Wikipediaを用いた語義曖昧性解消のための辞書の自動構築

胡 寅駿　　谷田 泰郎
シナジーマーケティング株式会社
{ko.inshun, tanida.yasuo}@synergy101.jp

1　はじめに

近年，マイクロブログなどのソーシャルメディアの普及により，自分の意見や感想をウェブ上により容易に発信することが可能となった．このようなデータは，マーケティングサイエンス・社会学のみならず，医療・言語学まで幅広い分野にも利用できる．例えば，荒牧ら [1] は，ソーシャルメディア上の情報を用いたインフルエンザ・サーベイランスに注目した．また，谷田ら [3] は，マイクロブログのテキストデータを利用して，発言者の社会的類型・価値観の推定方法を提案した．

上述のような言語処理を用いた研究においては，多くの場合は語義曖昧性解消 (word-sense disambiguation，以下 WSD) の課題があると考えられる．特にツイートデータなどを分析する際に，固有名詞・省略語の WSD (例えば,「マック」はマクドナルドなのか，Macintosh なのかへの判断) は，言語処理を利用したマーケティングサイエンス研究に極めて重要である．本研究では，日本語 Wikipedia のカテゴリ「曖昧さ回避」とテンプレート「Otheruses」の特性に注目し，WSD のための辞書を自動的に構築する手法を提案する．また，構築した辞書 (以下，Wiki 辞書) の評価を通して，辞書の特徴と限界を明らかにする．

2　Wikipediaの曖昧さ回避情報

Wikipedia には，リンク・カテゴリやテンプレートなどのさまざまな (半) 構造化の情報がある．本章では，提案手法に用いられるカテゴリ「曖昧さ回避」とテンプレート「Otheruses」について概観する．

2.1　カテゴリ「曖昧さ回避」

Wikipedia は，記事を分野別にまとめるためにカテゴリを導入しており，これらのカテゴリ情報は言語処理やデータマイニングにとって有力な資源である．中

図 1: 曖昧さ回避記事「LSA」

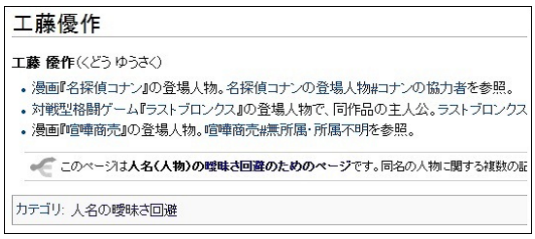

図 2: 人名の曖昧さ回避記事「工藤優作」

でも，記事の多義性を示すためのカテゴリ「曖昧さ回避」がある．このカテゴリを引用した記事 (以下，曖昧さ回避記事) を用いることで，WSD の効果が大いに期待される．図 1 に単語 LSA に対する曖昧さ回避記事を示す．この例では，日本語 Wikipedia における英語略語 LSA は,「アメリカ言語学会」・「潜在意味解析」など解釈が 4 通りあるため，カテゴリ「曖昧さ回避」の引用により LSA が多義であることを表現している。

また，カテゴリ「曖昧さ回避」からの派生としてカテゴリ「人名の曖昧さ回避」が存在する．このカテゴリを引用した記事では，記事名と同名或いは同姓同名の複数の人物が列挙されており，各人物に対する具体的な説明により人名の曖昧さを回避する．図 2 に人名の曖昧さ回避記事「工藤優作」のイメージを示す．

2.2　テンプレート「Otheruses」

Wikipedia において，テンプレートという定型文の入力を汎用化・簡便化するための仕組がある．その中

Copyright(C) 2014 The Association for Natural Language Processing.
All Rights Reserved.

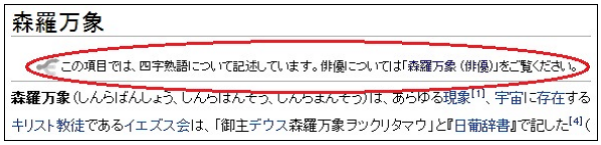

図 3: 記事「森羅万象」

のテンプレート「Otheruses」は，引用される記事に記事名のほか (該当記事以外) の用法を表示させる役割を担う．図 3 にテンプレート「Otheruses」を引用した記事 (以下，Otheruses 記事) の一例を示す．この例では,「森羅万象」は四字熟語のほかにも俳優の森羅万象として使われるため，テンプレート「Otheruses」を引用することにより,「森羅万象」が多義であることを表現している．なお，本研究では，Otheruses 記事が記述する実体を記事名の常用語義として取扱う．例えば，図 3 の場合は，四字熟語の森羅万象を単語「森羅万象」の常用語義とする．上述のテンプレート「Otheruses」以外にも，機能が似ているテンプレート「Otheruseslist」などを引用した記事が Wiki 辞書の構築に利用できる．

3 WSD のための辞書構築

本章では，曖昧さ回避記事や Otheruses 記事から語義名・定義ペア (以下，語義データ) を抽出する手法について説明し，抽出したデータを辞書として利用できるものに洗練する方法を検討する．

提案手法により構築した辞書は，基本的に < 単語, 語義名, 語義の定義 > トリプル群となるが，語義名と定義に対応する記事 (リンク先) が存在すれば，その記事の本文も補足情報として辞書に追加する．

3.1 曖昧さ回避記事の語義データ抽出

箇条書きされた語義　図 1 に示したように，曖昧さ回避記事において，語義群は箇条書きされており，1 項目を 1 語義と見なすことができる．Wikipedia においては，このような箇条書きしたものを記事に表示させるために，ソースコードにマークアップ (markup) と呼ばれるメタ文字列の記入が必要である．表 1 に箇条書きに関するマークアップの例を示す [1]．これらのマークアップは，行頭以外に記入されると箇条書きのメタ文字列として識別されないため，先頭マッチングの正規表現により簡単に語義の抽出を実現できる．

[1] http://ja.wikipedia.org/wiki/Help:早見表 より．

表 1: 箇条書きに関するマークアップの例

種類	入力内容	表示結果
箇条書き	* いち ** いち の いち * に	・ いち ・ いち の いち ・ に
番号付 箇条書き	# いち ## いち の いち # に	1. いち 1. いち の いち 2. に
定義の 箇条書き	; いち : いちの説明	いち いちの説明

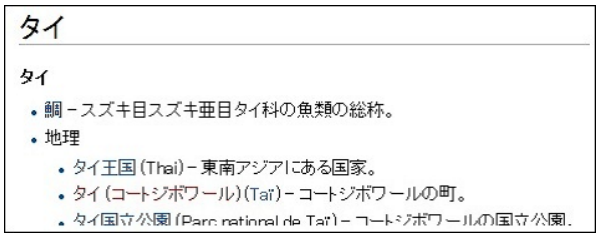

図 4: 曖昧さ回避記事「タイ」の一部

語義群の階層構造　曖昧さ回避記事における語義群は，階層構造になる場合も多く，その一例を図 4 に示す．この場合は，単語の語義を可能な限り細かく分けるために，途中の階層ではなく，最下層にあるものを語義として捉える．例えば，図 4 の曖昧さ回避記事「タイ」においては，

- 鯛 - スズキ目スズキ亜目タイ科の魚類の総称。
- タイ王国 (Thai) - 東南アジアにある国家。
- タイ (コートジボワール) (Ta) - コートジボワールの町。

などを「タイ」の語義とする．

階層構造となった語義の処理　表 1 に示したように，「箇条書き」や「番号付箇条書き」の階層は，ソースコードにおける「*」もしくは「#」の数から読み取れる．しかし，定義の箇条書き「;」・「:」が階層の構築に利用されることもしばしばあり，特に，この 2 つのマークアップが「*」・「#」と同じ記事に出現する場合は，箇条書きの階層関係が判断しにくくなる．例えば，記事「メイフィールド」のソースコードにおいては，

```
; 地名
*[[メイフィールド (都市)]]
```

という記述があるが,「; 地名」と「*[[メイフィールド (都市)]]」の階層関係が判断しにくい．このような階層関係を決めるためには，統一のルールが必要だと考えられる．そこで，本研究では，マークアップについて以下のように階層リスト (階層 1 を最上位の階層) を設定する．

- 「;」のみを利用した項目を階層 1 とする．
- 「:」のみを利用した項目を階層 2 とする．
- 「*」か「#」のみを利用した項目を階層 3 とする．
- それ以外の項目を，階層 $n+2$ とする．

なお，n は「;」・「:」「*」と「#」の総数である．

Copyright(C) 2014 The Association for Natural Language Processing.
All Rights Reserved.

語義データの抽出　本研究では，曖昧さ回避記事における語義データの抽出は以下の 5 ステップにより行う．

- 曖昧さ回避記事に対してノイズ排除 [2].
- 正規表現で項目ごとに分割 (語義の抽出).
- 設定した階層リストにより項目ごとに階層を判断.
- 項目ごとに下位階層がないものを語義として抽出.
- 抽出した語義に対して，さらに「 - 」「とは、」などの特徴文字列を利用して，語義データを抽出.

本手法により，例えば記事「マック」のソースコード

```
** [[日本マクドナルド]] - 日本のマクドナルドの
運営会社。
* [[プロ野球選手]]・[[金子誠]] の愛称。
```

から，語義データ {(語義名:[[日本マクドナルド]], 定義:日本のマクドナルドの運営会社。), (語義名:[[プロ野球選手]]・[[金子誠]] の愛称, 定義:)} が抽出できる．また，人名の曖昧さ回避記事は，曖昧さ回避記事の派生であるため，この手法が適用できる．

3.2　Otheruses 記事の語義データ抽出

テンプレート「Otheruses」は,「曖昧さ回避ページへの誘導」と「他の項目への誘導」の役割を担っている [3].「曖昧さ回避ページへの誘導」は，リンク先が曖昧さ回避記事で前節の手法により抽出できるため，Otheruses 記事から抽出する必要がない.「他の項目への誘導」の場合は，テンプレートに渡された引数が

1. 常用語義に対する定義 (省略可能)
2. 同名他項目 1 の定義
3. 同名他項目 1 の語義
4. 同名他項目 2 の定義
5. 同名他項目 2 の語義
6. .....

という順番になる．提案手法では，Otheruses 記事に対して，語義データを以下のように抽出する．

- テンプレート「Otheruses」の引用部分を取得.
- テンプレートに渡された引数を順次に抽出.
- 引数の文字列に「その他」が含まれる場合，その引数以降のものを削除.
- 常用語義の語義データを抽出
  - 記事名を語義名として取得.
  - 定義が省略されない場合，その定義を抽出.
- 常用語義以外の語義データを抽出.

この手法により，例えば記事「MAJOR」ソースコード

```
{{Otheruses|満田拓也の漫画作品|この作品を原
作とするテレビアニメ|メジャー (アニメ)|その他
の用法|メジャー }}
```

から，語義データ {(語義名:MAJOR, 定義:満田拓也の漫画作品), (語義名:メジャー (アニメ), 定義:この作品を原作とするテレビアニメ)} が抽出できる．また，テンプレート「Otheruseslist」などを引用した記事の語義データもこのような方法で抽出できる．

3.3　語義の展開

3.1 節と 3.2 節で述べた方法により抽出した語義データにはリンク情報が含まれている．例えば，語義データ {(語義名:[[日本マクドナルド]], 定義名:日本のマクドナルドの運営会社。)} では，語義名が Wikipedia の内部リンク [4] により記事「日本のマクドナルド」につながる．本研究では，このような語義名つながる記事を関連記事として語義データに追加する．上述の場合は，語義データを {(語義名:日本マクドナルド, 定義:日本のマクドナルドの運営会社。, リンク先:{ 日本マクドナルド })} に展開する．

4　Wiki 辞書の評価と考察

本章では，Wiki 辞書の実用性を調査するために，Wiki 辞書を用いた WSD の性能を評価し，辞書の特徴と限界について考察を行う．

4.1　基礎情報

本稿では，2013 年 12 月 19 日付の日本語版 Wikipedia ダンプデータ [5] に対して提案手法を適用し，79,590 単語に対する 366,716 語義の Wiki 辞書を抽出した [6]．表 2 に提案手法により抽出した Wiki 辞書の例を示す．

4.2　評価

本研究では，Lesk アルゴリズム [2] をもとに，Wiki 辞書による WSD システムを実装した．このシステムを用いて，ツイートとニュースのテキストデータにおける語義曖昧性を解消し，以下の 2 つの評価を行った．

[2] ここのノイズ排除とは，関連項目・注釈などの語義データ抽出に不用な情報を削除することである．

[3] http://ja.wikipedia.org/wiki/Template:Otheruses

[4] http://ja.wikipedia.org/wiki/Help:リンク

[5] http://dumps.wikimedia.org/jawiki/20131219/，ページ数 1,776,215

[6] 利用したデータ：曖昧さ回避記事 (人名の曖昧さ回避記事も含む) 46,488 件，Otheruses テンプレート (テンプレート「Otheruseslist」などを引用した記事も含む) 36,148 件

Copyright(C) 2014 The Association for Natural Language Processing.
All Rights Reserved.

表 2: 提案手法により抽出した Wiki 辞書の例

単語	語義	定義	関連記事
EU	欧州連合 (European Union)		{ 欧州連合 }
EU	エディンバラ大学 (Edinburgh University)	イギリス スコットランド...	{ エディンバラ大学 }
NLP	自然言語処理... の略称	人間の言語を...	{ 自然言語処理 }
NLP	神経言語プログラミング... の略称	自己啓発技法を...	{ 神経言語プログラミング }

表 3: 評価 1 の結果

課題単語	語義正解率	テキスト正解率
Mac	40%	60%
マック	5%	80%
コナン	0%	90%
ネタ	15%	10%
日本橋	60%	30%

表 4: 評価 2 の結果

テキスト種類	正解率 1	正解率 2
ツイート	46.67%	48.68%
ニュース・経済	47.19%	33.33%
ニュース・IT	39.56%	40.00%
ニュース・エンタメ	33.33%	35.29%
ニュース・政治	33.33%	25.00%
ニュース・スポーツ	46.74%	44.44%

評価 1　評価 1 においては,「マック」などの 5 単語のいずれかが含まれるツイートを 20 件ずつ抽出し，WSD の対象とした．そして，解消した結果に対して，課題単語ごとに語義名・定義の 2 要素による曖昧性解消の正解率 (語義正解率) と，語義名・定義と関連記事の 3 要素による正解率 (テキスト正解率) をそれぞれ評価した．表 3 に評価 1 の評価結果を示す．

評価 2　評価 2 においては，異なるメディアに対する WSD の性能を把握するために，ツイート 100 件とニュース 50 件 (経済などの 5 つの分野から 10 件ずつ) のテキストデータをランダムに抽出し，WSD の対象とした．また，曖昧性解消した結果に対して，すべての対象単語 (Wikipedia にエントリがある単語) の正解率 (正解率 1) と 機械翻訳や意味解析に利用できる対象単語の正解率 (正解率 2) をそれぞれ評価した．なお，評価 2 においては，語義名・定義と関連記事の 3 要素が利用された．表 4 に評価 2 の結果を示す．

4.3　Wiki 辞書の特徴と限界

前節の実験を通して，以下のことが確認できた．

- 関連記事が WSD に有効である (評価 1 と評価 2).
- 「日本橋」のような多義性がある地名には，関連記事による WSD が難しい (評価 1).
- 政治のニュースより IT とスポーツのニュースに対する WSD の効果が良好である (評価 2).
- 辞書ベースアプローチであるため，WSD の性能は語義データに依存する (評価 1 と評価 2).

また，Wiki 辞書の特徴 (+) と 限界 (−) としては，

+ 「マック」や「コナン」のような具体的な固有名詞の WSD に良好な効果が見られる (評価 1).
− 「ネタ」のような抽象的な単語・一般語彙の WSD には効果が見られない (評価 1).
+ 新語・造語が多いツイートデータに対する WSD も可能である (評価 1 と評価 2).

が挙げられる．

5　おわりに

本研究では，日本語 Wikipedia の曖昧さ回避記事と Otheruses 記事から，WSD のための辞書を自動的に抽出する手法を提案した．また，抽出した辞書を用いた WSD システムへの性能評価を通して，Wiki 辞書の新語・造語，或いは英語略語に対する WSD の有効性を確認した．

今後の課題として，WSD 対象に対して語義データに特徴語がない場合の対策方法を探索したい (i.e. 機械学習のアプローチ，語義データの関連情報の拡大)．また，Wiki 辞書の一般語彙への適用が難しいことに対して，ほかの資源 (i.e. 日本語彙大系，日本語 WordNet) との統合に大きな期待を寄せており，その統合方法も検討していきたい．

参考文献

[1] 荒牧英治, 増川佐知子, 森田瑞樹. 文章分類と疾患モデルの融合によるソーシャルメディアからの感染症把握. **自然言語処理**, Vol. 19, No. 5, pp. 419–435, 2012.

[2] Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone *Proc. of SIGDOC*, ACM pp. 24–26, 1986.

[3] 谷田泰郎, 馬場彩子, 河本裕輔, 藤井絵美子. 価値観モデルを利用したマイクロブログ発言者の社会的類型の推定. **言語処理学会 第 *19* 回年次大会**, pp. 628–631, 2013.

Copyright(C) 2014 The Association for Natural Language Processing.
All Rights Reserved.