森羅:Wikipedia構造化プロジェクト2018結果の分析と考察

小林暁雄　　中山功太　　関根聡
理化学研究所 AIP センター
{akio.kobayashi, kouta.nakayama, satoshi.sekine}@riken.jp

1 はじめに

自然言語理解の実現のための言語的及び意味的な大規模知識源の新たな構築方法の枠組みとして、理研 AIP では森羅:Wikipedia 構造化プロジェクト [5, 4] を推進している。このプロジェクトでは、拡張固有表現（ENE）[2] カテゴリに分類された Wikipedia 記事から、各カテゴリに設定された属性の属性値を可能な限り自動抽出することで、更新性・規模ともに優れた知識源を構築することを目標とした共有タスクを実施している。この共有タスクでは、”Resource by Collaborative Contribution（RbCC）”の考え（詳細は [4]）に基づき、各参加システムによって得られた結果をフィードバックすることで知識源を更新し、参加者に還元することで、実行委員と参加者が一体となって最終目標である知識源とその構築方法を作り上げていく。

その第 1 回となる森羅:Wikipedia 構造化プロジェクト 2018 では、ENE カテゴリのうち、Wikipedia 上に記事が多いと考えられるカテゴリなど 5 種類（人名、市区町村名、企業名、空港名、化合物名）を対象として、各属性値の抽出を目的とした共有タスクを実施した。この共有タスクには 8 チームが参加し、15 システムが提案された。提案されたシステムは、パターンベースや、系列ラベリング [3]、機械読解システム [1] に応用したものなど、参加者によって様々なアプローチが採用されていた。このため、各システムの出力結果もそれぞれ異なった傾向がみられた。そこで、森羅プロジェクトでは、RbCC に基づき、これらのシステムの出力から、属性ごと、Wikipedia 記事ごとに適切なものを選び出しより高精度な知識源を構築するためのアンサンブル手法の検討を行った（詳細は [6] を参照）。

一方で、すべての提案システムでも抽出できなかった属性値については、すべての参加システムにとって解決すべき課題である。特に、その原因がデータのスパースネスにあるものについては、例えばクラウドソーシングなどを活用して、訓練データの拡充を行うなどの対策を検討する必要がある。

抽出成功した属性値を精査するアンサンブル学習と、抽出漏れした属性値の調査の両輪で分析を進めることで、今回の共有タスクの結果に対する網羅的な調査・分析を行うことができる。これによって、次回以降の共有タスクにおいて解決すべき課題など、有用な知見を得ることができると期待される。

本稿では、これらのうちの、抽出漏れの属性値の調査・分析について報告する。特に、記事中の記述形式に着目して分析を行うことで各属性の特徴を明らかにするとともに、各属性をその特徴ごとにまとめたクラスに分類し、それぞれのクラスの課題について考察する。

2 調査内容と調査対象について

森羅:Wikipedia 構造化プロジェクト 2018 では、5 種類の ENE カテゴリを対象として、Wikipedia 記事から属性値を抽出したサンプルデータ（各カテゴリ約 600 記事）と、各 ENE カテゴリに分類された Wikipedia 記事の一覧を公開している。参加者は、一覧に含まれる対象カテゴリに分類された記事のうち、サンプルデータにはない記事から属性値を抽出するシステムを考案する。また、各システムの評価用に、サンプルデータと同様に人手により抽出を行った、サンプルデータと重複しない少数のテストデータ（各カテゴリ約 100 件）を用意した。本年度プロジェクトの最終報告回[1]にて、このテストデータによる評価を行い、どのシステムがどのカテゴリに強いのか、どの属性に特に有効だったのかなどについて報告を行った。また、値の抽出漏れについて、属性別の頻度や記事中の記述箇所に基づく大まかな特徴の分析についても報告したが、各属性値の記事中での記述形式に基づく調査・分析・考察は行っていない。このため、来年度以降のプロジェクト参加者に向けてより有益な情報を示すため、本稿では以下の調査・分析を行い、その結果と考察を報告する。

1. どのような特徴をもった属性値が抽出漏れしているのかを明らかにするため、抽出漏れした属性値の類型化を行う。また、比較のため、抽出に成功した属性値についても類型化を行い、それぞれを比較することで、抽出漏れしている属性の特徴を分析する。
2. この類型に基づき、属性間の関係性を調査する。類似した属性をまとめ上げた属性のクラスを設定、調査することで、次回共有タスクにおける課題を属性クラス別に検討可能とする。

本稿では、全システムの出力結果の総和に対して、テストデータ中の属性について、完全一致による評価を行い、完全一致した属性値について抽出成功、それ以外を抽出漏れと判断する。個別のシステムの結果に対しても同様に評価を行い、その結果を調査・分析することも可能だが、森羅プロジェクトではシステム自体の提出を要求していないため、抽出漏れの原因については推測することしかできない。このため、本稿では全システムの総和を対象としている。このため、本稿では、出力結果の精度に関わる、誤抽出については調査・分析を行わない。基本的にはシステム全体としての再現率の向上を目指した分析・考察を行う。精度の向上に関しては、各システムの結果を組み合わせるアンサンブル手法を考案し、スコア全体の向上を実現した。詳細は、[6] にて報告する。

[1] https://aip.riken.jp/labs/goalorient_tech/lang_inf_access_tech/森羅：wikipedia 構造化プロジェクト 2018/森羅プロジェクト 2018 最終報告会/?lang=ja

また、本稿では人名カテゴリを対象とする。人名カテゴリは、すべてのシステム結果を総和した際の再現率が最も低く、他のカテゴリと比較して 10%近く値の低い、65%となっている [6]。このため約三割の抽出漏れが存在しており、最も抽出漏れの調査が必要なカテゴリである。

また、文章中で属性を表す表現を「属性表現」と定義する。例えば、属性「家族」について、”親交を得たミシェル・ロビンソンと結婚。”という文章中には、属性値「ミシェル・ロビンソン」が含まれている。このとき、この属性値が記事の指す概念に対する家族であると判断するための表現「結婚」が属性値の直後に記載されている。この例では、この「結婚」が属性表現となる。

2.1 類型の定義

全参加システムの総和に対して抽出漏れとなっている属性値は、その記述形式が特異だったため、各システムはパターンによる網羅ができなかった、または分類が適切に行えなかったと考えられる。そのような記述形式の特徴を把握することができれば、次回以降の参加者に対して、抽出漏れを低減させるための有益な情報を提供できる。そこで、抽出漏れ、抽出成功の各属性値に対して、それぞれ記事中での記述を人手で確認し、類型化を行った。抽出の評価に用いたテストデータは、アノテーターが Wikipedia 記事を確認し、正解となる属性値を抽出することで構築されている。この際に抽出された値は文字列であり、記事中の記載位置などの情報は含まれていない。このため、記事中の複数箇所に該当する記述がある場合や、同一記事の同一属性内で、2 つの属性値間に包含が発生している場合がある。これについては、以下のルールに基づいて判断した。

- 複数箇所に記述がある場合、他の記事の同一属性の同様の属性値について、その出現場所の傾向から判断する。
- 値に包含がある場合（例えば、「居住地」属性に「東京都台東区浅草」と「浅草」の 2 つが含まれているなど）には、アノテーターがそれら複数の値を採用するに当たり、同一の記述から抽出している可能性はかなり低く、どちらも属性値であると判断できる表記が本文中に分かれて存在していると考えられる。このため、特に包含されている側の値については、それ単独で属性値として判断できる記述を検索し、類型の判断に用いた。

1 つ目については、例えば、出生地は Infobox や、人物の経歴の出だしの両方に記載される場合が多い。この際、同じ種類の Infobox を使用していて（例えば、同じスポーツをしている選手など）、かつ経歴に出生地が記載されていない記事からも出生地の値が抽出されている場合、この種の記事は Infobox から値を抽出していると判断した。

類型一覧

定形表現 Wikipedia 記事の特定の書式に従っており、単純なパターンで属性値が抽出可能なもの。（例:「生年月日」について、”原 哲夫（はら てつお、1961 年 9 月 2 日 - ）”の”1961 年 9 月 2 日”など。）

並列表記 複数の値が同一文、あるいは同一の Wiki 記法による一区画内に出現するもの。（例:「別名」について、”西郷三助・菊池源吾・大島三右衛門、大島吉之助などの変名”など。 ）

表 1: 属性値の類型と頻度

	定形表現	並列表記	Wiki/HTML
抽出成功	200	445	7
抽出漏れ	17	206	62
	推論	属性表現が値周辺に出現	その他
抽出成功	6	295	8
抽出漏れ	28	121	11

表 2: 「属性表現が値周辺に出現」型の属性値の小分類と頻度

	動詞/体言止め	名詞と修飾語	主述関係
抽出成功	185	60	50
抽出漏れ	81	39	1

Wiki/HTML 特定の Wiki 記法のブロックか HTML タグに値が区切られているもの。（表組みの各セル内に記載された値や、Infobox の小見出し名、アスキーアートの一部など。）

推論 属性表現が属性値の周辺に記述されておらず、推論によって属性・属性値間の関係を判断できるもの。（「代表作」について、”「LADY NAVIGATION」が初のミリオンセラーに” について、ミリオンセラーであるなら、代表作であると推論）

属性表現が値周辺に出現 属性表現が属性値の付近に記述されているタイプ。属性表現または属性値の文中での役割によって以下に細分される。

- **動詞/体言止め** 属性表現が動詞または体言止め（サ変）となっているタイプ。（例: 「学歴」について、”駒澤大学仏教学部中退。”より、属性表現は”中退”, 属性値は”駒澤大学” ）
- **名詞と修飾語** 属性表現または属性値の片方が名詞で、もう片方がそれを修飾しているタイプ。（例:「両親」について、”西郷吉兵衛隆盛の長男”より、属性表現は”長男”, 属性値は”西郷吉兵衛隆盛”）
- **主述関係** 属性表現が主語、属性値が述語となっている、あるいはその逆となっているタイプ。（例:上述の「家族」の例など）

属性値が複数のタイプに該当しうる場合、他のタイプに加えて並列表記に該当する場合は並列表記として換算した。これは、抽出漏れについて、並列表記されている複数の値のうち、一部を取りこぼしているケースが散見されたことから、本稿の属性値の再現率を向上させるという目的のために、取りこぼしを低減する方法を検討するためである。

3 調査と分析

3.1 全体の分析

人名カテゴリのテストデータ全体での属性値の類型毎の頻度を表 1 に示す。また、「属性表現が値周辺に出現」タイプの小分類の頻度を表 2 に示す。

並列表記は値が一度に複数個列挙されること、また上述の通り複数タイプに該当する場合には優先してこのタイプ

に分類されるため、頻度は抽出成功・抽出漏れともに最も多くなっている。次に頻度が抽出成功・抽出漏れともに多かったのが、属性表現が値周辺に出現するタイプの属性値である。これは、Infobox などの Wikipedia 特有の構造ではなく、記事本文から抽出する必要のある属性が多かったことを示している。

一方で、抽出成功数がその次に多かった定形表現では、人物一般に共通する属性「国籍」「ふりがな」「生・没年月日」などが大半（詳細は後述）を占めている。定形表現では、抽出成功数に対して抽出漏れがかなり少ない。これは、人名カテゴリについては、上述の属性のようなパターンで記述できる属性の種類があまり多くないことを示しており、属性表現が値周辺に出現するタイプの結果とも一致する。

前述のように、抽出漏れしている属性値は全体の 35%ほどなので、抽出漏れのほうが全体的に少なくなる。その中で、Wiki/HTML と推論については抽出漏れ件数の方が抽出成功しているものよりも多い。

Wiki/HTML については、特定のテーブル表記やリスト表記について、何らかの原因によって、どのシステムもそれらの表記の内部から値を抽出事ができなかったため、それらが抽出漏れとして計上されている。実際、この 62 件の値は特定の記事 2 件に含まれるものである。類似した記述は他の記事でも見受けられるが、他の記事では、何らかの記述がこの 2 記事と異なっており、部分的、あるいは全体的にテーブルやリストから値が抽出できたため、並列表記として計上されている。

推論については、本質的には知識源が必要な問題と考えられる（前述のミリオンセラーが代表作であるといった判断など）。今回の参加システムは知識源を用いているシステムは無いことから、抽出が難しい属性値であったと考えられる。一方で、そのような属性であっても 6 件抽出に成功しているため、今回の調査で発見できなかった特徴がこれら 6 件の周辺に存在している可能性がある。

属性表現が値周辺に出現するタイプの小分類については、まず「動詞/体言止め」タイプの抽出漏れが最多であり、逆に「主述関係」タイプについては、抽出漏れは 1 件のみであった。主語・述語に属性値・属性表現が記載されるような属性値は抽出が容易かったと考えられる。

「名詞と修飾語」タイプについては、最も抽出漏れの比率が多かった。データ中では、助詞で接続されているだけの表記のものに加えて、様々な記述のものが存在していた。修飾されている名詞自体も構文的な特徴を見つけ出すことが困難だった。このため、より詳細な調査を行うためには、システムの挙動を把握する必要があると考えられる。「動詞/体言止め」については、頻出する属性表現は抽出成功している傾向にあった。例えば、属性「家族」については、多くの記事で配偶者を表す属性値が「結婚」という属性表現の付近に出現しており、「結婚」にかかる属性値の抽出性能は高かった。一方で、他の親族を表す表現では、抽出漏れも多く存在していた。

これらの類型の特徴を用いて、属性をクラス分類することで、次回以降のプロジェクトの参加者が解決すべき課題をクラスごとに明確にする。

3.2 属性別の分析

属性毎の各属性値タイプの比率をグラフ 1、2 に示す。

それぞれのグラフより、どの属性値タイプが含まれているか、またその割合がどの程度か、抽出漏れの際にどの属性値タイプのものが残っているか、という観点から、属性の分類クラスを以下のように決定した。

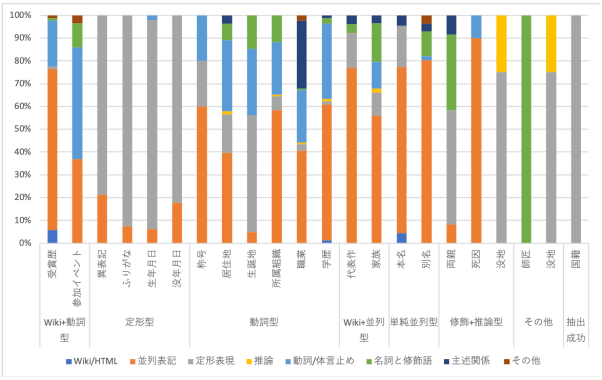

図 1: 抽出に成功した各属性の属性値の種別と比率

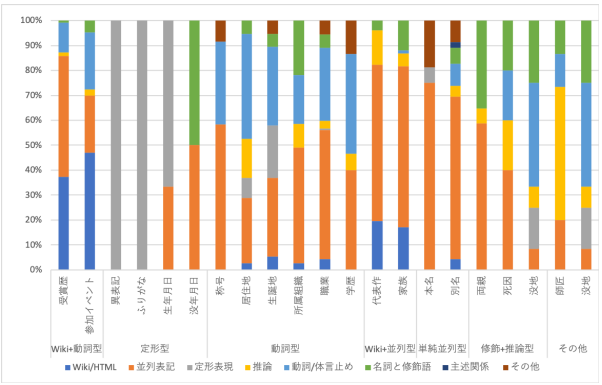

図 2: 抽出漏れした各属性の属性値の種別と比率

Wiki+動詞型クラス 「受賞歴」「参加イベント」が含まれる。参加システムにて学習できなかった、あるいはパターンで網羅できなかった Wiki 記法や HTML で括られた属性値の割合が多い属性が分類される。かつ、属性表現が動詞で記述（「受賞した」や「獲得した」、「参加した」、「戦った」など）されやすい属性からなるクラス。Wiki 記法や HTML に対する前処理などが今後の中心的な課題となる。

Wiki 型の属性値 ナビゲーションボックスのタイトル「TIME パーソン・オブ・ザ・イヤー」など

動詞型の属性値 ”2013 年に WWE 殿堂入り”より、「WWE 殿堂」 など

定形型クラス 「異表記」「ふりがな」「生年月日」「没年月日」が含まれる。定形表現に該当する属性値について、抽出成功、抽出漏れともに多い属性を分類。定形表現のうち、網羅しきれなかったパターンをどのように発見していくかが中心的な課題となる。

異表記・ふりがな ”上杉 謙信（うえすぎ けんしん） / 上杉 輝虎（うえすぎ てるとら）は、戦国時代の越後国の大名。”より、ふりがな「うえすぎけんしん」と、異表記「上杉 輝虎」、「うえすぎてるとら」（異表記は並列表現）

生年月日・没年月日 ”享禄 3 年 1 月 21 日（1530 年 2 月 18 日）”より、「享禄 3 年 1 月 21 日」と「1530 年 2 月 18 日」（並列表現）

動詞型クラス 「称号」「居住地」「生誕地」「所属組織」「職業」「学歴」が含まれる。動詞/体言止めと並列表現タイプの属性値が多かった属性が該当する。抽出成功と抽出漏れ間で比率は変動しているものの、いずれも動詞/体言止めと並列表現のタイプが大半を占めている。抽出成功側では、頻度の高い属性表現「卒業する」や「就任する」「生まれ」などが、抽出漏れ側では、「派遣され」「留学する」などの比較的頻度の低そうな動詞による属性表現が散見された。訓練データ中での頻度が低い、スパースな動詞にどのように対応していくのかが課題となる。

- 「称号」”（博士号などを）授与される”など。「職業」”（職業名）を営む”など。「学歴」”（学校名）卒業”など。「居住地」”（地名）で（職業名）を営む”など。「生誕地」”（地名）生まれ”など。「所属組織」”（組織名）（職名）就任”など。

Wiki+並列型クラス 「代表作」「家族」が含まれる。内部に並列表記が含まれる、Wiki テンプレートや HTML ドキュメントを処理する必要がある属性を分類するクラス。多くの属性値が Infobox や特定の小見出し以下に列挙されており、ここから取りこぼさないようにする手法の考案が課題となる。

- 次の例の属性値の列挙について、1 件でも抽出があれば、並列、そうでなければ HTML 表記などが特殊である可能性があるため、Wiki 型と判断。例 : Infobox の”代表作”属性の値 ”『注文の多い料理店』（1924 年）「雨ニモマケズ」（1931 年）...”

単純並列型クラス 「別名」「本名」が含まれる。Wiki/HTML が少なく、本文中の並列表現が多いクラス。単純並列型クラスは、同列の概念が列挙されている範囲を正しくラベリングする方法を考案することが課題となる。

- ”... 「神君」、「東照宮」、「権現様」（ごんげんさま）とも呼ばれて信仰される。”より、「神君」、「東照宮」、「権現様」、「ごんげんさま」など。

修飾+推論型クラス 「両親」「死因」「没地」が含まれる。属性表現が属性値の修飾語となることが多いクラス。”父親の〜”や”肺がんにより〜”など。修飾+推論型クラスは、動詞型クラス同様、スパースな表現に対応することが今後の課題。推論型の属性値については、記事中の記述によって推論の難易度が異なるため、非常に難しい課題と思われる。

その他 件数が 10 件未満で特徴の判断ができない属性を分類するためのクラス。「師匠」「時代」が該当。

4 考察

表 2、グラフ 1、2 より、絶対数だけでなく、どの属性にも並列表現が多いことがわかる。このため、Wiki+並列型クラスに限らず、全体的に並列表記中の取りこぼしを減らすことが、次回プロジェクトに向けての大きな課題である。リスト表記や Infobox 中での属性値の列挙などは、典型的な例であり、これらは前処理等によって多くの取りこぼしを減らすことができると考えられる。また、単純並列型クラスに該当するような、一文中に列挙されているような値については、並列構造解析などを行うことによって、取りこぼしを減らすことができる可能性がある。

また、属性表現が値周辺に出現するタイプの抽出漏れについては、動詞型・修飾+推論型クラスを中心に、定形型クラス（没年月日の抽出漏れは件数が 1 件のみ）を除く属性全般に出現している。こちらについては、出現頻度の低い動詞や修飾語に対応する処理が重要となってくると考えられる。属性値の近辺に属性表現が存在することから、系列ラベリング手法や、品詞情報や構文情報が属性値の抽出に有用であると考えられる。

本稿の調査は、非公開のテストデータを用いて調査を行っており、この調査で得られた実例は公開できない。また、抽出漏れが起きた原因として、サンプルデータ中での属性表現のスパースさなどが考えられるため、サンプルデータについても、今回と同様の分析を行う必要があると考えられる。

次回プロジェクトの共有タスクでは、HTML 中の値の出現箇所のオフセットを抽出するタスクに変更される。このため、特に並列表現からの値の抽出に関しては、どこからどこまでの範囲に各属性値が列挙されているのかを正しく推定する必要があるため、より重要な課題となると思われる。

5 終わりに

次回プロジェクトに向けて、属性値を、記事内での記述における特徴を類型化し、これに基づき属性を 7 種類のクラスに分類した。これにより、属性クラス毎に解決すべき課題が明らかになった。また、属性値を列挙する並列表記は人名カテゴリの属性全般で非常に数が多く、このタイプの取りこぼしを低減することが、重要な課題であることが明らかになった。

また、クラス別に課題が明らかになったことから、参加者に対して、どのような課題があり、どのようなアプローチが有効と期待できるかといった情報を提示できることが可能となった。これにより、それぞれの課題に対応する小問題を設定し、気軽に取り組んでもらうといったサブタスクなども検討すべきと考えられる。

参考文献

[1] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[2] Satoshi Sekine. Extended named entity ontology with attribute information. In *LREC 2008*, 2008.

[3] Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 2018.

[4] 関根聡, 小林暁雄, 安藤まや. Wikipedia 構造化プロジェクト「森羅 2018」. 言語処理学会第 25 回年次大会, 2019.

[5] 関根聡, 小林暁雄, 安藤まや, 馬場雪乃, 乾健太郎. Wikipedia 構造化データ「森羅」構築に向けて. 言語処理学会第 24 回年次大会, 2018.

[6] 中山功太, 小林暁雄, 関根聡. 共有タスクにおける ga 重み付け荷重投票を用いた属性値アンサンブル. 言語処理学会第 25 回年次大会, 2019.