

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

M2018SE006 加納 辰真

指導教員 青山 幹雄

1 研究背景

近年, Web API を用いたソフトウェア開発が広がり, 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. Web API エコシステムをグラフ理論に基づきグラフモデルを定義する研究は行われているが, 深層表現学習を用いて Web API エコシステムを分析する研究は少ない. 本研究では, 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 エコシステムの進化特性を分析する方法を提案する.

2 研究課題

本研究では, 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) 大規模開発データに適用し, 提案方法の有効性と妥当性を評価

3 関連研究

3.1 エコシステムの構造と進化

3.1.1. 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等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研究がある[10][12]. マッシュアップサービスとは, 複数の Web API を組み合わせることで生み出された高性能な Web サービスのことである. 公開されている Web API を Web 上から取得し, 関連のある Web API やマッシュアップサービス, 検索する際に必要となるタグをリンクさせてグラフモデルを定義している.

3.1.2. ソフトウェアエコシステムの進化

ソフトウェアエコシステムは, 他の競合となるソフトウェアエコシステムとともに進化している[8]. ソフトウェアエコシステムが相互に影響を与えるメカニズムを特定している. ソフトウェアエコシステムは, 他のソフトウェアエコシステムと相互にフィードバックを行うことで互いに進化している.

3.2 代替性を持つ Web API の推論

Web API エコシステムとは, Web API を介して関連技術や Web サービスを連携し, 価値を創出するシステムである[3]. Web API エコシステムの構造は Web API の特性上, Web API の利用停止や類似性をもつ他の Web API の誕生によって, 動的構造変化している. よって, 代替性を持つ Web API を推論することが重要である. 深層学習を用いて類似性を持つ Web API を提案する研究[4]やグラフデータベース(グラフ DB)を用いる研究[12], 統計方法を用いる研究[9]がある.

3.3 グラフベースの深層表現学習

3.3.1. 表現学習

表現学習とは, 画像, 音, 文章, 時系列データ等の要素を分散表現として抽象化する方法である.

ネットワークの表現学習とは, ネットワーク構造からノードの分散表現を獲得する方法である[2]. このような方法として, graph2vec[5]が提案されている. 計算された分散表現を使うことで, 既存の複雑なネットワークのクラスタリング方法よりもラベル推定や分類タスクを高い精度で実行できる. また, ネットワーク表現学習は大規模グラフの可視化へも応用されている.

3.3.2. GCN(Graph Convolutional Networks)[7]

GCN は, グラフ構造における畳込み演算を定義し, CNN と類似した振る舞いするモデルである. 近傍ノードの信号を集約して重ね合わせて, 畳込み演算を行う. グラフを入力として, 1層が1ポップ隣接したノードの情報の畳込みを行う. 柔軟な特徴抽出を可能とし, ノードの分類予測, ランキング, グラフ分類, リンク予測等のタスクにおいて従来方法よりも高い精度を挙げている.

4 アプローチ

Web API エコシステムを理解するためには, 大規模な開発データを分析する必要がある. 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 エコシステムの構築期間を一定期間に分けて, Web API エコシステムグラフモデルの特徴量を獲得する. Web API エコシステムの特徴量を分析し, 進化特性, 重要度および, 成熟度を評価することを可能にする.

アプローチを図1に示す. Web 上に公開されている大規模開発データから Web API データとマッシュアップサービスデータから, Web API エコシステムを Web API エコシステムグラフモデル化する. モデル化するには Web API とマッシュアップサービスのデータを取得し, その間をリンクさせて Web API エコシステムを構築する. 構築したグラフデータに深層表現学習を用いることで自動的に特徴量を獲得する. 獲得した特徴量を分析することで Web API エコシステムグラフモデルを評価する. 特徴量を利用することで期待する効果を得る.

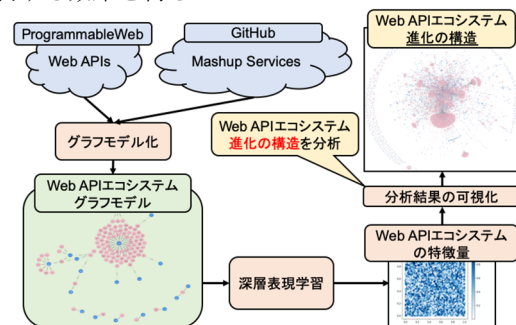


図 1 アプローチ

5 提案方法

5.1 Web API エコシステムグラフモデルの定義

Web API エコシステムを分析するために Web API エコシステムグラフモデルを作成する。本研究では、ProgrammableWeb 上のデータと GitHub から開発データを取得すると仮定している。Web API エコシステムグラフモデルのメタモデルを図 2 に示す。

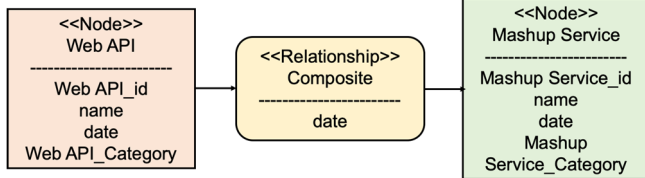


図 2 Web API エコシステムグラフモデルのメタモデル

5.1.1. ノードの定義

ノードは、対象となるオブジェクトで定義する。ノード周りの Web API やマッシュアップサービスとその関係を分析するために必要となる。Web API エコシステムグラフモデルでは、Web API とマッシュアップサービスをノードとした(表 1)。

表 1 Web API エコシステムグラフモデルのノードの定義

ノード	定義	プロパティ
Web API	Web API	Web API_id, name, date, Web API_Category
Mashup service	マッシュアップサービス	Mashup Service_id, name, date, Mashup_service_Category

5.1.2. エッジの定義

エッジは、内在関係で定義する。マッシュアップサービスに使用する Web API が存在すると、マッシュアップサービスノードから Web API ノードへとリンクする(表 2)。

表 2 Web API エコシステムグラフモデルのエッジの定義

エッジ	定義	プロパティ
Composite	マッシュアップサービスに使用する Web API	date

5.1.3. プロパティの定義

Web API エコシステムグラフモデルの各ノードおよびエッジに対してプロパティを定義する。Web API エコシステムグラフモデル上のプロパティに対するグラフ理論に基づいて Web API エコシステムの構造特性を分析する。任意のノードおよびエッジには、1 個以上のプロパティのセットが関連付けられる。プロパティの定義を表 3 に示す。

表 3 Web API エコシステムグラフモデルのプロパティの定義

プロパティ	定義
Web API_id	Web API の ID
name	名前
date	日付
Mashup service_id	マッシュアップサービスの ID
Web API_Category	Web API のカテゴリ
Mashup_service_Category	マッシュアップサービスのカテゴリ

5.2 Web API エコシステム分析方法のプロセス

本研究で提案する Web API エコシステム分析プロセスを図 3 に示す。

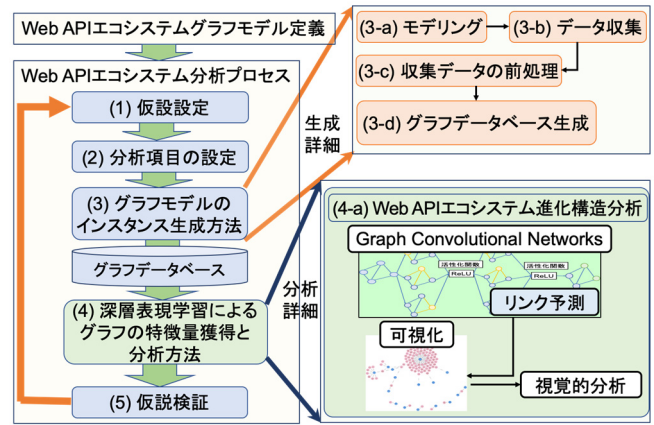


図 3 分析方法のプロセス

(1) 仮説設定

本研究では自然エコシステムの特徴が Web API エコシステムの特徴でも観測される。その特性によって Web API エコシステムの進化が推進されると仮説を立てた。設定した仮説は検証プロセスの結果に基づき必要があれば追加、変更する。

(2) 分析項目の設定

設定した仮説に基づいて、仮説ごとに明らかにすべき項目を列挙する。列挙した項目を統合して、分析項目を設定する。

(3) グラフモデルのインスタンス生成方法

(3-a) モデリング

分析対象の Web API エコシステムに Web API エコシステムグラフモデルで定義されていない特定のタイプのノードとエッジがある場合は Web API エコシステムグラフモデルを拡張する。

(3-b) データの収集

本研究で利用するデータは GitHub 上に公開されているマッシュアップサービスのデータと ProgrammableWeb 上に公開されている Web API のデータである。データ収集は主に、ソフトウェアプロジェクト管理プラットフォームが提供している Web API を使用する。GitHub 上のソフトウェアプロジェクトリポジトリからマッシュアップサービスを GitHub API を利用してデータを収集する。

ProgrammableWeb に公開されている Web API はスクレイピング技術を用いてデータを収集する。

(3-c) 収集データの前処理

収集したデータには、ノイズデータが存在する可能性がある。そこでグラフ DB ヘッダを挿入する前にノイズデータの削除などのデータ整形を行う。

(3-d) グラフデータベースの作成

グラフ DB 上に Web API エコシステムグラフモデルのインスタンスを作成し、前処理されたデータを挿入する。

(4) 深層表現学習によるグラフの特徴量獲得と分析方法

(4-a) Web API エコシステム進化構造分析

グラフ DB で作成された Web API エコシステムグラフモデルのインスタンスに対して、GCN の畳み込み演算によって、グラフ上のノードに対する特徴量を抽出する。抽出された特徴量を用いて、グラフ上でのリンク予測を行い、Web API

とマッシュアップサービスの関係を分析する。Web API とマッシュアップサービス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と重要度を分析する。

(5) 仮説検証

分析で得られた結果から、設定した仮説を検証する。必要に応じ、得られた結果に基づいて、仮説の追加や変更を行い、はじめからの一連のプロセスを繰り返す。

また、本研究ではこのプロセスで、設定された仮説に対する分析が深層表現学習によって表現できているかを検証する。

以上をふまえ、本研究での特徴量検証までのプロセスをもとに分析、評価を示す。

6 プロトタイプ実装

6.1 プロトタイプ実装の目的

本稿の提案方法でエコシステムに見られる特性が 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。自然エコシステムで見られる特性は以下の項目がある[1]。

(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

生態系の変化は、漸進的变化と突然変異を組み合わせた複雑系循環で構成する。

(4) 共通化と共通適応

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である。

共通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。

6.2 プロトタイプの構成

本研究で提案されている分析方法に基づいて作成したプロトタイプの構成を図4に示す。

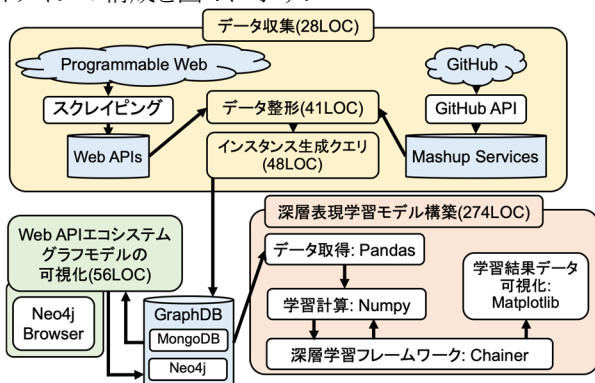


図 4 プロトタイプの構成

6.3 実装環境

表4,5にプロトタイプを実現するためのデータ収集、可視化分析の実装環境と深層表現学習モデルの構築に使用した実装環境を示す。

表 4 実装環境

コンポーネント名	名称	Version
OS	Ubuntu	18.04
Graphdatabase	Neo4j	3.5.9
	MongoDB	3.6.14
可視化ツール	Neo4j Browser	3.5.9
データ収集	Python	3.7.4
利用する外部API	GitHub API	v4

表 5 深層表現学習に用いた実装環境

コンポーネント名	名称	Version
OS	Ubuntu	18.04
実装言語	Python	3.7.4
フレームワーク	Chainer	6.4.0
データ加工ツール	Pandas	0.25.1
可視化ツール	Matplotlib	3.1.1
計算ライブラリ	Scikit-learn	0.21.3

7 実データへの適用による評価

7.1 適用目的と評価方法

Web API エコシステムの特徴を獲得するために本研究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、テストデータにわけて、正解率と誤差率を算出する。正解率の算出式は式1に示す。誤差率は Softmax-Cross-Entropy 関数で算出する。算出された値で深層表現学習の学習モデルを評価する。

$$\text{正解率} = \frac{\text{正解したデータ数}}{\text{総データ数}} \quad [1]$$

7.2 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、Web API とマッシュアップサービスのデータを分析対象とする。分析対象期間を表6に示す。本稿では2005年9月1日から2019年8月31日にかけて ProgrammableWeb と GitHub 上で発表された Web API とマッシュアップサービスについて分析を行う。

表 6 分析対象データ

分析対象	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

7.3 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と誤差率の推移と 200 epoch における学習モデルの結果を図5、表7に示す。

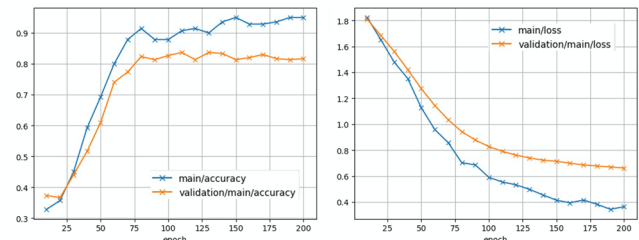


図 5 深層表現学習の学習結果

表 7 200epoch における学習結果

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

7.4 Web API エコシステムの可視化結果

Web API エコシステムグラフモデルの視覚的分析結果を図 6 に示す。Web API エコシステムは誕生してから一定期間ごとに組織化と分散を繰り返している。組織化は同じカテゴリでの連鎖, 分散はサブグラフの発散が挙げられる。図 7 に組織化と分散の例を示す。Web API エコシステム進化の構造を図 8 に示す。

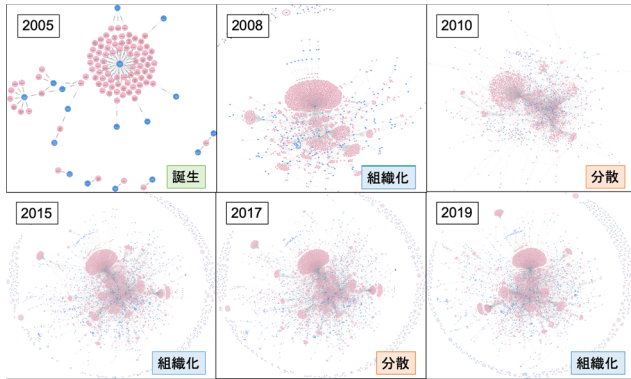


図 6 Web API エコシステム構造の進化

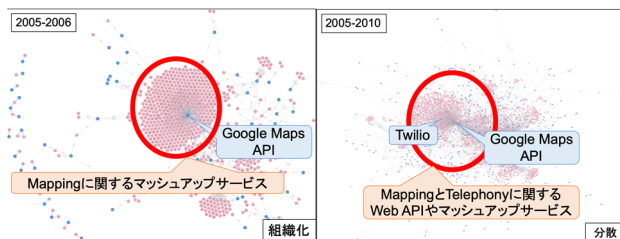


図 7 Web API エコシステムの組織化と分散の例

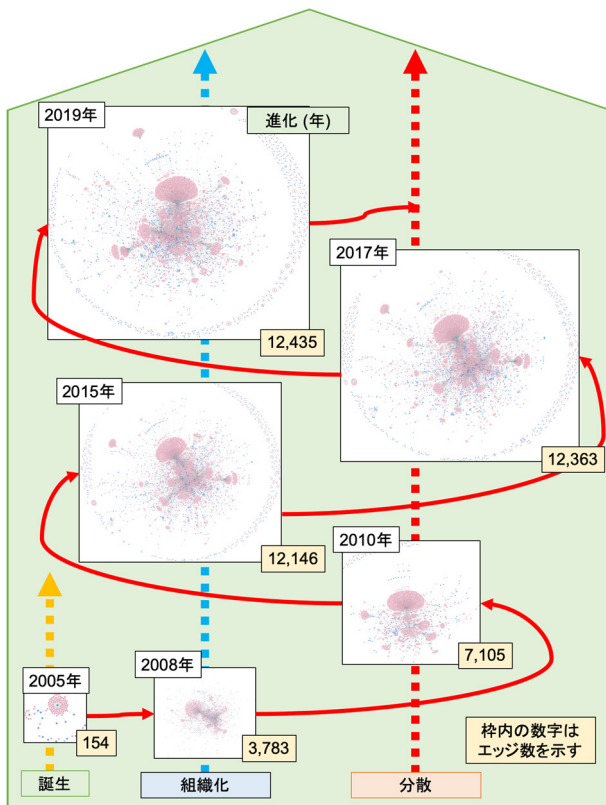


図 8 Web API エコシステム進化の変遷

8 考察

7.4 により, 自然エコシステムで見られる特性[1]が Web API エコシステムでも明らかであるか考察する。

(1) ポジティブフィードバックによる共進化

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り, この相互作用に共進化していることが明らかになった。

(2) 自己組織化

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

図 8 より組織化と突然変異が出現して分散することが循環することによって Web API エコシステムが進化することが明らかとなった。

(4) 共通化と共通適応

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。共通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。

9 今後の課題

- (1) 深層表現学習モデルの評価方法と妥当性検証
- (2) 分類精度の改善
- (3) グラフモデルの改善

10 まとめ

本研究では, Web API エコシステムの動的な構造変化に着目して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をした。提案方法により, Web API エコシステム進化の構造を可視化した結果, Web API エコシステムの進化特性を明らかにした。

参考文献

- [1] 青山 幹雄, サービス生態系工学への扉を開く, 情報処理学会ソフトウェア工学研究会ウィンターワークショップ 2008・イン・道後 論文集, Vol. 2008, No. 3, Jan. 2008, pp. 79-80.
- [2] 浅谷 公威, ネットワークの表現学習, 人工知能, Vol. 31, No. 4, 2016, pp. 587-593.
- [3] S. Jansen, et al., A Sense of Community: A Research Agenda for Software Ecosystems, Proc. of ICSE 2009, IEEE, May. 2019, pp. 187-190.
- [4] H. Labbaci, et al., A Deep Learning Approach for Web Service Interactions, Proc. of WI2017, ACM, Aug. 2017, pp. 848-854.
- [5] A. Narayanan, et al., graph2vec: Learning Distributed Representations of Graphs, Proc. of MLG 2017, Aug. 2017, pp. 1-8. arXiv: 1707.05005.
- [6] ProgrammableWeb, <https://programmableweb.com/>.
- [7] M. Schlichtkrull, et al., Modeling Relational Data with Graph Convolutional Networks, Proc. of ESWC 2018, LNCS, Vol. 10843, Springer, Jun. 2018, pp. 593-607.
- [8] J. Teixeira, et al., How Do Software Ecosystems Co-Evolve?, Proc. of ICSOB 2017, LNBP Vol. 304, Springer, Aug. 2017, pp. 115-130.
- [9] X. Wang, et al., Mashup-Oriented API Recommendation via Random Walk on Knowledge Graph, IEEE Access, Vol. 7, 2018, pp. 7651-7662.
- [10] M. Weiss, et al., Modeling the Mashup Ecosystem: Structure and Growth, R&D Management, Vol. 40, No. 1, Jan. 2010, pp. 40-49.
- [11] M. White, et al., Toward Deep Learning Software Repositories, Proc. of MSR 2015, IEEE, May 2015, pp. 334-345.
- [12] E. Wittem, 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.