Vol.2011-BIO-25 No.9
2011/6/23

生物情報解析ワークフローのための REST サービスの SOAP サービス変換手法

池田成吾[†1] 木戸善之[†2] 瀬尾茂人[†1]
竹中要一[†1] 松田秀雄[†1]

多数のツールやサービスを組合せる必要がある生物情報解析を円滑に実行するためにワークフローが利用されている．しかし REST サービスは仕様を記述するための標準的な形式が確立されていないため, ワークフローツールでの利用が困難である．そこで本研究では, 生物情報解析を行うワークフローツールで REST サービスを利用するための, SOAP サービスへの変換手法の提案を行う．提案手法では, REST サービスのドキュメントを URL 記載形式により分類し, 各分類ごとに機械学習で REST サービスの URL を抽出する．抽出した URL をもとに REST サービスを中継する SOAP サービスを自動生成する．提案手法を BioCatalogue に登録されている REST サービスに適用し, その有効性の検証を行った．

A method for using REST service in bioinformatics workflow by converting to SOAP service

Seigo Ikeda ,[†1] Yoshiyuki Kido ,[†2] Shigeto Seno ,[†1]
Yoichi Takenaka [†1] and Hideo Matsuda[†1]

In bioinformatics, workflows have used frequently to combine many tools or services. But it have been difficult to use REST services by workflow tools, because there is no way to read their specification computationally. In this research, we proposed a method for using REST services in workflows by converting REST services to SOAP services. This method classfies documents of REST services by their forms writing URL. And by using machine learning, this method extracts URLs of only REST services from classified documents. By using extracted URLs, this method generates SOAP services that access REST services. To show effectiveness of this method, we use it with example REST services registered in BioCatalogue.

1. はじめに

DNA 配列などの生物情報は, 様々な機関が実験によって取得し, それぞれの機関がデータベースとして Web 上から一般に公開している．近年, 実験技術の発展によって生物情報が急激に増加するようになり, 生物情報データベースの更新が頻繁に行われるようになった．そのため, 生物情報データベース全体を手元の計算機環境にダウンロードして利用してきた生物情報解析の研究者は, データが追加されるたびに最新の大容量なデータベースを頻繁にダウンロードしなければならなくなった．そこで, データベースを公開している機関は, Web サービスによってデータベースへの検索や配列の相同性検索などの解析に必要な処理を行えるようにすることで, データベースを入手せずに解析を行えるようにした．現在では, 多くの解析処理が Web サービスとして提供されており, 生物情報解析 Web サービスのレポジトリである BioCatalogue[1)] では, 2009 年だけで 100 個以上の Web サービスが新たに登録され, 2010 年までで 1400 個以上が登録されている[2)]．

生物情報解析では, ある処理の結果を別の処理の入力とするといった処理を用いて行う解析が多い．例えば, 相同性検索によって類似度を求めた DNA 配列群を, 系統樹解析処理の入力として進化系統樹を作成するといった解析などである．これらの処理の多くは Web サービスとして利用可能であり, ある Web サービスの結果を別の Web サービスの入力として用いることを, Web サービスの連携と呼ぶ．この Web サービスの連携を容易に実現できるようにするためにワークフローツールが利用される．ワークフローツールは, ワークフローで表現される処理の流れを容易に構成, 実行できるようにするツールであり, Web サービスをワークフロー内の処理として利用することができる．そして, 取り込んだ Web サービスの連携をワークフローとして構成することで, Web サービスとして処理の連携をグラフィカルユーザインタフェース (GUI) 上で容易に実現することができる．生物情報解析のためのワークフローツールとして, Taverna[3)] や Kepler[4)] など多くのツールが利用されている．

Web サービスは, SOAP (Simple Object Access Protocol) サービス[5)] と REST (REpresentational State Transfer) サービス[6)] に大別され, SOAP サービスでは仕様を記述するための言語 WSDL (Web Service Description Language)[7)] が存在する．WSDL で記述された仕様から SOAP サービスのインタフェースのコードを生成することが可能であり,

†1 大阪大学大学院情報科学研究科
†2 大阪大学臨床医工学融合研究教育センター

SOAP サービスを提供するプロバイダはサービスと一緒に公開している．一方, REST サービスは近年普及してきた簡易的な Web サービスであり, SOAP サービスにおける WSDL の様な仕様記述のための標準的な言語は策定されておらず, サービスプロバイダはそれぞれが HTML などを用いた自由形式のドキュメントで仕様情報を公開している．こうした規格の差異から SOAP サービスと REST サービスの両者を連携させるためのワークフローツールはまだ少なく, 生物情報解析ワークフローでの利用は困難であると言える．

本研究では, 生物情報解析ワークフローで REST サービスを容易に利用するために, REST サービスを SOAP サービスに変換する手法を提案する．変換元の REST サービスを利用し, REST サービスの結果をそのまま出力する SOAP サービスを生成することで変換を行った．入力として REST サービスのドキュメント (以下 REST ドキュメント呼ぶ) を受け取り, ドキュメントを解析することで仕様情報を抽出した．ドキュメントの解析は, その特徴からドキュメントをいくつかの種類に分類し, それぞれの特徴に対応する解析を行うことで精度を上げることを試みた．そして仕様情報から REST サービスを利用する SOAP サービスを生成し, 変換を行った．

以下 2 章では REST サービスをワークフローツールで利用する場合の問題点について述べる．3 章では, 提案手法である REST サービスから SOAP サービスの変換手法について述べる．そして, 4 章では提案手法の評価, 5 章で結論を述べる．

2. 生物情報解析ワークフローでの REST サービスの問題点

本章では生物情報解析ワークフローに関する技術的な詳細, サービスとワークフローについて述べる．その後, ワークフローツールで Web サービスを利用するための問題点について述べる．Web サービスは，ネットワーク上の別の計算機のプログラムを実行し, その結果を取得する技術である．Web サービスの実行は以下のような手順で行われる．まず, サービスを利用する側の計算機 (以降ではクライアントと呼ぶ) がサービスを公開している計算機 (以降ではサーバと呼ぶ) に対してサービスの実行要求を送信する．そして, サーバは実行要求を受け取り, サービスとして提供しているプログラムを実行し, 実行結果をクライアントに返信する．最後に, クライアントは返信された実行結果を受信する．Web サービスではプログラムがサーバ上で実行されるため, プログラムの実行に必要なデータはサーバ上にのみ配置していれば良く, クライアントにデータをダウンロードする必要がない．そのため, 検索などのデータベースへの解析処理を Web サービスとして提供することで, クライアントにはデータベースをダウンロードすることなく, サーバ上の解析処理を実行することができる．現在では, データベースを利用しない解析処理も Web サービスとして多く提供されている．

次にワークフローツールでの Web サービスの利用について述べる．ワークフローツールでは, ユーザが連携するスクリプトを記述することなく, GUI で有向グラフを記述する形でワークフローを実現するのが一般的である．ワークフローを構成するためには, ワークフロー上の処理の受け取ることができる入力パラメータの数やその型情報などの, 処理の仕様情報が明確にわかっている必要がある．そのため, ワークフロー上の処理として Web サービスを利用するためには, Web サービスが持っている仕様情報が必要になる．それらがコンピュータリーダブルな形式であれば, ワークフローツールでその仕様情報ファイルを読み込むことで, 自動的に Web サービスを利用することが可能になるが, 仕様情報が決められたフォーマット以外, 自然言語による自由形式な仕様情報しかない場合はユーザが手動でワークフローツールに認識させる必要があり, ワークフローツールを使う上で Web サービスの型情報などプログラミングの知識が必要となってくる．

SOAP サービスでは, WSDL ファイルから仕様情報を読み込むことができる．WSDL ファイルは, SOAP サービスの仕様情報を記述するために標準的に利用されており, ほとんどの SOAP サービスのものが公開されている．WSDL は, W3C によって標準化されている XML 形式の仕様記述言語である．WSDL では, SOAP サービスを利用するための URL や入力パラメータの型情報などの仕様情報の記述方法が規定されており, WSDL ファイルはその規定に沿って記述される．したがって, WSDL ファイルを規定で定められている通りに読み込むことにより, 容易に仕様情報を取得することができる．そのため, WSDL ファイルにより仕様情報を読み込むことで, SOAP サービスはワークフローツールで容易に利用することができる．

REST サービスにも WSDL と同じように, 仕様情報の記述方法を規定した WADL (Web Application Description Language)[8] が W3C により標準化されているが, あまり普及していない．実際に 2010 年 2 月時点で, BioCatalogue に登録されている REST サービスでは, WADL を公開しているサービスは 1 つもなかった．そのため現状では, 何らかの形式の仕様情報の説明文書を利用するしかないということになっている．そこで REST サービスでは, 主に自然言語で記述した HTML 形式のドキュメントで仕様情報を記述している．この REST ドキュメントには共通のフォーマットは存在せず, 記述形式がサービスごとに異なるため, どこにどの情報が記述されているかがわからず, 仕様情報を読み込むことが困難である．そのため, ワークフローツールでは SOAP サービスは容易に利用できるが REST サービスの利用は困難であるということが問題になっている．

3. REST サービスから SOAP サービスへの変換手法

本章では, 生物情報解析ワークフローで REST サービスを利用するための, REST サービスの SOAP サービスへの変換手法について述べる.

3.1 変換処理の概要

変換処理は入力として REST ドキュメントを受け取り, 変換結果の SOAP サービスの WSDL ファイルを出力する. 出力される WSDL ファイルをワークフローツールで読み込むことにより, 変換結果の SOAP サービスをワークフローに取り込むことができる. すなわち, 変換元の REST サービスの処理をワークフロー上で容易に利用することができる.

変換結果として生成される SOAP サービスは, 変換元の REST サービスの実行結果を取得するサービスである. 変換結果の SOAP サービスは, 変換元の REST サービスの URL にアクセスすることで REST サービスの実行結果を取得する. そして実行結果をそのまま出力することで, 変換元の REST サービスと同じ結果を出力する.

この SOAP サービスを生成するためには, REST サービスの入力パラメータなどの仕様情報が必要になる. そこで変換に必要な情報を取得するために, REST サービスの仕様情報を記述したドキュメントを入力として受け取る. 受け取ったドキュメントの内容を解析することで, 入力パラメータなどの仕様情報を抽出する. そして, 抽出した仕様情報から REST サービスを利用する SOAP サービスと, その SOAP サービスの WSDL ファイルを生成し, 公開する.

3.2 REST ドキュメントの解析

本節では変換の精度を決定する, REST ドキュメントの解析処理について述べていく.

3.2.1 解析処理の概要

本手法では REST サービスの仕様情報を記述したドキュメントを解析することで, サービスの変換に必要な入力パラメータなどの情報を抽出する. REST ドキュメントは, 図 1 のような Web で公開されている HTML 形式のドキュメントである. 内容は自然言語で記述されており決まったフォーマットを持っておらず, 使用例の URL のみ記述しているものや入力パラメータの内容を 1 つずつ説明しているものなど記述方式はドキュメントごとにそれぞれである. そのため仕様情報の取得のために, 自然言語解析や HTML タグ解析を行うことで必要な情報のみを抽出する必要がある.

REST ドキュメント中には, 同じホストから提供される複数の REST サービスの仕様が記述されていることが多い. そこで解析の出力は, REST ドキュメント中に記述されている

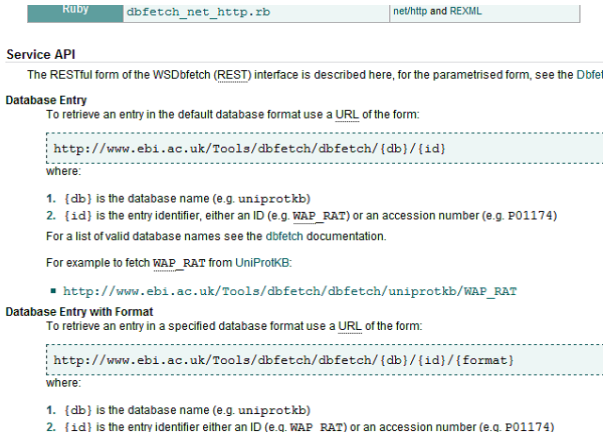

図 1 REST ドキュメントの例
Fig. 1 Example of REST document

全ての REST サービスの仕様情報とする. 変換処理では, SOAP サービスがメソッドと呼ばれる複数の処理を提供できることから, このメソッドに 1 つの REST サービスを対応させることで, 1 つの REST ドキュメントから 1 つの SOAP サービスへの変換を行う.

解析処理は, まず REST ドキュメントの特徴に対応した解析を行い, 結果が得られなかった場合にはドキュメント中のリンク先のページの解析を行うという手順で行った. この解析の詳細については 3.2.3 節で述べることにする.

3.2.2 ドキュメントの分類による解析処理

この解析処理では精度向上のため, REST ドキュメントを特徴ごとに分類し, その特徴に合わせた解析を行った. REST ドキュメントは決まったフォーマットを持たず, 様々な形式で記述されている. その中でも, いくつかのドキュメントには解析に利用することができるような共通した特徴が見られた. そこで, それらの特徴ごとに REST ドキュメントを分類し, それぞれの特徴に対応する解析を行うことで解析の精度を上げることを試みた. 具体的な分類とその解析方法については 3.3 節で述べることにする.

REST ドキュメントの分類の判別は, 全ての解析を試行し, 多くの結果が得られたものをその分類とした. 分類ごとの解析は, その分類の特徴に対応した解析しか行えない. そのため, 他の分類の REST ドキュメントに対しては的外れな解析を行うことになり, 解析結果を得ることができない. そこで入力された REST ドキュメントに対して, それぞれの分類に対

する解析を全て実行し, それぞれの解析で取得できた結果を比較し, 取得結果が最も多かった解析の分類をその REST ドキュメントに対する正しい分類として, その解析結果を出力する. ここでの解析結果の数とは, REST ドキュメントから抽出された REST サービスの数である.

3.2.3 サブドキュメントの解析

分類による解析で結果を取得できない場合には, サブドキュメントを解析し結果の取得を試みる. REST ドキュメントの中には, そのドキュメント中では REST サービスの仕様情報を記述せず, 仕様情報を記述しているドキュメントへのリンクを記載しているものがある. 以降では, このリンクのみを記載したドキュメントのことをメインドキュメントと呼び, リンク先の仕様情報を記述しているドキュメントのことをサブドキュメントと呼ぶ. このメインドキュメントに分類による解析を行っても, メインドキュメント中に仕様情報は記述されていないため結果を得ることができない. そこで, 分類による解析で閾値以上の結果を取得できない場合には, そのドキュメントがメインドキュメントであるとしてドキュメント中のリンクをたどり, リンク先のページをサブドキュメントとして解析し, その結果を出力するようにした. なお, この解析でリンク先をたどるのは 1 度までとした. サブドキュメントで分類による解析の結果が得られなかった場合に, この解析を無限に繰り返してしまうことを防ぐためである.

次にサブドキュメントへのリンクの判別方法について述べる. メインドキュメント中には, サブドキュメントへのリンク以外にも外部サイトへのリンクなどが多く含まれている. サブドキュメントの解析を行うためには, サブドキュメントへのリンクのみを判別して解析を行わなければならない. サブドキュメントへのリンクは, 「サービスの詳細情報へのリンク」としてドキュメント中で 1 か所にまとめて記述されていることが多い. そこで, REST ドキュメント中で複数が近距離で記述されているリンクをサブドキュメントへのリンクと判別するようにした. ここでの距離とは, リンク間に記述されている有効なコンテントの数である. 有効なコンテントとは, 同じタグによって修飾される 1 文字以上のテキストをさす. 有効なコンテントの定義は, 表 1 のようになる.

この距離が閾値以下となるリンク同士を, 同じクラスタとしてクラスタリングした. さらにサブドキュメントへの URL は, 同じドメイン部分を持つことが多いことから, クラスタ中のリンクのドメイン部分を調べ, 異なるものを別のクラスタとして分割するなどのフィルタリングを行った. その結果, 要素数が閾値以上になったクラスタでは複数のリンクが近距離で記述されていたものとして, そのクラスタ中のリンクをサブドキュメントへのリンクと判別するようにした.

表 1 有効なコンテントの定義
Table 1 Definition of valid content

HTML の記述	有効なコンテントの数 (距離)
$\langle a\rangle content\langle/a\rangle$	1
$\langle b\rangle\langle a\rangle content\langle/a\rangle\langle/b\rangle$	1
$\langle b\rangle content1\langle a\rangle content2\langle/a\rangle\langle/b\rangle$	2
$content1\langle br\rangle content2$	2

http://flybase.org/chadoxml/id/<FlyBase ID>

図 2 テンプレートとなる URL
Fig. 2 URL shown as template

そして, サブドキュメントへのリンクと判別されたリンクの示すページの解析を行い, それらの解析結果を合わせたものを REST ドキュメントの解析結果として出力する.

3.3 REST ドキュメントの分類とその解析方法

REST ドキュメントの特徴による分類は, テンプレート型, 通常型, フォーム型の 3 つとした. 本章では, それぞれの分類とその解析方法について述べていく.

3.3.1 テンプレート型

テンプレート型のドキュメントは, サービスを利用する URL を模式的に表したテンプレートとなる URL を持つドキュメントの分類である. REST サービスでは, 入力パラメータを REST サービスの URL 中の特定の位置に埋め込むことで入力を行うことができる. テンプレートとなる URL とは, URL 中の入力パラメータの位置をかっこで囲むなどして強調している図 2 のような URL である. このテンプレートなる URL によって仕様情報を記述している REST ドキュメントをテンプレート型と分類した.

テンプレート型のドキュメントの解析は, テンプレートとなる URL を抽出することで行った. 以下では, テンプレートとなる URL の抽出方法について述べる. REST ドキュメント解析では, サービスの入力パラメータの内容を取得する必要ある. テンプレートとなる URL は入力パラメータを強調しているため, 解析によって入力パラメータを取得することが容易である. また, テンプレートとなる URL によって解析を行うことで, 抽出した URL が重複したサービスのものであるかの判別を行わなくてもよくなる. REST ドキュメントによっては同じサービスの URL を, 異なる入力パラメータの具体例を解説するために, 入力内容を変

http://www.ebi.ac.uk/Tools/dbfetch/dbfetch?db=EMBL&id=J00231

図 3 入力の具体例を記述した URL
Fig. 3 URL to show examples of parameters

更した形で複数個の URL を記述している場合がある．この場合には，抽出した REST サービスの URL が異なるサービスのものであるかの判別が必要になる．しかし，テンプレートとなる URL では入力パラメータ部分を抽象的に表現するため，入力の具体例によって重複して記述されることがない．そのためテンプレートとなる URL を抽出することができれば，抽出した URL の重複を気にせずに解析を行うことができる．

テンプレートとなる URL の抽出には，決定木を用いて URL 周辺の記述形式を学習させることで行った．REST ドキュメント中には，テンプレートとなる URL の他にも多くの URL が記述さている．テンプレートとなる URL のみを抽出するためには，REST ドキュメントに記述されている URL がテンプレートとなる URL かそれ以外の URL かの判別を行う必要がある．テンプレートとなる URL は特殊な URL であるため，ドキュメント中で強調されて記述される場合が多い．URL を強調するための URL 周辺の記述形式はドキュメントごとに異なっていたが，その中にもある程度共通した形式が見られた．例えば，「Template URL」といった表題が付けられている，入力パラメータの解説が記述されているといった記述形式である．そこで，より「それらしい」記述形式が近距離に記述されている場合に高いスコアを付けることで，実際のテンプレートとなる URL 周辺の記述形式を決定木によって学習させ，その決定木を用いてテンプレートとなる URL の判別を行った．

3.3.2 通　常　型

通常型のドキュメントは，テンプレートとなる URL を持たず入力の具体例を記入した URL を示すことでサービスの利用法を解説しているドキュメントの分類である．入力の具体例を記述した URL は，テンプレートとなる URL とは異なり URL 中の入力パラメータを抽象的に記述せず，実際の入力の値を記入した図 3 のような URL である．通常型のドキュメントでは，REST サービスの URL は入力の具体例を記述した URL のみであり，テンプレートとなる URL は記述されていない．そのため，テンプレート型のドキュメントとは別の解析方法を行う必要がある．

通常型のドキュメントの解析は，REST サービスの URL を抽出することで行った．REST サービスの URL は解析の結果として必要な情報 1 つである．REST サービスの URL は REST サービスの利用法を示すものであり，その記述形式を解析することで入力パラメータなどの他の情報も取得できる．同じ REST サービスを複数の URL が重複して示していることの検出は，入力パラメータの位置を特定することで行うことができる．同じ REST サービスを示している URL 同士は，入力パラメータ部分に記入されている入力の具体例のみが異なっている．よって入力パラメータ以外の部分が一致していれば，同じ REST サービスの URL と見なすことができる．そのため REST サービスの URL のみを抽出することで，サービスの仕様情報を取得することができる．

REST サービスの URL の抽出は，ドキュメント中の URL の中で最も多いドメイン部分を持っていた URL を REST サービスとする方法で行った．REST ドキュメント中には，外部ページへのリンクなどの REST サービスの URL 以外の URL も多く記述されている．そのため，REST サービスの URL のみを抽出するためには，REST サービスの URL とそれ以外の URL を判別する必要がある．同じ REST ドキュメント中に記述されている REST サービスの URL は，同じドメイン部分を持つことが多い．また，REST サービスの URL 以外の外部ページへのリンクなどは，それぞれ異なるドメイン部分を持つ場合が多くなる．そのため REST ドキュメント中に記述されている URL のドメイン部分で，最も多いものが REST サービスの URL のドメイン部分であると予想できる．そこで，REST ドキュメント中の全ての URL のドメイン部分を調べ，最も多いドメイン部分を持つ URL を REST サービスの URL として抽出した．

サービスの入力パラメータの情報は，URL 中の非自然言語を入力パラメータ部分とする方法で行った．生物情報解析を行うサービスへの入力は，”GAATTC”といった塩基配列などの非自然言語である場合が多い．また，REST サービスの URL の入力パラメータ以外の部分は，get, search などの自然言語で構成される場合が多い．そこで，URL 中の語が自然言語であるかを調べ非自然言語であれば実際の入力を記入しているとして，その語の部分を入力パラメータであるとした．

3.3.3 フォーム型

フォーム型のドキュメントは，サービスを利用するための HTML フォームを持つドキュメントの分類である．HTML フォームは，HTML の form タグの機能によりテキストボックスなどのフォームにより入力を受け付け，入力内容を指定された URL に送信し，結果を表示するものである．この HTML フォームにより REST サービスを実行できるようにしているドキュメントをフォーム型と分類した．

フォーム型のドキュメントの解析は，HTML フォームの内容を解析することで行った．HTML フォームを構成するタグには，入力内容を送信する URL やそれぞれの入力に指定さ

れた入力名などが記入されている. それらを解析により取得することで, REST サービスの URL, 入力パラメータなどの情報を取得した.

4. 評価方法

本章では, 提案手法である REST サービスから SOAP サービスへの変換の精度を評価する方法について述べていく.

4.1 評価内容

本手法の評価は, 生物情報解析 Web サービスのレポジトリである BioCatalogue に登録されている REST ドキュメント 76 個の中で, 重複して登録されているものやドキュメントが存在しないものを除いた 39 個 (2011 年 2 月時点) のドキュメントをいくつ完全に変換できたかと, その変換成功率を調べることで行った. 正しい変換結果として, REST ドキュメントを実際に目で見て, 記述されている REST サービスとその入力パラメータを判別したものを使用した. 完全に変換できた REST ドキュメントとは, REST ドキュメントに記述されている全ての REST サービスとその入力パラメータを正しく取得し変換できたドキュメントをさす. 変換成功率は, 変換の精度を適合率, 再現率, F 値で算出したものとする. 適合率は取得した正解数を取得した全ての情報数で割った数値であり, 再現率は取得した正解数を全ての取得すべき正解数で割った数値である. F 値は, 適合率, 再現率の調和平均をとったものである.

これらの結果を REST ドキュメントの分類ごとに調べることで評価を行った. サブドキュメントでは REST ドキュメントが複数分割されており, またサブドキュメントごとに 1 つの REST サービスしか記載されていないため, サブドキュメントごとに適合率, 再現率, F 値を別に評価した.

4.2 評価結果

表 2 は BioCatalogue 中の REST サービスの分類別の数と, 分類ごとの完全に変換できたドキュメント数を表したものである. なお, フォーム型のドキュメントはサブドキュメントとしてのみ登場しているため, 0 個となっている.

表 3 は分類ごとの REST サービスの変換成功率である. 表 4 はサブドキュメントを持つ REST ドキュメントの数とそのサブドキュメントの数, そしてそれらの分類ごとの変換成功率である. サブドキュメントは, フォーム型, 通常型のドキュメントのみがあり, それらについてのみ記述している.

表 2 を見ると, 完全に変換できたドキュメントは 39 個中 14 個と 3 割程度だった. テンプレート型, サブドキュメントによるドキュメントは 5 割程のドキュメントの変換に成功していた. 変換に成功したドキュメントの割合の低下は, 分類による解析で対応できなかった未分類のドキュメントと, 通常型のドキュメントの変換の成功数が少なかったためである. 未分類のドキュメントへの対応は今後の課題となるだろう. 通常型のドキュメントは, 1 つの REST サービスに対応する URL を別のサービスのものとして, 重複して検出してしまうことがある. そのため, 他の分類より過剰な取得が多くなるので, 完全な変換の成功率が低くなったのではないかと考えられる.

表 3 を見ると, 全体を通して変換の精度は 8 割程度だった. テンプレート型の変換は, 適合率と再現率の両方が 9 割以上と比較的高精度だった. 通常型の変換は, 再現率は 9 割以上だったが適合率が 6 割程度とかなり低い値になっていた. 適合率が低く再現率が高いことから, 情報を過剰に取得してしまっていることがわかる. そのため, 1 つの REST サービスを

表 2 変換に成功したドキュメントの分類ごとの数
Table 2 Numbers of Documents converted perfectly

ドキュメントの型	分類ごとの数	完全に変換できた数
テンプレート型	10	4
通常型	17	5
フォーム型	0	0
未分類	3	0
サブドキュメントによるドキュメント	9	5
合計	39	14

表 3 分類ごとの変換成功率
Table 3 Success rate of conversion

ドキュメントの型	適合率	再現率	F 値
テンプレート型	0.9608	0.9333	0.9469
通常型	0.5833	0.9286	0.7165
未分類	N/A	0.0000	N/A
全体	0.7994	0.8914	0.8429

表 4 サブドキュメントでの変換成功率
Table 4 Success rate of conversion in Subdocuments

ドキュメントの型	適合率	再現率	F 値
通常型	1.0000	0.9574	0.9783
フォーム型	1.0000	0.8333	0.9091

表す URL を別のサービスのものとして, REST サービスの情報を重複して過剰に取得していることが原因であると考えられる.

表 4 を見ると, サブドキュメントでの変換は全体の変換より精度が高くなった. 特に通常型の変換は大きく精度が上がっている. これはサブドキュメント中に 1 つの REST サービスの仕様情報しか記述されておらず, 解析の難度が下がっているためと思われる.

5. おわりに

本研究では, REST サービスを SOAP サービスに変換することで, 生物情報ワークフローで利用できるようにする手法を提案した. 変換方法として, REST ドキュメントを解析することで REST サービスの仕様情報を取得し, REST サービスを利用する SOAP サービスを生成した. REST ドキュメントの解析は, ドキュメントを特徴ごとに 3 種類に分類し, それぞれの特徴に対応させた解析を行った. 評価結果から, 提案手法で REST サービスから SOAP サービスへの変換が 8 割程度行えることがわかった. これにより, ワークフローツールでの REST サービスの利用が容易になるだろう. 今後の課題として, 特に精度が悪かった通常型ドキュメントにおけるサービスの仕様情報の過剰取得の改善と, 未分類のドキュメントへの対応が考えられる.

6. 謝　　辞

本研究の一部は科研費 (22310125,22680023) の助成を受けたものである.

参 考 文 献

1) ”BioCatalogue The Life Science Web Service Registry,” http://www.biocatalogue.org/ (last access:2011/2/8)
2) J. Bhagat, F. Tanoh, E. Nzuobontane, T. Laurent, J. Orlowski, M. Roos, K. Wolstencroft, S. Aleksejevs, R. Stevens, S. Pettifer, R. Lopez, and C. a Goble, “ BioCatalogue: a universal catalogue of web services for the life sciences., ” Nucleic acids research, vol. 38 Suppl, Jul. 2010, pp. W689-94.
3) T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, ”Taverna: a tool for the composition and enactment of bioinformatics workflows,” Bioinformatics, Vol. 20, No. 17, pp.3045 3054, 2004.
4) I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, “ Kepler: an extensible system for design and execution of scientific workflows, ”Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004., pp. 423-424.
5) ”Simple Obeject Access Protocol (SOAP) 1.2.,” http://www.w3.org/TR/soap/ (last access:2011/2/11)
6) Fielding RT, ”Architectural Styles and the Design of Network-based Software Architectures.,” Ph.D. Thesis, UC Irvine. 2000
7) Roberto C, Jean JM, Arthur R, Sanjiva W, ”Web Services Description Language (WSDL) Version 2.0.,” http://www.w3.org/TR/wsdl20/ (last access:2011/2/11)
8) Marc H, ”Web Application Description Language (WADL),” http://www.w3.org/Submission/wadl/ (last access:2011/2/11