日本語 Wikipedia カテゴリを用いた記事のクラス分類手法の提案

小板橋佳晃
北海道大学
yk-wwm-aa@eis.hokudai.ac.jp

吉岡真治
北海道大学
理研 AIP
yoshioka@ist.hokudai.ac.jp

1 はじめに

インターネットの発達により人間の膨大な知識が蓄積されてきている。それに伴い、その膨大な知識を機械が利用できるように、機械が扱いやすい構造化された知識の構築が求められてきた。この構造化された知識の知識源として Wikipedia を対象とした研究が近年盛んに行われいる。主に、インフォボックス（ページの主題についての要約情報を提供することを目的とした、記事の右上に配置する形の規定フォーマットの表）やカテゴリといったデータに注目して構造化を行うプロジェクトとして、ページを単位とした固有物に対してメタデータを付与する DBpedia[1] や、それをオントロジーとして整理する YAGO[2] の研究などが代表的である。これに対し、近年、Wikipedia の本文から自然言語処理技術などを活用して、より詳細なメタデータを抽出する森羅 [3] プロジェクトが行われている。

この森羅プロジェクト内のタスクとして、本研究の目的は日本語版 Wikipedia の記事のクラス分類である。私達はこれまで Wikipedia カテゴリに対する分析を行い、カテゴリは記事をクラス分類する上で有益な情報を含むことがわかっている。故に、本研究で紹介する記事の分類手法は、カテゴリを用いて森羅プロジェクトより提供される分類済みの日本語版 Wikipedia の記事の分類を行っている。

2 研究概要

森羅プロジェクトは、辞書の目次をもとに約 200 のクラスを持つクラス階層を定義した拡張固有表現をもとに Wikipedia から属性値を抽出することで構造化された知識の構築を目的としている。タスクとしては、記事のクラス分類と属性値の抽出があり、現在以下二つのサブプロジェクトがある。日本語版 Wikipedia から属性値の抽出を行う森羅 2020-JP と 30 言語版の Wikipedia の記事のクラス分類をおこなう森羅 2020-ML である。本研究では、記事のクラス分類の他言語への拡張を想定し、日本語 Wikipeida の記事のクラス分類を行なっている。トレーニングデータは森羅 2020-ML で提供されるクラス分類済みの日本語版 Wikipedia の記事のデータを用いた。ただしこのデータには、マルチラベルのデータが含まれているが、ノイズになりかねないためシングルラベルのデータのみを使用した。カテゴリは 2020 年 3 月 1 日取得の Wikipedia カテゴリデータを使用した。

3 Wikipedia カテゴリについて

Wikipedia は、世界最大の百科事典である。その中で、カテゴリは閲覧性の向上を目的として記事を分類する役割を果たしている。Wikipedia は誰でも情報の追加や編集が可能であるため、カテゴリには以下の 3 つの定義が示されいる。１つ目は「分割を示すもの」、２つ目は「関連を表すもの」、３つ目は「ウィキペディアの骨組み」である。この中の分割を示すカテゴリは、クラス分類に役立つと考えられる。一方で、関連を表すカテゴリや定義からは外れたカテゴリも存在するため、カテゴリは記事のクラス分類に有益な情報を持つがノイズを多く持つと言える。

私たちは、これまでカテゴリとカテゴリ間の関係を分類し、Wikipedia カテゴリオントロジー [4] を構築してきた。この研究においてカテゴリは以下４つに分類した。

- topic：「北海道大学」「トヨタ自動車」のような具体的な事象を表すカテゴリ
- constrainedTopic：「日本のサッカー」「各年の日本」のような topic の性質を持つカテゴリ
- set：「大学」「企業」といったクラスを表すカテゴリ
- constrainedSet：「日本の大学」「各国の企業」の

ような set の性質を持つカテゴリ

また、我々はこの Wikipedia カテゴリオントロジーを用いた記事のクラス分類を試みた [5]。具体的には、記事につくカテゴリから包含関係を満たす親カテゴリを遡り対応する set カテゴリからクラスを推測する方法を提案した。しかし、図 1 のように対応する set を調べるためにカテゴリを上位に遡ると複数の set が候補として現れ、複数クラスが推定されてしまう課題が生じた。

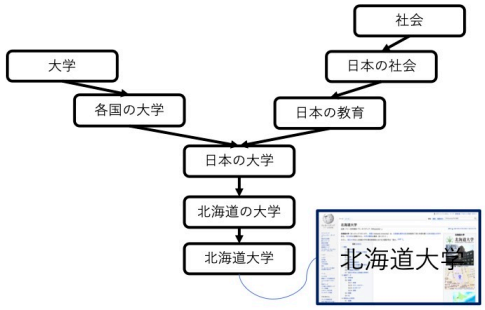

図 1　カテゴリの例

4　提案手法

[5] の方法で問題となった対応する Set を決定する際の問題を解決するために、本研究では constrainedSet の set の文字列部分を抽出することで上位カテゴリに遡らなくても対応する Set を決定する方法を提案する。具体的には、constraindSet の記述パターンに関するこれまでの分析から、constraindSet は「A の B」や「AB」のような記述パターンをもち、B の部分が Set を表す単語になっていることが多いことが分かっている。

4.1　カテゴリの set の抽出

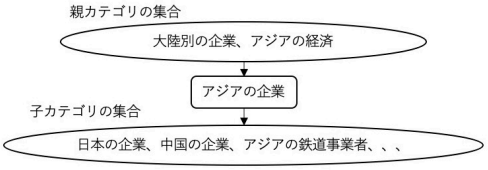

図 2　カテゴリの例

カテゴリの set を抽出すために、カテゴリの記述パターンを利用する。具体的には、図 2 の constrainedSet のカテゴリの例の通り、「アジアの企業」というカテゴリの上位カテゴリと子カテゴリを見ると「A の企業」というカテゴリが存在することがわかる。このように親カテゴリと子カテゴリで後半が一致する文字列を set として抽出した。このパターンによる制約部分の除去の手続きを全てのカテゴリ[1)]間の親子関係について適用し、set の候補を生成した。この候補の生成は、constrainedTopic の関係についても同様に行われることになるが、後半の set 候補の抽出の際に役に立たない情報として扱われることが期待される。

- 文字列パターンの照合
- 形態素解析の利用

文字列パターンの照合では、カテゴリの後方から順次文字列長を増やし親子カテゴリと比較し、同じ文字列パターンが存在する最長の文字列を set として抽出する。また、後方に親子カテゴリと同じ文字列パターンが存在せず、set を抽出できない場合は前方から順次文字列長を増やし親子カテゴリと比較し同じ文字列パターンが存在する最長の文字列をカテゴリから引いた文字列を set として抽出する。これは、図 2 の例において「A の企業」というカテゴリが親子カテゴリに存在しない場合でも前方からの文字列の一致で「アジアの」という文字列が抽出でき、これをカテゴリから引くことで「企業」という set が抽出できるからである。

形態素解析の利用では、後方から単語を順次増やし親子カテゴリと比較し、同じ文字列が存在する最長の文字列を set として抽出した。また、後方に親子カテゴリと同じ文字列が存在せず、set を抽出できない場合は前方から単語を順次増やし親子カテゴリと比較し同じ文字列パターンが存在する最長の文字列をカテゴリから引いた文字列を set として抽出した。

4.2　set 候補の抽出

文字列パターンの照合で抽出した set と形態素解析を利用して抽出した set のそれぞれにおいて、set 候補の抽出を行う。各 set に対して precision を計算し、実験をした結果 0.86 以上を set 候補とした。

$$prec(set, class) = \frac{num(set \cap class)}{num(set)} \tag{1}$$

$num(set)$: 同じ set を持つカテゴリ数
$num(set \cap class)$: 各クラスごとの同じ set を持つカテゴリ数

1)　編集者がメンテナンスをする際に使う隠しカテゴリについては除外した

4.3 カテゴリのクラス分類

文字列パターンの照合から抽出された set 候補と形態素解析を利用して抽出された set 候補と両方を使って抽出した set 候補で全てのカテゴリに対して、クラス分類を行った。その統計データは表 1 に示す。分類済みが 6 割という結果より、日本語 Wikipedia カテゴリオントロジーでは、set および、constrainedSet に対応するカテゴリが 68 %であるため、その全てが森羅のクラス分類と対応するものではないことが確認された。具体的には、set 部分が複数のクラスに対応する（アーチスト：人名 or 公演組織名）場合などが存在した。

表 1 分類、未分類のカテゴリの割合

	分類済み	未分類
文字列パターンの照合	61%	39%
形態素解析の利用	64%	34%
両方利用	67%	33%

4.4 記事のクラス分類

分類済みのカテゴリを使い記事のクラス分類を行う。具体的には、記事につく分類済みのカテゴリの精度 prec(set,class) が最大のものをその記事のクラスとした。また、記事につくカテゴリの中に記事と同名のカテゴリが存在した場合そのカテゴリの親カテゴリからクラス分類を行った。これは、その記事のクラスを表すカテゴリは記事よりもそのカテゴリにつくと考えたからである。

5 結果

実験は 5-fold の cross-validation で行い、f 値はその平均をとっている。実験データは, クラス数：196、記事数：804693 件となっている。文字列パターンの照合によりクラス分類をした結果の f 値ごとの分布を表 2 に、形態素解析の利用によりクラス分類をした結果の f 値ごとの分布を表 3 に、両方を利用しクラス分類した結果の f 値ごとの分布を表 4 に示す。また、全体での精度は、文字列パターンの照合：0.76、形態素分析の利用：0.69、両方利用：0.71 となった。precision を重視したカテゴリの活用を行なったために、一貫して適切な対応するカテゴリが作成・付与されていない場合などに、極端に recall が低い、場合によっては、対応するカテゴリが存在しない状況になった。全体での精度を**??**示す。

表 2 文字列パターンの照合を利用

f 値	クラス数	クラス例（上位 3 つ）
0.9~	13	競走馬名、音楽名、人名
0.8~	23	市区町村名、飛行機名、惑星名
0.7~	20	武器名、公演組織名、都道府県州群名
0.6~	18	家系名、天体名_その他、通過単位名
0.5~	10	罪名、鉱物名、非営利団体名
0.4~	13	鳥類名、国内地域名、地形名_その他
0.3~	14	催し物名_その他、神名、規則名_その他
0.2~	16	地位職業名、会議名、植物名
0.1~	9	ダム名、競技名、施設部分名、
0.0~	22	乗り物名_その他、運河名、病気名_その他

表 3 形態素解析を利用

f 値	クラス数	クラス例（上位 3 つ）
0.9~	9	音楽名、恒星名、人物名
0.8~	17	飛行機名、日付表現、年数期間
0.7~	13	鉄道路線名、法令名、神社寺名
0.6~	6	軍事基地名、鉱物名、競技施設名
0.5~	12	魚類名、衣類名、真菌類名
0.4~	14	海洋名、美術博物館名、公共機関名
0.3~	15	規則名_その他、地位職業名、国名
0.2~	15	宗教名、施設名_その他、哺乳類名
0.1~	15	主義方式名_その他、バーチャルアドレス名、国際組織名
0.0~	37	会議名、公園名、交通施設名_その他

表 4 文字列パターンの照合を利用

f 値	クラス数	クラス例（上位 3 つ）
0.9~	8	音楽名、恒星名、人名
0.8~	17	学校名、河川名、年数期間
0.7~	19	橋名、競技施設名、動物病院名
0.6~	15	通過単位名、鳥類名、料理名
0.5~	12	衣類名、魚類名、新聞名
0.4~	16	劇場名、発電所名、化合物名
0.3~	23	催し物名_その他、食べ物_その他、交通施設名_その他
0.2~	12	哺乳類名、政治的組織名_その他、キャラクター名
0.1~	14	国際組織名、ダム名、組織名_その他
0.0~	23	動植物園名、チャンネル名、芸術作品名_その他

6 考察

まず、文字列パターン照合を利用した結果を考察していく。f 値が高いクラスに関して抽出した set を見てみると、競走馬名では「の競走馬」「種牝馬」という set が抽出され、トンネル名では「州地方のトンネル」「の道路トンネル」といった set が抽出されており、クラス名と対応する set が抽出できていることがわかる。

一方で、f 値が低いクラスに関して抽出した set を見てみると、グループ企業では「企業グループ」という set が抽出されており precision は 0.82 であるが recall が 0.076 ととても低い。理由は、ページにつくカテゴリに「企業グループ」という set を持つカテゴリが存在せず、「企業」の set を持つカテゴリが多いからである。これは、編集者がグループ企業と企業の区別を意識せずにカテゴリ付けをしてしまっているからであると考えられる。また例外的に、precision が 0.034 と極端に低く recall も低いクラ

スが存在し、大陸地域名である。このクラスに分類されている記事を見てみるとノイズと思われるものがとても多く含まれており、これが原因だと考えられる。

次に、形態素解析を利用した結果を考察していく。f 値が高いクラスに関して抽出した set を見てみると、学校名では「航空学校」「私立中高一貫校」という set が抽出され、音楽名では「曲」「卒業ソング」という set が抽出されおり、クラス名と対応する set が抽出できていることがわかる。

一方で、f 値が低いクラスに関して抽出した set を見てみると、映像作品名では「スーパー戦隊シリーズオリジナルビデオ作品」「ビデオ・クラップ集」といった set は抽出できており、precision は 0.79 であるが recall が 0.11 ととても低い。カテゴリから set を抽出した段階では、「映像作品」「アニメ作品」といった set として妥当だと考えられるものが抽出できているがこれらの precision が低いため set 候補にならなかった。「映像作品」は、映像作品名以外に芸術作品__その他や公演名にも多く出てきている。この２つのクラスには、音楽系の映像作品が多く含まれていた。これらの記事は、両方のクラスに属すると見なすこともできるため、今後検討が必要である。

最後に、文字列パターン照合と形態素解析を利用したときの違いについて考察していく。実験結果より、文字列パターンの照合の方が f 値が高いクラスが多いことがわかる。precision と recall にも目を向けると両方とも文字列パターン照合の方が高い値を示しているが、recall の差が precision に比べて大きなっている。これは、形態素解析では単語ごとに区切って set を抽出しているが文字列パターンの照合で一文字づつ文字列を区切っているため、単語の途中まで文字列パターンが親子カテゴリと一致している場合でも set を抽出できるためである。具体的に、家系名クラスに分類されたカテゴリの数を見ると、文字列パターンの照合では 1097 個、形態素解析の利用では 637 個となっている。また、映画名クラスに分類されたカテゴリの数を見ると、文字列パターンの照合では 4175 個、形態素解析の利用では 3051 個となっている。

また、文字列パターンの照合と形態素解析の両者を使った結果、分類済みのカテゴリが増え recall は上がった precision が下がってしまった。両者を組み合わせる際は各々で set 抽出を抽出する際、precision を上げる必要がある。

7 おわりに

本研究は、30 言語版 Wikipedia の記事のクラス分類を念頭に、日本語 Wikipedia の記事の分類をカテゴリを用いて行なった。Wikipedia カテゴリの曖昧さやトレーニングデータのクラス分類の曖昧さにより記事のクラス分類がでうまくできないクラスが存在した。また、データが不十分なニッチなクラスも多くそれが原因でクラス分類ができないものも存在した。ニッチなクラスについてはトレーニングデータを増やすことで改善できると考えられる。一方で、本研究の手法により十分にクラス分類できたクラスもある。今後は、分類済みの日本語 Wikipedia のカテゴリと記事を使い言語間リンクを使って他言語の Wikipedia 記事の分類を行いたい。また、本研究で使った分類手法を他言でも行うことでさらなる精度の向上が期待できるため、それも合わせて今後検討していきたい。

参考文献

[1] Christian Bizer, Jens Lehmann, Georgi Kobi-larov, Soren Auer, Christian Becker, Richard Cy-ganiak, and Sebastian Hellmann. Dbpedia - acrystallization point for the web of data. In *WebSemantics: Science, Services and Agents on theWorld Wide Web, Vol. 7, No. 3, pp. 154 - 165*, 2013.

[2] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. Yago: A spatially and temporally enhanced knowledge base from wikipedia. In *A spatially and temporally enhanced knowledge base from Wikipedia. Arti cial Intelligence, Vol. 194, No. 0, pp. 28 61*, 2013.

[3] Satoshi Sekine, Masako Nomoto, Kouta Nakayama, Asuka Sumida, Koji Matsuda, and Maya Ando. Overview of shinra2020-ml task. In *In Proceedings of the NTCIR-15 Conference*, 2020.

[4] 中川崇教, 小板橋佳晃, 吉岡真治. Wikipedia オントロジーの構築. 人工知能学会研究会資料, Vol.48, No.7, pp.1-5, 2019.

[5] 中川崇教, 小板橋佳晃, 吉岡真治. Wikipedia カテゴリオントロジーを用いた wikipedia ページのクラス分類. 人工知能学会研究会資料, Vol.4, No.5, pp.1-4, 2020.