

特別研究報告書

表形式の言語資源からの
例示に基づくメタデータ抽出

指導教員 石田 亨 教授

京都大学工学部情報学科

石松 昌展

平成 21 年 2 月 2 日

表形式の言語資源からの例示に基づくメタデータ抽出

石松 昌展

内容梗概

現在 Web 上には用語集や対訳集など、多数の表形式の言語資源が HTML で記述され蓄積されている。これらの言語資源をラッピングして Web サービスとしてアクセスできるようにすると、資源間の連携が容易になる。例えば Web 上の医療用語集を Web サービス化して機械翻訳サービスと連携させることで、容易に医療の分野に特化した翻訳が行える。ラッピングには、HTML で記述された表形式の言語資源の内容を解釈し、ラッパープログラムにあわせてメタデータを抽出することが必要となる。しかしながら、Web 上には大量の言語資源が存在するため、抽出の作業には通常多大なコストがかかる。また、様々な構造の解釈を持つ表が存在するため、あらかじめ定めたモデルに従って自動的に処理するのは難しい。

そこで、様々な表からメタデータを自動的に抽出するために、与えられた例示に基づいて表構造を解釈する手法を用いる。言語資源が記述された表は表ごとに構造やその解釈が異なり、全ての表を一つの解釈で処理することができない。これを解決するために、人間に表の一部の解釈を例示として与えてもらい、その例示に基づいて表の解釈を行う手法を用いる。解釈を行う際には、表中のセルの大きさや配置に注目し、例示を与えられた部分と同じ並びのセルを探して繰り返し構造を発見する。これを表構造の一般化と呼ぶ。この一般化を正確に行うことができれば、表構造を正しく解釈することができる。しかしながら、この手法を言語資源に適用する際には、以下のような問題がある。

- 表構造の特徴のみに基づく一般化の誤り

表には構造が同じでも記述されている内容が全く違うようなものが多数存在する。従来手法の一般化の処理では、構造のみの比較で繰り返しを発見しようとする。このため、与えられた例示のどの部分が表のどの部分と一致しているかという判断を誤ってしまう場合がある。

- 与えるべき例示の不明確さ

従来手法では表を自動的に解析する際、ユーザが表に対して適切な例示を行わなければ正しい一般化はできない。しかしながら、何を例示として与えれば良いかは表を見ただけでは明確ではない。このため、ユーザが適

切な例示を与えることは難しく、それに基づく表の解析も失敗してしまうことになる。

このような問題を解決するために、言語資源の持つ制約を利用した。

構造の比較による一般化の誤りについては、繰り返すべきものとそうでないものの区別がつかない場合や、繰り返しの単位が分からない場合が存在する。そこで、言語資源が対訳単位の繰り返しで成り立っているという制約や、1つの対訳の中に同じ言語による複数の対訳文が存在しないという制約を利用して、一般化の誤りを防ぐための処理を実現した。

次に、ユーザの入力による不完全な例示に基づいて一般化を行うことを考える。まず、ユーザの例示を機械による一般化の処理に適したものに近づけるため、例示を行うべき要素をツールがユーザに示し、ユーザはそれに従うだけで例示が行えるようにした。この処理を実現するために、対訳中には必ず2言語以上の対訳文が存在することや、先述の制約などを利用した。また、言語資源をWebサービス化する際に必要となるメタデータの分析を行い、表を構成する要素のうち、どの要素に対して例示を行えば良いかを特定した。ツールは与えられた例示が先述の制約を満たしていることを確かめ、例示が誤っている場合にはユーザに例示の修正を促すようにした。また、与えられた例示による一般化の結果が言語資源の制約を満たすことを確認する処理を追加した。制約を満たしていない場合は、ツールが更なる例示をユーザに求めるようにした。

以上のようにして言語資源の持つ制約を利用することで、表形式の言語資源からメタデータを抽出してするツールを開発した。また、実際にWeb上にある50の言語資源を用いて評価を行った。本研究の貢献は以下の通りである。

- 一般化誤りの減少

言語資源の持つ制約を利用して処理を実現することで、Web上から収集した言語資源を用いて行った評価では、9割の表からメタデータを抽出することができた。

- ユーザの例示の支援

言語資源の持つ制約を利用してユーザの入力や一般化の結果のチェックを行うことで、一般化のために必要となる要素を全て含んだ例示をユーザに求めることができるようになった。また、ツールが例示の選択肢を提示することで、ユーザはインタラクションを行うだけで例示が行えるようになった。

Example-based Metadata Extraction from Language Resources in Table Formats

Masanobu ISHIMATSU

Abstract

The large amount of language resources such as parallel texts and multilingual glossaries are described in HTML and published on the Web. Wrapping these resources as Web services enables us to combine them easily. If we wrap a medical glossary as a Web service and combine it with a machine translator, then we can translate medical sentences easily, for example. In order to wrap a language resource as a Web service, it is needed to understand the contents of the resource and to extract metadata from the resource. However, it takes large costs to extract metadata from language resources because there are many language resources on the Web. Since each language resource has different contents in a same type of structure, it is difficult to extract automatically based on a predefined model.

Then we use a method to extract metadata from table formats based on examples. Since the language resources have various structures and interpretations, it is difficult to understand all language resources based on one model. Using examples which given by users to some parts of the table, we can realize a process to interpret table structures automatically. In order to interpret, we focus on the size and figure of cells in the table. If the cells in the table have same structures with examples, we assume that the cells should have same type of contents, and we regard them the “repetition”. With gathering repetitions from the table, we “generalize” the table structure. If we can generalize the table structures correctly, we can understand the contents of the table correctly. However, to apply this method to interpretation of language resources, there are some problems as follows:

- Errors of the generalization based on only features of table structures

There are many tables which have same structures but their contents are different. The method discovers the repetitions based on only features of table structures. Therefore, errors of generalization often occur.

- Obscurity of examples which should be given

In the method, we should give appropriate examples to generalize the table correctly. However, user cannot understand what examples should be given only by observing the table. So, it is difficult to give appropriate examples and to generalize correctly.

To solve these problems, we use the constraints of the language resources.

There are some errors of generalization such as errors of what should be repeated. So we use the constraints of language resources such that language resources should be described by the repetitions of parallel texts and that a parallel text should not have several expressions of same language. Using the constraints, we can realize the process that keeps out errors of generalizations.

In order to support users to give appropriate examples, we developed a process which shows possible properties of examples. The users can give examples only by choosing from the properties. To develop this process, we also use the constraint that there should be expressions about two or more languages in a parallel text. And, we analyze metadata which should be given to wrap language resources as Web services, then we specified what the properties are needed to wrap. The process checks whether the examples satisfy the constraints, and when the examples have some errors, suggest that users should revise the examples. In the same way, we developed the process which checks the results of generalizations. If results don't satisfy the constraints, the process asks users to try to give examples again.

In this way, we developed a tool which extracts metadata from language resources. And we evaluated this tool by using 50 language resources on the Web. The contributions of this work are as follows:

- Decrease of errors of generalizations

Using the constraints of language resources, we could extract metadata from 90 % of tables collected for our evaluation.

- Supporting users to give examples

For checking the user's examples and the results of generalizations, the user can give the appropriate examples to generalize the table structure correctly. Moreover, users can give examples easily because the tool shows the properties needed for generalizations.

表形式の言語資源からの例示に基づくメタデータ抽出

目次

第1章	はじめに	1
第2章	言語資源の観察	3
2.1	対象とする言語資源	4
2.2	表形式の言語資源の構造	5
2.2.1	表を解釈するための構造の形式化	5
2.2.2	構造の特徴による表の分類	6
2.2.3	記述法による表の分類	6
2.2.4	構成要素による分類	7
2.3	言語資源の持つ制約	8
2.3.1	対訳に関する制約	8
2.3.2	言語資源を Web サービス化するために必要なメタデータ	11
第3章	言語資源の持つ制約に基づくメタデータ抽出	11
3.1	人間の例示に基づく表からの情報抽出	11
3.1.1	例示	13
3.1.2	表構造の一般化	13
3.1.3	メタデータの抽出	15
3.2	言語資源の持つ制約の利用	15
3.2.1	言語資源の持つ制約に基づく表構造の一般化	15
3.2.2	ユーザとのインタラクションによる例示支援	18
第4章	評価	20
4.1	評価方法	21
4.1.1	サンプルデータ	21
4.1.2	評価基準	21
4.1.3	評価手順	21
4.2	評価結果	22
4.2.1	言語資源の持つ制約に基づく処理の有効度	22
4.2.2	言語資源から情報を抽出する際のコスト	23
4.3	考察	24

第5章	おわりに	26
	謝辞	26
	参考文献	27

第1章 はじめに

近年のインターネットの普及により，専門知識を持たないWebユーザでも容易に情報が発信できるようになった．また，日本に住む外国人の増加により，以前に増して外国語と接する機会が増えた．このような要因から，現在Web上には用語集や対訳集などの言語資源が多数存在する．これらの言語資源は，元来の医学用語等の専門分野に関するものだけではなく，小学生向けの簡単な英会話を集めたものや，各国の同じ意味のことわざを集めたものまで，その内容は様々である．また，これらの言語資源の多くは，表形式で記述されている．表は各言語を対訳ごとに並べて記述できるために視覚的に理解しやすく，作成が容易であるため，多くの言語資源提供者から利用されている．

このようなWeb上に大量に蓄積された資源を活用するために，これらの言語資源をWebサービス化することで，他の複数の言語サービスと連携させることを考える．言語資源や言語サービスを構築する際に，自らの手でデータを集め，資源を作成するには多大なコストがかかる．しかしながら，既存の言語資源を利用することができれば，少ないコストで高品質なサービスを提供することができる．このように，既にある言語資源や言語サービスを利用して，少ないコストで新たな言語サービスを提供できるようにする取り組みとして言語グリッドプロジェクト [1] が存在する．言語グリッドプロジェクトは，専門家によって開発された言語資源と，利用現場で作成された言語資源をWebサービスとして登録し，自由に組み合わせて利用することができる取り組みである．したがって，HTMLで記述された対訳集などの言語資源を，言語グリッドのラッパープログラムの形式にあわせて整形することができれば，言語グリッドに登録された様々なサービスで利用できるようになると考えられる．例えばHTMLで記述された医学用語集をWebサービス化すると，辞書連携翻訳サービスと連携させることで精度の高い医療分野の翻訳を行えるようになる．このような利用を考えると，Web上には多くの貴重な資源が蓄積されていることになり，これらをWebサービス化して連携することは非常に有用である．

言語資源をWebサービス化して連携させるためには，その言語資源にメタデータを付与して，共通の「意味」によって扱えるようにするというセマンティックWebのアプローチが有効である [2]．Web上の言語資源は人間が読むことを目的としているため，機械ではその意味を解釈することはできない．そこで連

携する言語サービスが共通に理解できる「意味」を言語資源にメタデータとして与えなければならない。メタデータを付与するためには、その言語資源に何が書かれているのかを理解し、情報を抽出しなければならない。しかしながら、Web 上には大量の言語資源が存在するので、手動でこの作業を行うのはコストが大きい。そこで、多数の表を自動的に処理できることが求められる。

これまでに、表から情報を抽出するための様々なアプローチが提案されてきた。それらは以下のように大きく二つに分けられる。

1. 先見的な表構造のモデルを用いた手法

処理の対象となる表に対して、先見的に与えられた表のモデルによって表を解釈したり [3, 4]、知識ベースによって表を解釈したり [5, 6] して、表からメタデータを抽出するもの。

2. 多数の学習例からの一般化によるモデルを用いた手法

事前に多数の学習例を一般化し、表のモデルを獲得する手法。得られたモデルを用いて、対象となる表からメタデータを抽出する [7, 8, 9]。

しかしながらこれらの手法は、本研究の処理対象である表には適用できない。なぜならば、本研究で扱う表は Web のエンドユーザによって記述されたものが多く、ブラウザを通して視覚的に得られるデータ構造と、その中に記述された情報の意味的な構造が表によって全く異なるためである。また、記述法が統一されていないため、様々な形式の表が存在する。このため、HTML で記述された表形式の言語資源は、それぞれに異なる記述を持つことになり、先見的に与えられた表のモデルをもとに解釈することは難しい。また、多数の学習例を用いる方法も、処理対象の表と同じ構造を持つ多数の学習例を用意することが困難であるため、適用できない。

このような背景を踏まえたうえで、上述の問題を解決するために人間の例示に基づいて表構造を解釈し、メタデータを抽出する手法 [10] を本研究では利用する。この手法では表ごとに人間が例示を与えることから、機械では解釈できないデータ構造と意味的構造の関連を機械に示すことができる。機械はこの例示に基づいて表全体を解釈すれば良い。ただし、ある表に基づいてユーザから与えられた例示は、一般に他の表では有効でないため、表ごとに人間が例示を与えなければならない。多数の表に対する処理を完全に自動化することは困難である。しかしながら、先述の通り、少ないコストで表からメタデータを抽出することができるため、本研究ではこの手法を用いる。この手法を言語資源に適

用する際には以下のような問題がある．

- 表構造の特徴のみに基づく一般化の誤り

表には構造が同じでも記述されている内容が全く違うようなものが多数存在する．従来の手法の一般化の処理では，構造のみの比較で繰り返しを発見しようとする．このため，与えられた例示のどの部分が表のどの部分と一致しているかという判断を誤ってしまう場合がある．例えば，繰り返すべきものとそうでないものの区別がつかない場合や，繰り返しの単位が分からない場合などが存在する．

- 与えるべき例示の不明確さ

従来の手法では表を自動的に解析する際，ユーザが表に対して適切な例示を行わなければ正しい一般化はできない．しかしながら，何を例示として与えれば良いかは表を見ただけでは明確ではない．このため，ユーザが表を観察して適切な例示を与えることは難しく，それに基づく表の解析も失敗してしまうことになる．

このような問題を解決するために，言語資源の持つ制約を利用して処理を実現する．例えば言語資源において，表は対訳の繰り返しで表現されることを利用し，表の解釈の精度を上げる．また，言語資源の構成要素や，Web サービス化のために抽出しなければならないメタデータが限られていることを利用して，エンドユーザの支援を行う．

以上のようにして本研究では Web 上の表形式の言語資源からメタデータを抽出するツールを開発する．本稿では，第 2 章で処理の対象となる言語資源に関する観察について述べる．第 3 章では，言語資源の持つ制約を利用したメタデータ抽出処理について述べる．第 4 章では実際に Web 上にある言語資源を用いて本ツールの評価を行う．最後に第 5 章で本研究のまとめを行う．

第 2 章 言語資源の観察

Web 上にある表形式の言語資源から対訳などの情報とメタデータを抽出し，Web サービス化するために，言語資源を観察しその特徴を調べた．

日本語	英語	備考
あ行 <u>戻る</u>		
アスワン	Aswan	滝:Nile川の滝
アマゾン河	Amazon	河:Andes山脈～ブラジル
ウラル川	Ural	川:ウラル山脈～カスピ海
オリノコ川	Orinoco	川:ベネズエラ
か行 <u>戻る</u>		
カンザス	Kansas	川:カンザス州北東部～Missouri川
ガンジス川	Ganges	川:Himalaya山脈～Bengal湾、インド北東部
黄河	Yellow River, Huan	川:中国

図 1: River

2.1 対象とする言語資源

本研究が対象とする言語資源は Web 上に蓄積された表形式の言語資源である。本研究で扱う言語資源を以下のように定義する。

1. 複数の言語で一つの意味の用語を表現しているような対訳集や用語集。
2. Web 上で表構造を用いて表現されているもの。

例えば図 1 は川の名前を英語と日本語で記述した対訳集である。この表は Web 上で HTML の表構造を用いて記述されている。このような表を本研究で扱う言語資源とする。

次に、言語資源を構成する要素を観察した。すると、表形式の言語資源に記述されるものは以下の 4 つの要素に分類できることが分かった。

1. 対訳の見出し語

図 1 の 1 行目の「日本語」や「英語」のように、その下（あるいは右）に続くセルに何語の対訳文が記述されているのかを示したもの。言語資源によっては存在しない場合もある。

2. 対訳文

言語資源の主たる要素となる対訳文。図 1 の 3 行目の「アスワン」や「Aswan」などで、対訳の見出し語で指定された言語で記述される。また、2 言語以上の同じ意味を表す対訳文の集合を対訳と呼ぶ。例えば「アスワン」と「Aswan」のセットは 1 つの対訳である。

3. 属性の見出し語

対訳の見出し語と同じように、その下（あるいは右）に続くセルにどのような種類の属性地が記述されているのかを示したもの。図1の一行目の「備考」がこれにあたる。

4. 属性の値

属性の値が記述されているセル。図1の3行目の右端のセルのように見出し語がある場合はその見出し語についての具体的な内容が示される。また、2行目のように見出し語が存在しない属性の値も存在する。

2.2 表形式の言語資源の構造

本研究では表形式の言語資源を扱う。表形式の言語資源は様々な構造を持つ。この節では、2.2.1で表を解釈するために構造を形式化し、その後表を処理するためのいくつかの観点から分類する。

2.2.1 表を解釈するための構造の形式化

1. セルの隣接

表の構造を解釈するために、表を行や列の配列とみなす。さらに、行や列はセルの配列とみなす。表には複数の行や列にまたがるセルが存在する場合があります。隣接するセルの幅が異なる場合、そこには通常異なる種類の情報が記述されている。例えば図1の2行目のセルは3列にまたがっている。この行は、上下の対訳などが記述されたセルとは異なり、索引のような役割をしている。このように、あるセルとその隣接するセルの幅の大小関係に着目することで、そのセルを含む行や列の構造を特徴付けることができる。

2. 同じ行や列内のセル

言語資源の対訳は、同じ行や列単位で表現されることが多い。例えば、図1の表では対訳は横に並べられ、属性等も含めて一行でまとめられている。また、上下の行の同じ列のセルには同じ種類の情報が記述されている。すなわち、同じ行や列のセルには強い関連があると考えられる。

3. 繰り返し構造

同じ行や列内で同じ特徴を持つセルの集合が複数回出現する場合には、それらのセルの集合には同じ情報が記述されていると考えることができる。例えば図1では、一行を一つのインスタンスと考えると、各行には同じ情報が記述されている。このように、表は同じ情報を持つセルの集合が繰り返されて表現されている。

PORTUGUESE	ENGLISH
Meu nome é	My name is
Como é seu nome?	What is your name?
Você é brasileiro/brasileira?	Are you Brazilian? (male/female)
O senhor é brasileiro?	Are you Brazilian? (male/mark of respect)
A senhora é brasileira?	Are you Brazilian? (female/mark of respect)

図 2: 1-dimensional table

ことば/Glossary		
	日本語	英語 English
場所/ Places	相談室	Counseling Room
	視聴覚室	Audio-Visual Room
	給食室	School Kitchen
	トイレ	Toilets
	ロッカー	Locker
	昇降口	Foyer/ Entrance hall
	下駄箱	Shoes Cupboard
用具	教科書	Textbook

図 3: Over-expanded label

2.2.2 構造の特徴による表の分類

言語資源から情報を抽出するためには、表の構造を解釈しなければならない。そこで、多くの言語資源を観察し構造上の特徴を調べたところ、以下の2つのタイプに分けることができた。したがって、これらの表の構造を持つ言語資源を解析できれば良いということになる。

1. 1-dimensional table

表の左端もしくは上端に属性が記述される。各属性の値は同じ行あるいは同じ列に記述される。図 2 にこの表の例を示す。

2. Complex table: Over-expanded label

複数の行や列にまたがるセルを含む表。これらのセルは、他の幅の小さいセルと隣接して属性の階層関係を表す場合や、連続するセルに同じデータが記述されている際に一つのセルにまとめられている場合がある。図 3 にこの表の例を示す。

2.2.3 記述法による表の分類

前節で述べたように、表は通常行や列に強い関連がある。このため、行の左端や列の上端には何らかの見出し語がつけられ、その隣や下のセル以降にその見出し語に対応する値が記述されることが多い。こうすることで、その表の構造を視覚的に分かりやすくすることができる。しかしながら、Web 上の表形式

GIS及び防災用語の多言語対応表 英語, 中国語及び日本語 - A

	英語	中国語	日本語
■	A horizon	A層位	A層位
■	abbreviation	縮写词	略記
■	abbreviations	縮写	略記

図 4: 全ての見出し語が記述されているもの

スクリーンセイバー	屏幕保護
自作着メロ	自編鈴音
音声発信	語音撥号
ボイス・メモ	語音記事
バイブレーション	振動提示

図 5: 見出し語が全く存在せず、値のみが記述されているもの

の言語資源には様々な記述法によるものがある。見出し語とその値の記述法に注目して表を分類すると以下ようになる。

1. 全ての見出し語が記述されているもの
全ての対訳にはその同じ行(列)の左端(上端)に言語の種類が記述され、属性にも全て見出し語がついている。図4のような表である。
2. 一部の見出し語のみ記述されているもの
例えば、図3では各対訳部分にはその言語を示す見出し語が上端に記述されているが、その対訳のカテゴリを示す左端の列には見出し語が存在しない。
3. 見出し語が全く存在せず、値のみが記述されているもの
表内のどこにも見出し語となるセルが存在しない。図5に例を示す。
4. 見出し語が階層構造になっているもの
図1に例を示す。全ての見出し語が記述されているだけでなく、見出し語が階層構造により整理されている。

2.2.4 構成要素による分類

表形式の言語資源は、それを構成する要素について分類することができる。以下では、対訳を含む言語資源について構成要素の面から分類する。

1. 対訳のみを含むもの
表中对訳のみが存在する。図2や図5のような表である。

2. 対訳のほかに要素を含むもの

分類や用途，意味などのメタデータとして利用可能な要素を含む．図1や図3のような表である．

2.3 言語資源の持つ制約

本研究では言語資源から対訳などの情報とメタデータを抽出する処理を行う．処理を実現する際に有用となる，言語資源の持つ制約を以下に述べる．

2.3.1 対訳に関する制約

言語資源は対訳が羅列された資源である．この対訳には以下のような制約がある．

1. 一言語で記述される対訳文

対訳とは一つの文を複数の言語で記述した対訳文の組である．それぞれの対訳文は同じ意味であり，それらはそれぞれ一言語で記述されなければならない．

2. 対訳に含まれる一言語の対訳文は一つ

対訳には複数の言語で記述された対訳文が存在するが，一つの言語に関して複数の対訳文が存在することはない．

3. 1つの言語資源に含まれる対訳のセットは1種類

例えば，ある言語資源に日本語，英語，中国語の対訳が記述されている場合，その言語資源に含まれる対訳は全て日本語，英語，中国語をセットとして持つ．

4. 言語資源は対訳単位の繰り返しで構成される

2.2.1 節で述べたように，表は繰り返し構造で表現される．本研究で扱う言語資源の表では，対訳が繰り返し現れる．

5. 表に含まれる各言語の対訳文の数と出現順序は一致する

上記の2，3，4より，言語資源から対訳を言語ごとに取り出した場合，言語ごとの対訳文の数と出現順序は一致する．

上記の制約を分かりやすく示すために，図6にOWLに基づく図1の言語資源の持つ制約の表現を示す．また，図7に図1の言語資源の対訳における図6の制約の例を示す．ここで，`hasPTs`とは言語資源が対訳を持っていることを表すプロパティであり，`hasExp`は対訳が対訳文を持っていることを表すプロパティである．`hasExp`の後ろにつく `Ja` や `En` は言語コードで，対訳文がその言

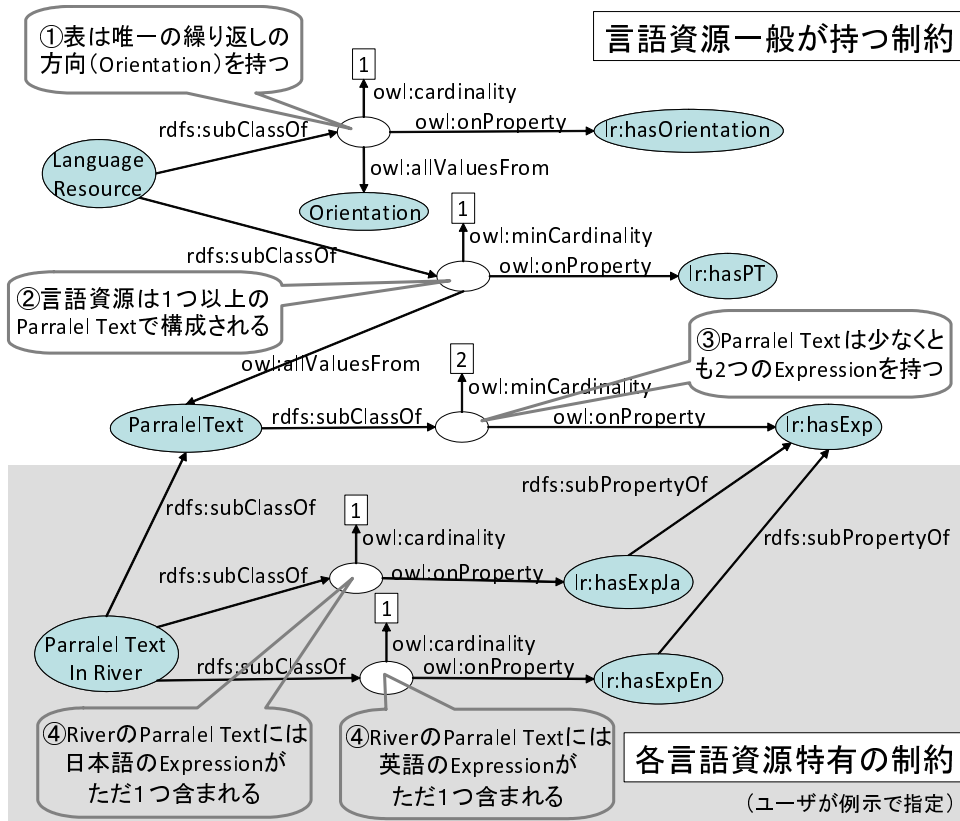


図 6: OWL に基づく図 1 の言語資源の持つ制約の表現

語コードで記述されていることを表している．言語資源の持つ制約は 2 種類に分けられる．1 つは言語資源一般が持つ制約で，もう 1 つは処理の対象となる言語資源に特化した制約である．後者の制約は，ユーザの例示によって得られる．このため，例示で与えられる制約は，言語資源一般の持つ制約を満たしていなければならない．例えば，図 6 の各言語資源特有の制約は，言語資源一般が持つ制約を全て満たしていなければならない．図 6 に記述されている制約は，具体的に以下ようになる．なお，図の番号は以下の番号と対応している．

1. 表は唯一の繰り返し方向を持つ

`hasOrientation` は表が繰り返しの方向を持つことを表している．`hasOrientation` には `cardinality` と `allValuesFrom` の二つの制約があり，それぞれ数が 1 であることと，構成する要素が `Orientation` に含まれる要素からのみ（実際には縦か横）であることが記述されている．

2. 言語資源は 1 つ以上の対訳で構成される

`hasPT` は言語資源が対訳 (`Parallel Text`) を持つことを表している．

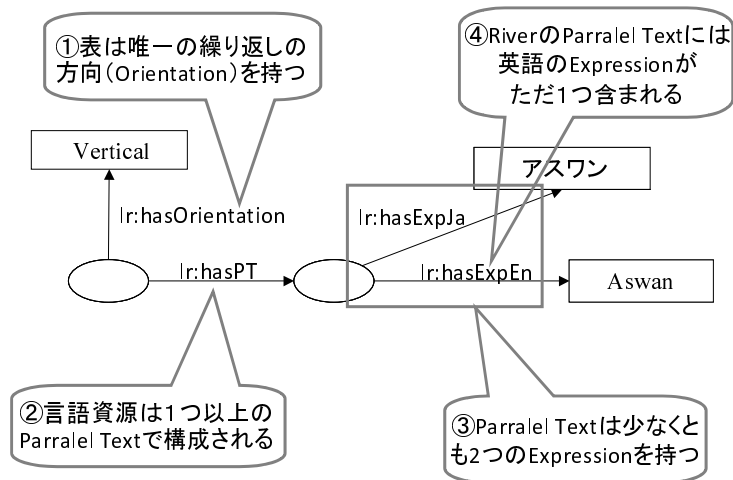


図 7: 図 1 の言語資源の対訳における図 6 の制約の例

hasPT には minCardinality と allValuesFrom の二つの制約があり、数が 1 以上であることと、構成する要素が Parralel Text に含まれる要素からのみであることが記述されている。従って、言語資源はひとつ以上の対訳を含んでいなければならない。

3. 対訳は少なくとも二つの対訳文を持つ

図の中部において、hasExp は対訳 (Parralel Text) が対訳文 (Expression) を持つことを表している。hasExp には mincardinality の制約があり、Parralel Text は少なくとも二つの hasExp を持つことが記述されている。

4. 1 つの対訳にはある言語に関する複数の対訳文は存在しない

図 6 の下部において、hasExpJa や hasExpEn の cardinality は全て 1 となっている。これは、1 つの対訳に含まれる日本語や英語の表現が必ず 1 つだけ存在することを示している。また上の図には明記されていないが、hasExpJa、hasExpEn はそれぞれ 1 つの言語で記述されるという制約を持っている。

5. 言語資源に含まれる対訳は 1 種類

4 より言語資源 River に含まれる対訳は必ず英語と日本語の対訳文を 1 つずつ持っていなければならない。すなわち、言語資源 River に含まれる対訳の種類は 1 種類ということになる。

6. 表は対訳単位の繰り返しで表現される

上記の 2、5 より、言語資源は 1 種類の対訳で構成されることが分かる。すなわち、言語資源の表はこの 1 種類の対訳単位の繰り返しで表現されると

いうことになる。

2.3.2 言語資源を Web サービス化するために必要なメタデータ

言語資源を Web サービス化するためには，ラッピングプログラムの形式に合わせてメタデータを付与しなければならない。ラッピングプログラムは，分類や言語の種類などを指定し，対訳文を呼び出すという処理を行う。このため，言語資源に記述されている情報の中で最も重要なのは対訳であり，次にその対訳文のカテゴリや用途などの情報が重要となる。よって，本研究では，言語資源を Web サービス化するために必要となるメタデータを以下のように定義する。対訳 同じ意味を表す対訳文のセットに対して付与される。

対訳文 パラメータとして，各言語の対訳文ごとに言語コード (ja, en など) を指定する。

属性 対訳についてのカテゴリや用途などの属性を示す。このメタデータは一つの対訳全体に対して記述される。パラメータとして「カテゴリ」などの属性の型を持つ。

第3章 言語資源の持つ制約に基づくメタデータ抽出

言語資源を Web サービス化し，複数の言語サービスと連携することが本研究の目的である。あるコミュニティ内でのみ使われている言語資源を，Web サービス化することで，他のコミュニティの言語サービスと連携させることができ，各言語サービスの品質を高めることができる。このように，言語資源の Web サービス化とその利用を示した図を図8に示す。

本研究では，言語資源の持つ制約を利用して表構造の内容を解釈する方法を提案する。図9にこの処理の全体的な流れを示す。以降の節では，この処理について説明する。

3.1 人間の例示に基づく表からの情報抽出

言語資源からメタデータを抽出するために，表構造を解釈しなければならない。先行研究として，人間が表のモデルを例示として与えて，情報を抽出する研究がある [10]。この研究のアルゴリズムを適用して，表からメタデータを抽出するツールを開発する。このアルゴリズムでは以下の3つのステップからなるアプローチをとる。以降でそれぞれのステップについて説明する。

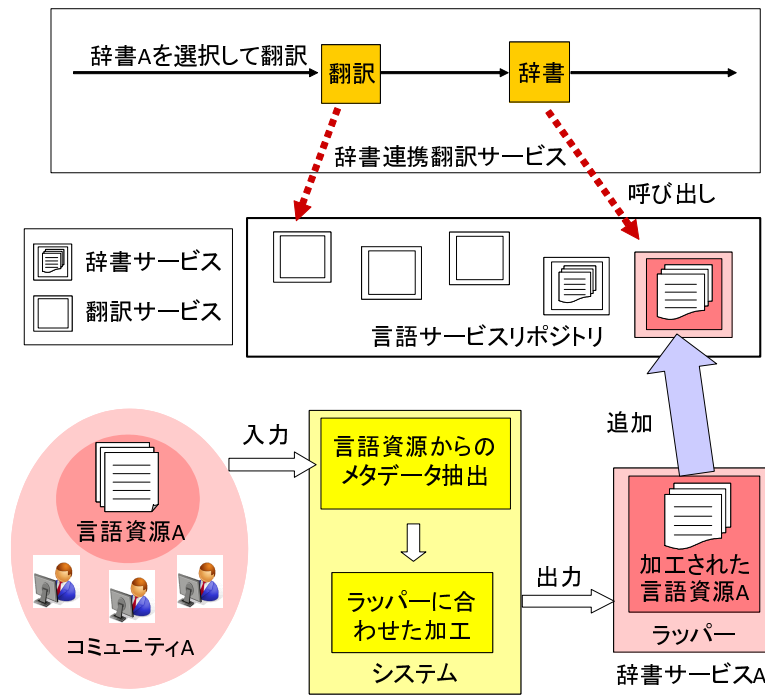


図 8: 言語資源の機械可読化とその利用

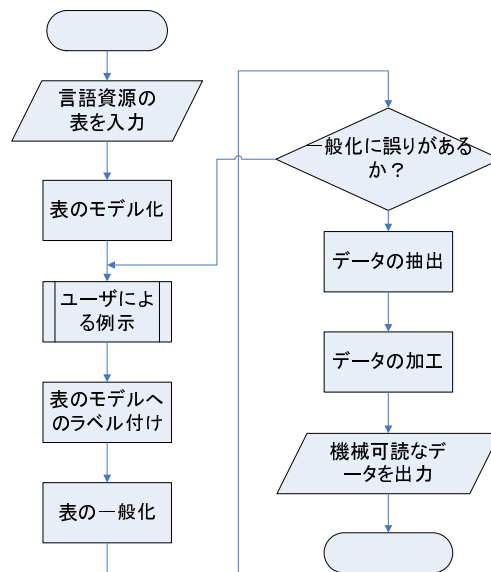


図 9: 処理の流れ

1. 例示によって表構造の解釈を与える
2. 解釈を与えた表構造の一般化
3. メタデータの抽出

日本語	英語	備考
あ行		
アスワン	Aswan	滝: Nile河の・・・
アマゾン河	Amazon	河: Andes山脈・・・
⋮	⋮	⋮

図 10: 図 1 の表のモデル

3.1.1 例示

まずツールは、HTML で記述された表を読み込んで表のモデルを作成する。表のモデルは、HTML の表から表中に含まれるセルの形とその中に記述されている情報を抽出したものである。図 10 に図 1 の表のモデルを示す。これに対し、人間が例示によって表構造の解釈を与える。そのために、まずユーザは表のどの部分が「対訳文」であり、どの部分が「属性」であるのかを示す。また、「見出し語」なのか「値」なのかも記述する必要がある。これを行うために、先行研究では RDF データが用いられている。情報が記述されたセルとセルの関係を RDF で記述することにより、表に意味的な解釈を与えることができるためである。図 1 の表に対する例示は図 11 のようになる。ツールはこれらの例示が行われた箇所を探し出し、その部分にラベル付けを行う。こうすることで、表に対するユーザの解釈を取り入れた表のシンボルを作ることができる。表のシンボルとは、セルを形のみで表したものに、例示によって与えられた解釈をラベル付けしたものの集合で、次の処理で表全体を解釈するために用いる。図 1 の表をシンボルの集合で表すと、図 12 のようになる。

3.1.2 表構造の一般化

ラベルを与えたセルによる構造の表現から繰り返し構造を発見し、繰り返しに対応する生成規則を得る。まず、表を表のシンボルの集合と考える。表のシンボルは、終端のシンボルと非終端のシンボルに分けられる。終端のシンボルは、たいてい言語資源における一つの対訳を表すセルの集合である。2.3 節で述べたように、言語資源は対訳の繰り返しであるから、対訳を表すシンボルはそれ以上分割することはできない。また、属性や見出し語が記述されたセルのみで構成された終端のシンボルも存在する。非終端のシンボルは複数のシンボル

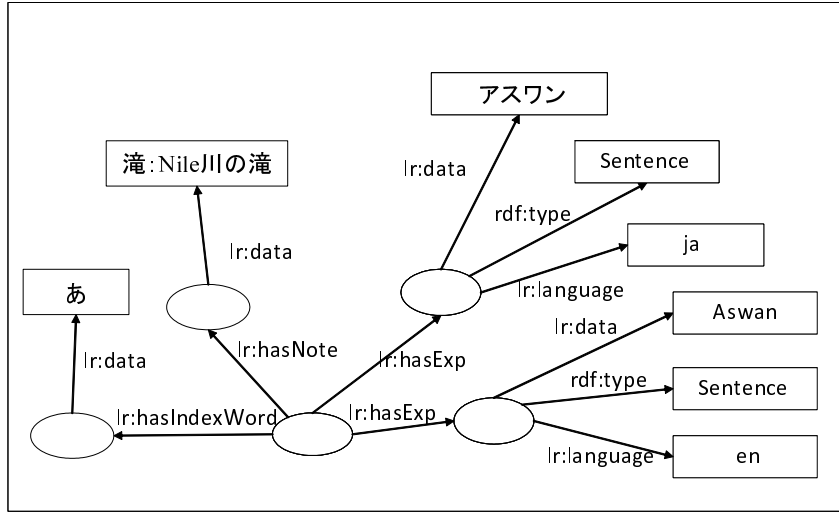


図 11: 図 1 の表に対する例示

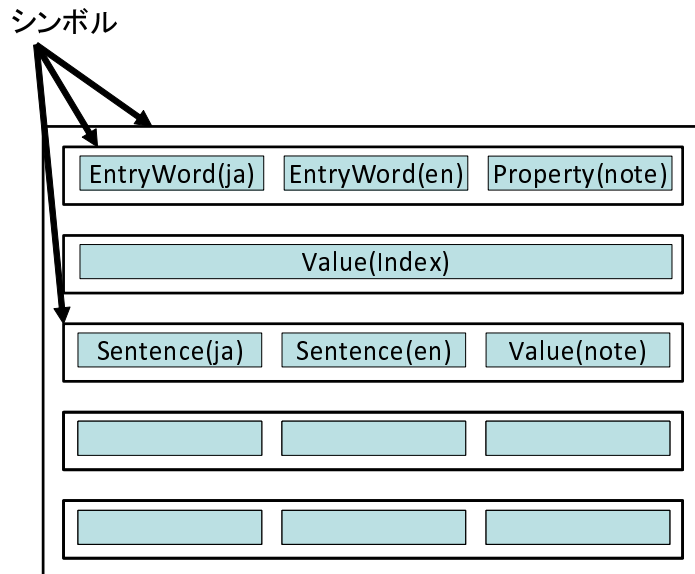


図 12: 図 1 のシンボル

から構成される．このように表はシンボルの繰り返しによって表現されている．
 ツールは例示によって解釈を与えられたシンボルを探し出し，そのシンボルと
 同じ構造を持つシンボルを，同じ情報を持つ繰り返しとみなす．このようにし
 て，表全体をシンボルとその繰り返しで表す．この処理を表構造の一般化と呼
 び，ツールは一般化された表のシンボルのパターンを元に，次の抽出の作業を
 行う．図 12 に図 1 の表のシンボルを示し，図 13 に一般化された表を示す．

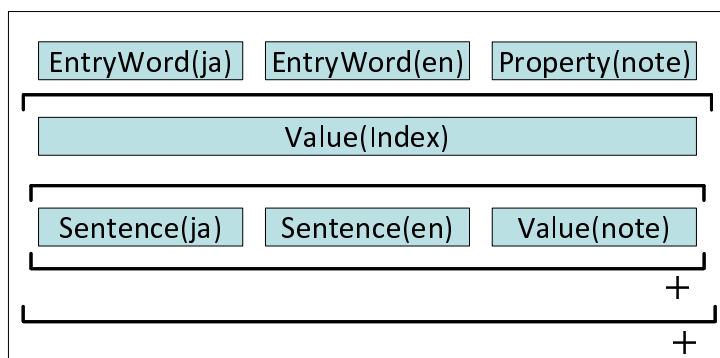


図 13: 図 1 の一般化

3.1.3 メタデータの抽出

ツールは表のモデルと表の一般化構造を比較し、一致した部分からメタデータを抽出していく。この処理を行うためにツールは、一般化構造から決定性オートマトンを作成する。これは、一般化構造を構成する各シンボルに対応する状態遷移を記述することで実現できる。また、決定性オートマトンを用いることで繰り返し構造や非終端のシンボルなどにも対応できる。

3.2 言語資源の持つ制約の利用

3.2.1 言語資源の持つ制約に基づく表構造の一般化

先述のアルゴリズムでは表の構造のみに基づいて一般化するため、しばしば誤りが生じる。この誤りの頻度を減少させるために、2.3 節で述べた言語資源の制約を利用する。2.3 節で述べた図 1 の持つ制約を利用して、処理を実現する。具体的には以下のような制約に着目して例示によって正しく一般化を行うためのチェックを行う。

1. 表は対訳単位の繰り返しで構成される
2. 1 つの対訳にはある言語に関する複数の対訳文は存在しない

まずはこれらの制約をユーザの例示が満たしているかのチェックを行う。具体的には、3.2.2 の節で述べるように、ユーザとインタラクションを行うことで例示を与えさせ、上記の二つの制約を満たすように誘導する。

表から繰り返し構造を発見するための手続きの詳細を示す。手続きは入力として表構造をセルの大きさや形で表した `seq` と言語資源の持つ制約を記述した `const` を受け取る。例として、図 9 の表をもとに説明する。この表の `seq` は図 12 に示すようなシンボルの集合である。また、`const` は図 6 に示すような制約であ

```

discoverRepetition(seq, const){
  requiredLabels = constで指定された, 必要となる
  要素のラベルの集合
  for each seqの部分配列sすべてについて(短いもの順){
    ① if s中にrequiredLabelsが全て含まれている
    ② if s中のrequiredLabelsがconstで指定された
      cardinalityを満たしている
      if seq中にsが繰り返して出現
        seq = seq中で繰り返して出現するsを
        (s)+に置き換えたもの
        return discoverRepetition(seq, const)
      else 例示からやり直し
      else 次に短い部分配列をsとする}
  return seq }

```

図 14: 言語資源の持つ制約に基づく表構造の一般化

る。次に, `const` から対訳を構成する要素の集合である `requiredLabels` を得る。図 9 の表の `requiredLabels` は, 対訳を表す `Sentence(ja)` と `Sentence(en)` である。これは主に対訳の中に必ず含まれていなくてはならない対訳文を表すラベルの集合である。これを用いて繰り返しを発見する処理に移る。まず手続きでは `seq` を可能な限り短いセルの集合に分割して, 部分配列の集合を得る。図 1 の表の場合, 最も短い部分配列は図 12 のそれぞれの行で示される 1 つのシンボルである。処理では, `seq` をこれらのシンボルの配列であると考え, その部分配列と一致する部分を繰り返しとみなして一般化していく。この繰り返しの基準となる部分配列を選ぶ際に, 言語資源の制約を利用して二つのチェックを行う。

まず, 図 14 の 1 のチェックは上記の制約の 1 と対応している。すなわち, 言語資源は対訳単位の繰り返しで構成されていなければならないため, 繰り返しの単位の中に, それを構成する要素を全て含んでいなければならない。図 12 の場合, 繰り返しの単位となる部分配列に `Sentence(ja)` と `Sentence(en)` という `requiredLabels` を全て含んでいなければならない。このチェックによって図 15 に示すような一般化の誤りを防ぐことができる。したがって, 図 12 の一行目や二行目はこのチェックを満たしていないため, 対訳とみなされない。その場合は次に短い部分配列である三行目のシンボルに繰り返しの基準を移して処理を行う。三行目は `requiredLabels` が全て含まれているため, このチェックをパスす

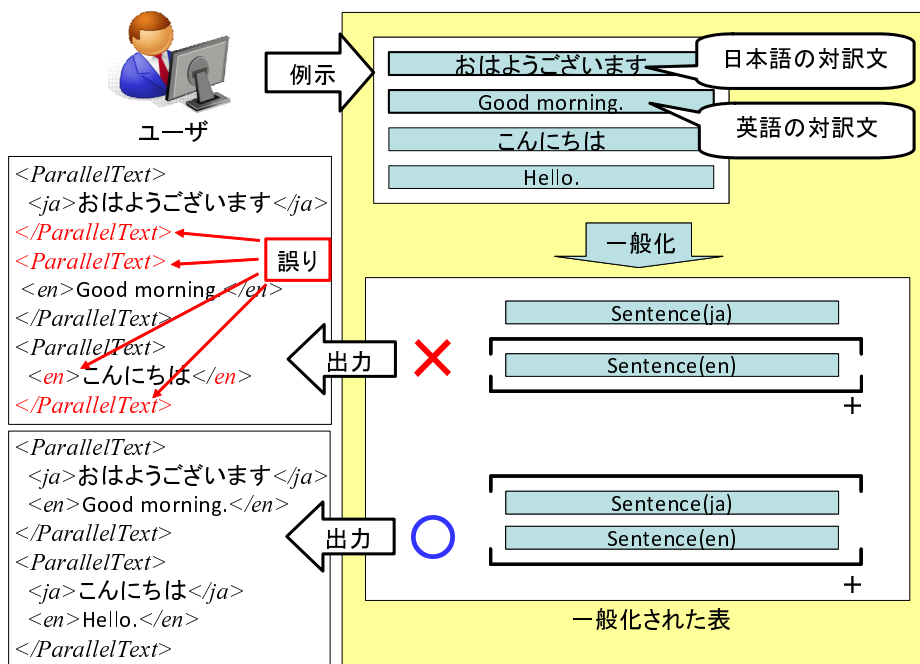


図 15: 対訳単位の繰り返しに関する一般化の誤り

ることができる。

requiredLabels を全て含んでいた場合には次のチェックに進む。ここでは requiredLabels で指定された Label を持つセルの数が、それぞれの Label で指定された数の制限を満たしているかをチェックする。図 14 では 2 の部分に当たる。これは上記の制約の 2 と対応しており、主に 1 つの対訳中にある言語に関する対訳文が複数存在していないかチェックしている。もし、複数存在した場合は一般化が誤っているということになる。この誤りに対して、本当にそこに対訳があるのか別の要素を誤って対訳だと認識しているのかを機械は判断できない。このため、ツールはユーザに例示を行うステップからのやり直しを求める。このチェックにより図 16 に示すような一般化の誤りを発見することができる。

これら二つのチェックをクリアした繰り返しの基準は言語資源の「対訳」に関する制約を全て満たしているため、「対訳」を表していると断定できる。実際に図 12 の三行目のシンボルは二つの条件を満たしており、対訳を表すシンボルとなっている。従ってツールはこの基準をもとに繰り返し構造を発見する。繰り返し構造が発見されると、それを繰り返しを表す新たな記号で置き換える。すると、seq の配列が変更される。図 12 では三行目以降が同じ構造をしているため、対訳であると判断され、三行目のシンボルの繰り返しであるということ

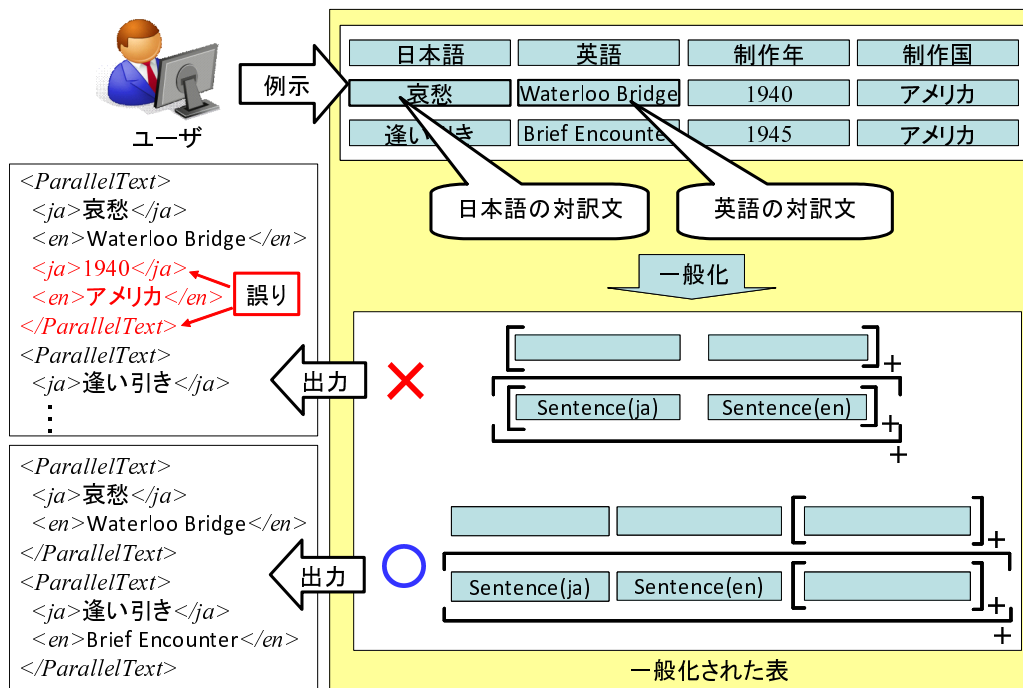


図 16: 対訳内の繰り返しに関する一般化の誤り

示す図 13 の三行目のような新たなシンボルに置き換えられる．この変更された配列に対して繰り返し手続きを適用することで，繰り返しの階層構造を発見することができる．表構造を一般化することができる．この処理を行うことで，図 1 の seq である図 12 からは図 13 のような一般化された表が出力として得られる．

また，このアルゴリズムの計算量と記憶量は，表形式の言語資源に含まれる取り出したい対訳の数を n とすると，次のようになる．計算量は，一つの部分配列につき表全体と比較するという処理を行うため，通常 $O(n^2)$ となってしまうが，繰り返しの基準となる部分配列を変更してさらに繰り返しを発見するという処理の回数をあらかじめ定めた定数とすることで， $O(n)$ となる．また，記憶量も深さ優先探索を行って繰り返し構造を発見し，発見した場合は新しい seq に対してのみ処理を行えばよいので， $O(n)$ となる．

3.2.2 ユーザとのインタラクションによる例示支援

例示に基づいて言語資源の表を一般化する手法では，ユーザの例示で処理が大きく左右される．適切な例示を行うためには，この手法のアルゴリズムを理解したうえで例示を与える必要がある．しかしながらそのようなツールが有用でないことは明らかである．また，先行研究は表中から新たなオントロジーを

獲得することを目的としているため、例示としてRDFデータを入力する必要がある。例えば図1のような言語資源に例示を行う場合、図11のようなメタデータを付与しなければならない。しかしながら、言語資源の提供者がRDFに関する知識を持っていないことは容易に想定される。したがって、一般ユーザでも扱えるようにするため、ツールが表示する選択肢から該当する項目を選択するだけで例示が行えるようにした。ユーザはツールとインタラクションを行いながら例示を行うことで、メタデータに関する知識を持っていなくても処理を実現することができる。これは2.3.2節で述べた、言語資源に付与すべきメタデータが「対訳文」と「属性」に限られることに着目している。以下に具体的なエンドユーザとツールのインタラクションの流れを示す。

1. 例示を行う座標の選択

ツールはまず例示を行いたい座標の選択をユーザに求める。ユーザは例示を行いたいセルの(x,y)座標を入力する。

2. セルに記述されている情報の種類の選択

次にツールは、そのセルに記述されている情報が以下に示す選択肢のどれに該当するかをたずねる。

- 対訳の見出し語 (Entry Word)

例えば、図2の表の「日本語」や「英語」のように、その下（あるいは右）に続くセルに何語の対訳文が記述されているのかを示したもの。言語資源によっては存在しない場合もある。

- 対訳文 (Sentence)

言語資源の主たる要素となる対訳文。実際の文章が記述されている。

- 属性の見出し語 (Property)

対訳の見出し語と同じように、その下（あるいは右）に続くセルにどのような種類の属性地が記述されているのかを示したもの。

- 属性の値 (Value)

属性の値が記述されているセル。

3. 言語コード

上記の選択肢のうち、対訳の見出し語が対訳文であるという選択がなされた場合、ツールはユーザに言語コードの入力を求める。

4. 属性の種類

上記の選択肢のうち、属性の見出し語か値であるという選択がなされた場

合，ツールはユーザにカテゴリや用途などの属性の種類を入力を求める．

以上の情報に対する例示をそれぞれの種類のセルについて一度行えば，例示の作業は終了となる．ただし，特殊な構造をもつ表の場合，複数箇所の同じセルに対して例示を行わなければならない場合もある．また，一つの対訳中で同じ言語コードが指定された場合は，2.3.1 節で述べた言語資源の持つ制約に基づいて，例示が正しくないことをユーザに示すようにした．さらに，対訳内に含まれる対訳文が二つ以上であるという制約を満たしているかのチェックも行い，足りない場合は更なる例示を求めるようにした．以上のような支援を行うことで，専門的な知識を持たないユーザでも，一般化のために必要となる最低限の例示を行えるようになった．図 17 に例示の入力を行う際の処理の流れを示す．

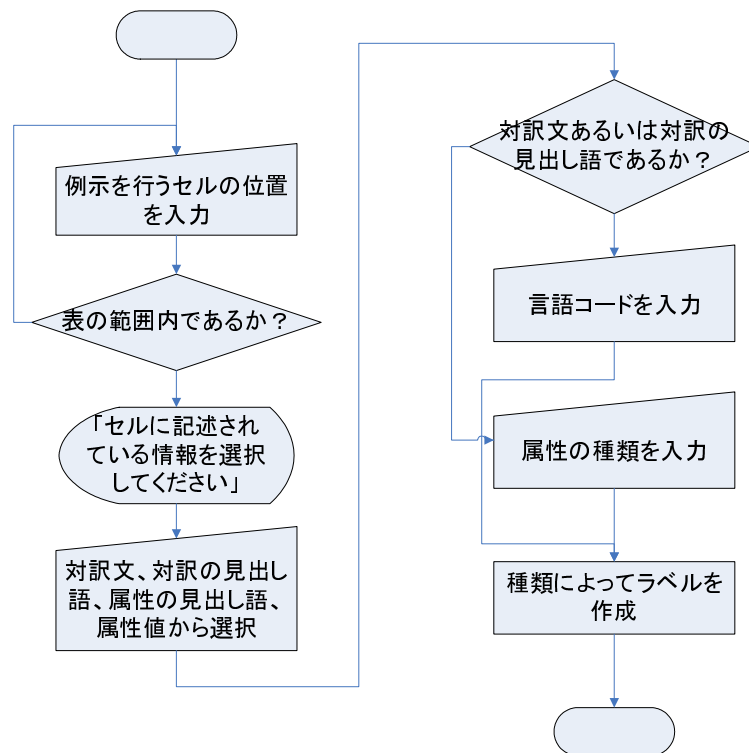


図 17: 例示支援

第4章 評価

前章で実装したツールが，HTML で記述された表からメタデータを抽出し，Web サービスのラッパープログラムに合わせて出力できるかの評価を行った．

また，その際に言語資源の持つ制約を利用する本手法が有効であるかを調べた．最後に，抽出のコストについても評価を行った．

4.1 評価方法

評価は以下のようにして行った．

4.1.1 サンプルデータ

Web 上にある言語資源のリンク集「翻訳と辞書：翻訳のためのインターネットリソース」(<http://www.kotoba.ne.jp/>) から多言語の表を持つページを 50 選んだ．多言語の表を持つページとは，以下の条件を満たすページである．

- HTML で記述された表がある
- 表内には多言語の対訳が記述されている

この 2 点を満たすページを対象に評価を行った．

4.1.2 評価基準

評価は以下の二点について行った．

1. 表形式の言語資源から情報を抽出することができるか

表形式の言語資源を一般化によって解析することができるのか，抽出した情報は言語サービスで利用できるのかを言語資源の制約を用いる場合と用いない場合で調べることで，本手法が有効であるのかを調べた．

2. 小さいコストで表から情報を抽出することができるか

表形式の言語資源から情報を抽出する際に要するコストを，手作業で行った場合と本手法を用いた場合とで比較した．

4.1.3 評価手順

評価は次のようにして行った．まず，表を選択し，例示を与え表中に記述された対訳を全て取り出した．全てを取り出せない場合は，可能な限り取り出した．次に，抽出した対訳を言語グリッドプレイグラウンドの Dictionary Creation というサービスのスキーマにあわせて出力した．この一連の流れの中で前節で述べた評価基準に対して以下のような評価の作業を行った．

1. 言語資源の持つ制約に基づく処理は有用であるか

まず，各々の表形式の言語資源が一般化によって解析できるのかを評価した．その際，対訳部分にのみ例示を与えることで，不完全な例示によって表構造を解釈できるかを検証した．この検証を，言語資源の制約を用いた処理と用いなかった処理とで行い，一般化の精度を比較した．次に，一般

表 1: 一般化による表構造の解釈

	制約を利用しない (%)	制約を利用する (%)
完全に解釈できる	30	54
部分的に解釈できる	16	40
一般化を誤っている	50	2
処理できない	4	4

化できると分かった言語資源について、抽出した情報の評価を行った。具体的には、言語資源から抽出した情報がそのまま言語サービスで使用できる場合を正解とし、再現率と適合率を計算した。

2. 小さいコストで表から情報を抽出することができるか

まず、言語資源から対訳を抽出するために行った例示の回数を数えた。次に、出力した情報がそのまま言語サービスで使える形でなかった場合、利用できるようにするための処理が必要な対訳の数を数えた。手作業の場合、一回の作業で一つの対訳が得られると考え、言語資源に含まれる対訳の数と、処理の際に例示を行ったり、抽出後に手を加えたりした対訳の数の合計とを比較することで、コストの差を明らかにした。

4.2 評価結果

4.2.1 言語資源の持つ制約に基づく処理の有効度

50の言語資源のうち、一般化によって構造を完全に解釈できるものと、部分的に解釈できるもの、一般化を誤っているもの、そもそも処理できないもの数はそれぞれ表1のようになった。完全に構造を解釈できるとは、例示に基づいて表を正しく一般化でき、正しいメタデータが抽出できると言うことである。部分的に解釈できるとは、正しく一般化できるが、抽出されたメタデータに誤っているものがある場合を言う。一般化を誤っているとは、例示に基づいて一般化を行った際に、繰り返し構造の誤った解釈などのために一般化を誤ってしまい、結果として例示とは異なる構造の解釈をしてしまう場合である。また、処理できないとは、一般化によって表を解釈することがそもそも不可能である場合を指す。次に、情報を抽出できた言語資源に関して抽出した情報の再現率と適合率を計算した。言語資源から抽出した情報は、そのまま言語サービスで使

表 2: 抽出した情報の有用性

	制約を利用しない	制約を利用する
適合率	70.02	84.54
再現率	70.20	84.72

表 3: 抽出する際のコスト

	全体に占める割合 (%)
最低限の例示で情報が抽出できた言語資源	96.00
抽出後の情報が言語サービスでそのまま使えた言語資源	44.00
上記の両方を満たす言語資源	40.00

用できる場合を正解とし、その他は不正解とみなした。以下に、言語資源の持つ制約を利用しなかった場合とした場合について、何らかの抽出が行えた表の抽出結果を適合率と再現率で評価した結果を表 2 に示す。

この評価において、抽出した情報がそのまま使えないと判断したのは以下のような場合であった。

- 対訳とは関係のないものが入っている
対訳の後に読み仮名がふってあるものや見出し語が部分的に挿入されているものは誤りとみなした。
- 言語が一致していない
対訳文の中に別の言語による記述がある場合は失敗とみなした。

4.2.2 言語資源から情報を抽出する際のコスト

一般化によって言語資源から情報を抽出できたものに対して、そのコストを評価した。まず、例示に関して、最低限の例示で抽出を行えたものと、そうでないものの割合を調べた。最低限の例示とは、言語資源の構造を解釈するために、見出し語の部分とその値の部分にそれぞれ一度程度の例示を与えることとした。また、抽出後の情報が言語サービスで全てそのまま使えた言語資源の割合を調べた。結果は表 3 のようになった。

また、言語サービスでそのまま使えなかった情報の平均は、言語資源あたり、11.45 個であった。逆に、これらの言語資源から得られたそのまま使える情報の平均は、196.9 個であった。

4.3 考察

多数の言語資源を処理する中で、ほとんどの表は構造の解釈を行えたが、いくつかの表は行えなかった。以下に、抽出を行えなかったいくつかの要因とその対処案を示す。

1. 1つのセルに複数の言語で対訳が記述されている

対訳文は1言語で記述されなければならない。しかし、1つのセルの中に複数の対訳文が記述されている場合は、それぞれの対訳文を区別せずに1つの対訳文として抽出せざるを得ない。例えば図18では一つのセルの中に改行を用いて複数言語の対訳が記述されている。このような場合は抽出した情報を対訳文として扱うことができず、抽出は失敗となってしまった。しかしながら対訳文が混在していても人間にはそれぞれが区別できるような工夫が改行などを用いてなされている。この工夫のルールをきちんと定義できれば、対訳を抽出することができる。

2. 1列に複数の言語が存在する

言語資源の構成として、本研究ではある言語資源に含まれる対訳のセットは1種類であることを前提としている。この前提に反するものがいくつかの言語資源に見られた。例えば図20の表の「原題」という列には様々な言語による記述が含まれるため、対訳として抽出することができなかった。これは、「原題」という列が英語であると解釈して処理を行ったために起こった失敗である。言語資源は1種類の対訳しか持たないという制約に従えば、この「原題」という列は各国の言葉が混在しているため、対訳文の集合ではなく対訳に付与される属性の集合である。しかしながら、言語資源から複数の種類の対訳を抽出できるようになれば、さらに有用なツールが実現できると考えられる。

3. 空白セルの存在

表中に空白セルがあるために対訳として成り立たないにもかかわらず構造が同じため対訳として抽出されてしまうような場合があった。例えば図19は日本語と韓国語の列にそれぞれ空白が存在する。このため、どちらかが空白のセルである場合、抽出は失敗となってしまった。これを単純に解決するならば、対訳文であると例示されたセルが空白であった場合、その対訳文を含む対訳を抽出しなければ良い。しかしながら、図19の表は、韓国

英語	日本語
Absence makes the heart grow fonder 遠ざかるほど、想いはつめる	側に居ないと、想いはつめる
A bad thing never dies. 悪事は決して絶えない	憎まれっ子世にはばかいる

図 18: 1つのセルに複数の対訳文を持つ表

日本語	それに相当する韓国 のことわざ	韓国語の直訳 <small>ほとんど同じ 訳語になる場合や、訳しよのない 場合は空欄としてあります。</small>
愛想が尽きる	정나미가 떨어지다	
朝飯前	누워서 떡먹기	寝ながらもちを食う
	식은 죽 먹기	冷めたかゆを食う
後のまつり	행차후의 나팔	行列の後のラッパ

図 19: 空白のセルを含む表

邦題	制作年	制作国	中国語	原題
8 1/2	1962	イタリア	8 1/2	Otto e Mezzo
Xファイル	1998	アメリカ	X档案-征服未来	X-file
Z	1969	フランス	Z	Z

図 20: 映画タイトル対訳

語の部分が空白の場合と日本語の部分が空白の場合では意味が違う。韓国語が空白の場合は、その対訳文が存在しないという意味であるが、日本語が空白の場合は上のセルと同じ対訳文が入るという意味になっている。これは、この表が日本語を使用するユーザに向けてかかれたものであることを考えれば理解できる。このような、言語資源提供者が対象としている言語の対訳文には、特別な制約がある。この制約を処理に取り入れることができれば、単純に解決するよりも有用なツールが実現できる。

4. 見た目と実際のコードの違い

HTMLでWeb上に言語資源を提供しているユーザは、表を見やすくするために様々な工夫をしている。そのため、見た目では単純な構造に見えても、実際はかなり複雑な構造をしていることがある。例えば、対訳ごとのまとめを見せるために、対訳を含む2つの行の間に、ブラウザ上では表示されない行が挟んであるものなどがあつた。これは、例示を行う際に、ツールが読み込んだ通りの構造をユーザに示す機能を持たせることで解決できる。

以上のようにして、それぞれの問題は解決できると考えられる。しかしながら、これは今後の課題とする。

次に、表を処理する際に、前処理と後処理を行わなければならない表が多数あった。前処理としては、`</tr>` タグが全く存在しない場合や、規定された幅の分のセルが存在しない場合等があり、自動的に処理できなかったため、手作業で保管した。後処理としては、英語の対訳文を記述したセル中であっても、日本語で説明がなされていたりして、英語の対訳文の中に日本語が混じっていることがあったため、これを取り除く必要があった。

第5章 おわりに

対訳集などの表形式の言語資源を Web サービス化して連携するために、表からメタデータを抽出する。その際、一般化の誤りや例示の不明確さなどの問題があったため、言語資源の持つ制約を利用して解決した。また、実際に Web 上にある表形式の言語資源を用いて評価を行った。

本研究の貢献は以下の通りである。

- 一般化誤りの減少

言語資源の持つ制約を利用して処理を実現することで、Web 上から収集した言語資源を用いて行った評価では9割の表からメタデータを抽出することができた。

- ユーザの例示の支援

言語資源の持つ制約を利用してユーザの入力や一般化の結果のチェックを行うことで、一般化のために必要となる要素を全て含んだ例示をユーザに求めることができるようになった。また、ツールが例示の選択肢を提示することで、ユーザはインタラクションを行うだけで例示が行えるようになった。

今後の課題としては、表構造の一般化が行えなかった表に対して新たに言語資源の持つ制約に基づいた処理を加え、抽出できるようにしようと考えている。

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石田亨教授に厚く御礼申し上げます。また、日頃より時間を惜しまず様々なご助言とご協力をいただきました石田・松原研究室の皆様にも心より感謝いたします。

参考文献

- [1] Ishida, T.: Language Grid : An Infrastructure for Intercultural Collaboration, *IEEE/ISPJ Symposium on Applications and the Internet (SAINT-06)*, pp. 96–100 (2006).
- [2] 林良彦: セマンティック Web と言語資源・言語技術, *情報処理学会論文誌*, Vol. 48, No. 8, pp. 857–863 (2007).
- [3] Chen, H., Tsai, S. and Tsai, J.: Mining Tables from Large Scale HTML Texts, *18th International Conference Computational Linguistics (COLING2000)*, pp. 166–172 (2000).
- [4] Wang, H., Wu, S., Wang, I., Sung, C., Hsu, W. and Shih, W.: Semantic search on Internet Tabular Information Extraction for Answering Queries, *9th International Conference on Information and Knowledge Management (CIKM2000)*, pp. 243–249 (2000).
- [5] Embley, D., Tao, C. and Liddle, S.: Automatically Extracting Ontologically Specified Data from HTML Tables with Unknown Structure, *21st International Conference Conceptual Modeling (ER2002)*, pp. 322–337 (2002).
- [6] Tijerino, Y., Embley, D., Lonsdale, D. and Nagy, G.: Ontology Generation from Tables, *4th International Conference on Web Information Systems Engineering (WISE2003)*, pp. 242–252 (2003).
- [7] Hurst, M.: Layout and Language : Beyond Simple Test for Information Interaction-Modelling the Table, *2nd International Conference on Multimodal Interfaces (ICMI-99)*, pp. 243–249 (1999).
- [8] Pivk, A., Cimiano, P. and Sure, Y.: From Tables to Frames, *3rd International Semantic Web Conference (ISWC-04)*, pp. 166–181 (2004).
- [9] 新里圭司, 鳥澤健太郎: HTML 文書からの単語意味クラスの単純な自動獲得手法, *情報処理学会論文誌*, Vol. 48, No. 6, pp. 2140–2152 (2007).
- [10] 田仲正弘, 石田亨: 表構造の一般化に基づくオントロジの獲得, *情報処理学会論文誌*, Vol. 47, No. 5, pp. 1530–1537 (2006).