ノード数	19,793	6,401

### (3) 深層表現学習の学習結果

深層表現学習モデルの有効性を評価するために正解率，誤差率の推移と 200epoch における学習結果を評価した(図 3，表 2)。

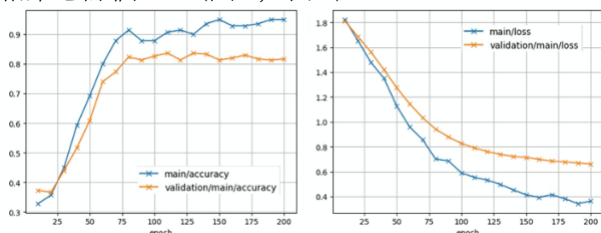


図 3 深層表現学習の学習結果

表 2 200epoch における学習結果

エポック数	訓練正解	訓練誤差	テスト正解	汎化誤差
200	0.95	0.36	0.82	0.66

### (4) Web API エコシステムの可視化結果

Web API エコシステムの可視化結果を示す(図 4)。

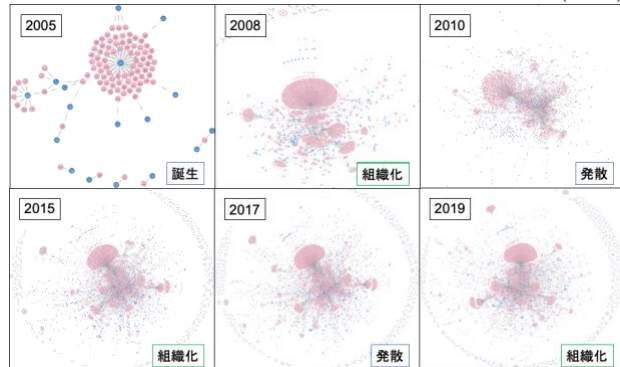


図 4 Web API エコシステムの進化の構造

Web API エコシステムは，誕生してから組織化と発散を循環することが明らかとなった。

## 8. 考察

Web API エコシステムの可視化により，生態系で見られる特徴が Web API エコシステムで明らかであるか考察する。

### (1) 個体間の変化の連鎖がもたらすフィードバック機構

図 4 の組織化から Web API エコシステムでは Web API の開発による Web API の進化がマッシュアップサービスの進化の連鎖となることが明らかになった。

### (2) 自己組織化

図 4 の組織化から Web API エコシステムでは同じカテゴリで分類される Web API とマッシュアップサービスの連鎖によって組織を形成することが明らかになった。

### (3) 複雑系循環 (Complex System Cycle)

図 4 の発散から Web API エコシステムの突然変異として，異なるカテゴリの Web API で構成されるマッシュアップサービスの出現が観測された。(2)の組織化と突然変異が循環することによって Web API エコシステムの進化することが明らかになった。

### (4) 共通化 (Coevolution) と共適応 (Coadaptation)

図 4 の組織化から Web API エコシステムでは，Web API の仕様変更とマッシュアップサービスに使用する Web API の変化との間に依存関係がある。共適応はエコシステム内の自己再組織化とそれをビジネスシステムに適応することとの 2つの進化パターンが確認できた。

## 今後の課題

今後の課題として次の 3 点を挙げる。

- (1) 深層表現学習モデルの妥当性検証
- (2) 分類精度の向上
- (3) グラフモデルの改善

## まとめ

本稿では，Web API エコシステムの動的な構造変化に焦点を当てて，深層表現学習を用いた Web API エコシステム分析方法を提案した。Web API エコシステムの進化の構造を可視化することで自然エコシステムで見られる特徴が Web API エコシステムでも見られることを明らかにした。

## 参考文献

- [1] H. Labbaci, et al., A Deep Learning Approach for Web Service Interactions, Proc. of WI 2017, ACM, Aug. 2017, pp. 848-854.
- [2] M. Schlichtkrull, et al., Modeling Relational Data with Graph Convolutional Networks, Proc. of ESWC 2018, LNCS, Vol. 10843, Springer, Jun. 2018, pp. 593-607.
- [3] M. Weiss, et al., Modeling the Mashup Ecosystem: Structure and Growth, R&D Management, Vol. 40, No. 1, Jan. 2010, pp. 40-49.
- [4] E. Wittern, et al., API Harmony: Graph-based Search and Selection of APIs in the Cloud, IBM J. of Research and Development, Vol. 60, Mar. 2016, pp. 1-11.