# グラフクラスタリングによる 単語用例クラスタリング

相原 功昌　佐々木 稔　新納 浩幸
茨城大学工学部情報工学科

## 1　はじめに

ある単語について大量の用例文を抽出し、その用例文集合を語義に基づいて自動的に分類する手法の開発を行っている。国語辞典では知りたい意味についての用例が少なく、多くの語義別用例を分類、提供することでユーザは理解を深めることができる。また、用例文集合の中には、辞典には記載されていない語義として利用された用例も存在するため、辞書の改訂作業を支援することにも応用が可能である。

用例文集合から自動的に語義を分類するために利用される方法として、教師あり学習や半教師ありクラスタリングなどが存在する。教師あり学習ではあらかじめ手作業で語義ごとに分類した教師データを利用して新しい用例を分類する。その場合、大量の教師データを語義ごとに用意することや新しい語義を見つけることへの対応は難しい。また、半教師ありクラスタリングでは、分類する用例文集合内において少量の制約条件を与えることで、大量の教師データを必要とせず、新しい語義への対応も可能な語義別の分類を行うことができる。

これらの手法の他に、最近では HyperLex [1] などといったグラフ分割によるクラスタリングを利用して共起する単語の分類を行う手法が提案されている。グラフを利用する場合、用例文集合に出現する単語をノード、2 単語の共起頻度をエッジとして表現する。このようにして表現したグラフにおいて、高い頻度で共起する単語の組は同じ語義で使用されていると仮定して、共起頻度の高いエッジを持つノード群を求める。それらのノード群に対応する単語集合が同じ語義で共起しやすい単語の集合となる。近年、グラフクラスタリング技術は高速化や性能向上が盛んに行われ、今後も期待できる手法になると考えられる。しかし、単語共起をグラフにより表現する場合、用例文集合に存在する単語が少ないため、共起しやすい特徴を効果的に捉えることが難しいことが大きな問題となる。単語共起のみでグラフを表現すると、頻度の高い共起のみがクラスタとして残るが、頻度は低いが特徴的な単語共起を捉えることは非常に難しい。

本稿では、従来は出現単語をグラフにより表現する手法を改良し、単語の概念をグラフ内に組み込むことで用例文集合に存在する単語の少なさを改善することを提案する。具体的には各出現単語に対して分類語彙表 [2] の分類番号を検索し、その分類番号をグラフ内に追加し、それにより得られたグラフ構造に対してクラスタリングを行う。そのクラスタリング結果に対して、各クラスタ内の単語を含む用例文を導き、用例文を語義毎に分類することを試みる。この用例文のクラスタリング結果を分析し、用例文集合から語義別用例文集合への分類性能を調査する。

## 2　システム概要

本研究では、用例文を分類したい単語（キーワード）を入力すると、その単語の用例文を自動的にクラスタリングして、結果を各クラスタごとに表示するシステムを作成する。そして、このシステムを用いて用例文集合から語義別用例文集合への分類性能を調査することを目的としている。本節ではこの用例文クラスタリングシステムについて、グラフ構造作成部と名詞クラスタリング部、用例文クラスタリング部に分けて述べる。

### 2.1　グラフ構造作成部

キーワードの用例文集合に対して、形態素解析器 mecab を用いて用例文ごとに形態素解析し、名詞だけ

を抽出する。抽出された名詞を、分類語彙表を用いて概念に変換する。ここで、用例文ごとに抽出された名詞は、キーワードと共起しているものと仮定する。

分類語彙表には、一つの単語に対して複数の分類番号が記載されていることがある。分類番号が一意でない場合、どの分類番号に変換するのかが問題となる。しかし、どの分類番号に変換するべきかという情報は持っていないので、全ての分類番号が出現する確率は一様分布に従うと仮定する。従って、一つの単語が複数の概念番号に変換されることがある。また、分類番号は７桁のうち上５桁までを使用し、同じ分類番号の単語がまとめられる。

全ての概念の延べ出現数と、全ての概念の出現頻度を数え、一文に出現する全ての概念の組み合わせに対して相互情報量 [3] を求める。そして、各概念がノード、一文で共起する概念の組み合わせがエッジ、エッジの重みを相互情報量をとする重みつき無向グラフが作成される。

### 2.2 名詞クラスタリング部

作成した概念のグラフ構造を、マルコフクラスタリング（mcl） [4] や Normalized Cut アルゴリズム（graclus） [5] でクラスタリングする。これにより、繋がりの強い概念が同じクラスタにまとめられ、出力される。

mcl は、概念の集合をいくつのクラスタに分類するかを指定しないが、graclus は指定する必要がある。そこで、キーワードの語義数と概念の集合を分類する数が等しいと仮定し、このキーワードの語義数を graclus において指定する。

### 2.3 用例文クラスタリング部

概念のグラフ構造をクラスタリングした結果に対して、各クラスタ内の単語を含む用例文を導き、用例文を語義毎に分類する。一つの用例文に単語が複数存在し、それぞれ違ったクラスタに含まれる場合、全てのクラスタに用例文を導くということは用例文が異なる語義に重複して導かれることを意味する。そのため、一度どこかのクラスタに導かれた用例文は、以降どのクラスタにも導かれないようにするなど、工夫が必要である。本稿では、用例文の重複を許可する場合と禁止する場合について実験している。

## 3 実験方法

検索用データは、「現代日本語書き言葉均衡コーパス」[6] の DVD に収められえている書籍コーパスのサンプル (XML) から、日本語だけを抽出し短文を生成する。この生成した短文を検索エンジン HyperEstraier で検索できるようにする。この際 HyperEstraier 自身が、自らが検索しやすいようにデータを整形・保存したデータベースを作成する。

キーワードを入力し、用例文クラスタリングシステムを実行すると、用例文がクラスタリングされ、Web ブラウザ上に語義別用例文集合が集合毎に表示される。

そこで、あるキーワードについて

1. グラフ構造作成部において、抽出された名詞を
   (a) 分類語彙表の分類番号を考慮しない。
   (b) 分類語彙表の分類番号を考慮する。
2. 名詞クラスタリング部においてグラフクラスタリングを行うツールに
   (c) mcl を用いる。
   (d) graclus を用いる。
3. 用例文クラスタリング部において
   (e) 名詞クラスタリングの結果から直接用例文を導く。この場合、同一の用例文に属する概念が別々のクラスタに分類された時、それぞれのクラスタに用例文を重複して導くことを許可する。
   (f) 複数のクラスタに重複して導かれた用例文は、クラスタの要素の少ない順に導き、一度導かれた用例文はその他のクラスタには導かせない（重複して導くことを禁止する）。
4. 使用する用例文のデータにおいて
   (g) HyperEstraier で検索できる用例文数が約 9 万 6 千件のデータベースを NDB とし、NDB を用いて用例文を検索する。
   (h) HyperEstraier で検索できる用例文数が約 23 万件のデータベースを ADB とし、ADB を用いて用例文を検索する。

の、(a)〜(h) 全ての組み合わせについて用例文分類結果を出し、手動で語義別に分類した正解のデータと比較する。

表 1　正解データ

| | NDB | ADB |
|---|---|---|
| 「走る」の語義数 | 12 | 12 |
| 出現した語義数 | 7 | 9 |
| 有効用例文数 | 45 | 121 |

# 4　実験結果

キーワードを「走る」として用例文を分類したときの正解データを表 1 に、分類結果を解析したデータを表 2 示す。「走る」の語義は『デジタル大辞泉＜小学館＞』のデータを基にし、適切だと思われる数を導き出した。

## 4.1　評価方法

用例文クラスタリングの結果について、有効用例文数・クラスタ数・用例文数が 1 のクラスタ数・正解数・正解率・抽出された語義の 6 つの項目について比較する。ここで、有効用例文数とは、ノイズを含めない用例文の数を表す。ここでいうノイズとは、ある単語の一部にキーワードが含まれているために、本当は別の単語であるにも関わらずクラスタリングされてしまった用例文を指す。クラスタ数は用例文がいくつのクラスタに分かれたかを示す。用例文数が 1 のクラスタは、用例文が分類ができていないとみなし、無効クラスタとする。クラスタ毎に、分類された用例文がどの語義で使われているのかを正解データから調べ、クラスタ内で一番多く使われていた語義の用例文数を正解数として数え、正解率を求める。

正解率は式 1 で求める。

$$\text{正解率} = \frac{\text{分類結果の正解数}}{\text{分類結果の有効用例文数}} \qquad (1)$$

また、評価の指標には正解率だけでなく、抽出された語義数、無効クラス数なども含め、総合的に評価する。

# 5　検討・考察

用例文クラスタリングの結果について、以下の組について、概念に変換する場合 (a) と変換しない場合 (b) の違いについて検討・考察する。

1. グラフクラスタリングに mcl を用いる場合 (c) と graclus を用いる場合 (d)

   (c) の場合、概念を考慮したほうが正解率が高くなり、無効クラスタ数が減少した。また、概念を考慮しないと、無効クラスタの数が多くなる。つまり、mcl では名詞が細かく分かれすぎてしまうということがわる。概念を考慮すると、名詞がある程度まとめられているので、細かく分かれすぎてしまうという事態を解消できている。
   (d) の場合、概念を考慮すると正解率が下がるということがわかった。

2. 用例文が複数のクラスタに重複して導かれることを許可する場合 (e) と禁止する場合 (f)

   (e) の場合に比べ、(f) の場合のほうが有効用例文数が減るため、正解率にも影響を及ぼす。mcl で概念を考慮しない場合は、正解数が有効用例文数よりも大幅に減るため、正解率が下がる。概念を考慮する場合は、有効用例文数が半分程度になるのに対し、正解数はそこまで減らない。つまり、正しく分類されていたにも関わらず、用例文数の多い語義の用例文が重複してこれらの結果を隠す形になっていたクラスタがあったと考えられる。

次に、(g)・(h) において、正解率・抽出された語義数について最も良い成果を得られた (a)〜(f) の組み合わせについて検討・考察する。ここで、(b,c,f) は (b) と (c) と (f) の方法の組み合わせを表す。

1. 正解率

   (g)：(b,c,f)
   (h)：(b,c,f)

   (g)、(h) ともに (b,c,f) が最も正解率が高かった。
2. 抽出された語義数

   (g)：(b,c,e)・(b,c,f)
   (e)、(f) に関わらず (b,c) ならば最も多くの語義が抽出された。
   (h)：(a,c,f)
   (h) では、(a,c,f) が最も多くの語義が抽出された。しかしこの方法では、同じ語義なのに別のクラスタで少ない用例文がまとまっていることが多かった。同じ語義ならば、一つのクラスタにまとまるほうが望ましい。

表 2　用例文クラスタリングの解析結果

| | | NDB(g) | | | | ADB(h) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | mcl(c) | | graclus(d) | | mcl(c) | | graclus(d) | |
| | 重複 | 許可 (e) | 禁止 (f) | 許可 | 禁止 | 許可 | 禁止 | 許可 | 禁止 |
| 概念に | 有効用例文数 | 39 | 37 | 176 | 45 | 142 | 121 | 378 | 121 |
| 変換 | クラスタ数 | 32 | 32 | 12 | 10 | 86 | 86 | 12 | 12 |
| しない | 無効クラスタ数 | 24 | 25 | 0 | 1 | 50 | 53 | 0 | 2 |
| (a) | 正解数 | 7 | 4 | 72 | 17 | 61 | 47 | 114 | 42 |
| | 正解率 | 0.1795 | 0.1081 | 0.4091 | 0.3778 | 0.4296 | 0.3884 | 0.3016 | 0.3471 |
| | 抽出された語義数 | 2 | 2 | 2 | 3 | 7 | 6 | 4 | 5 |
| 概念に | 有効用例文数 | 91 | 45 | 247 | 45 | 255 | 121 | 522 | 121 |
| 変換 | クラスタ数 | 11 | 11 | 12 | 9 | 28 | 28 | 12 | 11 |
| する | 無効クラスタ数 | 1 | 1 | 0 | 4 | 7 | 7 | 0 | 2 |
| (b) | 正解数 | 39 | 25 | 93 | 16 | 103 | 59 | 170 | 40 |
| | 正解率 | 0.4286 | 0.5556 | 0.3765 | 0.3556 | 0.4039 | 0.4876 | 0.3257 | 0.3306 |
| | 抽出された語義数 | 4 | 4 | 2 | 3 | 6 | 6 | 2 | 5 |

実験結果から、最も精度が高い組み合わせは「グラフ構造は概念を考慮して作成し、mcl でグラフクラスタリングして、用例文が複数のクラスタに重複することを禁止する (b,c,f)」であった。これは、他の組み合わせから総合的に判断して、正解率が高く、抽出された語義数が多く、クラスタ数に対する無効クラスタ数の割合が低い組み合わせを選択した結果によるものである。

## 6　まとめと今後の課題

本稿では、概念を考慮して作成したグラフ構造をクラスタリングし、その結果を用いて用例文を語義毎に分類した。また、この用例文のクラスタリング結果を分析し、用例文集合から語義別用例文集合への分類性能を調査した。その結果、概念を考慮した場合、マルコフクラスタリングを利用したほうが分類性能が良くなった。固有値を用いたクラスタリングでは、今回の実験では効果がみられなかった。従って、マルコフクラスタリングを利用してクラスタリングしたほうが、概念を考慮するメリットがあると考えられる。固有値を利用たクラスタリングによる方法は、違ったアプローチによって分類性能が良くなる可能性がある。

今後の課題として、分類性能を良くするために

係り受けも考慮する。

エッジの重みが一定以下の概念の組み合わせは、グラフ構造の作成に使用しない。

ノイズを取り除いてグラフクラスタリングを行う。

ということが考えられる。

## 参考文献


[1] Jean Véronis: ”HyperLex: lexical cartography for information retrieval”, Computer Speech & Language, Vol.18-3, pp. 223–252 (2004).

[2] 独立行政法人 国立国語研究所: ”分類語彙表－増補改訂版” (2004).

[3] 村田 昇: ”情報理論の基礎—情報と学習の直観的理解のために”, サイエンス社 (2005).

[4] Stijn van Dongen: ”Graph Clustering by Flow Simulation”, Ph.D thesis, University of Utrecht (2000).

[5] I. Dhillon, Y. Guan, and B, Kulis: ”Weighted Graph Cuts without Eigenvectors: A Multilevel Approach”, IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI), Vol.29-11, pp. 1944–1957 (2007)

[6] 独立行政法人 国立国語研究所: ”「現代日本語書き言葉均衡コーパス」モニター公開データ（2008 年度版）” (2008).