

機械学習サービスを登録・提供するサービス基盤の構築に向けて

Toward a Construction of Service Platform for Machine Learning Technologies

中口 孝雄 (京都情報大学院大学)

Takao Nakaguchi (The Kyoto College of Graduate Studies for Informatics)

Abstract

サービスコンピューティングによるプログラミングインタフェース定義の明確化・標準化と機能のコンポーネント化は、再利用性を向上させ、サービス合成による新しいサービスの創出を可能にした。一方近年急速に発達している機械学習を応用した技術では、様々な実装がオープンソースソフトウェアやクラウドサービスを通じて公開されているものの、プログラミングインタフェースやデータフォーマットは実装毎に異なり、コンポーネント化やその組み合わせにより新しい複合サービスを作成することは難しい。本稿では、サービスコンピューティングを、機械学習を応用した技術に適用し、再利用性の向上やサービス合成の実現を行うために必要な手法について検討し、機械学習サービスのインタフェースとサービスプロトコル拡張を提案する。さらに、提案したインタフェースを用いて複合サービスが作成可能であることを実証する。

Services computing encourages clarification and standardization of programming interface, and componentize of functions to achieve reusability and create new services by composition of existing services. Though many open-source implementations and cloud services of machine learning technologies, which has been rapidly developing in recent years, are provided, their programming interfaces or data formats are difference even for same type of technologies, it is hard to componentize or creations of new services by services composition. In this paper, we examine the methods necessary to apply service computing to machine learning technologies to achieve reusability and realize services composition of machine learning services. In addition, we propose service interfaces for machine learning services, extensions for service invocation protocol and demonstrate that we can create new services by combining services that have proposed service interface.

1. はじめに

サービスコンピューティングは、コンピュータにより提供される機能を部品化して利用するために必要な技術の総称である [1]。情報システムを構成するためには様々な機能を開発する必要があるが、サービスコンピューティングは呼び出しインタフェース定義の明確化と、ネットワークを通じて利用するためのプロトコルを定義することにより、複数台のマシンに分散した機能を必要に応じて利用し一つのシステムを構成することを可能にした。このようなアーキテクチャは、サービス指向アーキテクチャ (Service Oriented Architecture, SOA) [2][3] と呼ばれている。サービス指向アーキテクチャでは、ネットワークを通じて利用可能な機能をサービスと

呼ぶ。サービスはそれを利用するプログラムとは独立して動作しており、その呼び出しインタフェースは機械読み取り可能な形式で定義され、公開される。ネットワーク経由でインタフェース定義に基づいたリクエストを送信すると、処理が行われ、同じくインタフェース定義に基づいた形式で結果を受信できる。インタフェースの記述仕様としては、WSDL (Web Services Description Language) [4] が、ネットワークプロトコルの仕様としては SOAP [5] が定義されている。WSDL はプログラミングインタフェース定義言語の一種であり、XML (eXtensible Markup Language) [6] を用いてサービス名やメソッド名、引数の情報を記述する。サービスが提供されているネットワークアドレスであるエンドポイントも WSDL により記述する。エンドポイントと

しては、URL が使用されるのが一般的である。

サービスは独立して稼働しており、定義され公開されたインタフェースに基づきネットワーク経由で利用できる。アプリケーションプログラムから利用するだけでなく、複数のサービスを組み合わせる新しいサービス(以下、複合サービス)を作成する、サービス合成 [7] も行われる。サービス合成の記述仕様の一つに、WS-BPEL (Web Services Business Process Execution Language Version 2.0) [8] がある(以下 BPEL)。BPEL は XML を採用したプログラム記述言語であり、リクエストの作成やサービスの呼び出し、受け取ったデータの加工、条件分岐やループ、平行実行など、一般的なプログラム記述言語が備える機能を持っている。また、BPEL で記述された手順自体もサービスとして利用可能となる。複合サービス自体もサービスであり、それを含んだ複合サービスも作成可能である。そのため、階層的な複合サービスが構成される。このように、サービスコンピューティングでは、機能を独立したサービスとして分割し、それを組み合わせた新しいサービスを構築可能とする。

サービスコンピューティングの応用例としては、航空券の手配とホテルの予約のような商用サービスの他にも、自然言語処理などの研究成果をサービス化し合成して利用するものも提供されている。言語グリッド [9] はその例の一つであり、自然言語処理分野での研究成果や非営利組織が提供する言語コーパス、商用の言語処理機能などがサービスとして提供され、それらを組み合わせた複合サービスも作成できる。

近年、機械学習技術やとりわけ深層学習を用いた技術が発展し、オープンソースソフトウェアや学習データ、クラウドサービスが提供されている。これらの技術の発展は非常に速く、絶えず新しいアプローチが提案され、ソフトウェアやデータも更新される。一方で従来のサービスコンピューティングで用いられたような、インタフェースの定義や標準化は十分ではなく、ソフトウェア毎に独自のインタフェースを持っているため、それを利用するアプリケーションはソフトウェア毎に呼び出し処理を記述する必要があり、同種のサービスであっても容易に切り替えて利用することが難しい。本稿では、機械学習を応用した機能においてもサービスコンピューティングのアプローチを提供し、再利用性を向上させサービス合成を可能にするための手法について検

討する。本稿は以下のように構成する。2章で言語グリッドにおけるサービスコンピューティングの適用について述べ、3章で機械学習サービスのインタフェース定義、4章でプロトコルについて提案する。5章で実際に機械学習サービスとそれを組み合わせたサービスを作成することで、提案を検証し、6章で結論を述べる。

2. 言語グリッドとサービスコンピューティング

言語グリッド [9] はサービスコンピューティングの自然言語処理分野への応用である。インターネット上では対訳辞書や対話コーパスなどのデータ、辞書や形態素解析、機械翻訳などのプログラム、およびネットワーク経由でそれらを利用できるようにしたクラウドサービスなど、研究用や商用含め様々な自然言語処理に関する資源が公開されているが、それらは独自のデータフォーマットやプログラミングインタフェースを持っており、アプリケーションプログラムから利用するには個別にプログラムを記述する必要がある。言語グリッドでは、自然言語処理分野にサービスコンピューティングを導入するため、辞書や形態素解析などの種別毎にサービスインタフェースを定義し、それに基づいて言語資源をサービス化した(以下、原子サービス)。言語グリッドの構成を図 1 に示す。サービスは、標準インタフェースと言語資源のインタフェースの変換を行い、また提供形態(データのみ、ネイティブのプログラム、クラウドサービス)の違いを隠蔽する。サービスは言語グリッドのコアノードに登録され、コアノードを通じて呼び出される。これにより、サービスを利用するアプリケーション(多言語アプリケーション)は、言語資源のインストール方法や起動方法、提供形態やインタフェースの違いを意識せず、言語サービスを利用できる。また、コアノードでは、言語サービスが呼び出される際に、言語サービス提供者が設定した情報を基に、サービスの呼び出しが許可されているか、回数制限を超過していないかを検証する。これにより、言語サービスの提供者は、自身が提供する言語サービスのアクセス制御を行える。

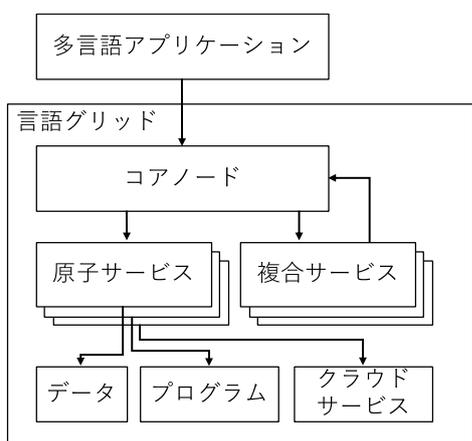


図1 言語グリッドの構成

表1 言語グリッドのサービスインタフェース例

種別	インタフェース
形態素解析サービス	Morpheme[] analyze(String language, String text)
	Morpheme {String word, String lemma, String partOfSpeech}
辞書検索サービス	Translation[] search (String sourceLang, String targetLang, String headWord, String matchingMethod)
	Translation {String headWord, String[] targetWords}
機械翻訳サービス	String translate (String sourceLang, String targetLang, String source)

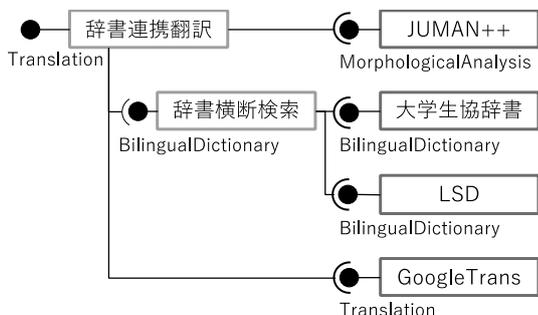


図2 複合サービスの構成例

サービスはその種別毎に、プログラミングインタフェースが定義されている。インタフェースは戻り値の型、メソッド名、引数で構成される。表1に、その一部を示す。自然言語処理分野では利用の容易さやデータ作成や実装の都合上様々な形式のデータやプログラミングインタフェースが利用されており、一つに統一することは難しい。そのため言語グリッドでは、同じ種類の言語資源に多くみられる特徴と、ネットワーク経由で利用するために必要十分

なパラメータおよび戻り値を備えるという観点からインタフェースが定義されている。例えば形態素解析では、入力に言語とテキストを受け取り、テキストを形態素に分割した結果を返す。形態素は、単語 (word)、原型 (lemma)、品詞 (partOfSpeech) から構成される。品詞は言語により異なるが、名詞や動詞など、多くの言語に見られる共通の品詞が定義されている。例えば同じ日本語向けの形態素解析でも品詞の定義は異なるが、標準インタフェースを定義しサービスの実装がアダプタの役割を担うことで、その差異を吸収している。

複合サービスはサービスを組み合わせたサービスであり、原子サービス同様言語グリッドに登録され、言語グリッドを通じて利用される。複合サービスからサービスが呼び出される際も、一旦言語グリッドにリクエストが送信され、アクセス制御が行われた上で該当のサービスに転送される。複合サービスもサービスインタフェース定義に基づきサービスを利用している。言語グリッドでは実行時に複合サービスが呼び出すサービスを指定する機構 [10] を導入しており、サービス構成の動的な変更が可能である。図2に複合サービスの構成例を示す。この構成では、日本語形態素解析である JUMAN++ [11] と、大学生協辞書および LSD [12] (Life Science Dictionary) を組み合わせた辞書横断検索サービス、および機械翻訳である GoogleTrans を組み合わせている。利用する辞書や言語により構成は変更されるが、アプリケーションや複合サービスを修正する必要はない。

すなわち言語グリッドでは、自然言語処理分野の技術に対して、以下の機能を提供している。

- サービス情報の検索・登録・削除・変更機能
- サービスの実行およびアクセス制御機能
- 複合サービスの実行および動的構成変更機能

これらの機能およびサービス自体の機能を利用して、多言語アプリケーションが構築される。例として、日本の農業専門家とベトナムの農家を繋ぐ多言語農業支援 [13] や、国際会議における講演の多言語化 [14] が挙げられる。これらのシステムでは、言語グリッドで定められた言語サービスインタフェースに基づいてアプリケーションプログラムが構築され、利用者の言語やサービスの提供状況、品質などに応じて構成が変更されながら複合サービス

が利用された。

機械学習、特に近年大きく発達した深層学習を用いた技術は、オープンソースライブラリの提供や企業が用意するクラウドを通じた提供も盛んであり、様々な機能が利用できる。しかしサービスコンピューティングの導入は進んでおらず、またインタフェース定義の標準化も行われておらず、各機能が実際に利用できるようになるまでの手順もクラウド毎やオープンソースライブラリ毎に異なるため、言語グリッドのように手軽に利用し柔軟に組み合わせることはできない。以降の章では、言語グリッドで提案された手法を機械学習や深層学習を用いた機能に対して適用するにあたり、適用の容易性や、新たに必要となる仕組みの要件について考察する。

3. 機械学習サービスとインタフェース定義

言語グリッドでは、コーパスなどの言語データや言語処理プログラム毎にサービスインタフェースが定義され、そのインタフェースに基づいてサービスが作成された。サービスコンピューティングを機械学習に適用するにあたっては、まずサービスインタフェースを検討する。サービスインタフェースはサービスとそれを利用するアプリケーションプログラムの接続点であり、抽象度が重要になる。サービスの実装の細部に着目した抽象度の低い定義だと、固有の実装に依存しやすく、複数の実装を同じインタフェースで提供することが難しくなる。例えば画像識別において vgg19 などの深層学習が利用するモデルの種類をパラメータとして採用すると、モデルを用いた認識手法を実装することが前提となり、また予めどのような文字列がどのようなモデルを表すかという知識がサービスとアプリケーションの間で共有される必要が生じるため、柔軟なサービスの提供を阻害する。一方実装上の違いを過度に抽象化すると、メソッド名やパラメータだけではサービスの振る舞いが定まらず、同種の実装を用いて振る舞いの異なる複数のサービスを提供するなど、インタフェース定義以外の手段でサービスの振る舞いを区別する必要が生じる。

また、サービスの種別によっては、従来の Web サービスにおいて一般的であるリクエスト-レスポンスの対応が取れない場合がある。例えば音声認識は人の音声を変換する技術であるが、逐次的な変換を行う場合、どの程度の長さの音声が一文

のテキストに対応するかは実際に認識処理を行わないと明らかにならない。つまり、複数のリクエストが一つのレスポンスになることが起こり得る。そのためメソッド呼び出しのたびに結果が返されるインタフェースではなく、結果がサービスから送られてきたことを通知できるインタフェースである必要がある。

さらに、多くの深層学習技術では、利用者の用意したデータを用いてモデルを一から学習し作成することや、既存のモデルをチューニングして特定のタスクに特化したモデルを作成できる。つまり、サービスが利用される文脈として、学習によるモデルの作成と、作成されたモデルの利用という 2 種類が存在する。学習には通常時間がかかるため、一回のリクエスト/レスポンスでは完結しない。そのため進行している処理の状態確認やキャンセルなどの操作が提供される必要がある。さらに、学習後のモデルは新たな推論に使用されるため、これもサービスとして扱える必要がある。すなわち、サービス呼び出しの結果新たなサービスが生成されることがインタフェース定義上明示される必要がある。サービスを作成するサービスでは、以下のことが行える必要がある。

- サービスの作成
- サービスの作成が完了したかどうかの照会

作成するサービスの種類によって必要な情報は異なるため、サービスの作成そのもののインタフェースは標準化できない。また、サービスの識別子はそれを管理する基盤により管理されるため、言語グリッドのようなアーキテクチャの場合、サービス作成サービスが作成したサービスに発行する ID を、コアノードが管理する ID に変換する必要がある。

表 2 機械学習サービスのインタフェース例

種別	インタフェース定義
画像識別 (Image Classification)	ClassificationResult[] classify(byte[] image, String format, int maxResults)
	ClassificationResult {String label, double accuracy}
物体検出 (Object Detection)	DetectionResult[] detect (byte[] image, String format, int maxResults)
	BoundingBox {int x, int y, int width, int height}
	Detection {BoundingBox area, String label}
	DetectionResult {Detection detection, double accuracy}
音声認識 (Speech Recognition)	String startRecognition (String audioFormat, RecognitionReceiver receiver)
	String processRecognition (String recognitionId, byte[] audio)
	void finishRecognition (String recognitionId)
音声認識結果 受け取り (Speech Recognition Result Receiver)	void onRecognitionResult (RecognitionResult[] results)
	RecognitionResult {int id, String transcript, boolean fixed, double accuracy}

表 3 サービス作成サービスのインタフェース例

種別	インタフェース定義
画像識別作成 (Image Classification Creation)	String createImageClassification (byte[] [] images, String[] formats, String[] labels)
	String createImageClassification (String resourceUrl)
	boolean isAvailable (String serviceId)
	void remove (String serviceId)
物体検出作成 (Object Detection Creation)	String createObjectDetection(byte[][] images, String[] formats, Detection[] detections)
	String createObjectDetection (String resourceUrl)
	boolean isAvailable (String serviceId)
	void remove (String serviceId)

以上を踏まえて設計した機械学習サービスの例を表 2 に示す。表 2 には、機械学習の一種である、画像識別と物体検出、音声認識を利用するためのインタフェースが示されている。画像認識に利用されている keras[16] や物体検出を行う技術である

M2Det[17], 音声認識技術である ESPnet[18][19][20] の入出力を元にこのインタフェースを設計した。画像識別は 1 枚の画像に対して、そこに写っているものが何であるかを推定する技術である。画像を入力とし、テキストを出力とする。物体検出は画像内の複数の物体の位置と名前を検出する技術であり、画像を入力として検出結果を出力とする。また、画像のデータ形式は、画像のバイナリデータ (byte[]) とフォーマット (String) で構成されている。どちらのサービスもインタフェース上は内部実装に依存した情報は存在せず、ただし画像識別特有の、画像とラベルが対応付けられている状態は反映されている。音声認識は、音声認識サービスと音声認識結果受け取りサービスの 2 つから構成されている。前述の通り音声認識は、入力が連続的に発生し、また出力は入力と 1 対 1 では無い。そのためインタフェースにはコールバックを採用し、startRecognition メソッドに音声認識結果受け取りサービスを渡す形を採用している。この受け取りサービスは利用側で実装する想定である。

表 3 では、サービスを作成するサービスのインタフェース例を示している。画像識別サービスの作成では、画像とそのフォーマット、および画像に対応するラベルを指定し、学習を開始する createImageClassification メソッドを定義している。画像とラベルが対応づけられたデータを学習する方法は、画像認識技術において一般的なものである。ただし精度を向上させるためには大量のデータが必要となる場合が多いため、データの位置を、URL を用いて表すメソッドも定義している。近年ストレージサービスが普及しており、それらストレージサービスが格納するデータを指し示す URL を指定するか、または従来の HTTP を用いて画像ファイルとラベル情報が圧縮されたファイルを指定する方法を想定している。また、前述の通り学習には長時間を要するため、学習が完了してサービスが利用可能になったかを調べるための isAvailable メソッド、およびサービスが不要になった際に削除するための remove メソッドを定義している。物体検出作成についても画像識別同様に、作成、利用可否取得、削除の 3 種類に対応したメソッドを定義しているが、一つ目の createObjectDetection の第 3 引数では、表 2 の物体検出で定義されているデータ構造である Detection が使用されている。正解データを学習させてモデルを作成する機械学習で

は、このように、サービス作成と作成されたサービスの利用で同じデータ構造が用いられる。

4. 機械学習サービスのプロトコル

前述の通り、サービスの呼び出しには SOAP が用いられる。SOAP はリクエスト-レスポンス形式のプロトコルであり、一つのリクエストで一つのレスポンスが返されることを想定している。他にも REST[21] や JSON-RPC[22] などのプロトコルも存在する。REST はデータフォーマットに XML を用いるプロトコルであり、JSON-RPC では同様に JSON(JavaScript Object Notation)[23] を用いる。どちらもリクエスト-レスポンス形式を想定している。しかし、音声認識に代表されるように、機械学習ではリクエストとレスポンスの数が異なるものが存在する。そのため SOAP などの従来のプロトコルをそのまま使用することはできない。リクエスト数とレスポンス数が非対称であることを許容するプロトコルを用いて、その上にサービス呼び出しを実現する必要がある。本稿では、従来用いられていた SOAP や JSON-RPC の代わりに、WebSocket[24] を用い、その上に遠隔メソッド呼び出しを実現する仕組みを構築する。SOAP や JSON-RPC は、メソッド呼び出しをデータ列に表現し、それをサービスに送信し、サービス側ではそのデータ列からメソッドやパラメータを復元しその情報を元にメソッド呼び出しを行う。また、そのメソッドが戻り値を持つ場合は、戻り値をデータ列に変換して呼び出し元に返送し、呼び出し側ではデータ列から戻り値を復元して、最初に遠隔呼び出しが行われた際の戻り値として呼び出し元に返す。本稿で提案する WebSocket を用いた

サービス実行では、通常のプロセスとは別に、コールバックによるサービスからの情報通知を実装する。コールバックが行われる際のシーケンスを図3に示す。startRecognition および processRecognition、onRecognitionResult は、同一プログラム内でのメソッド実行を示す。invocation および callback は、通信経路上に流れるデータパケットを表している。スタブとハンドラはそれぞれネットワーク接続を行っている相手に対してメソッド呼び出しをデータ列に変換して送信する役割、受信したデータ列をメソッド呼び出しに変換して実際にメソッド呼び出しを実行する役割を担う。従来のサービスの呼び出しでは、アプリケーションがサービスを呼び出し、サービスはアプリケーションを呼び出さないため、アプリケーションがスタブを持ち、サービスがハンドラを持っていた。本稿で提案する方式では、コールバック、つまりサービスからアプリケーションへの呼び出しが行われるため、双方がスタブとハンドラを備える構成とした。この構成により、音声認識のような、リクエストとレスポンスの数が非対称な場合に、コールバックを利用することで結果を返すことができる。

5. 機械学習サービスの合成

2章で述べたように、言語グリッドでは複数のサービスを組み合わせた複合サービスを作成し、利用している。また、利用する言語やサービスの特性に応じ、実行時に実際に呼び出されるサービスを切り替えることができる。複数の翻訳サービスを合成し障害発生時に代替のサービスを呼び出すフェイルオーバー翻訳サービスや、複数の翻訳サービスでの翻訳結果の品質を比較しより正確な結果を返すサー

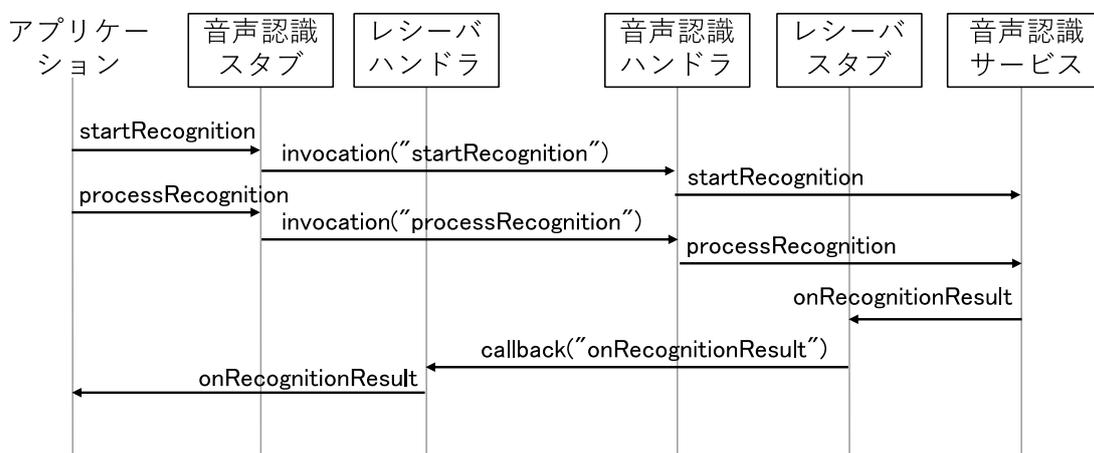


図3 コールバックを導入したシーケンス

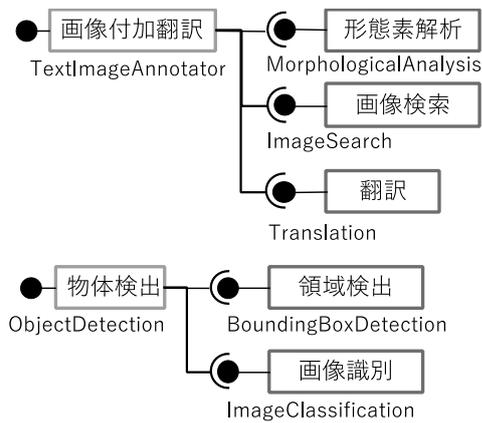


図4 複合サービスの例

サービスなど、機能的、非機能的双方の観点から様々な複合サービスが作成され、利用されている。機械学習に関しても同様に、物体検出を、物体を含む領域を特定するサービスと領域内の画像識別を行うサービスを組み合わせた複合サービスとして提供し、画像識別部分を実行時に指定するものや、複数の画像識別から accuracy の高い結果を選択して返す複合サービスなどが作成可能である。このような複合サービスの例を図4に示す。図内の画像付加翻訳は、機械翻訳と同時に、元言語に含まれるキーワードに関連付けられる画像を検索し、翻訳結果とともに返すサービスである。例えば京都の山鉾巡行など、単純に翻訳しただけでは文化の違いから内容の理解が困難な場合でも、理解を促進するサービスとして利用できる。

さらに、3章で提案したサービスを生成するサービスやコールバックを用いたサービスを利用することで、画像識別と機械翻訳と画像識別作成を組み合わせ、画像とそのラベルの翻訳結果を学習させた新たな画像識別サービスを作成したり、音声認識のコールバック結果を機械翻訳で翻訳する複合サービスを作成することが可能になる。現在様々な分野において機械学習が適用され、利用されているが、大量の訓練データを必要とする深層学習を用いるアプローチが主流であり、データが手に入らない言語では十分精度をあげることが難しい。また、新たに機械学習を導入する場合検証にも時間がかかり、初期費用が高くなり易い。機械学習のインタフェースを定義し、容易に組み合わせる複合サービスを作成できるプラットフォームを提供することで、資源が手に入りやすく音声翻訳が提供されていない言語でも、代わりに音声認識と翻訳を組み合わせるサービスを作成することができる。また機械学習をサービ

スとして利用可能にしておくことで、対象のデータの収集・学習が完了するより前に、似たデータで学習された既存のサービスを用いてプロトタイピングを行いシステムの開発期間を短縮するといった利用方法にもつながると考える。

6. おわりに

本稿ではサービスコンピューティングと複合サービス技術を近年発展している機械学習に適用するための課題を検討し、サービスインタフェース定義やプロトコルの拡張を行った。サービスインタフェース定義では、機械学習サービスの特徴を適切に反映するため、リクエストとレスポンスが1対1では無い場合に対応するコールバックサービス、およびサービスを生成するサービスを導入したインタフェース定義を行った。さらにその実現のため、サービス呼び出しに使用されるプロトコルの拡張を行った。最後に提案・導入したインタフェースを用いて、実際に機械学習サービスを用いた複合サービスが作成できることを示した。

本研究は JSPS 科研費 JP19K20243 の助成を受けたものである。

参考文献

- [1] 青山 幹雄, “サービスコンピューティングへの誘い,” 電子情報通信学, 情報・システムソサイエティ誌, Vol.20.1, pp.8-9, 2015
- [2] Thomas Erl, “Service-oriented architecture: concepts, technology, and design,” Pearson Education India, 2005.
- [3] M.Papazoglou, and W.Heuvel, “Service oriented architectures: approaches, technologies and research issues,” The VLDB Journal 16.3, pp.389-415, 2007.
- [4] Robert Chinnici, et al., “Web Services Description Language Versio 2.0,” W3C, 2007, <https://www.w3.org/TR/wsdl/> (2020/12/6 取得)
- [5] Martin Gudgin, et al., “SOAP Version 1.2,” W3C, 2007, <https://www.w3.org/TR/soap12-part1/> (2020/12/6 取得)
- [6] Tim Bray, et al., “Extensible Markup Language (XML) 1.0,” W3C, 2008, <https://www.w3.org/TR/xml/> (2020/12/6 取得)
- [7] Schahram Dustdar, and Wolfgang Schreiner. “A survey on web services composition,” International journal of web and grid services 1.1, pp.1-30, 2005.
- [8] Jordan Diane, et al., “Web services business process execution language version 2.0,” OASIS Standard, 2007, <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html> (2020/12/6 取得)
- [9] Toru Ishida, Ed., “The Language Grid: Service-

Oriented Collective Intelligence for Language Resource Interoperability,” Springer, 2011.

- [10] 中口 孝雄, et al., “高階関数を導入した階層的サービス合成記述,” 電子情報通信学会論文誌 B 99.10, pp.834-842, 2016.
- [11] Hajime Morita, et al., “Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model,” Proceedings of EMNLP 2015, Conference on Empirical Methods in Natural Language Processing, pp.2292-2297, Sep. 2015.
- [12] Shuji Kaneko, et al., “Life Science Dictionary: A Versatile Electronic Database of Medical and Biological Terms,” Dictionaries and Language Learning: How can Dictionaries Help Human & Machine Learning. The 3rd Asialex Biennial International Conference (Murata M, Yamada S, Tono Y, eds.), 2003.
- [13] 林 冬恵, et al., “言語グリッドを用いた多言語農業支援,” 電子情報通信学会論文誌 B 100.9, pp.714-721, 2017.
- [14] 中口 孝雄, et al., “国際シンポジウムのための多言語支援システム,” 情報処理学会論文誌 58.1, pp.197-204, 2017.
- [15] Yann LeCun, et al., “Deep learning,” nature 521.7553, pp.436-444, 2015.
- [16] François Chollet, “Keras,” <https://github.com/keras-team/keras> (2020/12/6 取得).
- [17] Zhao Qijie, et al., “M2det: A single-shot object detector based on multi-level feature pyramid network,” Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019.
- [18] Shinji Watanabe, et al., “ESPnet: End-to-End Speech Processing Toolkit,” Proc. Interspeech'18, pp.2207-2211, 2018
- [19] Suyoun Kim, et al., “Joint CTC-attention based end-

to-end speech recognition using multi-task learning,” Proc. ICASSP'17, pp. 4835-4839, 2017.

- [20] Shinji Watanabe, et al., “Hybrid CTC/Attention Architecture for End-to-End Speech Recognition,” IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp.1240–1253, Dec. 2017.
- [21] Roy T. Fielding and Richard N. Taylor, “Principled design of the modern web architecture,” ACM Trans. Internet Technol., 2(2), pp.115–150, May 2002.
- [22] JSON-RPC Working Group, “JSON-RPC 2.0 Specification,” 2010, <https://jsonrpc.org/specification> (2020/12/6 取得)
- [23] Tim Bray, “The JavaScript Object Notation (JSON) Data Interchange Format,” STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017.
- [24] Ian Fette and Alexey Melnikov, “The WebSocket Protocol”, RFC 6455, DOI 10.17487/RFC6455, December 2011.

◆著者紹介

中口 孝雄 Takao Nakaguchi

京都情報大学院大学 准教授
株式会社コネクションズ 代表取締役
京都大学 大学院情報学研究科修了 博士（情報学）
京都情報大学院大学修了 情報技術修士（専門職）
京都コンピュータ学院卒業 専門士
専門分野：サービスコンピューティングウェブサービス
最近の関心事：共同作業のための通信機能のサービス化