2021 年度

修　士　論　文

グラフ埋め込みを用いた
サービスクラスタリング

指導教員: 村上 陽平

立命館大学大学院 情報理工学研究科
博士課程前期課程 情報理工学専攻
計算機科学コース

学生証番号: 6611200012-8

氏名: 大久保 弘基

グラフ埋め込みを用いたサービスクラスタリング

大久保 弘基

内容梗概

複数の Web サービスを組み合わせて作成されたサービスのことを複合サービスという．複合サービスにより，個々に提供される原子サービスよりも拡張した機能を提供することができ，また既存サービスを再利用することが可能になった．このような複合サービスでは，Web サービスの提供中止や，サーバのメンテナンスなどにより，複合サービスに組み合わされた 1 部分の Web サービスが利用できなくなることがある．これらの原因により，連鎖的に複合サービス全体の機能に影響が生じる．この問題に対して，WEB サービス記述ファイル（以下，WSDL ドキュメントと呼ぶ）やサービス説明文をテキストマイニングし，サービス機能ごとにクラスタリングする研究がある．しかしながら，これらの手法ではテキストの記述内容に大きく依存するため，サービス提供者の命名規則による影響を受けやすい．

そこで，本研究では，サービス提供者の命名規則による影響を受けない，複合サービスによるサービス呼び出しに着目し，複合サービスのサービス依存関係に基づくサービス機能ごとのクラスタリング手法を提案する．具体的には，同一の複合サービスに組み合わされたことのある原子サービスの連携関係をグラフで表し，そのグラフのトポロジカルな特徴から各ノード間の類似度を計算することで，サービス機能ごとのクラスタリングを行う．本手法の実現に辺り，取り組むべき課題は以下の２点である．

文脈を考慮したサンプリング戦略

各原子サービスがネットワーク全体において，どのような意味を持つのか理解するために，各原子サービスの特徴となる近傍を適切にサンプリングすることが必要である．さらに，原子サービスは複数の複合サービスによって，異なる他の原子サービスと連携される可能性があるため，近傍の原子サービスが同一の複合サービスで連携されているとは限らない．同一の複合サービスで連携された場合と区別するために，文脈を考慮したサンプリングが必要である．

ノード分散表現の学習

ネットワークから抽出したトポロジー情報からクラスタリングを行うためにそれぞれの原子サービスの分散表現を獲得することが必要である．また，周辺

のネットワークが似ているノード同士の分散表現は近くなるような学習方法が必要である．

１つ目の課題に対しては，グラフ埋め込み技術に倣い，ノード探索を行うことでグラフのトポロジカルな特徴をサンプリングした．同一の複合サービスに組み合わされた原子サービスをノード，連携関係をエッジとしたネットワークを構築し，そのネットワークに対しノード探索を行って近傍をサンプリングした．また，ノード探索を行う際に，同一複合サービスで連携された原子サービスを優先する探索を行うことで，連携元の複合サービス情報を，ノード探索によりサンプリングされるノードリストに埋め込むためである．

２つ目の課題に対しても，グラフ埋め込み技術に倣い，ノード探索結果を単語分散表現の技術である skip-gram model で学習させることで，各ノードのベクトル表現を生成した．さらに，skip-gram model より高精度である BERT で学習することで，更なる精度向上を図った．

提案手法によってグラフのトポロジー情報から生成したクラスターを，人手で付与したカテゴリ情報に基づく正解クラスターと比較することで，生成したクラスターがサービス機能ごとにクラスタリングされていたかを評価することで，提案手法の有効性を検証した．本研究の貢献は以下のとおりである．

文脈を考慮したサンプリング戦略

ノード探索方法をランダムウォーク，幅優先探索にバイアスをかけた探索，深さ優先にバイアスをかけた探索，同一複合サービスに組み合わされる原子サービスを優先する探索の４つで評価した．その結果，正解クラスタを Primary カテゴリと Secondary カテゴリで作成し，正解クラスタが重複する対応付けを行なった場合，同一複合サービスに組み合わされる原子サービスを優先する探索のクラスタリング精度は他手法に比べて 5%上昇した．

ノード分散表現の学習

ネットワークのノードのサンプリング結果を各ノードが空白で区切られたテキストデータに加工し，従来の単語分散表現学習モデルで学習することで，各ノードの分散表現を獲得することができた．また，正解クラスタを Primary カテゴリと Secondary カテゴリで作成し，正解クラスタが重複する対応付けを行なった場合，skip-gram model を用いたグラフ埋め込みのクラスタリング精度は BERT を用いたグラフ埋め込みと比べて 15%上昇した．

Service Clustering with Graph Embedding

Koki Okubo

Abstract

Creating a new service by combining multiple web services is called the Mashup service. The Mashup service functionality is more extended than individually provided atomic services, by reusing existing web services. In such a Mashup service, if some of the combined web services become unavailable, the function of the entire Mashup service will be affected. In response to this problem, there is research being conducted on text mining web service description files, also known as WSDL, and clustering services by function. However, in the existing research, terms that represent service functions are specified by WSDL text mining. They are highly dependent on the description contents, and thus are easily affected by how the providers name and describe the services.

In this study, we propose a clustering method for each service function based on service dependencies of Mashup service. Specifically, the cooperative relationship of services combined in a Mashup service is represented as a graph, and by calculating the similarity between each node based on the topological characteristics of the graph, the services are clustered with services with similar functions. There are two issues to be addressed in realizing this method.

Contextual sampling

To understand the meaning of each atomic service in the whole network, it is necessary to properly sample the characteristic neighborhood of each atomic service. However, atomic service may also be linked to other atomic services by multiple Mashup services. Contextual sensitive sampling is needed to distinguish between cases linked by the same composite service.

Learning node distributed representation

From the topology information extracted from the network, we need to obtain a distributed representation of each atom service for clustering. In addition, we require a learning method that allows the surrounding network to be as close to the distributed representation of similar nodes as possible.

For the first task, we sampled the topological features of the graph by performing a node search, following the graph embedding technique. We

constructed a network with atomic services combined into the same Mashup service as nodes and coordination relationships as edges, and performed node search on the network to sample the neighborhood. This is due to the fact that when node retrieval is performed, the atomic services linked by the same Mashup service are walked preferentially, and the Mashup service information of the link source is embedded in the node list sampled by node retrieval.

For the second task, following the graph embedding technique, we generated a vector representation of each node by training the node search results with a skip-gram model, a technique for word distributed representation. In addition, we used BERT, which is more accurate than the skip-gram model.

We verified the effectiveness of the proposed method by comparing the clusters generated by the proposed method based on the topological information of the graph with the correct clusters based on manually assigned category information, and by evaluating whether the generated clusters are clustered by the service function. The contributions of this research are as follows.

Contextual sampling

We evaluated four node-walking schemes: random walk, biased breadth-first search, biased depth-first search, and preferential search for the same Mashup service. As a result, the clustering accuracy of the search prioritizing atomic services combined with the same Mashup service improved by 5% when the correct clusters were created in Primary and Secondary categories and the correct clusters were mapped to overlapping clusters.

Learning node distributed representation

The network nodes sampling results were converted into text data in which each node was separated by a blank space, and the distributed representation of each node was obtained by training with a conventional word distributed representation learning model. The clustering accuracy of graph embedding using the skip-gram model was 15% higher than that of graph embedding using BERT when the correct clusters were created in Primary and Secondary categories and the correct clusters were mapped to overlapping clusters.

グラフ埋め込みを用いたサービスクラスタリング

目次

第1章 はじめに

近年，Web サービスにおいて，サービスコンピューティングという概念が急速に発達している．サービスコンピューティングとは，迅速かつ柔軟なサービス開発を目標として開発された技術である．例えば，複数の Web サービスを組み合わせた複合サービスは個々に提供される原子サービスより，拡張した機能を提供でき，既存の Web サービスを再利用できる．

１つ目の，拡張した機能を提供できることについて説明する．一般的に知られている Web サービスは Google Maps API のような地理に関する情報を提供する Web サービスや，Twitter API のような提供者が保有するデータにアクセスできる Web サービスなどである．これらの単純な機能を持つ Web サービスを原子サービスと呼ぶ．個々に提供される原子サービスより，拡張された機能を提供できるとは，これらの原子サービスを複数個組み合わせることによって，本来提供される原子サービスよりも複雑な機能を提供することができる．例えば，Google Maps API と天気情報を提供する API を組み合わせることで，地図上に各地の天気情報を表示する複合サービスを開発することができる．他にも，原子サービスは API で連携されるので，組み合わせられている一部 Web サービスを変更，または増やすことで，複合サービスの機能を拡張することができる．

２つ目の，既存の Web サービスを再利用できることについて説明する．複合サービスは，オンライン上で提供される Web サービスの API を用いて開発される．これにより，開発に必要だったコストを抑え，迅速な開発を可能とする．

このような複合サービスでは，サービスの提供中止や，サーバのメンテナンスなどにより，組み合わされた一部のサービスが利用できなくなることがある．これらの原因により，連鎖的に複合サービス全体の機能に影響が生じる[1]．このような障害が発生した場合，複合サービスのサービス提供者は利用できなくなったサービスと代替可能な類似サービスを発見し，組み替える必要がある．しかしながら，オンライン上に存在する膨大な数の Web サービスの中から類似サービスを発見するのは非常に困難である．このような場合，Web サービスを自動的に分類することが必要になり，クラスタリングは Web サービスを分類するための非常に効率的なアプローチの１つである．このアプローチに関する研究として，Web サービス記述ファイル（以下，WSDL ドキュメントと呼称）を用いてサービス機能ごとにクラスタリングする研究がある．しかしながら，この手

法では WSDL ドキュメントのテキストマイニングによりサービス機能を表す機能用語を特定しており，記述内容に大きく依存するため，Web サービス提供者の命名規則による影響を受けやすい．

そこで，本研究では，サービス提供者の命名規則による影響を受けない，複合サービスのサービス依存関係に基づく類似機能サービスのクラスタリング手法を提案する．具体的には，複合サービスにより組み合わされた原子サービスの連携関係をグラフで表し，Network Embedding（以下，グラフ埋め込み技術と呼称）による各ノードの分散表現を用いて類似機能サービス群にクラスタリングする．本手法の実現にあたり，取り組むべき課題は以下の通りである．

文脈を考慮したサンプリング戦略

各原子サービスがネットワーク全体において，どのような意味を持つのか理解するために，各原子サービスの特徴となる近傍を適切にサンプリングすることが必要である．さらに，原子サービスは複数の複合サービスによって，異なる他の原子サービスと連携される可能性があるため，近傍の原子サービスが同一の複合サービスで連携されているとは限らない．同一の複合サービスで連携された場合と区別するために，文脈を考慮したサンプリングが必要である．

ノード分散表現の学習

ネットワークから抽出したトポロジー情報からクラスタリングを行うためにそれぞれの原子サービスの分散表現を獲得することが必要である．また，周辺のネットワークが似ているノード同士の分散表現は近くなるような学習方法が必要である．

以下，本論文では，２章において従来の研究のクラスタリング手法として，インタフェース情報に基づくクラスタリング，オントロジーに基づくクラスタリング，サービス説明文に基づくクラスタリングについて説明する．次に，３章では近傍に基づくクラスタリングとして用いたコサイン類似度，ジャカード類似度のサービス依存関係グラフ，またその計算方法について説明する．続いて，４章ではノードの分散表現に基づくクラスタリングであるグラフ埋め込み技術について説明を行い，ネットワークのサンプリング手法やグラフ埋め込みで用いる skip-gram model と BERT について説明する． ５章では３章と４章で説明を行なったクラスタリング手法に対する考察を行い，最後に今後の展望や課題について述べて結論とする．

第2章 サービスクラスタリング

本章では，既存の類似機能サービスのクラスタリング手法について説明する．サービスクラスタリングは，複合サービスにおいて，その構成要素のサービスが利用できないときに，類似機能サービスを見つけるのに有用である．

複合サービスが使用できなくなった場合のサンプルケースとして，表 1 のケースを用いる．この複合サービスが正常に動作している場合は図 1 左側のように地図上に対応する各地方の天気に関する情報が表示される．しかし，組み合わされたサービスである地図情報サービスがサービスの提供停止やサーバのメンテナンスなどにより使用できない場合，図 1 右側のような何も表示されていない地図上に天気情報だけが表示される，または，この複合サービス全体が動作しないことが考えられる．このような使用できないサービスが発生した場合，複合サービスの提供者は使用できないサービスと機能的に類似する代替可能なサー

表 1：使用できない複合サービスのサンプルケース

複合サービスの機能	組み合わされたサービス	故障状況
地図上に各地方の天気情報を表示する	・地図情報サービス ・天気情報サービス	地図情報サービスが使用不可

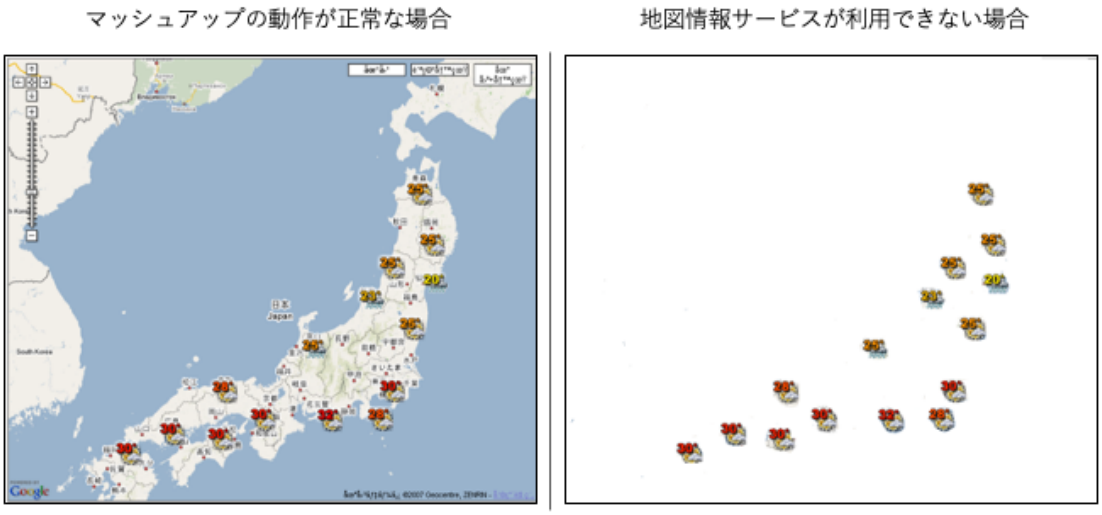

図 1：複合サービスの動作例

ビスの発見が必要となる．しかしながら，オンライン上に存在する膨大な数の Web サービスの中から類似サービスを発見するのは非常に困難である．このような場合，Web サービスを自動的に分類することが必要になり，クラスタリングは Web サービスを分類するための非常に効率的なアプローチの１つである．本章では，このアプローチに関する研究として，インタフェース情報に基づくクラスタリング手法やオントロジーに基づくクラスタリング手法，サービス説明文に基づく手法について説明する．

2.1 インタフェース情報に基づくクラスタリング

インタフェース情報に基づくクラスタリング手法において，ユーザが推薦されるサービスはユーザが指定したサービスの機能を表す単語に類似するものである．Elgazzar らは，WSDL ドキュメントからサービスの機能を表す用語をテキストマイニングし，それらを機能的に類似した Web サービスグループにクラスター化する方法を提案した[2]．この提案手法の実現には以下のステップが必要になる．

1. WSDL ドキュメントからクラスターを作成する
2. クラスターからユーザの要求する Web サービスを発見する

１つ目の WSDL ドキュメントからクラスターを作成するステップについて説明する．アルゴリズムの全体像を図 2 に示す．はじめに検索エンジンのクローラーを使い，インターネット上から WSDL ドキュメントをクロールする．WSDL ドキュメントのサンプルを図 3 に示す．次に WSDL ドキュメントから Web サービスの意味や動作などを表す用語を抽出するために，WSDL タイプ，WSDL メッセージ，WSDL ポート，および Web サービス名から単語を抽出する．これらの単語を結合して，Web サービスを機能的に類似したグループにクラスター化する．これは検索エンジンが Web サービスを識別し，ユーザ要求を満たす Web サービスを発見するための処理である．

２つ目のクラスターからユーザの要求する Web サービスを見つけるステップについて説明する．はじめにユーザからサービス検索エンジンに目的の単語でクエリを実行する．次に１つ目のステップで作成されたクラスターと入力されたクエリを意味的に一致させる．これは例えば，「vehicle」と「car」のような意味は同じだが単語が違うもの同士を一致させるための処理である．最後に，要求された目的を満たす最も関連性の高い Web サービスをユーザに返す．

この手法を上記の複合サービスのサンプルケースに適応した場合，ユーザは目的の単語を「map」とし，Web 検索エンジンにクエリを実行する．その後，事前にクラスター化された Web サービスから「map」に関連する Web サービスを発見し，ユーザにその結果を提示する．しかしながら，この手法では，WSDL ドキュメントのテキストマイニングによりサービスの機能を表す用語を特定しており，記述内容に大きく依存するため，提供者の命名規則による影響を受けやすい問題がある．

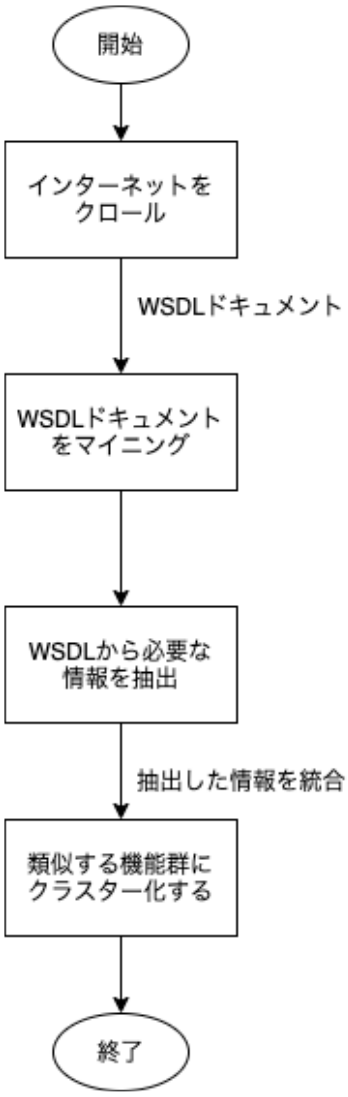

図 2：WSDL ドキュメントからクラスターを作成するフローチャート

```
<?xml version=”1.0” encoding=”shift_jis”?>
<wsdl:definitions
～省略～
        <wsdl:message name=”getDataRequest”>
                <wsdl:part name=”text” type=”xsd:string”/>
        </wsdl:message>
        <wsdl:message name=”getDataResponse”>
                <wsdl:part name=”getDataReturn” type=”xsd:string”/>
        </wsdl:message?>
        <wsdl:portType name=”sample”>
                <wsdl:operation name=”getData”>
                        <wsdl:input name=”getDataRequest”/>
                        <wsdl:output name=”getDataResponse”/>
                </wsdl:operation>
        </wsdl:portType>
～省略～
</wsdl:definitions>
```

図 3：WSDL ドキュメントのサンプル

2.2 オントロジーに基づくクラスタリング

Xie らは，ドメインオントロジーの構造を分析し，それらを機能的に類似した Web サービスグループにクラスター化する方法を提案した[3]. Web サービス機能と進捗状況を使用して，類似度を計算する．オントロジーとは概念，プロパティ，概念の様々な機能と属性，およびスロットの制限の正式な使用である．またドメインオントロジーは概念の説明とドメイン内の概念間の関係である．本節では，ドメインオントロジー階層２つの概念関係を計算することにより，概念間の類似性を取得する．類似サービスのクラスター化手順を図 4 に示す.

はじめに，概念測定の深さ，経路，概念密度，アンチセンス間の測定について説明する．階層に基づく概念の意味的類似性の影響要因は次の通りである.

1. 階層の深さ：ドメインオントロジー階層はより深く，概念分類がより網羅

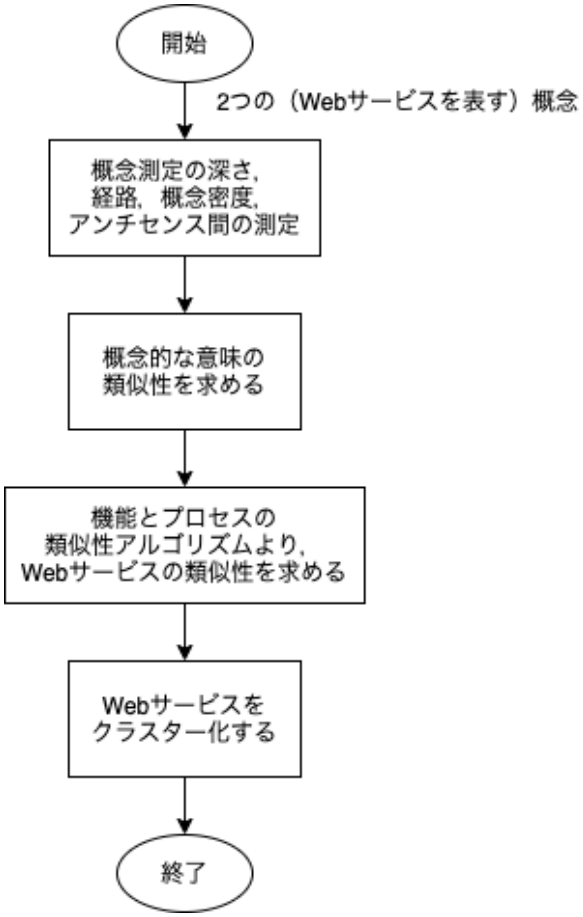

図 4：オントロジーに基づくクラスタリングのフローチャート

的であり，階層がない２つの概念がより類似していることを表す.

2. 最短経路のステップ：２つの概念のステップが多いほど，それらの距離は遠く，類似性は低くなる.
3. 概念の深さ：２つの概念がより深い場合，概念がより包括的に分析されることを意味する.
4. 概念の密度：概念の深さと同様に，概念にさらに兄弟の概念がある場合，概念がより包括的に分析されることを意味する．したがって，密度の高い概念ほど類似性は高くなる.

上記の分析から，概念測定の深さと経路が計算される．また，概念密度はある概念と同じ概念に属する概念の数とドメインオントロジー階層のない全ての概念の数から計算することができる．さらに，反対語の関係をアンチセンス関係

と呼び，これは類似度の計算で肯定的な役割を果たすので，類似性の計算中に２つのアンチセンス概念の意味的類似性を減らす.

次に，２つ目の概念的な意味的類似性の求め方を説明する．これは１で求めた概念測定の深さ（Dep），経路（Path），概念の密度（p），アンチセンス間の測定（Anti）を用いることで以下のように計算できる．A, B は概念を表し，α, β, δは調整係数であり，２つの概念のパス，密度，およびアンチセンス関係の異なる重みを表す.

$$Sim(A,B) = Dep(A,B) \times (\alpha Path(A,B) + \beta p(A,B) + \delta Anti(A,B))$$

続いて，３つ目の機能とプロセスの類似性のアルゴリズムにより，WEB サービスの類似性の求め方を説明する．WEB サービスの類似性を求めるには機能の類似性とプロセスの類似性をそれぞれ求める必要がある．定義を下記に示す.

WSA, WSB：サービスの説明で，名前と記述内容を含む

AStr, Bstr：プロセスの概念文字

AC, BC：WSA または WSB から概念を抽出したもの

プロセスの類似性は２つ目で用いた計算の引数に Astr と Bstr を入力することで求める．機能の類似性を求めるためには入出力の一致度とサービス記述内容の類似性を求める必要がある．入出力の一致度は WSA と WSB を用いることで求めることができる．サービス記述内容の類似性は２つ目で用いた計算の引数に AC, BC を入力することで計算できる．最後に，サービス類似度は機能類似度とプロセス類似度を足し合わせることで計算できる.

最後に４つ目の WEB サービスをクラスター化する手法については，３つ目までに求めた WEB サービスの類似度と k-means を使用することで類似サービスグループにクラスター化することができる.

しかしながら，この手法では類似度アルゴリズムで密度と関係の重みが考慮されるため，オントロジーの構造を正確に構築する必要がある.

2.3 サービス説明文に基づくクラスタリング

Min らは, Web サービスを word2vec によって拡張された LDA モデルを用いて, 機能別にクラスタリングする手法を提案した[4]． LDA モデルとは文書のトピックを推定するモデルで，文書の分類や検索などに利用される手法である.

一般的な WSDL から抽出した特徴を利用しサービスを機能ごとにクラスタリングする手法（WSDL ベース）では，WSDL に記述される単語は限られており，

クラスタリングの品質が低くなる可能性が考えられた．また WSDL ベースはテキスト情報がないため，Web サービス間の意味的な関係を捉えることができない．そこで，サービス説明文と LDA モデルを用いて，Web サービスを機能ごとにクラスタリングする．また，LDA モデルを word2vec に拡張することで，高品質な単語ベクトルを利用する．この word2vec に拡張した LDA モデルは WE-LDA と呼ばれる．WE-LDA の全体的なアルゴリズムを図 5 に示す．WE-LDA では，開発者が Web サービスの機能を説明するために作成したテキスト（サービス説明文）を入力とし，サービス説明文に含まれる全用語の単語ベクトルを word2vec で学習する．次に，k-means++アルゴリズムを用いて，単語ベクトルの類似性に基づき，これらの単語を異なるグループにクラスタリングする．その後，単語のクラスタ情報は LDA モデルに基づいて，Web サービスの分散表現を訓練するために使用される．最後に，k-means などのクラスタリングアルゴリズムに基づき，全ての Web サービスを異なる機能ドメインにクラスタリングするか，潜在的なトピックに従って，同じトピックに属する Web サービスを一緒にクラスタリングすることができる．

しかしながら，この手法はサービス説明文からその Web サービスの機能を表す機能用語を特定しており，その特徴に基づきクラスタリングを行うため，開発者の命名規則による影響を受けやすい．

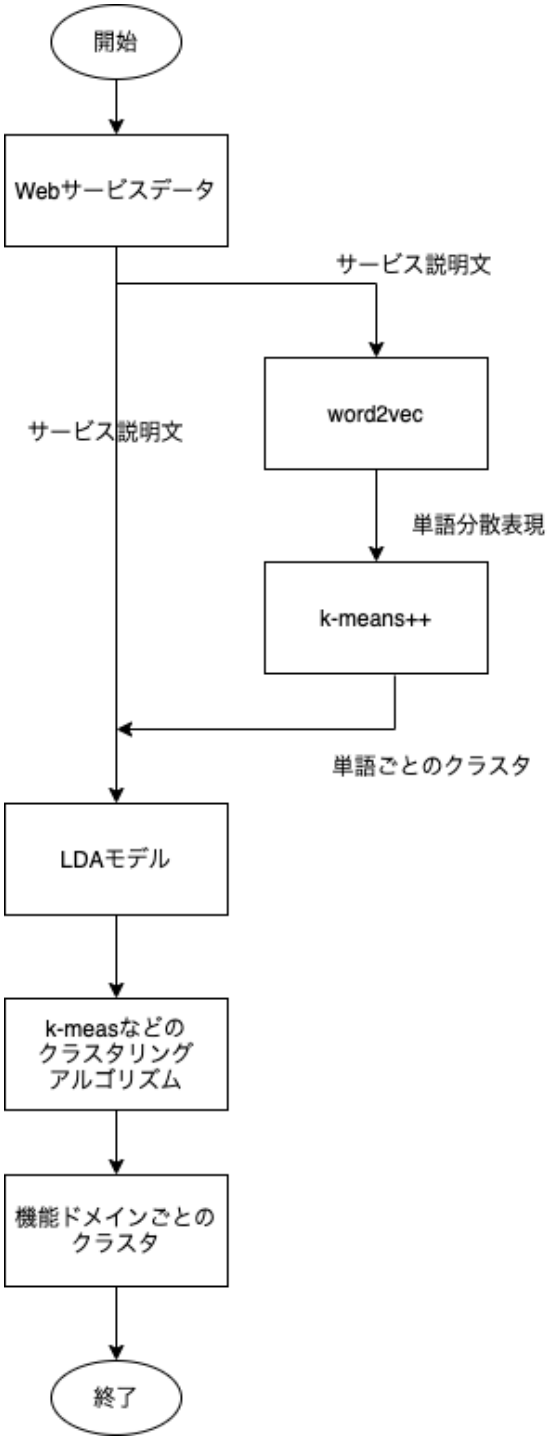

図 5：WE-LDA のフローチャート

第3章 近傍に基づくクラスタリング

本章では，コサイン類似度，ジャカード類似度の計算で用いるネットワークの定義とその計算方法について説明する．

3.1 サービス連携の依存グラフ

複合サービスに組み合わされている原子サービスデータを用いて，複合サービスによるサービス連携の依存グラフを構築する．具体的には，図 6 のようなグラフを作成する．このグラフでは，ある複合サービスで”Google Maps”と”Yahoo Maps”と”Flicker”が組み合わされていたことから，この３つのノードを生成し，エッジを繋いでいる．また，他の複合サービスで”Flicker”と”Twitter”が組み合わされていた場合，新しく”twitter”ノードを作成し，”Flicker”ノードとエッジを繋ぐ．

3.2 コサイン，ジャカード類似度を用いた類似度計算

サービス間の類似度の計算にはコサイン類似度とジャカード類似度を用いる．コサイン類似度では，２つのベクトルの角度の近さで類似度を算出する．具体的には，２つのベクトルを A, B とすると，以下の式によって算出できる．

$$cos(A,B)\frac{A\cdot B}{|A||B|}$$

またジャカード類似度では，２つの集合同士の和集合の割合で類似度を算出する．具体的には，２つの集合を U, V とすると，以下の式によって算出できる．

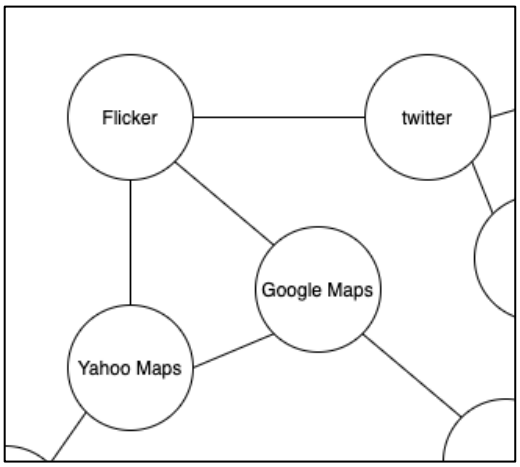

図 6：サービス連携の依存グラフ例

$$jaccard(U,V)\frac{|U \cap V|}{|U \cup V|}$$

それらを用いて類似度計算を行うアルゴリズムを図 7 に示す．コサイン類似度，ジャカード類似度を用いた類似度計算では，まず 3.1 で述べたネットワークを構築し，全サービスの隣接ノード群を取得する．図 6 を用いると”Flicker”の隣接

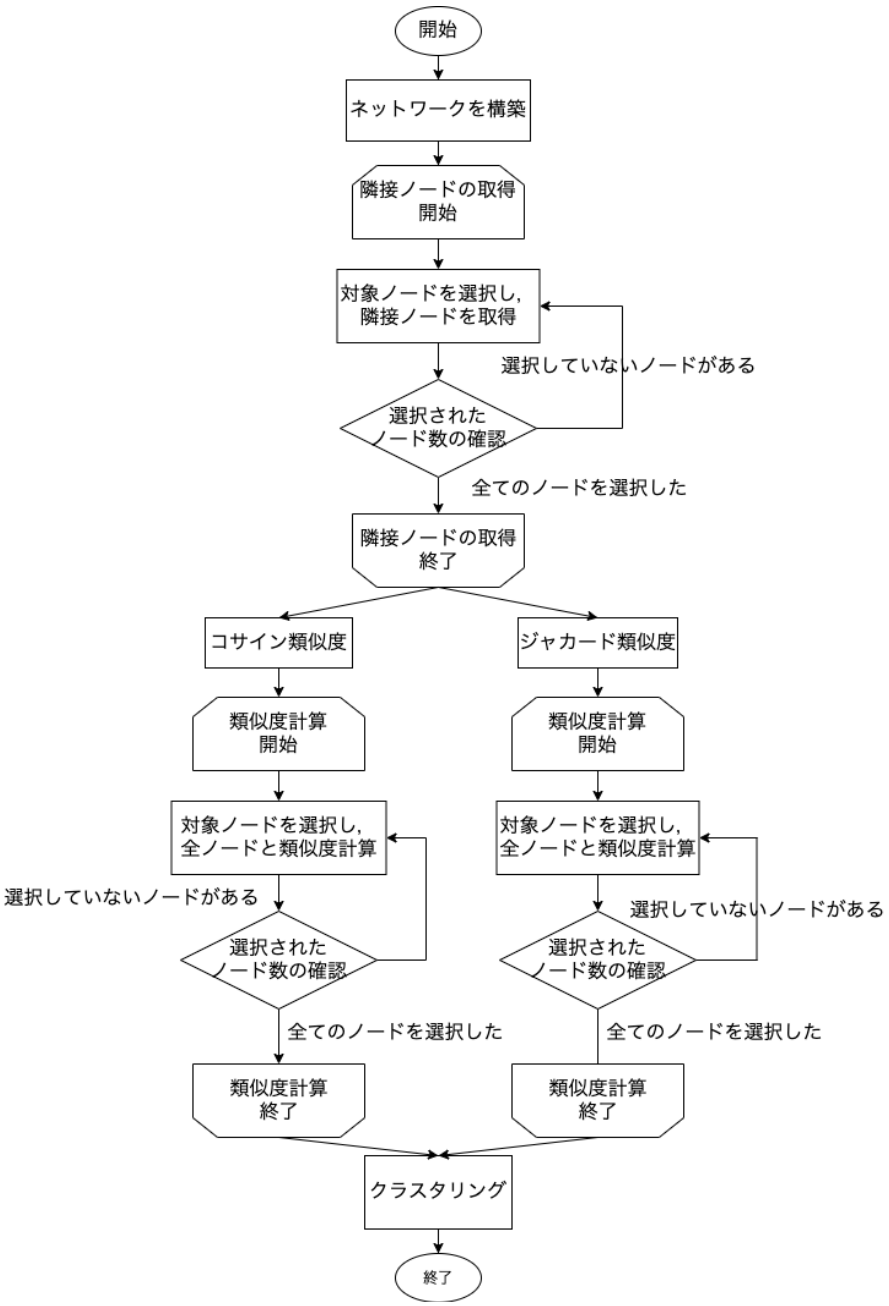

図 7：コサイン，ジャカード類似度計算のフローチャート

ノード群は”Yahoo Maps”と”Google Maps”と”Twitter”になる．次に，全サービスの総当たりの類似度を計算し，その結果を用いて類似機能サービス群にクラスタリングする．

第4章 ノード分散表現に基づくクラスタリング

4.1 グラフ埋め込み

グラフ埋め込みとはネットワークデータのノードやエッジ，サブグラフなどの分散表現を得る手法のことであり，特に skip-gram model をネットワーク構造に対して拡張した手法がいくつか提案されている. 本節では, skip-gram model のネットワーク構造への拡張手法を説明した後，skip-gram mode と同じ自然言語処理モデルである BERT のネットワーク構造への拡張手法を説明し，最後にネットワークデータのサンプリング手法として同一複合サービス優先のサンプリングについて説明する.

4.2 skip-gram model に基づくグラフ埋め込み

skip-gram model[6]とはニューラルネットワークモデルの１つで，単語分散表現を得る学習モデルのことである．単語の分散表現の獲得には，意味が近い単語は周辺単語も似ているというという学習により実現する．以下の２つの文を例として説明する.

- 私は琵琶湖線の南草津駅周辺に住んでいる
- 京都の山科には琵琶湖線と湖西線が走っている

上記の文の「南草津」と「山科」に注目したとき，これら２つの単語は共に「琵琶湖線」という単語から近い位置にあることから，これら２つの単語は比較的近い属性であると予測することができる．この時の「南草津」や「山科」のような単語を入力データ，「琵琶湖線」のような周辺単語を教師データとし，単語に対するその周辺単語の重みを学習させることで，各単語の分散表現を得ることができる．この手法で得た単語ベクトルに対して単純な代数演算を実行すると，例えば，$King$ベクトル$-$ Manベクトル$+$ $Woman$ベクトルが１つのベクトルになることが示され，これは Queen ベクトルという単語ベクトルに最も近くなる．他にも，skip-gram model と似た手法として cbow model[7]が存在する．skip-gram model が中心語から周辺単語を予測するのに対し，cbow model は周辺単語から中心単語を予測する．また，一般的には cbow model よりも skip-gram model の方が予測精度は高い. skip-gram model をネットワーク構造に対応させると，入力データは各ノード，教師データは各ノードのサンプリングで表すこと

ができる．

skip-gram model を用いたグラフ埋め込み技術として node2vec[8]が存在する．node2vec の大まかなアルゴリズムを図 8 に示す．node2vec を実行する事前準備として，任意のデータで構築したネットワークが必要になる．はじめに，構築したネットワークからサンプリングの始点となるノードを選択し，ランダムウォークで任意の長さのサンプリングを行う．全てのノードからサンプリングを

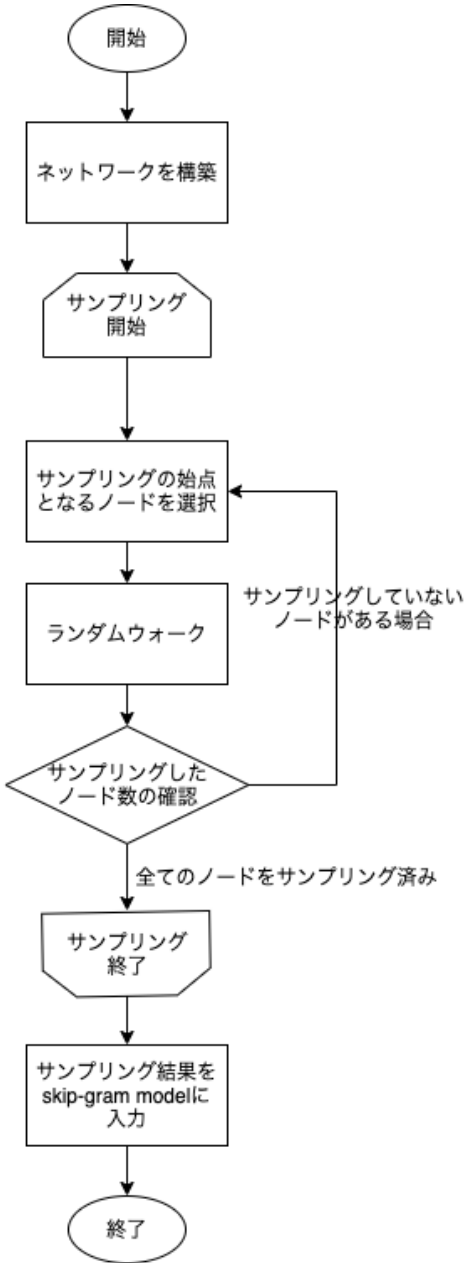

図 8：node2vec のフローチャート

行えた場合，それらを各ノードが空白で区切られたテキストデータとして扱い，skip-gram model し各ノードの分散表現を獲得する．node2vec の特徴として，ランダムウォークでノードのサンプリングを行う際，幅優先探索に似た探索と深さ優先探索に似た探索の度合いをハイパーパラメーターで制御することができる．ランダムウォークは静的エッジの重みに基づいて次のノードをサンプリングするが，ネットワークの構造を考慮し，検索手順をガイドして異なるタイプの近隣ネットワークを探索することができない．さらに，幅優先と深さ優先が競合または排他的ではない．そこでパラメーターreturn paremeter（以下，p と呼称）および In-out parameter（以下，q と呼称）を調整してサンプリングを行う．これらを使用した２次のランダムウォークを書きに定義する．またエッジ(t, x)を横断し，次に探索するノード$s_i(i = 1,2,3)$を考える（図 9）．d_{tx}はノード t と x 間の最短経路距離であり，d_{tx}は{1,2,3}のいずれかである．

パラメーターp は歩行中のノードを再訪する確率を制御する．このパラメーターを高い値（>max(q,1)）に設定することで既に探索したノードをサンプリングする可能性を低くすることができる．一方，低い値（<min(q,1)）にすることでサンプリングの開始ノードから離れにくくすることができる．

パラメーターq は幅優先探索に似た探索と深さ優先探索に似た探索の確率を

$$\alpha_{pq}(t,s) = \begin{cases} \frac{1}{p}\ if\ d_{ts} = 0 \\ 1\ if\ d_{ts} = 1 \\ \frac{1}{q}\ if\ d_{ts} = 2 \end{cases}$$

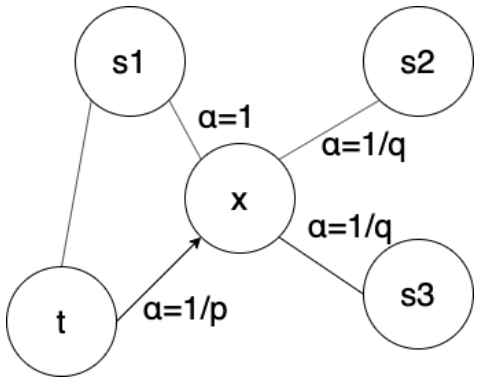

図 9：ノードの遷移例

制御する．q>1 の場合，ランダムウォークはノード t に近いノードに偏り，幅優先探索に似た探索を行うことができる．対照的に，q<1 の場合はノード t からさらに離れたノードを訪問する傾向があり，深さ優先探索に似た探索を行うことができる．しかし，幅優先探索に似た探索を重視しすぎると近いノードの情報しか得ることができず，一方で，深さ優先探索に似た探索を重視しすぎるとサンプリング範囲が大きく深くなるので，この２つのバランスが重要である．

4.3 BERT を用いたグラフ埋め込み

4.3.1 BERT

BERT は skip-gram mode と同じ自然言語処理モデルで，テキストデータを入力とする，文脈を理解することができる自然言語処理である．例えば，”parking on a hill with no carb”という文章があったとき，従来の自然言語処理では”no”が文章中のどの単語にかかるか理解することができなかった．しかし，BERT では”no”が”carb”にかかることを理解することができる．これは従来の自然言語処理とは異なり，BERT は全ての層で左右両方の文脈を共同で条件付けることにより、ラベルのないテキストから深い双方向表現を事前学習するよう設計されているからである．また，事前学習された BERT モデルを利用することで，少ない学習データでタスク固有のアーキテクチャを大幅に変更することなく，質問応答や言語推論などの幅広いタスクのための最先端のモデルを作成することができる．BERT は，ラベルなしデータを用いて行われる事前学習と，事前学習の重みを初期値としてラベルありデータで行われるファインチューニングの２ステップで学習される．また，事前学習では以下の２つのタスクを行う．

- Masked Language Model（以降，MLM と呼称）
- Next Sentence Model（以降，NSM と呼称）

１つ目の MLM について説明する．ここでは，入力された文章の１５％のトークンをランダムにマスクし，そのマスクされたトークンを予測するタスクを行う．この場合，マスクトークンに対応する最後の隠れベクトルは，語彙に対する出力の softmax に渡される．オートエンコーダーのノイズ除去とは対照的に，入力全体を再構築するのではなく，マスクされた単語を予測するだけである．しかし，ファインチューニングでは出てこない[MASK]トークンを事前学習で使用しているため，事前学習とファインチューニングの間にミスマッチが生じる問題がある．そこで，マスクするトークンを常に[MASK]トークンに置き換えるので

はなく，以下の置き換えパターンを適用することで解決する．また合わせて，それぞれのパターンを選択する確率を記載する．

- 80%：[MASK]トークンで置き換える
  例）my dog is big -> my dog is [MASK]
- 10%：ランダムで選んだトークンで置き換える
  例）my dog is big -> my dog is pen
- 10%：元々のトークンで置き換える
  例）my dog is big -> my dog is big

学習データ生成器は，予測のために入力文の１５％のトークンをランダムに選択する．そして，i 番目のトークンが選択された場合は，i 番目のトークンを８０％の確率で[MASK]トークンに置き換え，１０％の確率でランダムなトークンに置き換え，１０％の確率で変更されていない i 番目のトークンに置き換える．その後，置き換えられたトークンを周りの文脈から予測するタスクを行うことで，トークンに対応する文脈情報を学習することができる．

２つ目の NSM について説明する．ここでは，２つの文を選んでそれらが連続した文かどうかのタスクを行う．質問応答や自然言語推論などの下流の重要なタスクの多くは，言語モデリングでは直接捉えることのできない２つの文の関係性を理解することに基づいている．MLM において，単語に関しての学習を行うことができるが，文単位の学習を行うことはできない．そこで，２つの入力文に対して，その２つの文が隣り合う文であるかを当てる事前学習を行う．具体的には，文 A と文 B を選択する際に，５０％は B が実際に A に続く文（IsNext）とし，５０％はコーパスからのランダムな文（NotNext）とする．また学習を行う際に，２つの文を[SEP]というトークンで分け，isNext か NotNext を分類する[CLS]トークンが文頭に付与される．具体的な，タスク内容を図 10 に示す．例１では，”he went to [MASK] store”と，実際に続く文章である”he bought [MASK]”を[SEP]トークンで接続している．また１つ目の文章”he wen to [MASK] store”の文頭に[CLS]トークンを付与し，この場合は isNext でラベル付けしている．一方で，例２では，”he wen to [MASK] store と，コーパスからランダムで選択された”my dog is [MASK]”を[SEP]トークンで接続している．また例１と同様に”he went to [MASK] store”の文頭に[CLS]トークンを付与した後，この場合は NotNext でラベル付けしている．このように，隣り合った２つの文の予測タスクを行うことで，文単位での意味表現を学習している．

例１：
[CLS] he went to [MASK] store [SEP] he bought [MASK]
[CLS] -> isNext
例２：
[CLS] he went to [MASK] store [SEP] my dog is [MASK]
[CLS] -> NotNext

図 10：Next Sentence Prediction(NSP)のタスク例

最後に，ファインチューニングについて説明する．ファインチューニングとは，既存の学習済みモデルを転用して，新たなモデルを生成する方法のことである．BERT はトランスフォーマーの self-attention メカニズムにより，適切な入力と出力を入れ替えることで，少ない学習データをモデル化することができる．一般的には，双方向の cross-attention メカニズム適用前にテキストペアを独立してエンコードすることである．BERT は self-attention メカニズムを用いて連結されたテキストペアを符号化することで，２つの文の間の cross-attention メカニズムが効果的に含まれるため，代わりに self-attention メカニズムを用いてこれら２つの段階を統一する．各タスクについて，タスク固有の入力と出力を BERT に単純にプラグインし，全てのパラメータをエンドツーエンドで微調整している．事前学習で得られた文 A と文 B は以下のテキストペアに類似する．

- 言い換えにおける文のペア
- 含意における仮説-前提のペア
- 質問応答における質問とパッセージのペア
- テキスト分類やシーケンスタグ付けにおける縮退したテキスト-∅のペア

また出力では，トークン表現はシーケンスタグ付けや質問応答などのトークン・レベルのタスクのための出力層に供給され，[CLS]表現は entailment や感情分析などの分類のために出力される．

4.3.2 BERT の適用

BERT でテキスト埋め込みを行うための事前準備として，学習モデルを作成する必要がある．一般的なテキスト埋め込みでは上記の URL 先[1]に載せられてい

[1] https://github.com/google-research/bert

る事前学習モデルを用いるが，本手法で用いるテキストデータには一般的な単語で表せないサービス名が含まれるため，独自の事前学習モデルの用意が必要になる．BERT の学習モデルを作成するために，はじめに学習データの txt ファイルを create_pretraining_data.py を用いて tfrecord ファイルに変換する．実行するために必要なファイルは学習データ（txt 形式）と vocab.txt である．学習データは後述するネットワークのノードのサンプリング方法を用いて作成する．vocab.txt は学習データに出現する単語の一覧と学習の際に用いるトークンを改行区切りで記述したファイルである．トークンの種類とそれぞれの役割を下記に示す．

- [CLS]：入力された文章の頭につけるトークン
- [SEP]：文章の区切りにつけるトークン
- [MASK]：学習の工程で文章の一部を置き換える際に使用されるトークン
- [UNK]：vocab.txt に存在しない単語を表すトークン
- [PAD]：create_pretraining_data.py の引数の max_length の数値以下の文章を字埋めする際に用いられるトークン

最後に，run_pretraining.py を実行し，BERT の学習モデルを作成する．実行するために必要なファイルは create_pretraining_data.py で作成した tfrecord ファイルと BERT の設定を記述した bert_config.json が必要となる．

BERT を用いたグラフ埋め込みの大まかなアルゴリズムを図 11 に示す．node2vec と同様に，任意のデータで構築したネットワークからサンプリングの始点となるノードを選択し，ランダムウォークを行う．全てのノードからサンプリングを行えた場合，それらを空白で区切られたテキストデータを作成する．このテキストデータは前述の事前学習モデルを作成する際に用いる学習データにも用いられる．作成したテキストデータと事前学習モデルを extract_feature.py を用いてテキスト埋め込みを行うことで，テキストデータの各単語（ネットワークのノード名）の分散表現を獲得することができる．しかし，ここで得られる単語ベクトルは skip-gram model のような 1 単語 1 ベクトルではなく，1 単語複数ベクトルとなる．これは BERT の性質上，単語ベクトルは文章に依存するため，単語が出現する箇所によりベクトルが変化するからである．1 単語 1 ベクトルの形式に変換するために，同じ単語間のベクトル，異なる単語間のベクトルの差を調査した．その結果，同じ単語間のベクトルはコサイン類似度 0.8 以上を示し，異なる単語間のベクトルはコサイン類似度 0.7 以下を示した．以上より，同

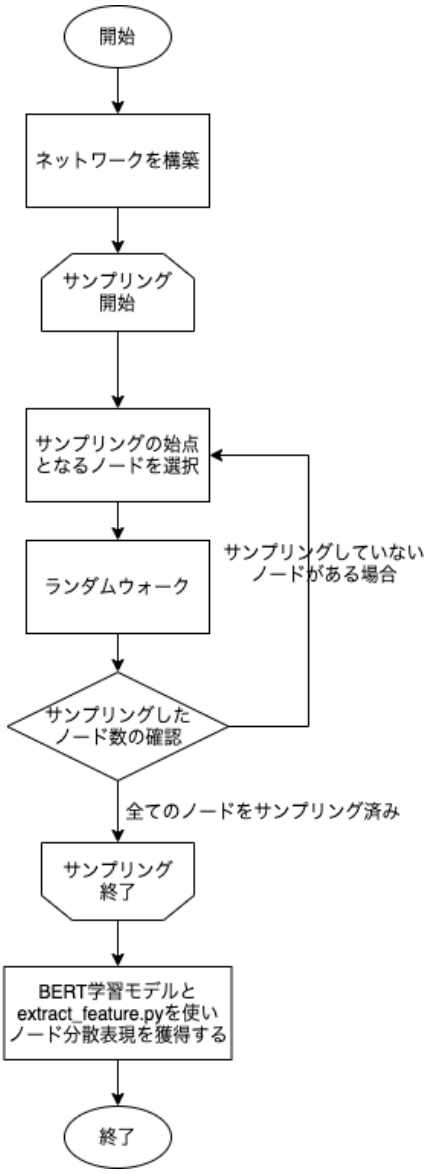

図 11：BERT を用いたグラフ埋め込みのフローチャート

じ単語間のベクトルは近く，異なる単語間のベクトルは同じ単語間よりも遠いベクトルになることが判明した．よって，本研究では，1 単語に生成される複数のベクトルの平均ベクトルを，その単語のベクトルとして扱う．

4.4 同一複合サービス優先のサンプリング

node2vec のサンプリング手法とは別のアプローチとして，訪問中ノードと同一複合サービスに組み合わされているノードを優先して探索する手法を説明す

る．実験で使用するネットワークは複合サービスにより組み合わされたことのある原子サービスの連携関係でネットワークを構築しており，それぞれの原子サービスがどの複合サービスで組み合わされたのか表されていない．また，同一複合サービスで組み合わされているノード同士は完全ネットワークになり，構築されるネットワークはそれぞれのノードが非常に密接するネットワークになるため，node2vec のサンプリング手法ではそれぞれのノードの特徴を取得するのは非常に難しいと考えられる．そこで，サンプリング結果に複合サービス情報を反映させるために，同一複合サービスを優先するサンプリング手法を行う．このサンプリング手法の大まかなアルゴリズムを図 12 に示す．任意のネットワークを構築した後，はじめにサンプリングの始点になるノードをランダムで選択する．次に，始点ノードが含まれる複合サービスの組み合わせ事例からランダムで優先して探索する複合サービス名を選択する．続いて，訪問先ノードをランダムで選択し探索する．この際，訪問先ノードが前述した優先して探索する複合サービスに組み合わされるサービスか確認する．最後に，優先して探索する複合サービスに組み合わされるサービスを全て探索し終えた後，任意の学習モデルにサンプリング結果を入力することで各ノードの分散表現を得ることができる．例えば，skip-gram model で学習させる場合は，サンプリング結果の各ノードを空白で区切った文字列にし，テキストデータとして skip-gram model で学習させることができる．

このサンプリング手法のサンプリングの特徴を図 13 のネットワークを用いて説明する．これは 3 つの複合サービスの組み合わせデータから構築されており，組み合わせのパターンは[サービス A，サービス B，サービス C，サービス D]と[サービス C，サービス Y]と[サービス X，サービス Y，サービス Z]である．また，図 13 を用いて作成した同一複合サービスを優先してサンプリングする手法のサンプリング結果を図 14 上部に示す．また比較対象として，図 14 下部にランダムウォークのサンプリング結果を示す．図 14 のランダムウォークのサンプリング結果は図 13 のネットワークにおいて，サービス A を始点として，サービス C，サービス Y，サービス Z の順で探索を行った結果である．ランダムウォークやランダムウォークをベースとするサンプリング手法は複数の複合サービスを横断するように探索するので，サンプリング結果にはネットワーク上における各ノードの前後関係しか表れていない．これにより，全てのノードが密接に接続される複合サービスに組み合わされたサービスの連携関係ネットワークに

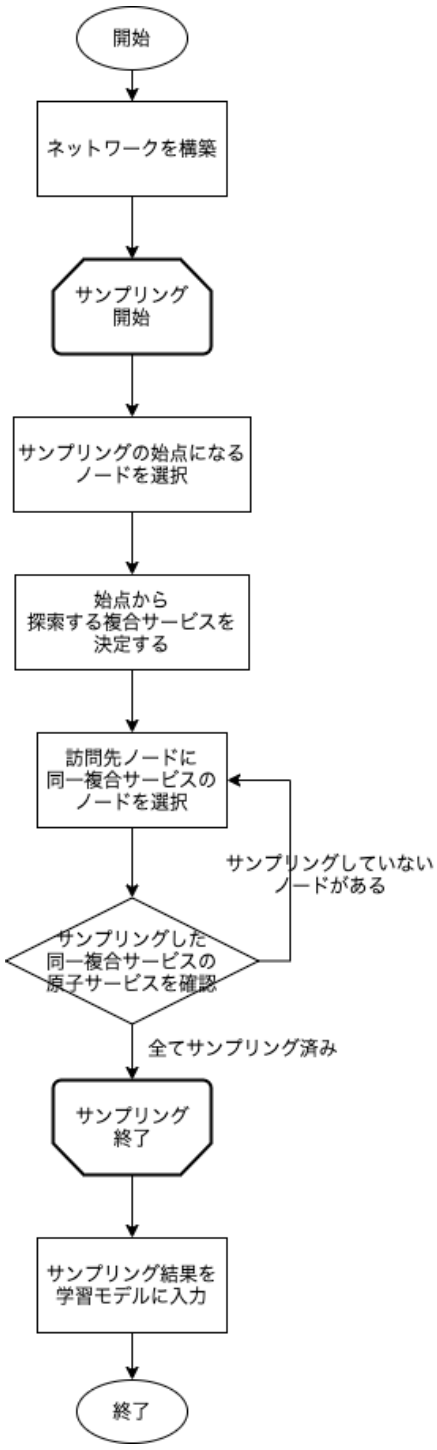

図 12：同一複合サービスを優先して探索する手法のフローチャート

おいて，全てのノードが近いベクトルで表され，類似機能サービスごとにクラスタリングするのは困難であると考えられる．そこで，同一複合サービスを優先す

るサンプリングを行う．図 14 の同一複合サービスを優先するサンプリング結果は図 13 のネットワークにおいて，サービス A を始点として，サービス C，サービス B，サービス D の順で探索を行った結果である．同一複合サービスを優先するサンプリング結果は複合サービスごとの塊が出現する特徴がある．これにより，複合サービスによる各サービスの組み合わせ情報をサンプリング結果に表すことができる．また，このサンプリング結果を skip-gram model で学習させると，似た組み合わせ同士のノード間のベクトルが近くなることが期待できる．4.2 で述べたように，skip-gram model は意味が近い単語同士は周辺単語も似ている学習を行い，これをネットワークのノードのサンプリング結果で解釈すると，周辺ノードが似ているノード同士はベクトルが近くなると解釈できる．また，同一複合サービスを優先するサンプリング手法において，周辺ノードは同一複合サービスで組み合わされたノード群になるので，似た構成を持つ複合サービスに組み合わされたノード同士のベクトルは近くなることが期待できる．

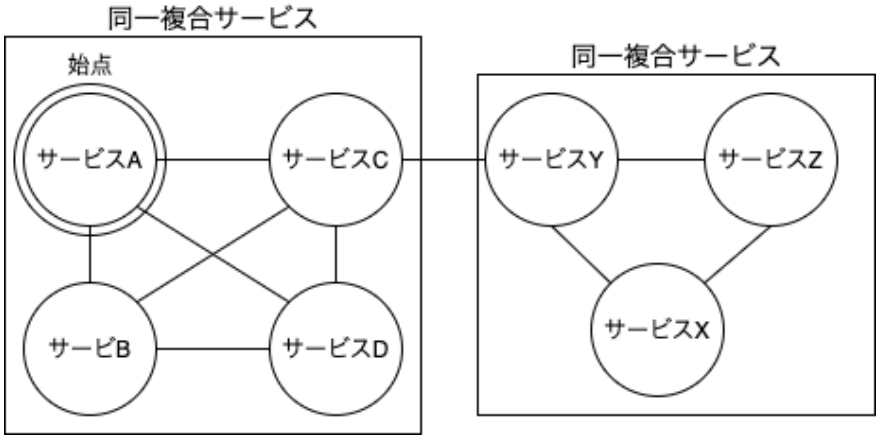

図 13：サンプリングを行うネットワーク例

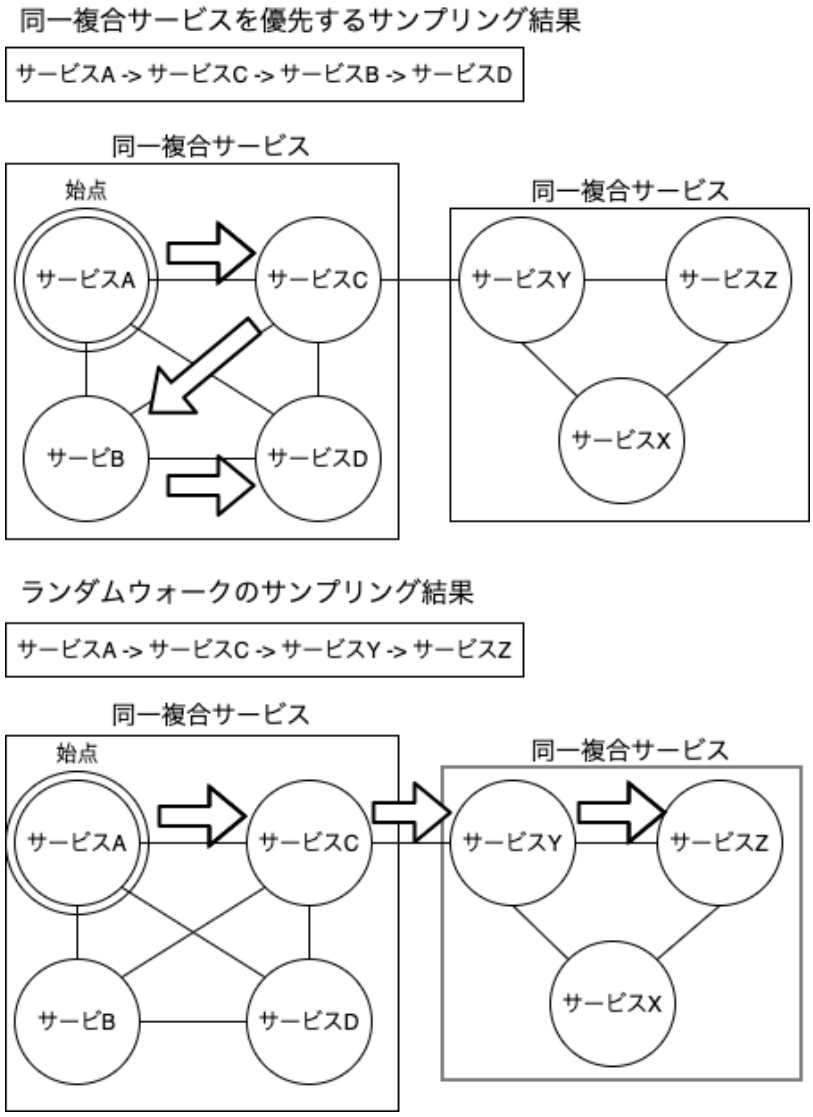

図 14：各手法のサンプリング例

第5章 評価

第 4 章で説明したグラフ埋め込みの手法を用いて複合サービスに組み合わされる原子サービスを類似機能サービスグループにクラスタリングする．本章では，はじめに実験データを説明する．次に，それぞれの手法を比較する際に使用する評価指標について説明する．続いて，ベースラインとなるサービス説明文に基づくクラスタリングについて説明する．次に，生成クラスタと正解クラスタの対応づけのパターンについて説明する．最後に，node2vec の探索パラメーター別のクラスタリング精度，node2vec のサンプリング戦略別のクラスタリング精度，BERT を用いたグラフ埋め込みにおけるクラスタリング精度，高精度のサンプリング戦略と埋め込み技術を用いたクラスタリング精度それぞれを説明した後，その結果について考察する．

5.1 評価手法

5.1.1 実験データ

実験データは Programmableweb[1]に掲載されている複合サービスデータ 7606

```
["items", {"vid":"81386", "uid":"61422", "title":"News Map", "type:"mashup"},
        ["fieldapi",
                ["target_id", "62687"],
                ["target_id", "62697"]],
        ["fieldapiendpoint"], ["filedmashupurl"]]
```

図 15：Programmableweb から取得したデータの例

```
["items", {"vid":サービスID, "uid":ユーザI D, "title":サービス名,
"type:マッシュアップなら"mashup", WEBサービスなら"api"},
        ["fieldapi",
                ["target_id", 組み合わされたサービスI D],
        ["fieldapiendpoint"], ["filedmashupurl"]]
```

図 16：Programmableweb から取得したデータのフォーマット

[1] https://www.programmableweb.com

件を用いる．取得したデータには，複合サービスと組み合わされている原子サービスが記述されている（図 15）．このフォーマットの定義を図 16 に示す．取得したデータから target_id のみを取得し，ノードを target_id とし，エッジを組み合わされたことのあるサービス間で繋ぐグラフを構築する構築したネットワークの全体像を図 17 に示す. また, 作成したネットワークは図 18 に示す通り，スケールフリーネットワークになっている．

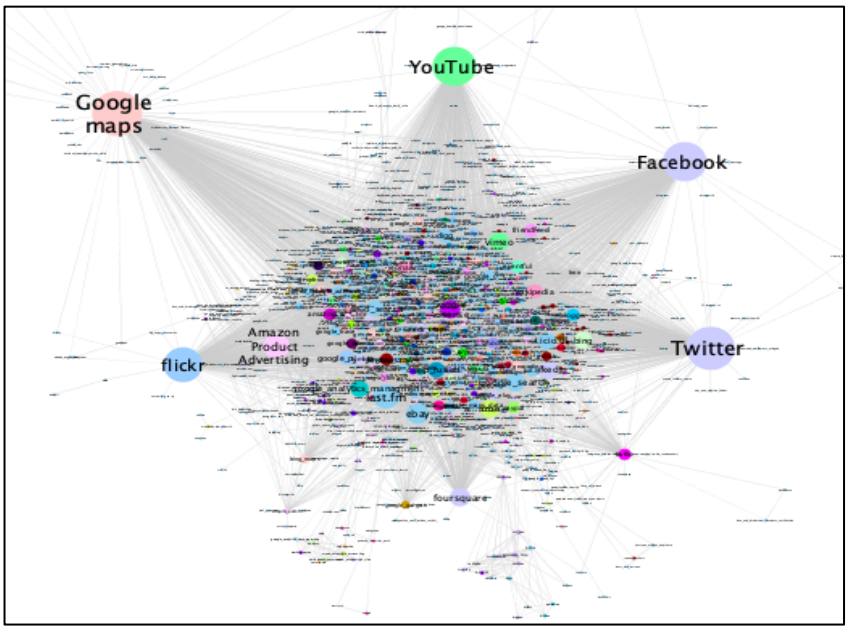

図 17：作成したサービス連携関係ネットワーク

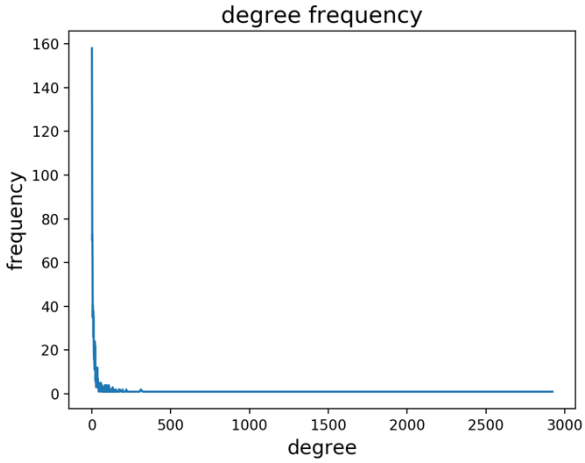

図 18：次数分布

5.1.2 評価指標

各手法により生成されるクラスタリング結果の有効性を評価する指標として以下の purity 値を求めた[9].

$$Purity = \frac{1}{N}\Sigma_i \max_j \left|sc_i \cap c_j\right| \quad (1)$$

数式(1)では，各手法において生成したクラスタ sc と正解クラスタ c の要素を比較し，最も共通項の多い組み合わせを作り，共通項の総和をサービス総数 N で割っている．これにより，対象となる全サービスの何割が正しいクラスタに所属しているか数値的に判断することができる．ここでの，正解クラスタは Programmableweb に掲載される人手で付与された各サービスのカテゴリ情報を基に作成している．

5.1.3 サービス説明文に基づくクラスタリング

グラフ埋め込みを用いたサービスクラスタリング手法の有効性を示す基準として，サービス説明文に基づくサービスクラスタリング手法を用いる．一般的な文章間の類似度を計算する際に用いられる技術は word2vec や TF-IDF などであるが，本研究では，それらよりも予測精度の高い BERT を用いて文章間の類似度計算を行う．サービス説明文の類似度計算を行うための手順は以下の通りである．

1. サービス説明文の分散表現の獲得
2. サービス説明文間の類似度計算

１つ目のサービス説明文の分散表現の獲得は BERT の事前学習モデルを用いて，各説明文の分散表現を獲得した．事前学習モデルは，10 億組以上の大規模かつ多様な学習用データセットで学習された all-mpnet-base-v2[1]を用いる．サービス説明文を入力とし，上記の事前学習モデルを使うことで７６８次元の分散表現を獲得することができる．

２つ目のサービス説明文間の類似度計算では，3.2 で用いたコサイン類似度を用いる．２つのサービス説明文の７６８次元の分散表現を計算することで，これら２つのサービス説明文間の類似度を計算することができる．計算結果の類似度が高い場合は，２つのサービス説明文は似た内容と判断することができ，該当する２つのサービスは似た機能であると判断できる．一方で，計算結果が低い場

[1] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

合は，２つのサービス説明文は違う内容であると判断でき，該当する２つのサービスは異なる機能であると判断できる．

上記の手順により得られる総当たりのサービス説明文間の類似度を k-means クラスタリングを用いることで，類似機能サービス群にクラスタリングする．

5.1.4 生成クラスタと正解クラスタの対応付けのパターン

5.1.2 の purity 値を計算する際の生成クラスタと正解クラスタの対応付けにおいて，以下の４つのパターンが想定される．

- pattern1 : 正解クラスタを Primary カテゴリのみで作成し，正解クラスタが重複しない対応付けを行う
- pattern2 : 正解クラスタを Primary カテゴリと Secondary カテゴリで作成し，正解クラスタが重複しない対応付けを行う
- pattern3 : 正解クラスタを Primary カテゴリのみで作成し，正解クラスタが重複する対応付けを行う
- pattern4 : 正解クラスタを Primary カテゴリと Secondary カテゴリで作成し，正解クラスタが重複する対応付けを行う

これら４つのパターンに分類される理由は以下の通りである．

- 各サービスのカテゴリ情報が複数付与されている場合があるため
- 複数の生成クラスタが共通の正解クラスタと対応付けされる場合があるため

はじめに，各サービスのカテゴリ情報が複数付与されている場合があることについて説明する．5.1.2 で述べたように，正解クラスタは Programmableweb から取得した人手で付与されたカテゴリ情報を基に作成する．Programmableweb にカテゴリ情報を登録する際に，主なカテゴリを表す Primary カテゴリと他に想起されるカテゴリを表す Secondary カテゴリを付与することができるため，１つのサービスに複数のカテゴリが付与されている場合が生じる．例えば，地図情報サービスとして有名な Google Maps API は Primary カテゴリに”Mapping”，Secondary カテゴリに”Viewer”が付与されている．また，Secondary カテゴリは複数付与することができるため，１つのサービスに３つ以上カテゴリが付与されていることがある．Programmableweb のカテゴリ情報はカテゴリ情報登録者による影響を受けやすいため，Primary カテゴリのみで正解クラスタを作成するパターンと，Primary カテゴリと Secondary カテゴリで正解クラスタを作成するパターンに分けた．

続いて，複数の生成クラスタが共通の正解クラスタと対応付けされる場合があることについて説明する．purity 値の計算を行う工程で，最も共通する要素が多い生成クラスタと正解クラスタの組み合わせを見つける工程がある．生成される各クラスタが類似機能サービスでクラスタ化されている時，正解クラスタは唯一に決定する．しかし，類似機能サービスのクラスタが複数生成された場合，共通の正解クラスタと対応付けされる場合が生じる．類似機能サービスクラスタが複数生成され，対応付けされる正解クラスタが重複しないように対応付けを行った場合，２つ目以降の類似機能サービスクラスタは要素数が次に多いカテゴリで対応付けされることが考えられる．この理由により，正解クラスタが重複する対応付けのパターンと，正解クラスタが重複しない対応付けのパターンに分けた．

5.1.5 node2vec の探索パラメータ別のクラスタリング精度

node2vec の探索パラメーター変化によるクラスタリング精度比較を行う．node2vec の探索パラメーターは探索の長さ（以下，walk_length と呼称），１つのノードからサンプリングする回数（以下，num_walk と呼称），4.2 で説明した node2vec のランダムウォークに関するパラメーターp と q の４つである．これらの探索パラメーターを変化させ，それぞれのパラメーターがいくつの時にクラスタリング精度が高くなるか検証した．

5.1.6 node2vec のサンプリング戦略別のクラスタリング精度

node2vec のサンプリング手法戦略によるクラスタリング精度比較を行う．比較するサンプリング手法は以下の通りである．

- 5.1.5 で最もクラスタリング精度が高かった node2vec のパラメーターを使ったサンプリング手法
- 同一複合サービスを優先して探索するサンプリング手法

これらのサンプリング手法を比較して，どのサンプリング手法が，クラスタリング精度が高くなるか検証した．

5.1.7 BERT を用いたグラフ埋め込みにおけるクラスタリング精度

BERT を用いたグラフ埋め込みにおけるクラスタリング精度の検証を行う．skip-gram model を用いたグラフ埋め込みとクラスタリング精度と比較することで，BERT を用いた場合の有効性を検証する．また，ネットワークのノードのサンプリング手法は同一複合サービスを優先して探索するサンプリング手法を用いる．

5.1.8 高精度のサンプリング戦略と埋め込み技術を用いたクラスタリング精度

5.1.6 と 5.1.7 で高精度と評価されたサンプリング戦略と埋め込み技術を用いたグラフ埋め込みのクラスタリング精度を評価する．比較手法として，第 3 章で述べた近傍に基づくクラスタリングと，5.1.3 で述べた BERT を用いたサービス説明文に基づくクラスタリングを用いる．これらの手法を比較することで，グラフ埋め込み手法の有効性を評価する．

5.2 結果

5.1 で説明した評価指標を用いてグラフ埋め込みを用いて複合サービスに組み合わされる原子サービスを類似機能サービスグループにクラスタリングした結果を以下に示す．また提示する図や表に記述される pattern1 などは 5.1.4 で説明したものである．

1. node2vec の探索パタメータ別のクラスタリング精度

はじめに，次元数を変化させた結果を図 19 に示す．全ての次元数において，purity 値は大きく変化しないが，pattern1 と pattern2 において，次元数= 8 の時に purity 値が明らかに高くなっていることが見て取れる．

次に，パラメーターwalk_length を変化させた結果を図 20 に示す．図 20 の pattern2，3，4 から walk_length が 30 と 100 の時に purity 値が高くなることがわかる．また，walk_length が小さい数値から比較した時，walk_length=30 の時にピークを迎えた．

続いて，パラメーターnum_walk を変化させた結果を図 21 に示す．図 21 の結果から，全ての pattern において，num_walk=50 の時にピークになり，それ以降は purity 値に大きな変化は見られなかった

次に，パラメーターp の値を変化させた結果を図 22 に示す．p は探索時に 1 つ前のノードを探索する確率 1/p の値である．図 22 では p=0.1，0.5，1，2，10 の時の purity 値を示している．全ての場合において，purity は大きく変わらないが，pattern2 において，p=0.1 の時に purity 値が最も高くなった．

最後に，パラメーターq を変化させた結果を図 23 に示す．q は探索方法を決める数値で，$q > 1$ の時に幅優先探索に似た探索，$q < 1$ の時に深さ優先探索に似た探索を行う．図 23 より，$q > 1$ よりも $q = 1$，$q < 1$ の時に purity が高くなることが見て取れる．

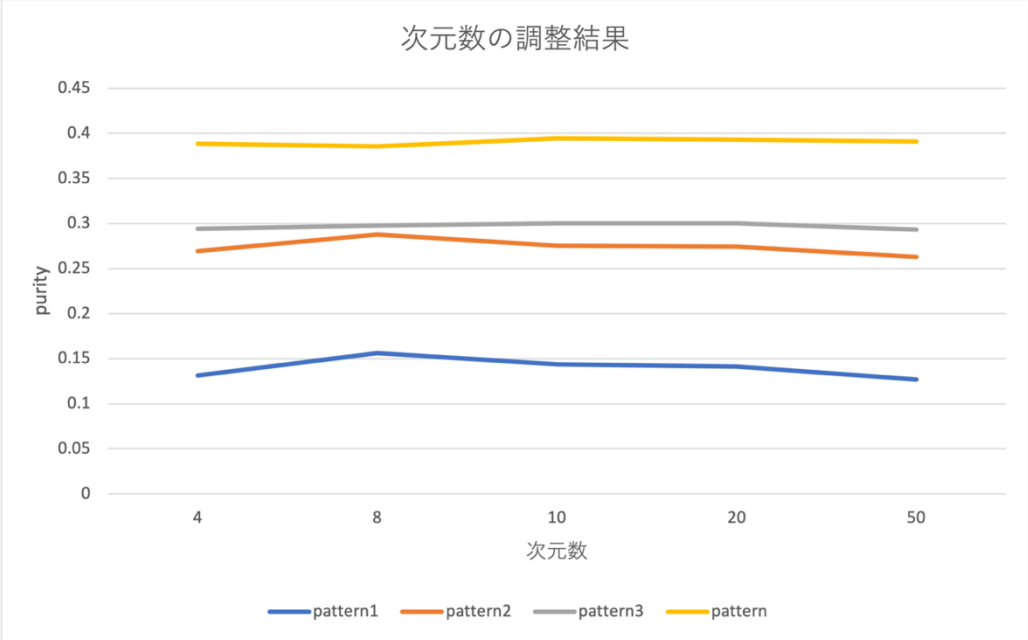

図 19：node2vec の次元数を変化させた purity 値の結果

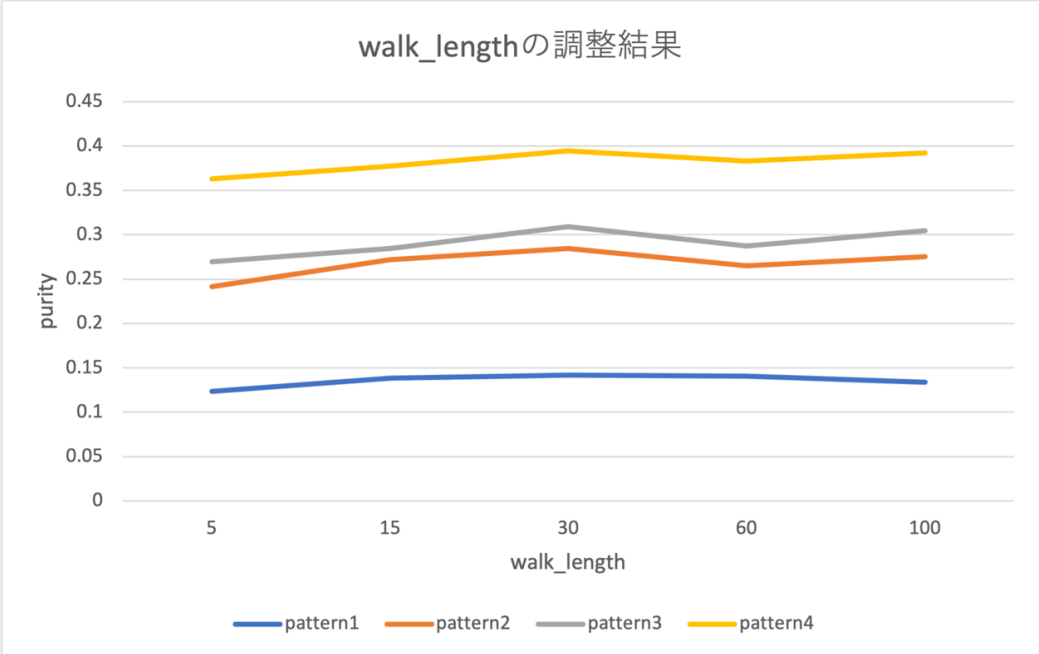

図 20：walk_length を変化させた purity 値の結果

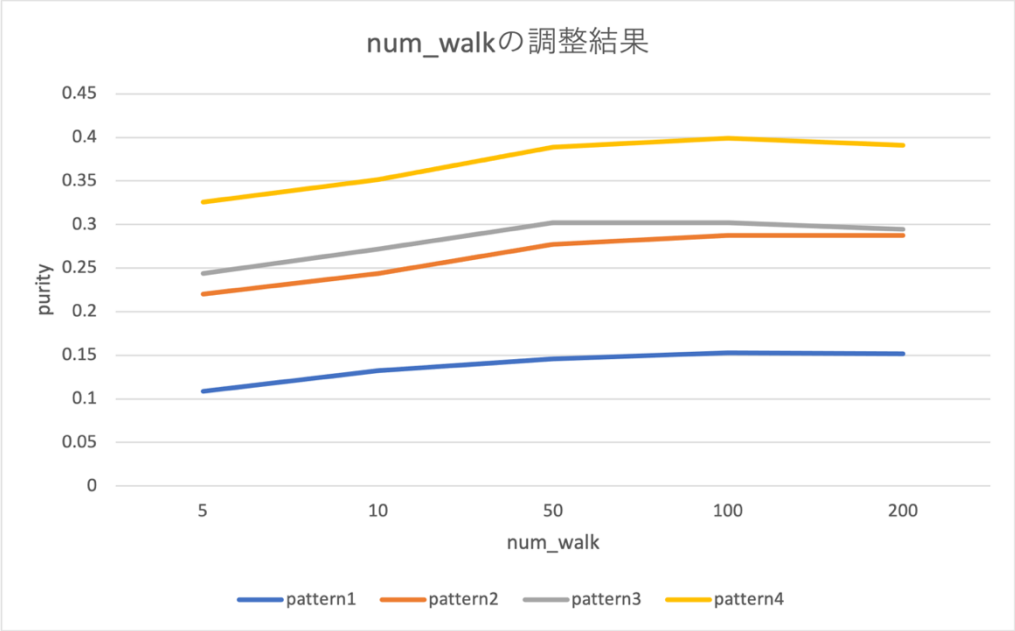

図 21：num_walk を変化させた purity 値の結果

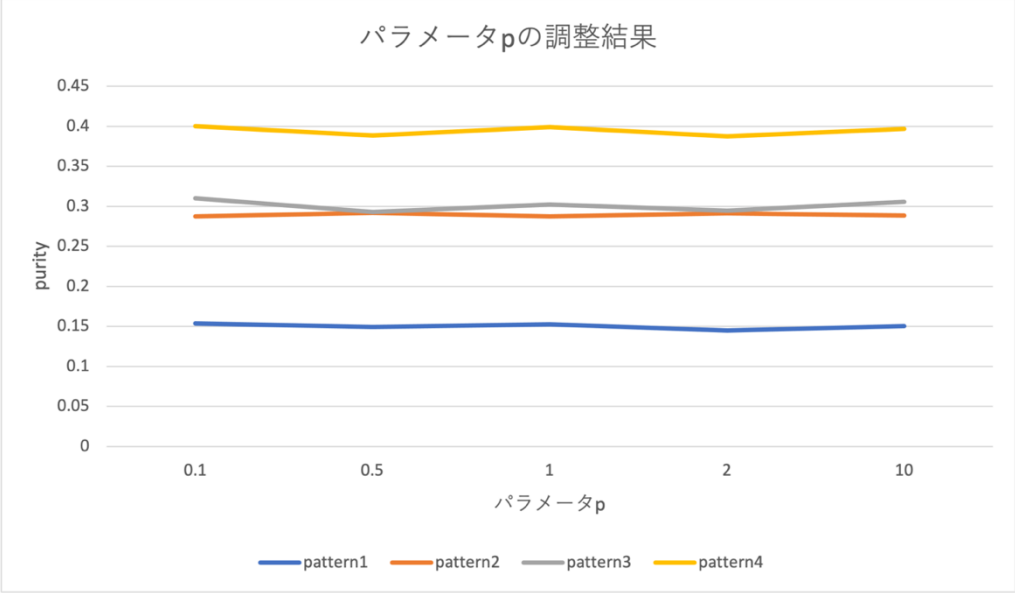

図 22：パラメーターp（1 つ前の訪問ノードに戻る確率）を変化させた purity 値の結果

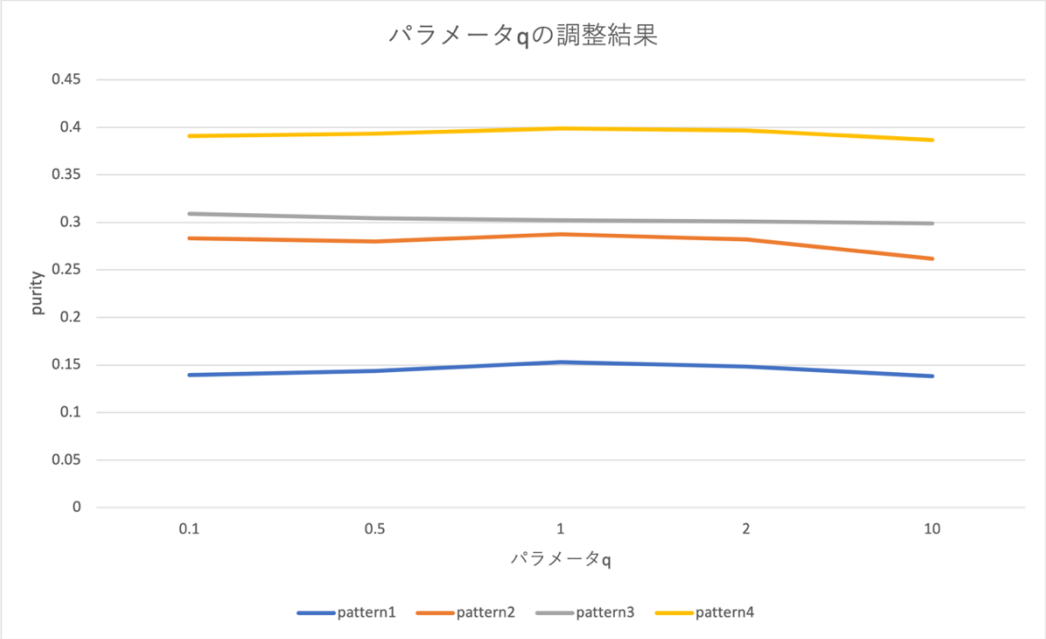

図 23：パラメーターq（q<1：深さ優先探索に似た探索，q>1：幅優先探索に似た探索を行う確率）を変化させた purity 値の結果

2. node2vec のサンプリング戦略別のクラスタリング精度

pattern ごとの node2vec のサンプリング手法，同一複合サービスを優先するサンプリング手法の purity 値，適合率の平均を表 2 表 3 表 4 表 5 に示す．また，node2vec のサンプリング手法はランダムウォークがベースとなっているので，表ではランダムウォークと記載する．

表 2 の pattern1 の結果において，同一複合サービスを優先してサンプリング手法の purity 値が最も高くなった（purity=0.1749）．また適合率の平均においても同一複合サービスを優先してサンプリングする手法が最も数値が高くなった（適合率の平均=0.1997）．

表 3 の pattern2 の結果において，同一複合サービスを優先するサンプリング手

表 2：pattern1 における各サンプリング手法のクラスタリング精度

	ランダムウォーク	同一複合を優先するサンプリング
purity	0.1348	0.1749
適合率の平均	0.1848	0.1997

表 3：pattern2 における各サンプリング手法のクラスタリング精度

	ランダムウォーク	同一複合を優先するサンプリング
purity	0.2494	0.2752
適合率の平均	0.3269	0.2964

表 4：pattern3 における各サンプリング手法のクラスタリング精度

	ランダムウォーク	同一複合を優先するサンプリング
purity	0.2943	0.3278
適合率の平均	0.39	0.349

表 5：pattern4 における各サンプリング手法のクラスタリング精度

	ランダムウォーク	同一複合を優先するサンプリング
purity	0.3797	0.4267
適合率の平均	0.4903	0.4493

法の purity 値が最も高くなった（purity=0.2752）．一方で，適合率の平均においては node2vec のサンプリング手法が最も高くなった（適合率の平均=0.3269）．
表 4 の pattern3 の結果において，同一複合サービスを優先するサンプリング手法の purity 値が最も高くなった（purity=0.3278）．一方で，適合率の平均においては node2vec のサンプリング手法が最も高くなった（適合率の平均=0.39）．
表 5 の pattern4 の結果において，同一複合サービスを優先するサンプリング手法の purity 値が最も高くなった（purity=0.4267）．一方で，適合率の平均においては node2vec のサンプリング手法が最も高くなった（適合率の平均=0.4903）．

3. BERT を用いたグラフ埋め込みにおけるクラスタリング精度

BERT を用いたグラフ埋め込みと，比較手法である skip-gram model を用いたグラフ埋め込みの purity 値と適合率の平均の結果を表 6 表 7 表 8 表 9 に示す．

表 6：pattern1 における各埋め込み手法のクラスタリング精度

	skip-gram model	BERT
purity	0.1749	0.1033
適合率の平均	0.1997	0.1022

表 7：pattern2 における各埋め込み手法のクラスタリング精度

	skip-gram model	BERT
purity	0.2752	0.191
適合率の平均	0.2964	0.1721

表 8：pattern3 における各埋め込み手法のクラスタリング精度

	skip-gram model	BERT
purity	0.3278	0.2089
適合率の平均	0.349	0.2315

表 9：pattern4 における各埋め込み手法のクラスタリング精度

	skip-gram model	BERT
purity	0.4267	0.2741
適合率の平均	0.4493	0.2742

全ての対応付けのパターンにおいて，skip-gram model の方が BERT よりも purity 値や適合率の平均が高くなることが見て取れる．

4. 高精度のサンプリング技術と埋め込み技術を用いたクラスタリング精度

同一複合サービスを優先するサンプリング手法をとり，skip-gram model を用いるグラフ埋め込み手法と，比較手法であるコサイン類似度またはジャカード類似度を用いた近傍に基づくクラスタリング, BERT を用いたサービス説明文に基づくクラスタリングの purity 値と適合率の平均の結果を表 10 表 11 表 12 表 13 に示す．

表 10 の pattern1 の結果において，グラフ埋め込み手法の purity 値が最も高くなった（purity=0.1749）．また，適合率の平均においてもグラフ埋め込み手法が最も高くなった（適合率の平均=0.1997）．

表 11 の pattern2 の結果において, コサイン類似度を用いた近傍に基づくクラスタリング手法の purity 値が最も高くなり（purity=0.2752），次に高いのはグラフ埋め込み手法であった（purity=0.2752）．一方で，適合率の平均においてはグラフ埋め込み手法が最も高くなった（適合率の平均=0.2964）

表 12 の pattern3 の結果において, BERT を用いたサービス説明文に基づくクラスタリング手法の purity 値が最も高くなり（purity=0.3573），次に高いのはグラフ埋め込み手法であった（purity=0.3278）．また，適合率の平均においても最も

表 10：pattern1 における各クラスタリング手法の精度

	同一複合サービスを優先するサンプリング + skip-gram model	コサイン	ジャカード	説明文
purity	0.1749	0.1359	0.1483	0.1191
適合率の平均	0.1997	0.1393	0.1586	0.1733

表 11：pattern2 における各クラスタリング手法の精度

	同一複合サービスを優先するサンプリング + skip-gram model	コサイン	ジャカード	説明文
purity	0.2752	0.282	0.2651	0.2505
適合率の平均	0.2964	0.2042	0.2093	0.2663

表 12：pattern3 における各クラスタリング手法の精度

	同一複合サービスを優先するサンプリング + skip-gram model	コサイン	ジャカード	説明文
purity	0.3278	0.2955	0.2865	0.3573
適合率の平均	0.349	0.2463	0.2452	0.3832

表 13：pattern4 における各クラスタリング手法の精度

	同一複合サービスを優先するサンプリング + skip-gram model	コサイン	ジャカード	説明文
purity	0.4267	0.3853	0.3741	0.4696
適合率の平均	0.4493	0.3063	0.3052	0.5042

高い手法はサービス説明文に基づく手法であり（適合率の平均=0.3832），次にグラフ埋め込み手法が高くなった（適合率の平均=0.349）.

表 13 の pattern4 の結果においても patter3 と同様に，purity 値と適合率の平均が最も高いのはサービス説明文に基づく手法であり，次に高いのはグラフ埋め込み手法であった.

5.3 考察

本節では，5.2 節で提示した結果を受けて node2vec の探索パラメーター変化によるクラスタリング精度に関して考察を与えた後，node2vec のサンプリング戦略別のクラスタリング精度に関して考察を与え，最後に BERT を用いたグラフ埋め込みにおけるクラスタリング精度に関して考察を行う. 最初に, node2vec

の探索パラメーター変化によるクラスタリング精度の考察について述べる．

はじめに，次元数についての考察を述べる．図 19 より，次元数ごとのクラスタリング精度は大きく変わらないが，pattern1 の対応付け方法である正解クラスタを Primary カテゴリのみで作成し，正解クラスタが重複しない対応付けにおいて，次元数=8 の時に明らかに purity 値が上がっていた．このことから，妥当な次元数は８であると定めた．一般的な単語分散表現の次元数は 768 次元であるが，使用しているネットワークが 1312 ノードであることを踏まえると極端に少ない次元数ではないと考えられる．しかし，この次元数は現在使用しているネットワークを対象としたときの妥当な次元数である．従って，今後複合サービス事例が増えることにより，ネットワークノードが増えることが予想されるので，妥当な次元数を適宜検証する必要があると考えられる．

次に，walk_length と num_walk についての考察を述べる．図 20 図 21 の結果より，walk_length と num_walk は数値が高いほどクラスタリング精度は上がった．しかし，それぞれの数値が大きくなるほどクラスタリング精度の変化は小さくなっていることもわかる．このことから，walk_length=30 の時，num_walk=50 の時点でおおよそのネットワークの特徴を学習できていたと考えられる．無限大に数値を増やした場合，クラスタリング精度に変化か出なくなる，または下がることが考えられ，また数値が増えるほど学習時間がかかるので，walk_length=30，num_walk=50 の数値は妥当であると考える．

続いて，パラメーターp について考察を述べる．パラメーターp はサンプリング時に１つ前のノードに戻る確率に関係する数値である．図 22 の結果より，パラメーターp を変化させたことによるクラスタリング精度の影響は少ないことがわかる．これは図 17 で示したように，本研究におけるネットワーク構造が非常に密な構造であることが関係すると考えられる． １つ前のノードとは現在参照中のノードから見た時の１つ前のノードであり，別のノードに遷移すると１つ前のノードの対象が変化する．これにより，密なネットワークにおいて，２ホップ目以降にサンプリング済みのノードを訪問する可能性が高いので，パラメーターp 変化によるクラスタリング精度への影響は少ないと考えられる．

最後に，パラメーターq について考察を述べる．図 23 の結果より，パラメーターq の調整によるクラスタリング精度は少ない変化であるが，$q = 1$ または $q < 1$ であるときに高くなることがわかる．これはランダムウォークまたは深さ優先探索に似た探索を行った時にクラスタリング精度が上がったことを表す．少な

いクラスタリング精度の変化であった理由は実験で用いている密なネットワークにあると考えられる．深さ優先探索やランダムウォークを行うサンプリングを行なっても，一度訪問したノードを再訪する可能性が高く，パラメーターに関係なく幅優先探索に近いサンプリング方法になっていたと考えられる．しかし，このような密なネットワークであっても幅優先探索に似た探索の時のクラスタリング精度が低かったことから，一度の探索で周辺ネットワークを網羅するサンプリング方法よりも，深さ優先のような遠くのノードを探索するサンプリング方法をとることで，より多くノードとの関係を表した方が有効であると考えられる．

次に，node2vec のサンプリング戦略別のクラスタリング精度について述べる．表 2 表 3 表 4 表 5 の結果より，node2vec と同一複合サービスを優先するサンプリングの purity 値を比較した時，いずれの対応付けパターンにおいても同一複合サービスを優先するサンプリングの方が purity 値は高くなることがわかる．これはサンプリング結果に組み合わされられた複合サービスの情報を反映した結果であると考えられる．実験で用いているネットワークはどの原子サービス同士がどのタイミングで組み合わせられたか表されておらず，node2vec のようなランダムウォークベースのサンプリング方法では隣接ノードは同一複合サービスで組み合わせられたと学習される可能性があった．そこで，同一複合サービスを優先するサンプリング手法により，サンプリング方法に同一複合サービスごとの塊を作り skip-gram model で学習することで，似たようなサンプリングの塊を持つ原子サービス同士は近い関係であると学習させることができたと考えられる．しかし，適合率の平均においては，pattern1 以外の場合は同一複合サービスを優先するサンプリング手法より node2vec の方が高かった．これは 1 つのクラスター内にノイズとなる異なるカテゴリサービスが所属していることを表す．purity 値はどのくらいのサービスが正しいクラスタに所属しているかを表し，適合率はそれぞれのクラスタにおける正しい要素の割合を表す．同一複合サービスを優先するサンプリング手法の purity 値は高く，適合率の平均は低かったことより，特定のカテゴリは正しくクラスタリングされていたと考えることができる．

続いて，BERT を用いたグラフ埋め込みにおけるクラスタリング精度について考察を述べる．ここでは，同一複合サービスを優先するサンプリングを行なった結果を skip-gram model で学習する場合と，BERT で学習する場合で比較し

た．表 6 表 7 表 8 表 9 の結果より，どの対応付けパターンにおいても，purity 値と適合率の平均は skip-gram model で学習させた方が高くなっていることがわかる．このことから，本研究においては BERT を用いたグラフ埋め込みは有効ではないことがわかる．skip-gram model より予測精度が高いとされる BERT が有効ではなかった理由は２つ考えられる． １つ目の理由はオンライン上で公開されている大規模なデータセットで学習された事前学習モデルを利用することができず，独自に用意した事前学習モデルではそれぞれのサービスの特徴を表すのが困難であったと考えられる．BERT にテキストデータとして入力されたサンプリング結果は各サービスの名前で構成されており，それぞれが独自の命名規則に基づいた名前であった．これにより，BERT の大規模事前学習モデルには出現しない単語が多く，特徴を得ることができなかった．そこで，BERT を利用するためにサンプリング結果を基に独自の事前学習モデルを作成した．しかし，サンプリング結果は実際のテキストデータと違い，テキストの構造に特徴がなく，後述する BERT 独自の学習方法が有効ではなかったと考えられる． ２つ目の理由は，BERT 独自の学習方法が有効ではなかったと考えられる．BERT は 4.3.1 で述べたように入力テキストを左右それぞれから学習する．しかし，左右それぞれから学習する方法は，ネットワークノードのサンプリング結果には有効ではない．理由は，左右どちらから入力されても成り立つからである．逆の順番から入力された場合は，それは別のノードを始点としたサンプリング結果になり，１つのサンプリング結果が２つの意味を持ってしまうことにより，学習結果に悪影響が現れたと考えられる．

最後に，高精度のサンプリング技術と埋め込み技術を用いたクラスタリング精度について考察を述べる．表 10 表 11 表 12 表 13 の結果より，全てのパターンにおいて，グラフ埋め込み手法の purity 値と適合率の平均は最も高いもしくは次点で高い数値を出している．特に，pattern1 や pattern2 の正解クラスタが重複しない対応付けにおいては BERT を用いたサービス説明文に基づくクラスタリング手法よりも高いクラスタリング精度を示している．これは類似機能サービスクラスターが唯一に決定する場合，サービス説明文に基づく手法よりも高いクラスタリング精度を出すことを表す．一方で，pattern3 と pattern4 において，purity 値と適合率の平均はサービス説明文に基づく手法が最も高精度なクラスタリング精度を出す．しかし，グラフ埋め込み手法は近傍に基づく手法より purity 値と適合率の平均は高くなっている．これは各ノードの周辺ノードデータ

の違いが関係すると考えられる．近傍に基づく手法では各ノードの隣接ノード情報しか考慮していないのに対し，グラフ埋め込み手法では隣接ノードよりもさらに遠いノードまでの経路をサンプリングしている．また，そのサンプリング結果を分散表現で出力するため，サンプリングの長さや個数が増えるほど各ノードの違いを分散表現から得ることができる．そのため，実験データの複合サービスの種類が増え，ネットワーク上で様々なノード間の連携関係が増えるほどグラフ埋め込み手法の精度は上がると考えられる．また，BERT を用いたサービス説明文に基づくクラスタリング手法は BERT の事前学習モデルを用いてサービス説明文の分散表現を獲得するため，サービス説明文の個数に影響されない．したがって，今後複合サービス事例が増え続けることが予想されるグラフ埋め込み手法は，将来的にはサービス説明文に基づく手法よりも高いクラスタリング精度になることが考えられる．

以上のことから，以下の結論が導ける．同一複合サービスを優先するサンプリング手法は，node2vec のようなランダムウォークベースのサンプリング方法より有効である．しかし，特定のカテゴリのサービスだけ正しくクラスタリングされていることが考えられ，汎用的なクラスタリング結果にするためには改善の余地が存在する．

第6章 おわりに

本研究では，従来の WSDL やサービス説明文を用いて自然言語処理モデルで学習するテキストベースのクラスタリング手法の代わりに，複合サービスのサービス依存関係に基づくクラスタリング手法を提案した．そして，同一複合サービスを優先してサンプリングするクラスタリング手法は，近傍に基づくクラスタリング手法や node2vec を用いたクラスタリング手法より有効であることを示した．また，skip-gram model を用いるグラフ埋め込みは BERT を用いたグラフ埋め込みよりクラスタリング精度は向上することを示した．

本研究の貢献は以下の２点である．

文脈を考慮したサンプリング戦略

ノード探索方法をランダムウォーク，幅優先探索にバイアスをかけた探索，深さ優先にバイアスをかけた探索，同一複合サービスに組み合わされる原子サービスを優先する探索の４つで評価した．その結果，正解クラスタを Primary カテゴリと Secondary カテゴリで作成し，正解クラスタが重複する対応付けを行なった場合，同一複合サービスに組み合わされる原子サービスを優先する探索のクラスタリング精度は他手法に比べて 5%上昇した．

ノード分散表現の学習

ネットワークのノードのサンプリング結果を各ノードが空白で区切られたテキストデータに加工し，従来の単語分散表現学習モデルで学習することで，各ノードの分散表現を獲得することができた．また，正解クラスタを Primary カテゴリと Secondary カテゴリで作成し，正解クラスタが重複する対応付けを行なった場合，skip-gram model を用いたグラフ埋め込みのクラスタリング精度は BERT を用いたグラフ埋め込みと比べて 15%上昇した．

今後，実世界において複合サービスの障害に対し，グラフ埋め込みを用いたクラスタリング手法を用いる場合，ユーザーが類似機能サービスを検索して組み替えるのではなく，システムが自動で組み替えてくれることが望ましいと考えられる．これを実現するためには，各クラスタが類似機能なクラスタであるだけでなく，入力データや出力データも近いものである必要がある．

本研究のクラスタリング手法は原子サービスの連携関係のみに着目したクラスタリング手法であるため，入力データや出力データ，また組み合わされる複合サービスの機能における利用の順番などを考慮することで，より精度の高いクラ

スタリング結果を得られると考えられる．また，本研究の連携関係に着目したクラスタリングはソースコード中のメソッドなどに応用することが可能であると考える．これにより，オンライン上に数多く存在するソースコードから類似機能を持つメソッドを探すことができ，再利用につながると考える．

本手法のクラスタリング結果は，サービス説明文を用いたテキストベースのクラスタリングに比べ，クラスタリング精度は高くなかった．しかし，従来の node2vec のようなランダムウォークベースのクラスタリング精度より高かったことから，DeepWSC[9]のようなテキストベースとグラフ埋め込みを混合させたクラスタリング手法のクラスタリング精度の向上に繋がったと考える．

謝辞

本研究を行うにあたり，熱心なご指導，ご助言を賜りました指導教官の村上陽平准教授に深謝申し上げます．また普段からお世話になっている社会知能研究室の皆さまにも感謝の意を表します．

参考文献

[1] Kemas Muslim Lhaksmana, Yohei Murakami, Toru Ishida, “Analysis of Large-Scale Service Network Tolerance to Cascading Failure, “IEEE Internet of Things Journal, Vol. 3, No. 6, pp. 1159-1170, 2016.

[2] Khalid Elgazzar, Ahmed E. Hassan and Patrick Martin: Clustering WSDL Documents to Bootstrap the Discovery of Web Service, Proceedings of IEEE International Conference on Web Services, pp. 147-154 (2010)

[3] Ling-li Xie, Fu-zan Chen, Ji-song Kou: Ontology-based semantic web service clustering, Proceedings of IEEE 18th International Conference on Industrial Engineering and Engineering Management, pp. 2075-2079 (2011)

[4] Min Shi, Jianxun Liu, Dong Zhou, Mingdong Tang, and Buqing Cao, “WE-LDA: a word embeddings augmented LDA model for web services clustering,” in *IEEE International Conference on Web Services (ICWS)*, 2017, pp. 9–16.

[5] J.Devlin,M.-W.Chang,K.Lee,andK.Toutanova,“BERT:Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805. [Online]. Available: http://arxiv.org/abs/1810.04805

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ICLR Workshop*, 2013.

[7] Tom Kenter, Alexey Borisov, and Maarten de Rijke. Siamese CBOW: Optimizing word embeddings for sentence representations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL’16).

[8] Aditya Grover, Jure Leskovec: node2vec: Scalable Feature Learning for Networks, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855-864 (2016)

[9] Guobing Zou, Zhen Qin, Qiang He, Pengwei Wang, Bofeng Zhang, Yanglan Gan: “DeepWSC: Clustering Web Services via Integrating Service Composability into Deep Semantic Features”, IEEE Transactions on Services Computing, 2020.