JAIST Repository
https://dspace.jaist.ac.jp/

Title	用例のクラスタリングに基づく単語の新語義の発見
Author(s)	田中，博貴
Citation	
Issue Date	2009-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8107
Rights	
Description	Supervisor:白井　清昭　准教授，情報科学研究科，修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

用例のクラスタリングに基づく単語の新語義の発見

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

田中 博貴

2009 年 3 月

修 士 論 文

用例のクラスタリングに基づく単語の新語義の発見

指導教官　白井 清昭 准教授

審査委員主査　白井 清昭 准教授
審査委員　島津 明 教授
審査委員　東条 敏 教授

北陸先端科学技術大学院大学
情報科学研究科情報処理学専攻

0710043 田中 博貴

提出年月: 2009 年 2 月

Copyright © 2009 by Tanaka Hiroki

目 次

第 1 章　はじめに **1**

1.1　研究の背景と目的 1

1.2　本論文の構成 2

第 2 章　関連研究 **4**

2.1　クラスタリング 4

2.1.1　K-means 法 4

2.1.2　半教師あり手法 5

2.2　語義識別 5

2.3　語義曖昧性解消 5

2.4　本研究との関連と本研究との相違 6

第 3 章　クラスタリング **7**

3.1　九岡の手法 7

3.1.1　ベクトルの作成 7

3.1.2　複数の特徴ベクトルの組み合わせ 9

3.1.3　クラスタリングアルゴリズム 10

3.2　改良手法 10

3.2.1　KKZ 法 10

3.2.2　トップダウン分割法 11

第 4 章　クラスタと辞書との対応付け **13**

4.1　用例クラスタの特徴ベクトル 13

4.2　語義の特徴ベクトル 14

4.3　クラスタと語義との対応付け 19

4.4　特徴ベクトルの作成方法の改良 19

4.4.1　対象単語の取り扱い 19

4.4.2 単語の出現頻度の取り扱い 20
4.4.3 単語のクラスタ頻度を考慮 20
4.5 語義の特徴ベクトルの過疎性への対応 21
4.5.1 特徴ベクトルの構築方法の改良 21
4.5.2 類似度計算方法の改良 23
4.5.3 特徴ベクトルの補正 24
4.6 小分類の語義の特徴ベクトルによる対応付け 25

第 5 章 新語義判定 **26**
5.1 既存語義近接度 26
5.1.1 既存語義との類似度の分散 26
5.1.2 既存語義との類似度の最大値と最小値の開き 27
5.1.3 既存語義との類似度の最大値 27
5.2 新語義の判定 27
5.2.1 既存語義近接度による新語義の判定 27
5.2.2 既存語義近接度の差による新語義の判定 28

第 6 章 評価 **30**
6.1 実験データ 30
6.2 評価実験 31
6.2.1 用例のクラスタリング 31
6.2.2 用例クラスタと辞書の語義との対応付け 36
6.2.3 新語義判定 51

第 7 章 おわりに **61**
7.1 まとめ 61
7.2 今後の課題 62

付 録 A 実験に用いた対象語とその語義 **66**

図 目 次

3.1 K-means での初期クラスタの様子 11
3.2 KKZ 法における初期クラスタの作成 11

4.1 単語の共起行列 14
4.2 岩波国語辞典における辞書 ID 15
4.3 「目」の語義の定義文とそれに対応する語義分類の様子 16
4.4 「モデル」の語釈文 16
4.5 語釈文タイプ分類アルゴリズム 17

5.1 新語義の判定 28

6.1 K-Var で閾値の値に対する F 値の変動 52
6.2 K-$Diff$ で閾値の値に対する F 値の変動 52
6.3 K-Max で閾値の値に対する F 値の変動 53
6.4 類似度の値と対応付け成果との相関 60

表 目 次

4.1 共起行列の詳細 18
4.2 語義の特徴ベクトルで値が 0 である素性の個数 22

6.1 対象単語の一覧 30
6.2 単独の特徴ベクトルを用いたクラスタリングの purity 33
6.3 単独の特徴ベクトルを用いたクラスタリングの語義識別率、新語義識別率 . 34
6.4 複数の特徴ベクトルを用いたクラスタリングの purity 34
6.5 複数の特徴ベクトルを用いたクラスタリングの語義識別率、新語義識別率 . 35
6.6 w_e の値に対する正解率の変動 37
6.7 4.4 節の手法の正解率 38
6.8 top_x の値に対する正解率の変動 39
6.9 top_I の値に対する正解率の変動 40
6.10 sm の値に対する正解率の変動 41
6.11 式 (4.13) の補正による語義の対応付けの正解率 ($w_e = 0.0$) 42
6.12 式 (4.13) の補正による語義の対応付け正解率 ($w_e = 1.0$) 43
6.13 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 0.0$) 44
6.14 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 1.0$) 45
6.15 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 0.5$) 46
6.16 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 1.5$) 47
6.17 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 2.0$) 47
6.18 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 2.5$) 48
6.19 小分類の語義の特徴ベクトルを用いての対応付けの結果 49
6.20 自動作成した用例クラスタと辞書の語義との対応付けの結果 50
6.21 手法 RDK1 を用いての新語義判定の結果 51
6.22 最適な閾値を設定したときの実験結果 54
6.23 自動作成クラスタで、$K\text{-}Var$ を用いての新語義判定の結果 54

6.24 自動作成クラスタで、$K\text{-}Diff$ を用いての新語義判定の結果 55
6.25 自動作成クラスタで、$K\text{-}Max$ を用いての新語義判定の結果 55
6.26 Cor_{total} による評価（正しい用例クラスタ） 56
6.27 Cor_{total} による評価（自動作成クラスタ） 57
6.28 1つの用例を1クラスタとした時の新語義判定の評価 58

第1章　はじめに

1.1　研究の背景と目的

単語の意味は時と共に変化し、新しい語義は日々生まれてきている。そこで本研究では、コーパスから新しい単語の意味を自動的に発見することを目的とする。もし新しい単語の意味を自動的に発見することができれば次のような利点がある。

1. 新語義をいち早く知ることができる

   単語の意味は日々変化するが、膨大な量のテキストを常に観察し、単語の意味の変化を観察することは不可能である。新語義を自動的に発見できる技術があれば、単語の意味の変化を効率的に把握できる。

2. 辞書編纂作業のサポート

   新語義をいち早く知ることにより、その時々に適した内容の辞書を構築できる

3. 語義曖昧性解消における辞書にない語義を判別できないという問題の解消

3について詳しく述べる。文中で使用されている単語の意味を判別する語義曖昧性解消（Word Sense Disambiguation;WSD） は、単語の意味は岩波国語辞典のような辞書によって予め定義されていることが一般的である。そして、曖昧な単語に対して、辞書に記載されている語義のいずれかを選択することで語義の曖昧性を解消する。しかし、既存の辞書に定義されている語義の中から適切なものを選択するという手法では、辞書に記載されていない新語義を判別する場合、必ず間違った語義を選択してしまう。新語義を自動的に発見することができれば、辞書の整備が容易になり、上記の問題を解決できる。

また、あらかじめ語義を定義せず、単語のインスタンス（用例）をクラスタリングすることで単語の意味を自動的に弁別する研究もある [1,2]。これらの手法は、語義の定義は辞書に寄らないため、単語が辞書に記載されていない意味で使われる例文も適切に取り扱うことが出来る。しかし、先行研究では、同じ意味を持つ単語の用例をまとめたクラスタ

を作成することはできるが、作成された用例クラスタがどのような意味を持つのかといった意味の解釈は行われていない。

本論文の目的は、クラスタリングによって得られた用例クラスタに対して意味の解釈を行うことにより、コーパスから新語義を自動的に発見する手法を提案する。用例クラスタの意味の解釈とは、クラスタ内における用例の単語の意味が既存の辞書の意味のどれに該当するか、あるいはどの意味にも該当しない新語義であるかを判定する処理を指す。

本論文における新語義の定義について述べる。本論文では、既存の辞書に記載されていない意味で使用されている語義のことを新語義と定義する。新語義の具体的な例を挙げる。岩波国語辞典における「反応」という単語の語釈文は以下の2つである。

1. 生体反応 「薬物反応」

2. 化学反応 「可逆反応」

上記2つの語釈文の意味に該当しない「反応」を本研究では新語義であると見なす。今、「反応」の例文として以下の2つがあるとする。

1. 他のパソコンには反応するのに私のには …

2. 何キロ以上でオービスが反応しますか？

この2つの例文中で使用されている「反応」は、どちらも「非生物による反応」を表わしており、上記で述べた辞書における「反応」の語釈文のどちらにも該当しない。よってこの2つの例文における「反応」は新語義である。

しかし、この「非生物による反応」は、辞書の1番目の語釈文「生体反応」の意味と同じであるとも考えることもできる。それは、パソコンやオービスを人間のように比喩することによって「反応」という単語を用いていると言えなくも無いからである。しかし、本研究では、本質的な意味は同じでも、本来の意味とは異なる文脈で使われている場合には、新語義であるとみなす。この例では、辞書の意味の「反応」の主語は生物であるが、例文における「反応」の主語は非生物であり、文脈が異なるため、新語義であるとみなす。また、本研究における新語義は、既存の辞書に記載されていない、単語の新しい用法のことであるとも言える。

1.2 本論文の構成

本論文の構成は次の通りである。

2章では、語義曖昧性解消、語義識別に関する先行研究と本研究との関連と違いについて

述べる。
3 章では、単語の用例をクラスタリングする手法について述べる。
4 章では、クラスタと辞書との対応付けの手法について述べる。
5 章では、対応付けの結果を基にした、新語義の判定の手法について述べる。
6 章では、手法に対する評価実験と、それによって得られた結果について述べる。
7 章では、本研究のまとめ、および今後の課題について述べる。

第2章　関連研究

本論文では、既存の辞書に定義されていない新語義の発見を目的としている。その大まかな手順は以下の通りである。コーパスから単語のインスタンス（用例）をいくつか収集し、クラスタリングの手法を用いて同じ意味を持つインスタンスをまとめてクラスタを作成する。これは一般に語義識別と呼ばれるタスクである。次に、用例のクラスタに対し、既存の辞書の語義の中から対応する語義を選択する。この処理は、語義曖昧性解消とほぼ同様の問題設定となっている。最後に、既存の語義の関連度が低いクラスタを新語義とみなす。

ここでは本研究に関連するクラスタリング、語義識別、語義曖昧性解消に関する過去の研究について述べ、最後に本研究との関連と相違について述べる。

2.1　クラスタリング

2.1.1　K-means 法

K-means 法とは、分割型クラスタリングの一種である。分割型クラスタリングとは、データを k 個のクラスタにランダムに割り当て、クラスタの質が高まるように各データのクラスタへの再割り当てを繰り返す手法である。データの再割り当ては、対象データを、クラスタの重心とその要素データの類似度が最大であるクラスタに割り当てることにより行う。具体的には以下の手順をとる。

1. 対象となるデータをランダムに k 個のクラスタに分割する
2. クラスタの重心を計算する
3. 各データが属するクラスタを、重心との類似度が最大であるクラスタに変更する。
4. クラスタの重心やデータの割り当てが収束するまで 2,3 を繰り返す

ただし、K-means 法では初期クラスタをランダムで作成しているため、初期化によって異なるクラスタリング結果が得られるという問題がある。

2.1.2 半教師あり手法

クラスタリングの対象となるデータのいくつかに、クラスタに関する情報を与えた上でクラスタリングを行うという手法もある。これを半教師ありクラスタリング手法という。Klein らや Kiri らは、クラスタリングの際に must-link や cannot-link と呼ばれる制約を予めクラスタリングの対象となるデータに与えた上でクラスタリングを行い、クラスタリングの質を向上させている [3][4]。must-link とは、これを持つデータ同士は同じクラスタに属すという制約を意味し、cannot-link とは、これを持つデータ同士は同じクラスタには属さないという制約を意味する。また、Klein らは制約をより有効に利用するため能動的な学習アルゴリズムも提案している。

一方、新納らはデータクラスタリングの際に同じような制約を用いているが、直接的にクラスタリングに用いるのではなく、ある程度制約を用いずにクラスタリングを行い、出来上がったクラスタリング結果を制約に応じて修正するという方法をとっている [5]。この方法によりクラスタリング手法に依らない制約の使用方法を提案している。

2.2 語義識別

語義識別とは、予め語義を定義せず単語のインスタンス（用例）をクラスタリングすることで単語の意味を自動的に弁別するという技術である。Schütze は英語を対象とし、コーパスに出現する単語のインスタンス毎に一つの特徴ベクトルを作成し、クラスタリングによって同じ意味を持つインスタンスのクラスタを獲得する手法を提案している [6]。特徴ベクトルの素性として、文脈に出現する単語の基本形を用いている。一方、九岡は日本語を対象とした語義識別の手法を提案している [7]。九岡は Schütze と同様にベクトルをクラスタリングしているが、隣接ベクトル・連想ベクトル・文脈ベクトル・トピックベクトルと呼ばれる 4 種類の特徴ベクトルと、それらを組み合わせる手法を提案している。なお、九岡の手法に関しては 3.1 節で詳しく述べる。

2.3 語義曖昧性解消

語義曖昧性解消（Word Sense Disambiguation:WSD）とは、文中の単語が、辞書に定義されている複数の意味のうち、どの意味で使用されているかを自動的に判別する技術のことであり、自然言語処理における重要な基礎技術のひとつである。近年、教師あり学習に基づく手法が盛んに行われ、成功を収めている。

通常の WSD の問題設定は、特定の文脈中に出現した単語の意味を、辞書などによってあらかじめ定義された語義の中から選択するというものである。しかし、単語が辞書に記載されていない意味で使用されている場合、必ず間違った語義を選択してしまうこととなる。それに対し、菊田らは、辞書に定義された語義に「未定義語義」を加えた語義集合の中から対象語の語義を選択することで、新語義を考慮した WSD の手法を提案している [8]。彼らは、「未定義語」というラベルのついた大量の語義タグ付きコーパスは存在しないことから、教師なし学習アルゴリズムである EM アルゴリズムを用いて Naive Bays モデルを学習し、語義の曖昧性を解消している。また、初期の確率を推定する際、語義タグ付きコーパスを用いることで学習の精度を向上させることを試みている。

2.4 本研究との関連と本研究との相違

本研究では、語義識別の手法を用いて単語の意味を弁別し、弁別した単語が既存の辞書のどの語義に該当するかの対応付けを行い、対応付けに用いた既存の語義との関連度が低い場合にはそれを新語義とみなす。単語の意味を弁別する点では九岡らの研究と同じだが、各クラスタが既存の辞書のどの意味に該当するかを選択している点で九岡らの研究と異なっている。

また、新語義を発見するためのアプローチとして以下の 2 つが挙げられる。

1. 教師なしクラスタリングをして作成されるクラスタが新語義であるか否かを判定

2. 個別の用例に対して新語義であるか否かを判定

2. は菊田らの研究で用いたアプローチであるが、本研究では 1. を用いる。これは、個々の用例に対して新語義であるかの判定を行うよりも、似た意味を持つ用例をまとめてクラスタを作成し、それに対して新語義か否かの判定を行う方が、周辺に現れる単語など判定に用いることのできる情報が多いため、新語義かどうかをより正確に判定できるという予想に基づいている。また、WSD を行うモデルを学習する際、完全に教師無しのデータを用いている点で菊田らの研究と異なっている。

第3章　クラスタリング

2.4節でも述べたが、新語義を発見をするためのアプローチとしては、以下の2つが考えられる。

1. 教師なしクラスタリングをしてクラスタが新語義であるか否かを判定し、新語義を発見する
2. 個別の用例に対し新語義であるか否かを判定し、新語義を発見する

これら2つのうち本研究では1. の方法を採用する。本研究では、基本的には九岡の手法を用いて用例のクラスタリングを行う。3.1節では九岡の手法を簡単に紹介する。3.2節では、九岡によるクラスタリング手法を改良する試みについて述べる。

3.1　九岡の手法

九岡は複数の特徴ベクトルを用いて単語のインスタンス（用例）をクラスタリングしている。3.1.1項では特徴ベクトルの作成方法、3.1.2項では複数の特徴ベクトルの組み合わせてクラスタリングを行う手法、3.1.3項ではクラスタリングアルゴリズムについてそれぞれ説明する。

3.1.1　ベクトルの作成

九岡は単語の用例を以下の4種類の特徴ベクトルを用いて表現している。

1. 隣接ベクトル

   対象単語 w の直前または直後に現れる単語で特徴付けるベクトルのこと。w の前後1語の単語の出現形ならび品詞をベクトルの要素とする。ここでの品詞は、茶筌の品詞体系を用いている。

2. LDA 拡張文脈ベクトル

   対象単語 w の周辺に現れる単語で特徴付けるベクトルのこと。前処理として以下を行う。

   (a) コーパスから、単語 c_k を行、文書 d_l を列とする共起行列 A_c を作成する。ここで A_c の要素 $a_{i,j}$ は、単語 c_k が文書 d_l に出現した回数とする。

   (b) 作成した行列 A_c に対して LDA（Latent Dirichlet Allocation）を適用する。LDA により、トピックと単語の関連性を表わすパラメタを学習する。

   (c) 各トピック z_m に対し、そのトピックと最も関連性の高い 300 個の単語の集合 Z_m を作成する。

   以上の前処理を経て、インスタンス w_i の LDA 文脈ベクトル $\vec{c_i}$ を以下のように作成する。

   (a) w_i の周辺に自立語 c_{ij} が出現した場合、$\vec{c_i}$ 中の c_{ij} の重みを 1 にする。

   (b) 上記の（a）が起きた場合で、c_{ij} が Z_m に含まれている場合、Z_m 中の残りの単語について、$\vec{c_i}$ 中のその単語の重みを 0.5 とする。

3. 連想ベクトル

   対象単語 w の周辺に現れる単語で特徴付けるベクトルのこと。事前にコーパスから単語の共起行列 A_a を作成する。この時、行列 A_a の要素 $a_{i,j}$ は、単語 c_i と単語 c_j が同じ文書に共起した回数である。連想ベクトル $\vec{a_i}$ は、w_i の周辺に現れる自立語 c_j に対する共起ベクトル $\vec{o}(c_j)$ の和とする。

4. トピックベクトル

   PLSI によって推定されるトピックによって、対象単語 w を特徴付けるベクトルのこと。前処理として以下を行う。

   (a) 文脈ベクトルと同様に、単語を行、文書を列とする共起行列 A_c を作成する。

   (b) 共起行列 A_c に対して、PLSI を適用し、トピックと単語の関連性を表わす確率パラメタを学習する。

   (c) インスタンス w_i を含む文書 d_i を、PLSI の学習データに含まれない未知の文書とみなし、EM アルゴリズムを用いて確率パラメタ $P(z_m|d_i)$ を推定する。

以上の前処理を経て、w_i に対するトピックベクトル $\vec{t_i}$ を式 (3.1) と定義する。

$$\vec{t_i} = (P(z_1|d_i) \cdots P(z_M|d_i))^T \tag{3.1}$$

ここでの M とは、PLSI の隠れ変数の数を表わしている。九岡は M=50 としている。

3.1.2 複数の特徴ベクトルの組み合わせ

九岡らは、複数の特徴ベクトルによるクラスタリングの結果を比較して、最も良い結果が得られると思われる特徴ベクトルを単語毎に選択する手法を提案している。具体的には、特徴ベクトルごとのクラスタリングの結果を C_n とした時、それぞれに対し評価関数 $eval(C_n)$ を算出し、その値が最大となる特徴ベクトルを選択する。評価関数の例を以下に示す。

- 相対的クラスタ内類似度

$$rel_intra(C) = \sum_{\pi_j \in C} \frac{1}{N_j} \sum_{\vec{v_i} \in \pi_j} \frac{sim(\vec{g_j}, \vec{v_i})}{\max_{\vec{v_i}} sim(\vec{g_j}, \vec{v_i})} \tag{3.2}$$

ここでの C はクラスタリングの結果を、π_j は j 番目のクラスタを、$\vec{g_j}$ は π_j の重心ベクトルを、$\vec{v_i}$ は π_j の要素である特徴ベクトルを、N_j はクラスタ π_j の要素数をそれぞれ表わしている。rel_intra は、クラスタ内の要素が近ければ近いほど高い値をとる評価値である。

- 相対的クラスタ凝集度

まず、rel_inter を以下のように定義する。

$$rel_inter(C) = \sum_{\pi_j \in C} \frac{sim(\vec{G}, \vec{g_j})}{\max_{\vec{g_j}} sim(\vec{G}, \vec{g_j})} \tag{3.3}$$

ここでの π_j は j 番目のクラスタを、$\vec{g_j}$ は j 番目のクラスタの重心ベクトルを、$\vec{G}$ は $\vec{g_j}$ の各 π_j についての平均をそれぞれ表わしている。rel_inter はクラスタの重心が互いに近ければ近いほど高い値をとる。次に相対的クラスタ凝集度 rel_coh を以下のように定義する。

$$rel_coh(C) = \frac{rel_intra(C)}{rel_inter(C)} \tag{3.4}$$

rel_coh は rel_intra が高ければ高いほど、また rel_inter が低ければ低いほど、高い評価値をとる。

3.1.3 クラスタリングアルゴリズム

九岡はクラスタリングアルゴリズムにK-means法を用いている。K-means法は、対象データをクラスタの重心とそのデータ間の類似度が最大であるクラスタに割り当てるというアルゴリズムである。九岡はベクトル間の類似度関数をsimとして、式(3.5)のようなコサイン類似度を用いている。

$$sim(\vec{v_i}, \vec{g_j}) = \frac{\vec{v_i} \cdot \vec{g_j}}{|\vec{v_i}| \cdot |\vec{g_j}|} \tag{3.5}$$

ここで、$\vec{v_i}$は対象語w_iの特徴ベクトルを、$\vec{g_j}$はクラスタjの重心ベクトルをそれぞれ表わしている。

また九岡は、K-means法における初期クラスタをランダムに作成しているが、その際各クラスタに含まれる要素の数をほぼ同じにするという工夫を施している。

3.2 改良手法

本節では、九岡の用例クラスタリング手法を改良する2つの手法について述べる。なお、ここであげる手法は既存の手法をそのまま適用し、新しい手法を提案しているわけではない。

3.2.1 KKZ法

九岡が用いたK-means法は、初期クラスタの決め方によって最終結果が異なる。そのため初期クラスタによってクラスタの品質が異なるという問題点が存在する。例えば、図**??**で、左側の図のように初期クラスタが互いに固まっている場合と、右側の図のように初期クラスタがばらけている場合とでは、右側の方が最終的に作成されるクラスタの質は良いと思われる。

本研究では、K-means法の初期クラスタの作成方法としてKKZ法を採用した[9]。KKZ法とは、お互いになるべく遠いK個の事例を選出し、それらを初期クラスタの核にするというものである。 具体的には以下の手順をとる。

1. 一番始めの初期点として、多次元空間の原点から最も遠い要素を選択する。

2. 選択した要素（以下Aとする）から最も遠い要素（以下Bとする）を選択する。

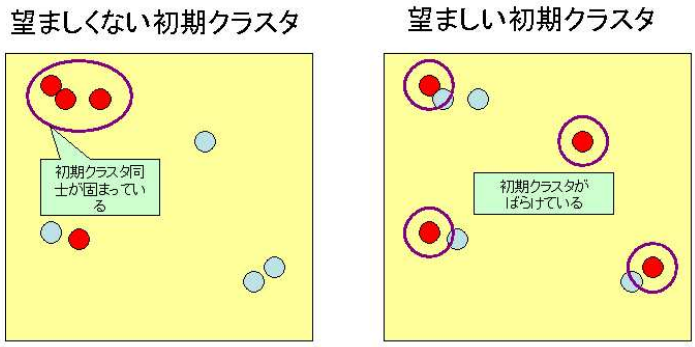

図 3.1: K-means での初期クラスタの様子

3. 全ての選択された要素（現在は A と B）から最も遠い点を選択する。

4. 3 の処理をクラスタの個数を満たすまで繰り返す。

上記の手続きを図解したのが図 3.2 である。

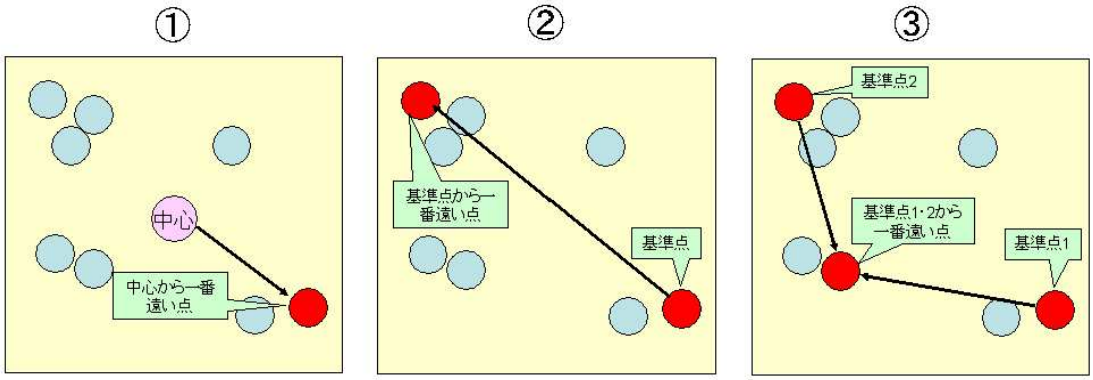

図 3.2: KKZ 法における初期クラスタの作成

これによって多くの種類の要素をクラスタの核に取り込むことが可能であり、クラスタリング精度が向上する。また、初期クラスタが一意に決定するため、クラスタリング結果が初期クラスタに依存するという問題も生じない。

3.2.2 トップダウン分割法

本研究では K-means 法以外のクラスタリングアルゴリズムを用いることも試行した。具体的には、トップダウン分割法と呼ばれるアルゴリズムを採用した。トップダウン分割法とは、始めに全てのインスタンスを一つのクラスタにまとめ、定められた個数のクラス

タが得られるまでトップダウン式にクラスタの分割を繰り返す方法である。クラスタの分割は、以下の評価関数 I_2 が最大になるように行う。

$$I_2 = \sum_{i=1}^{K} \sqrt{\sum_{v,u \in S_i} sim(v,u)} \tag{3.6}$$

ここでの、K はクラスタの総数を、S_i は i 番目のクラスタ内にある要素の集合を、$sim(v,u)$ は要素 v と要素 u の類似度をそれぞれ表わしている。なお、ここでは $sim(v,u)$ としてコサイン類似度を用いた。トップダウン分割法の実装には、汎用クラスタリングツール CLUTO を用いた。

第4章　クラスタと辞書との対応付け

本章では、クラスタリングによって生成された用例クラスタが辞書におけるどの意味に対応するかを決定する手法について述べる。また本研究では、辞書の意味に対応付けすることができないクラスタを新語義とみなす。ただし、本章では、新語義を発見するための前処理として、用例クラスタは辞書におけるいずれかの語義と対応すると仮定し、クラスタと辞書の語義とを対応づけることを目的とする。

具体的には、対象単語を w としたとき、コーパスから抽出された複数の w の用例をクラスタリングし、作成された用例クラスタ C_i から特徴ベクトル $\vec{c_i}$ を、辞書によって定義された w の語義 S_j から特徴ベクトル $\vec{s_j}$ を作成し、ベクトル間の類似度が最大となる語義 S_j を選択する。

4.1　用例クラスタの特徴ベクトル

用例クラスタの特徴ベクトル $\vec{c_i}$ は、クラスタ C_i に含まれる用例において、対象語 w の周辺に出現する単語をベクトルの要素とする。ただし、クラスタリングによって作成された用例クラスタの中には用例数が少ないものもある。実際に用例のクラスタリングを試みたところ、1 個または 2 個の用例のみで 1 つのクラスタが形成されることもあった。そこで、ベクトルがスパースになるのを避けるため単語間の間接的な共起も考慮する。

まず、図 4.1 のような単語の共起行列 A を作成する。

A_f は、A の行に対応し、用例クラスタの特徴ベクトル $\vec{c_i}$ の素性となる単語から構成される単語集合である。ここでは、BCCWJ の Yahoo!知恵袋コーパスにおいて、出現頻度上位 10,000 の自立語の集合を A_f とした。なおここでの自立語とは茶筌を用いた形態素解析の結果で品詞が名詞・動詞・副詞・形容詞である単語を指す。ただし、式 (4.1) に定義する $df(t)$ が 0.01 以上の単語 t は一般的すぎるとみなして除外した。

$$df(t) = \frac{\text{単語 } t \text{ が出現する文書数}}{\text{コーパスにおける全文書数}} \tag{4.1}$$

一方、A の列に対応する単語集合 A_t は、用例クラスタにおいて対象単語の周辺に出現すると仮定される単語の集合である。基本的には A_t と A_f は同じくコーパスにおける出

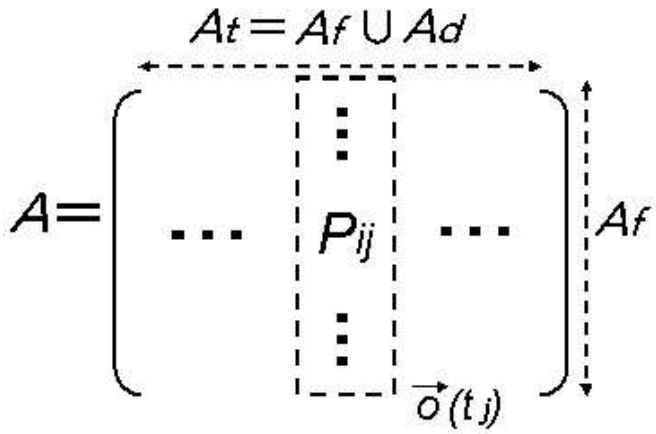

図 4.1: 単語の共起行列

現頻度の高い単語の集合とする。ただし、実際には A_t は A_f と A_d の和集合としている。この理由ならび A_d の定義については次項で述べる。

最後に行列の要素 P_{ij} は単語 t_i と t_j の文書内共起確率である。正確には、P_{ij} は j 列目の単語 t_j が出現する文書があったとき、同じ文書内に i 行目の単語 t_i が出現する確率 $P(t_i|t_j)$ である。P_{ij} は Yahoo!知恵袋コーパスから学習する。また、j 列目のベクトルを単語 t_j の共起ベクトル $\vec{o}(t_j)$ とする。$\vec{o}(t_j)$ は、単語 t_j が他の単語とどの程度共起しやすいかを表わすベクトルである。

クラスタの特徴ベクトル $\vec{c_i}$ を式 (4.2) と定義する。

$$\vec{c_i} = \frac{1}{N} \sum_{e_{ik} \in C_i} \sum_{t_l \in e_{ik}} \vec{o}(t_l) \tag{4.2}$$

e_{ik} は用例クラスタ C_i に含まれる用例を表わし、t_l は用例 e_{ik} の文脈に出現し、かつ A_t の要素である自立語を表わしている。また、N は $\vec{c_i}$ の大きさを 1 にするための正規化定数である。共起ベクトル $\vec{o}(t_l)$ の和をクラスタの特徴ベクトルとすることにより、用例の文脈に直接出現する単語 t_l ではなく、対象語 w と間接的に共起する単語の特徴が $\vec{c_i}$ に反映される。

4.2 語義の特徴ベクトル

語義の特徴ベクトル $\vec{s_j}$ は辞書の語釈文から作成する。本研究では語義の定義に用いる辞書として岩波国語辞典 [10] を用いた。岩波国語辞典の語義には、語義の分類として大分類・中分類・小分類とがある。本研究では、対象語の語義の定義として中分類の語義を用

いる。岩波国語辞典では語義には、図 4.2 に示す語義 ID が付与されている。また、語義 ID では、大分類、中分類、小分類の語義が区別されている。

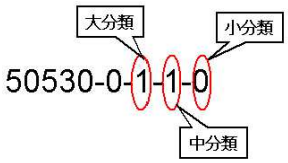

図 4.2: 岩波国語辞典における辞書 ID

岩波国語辞典における「目」の語義の階層構造を図 4.3 に示す。例えば、中分類の語義「50530-0-1-1-0」の下位の語義として小分類の語義は「50530-0-1-1-0」、「50530-0-1-1-1」、「50530-0-1-1-2」、「50530-0-1-1-3」の 4 つがある。同様に大分類の語義「50530-0-1-0-0」の下位には「50530-0-1-1-0」、「50530-0-1-1-1」、「50530-0-1-1-2」、「50530-0-1-1-3」、「50530-0-1-2-0」、「50530-0-1-3-0」、「50530-0-1-4-0」、「50530-0-1-4-1」、「50530-0-1-4-2」、「50530-0-1-4-3」、「50530-0-1-4-4」、「50530-0-1-5-0」、「50530-0-1-6-0」の 13 個の語義が該当する。

しかし、岩波国語辞典の語釈文は、その全てが単語の意味を説明した定義文ではない。そこで、本研究では岩波国語辞典の語釈文を以下の 4 つのタイプに分類した。

1. 定義文：単語の意味を説明した文
2. 例文：その語義の典型的な用例
3. 参照見出し：別の見出し語または語義への参照
4. その他：上記三つに当てはまらない文で、見出しの英語表記や注釈などが該当

図 4.4 は岩波国語辞典における「モデル」の 4 つの語義の語釈文と、そのタイプ分けの例である。これら 4 つの分類のうち、「参照見出し」と「その他」は単語の意味を表わしていると言い難いと判断したため、辞書の語義の特徴ベクトルの作成には用いなかった。

岩波国語辞典における語釈文を以上の 4 つのタイプに分類するプログラムを作成した。分類に使用したパタンマッチによる自動タイプ分けのプログラムのアルゴリズムを図4.5 に示す。

辞書ID	語釈文	小分類	中分類	大分類
50530-0-1-1-0	生物の、物を見る働きをする器官。また、その様子・働き。▽「眼」とも書く。	s1	s1	s1
50530-0-1-1-1	眼球・視神経から成る器官。「―を皿のようにして見る」「鵜の―鷹の―」(熱心に捜し求める目つき・態度)「―を開く」(自覚したり、知識を得たりする意にも)「―をつぶる」(見て見ないふりをする、また、死ぬ意にも)「―がつぶれる」(視力を失う)「―がさえる」(眠るべき時に、眠くならない。また、物事の本質を見抜く力を発揮する。↓(ウ))「―がすわる」(怒りや酔いで、じっと一点を見詰め、目の玉を動かさない)「―が回る」(めまいがする。転じて、非常に忙しい)「―から鼻へ抜ける」(抜け目がなくすばやい、または知恵が速く働くことの形容)「―と鼻の間」(非常に近いことのたとえ)「―と鼻の先」(同上)「―が真っ暗になる」(突然の不幸などで、希望を失ったり手立てが見えず、どうしたらよいか分からない状態に陥る)「―に見えて」(見てわかるほどはっきり)「―に物を見せる」(思い知るようにしかける)「―を光らす」(よく注意する)「―を凝らす」「―を丸くする」(びっくりする)「―を剥く」(怒ったり驚いたりして目を大きく開く)「―を喜ばせる」(美しい物などを見て喜び楽しむ)「―の保養」(同上)「―の正月」(同上)「―を細める」(喜ぶ意にも)「酒に―が無い(=すっかり心を奪われる)」	s2		
50530-0-1-1-2	目(1)(ア)の様子。目つき。「―は口ほどに物を言う」「はたから変な―で見られる」「色っぽい―をする」	s3		
50530-0-1-1-3	見ること。見えること。また、視力。更に、注意(力)。洞察力。「―に触れる」「お―に掛ける」(お見せする)「お―に掛かる」(お会いする)「―を掛ける」(転じて、面倒を見て引き立ててやる)「―がくらむ」(めまいがする。また、心が奪われて正しく判断できなくなる)「―をさます」(迷いから脱却する意にも)「―に障る」「―のかたき」(目ざわりで憎くて仕方ないもの)「―に立つ」「―につく」「―にする」「―もあてられない」(あまりひどくて見ていられない)「―もくれない」(無視する)「長い―で見る」(現状だけで判断せず、将来を期待して気長に見る)「―を奪う」(すばらしさなどで見とれさせる)「―を引く」「―をつける」「―をとめる」「―を配る」「―が届く」(注意・監視がそこまで及ぶ)「―を離す」「―を盗む」(人に気づかれないように何かをする)「―をくらます」(ごまかして気づかれないようにする)「―を疑う」(意外な事に接し、これが事実なのか、見そこないではないかと驚く)「―を通す」(一通り見る。ざっと読む)「―が近い」(近眼)「―がいい」「―が高い」(ものを識別する力がある)「―が肥えている」(同上)「―を利かす」	s4		
50530-0-1-2-0	目(1)(ア)に見える姿・様子。「―ばかりよくて実質は悪い」	s5	s2	
50530-0-1-3-0	ある物事に出会うこと。経験。体験。また、局面。「ひどい―を見る」「落第の―」	s6	s3	
50530-0-1-4-0	形が目(1)(ア)に似ているもの。「台風の―」「―」	s7	s4	
50530-0-1-4-1	縦横に(線状に)並び交わったものによって囲まれてできた、すき間。「―が粗い布地」「網の―」「碁盤の―」「―が詰んでいる」	s8		
50530-0-1-4-2	縦に並んだものの、すき間。「のこぎりの―」「櫛の―」	s9		
50530-0-1-4-3	さいころの面につけた点状のしるし。「六の―」「―が出る」(思いどおりの目が現れる。転じて、運がついて事がうまく運ぶ)	s10		
50530-0-1-4-4	量を示すしるしの刻み。目盛り。「秤の―」	s11		
50530-0-1-5-0	秤で量った重さ。「―がかかる」(相当の重みがある)「―減り」。また、「匁」の別称。「一貫二〇〇―」	s12	s5	
50530-0-1-6-0	木材の表面にあらわれた模様。木目。「―が細かい板」「柾―」	s13	s6	
50530-0-2-1-0	順序を表す時に添える語。「三番―の問題」「二つ―の角を曲がる」「十軒―」	s14	s7	s2
50530-0-2-2-0	点または線のようになって、他と区別される状態になった所。「結び―」「割れ―」「境―」。物事の境となるような情況。「季節の変わり―」「彼の勢力も落ち―になる」「勝ち―がない」(勝つ見込みがない)	s15	s8	
50530-0-2-3-0	《量・程度に関係する形容詞の語幹などに付いて》どちらかと言えば、その性質を持つ意を表す。「話を―に打ち切る」「大きー な方を選ぶ」「細―の糸」「幾分早―に出掛ける」「控えー な人」	s16	s9	

図 4.3: 「目」の語義の定義文とそれに対応する語義分類の様子

S1 ① 型。模型。③▽↓もけい
S2 ① 手本。模範。②「これを―にしてやれば間違いない」
S3 ① 美術製作の対象となるもの・人。文学作品の人物の素材となる人。
S4 ①「ファッション　モデル」の略。流行の服装をして、客にみせたり写真に撮られたりするのが職業の(女の)人。④▽model

図 4.4: 「モデル」の語釈文

語釈文を左から1つの形態素ごとに順に調べる。 現在参照している文字が「・↓・⇔・▽ の四つの場合それぞれのcaseに移行し、そうでなければ調べた文字は定義文とする。	
case１ "「"	"「" の次の文字から"」"が出現するまで、配列examに格納しておく。 "」"が出現した時、以下の条件を調べ該当した場合、配列examを例文として扱い、該当しない場合は、配列examを定義文として扱う。
	条件1 ：見出し語を表わすタグが配列examに存在する。 例えば、単語「目」のsense id="50530-0-1-1-1" の語釈文での「鵜の<EX>目</EX>鷹の <EX>目</EX>」の部分が該当する。<EX>は見出し語を表わすタグである。
	条件2 ："・"が配列exam中に存在する。 例えば、単語「亜」のsense id="1-0-0-2-1"の語釈文 「亜細亜」の略。「亜欧・東亜」 の場合、「亜欧・東亜」の部分がこれに該当する。
	条件3 ：「」の後が文末である。 例えば、単語「亜」のsense id="1-0-0-2-1"の語釈文 「亜細亜」の略。「亜欧・東亜」 の場合、「亜欧・東亜」の部分は文章の末尾にあるため例文となり、「亜細亜」の部分は定義文となる
	条件4 ：「」が連続して用いられている。 例えば、単語「足」のsense id="615-0-0-1-2"の語釈文 足首から下。「—の裏」「—が地につく」（うわつかず着実なことのたとえにもいう）「—を洗う」（↓あらう(1)） の場合、「—の裏」「—が地につく」の部分がこれに該当する。 この場合、「—の裏」と「—が地につく」は例文と見なされる。
case２ "↓"	"↓"の次の文字から文末、もしくは"。"までを参照見出しとする。 例えば、単語「アウトプット」のsense id="171-0-0-0-0"の語釈文 ↓しゅつりょく。⇔インプット。▽output の場合、"しゅつりょく。"の部分がこれに該当する。この場合、"しゅつりょく。"は参照見出しと見なされる。
case ３ "⇔"と"▽"	"⇔"と"▽"の次の文字から文末、もしくは"。"までをその他とする。 例えば、単語「アウトプット」のsense id="171-0-0-0-0"の語釈文 ↓しゅつりょく。⇔インプット。▽output の場合、"インプット。"と"output"の部分がこれに該当する。この場合"インプット。"と "output"はその他と見なされる。

図 4.5: 語釈文タイプ分類アルゴリズム

次に、「定義文」に分類された語釈文のみを用いて語義 S_j の特徴ベクトル $\vec{s_j}$ を式（4.3）のように作成する。

$$\vec{s_j} = \frac{1}{N} \sum_{t_k \in d_j} \vec{o}(t_k) \tag{4.3}$$

d_j は語義 S_j の語釈文における定義文を、t_k は d_j に出現し A_t の要素である自立語を表わす。また N は正規化定数である。ただし、語義 S_j は岩波国語辞典における中分類の語義である。$\vec{s_j}$ は、用例クラスタの特徴ベクトル $\vec{c_i}$ と同様に、共起ベクトルの和を正規化することによって得られる。

図 4.1 の共起行列における単語集合 A_d は、辞書全体における語釈文のうち「定義文」に出現する自立語の集合である。ここでの自立語とは、岩波国語辞典に予めタグ付けされている形態素解析結果で、品詞が名詞・動詞・副詞・形容詞である単語を指す。共起行列の作成については 4.1 節で述べたが、A_t を A_f と A_d の和集合にした目的は、コーパスにおける出現頻度が低いため A_f には含まれないが辞書定義文には出現する自立語に対して、共起ベクトルを得るためである。なお、A_t の要素のうち、コーパスから共起行列を作成した際、他の単語と共起せず、共起ベクトルが 0 ベクトルになった単語は除外してある。

A_t、A_f、A_d の定義と単語数を表 4.1 にまとめる。

表 4.1: 共起行列の詳細

記号	構成	要素数
A_f	Yahoo!知恵袋コーパスに出現する自立語	10,000 語
A_d	辞書の「定義文」の自立語	43,743 語
$A_f \cup A_d$	Yahoo!知恵袋の自立語＋辞書の「定義文」の自立語	46,694 語
A_t	$A_f \cup A_d$ の要素の中で、共起ベクトルが 0 ベクトルでない要素	24,086 語

辞書の語釈文のうち「定義文」だけでなく「例文」もまた単語の意味を識別する有力な手がかりになると考えられる。そこで、語義の特徴ベクトルを作成するもう一つの方法として、定義文と例文の両方を用いて $\vec{s_j}$ を式 (4.4) のように作成する手法を提案する。

$$\vec{s_j} = \frac{1}{N}\Big(\sum_{t_k \in d_j} \vec{o}(t_k) + \sum_{t_l \in e_j} w_e \cdot \vec{o}(t_l) \Big) \tag{4.4}$$

e_j は語義 S_j の語釈文中の例文を、t_l は e_j に出現する自立語を、N は正規化定数をそれぞれ表わしている。また、w_e は例文に出現する自立語の共起ベクトルに対して与えられる重みである。意味の説明文である「定義文」よりも、語義の使用例である「例文」の方が、コーパスから作成された用例クラスタの文脈に出現する単語と似ている単語が出現する

傾向が強いと予想される。そのため、例文に出現する自立語の共起ベクトルに高い重みを与えるようにした。なお、w_e の値は実験的に決定する。

4.3 クラスタと語義との対応付け

クラスタと辞書の語義との対応付けの手続きは以下の通りである。まず、作成された用例クラスタ C_i の特徴ベクトル $\vec{c_i}$ と、語義 S_j の特徴ベクトル $\vec{s_j}$ との類似度を測る。ここでは、ベクトル間の類似度はコサイン類似度とする。クラスタ C_i に対し、語義 S に含まれる全ての語義との類似度を求め、類似度が最大となる語義 S_j を選択する (式 (4.5))。

$$S_{selected}(C_i) = \arg\max_{S_j} sim(\vec{c_i}, \vec{s_j}) \tag{4.5}$$

4.4 特徴ベクトルの作成方法の改良

ここでは、特徴ベクトルの作成方法の改良案について述べる。具体的な案は以下の 3 つである。

1. 特徴ベクトルを作成する際に、対象単語に該当する共起ベクトルを考慮しない（4.4.1 項）
2. 単語の出現頻度を無視して特徴ベクトルを作成する（4.4.2 項）
3. 用例における単語のクラスタ頻度を考慮する（4.4.3 項）

以降それぞれに関して詳しく述べる。

4.4.1 対象単語の取り扱い

対象単語とは、新語義を発見する対象となる単語を指す。用例クラスタは対象単語の使用例を集めたものであるので、用例の中に少なくとも 1 つの対象単語を含む。4.2 節では、共起ベクトルの和に対象単語を用いていたが、ここでは対象単語を共起ベクトルの和に用いないことを試みた。なぜなら、対象単語は全ての用例クラスタに含まれている単語であるため、用例クラスタの個別の意味を示しているものではないと考えたためである。

4.4.2 単語の出現頻度の取り扱い

用例の特徴ベクトルを作成する際、式 (4.2) のように対象語 w の周辺に出現する単語に該当する共起ベクトルを足す。例えば、用例の文脈に出現する自立語 t_l が、「w_1 w_2 w_1 w_3 w_1」の場合、出現する 5 つの単語の共起ベクトルを足し合わせて特徴ベクトルを作成する。

ここで w_1 という単語に着目する。w_1 は文章中で複数回出現している単語である。共起ベクトルの足し算を行った際、w_1 は出現回数分（上記の例では三回）足されることとなる。これに対し、共起ベクトルの和を計算する際、単語の出現頻度を考慮しない方法も試みる。具体的には、複数回出現する単語に対し、出現回数分だけベクトルを足すのではなく一回だけ足すことにする。言い換えれば、式 (4.2) において、用例クラスタの中に出現する単語の異なりの集合を e_{ik} とする。

4.4.3 単語のクラスタ頻度を考慮

ある対象単語に対して複数の用例クラスタが得られた時、ある単語が 1 つのクラスタのみに含まれていれば、その単語はクラスタの特徴を表わしているといえる。一方、複数のクラスタに出現する単語は、個々のクラスタの意味を特徴づける単語とはいえない。

ここでは、単語が出現するクラスタの数をクラスタ頻度とし、これを基に対象語の周辺に出現する単語に重みを付けることを考える。具体的には、用例の特徴ベクトルを作成する式 (4.2) を式 (4.6) のように変更する。

$$\vec{c_i}' = \frac{1}{N} \sum_{e_{ik} \in C_i} \sum_{t_l \in e_{ik}} \log\{\frac{cl_n + 1}{cl_n(t_l)}\}\vec{o}(t_l) \tag{4.6}$$

ここで cl_n はクラスタの総数を、$cl_n(t_l)$ は単語 t_l が出現するクラスタの数をそれぞれ表わしている。これにより、クラスタ頻度が高い単語に対しては低い重みを、クラスタ頻度が低い単語に対しては高い重みを加えることになる。また、式 (4.6) 中の $\log\{\frac{cl_n+1}{cl_n(t_l)}\}$ で分子 cl_n に 1 を足している理由は、クラスタの個数 cl_n とクラスタ出現頻度 $cl_n(t_l)$ が同じ値の時、$\log\{\frac{cl_n+1}{cl_n(t_l)}\} = 0$ となり、t_l に対する共起ベクトル $\vec{o}(t_l)$ が足されなくなるのを避けるためである。

用例クラスタと同様のことが辞書の語義にも行える。ここでは、語義の特徴ベクトルの計算式 (4.4) を式 (4.7) のように変更する。

$$\vec{s_j}' = \frac{1}{N}\Big(\sum_{t_k \in d_j} \log\{\frac{sc_n + 1}{sc_n(t_k)}\}\vec{o}(t_k) + \sum_{t_l \in e_j} w_e \log\{\frac{sc_n + 1}{sc_n(t_l)}\}\vec{o}(t_l)\Big) \tag{4.7}$$

ここで sc_n は語義の総数を、$sc_n(t_l)$ は単語 t_l が出現する語義の数をそれぞれ示している。

4.5　語義の特徴ベクトルの過疎性への対応

一般に、辞書における定義文や例文の長さは短いため、作成された語義の特徴ベクトル $\vec{s_j}$ は過疎（スパース）になりやすい。ここで、特徴ベクトルがスパースであるとは、多くの素性（次元）に対する特徴ベクトルの値が 0 となる状態を指す。例えば、図 4.4 の語義 S_1 においては、定義文に出現する自立語は「型」と「模型」の二つしかなく、ベクトル $\vec{s_j}$ におけるほとんどの次元の値が 0 になる。表 4.2 に式 (4.4) で特徴ベクトルを作成した時の、値が 0 となる特徴ベクトルの素性の数を示す。なお、表 4.2 における単語は 6 章の評価実験の際に用いた対象単語である。表 4.2 から、同じ対象語の語義でも、値が 0 となる特徴ベクトルの素性の数に大きな差が生じていることがわかる。

用例クラスタと辞書の語義との対応付けを試みた結果、定義文が比較的長くスパースではない特徴ベクトルを持つ語義の方が、定義文が短くスパースな特徴ベクトルを持つ語義と比べて、一貫して用例クラスタの特徴ベクトルとの類似度が高くなる傾向が強いことがわかった。その結果、用例クラスタは常に同じ語義に対応付けられてしまう。例えば「地方」という単語では、全ての用例クラスタに対して「32849-0-0-3-0」の語義が選択されていた。実際に表 4.2 をみると、「32849-0-0-3-0」の語義は、他の語義と比べて値が 0 となる素性の数が少なく、特徴ベクトルがスパースではないことがわかる。この問題を解決するためには、語義の特徴ベクトルを構築する際に、値が 0 となる素性の数が語義によって大きく異なることがないようにする必要がある。

そこで本節では、上記で述べた問題の対策について述べる。ここでは以下の 3 つの手法を提案する。

1. 辞書定義文の特徴ベクトルの構築方法を改良する
2. クラスタと辞書定義文との類似度の計算方法を改良する
3. 特徴ベクトルに対する補正を行う

それぞれ、以降で詳しく述べる。

4.5.1　特徴ベクトルの構築方法の改良

本項では、辞書語釈文の特徴ベクトルの構築方法を改良する。語義の特徴ベクトルを構築する際、共起ベクトルを足す単語の個数に差があることによって、0 となる素性の数が語義によって大きく異なるといった問題がおきていると考えられる。そのため共起ベクト

表 4.2: 語義の特徴ベクトルで値が 0 である素性の個数

対象語	語義 ID	0 の個数
モデル	51379-0-0-1-0	8602
	51379-0-0-2-0	7865
	51379-0-0-3-0	129
	51379-0-0-4-0	110
ネタ	39686-0-0-1-0	8152
	39686-0-0-2-0	9601
	39686-0-0-3-0	6127
	39686-0-0-4-0	5201
カバー	8794-0-0-1-0	2101
	8794-0-0-2-0	3115
ウイルス	3327-0-0-1-0	1196
	3327-0-0-2-0	5284
ソース	29756-0-0-1-0	3864
	29756-0-0-2-0	9423
肉	38855-0-0-1-0	1200
	38855-0-0-2-0	1981
	38855-0-0-3-0	945
	38855-0-0-4-0	7383
	38855-0-0-5-0	8894
	38855-0-0-6-0	6243
サービス	18566-0-0-1-0	5167
	18566-0-0-2-0	4738
	18566-0-0-3-0	4130
地方	32849-0-0-1-0	4385
	32849-0-0-2-0	4251
	32849-0-0-3-0	520
アルバム	1399-0-0-1-0	4066
	1399-0-0-1-0	3976
コード	16950-0-0-1-0	4864
	16950-0-0-2-0	6297
	16950-0-0-2-0	8928
自分	22038-0-0-1-0	383
	22038-0-0-1-0	311
場合	40333-0-0-1-0	61
	40333-0-0-1-0	6611

対象語	語義 ID	0 の個数
時間	20676-0-0-1-0	196
	20676-0-0-2-0	1167
	20676-0-0-3-0	391
	20676-0-0-4-0	390
意味	2843-0-0-1-0	135
	2843-0-0-2-0	1626
	2843-0-0-3-0	1089
電話	35881-0-0-1-0	1309
	35881-0-0-2-0	3440
一緒	2485-0-0-1-0	267
	2485-0-0-2-0	1907
目	50530-0-1-1-0	0
	50530-0-1-2-0	458
	50530-0-1-3-0	102
	50530-0-1-4-0	42
	50530-0-1-5-0	1117
	50530-0-1-6-0	2337
	50530-0-2-1-0	1211
	50530-0-2-2-0	862
	50530-0-2-3-0	3
以前	2132-0-0-1-0	546
	2132-0-0-2-0	217
代	30531-0-0-1-0	2371
	30531-0-0-2-0	1351
	30531-0-0-3-0	5901
	30531-0-0-4-0	313
	30531-0-0-4-0	2427
顔	7272-0-0-1-0	4
	7272-0-0-2-0	361
	7272-0-0-3-0	2
系	14288-0-0-1-0	1753
	14288-0-0-2-0	2335
	14288-0-0-3-0	9410
	14288-0-0-4-0	8341
	14288-0-0-5-0	2527
郵便	52590-0-0-1-0	2833
	52590-0-0-2-0	5298
反応	42525-0-0-1-0	1445
	42525-0-0-2-0	3695

ルを足す単語の個数を均一にすることによってこの問題は回避でき、正解率の向上も期待できる。

具体的な手法は以下の通りである。

1. 式 (4.4) で特徴ベクトルを作成する。
2. 各特徴ベクトルで、ベクトルの要素を値の降順にソートする。以降、ベクトルの要素は値の大きい順に並んでいるものとする。
3. 降順に並んでいるベクトルの要素の上位 top_x 語の単語リスト A_{top} を作成する。top_x の値は実験的に決定する。
4. 単語リスト A_{top} を用いて特徴ベクトルを作成する（式 (4.8)）。

$$\vec{s_j}' = \frac{1}{N} \sum_{t_k \in A_{top}} \vec{o}(t_k) \tag{4.8}$$

4.5.2 類似度計算方法の改良

本項では、クラスタと語義の特徴ベクトルの類似度の計算方法に改良を加える。これまで述べた手法では、類似度はコサイン類似度を用いている (式 (4.9))。

$$sim(\vec{x}, \vec{y}) = \frac{\sum_i x_i y_i}{|\vec{x}| \cdot |\vec{y}|} \tag{4.9}$$

しかし、0 の要素が多いベクトルは、式 (4.9) の分子の内積の値が小さくなり、ベクトル間の類似度も小さくなる傾向がある。

そこで、式 (4.9) の分子で和をとる $x_i y_i$ の個数を一定にすることを考える。具体的には式 (4.9) の $x_i y_i$ の値を降順にソートし、その上位から top_I 個の $x_i y_i$ の値の総和を類似度の値として用いる（式 (4.10)）。top_I の値は実験的に決定する。

$$sim(\vec{c_i}, \vec{s_j})' = \sum_{i}^{top_I} (x_i y_i) \tag{4.10}$$

ここでの $x_i y_i$ は降順にソートされた $\vec{c_i}$ と $\vec{s_j}$ の要素の積の値を表わしている。

4.5.3 特徴ベクトルの補正

本項では、4.3 節で述べた手法で作成した特徴ベクトルに改良を加える。具体的には、語義の特徴ベクトルに対し補正を行い、ベクトルの値が 0 となる素性の数を減らすことを試みる。その方法として以下の 2 つをあげる。

1. 加算スムージング

2. 補正用ベクトルの加算

一つ目は、式 (4.11) で表わされるような単純な加算スムージングを施すというものである。

$$\vec{s_j}' = \begin{pmatrix} s_1 + sm \\ s_2 + sm \\ \cdot \\ \cdot \\ \cdot \\ s_m + sm \end{pmatrix} \tag{4.11}$$

s_m は式 (4.3) や式 (4.4) で作成した語義 S_j の特徴ベクトル $\vec{s_j}$ の m 行目の値を表わしており、m は $\vec{s_j}$ の次元数を、sm は全ての次元に加える定数をそれぞれ表わしている。

2 つ目は、式 (4.3) や式 (4.4) で作成した特徴ベクトルを基に補正用ベクトルを作成し、それを元のベクトルに加算する手法である。補正用ベクトルは以下の式 (4.12) で表わされる。

$$\vec{h_{s_j}} = \frac{1}{N}\Big(\sum_{t_m \in A_h} \vec{o}(t_m)\Big) \tag{4.12}$$

A_h は $\vec{s_j}$ の素性となる単語のうち、値が 0 でない単語の集合を表わしている。N は正規化定数である。この $\vec{h_{s_j}}$ を用いて式 (4.13) で示すようにベクトルの補正を行う。

$$\vec{s_j}' = \frac{1}{N}\Big(\vec{s_j} + w_c \cdot \vec{h_{s_j}}\Big) \tag{4.13}$$

$\vec{s_j}$ は式 (4.3) や式 (4.4) で作成した語義 S_j の元の特徴ベクトル、N は正規化定数である。w_c は補正ベクトルの重み付けに用いるためのパラメタで、この値は実験的に決定する。

また、式 (4.13) は元の特徴ベクトルの素性の値を考慮してはいない。そのため、元の特徴ベクトルの素性の値に応じて補正ベクトルに重みを付け加えることを試みる。具体的には式 (4.14) で $\vec{s_j}$ を補正する。

$$\vec{s_j}' = \frac{1}{N}\Big(\vec{s_j} + w_c \cdot s_j(t_m) \cdot \vec{h_{s_j}}\Big) \tag{4.14}$$

ここで $s_j(t_m)$ は、素性 t_m に対するベクトル $\vec{s_j}$ の値を示している。

4.6　小分類の語義の特徴ベクトルによる対応付け

本研究における語義の定義は、岩波国語辞典の中分類に該当する語義である。つまり、用例クラスタに対して中分類の語義を対応づけることが目標である。これには以下の2つの方法が考えられる。

手法1 中分類単位で特徴ベクトルを作る。下位の分類の語義の定義文や例文に使用される単語を全て使って特徴ベクトルを作成する。

手法2 小分類単位で特徴ベクトルを作る。クラスタに対し、小分類の語義を割り当てた後、対応する中分類の語義に変換する。

これまでは、手法1によってクラスタに対する語義の対応付けを行うことを考えていた。しかし、上記の2つの手法には一長一短がある。手法1は手法2に比べて、1つの特徴ベクトルを構築する際により多くの単語の共起ベクトルを足すことになる。つまり、特徴ベクトルが持つ情報量が多いと言える。しかし、手法1では小分類の語義をまとめて一つの特徴ベクトルを作るため、小分類のような細かい意味の区別がつかなくなる可能性もある。もし、小分類の語義のような細かい意味を持つ用例が集まってクラスタができている場合、手法2の方が手法1より対応付けに適していると思われる。そこで、手法1と手法2による語義の対応付けを行い、両者を実験的に比較する。

第5章　新語義判定

本章では、用例クラスタが辞書における既存の語義のいずれかにも該当しない新語義の用例を集めたものであるかを判定する手法について述べる。具体的には、ある対象単語に対し、n 個の用例クラスタ C_i が作成されたとき、4 章の手法によって C_i と辞書の既存の語義との類似度を求め、その結果を新語義判定の基準とする。以下の 5.1 節で既存語義近接度 K_i の求め方を、5.2 節で新語義の判定方法をそれぞれ説明する。

5.1　既存語義近接度

既存語義近接度とは、用例クラスタ C_i が、どのくらい既存の語義の意味に近いかを表わした指標のことである。既存語義近接度 K_i の求め方として以下の 3 つをあげる。

1. 既存語義との類似度の分散
2. 既存語義との類似度の最大値と最小値の開き
3. 既存語義との類似度の最大値

5.1.1　既存語義との類似度の分散

式 (5.1) のようなクラスタと辞書との類似度の分散値を既存語義近接度とする

$$K\text{-}Var_i = \frac{1}{n}\sum_{j=0}^{n}(sim(\vec{c_i}, \vec{s_j}) - \overline{sim_{\vec{c_i}}})^2 \tag{5.1}$$

n は対象単語 w に対して辞書で定義されている語義の総数、$\overline{sim_{\vec{c_i}}}$ はクラスタ C_i と辞書の語義との類似度の平均値をそれぞれ表わしている。既存語義のクラスタは、それに対応する辞書の語義との類似度がそれ以外の語義との類似度と比べて大きいと予測される。これに対し、新語義のクラスタは既存のどの語義とも似ていないため、どの語義との類似度も似たような値になり、類似度の分散が小さいと思われる。そのため、この $K\text{-}Var_i$ の値が小さいクラスタほど新語義クラスタである可能性が高い。

5.1.2 既存語義との類似度の最大値と最小値の開き

式 (5.2) のように類似度の最大値と最小値の差を既存語義近接度とする。

$$K\text{-}Diff_i = \max_j sim(\vec{c_i}, \vec{s_j}) - \min_j sim(\vec{c_i}, \vec{s_j}) \tag{5.2}$$

$K\text{-}Diff_i$ が大きい値であればあるほど、用例クラスタがある 1 つの語義に対して大きい類似度をとり、用例クラスタがその語義に対応する可能性が高いと言える。一方、新語義のクラスタ、どの語義とも低い類似度の値をとり、1 つの語義の類似度が突出して高いことはないと予想される。そのため、$K\text{-}Diff_i$ の値が低いクラスタは新語義である可能性が高い。

5.1.3 既存語義との類似度の最大値

式 (5.3) のように既存語義との類似度の最大値を既存語義近接度とする。

$$K\text{-}Max_i = \max_j sim(\vec{c_i}, \vec{s_j}) \tag{5.3}$$

新語義クラスタは既存語義と違う用法で使用している例文の集合であるため、既存語義とあまり高い類似性はないと思われる。この場合、$K\text{-}Max_i$ が小さければ小さいほど、その用例クラスタはどの語義ともそれほど類似性を持っていないため、新語義である可能性が高い。

5.2 新語義の判定

5.2.1 既存語義近接度による新語義の判定

前節で述べた、$K\text{-}Var_i$・$K\text{-}Diff_i$・$K\text{-}Max_i$ の値をそれぞれ算出し、それを基に新語義の判定を行う。単純な新語義の判定方法として、$K\text{-}Var_i$・$K\text{-}Diff_i$・$K\text{-}Max_i$ に対して閾値を決め、閾値以下ならその語義を新語義とみなすという判定方法が考えられる。しかし、予備調査により、用例クラスタに対し $K\text{-}Var_i$・$K\text{-}Diff_i$・$K\text{-}Max_i$ を求めてみたところ、新語義のクラスタほどこれらの値が低いという顕著な傾向は観察できなかった。そのため、既存語義近接度に対して、閾値を決めて新語義か否かを判定するという方法は有効ではないことがわかった。

5.2.2 既存語義近接度の差による新語義の判定

前項で述べた通り、既存語義近接度を算出し、その値を閾値と比較して新語義判定を行うことは困難である。そこで、$K\text{-}Var_i$・$K\text{-}Diff_i$・$K\text{-}Max_i$ の値を既存語義と新語義の境界の特定に用いることで、新語義の判定を行うこととする。まず用例クラスタを K_i の降順にソートする。以下、用例クラスタ C_i は、K_i の大きい順に並んでいるものとする。

単純な新語義判定の方法として、ソート後に一番最後の要素である用例クラスタを新語義クラスタとみなすという手法が考えられる。しかし、この場合以下の 2 つの問題が生じる。

1. 新語義クラスタが対象単語に対し必ず一つ検出される。しかし、対象単語によっては新単語が存在しないことも十分に考えられる。

2. 新語義クラスタの認識が一つしかできないため、新語義が複数ある場合に対応できない。

そこで上記のように並べられた C_i に対し、既存語義と新語義とを分ける境界を発見する (図 5.1)。

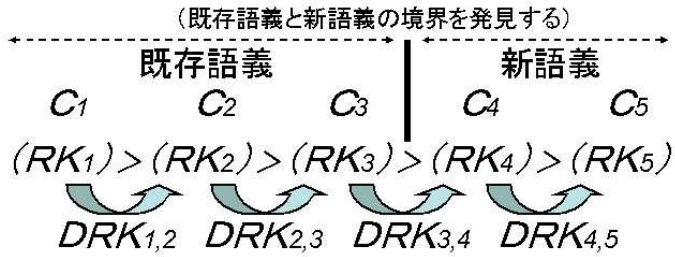

図 5.1: 新語義の判定

まず、相対既存語義近接度 RK_i を、最も大きい既存語義近接度 K_1 に対する K_i の相対値（$=K_i/K_1$）とする。さらに、隣接する用例クラスタ C_i と C_{i+1} の相対既存語義近接度の差を $DRK_{i,i+1}$ とする（式 (5.4)）。

$$DRK_{i,i+1} = RK_i - RK_{i+1} \tag{5.4}$$

この $DRK_{i,i+1}$ の値を利用して、境界発見のための二つの手法を提案する。

- DRK1

  $1 \leq i \leq N-1$ のうち、$DRK_{i,i+1}$ が最も大きい i を見つけ、$i+1$ 番目以降の用例ク

ラスタを C_{i+1}⋯ C_N は新語義であると判定する。これは、既存語義近接度の差が大きいところが既存語義と新語義との境界になっているという仮定に基づく。

- DRK2
  DRK1 の条件に加え、最大の $DRK_{i,i+1}$ の値が閾値 T_k よりも大きいときに限り、用例クラスタ C_{i+1}⋯ C_N を新語義と判定する。T_k 以下の場合は新語義に相当するクラスタは存在しないものとする。すなわち、既存語義近接度の差が十分大きくなければ既存語義と新語義との境界とはみなさない。

また、DRK2 で閾値 T_k を設定する際、用例クラスタと辞書の語義との類似度の大きさは対象単語によってばらつきがあるため、既存語義近接度 K_i の差に対して閾値を設定することは困難である。そのため、相対化した既存語義近接度 RK_i に対して閾値 T_k を設定した。DRK2 では先に挙げた二つの問題点は解決されると思われる。

第6章　評価

本章では、各提案手法の評価実験について述べる。

6.1　実験データ

ここでは、実験の対象となるデータについて述べる。

まず、実験に用いるコーパスとして、Yahoo!知恵袋コーパスを用意した。Yahoo!知恵袋[1]とは、インターネット電子掲示板サイトの 1 つで、質問と質問に対する答えによって構成されている。本コーパスは、Yahoo!知恵袋に投稿された質問とベストアンサーの組 45,725 件からなるコーパスである。ここで、Yahoo!知恵袋コーパスを用いたのは、新聞記事と比べ日常的に使用されている自然な日本語文が得られ、多様な語義が出現すると考えたためである。次に、実験対象となる単語を選定した。Yahoo!知恵袋コーパスにおいて出現数が少なくとも 100 件あり、2 つ以上の語義で使われているような単語 23 個を対象語とした。そして、対象語のそれぞれに対し、そのインスタンスをコーパスから 100 個ランダムに選択し、人手で語義を付与した。正解の語義は、岩波国語辞典で対象語の見出しを引いて、中分類の語義の中に該当すると思われる語義があればそのラベルを、なければ新語義のラベルをそれぞれ付与した。また、同じ対象語で新語義が二種類以上出現した場合、それらを別々の語義としてラベル付けしている。これらの正解の語義の情報は、用例のクラスタリング、クラスタに対する語義の対応付け、新語義の判定評価に用いる。実験に用いた 23 個の対象単語を表 6.1 に示す。また、各単語の辞書の定義文、ならびに新語義を付録 A に記す。

表 6.1: 対象単語の一覧

モデル、ネタ、カバー、ウイルス、ソース、肉、サービス、地方、アルバム
コード、自分、場合、時間、意味、電話、一緒、目、以前、代、顔、系、郵便、反応

[1]http://chiebukuro.yahoo.co.jp/

6.2 評価実験

前節で作成したテストデータを使用して以下の 3 種類の評価実験を行った。

1. 単語の用例のクラスタリング
2. 用例クラスタと辞書の語義との対応付け
3. 新語義判定

以降、各項で詳しく説明する。

6.2.1 用例のクラスタリング

ここでは、用例クラスタリングの評価を行う。3 章で提案した以下の 3 つのクラスタリング手法を比較する。

- クラスタリングアルゴリズム
    1. K-means（3.1 節）
    2. K-means+KKZ 法（3.2.1 節）
    3. トップダウン分割法（3.2.2 節）

また、用例のベクトルの作成方法として九岡が提案する以下の 6 種類の手法を比較した。

- ベクトルの作成方法（九岡の手法を用いる）
    1. 単独の特徴ベクトル
        (a) 隣接ベクトル
        (b) LDA 拡張文脈ベクトル
        (c) 連想ベクトル
        (d) トピックベクトル
    2. ベクトルを組み合わせる手法
        (a) 評価関数として rel_intra を用いる
        (b) 評価関数として rel_coh を用いる

いずれも作成するクラスタ数は 10 とした。評価は、クラスタ内の純度 purity と、語義識別率、新語義識別率の 3 つの観点から行うものとする。以下、それぞれの定義について述べる。

Purity

$$Purity = \sum_i \frac{|C_i|}{N} \times \frac{\max_j n_i(S_j)}{|C_i|} \tag{6.1}$$

ここで、N はインスタンスの総数、$|C_i|$ は個々のクラスタの含まれる用例の数、$n_i(S_j)$ はクラスタ C_i の中で語義が S_j である用例の数をそれぞれ表わしている。つまり、大きなクラスタにおいて同じ語義が占める割合が高ければ、Purity は高い値をとる。

語義の識別率

まず、用例クラスタに対し、そのクラスタにおいて最も多く出現する語義を「クラスタの語義ラベル」とし、そのクラスタは「語義ラベル」の意味に対応するとみなす。次に、語義識別率を式 (6.2) のように定義する。

$$\text{語義識別率} = \frac{\text{クラスタの語義ラベルの異なり数の総和}}{\text{真の語義数の総和}} \tag{6.2}$$

ここでの「真の語義の総和」とは、23 個の対象語について、クラスタリングを行う用例集合に出現する語義の数の和を表わす。ある語義の用例がコーパスの中には存在していても、その語義がクラスタの中では他の語義に比べて常に少数しか存在しなければ、その語義はクラスタとして識別できたとはいえない。つまり、ある語義の用例がクラスタ内で最多であるクラスタが作成されて始めて、その語義は用例クラスタとして識別できたと言える。語義の識別率は、コーパスに出現した語義のうち、上記のように用例クラスタとして識別できたものの割合を示している。

また、新語義の語義識別率を式 (6.3) に示す。新語義識別率は、コーパスに出現した新語義のうち、用例クラスタとして識別できたものの割合である。

$$\text{新語義識別率} = \frac{\text{クラスタの新語義ラベルの異なり数の総和}}{\text{新語義の語義数の総和}} \tag{6.3}$$

単独の特徴ベクトルを用たときの purity を表 6.2 に、語義識別率と新語義識別率を表 6.3 に示す。なお、表での「KKZ」は K-means+KKZ 法を、「Top」はトップダウン分割法をそれぞれ表わしている。

複数の特徴ベクトルを組み合わせたクラスタリングの purity を表 6.4 に、語義識別率と新語義識別率を表 6.5 に示す。

表 6.2、表 6.4 をみると、purity は、K-means 法・トップダウン分割法に比べて K-means + KKZ 法の方が若干上回った。これは、K-means + KKZ 法が初期クラスタとして多様な

表 6.2: 単独の特徴ベクトルを用いたクラスタリングの purity

	隣接			連想			トピック			LDA 拡張文脈		
	K-meas	KKZ	Top	K-means	KKZ	Top	K-means	KKZ	Top	K-means	KKZ	Top
モデル	0.68	0.66	0.69	0.85	0.88	0.88	0.83	0.79	0.86	0.76	0.71	0.78
ネタ	0.7	0.62	0.63	0.61	0.62	0.68	0.63	0.61	0.63	0.64	0.6	0.63
カバー	0.62	0.6	0.69	0.69	0.69	0.71	0.7	0.74	0.69	0.61	0.53	0.61
ウイルス	0.86	0.87	0.87	0.99	0.99	0.99	0.95	0.95	0.95	0.85	0.87	0.9
ソース	0.762	0.762	0.733	0.73	0.73	0.75	0.911	0.901	0.881	0.802	0.782	0.762
肉	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.98	0.96	0.96	0.97	0.96
サービス	0.67	0.69	0.663	0.67	0.7	0.693	0.7	0.69	0.703	0.67	0.7	0.663
地方	0.72	0.74	0.76	0.74	0.65	0.76	0.72	0.71	0.76	0.69	0.66	0.75
アルバム	0.91	0.92	0.91	0.91	0.91	0.91	0.91	0.95	0.91	0.91	0.91	0.91
コード	0.56	0.55	0.56	0.758	0.717	0.79	0.73	0.74	0.75	0.63	0.64	0.68
自分	0.68	0.68	0.72	0.64	0.64	0.64	0.66	0.66	0.71	0.66	0.69	0.69
場合	0.72	0.67	0.75	0.61	0.59	0.58	0.67	0.72	0.62	0.66	0.61	0.65
時間	0.8	0.81	0.8	0.64	0.65	0.61	0.59	0.65	0.59	0.67	0.64	0.59
意味	0.71	0.73	0.72	0.704	0.714	0.69	0.73	0.72	0.73	0.69	0.69	0.69
電話	0.78	0.78	0.84	0.75	0.76	0.76	0.75	0.75	0.73	0.68	0.77	0.75
一緒	0.82	0.95	0.88	0.83	0.83	0.82	0.82	0.82	0.83	0.83	0.82	0.82
目	0.76	0.73	0.72	0.55	0.56	0.5	0.59	0.56	0.62	0.65	0.58	0.54
以前	0.87	0.91	0.88	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87	0.87
代	0.869	0.919	0.81	0.798	0.768	0.77	0.707	0.758	0.73	0.758	0.677	0.68
顔	0.7	0.63	0.7	0.7	0.76	0.68	0.67	0.71	0.65	0.66	0.64	0.68
系	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.87	0.86	0.86	0.86
郵便	0.9	0.9	0.9	0.92	0.92	0.9	0.91	0.91	0.91	0.9	0.9	0.9
反応	0.8	0.78	0.78	0.82	0.83	0.82	0.81	0.8	0.81	0.79	0.79	0.82
purity	0.77	0.771	0.775	0.765	0.765	0.766	0.769	0.776	0.772	0.748	0.735	0.747

要素を選択できるためにクラスタリングの精度が向上し、purity が他の手法と比べて高くなったと思われる。

表 6.3、表 6.5 をみると、語義認識率も、K-means + KKZ 法は他の手法と比べて高い値であった。一方、新語義識別率はトップダウン分割法が一番高かった。中でも単独のベクトルでは、連想ベクトルとトピックベクトルの場合、ベクトルの組み合わせでは評価関数 rel_coh を用いた場合に、新語義識別率は 0.421 という値であった。しかし、0.421 は決して十分ではなく、新語義をもつ用例がクラスタとしてまとめることができたとは言い難い。

表 6.3: 単独の特徴ベクトルを用いたクラスタリングの語義識別率、新語義識別率

	語義数（新語義の語義数）												
		隣接			連想			トピック			ＬＤＡ拡張文脈		
		K-means	KKZ	Top	K-means	KKZ	Top	K-means	KKZ	Top	K-means	KKZ	Top
モデル	5(2)	2(1)	2(1)	2(1)	2(1)	3(1)	3(1)	2(1)	2(1)	2(1)	2(1)	2(1)	2(1)
ネタ	5(2)	3(2)	2(1)	3(2)	4(2)	2(0)	4(2)	3(1)	2(1)	3(1)	3(2)	3(2)	3(2)
カバー	4(2)	3(1)	4(2)	3(1)	3(1)	3(1)	4(2)	3(1)	4(2)	4(2)	3(1)	4(2)	3(1)
ウイルス	2	2	2	2	2	2	2	2	2	2	1	2	2
ソース	4(2)	2(1)	2(1)	2(1)	1(0)	1(0)	2(1)	3(1)	3(1)	2(1)	2(1)	3(1)	2(1)
肉	4	1	1	1	1	1	1	1	2	1	1	2	1
サービス	5(2)	1(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	3(1)	1(0)	3(0)	2(0)
地方	2	2	2	2	2	2	2	2	2	2	2	2	2
アルバム	2	1	2	1	1	1	1	1	2	1	1	1	1
コード	4(1)	3(0)	3(0)	3(0)	3(0)	3(0)	3(0)	3(0)	3(0)	3(0)	3(0)	2(0)	3(0)
自分	3(1)	2(0)	2(0)	2(0)	2(0)	2(0)	1(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)
場合	4(2)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)
時間	4	3	2	2	2	3	2	2	4	2	2	2	2
意味	3	2	2	2	1	2	1	2	2	2	1	1	1
電話	3(1)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)
一緒	2	1	2	2	2	2	1	1	2	2	2	1	1
目	5	2	3	2	2	2	2	2	2	2	2	2	2
以前	2	1	2	2	1	1	1	1	1	2	1	1	1
代	6(1)	2(0)	2(0)	2(0)	2(0)	2(0)	3(1)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)
顔	3	2	2	2	2	3	2	2	2	2	2	3	2
系	4(1)	1(0)	1(0)	1(0)	1(0)	1(0)	1(0)	1(0)	1(0)	2(0)	1(0)	1(0)	1(0)
郵便	3(1)	1(0)	1(0)	1(0)	2(0)	2(0)	1(0)	2(0)	2(0)	2(1)	1(0)	1(0)	1(0)
反応	3(1)	2(1)	1(0)	1(0)	3(1)	3(1)	3(1)	2(1)	3(1)	2(1)	2(1)	2(1)	2(1)
合計	82(19)	43(6)	46(5)	44(5)	45(5)	47(3)	46(8)	45(5)	51(6)	49(8)	41(6)	46(7)	42(6)
語義識別率		0.524	0.561	0.537	0.549	0.573	0.561	0.549	0.622	0.598	0.5	0.561	0.512
新語義識別率		0.316	0.263	0.263	0.263	0.158	0.421	0.263	0.316	0.421	0.316	0.368	0.316

表 6.4: 複数の特徴ベクトルを用いたクラスタリングの purity

	rel_intra			rel_coh		
	K-means	KKZ	Top	K-means	KKZ	Top
モデル	0.82	0.83	0.88	0.66	0.82	0.88
ネタ	0.6	0.62	0.68	0.64	0.62	0.63
カバー	0.72	0.6	0.71	0.72	0.6	0.69
ウイルス	0.92	0.95	0.99	0.95	0.95	0.95
ソース	0.901	0.891	0.75	0.901	0.901	0.733
肉	0.96	0.98	0.96	0.97	0.98	0.96
サービス	0.69	0.71	0.693	0.67	0.69	0.663
地方	0.69	0.66	0.76	0.76	0.74	0.76
アルバム	0.91	0.96	0.91	0.91	0.96	0.91
コード	0.77	0.77	0.79	0.74	0.73	0.75
自分	0.65	0.69	0.64	0.66	0.68	0.64
場合	0.74	0.67	0.58	0.67	0.67	0.75
時間	0.61	0.81	0.61	0.83	0.81	0.8
意味	0.73	0.73	0.69	0.69	0.72	0.72
電話	0.75	0.78	0.76	0.85	0.78	0.84
一緒	0.94	0.95	0.82	0.87	0.95	0.88
目	0.55	0.58	0.5	0.58	0.73	0.5
以前	0.87	0.91	0.87	0.87	0.91	0.87
代	0.879	0.919	0.77	0.879	0.919	0.81
顔	0.71	0.67	0.68	0.66	0.69	0.7
系	0.88	0.86	0.86	0.86	0.86	0.86
郵便	0.9	0.91	0.9	0.9	0.91	0.9
反応	0.78	0.78	0.82	0.78	0.82	0.82
purity	0.781	0.793	0.766	0.783	0.802	0.783

表 6.5: 複数の特徴ベクトルを用いたクラスタリングの語義識別率、新語義識別率

	総語義数（新語義の語義数）						
		rel_intra			rel_coh		
		K-means	KKZ	Top	K-means	KKZ	Top
モデル	5(2)	2(1)	3(1)	3(1)	2(1)	3(1)	3(1)
ネタ	5(2)	2(1)	2(1)	4(2)	3(2)	2(1)	3(2)
カバー	4(2)	3(1)	4(2)	4(2)	3(1)	4(2)	3(1)
ウイルス	2	2	2	2	2	2	2
ソース	4(2)	2(1)	3(1)	2(1)	2(1)	3(1)	2(1)
肉	4	1	2	1	2	3	1
サービス	5(2)	2(0)	2(0)	2(0)	1(0)	2(0)	2(0)
地方	2	2	2	2	2	2	2
アルバム	2	1	2	1	1	2	1
コード	4(1)	3(0)	3(0)	3(0)	3(0)	3(0)	3(0)
自分	3(1)	2(0)	2(0)	1(0)	2(0)	2(0)	1(0)
場合	4(2)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)
時間	4	2	2	2	2	2	2
意味	3	2	2	1	2	2	2
電話	3(1)	2(0)	2(0)	2(0)	2(0)	2(0)	2(0)
一緒	2	2	2	1	2	2	2
目	5	2	2	2	2	3	2
以前	2	1	2	1	1	2	1
代	6(1)	2(0)	2(0)	3(1)	2(0)	2(0)	2(0)
顔	3	2	2	2	2	2	2
系	4(1)	2(0)	1(0)	1(0)	1(0)	1(0)	1(0)
郵便	3(1)	2(0)	2(0)	1(0)	1(0)	2(0)	1(0)
反応	3(1)	1(0)	1(0)	3(1)	2(1)	2(1)	3(1)
合計	82(19)	44(4)	49(5)	46(8)	44(6)	52(6)	45(6)
語義識別率		0.537	0.598	0.561	0.537	0.634	0.549
新語義識別率		0.211	0.263	0.421	0.316	0.316	0.316

6.2.2　用例クラスタと辞書の語義との対応付け

ここでは、用例クラスタと辞書の語義との対応付けの結果について述べる。ここで用いる用例クラスタは、本来ならば3章で説明した手法で自動的に作成されたクラスタである。しかし、自動的に作成されたクラスタは以下の問題を含む。

1. 違う語義を持つ用例が1つのクラスタにまとめられるという誤りを含む。
2. 6.2.1項で述べたように語義識別率・新語義識別率が低いことから、現状の用例クラスタは十分に語義の識別がなされているとは言い難い。
3. 対象語によっては、クラスタの語義ラベルが全て同じであるという極端な偏りが存在する。

そのため、ここでは用例クラスタを語義と対応付ける手法を評価するために、人手で作成した完全に正しい用例クラスタを用いた。具体的には、コーパスから抽出された用例に対して人手で語義を付与し、同じ語義を持つ用例をまとめてクラスタを作成した。評価関数としては式(6.4)で算出する正解率を用いる。なお、ここでは用例クラスタと既存の辞書の語義との対応付けの正確さを測るため、新語義とラベル付けされたクラスタは評価からは除外している。

$$\text{正解率} = \frac{\text{正しい語義に対応付けられたクラスタの数}}{\text{辞書の語義に対応するクラスタの総数}} \tag{6.4}$$

6.2.2.1 4.2 節の手法の評価

4.2 節では語義の特徴ベクトルを構築する際、定義文に出現する自立語の共起ベクトルの和を求める手法 (式 (4.3)) と、定義文と例文に出現する自立語の共起ベクトルの和を求める手法 (式 (4.4)) とをそれぞれ提案した。式 (4.4) における例文の重みを変えた時の、正しい語義に対応付けることができたクラスタの数ならびに正解率を表 6.6 に示す。$w_e = 0.0$ の時は、例文を用いない手法 (式 (4.3)) を表わす。表中の「クラスタ数」は対象単語ごと

表 6.6: w_e の値に対する正解率の変動

		w_e										
	クラスタ数	0.0	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	10.0
モデル	3	2	2	2	2	2	2	2	2	2	2	2
ネタ	3	1	1	1	1	1	1	1	1	1	1	0
カバー	2	1	2	1	1	1	1	1	1	1	1	1
ウイルス	2	1	1	1	1	1	1	1	1	1	1	1
ソース	2	1	1	1	1	1	1	1	1	1	1	1
肉	4	0	2	2	2	2	2	2	2	2	2	2
サービス	3	1	1	1	1	1	1	1	1	1	1	1
地方	2	1	0	0	0	0	0	0	0	0	0	0
アルバム	2	1	1	1	1	2	2	2	2	2	2	1
コード	3	1	2	2	2	2	2	2	2	2	2	1
自分	2	1	1	1	1	1	0	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1	1
時間	4	2	2	1	1	1	1	1	1	1	1	1
意味	3	1	1	1	1	1	1	1	1	0	1	0
電話	2	1	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1	1
目	5	0	2	2	2	2	2	2	2	2	2	1
以前	2	1	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	3	2	2	2	2	2	1
顔	3	1	1	1	1	1	1	1	1	1	1	1
系	3	1	1	1	1	1	1	1	1	1	1	1
郵便	2	1	2	2	1	1	1	1	1	1	1	1
反応	2	1	1	1	1	1	1	1	1	1	1	1
合計	63	25	31	29	28	29	27	28	28	27	28	22
正解率		0.396	0.492	0.46	0.444	0.46	0.428	0.444	0.444	0.428	0.444	0.349

のクラスタ（新語義に対応するクラスタは除く）の数である。表中の「正解率」の行は対応付けの正解数を、それ以外の行は各対象単語ごとの対応付けに正解したクラスタの数を表わしている。

辞書とクラスタとの対応付けにおいて、辞書語釈文中の「例文」を用いた方が正解率が高かった。「例文」は語義の「定義文」と比べると、クラスタにおける用例と似ている。そのため辞書とクラスタとの関連度を測る際に重要な手がかりとなりえる。このことは直感的にも理解しやすい。しかし、w_e の値を大きくし、「例文」に重点をおきすぎるとかえって正解率は下がってしまった。これは「例文」の中にある単語によって、特徴ベクトルにノイズが入ってしまう場合があるからと思われる。この実験では、$w_e = 1.0$ のときに正解率がもっとも高かった。

6.2.2.2　4.4 節の手法の評価

4.4 節では、特徴ベクトルの作成方法に対する以下の 3 つの改良案について述べた。

1. 特徴ベクトルを作成する際に、対象単語に該当する共起ベクトルを考慮しない（4.4.1 項）
2. 単語の出現頻度を無視して特徴ベクトルを作成する（4.4.2 項）
3. 用例における単語のクラスタ頻度を考慮する（4.4.3 項）

それぞれの改良案について、対する正しい語義に対応付けることができたクラスタの数ならびに正解率を表 6.7 に示す。なお、w_e は 1.0 とした。

表 6.7: 4.4 節の手法の正解率

		手法		
	クラスタ数	4.4.1 項	4.4.2 項	4.4.3 項
モデル	3	2	1	2
ネタ	3	1	1	1
カバー	2	1	1	1
ウイルス	2	1	1	1
ソース	2	1	1	1
肉	4	2	1	1
サービス	3	1	2	2
地方	2	0	0	0
アルバム	2	1	1	1
コード	3	2	1	2
自分	2	1	1	1
場合	2	1	1	1
時間	4	2	1	1
意味	3	1	1	1
電話	2	1	1	1
一緒	2	1	1	1
目	5	2	2	2
以前	2	1	1	1
代	5	3	3	3
顔	3	1	1	2
系	3	1	1	1
郵便	2	2	1	2
反応	2	1	1	1
合計	63	30	26	30
正解率		0.476	0.412	0.476

4.4 節で説明した 3 つの方法では、対応付けの正解率は向上せず、逆に正解率が下がってしまった。

6.2.2.3　4.5 節の手法の評価

4.5.1 項では、式 (4.4) で特徴ベクトルを作成し、作成した特徴ベクトルの要素をベクトルの値の降順に並べかえ、値の上位 top_x 語で単語リスト A_{top} を作成し、A_{top} を用いて特徴ベクトルを作成する手法（式 (4.8)）を提案した。top_x の値を変えた時の正解率を表 6.8 に示す。なお、式 (4.4) における w_e は 1.0 とした。

表 6.8: top_x の値に対する正解率の変動

		top_x の値				
	クラスタ数	10	20	30	50	100
モデル	3	1	1	1	1	2
ネタ	3	1	2	2	1	1
カバー	2	1	1	1	1	1
ウイルス	2	2	2	2	2	2
ソース	2	2	2	2	2	2
肉	4	2	2	1	2	2
サービス	3	1	1	1	2	2
地方	2	1	1	1	0	0
アルバム	2	2	2	2	2	2
コード	3	2	2	2	2	3
自分	2	1	1	1	1	1
場合	2	1	1	1	1	1
時間	4	1	2	2	1	1
意味	3	1	3	3	2	1
電話	2	1	1	1	1	1
一緒	2	1	1	1	1	1
目	5	1	0	1	2	1
以前	2	1	1	1	1	1
代	5	2	2	3	3	3
顔	3	0	1	1	1	1
系	3	0	0	1	1	2
郵便	2	1	1	1	1	1
反応	2	1	1	1	1	1
合計	63	27	31	33	32	33
正解率		0.428	0.492	0.523	0.507	0.523

特徴ベクトルを作成する際に用いる単語リスト A_{top} の量が多いほど対応付けの正解率が高くなっている。この結果から、特徴ベクトルの構築の際に和を求める単語の量は多ければ多いほど対応付けの正解率が高いことが言える。この実験では、$top_x = 30$ と $top_x = 100$ のときに正解率が最も高かった。

4.5.2 項では、クラスタと語義の特徴ベクトルの類似度の計算方法を式 (4.10) に変更する手法を提案した。top_I の値を変えた時の正解率を表 6.9 に示す。なお、式 (4.4) における w_e は 1.0 とした。

表 6.9: top_I の値に対する正解率の変動

			類似度計算に用いる内積要素の個数								
	クラスタ数	補正なし	10	20	30	40	50	100	150	200	300
モデル	3	2	3	3	3	3	2	2	2	2	2
ネタ	3	1	0	0	0	0	0	1	1	1	1
カバー	2	2	1	1	1	1	1	2	2	2	2
ウイルス	2	1	2	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	2	1	1	1	1	1	1	1	1	1
サービス	3	1	1	2	2	2	2	2	2	2	1
地方	2	0	0	0	1	1	1	1	1	1	0
アルバム	2	1	2	2	2	2	2	2	2	2	2
コード	3	2	1	2	2	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	0	0	0	1
場合	2	1	1	0	1	1	1	1	1	1	1
時間	4	2	1	1	1	2	2	2	1	2	2
意味	3	1	1	1	1	1	1	2	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	0	0	0	0	0	0	0	0	0
以前	2	1	0	1	1	1	1	1	1	1	1
代	5	3	1	2	2	2	1	3	4	4	4
顔	3	1	1	1	1	1	1	1	2	2	2
系	3	1	1	1	1	1	1	1	1	1	1
郵便	2	2	2	2	2	2	2	2	2	2	2
反応	2	1	1	1	1	1	1	1	1	1	1
合計	63	31	24	27	29	30	28	32	32	33	32
正解率		0.492	0.38	0.428	0.46	0.476	0.444	0.507	0.507	0.523	0.507

$top_I = 100$ 以上のときには、正解率は若干向上した。しかし、十分な精度の向上は得られなかった。

4.5.3 項では、4.2 節で述べた手法で作成した特徴ベクトルに対し補正を行い、ベクトルの値が 0 となる素性の数を減らす手法を 2 種類提案した。1 つは加算スムージング（式 (4.11)）で、もう 1 つは補正用ベクトルを加算する手法（式 (4.13)・式 (4.14)）である。

式 (4.4) における w_e の値を 0.0、1.0 とした場合で、sm の値を変えた時のそれぞれの正解率を表 6.10 に示す。

表 6.10: sm の値に対する正解率の変動

		$w_e = 0.0$						$w_e = 1.0$					
		スムージング値						スムージング値					
	クラスタ数	0	0.005	0.05	0.5	1	3	0	0.005	0.05	0.5	1	3
モデル	3	2	2	2	2	2	2	2	2	2	2	2	2
ネタ	3	1	1	1	1	1	0	1	1	1	1	1	0
カバー	2	1	1	1	1	1	1	2	1	1	1	1	1
ウイルス	2	1	1	1	1	1	1	1	1	1	1	1	1
ソース	2	1	1	1	1	1	1	1	1	1	1	1	1
肉	4	0	0	0	0	0	0	2	2	2	1	0	0
サービス	3	1	1	1	1	1	1	1	1	1	1	1	1
地方	2	1	1	1	1	1	1	0	0	0	0	0	0
アルバム	2	1	1	1	1	1	1	1	1	1	1	1	1
コード	3	1	1	1	1	1	1	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1	1	1
時間	4	2	2	2	2	2	2	2	2	2	2	2	2
意味	3	1	1	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1	1	1
目	5	0	0	1	1	1	1	2	2	2	2	2	2
以前	2	1	1	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	3	2	3	3	3	3	3	3
顔	3	1	1	1	1	1	1	1	1	1	1	1	1
系	3	1	1	1	1	1	1	1	1	1	1	1	1
郵便	2	1	1	1	1	1	1	2	2	2	2	2	2
反応	2	1	1	1	1	1	1	1	1	1	1	1	1
合計	63	25	25	26	26	26	24	31	30	30	29	28	27
正解率		0.396	0.396	0.412	0.412	0.412	0.38	0.492	0.476	0.476	0.46	0.444	0.428

表 6.10 をみると、式 (4.4) における w_e の値を 0.0 にした場合には若干の正解率の向上がみられたが、w_e の値を 1.0 にした場合は、逆に正解率は低下した。これは、加算スムージングのように全ての要素に対し一律に値を足すと、元のベクトルの特徴が失われるためと考えられる。

次に、補正ベクトルを足す手法の評価を行う。まず、補正ベクトルを足すときに、元のベクトルの要素による重み付けを行わない手法（式(4.13)）を評価した。ここでは、w_c の値を 0ー100 まで変動させた。また、式 (4.4) における例文の重み w_e が 0.0 のときと 1.0 のときで実験を行った。$w_e = 0.0$ のときの結果を表 6.11 に、$w_e = 1.0$ のときの結果を表 6.12 に示す。

表 6.11: 式 (4.13) の補正による語義の対応付けの正解率 ($w_e = 0.0$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	2	2	2	2	2	2
ネタ	3	1	1	1	1	1	1	1	1	1	1
カバー	2	1	1	1	1	1	1	1	1	1	1
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	0	0	0	1	0	0	0	1	1	1
サービス	3	1	1	1	1	1	2	2	2	2	2
地方	2	1	1	1	1	1	1	1	1	1	1
アルバム	2	1	1	2	2	2	2	2	2	2	2
コード	3	1	1	1	1	3	3	3	3	3	3
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	2	2	2	2	1	1	1	1	1	1
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	0	0	1	0	0	0	0	0	0	0
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	3	3	3	3	3	2
顔	3	1	1	1	1	1	1	1	1	2	2
系	3	1	1	1	1	0	0	0	0	0	0
郵便	2	1	1	1	1	1	1	1	1	1	1
反応	2	1	1	1	1	2	2	2	2	2	2
合計	63	25	25	28	28	28	29	29	30	31	30
正解率		0.396	0.396	0.444	0.444	0.444	0.46	0.46	0.476	0.492	0.476

表 6.11 と表 6.12 をみると、補正ベクトルを足した方が足さないときより正解率が向上した。補正用ベクトルの加算は、辞書の語釈文に出現する単語の共起の共起の関係である。そのため補正する前の元となる特徴ベクトルに比べ、その語義に対する特徴は薄まっているものと思われる。しかし実験では、元となる特徴ベクトルよりも補正の方に比重をかけたほうが正解率は向上した。

また、例文を用いた特徴ベクトル（式 (4.4) における $w_e = 1.0$ の場合）に対して、補正ベクトルを足す手法は、例文を使用しない特徴ベクトル（式 (4.4) における $w_e = 0.0$ の場合）と比べて、正解率が高かった。このことから、補正ベクトルを足す場合でも例文を用いた方が良いということが言える。

表 6.12: 式 (4.13) の補正による語義の対応付け正解率 ($w_e = 1.0$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	3	3	3	2	2	1
ネタ	3	1	1	1	1	1	1	2	2	2	2
カバー	2	2	1	1	2	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	1
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	2	2	1	1	0	0	0	1	1	1
サービス	3	1	1	1	1	2	2	2	2	2	1
地方	2	0	0	0	0	1	1	1	1	1	1
アルバム	2	1	1	2	2	2	2	2	2	2	2
コード	3	2	2	1	1	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	0
時間	4	2	2	2	2	2	1	1	2	1	1
意味	3	1	1	1	1	2	1	1	1	1	1
電話	2	1	1	1	1	2	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	2	2	2	1	1	0	0	0	0
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	3	2	2	2	1	1
顔	3	1	1	1	1	2	2	2	1	1	1
系	3	1	1	1	1	1	1	1	1	1	1
郵便	2	2	2	2	2	1	1	1	1	1	1
反応	2	1	1	1	1	2	2	1	1	1	1
合計	63	31	30	30	31	36	32	31	31	29	25
正解率		0.492	0.476	0.476	0.492	0.571	0.507	0.492	0.492	0.46	0.396

次に、補正ベクトルを足すときに、元のベクトルの要素による重み付けを行う手法（式(4.14)）を評価した。ここでも、w_c の値を 0ー100 まで変動させた。また、式 (4.4) における例文の重み w_e が 0.0 のときと 1.0 のときで実験を行った。$w_e = 0.0$ のときの結果を表 6.13 に、$w_e = 1.0$ のときの結果を表 6.12 に示す。

表 6.13: 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 0.0$)

		w_c									
	クラスタ数	0.0	× 0.01	× 0.5	× 1.0	× 5.0	× 8.0	× 10.0	× 15.0	× 20.0	× 100.0
モデル	3	2	2	2	2	2	2	2	2	2	2
ネタ	3	1	1	1	1	1	1	1	1	1	1
カバー	2	1	1	2	2	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	0	0	0	0	0	0	0	0	0	0
サービス	3	1	1	1	1	1	2	2	2	2	2
地方	2	1	1	1	1	1	1	1	1	1	1
アルバム	2	1	1	2	2	2	2	2	2	2	2
コード	3	1	1	1	1	3	3	3	3	3	3
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	2	2	2	2	1	1	1	1	1	1
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	0	0	1	0	0	0	0	0	0	0
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	2	3	3	3	3	3	3	3	1
顔	3	1	1	1	1	1	1	1	1	1	2
系	3	1	1	1	1	0	0	0	0	0	0
郵便	2	1	1	1	1	1	1	1	1	1	1
反応	2	1	1	1	1	2	2	2	2	2	2
合計	63	25	24	29	28	29	30	30	30	30	29
正解率		0.396	0.38	0.46	0.444	0.46	0.476	0.476	0.476	0.476	0.46

表 6.13 と表 6.12 の場合でも、元の特徴ベクトルで例文を用いた方が、用いないときと比べ、全体的な正解率が高かった。

また、表 6.11 と表 6.13 を比較すると、表 6.13 の方が、若干正解率が高い。表 6.12 と表 6.14 とを比較しても、表 6.14 の方が全体的な正解率は高かった。

これらのことから、以下の 2 つのことが言える。

1. 特徴ベクトルを作成する際に、例文を用いた方が対応付けの正解率が上昇する。

2. 補正ベクトルを足す場合、元のベクトルの要素による重み付けを行った方が良い。つまり、式 (4.13) より、式 (4.14) を用いた方が対応付けの正解率が上昇する。

表 6.14: 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 1.0$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	3	3	3	3	3	3
ネタ	3	1	1	1	1	2	2	2	2	2	2
カバー	2	2	1	1	2	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	2	2	2	1	1	0	0	0	0	0
サービス	3	1	1	1	1	2	2	2	2	2	2
地方	2	0	0	0	0	1	1	1	1	1	1
アルバム	2	1	1	2	2	2	2	2	2	2	2
コード	3	2	2	1	1	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	2	2	2	2	2	2	2	2	2	2
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	2	2	2	1	1	1	1	1	1
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	4	3	3	3	3	3
顔	3	1	1	1	1	1	2	2	2	2	2
系	3	1	1	1	2	1	1	1	1	1	1
郵便	2	2	2	2	2	1	1	1	1	1	1
反応	2	1	1	1	1	1	2	2	2	2	2
合計	63	31	30	31	32	35	35	35	35	35	35
正解率		0.492	0.476	0.492	0.507	0.555	0.555	0.555	0.555	0.555	0.555

次に、例文の重み w_e と補正ベクトルの重み w_c の最適な組み合わせを調査する。w_e と w_c の組み合わせ方によって対応付けの正解率がさらに向上することが予想される。調査方法としては、今まで通り w_e の値を固定し、w_c の値を 0ー100 まで変動させる方法をとる。ただし、w_e はこれまでの実験より、あまり値を高くしても正解率の向上は期待できない。そのため、調べる w_e の値を 0.5、1.5、2.0、2.5 の 4 つとした。また、辞書の語義の特徴ベクトルを作成する式は、より正解率が高くなると予想される、式 (4.4) と式 (4.14) を用いた。

$w_e = 0.5$ のときの結果を表 6.15 に、$w_e = 1.5$ のときの結果を表 6.16 に、$w_e = 2.0$ のときの結果を表 6.17 に、$w_e = 2.5$ のときの結果を表 6.18 にそれぞれ示す。

表 6.15: 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 0.5$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	2	2	3	3	3	3
ネタ	3	0	0	1	1	2	2	2	2	2	2
カバー	2	1	1	1	2	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	1	1	1	1	1	0	0	0	0	0
サービス	3	1	1	1	1	2	2	2	2	2	2
地方	2	1	1	1	1	1	1	1	1	1	1
アルバム	2	2	2	2	2	2	2	2	2	2	2
コード	3	2	2	2	2	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	1	1	1	1	1	1	1	1	1	1
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	2	2	2	1	0	0	0	0	0
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	3	2	2	2	2	2
顔	3	1	1	1	1	1	2	2	2	2	2
系	3	2	2	2	2	1	1	1	1	1	1
郵便	2	1	1	1	1	1	1	1	1	1	1
反応	2	1	1	1	1	1	2	2	2	2	2
合計	63	29	29	31	32	32	31	32	32	32	32
正解率		0.46	0.46	0.492	0.507	0.507	0.492	0.507	0.507	0.507	0.507

表 6.15、表 6.16、6.17、6.18 より、例文の重み w_e と補正ベクトルの重み w_c の最適な組み合わせは、「$w_e = 2.0$、$w_c = 5.0$」であった。このときの正解率は 0.619 であった。しかし、一番良い正解率でも約 6 割であることから、提案手法の更なる改善が必要である。

表 6.16: 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 1.5$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	3	3	3	3	3	3
ネタ	3	1	1	1	1	1	2	2	2	2	2
カバー	2	1	1	1	1	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	2	2	2	1	1	0	0	0	0	0
サービス	3	1	1	2	2	3	2	2	2	2	2
地方	2	0	0	0	0	1	1	1	1	1	1
アルバム	2	1	1	2	2	2	2	2	2	2	2
コード	3	2	2	1	1	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	1	1	2	2	2	2	2	2	2	2
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	2	2	2	2	1	1	1	1	1
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	4	3	3	3	3	3
顔	3	1	1	1	1	2	2	2	2	2	2
系	3	1	1	1	1	1	1	1	1	1	1
郵便	2	2	2	2	1	1	1	1	1	1	1
反応	2	1	1	1	1	1	2	2	2	2	2
合計	63	29	29	32	30	37	35	35	35	35	35
正解率		0.46	0.46	0.507	0.476	0.587	0.555	0.555	0.555	0.555	0.555

表 6.17: 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 2.0$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	3	3	3	3	3	3
ネタ	3	1	1	1	1	1	2	2	2	2	2
カバー	2	1	1	1	1	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	2	2	2	1	1	0	0	0	0	0
サービス	3	1	1	1	2	3	2	2	2	2	2
地方	2	0	0	0	0	1	1	1	1	1	1
アルバム	2	1	1	2	2	2	2	2	2	2	2
コード	3	2	2	1	1	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	1	1	1	2	2	2	2	2	2	2
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	2	2	2	3	3	2	2	2	1
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	4	3	3	3	3	3
顔	3	1	1	1	1	2	2	2	2	2	2
系	3	1	1	1	1	2	1	1	1	1	1
郵便	2	1	1	1	1	1	1	1	1	1	1
反応	2	1	1	1	1	1	2	2	2	2	2
合計	63	28	28	29	30	39	37	36	36	36	35
正解率		0.444	0.444	0.46	0.476	0.619	0.587	0.571	0.571	0.571	0.555

表 6.18: 式 (4.14) の補正による語義の対応付けの正解率 ($w_e = 2.5$)

		w_c									
	クラスタ数	0.0	0.01	0.5	1.0	5.0	8.0	10.0	15.0	20.0	100.0
モデル	3	2	2	2	2	3	3	3	3	3	3
ネタ	3	1	1	1	1	1	2	2	2	2	2
カバー	2	1	1	1	1	2	2	2	2	2	2
ウイルス	2	1	1	2	2	2	2	2	2	2	2
ソース	2	1	1	1	1	1	1	1	1	1	1
肉	4	2	2	2	1	1	0	0	0	0	0
サービス	3	1	1	1	2	2	3	2	2	2	2
地方	2	0	0	0	0	1	1	1	1	1	1
アルバム	2	2	2	2	2	2	2	2	2	2	2
コード	3	2	2	1	1	2	2	2	2	2	2
自分	2	1	1	1	1	1	1	1	1	1	1
場合	2	1	1	1	1	1	1	1	1	1	1
時間	4	1	1	1	2	2	2	2	2	2	2
意味	3	1	1	1	1	1	1	1	1	1	1
電話	2	1	1	1	1	1	1	1	1	1	1
一緒	2	1	1	1	1	1	1	1	1	1	1
目	5	2	2	2	2	3	3	3	3	3	2
以前	2	1	1	1	1	1	1	1	1	1	1
代	5	3	3	3	3	4	3	3	3	3	3
顔	3	1	1	1	1	2	2	2	2	2	2
系	3	1	1	1	1	2	1	1	1	1	1
郵便	2	1	1	1	1	1	1	1	1	1	1
反応	2	1	1	1	1	1	2	2	2	2	2
合計	63	29	29	29	30	38	38	37	37	37	36
正解率		0.46	0.46	0.46	0.476	0.603	0.603	0.587	0.587	0.587	0.571

6.2.2.4　4.6 節の手法の評価

4.6 節では、小分類の語義で特徴ベクトルを作成し、用例クラスタに対し小分類の語義を割り当てた後、対応する中分類の語義に変換するという手法を提案した。

表 6.19 は小分類の語義毎に特徴ベクトルを作成したときの対応付けの結果である。なお、ここでの実験に用いたパラメータは、本節での実験結果で一番高い正解率をだした「w_e=2.0(式 (4.4))、w_c=5.0(式 (4.14))」を使用している。

表 6.19: 小分類の語義の特徴ベクトルを用いての対応付けの結果

	クラスタ数	小分類
モデル	3	3
ネタ	3	1
カバー	2	2
ウイルス	2	2
ソース	2	1
肉	4	1
サービス	3	3
地方	2	1
アルバム	2	2
コード	3	2
自分	2	1
場合	2	1
時間	4	2
意味	3	1
電話	2	1
一緒	2	1
目	5	2
以前	2	1
代	5	4
顔	3	1
系	3	2
郵便	2	1
反応	2	1
合計	63	37
正解率		0.587

小分類の語義の特徴ベクトルを用いての用例クラスタと辞書の語義の対応付けは、中分類の語義毎に特徴ベクトルを作成した場合とほとんど変わらない正解率であった。しかし、対象単語が「目」「顔」の時に、中分類と比べてそれぞれ 1 つずつ正解クラスタ数が減った。このことから、小分類を用いての対応付けは正解率向上に対する効果が薄いことがわかった。

6.2.2.5 自動作成クラスタを対象とした実験

表 6.20 は自動作成したクラスタでの対応付けの結果を示す。なお、ここでの実験に用いたクラスタリング結果は、6.2 節で記した結果で purity が一番高い「rel_coh で特徴ベクトルを組み合わせたクラスタリング」と、語義識別率・新語義識別率が一番高い「トピックベクトル」の結果を用いている。また、語義の対応付け手法のパラメタは 6.2.2.4 と同じく「w_e=2.0(式 (4.4))、w_c=5.0(式 (4.14))」を使用している。

表 6.20: 自動作成した用例クラスタと辞書の語義との対応付けの結果

	(対応付け正解数/新語義以外のクラスタの総数)					
	rel_coh			topic		
	K-means	KKZ	Top	K-means	KKZ	Top
モデル	(4/5)	(4/6)	(4/6)	(4/5)	(4/6)	(4/5)
ネタ	(0/7)	(0/8)	(0/6)	(0/7)	(0/9)	(1/9)
カバー	(4/6)	(5/6)	(6/7)	(6/7)	(5/7)	(4/6)
ウイルス	(10/10)	(7/10)	(9/10)	(10/10)	(8/10)	(9/10)
ソース	(6/6)	(5/5)	(9/9)	(6/7)	(5/5)	(5/5)
肉	(6/10)	(3/10)	(6/10)	(5/10)	(4/10)	(3/10)
サービス	(8/10)	(7/10)	(7/10)	(8/10)	(8/10)	(6/9)
地方	(1/10)	(4/10)	(3/10)	(4/10)	(4/10)	(4/10)
アルバム	(7/10)	(7/10)	(7/10)	(8/10)	(7/10)	(7/10)
コード	(7/10)	(8/10)	(8/10)	(6/10)	(8/10)	(8/10)
自分	(6/10)	(7/10)	(5/10)	(5/10)	(5/10)	(3/10)
場合	(3/10)	(6/10)	(5/10)	(4/10)	(4/10)	(4/10)
時間	(6/10)	(7/10)	(6/10)	(5/10)	(5/10)	(6/10)
意味	(7/10)	(6/10)	(8/10)	(7/10)	(5/10)	(7/10)
電話	(8/10)	(9/10)	(7/10)	(10/10)	(10/10)	(9/10)
一緒	(4/10)	(4/10)	(3/10)	(6/10)	(7/10)	(5/10)
目	(3/10)	(3/10)	(1/10)	(3/10)	(3/10)	(3/10)
以前	(0/10)	(2/10)	(2/10)	(4/10)	(2/10)	(4/10)
代	(7/10)	(10/10)	(8/10)	(7/10)	(8/10)	(7/10)
顔	(5/10)	(1/10)	(3/10)	(2/10)	(1/10)	(2/10)
系	(0/10)	(0/9)	(0/10)	(1/10)	(1/10)	(1/10)
郵便	(10/10)	(7/10)	(8/10)	(9/10)	(7/10)	(8/9)
反応	(8/9)	(6/7)	(6/8)	(8/9)	(6/7)	(8/8)
合計	(120/213)	(118/211)	(121/216)	(128/215)	(117/214)	(118/211)
正解率	0.563	0.559	0.56	0.595	0.546	0.559

表中の「正解率」の行は対応付けの正解数を、それ以外の行は「/」より左側が対応付けに正解したクラスタの数を、「/」より右側が新語義を除くクラスタの総数を表わしている。なお、表での「KKZ」は K-means+KKZ 法を、「Top」はトップダウン分割法をそれぞれ表わしている。正解率は最大で 0.595 であり、正しい用例クラスタを対象としたときよりも劣る。

6.2.3 新語義判定

ここでは、新語義の判定の結果について述べる。本項でも前項での実験と同様に、人手で作成した完全に正しい用例クラスタを用いて新語義判定手法の評価を行う。新語義判定の評価関数を以下の 3 つとする。

$$\text{精度} = \frac{\text{正しく新語義と判定されたクラスタの数}}{\text{システムが新語義と判別したクラスタの数}} \tag{6.5}$$

$$\text{再現率} = \frac{\text{正しく新語義と判定されたクラスタの数}}{\text{新語義に対するクラスタの数}} \tag{6.6}$$

$$F\text{ 値} = \frac{2 \times \text{精度} \times \text{再現率}}{\text{精度} + \text{再現率}} \tag{6.7}$$

まず、既存語義近接度を、5.1 節で述べた $K\text{-}Var$、$K\text{-}Diff$、$K\text{-}Max$ とし、5.2.2 項で述べた手法 RKD1 と手法 RKD2 を用いて新語義判定を行った。それぞれの実験結果を述べる。ここでは、前節の実験結果で一番高い正解率であった 4.5 節の手法で、パラメタを「w_e=2、w_c=5」として算出した用例クラスタと語義との類似度の値を用いる。

手法 RKD1 でのそれぞれの F 値 を以下の表 6.21 に記す。新語義判定の F 値は $K\text{-}Max$ が最も高く、0.510 であった。

表 6.21: 手法 RDK1 を用いての新語義判定の結果

用いる既存語義近接度	精度	再現率	F 値
$K\text{-}Var$	0.204	0.526	0.294
$K\text{-}Diff$	0.214	0.473	0.295
$K\text{-}Max$	0.428	0.631	0.510

次に手法 DRK2 の結果を記す。手法 DRK2 では、閾値によって結果が大きく異なる。また、既存語義近接度の種類によって最適な閾値の値も異なることも予測される。そのため、3 つの既存語義近接度のそれぞれに対し、閾値の値を変化させて F 値を算出するという実験を行った。実験結果を図 6.1、図 6.2、図 6.3 に示す。

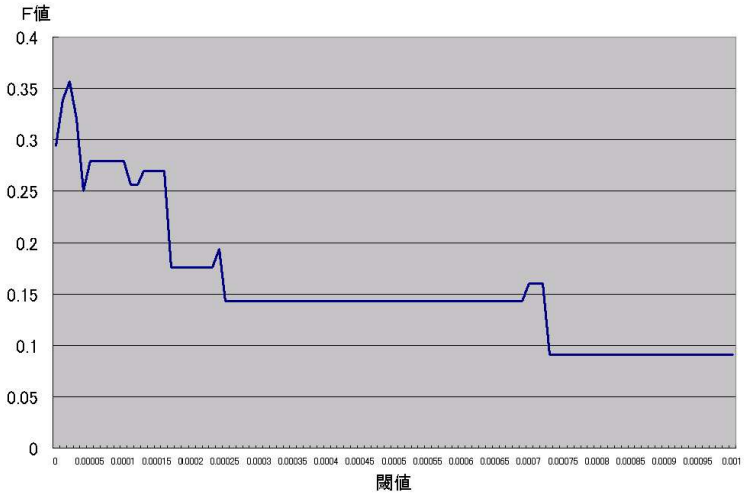

図 6.1: K-Var で閾値の値に対する F 値の変動

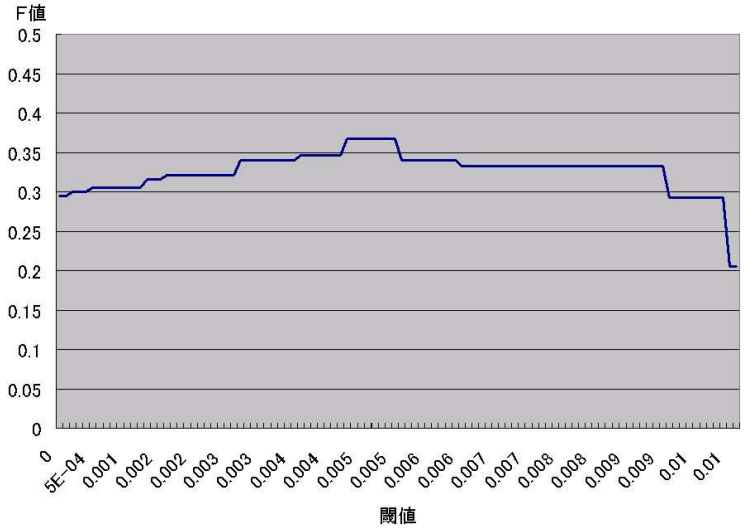

図 6.2: K-$Diff$ で閾値の値に対する F 値の変動

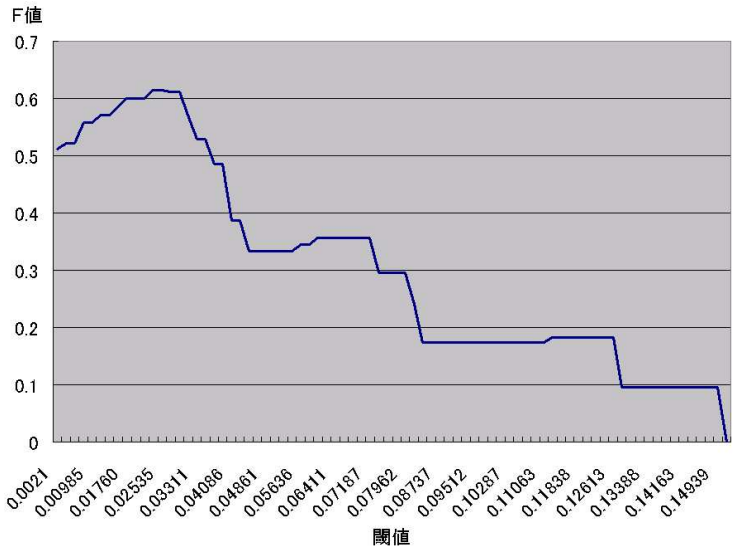

図 6.3: K-Max で閾値の値に対する F 値の変動

上記の実験で一番高いF値を算出した閾値ならびにそのときの精度、再現率、F値を表6.22にまとめる。

表 6.22: 最適な閾値を設定したときの実験結果

用いる既存語義近接度	閾値の値	精度	再現率	F値
$K\text{-}Var$	0.00002	0.270	0.526	0.357
$K\text{-}Diff$	0.0046	0.3	0.473	0.367
$K\text{-}Max$	0.025	0.6	0.631	0.615

完全に正しい用例クラスタを用いた場合、他の既存語義近接度と比べ、類似度の最大値を既存語義近接度Kとして用いた$K\text{-}Max$の場合に、良い結果が得られた。

次に、3章で述べたクラスタリング手法を用いて作成したクラスタに対して新語義判定を行う。特徴ベクトルは6.2.1項の実験でpurityの一番高かった「rel_cohで特徴ベクトルを組み合わせたクラスタリング」と語義識別率の一番高かった「トピックベクトル」とした。クラスタリング手法は、K-means法、K-means+KKZ法、トップダウン分割法の3通りを試した。新語義判定の手法はDRK2とした。既存語義近接度を$K\text{-}Var$としたときの結果を表6.23に、$K\text{-}Diff$にしたときの結果を表6.24に、$K\text{-}Max$にしたときの結果を表6.25にそれぞれ示す。いずれの場合も閾値はF値の値が一番高かった表6.22の値とした。

表 6.23: 自動作成クラスタで、$K\text{-}Var$を用いての新語義判定の結果

用いる特徴ベクトル	クラスタリング手法	精度	再現率	F値
rel_cohで特徴ベクトルを組み合わせたクラスタリング	K-means	0.111	0.631	0.188
	K-means+KKZ法	0.125	0.761	0.214
	トップダウン分割法	0.056	0.571	0.102
トピックベクトル	K-means	0.102	0.882	0.182
	K-means+KKZ法	0.104	0.777	0.184
	トップダウン分割法	0.140	0.947	0.244

自動クラスタリングのデータで実験を行った場合でも、既存語義近接度に類似度の最大値である$K\text{-}Max$を用いた場合が他と比べて若干F値が高かった。

表 6.24: 自動作成クラスタで、$K\text{-}Diff$ を用いての新語義判定の結果

用いる特徴ベクトル	クラスタリング手法	精度	再現率	F 値
rel_coh で特徴ベクトルを	K-means	0.151	0.631	0.244
組み合わせたクラスタリング	K-means+KKZ	0.131	0.619	0.216
	トップダウン分割法	0.07	0.5	0.122
トピックベクトル	K-means	0.095	0.529	0.162
	K-means+KKZ	0.104	0.555	0.175
	トップダウン分割法	0.121	0.684	0.206

表 6.25: 自動作成クラスタで、$K\text{-}Max$ を用いての新語義判定の結果

用いる特徴ベクトル	クラスタリング手法	精度	再現率	F 値
rel_coh で特徴ベクトルを	K-means	0.363	0.210	0.266
組み合わせたクラスタリング	K-means+KKZ 法	0.241	0.333	0.28
	トップダウン分割法	0.153	0.142	0.148
トピックベクトル	K-means	0.25	0.176	0.206
	K-means+KKZ 法	0.1	0.166	0.125
	トップダウン分割法	0.105	0.105	0.105

次に、既存語義への対応付けと新語義の判定がどの程度正確に行われたかを評価する。ここでは、式(6.8)に示した、Cor_{total}という評価基準で評価する。つまり、新語義に対応するクラスタに対しては新語義であると判定し、既存語義に対応するクラスタに対しては正しい辞書の語義に対応づけるほど、Cor_{total}は高くなる。

$$Cor_{total} = \frac{\text{正しい語義に対応付けられたクラスタの数} + \text{正しく新語義と判定されたクラスタの数}}{\text{クラスタの総数}} \tag{6.8}$$

ここでは、人手で作成した完全に正しい用例クラスタと3章で述べたクラスタリング手法を用いて作成した用例クラスタの2つに対して実験を行った。

まず、人手で作成した完全に正しい用例クラスタに対する実験について説明する。ここでは、前節の実験結果で一番高い正解率であった4.5節の手法で、パラメタを「w_e=2.0、w_c=5.0」として算出した用例クラスタと語義との類似度の値を用いる。新語義判定の手法はDRK2とし、いずれの場合も閾値はF値の値が一番高かった表6.22の値とした。実験結果を表6.26に示す。

表 6.26: Cor_{total}による評価（正しい用例クラスタ）

既存語義近接度	Cor_{total}
$K\text{-}Var$	0.414
$K\text{-}Diff$	0.463
$K\text{-}Max$	0.597

次に、クラスタリング手法を用いて作成した用例クラスタに対する実験について説明する。クラスタリングに用いる特徴ベクトルは、6.2.1項の実験でpurityの一番高かった「rel_cohで特徴ベクトルを組み合わせたクラスタリング」と語義識別率の一番高かった「トピックベクトル」とした。クラスタリング手法は、K-means法、K-means+KKZ法、トップダウン分割法の3通りを試した。また、新語義判定の手法は、正しい用例クラスタに対する実験と同様DRK2とし、いずれの場合も閾値はF値の値が一番高かった表6.22の値とした。

自動作成クラスタに対して、既存語義近接度$K\text{-}Var$、$K\text{-}Diff$、$K\text{-}Max$を用いた時のCor_{total}の値を表6.27に示す。

Cor_{total}の値をみると、完全に正しい用例クラスタを用いた場合、自動クラスタリングのデータを用いた場合の両方とも、既存語義近接度に$K\text{-}Max$を用いる場合が良い結果となった。

表 6.27: Cor_{total} による評価（自動作成クラスタ）

用いる特徴ベクトル	クラスタリング手法	既存語義近接度	Cor_{total}
rel_coh で特徴ベクトルを組み合わせたクラスタリング	K-means	$K\text{-}Var$	0.297
		$K\text{-}Diff$	0.400
		$K\text{-}Max$	0.525
	K-means+KKZ 法	$K\text{-}Var$	0.301
		$K\text{-}Diff$	0.349
		$K\text{-}Max$	0.491
	トップダウン分割法	$K\text{-}Var$	0.230
		$K\text{-}Diff$	0.378
		$K\text{-}Max$	0.513
トピックベクトル	K-means	$K\text{-}Var$	0.241
		$K\text{-}Diff$	0.379
		$K-Max$	0.543
	K-means+KKZ 法	$K\text{-}Var$	0.284
		$K\text{-}Diff$	0.349
		$K\text{-}Max$	0.448
	トップダウン分割法	$K\text{-}Var$	0.300
		$K\text{-}Diff$	0.334
		$K\text{-}Max$	0.473

また、クラスタリングを新語義発見に用いる有効性を測るため、1 つの用例を 1 クラスタとした時の F 値をそれぞれ求めた（表 6.28）。

表 6.28 の結果は表 6.22 の結果よりもかなり劣る。したがって、新語義発見を行う際に、個々の用例毎に新語義かどうかを判定するよりも、用例を一度クラスタリングし、クラスタ毎に判定を行うほうが有効であることが言える。

既存近接度 $K\text{-}Max$ は、表 6.25 の結果をみる限り、正しい用例クラスタを用いた場合に比べると、自動的に作成されたクラスタに対しては、F 値が他の 2 つとそれほど差はでなかった。また、1 つの用例を 1 つクラスタとみなしたときの表 6.28 の実験結果では、一番低い F 値をとっている。言い換えれば、$K\text{-}Max$ は正しい用例クラスタを対象としたでしか有効ではない。この原因の可能性を考察する。

1. 実験に用いた完全に正しい用例クラスタの中で、新語義のクラスタに含まれる用例の数が、既存語義のクラスタに含まれる用例の数と比べて少ない。

表 6.28: 1 つの用例を 1 クラスタとした時の新語義判定の評価

用いる既存語義近接度	閾値	精度	再現率	F 値
$K\text{-}Var$	0.00002	0.081	0.982	0.150
$K\text{-}Diff$	0.0046	0.118	0.75	0.204
$K\text{-}Max$	0.025	0.153	0.034	0.056

2. 1. のため、前者の特徴ベクトルが後者の特徴ベクトルのものと比べスパースになりやすい。

3. 2. のため、新語義のクラスタと辞書の語義の特徴ベクトルとの類似度が低くなりやすい。

4. 3. のため、新語義のクラスタの最大類似度は既存語義と比べ低くなる傾向が高くなる。

5. 4. のため、各クラスタを辞書の語義との最大類似度の値で降順で並べ替えた場合、新語義のクラスタが要素の最後の方となる傾向が高くなる。

6. 5. での並び方は、既存近接度 $K\text{-}Max$ にとって、効力を発揮できる環境である。

7. 6. のため、他の 2 つの既存近接度に比べ新語義判定が行いやすくなっている。

6.2.3.1　新語義判定に類似度を用いる妥当性

5 章でも述べたが、本研究では新語義判定に用例クラスタと辞書の既存の語義との類似度を用いている。既存の語義との類似度を新語義判定に用いることは、直感的には正しいと思われる。しかし、既存語義との類似度の計算がどの程度妥当であるかを調べる必要がある。そこで、用例クラスタと辞書の語義との対応付けの結果を用いて、新語義判定のために用例クラスタと辞書の既存の語義との類似度を用いることの妥当性について調査する。

具体的には、用例クラスタと辞書の語義との対応付けの結果で正解であったクラスタと不正解であったクラスタとの類似度の値を比較する。もし、類似度の値が大きければ大きいほど対応付けの正解率が高ければ、用例クラスタと既存の辞書の語義との類似度を正しく計算できており、類似度に基づく新語義判定は妥当であるという事が言える。しかし、対象単語毎に用例クラスタと辞書の語義との類似度の値にはばらつきがある。そのため、ここでは類似度の値そのものではなく、相対化した類似度（既存語義近接度を $K\text{-}Max$ としたときの RK）の値で比較する。

用例クラスタと辞書の語義との対応付けの正解率が一番高かった時の手法でパラメタ w_e=2、w_c=5 のときの結果を図 6.4 に示す。縦軸は、クラスタと辞書との類似度の値をその対象単語ごとに相対化した値 RK を表わしている。丸印が正しい語義を対応づけたクラスタを、四角印が間違った語義を対応づけたクラスタをそれぞれ表わしている。

図 6.4 から若干ではあるが、相対類似度の値が高い方が対応付けの結果で正解であり、相対類似度の値が低い方が対応付けが失敗している場合が多いように思われる。

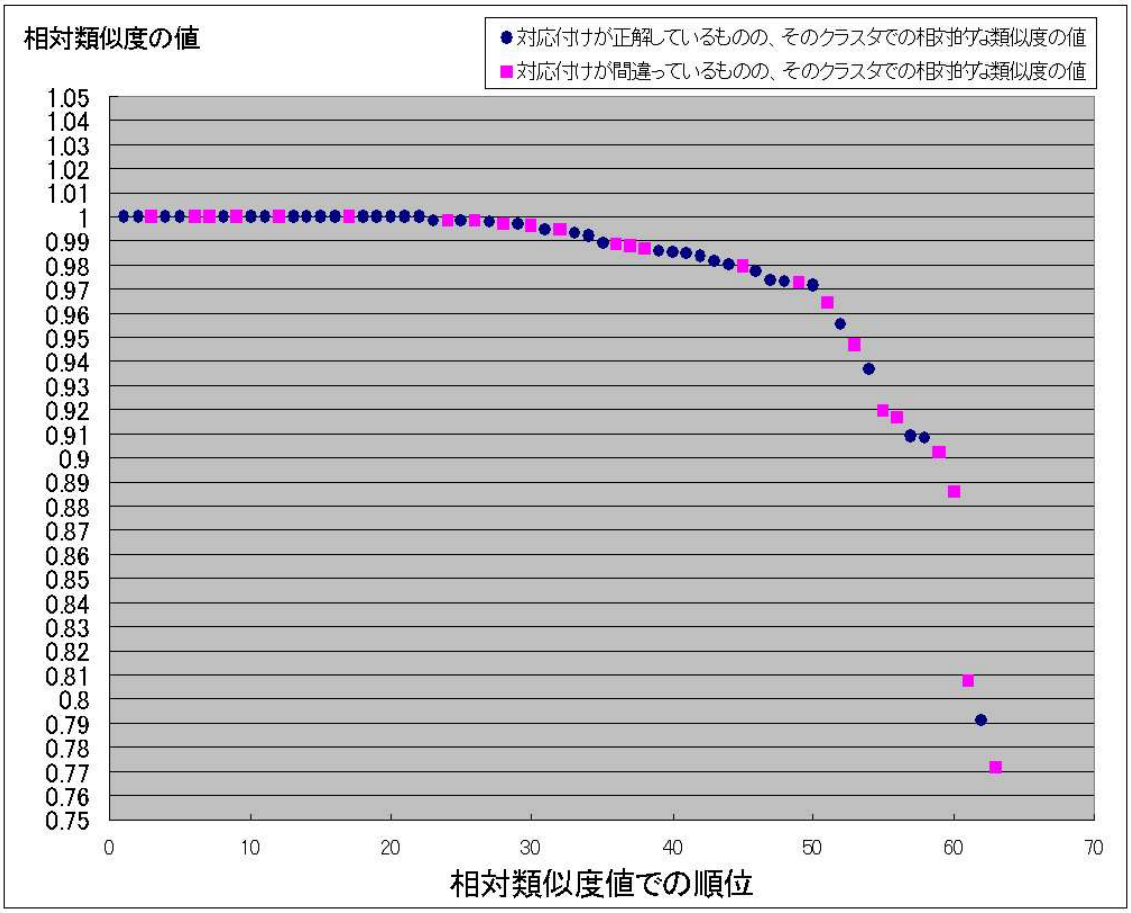

図 6.4: 類似度の値と対応付け成果との相関

第7章　おわりに

本論文では、同じ語義を持つ用例をまとめるクラスタリングの手法の改良と、コーパスから作成された用例クラスタに対する意味の解釈、すなわち用例クラスタに対応する辞書の語義を選択する手法と、辞書のどの語義にも対応しない新語義であるかを判定する手法について述べた。本章では、本研究で得られた知見と今後の課題について述べる。

7.1　まとめ

提案手法について以下にまとめる。

1. 用例クラスタリング

   九岡の手法を用いたベクトルの作成方法とクラスタリングアルゴリズムを組み合わせた用例クラスタリングの実験を行った。

   (a) ベクトルの作成方法（九岡の手法を用いる）

      i. 単独 4 種類（隣接ベクトル、LDA 拡張文脈ベクトル、連想ベクトル、トピックベクトル）

      ii. 組み合わせ 2 種類（rel_intra 、rel_coh）

   (b) クラスタリングアルゴリズム

      i. K-means

      ii. K-means+KKZ 法

      iii. トップダウン分割法

   実験の結果、最も高い purity と語義識別率が得られたのは、組合わせの評価関数に rel_coh を用い、クラスタリングアルゴリズムに K-means+KKZ 法を用いた場合であった。

一方、新語義識別率が最も高かったのは、クラスタリング手法にトップダウン分割法を用い、単独では連想ベクトルとトピックベクトル、組み合わせでは評価関数にrel_coh を用いた場合であった。

2. クラスタと辞書との対応付け

   用例クラスタと辞書定義文との特徴ベクトルを作成し、両者の対応付けを試みた。また、大きく分けて 3 つの改良案を提案した。

   (a) 特徴ベクトルの作成方法の改良

   (b) 特徴ベクトルの過疎性への対応

   (c) 小分類の語義の特徴ベクトルでの対応付け

   その結果、4.5 節の「語義の特徴ベクトルの過疎性への対応」で述べた、補正用ベクトルの加算が一番効果的であった。また、その時の実験で一番正解率が高かったのは、語義の特徴ベクトルを作成する際に、辞書の定義文と例文の両方を用い、元のベクトルの大きさを考慮した補正を加える手法である。また、その時のパラメタは「$w_e = 2.0$,$w_c = 5.0$」であった。

3. 新語義判定

   新語義を判定する指標としての $K\text{-}Var$、$K\text{-}Diff$、$K\text{-}Max$ の 3 種類の既存語義近接度を提案した。3 種類の中では、$K\text{-}Max$ が一番新語義判定に有効であった。また、$K\text{-}Max$ の閾値 T_k を 0.0025 としたとき新語義判定の F 値が最大となった。

7.2 今後の課題

本研究の今後の課題は以下の通りである。

- オープンテストで実験を行う。

  本論文におけるパラメタ、すなわち例文の重み w_e や補正の重み w_c などの調整はテストデータを用いている。そのため本論文の実験はクローズドテストであるといえる。

- 対応付けにおける正解率のさらなる向上を目指す。

  共起行列 A の作成方法を変更することによって正解率の向上が図れると思われる。現在、共起行列における辞書側の単語は、岩波国語辞典に予めタグ付けされた形態

素解析結果を用いているが、茶筌での形態素解析結果を用いることが改良案として挙げられる。また、辞書の語義に関する情報を拡張するため以下の3つの方法が考えられる。

  - シソーラスから得られる、その語義の上位語を用いる
  - 辞書における参照見出しを用いて、参照先の見出し語の定義文の情報を加えて特徴ベクトルを作成する
  - インターネットのキーワード検索により得られる関連語を利用する。例えば、ウェブの検索結果から類似する単語の集合を作成する試みが報告されている[11]。

- 新語義に対する意味の識別

  新語義と判定された用例クラスタが複数ある場合には、それらが同じ意味なのかそれとも別々の意味なのかの判定を行う。

- クラスタリングの改良

  例えば、人手でクラスタリングの制約を加えた半教師ありの手法を用いる。

謝辞

本研究を進めるにあたり, 丁寧な御指導、御教授を賜りました白井清昭准教授, 島津明教授, 中村誠助教,Nguyen Minh Le 助教に心より深く感謝致します。また, 白井研究室・島津研究室の皆様方には, 本研究に関する貴重なご支援をいただきました。この場を借りて感謝申し上げます。

参考文献

[1] Stefan Bordag. Word Sense induction: Triplet-based clustering and automatic evaluation. In Proceedings of the EACL, pp.137-144,2006.

[2] Fumiyo Fukumoto and Jun'ichi Tsujii. Automatic recognition of verbal polysemy.In Proceedings of the COLING, pp.762-768,1994.

[3] Dan Klein,Sepandar D.Kamvar,Christopher D.Manning. From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. Proceedings of the 19 International Conference on Machine Learing, pp.307-314,2002

[4] Kiri Wagstaff,Claire Claire,Seth Rogers,Stefan Schroedl. Constrained K-means Clustering with Background Knowledge. Proceedings of the Eighteenth International Conference on Machine Learing, pp.557-584,2001.

[5] 新納浩幸, 佐々木稔, 村上司. 制約を修正に用いた半教師有りクラスタリング. 情報論的学習論ワークショップ,2006.

[6] Hinrich Schütze. Automatic Word Sense Discrimination. Computational Linguistics, Vol.24,No.1,pp.97-123,1998.

[7] 九岡佑介, 白井清昭. コーパスからの単語の意味の発見. 修士論文. 北陸先端科学技術大学院大学 情報科学研究科 2008

[8] 菊田篤史, 白井清昭. 未定義語の判別を含む語義曖昧性解消. 修士論文. 北陸先端科学技術大学院大学 情報科学研究科 2006

[9] Katsavounidis I.,C.Kuo and Z.Zhang. A New Initialization Technique for Generalized Lioyd Interation.IEEE Singnal Proc.Lettrts 1(10),pp.144-146,1994

[10] 西尾実, 岩淵悦太郎, 水谷静夫. 岩波国語辞典 第五版. 岩波書店,1994

[11] Yutaka Matuo,Takeshi Sakaki,Koki Uchiyama,Mituru Ishizuka. Graph-based Word Clustering using a Web Search Engine, Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP2006), pp.542-550,2006

付 録A　実験に用いた対象語とその語義

実験に用いた23個の対象語とその語義を挙げる。なお、既存語義に対しては岩波国語辞典の語釈文を、新語義に対しては「(新語義)」というマークをつけた上でその意味の説明を掲載した。

- モデル
  s1：型。模型。▽↓もけい
  s2：手本。模範。「これを*モデル*にしてやれば間違いない」
  s3：美術製作の対象となるもの・人。文学作品の人物の素材となる人。
  s4：「ファッション　モデル」の略。流行の服装をして、客に見せたり写真に撮らせたりするのが職業の（女の）人。▽ｍｏｄｅｌ
  a（新語義）：機械・自動車などの型式
  b（新語義）：事象を定式化したもの

- ねた
  s1：新聞記事などの材料。
  s2：手品の仕掛け。
  s3：証拠。「*ねた*があがる」
  s4：（料理の材料としての）食物。「すしの*ねた*」▽「たね」をさかさまにした隠語から。
  a（新語義）：物語などの核心
  b（新語義）：うそ

- カバー
  s1：名他の物を覆うのに使うもの。覆い。ふた。書物の表紙の上にかけるもの。「*カバー*ガール」（雑誌の表紙をかざる女性モデル）靴や靴下の上から覆ってはくもの。
  s2：足りないところを補うこと。損失や不足の補い。「赤字分を*カバー*する」野球で、野手の守備動作を他の野手が援護すること。▽ｃｏｖｅｒ
  a（新語義）：カバーバージョン。楽曲を別の人が演奏・または歌ったりすること
  b（新語義）：覆う行為

- ウイルス
  s1：細菌より小さく、光学顕微鏡では見えない、一群の微生物。特定の細胞に依存して増殖する。狂犬病・天然痘・小児麻痺・モザイク病の病原体など、種類が多い。濾過性病原体。ビールス。ヴィールス。ヴァイラス。▽ラテンｖｉｒｕｓ
  s2：電子計算機で、ソフトウェアにひそかに仕掛けて、正常な働きをさせなくしたり格納データを消し去ったりする、悪性のソフトウェア。▽(1)の比ゆで、「感染」「潜伏」「発病」「増殖」などの語も、(1)と類比的に使う。ウイルス(2)を見つけ出して取り除くソフトウェアを「ワクチン」と言う。

- ソース
  s1：西洋料理に調味料として使う汁。種類が多い。「*ホワイトソース*」「*ウースターソース*」▽ｓａｕｃｅ
  s2：出どころ。源泉。「*ニュースソース*」▽ｓｏｕｒｃｅ
  a（新語義）：ソースコード

- 肉
  s1：動物のからだの、皮膚におおわれ、骨格を包む、やわらかな物質。「肉がつく」「筋肉・骨肉・髀肉・血肉・肉片・肉塊・肉質」食用にする鳥獣のにく。「肉を食う」「肉類・魚肉・獣肉・鶏肉・牛肉・酒池肉林・肉食・肉汁・肉牛」
  s2：精神に対する、人間の物質的要素。からだ。「血わき肉おどる」「肉体・肉感・肉欲・肉弾」
  s3：血縁で最も近い関係にある。「肉親」非常に近い。「肉薄」機械・機具を使わない。直接。「肉眼・肉声・肉筆」
  s4：果実の皮と種子との間にあるやわらかい部分。み。「果肉・竜眼肉」
  s5：「印肉」の略。「朱肉・肉池」
  s6：ふとりかげん。あつみ。ふとさ。「肉の厚い葉」「肉づき・中肉中背」

- サービス
  s1：客に対するもてなし。接待。優遇。「―のよい旅館」
  s2：商売で、値引きしたり客の便宜を図ったりすること。「百円―しておきます」「アフター*サービス*」
  s3：奉仕。「家庭*サービス*」
  b（新語義）：情報関連。サーバが提供するもの

- 地方
  s1：（国内の）ある一定の地域。
  s2：首府以外の地域。⇔中央。「*地方*機関」
  s3：旧軍隊で、軍以外の一般社会。「*地方*人の考え方が抜けない」

- アルバム
  s1：写真帳。記念帳。「卒業*アルバム*」
  s2：（特定のテーマにより）いくつかの曲を収めたＬＰレコードやＣＤなど。▽ａｌｂｕｍ

- コード
  s1：ゴムなどで絶縁した電線。主に室内用。▽ｃｏｒｄ
  s2：符号。「*コード*ブック」こういうことをしないようにと定めた準則。倫理規定。「プレス―」▽ｃｏｄｅ（＝法典。おきて）
  s3：楽器の弦。和音。▽ｃｈｏｒｄ

- 自分
  s1：その人自身。「―のことは―でせよ」「―で言ったくせに」「太郎は弟に―の（＝太郎の）荷物を持たせた」
  s2：《代名詞的に》わたくし。「―は二等兵であります」＜関連＞己・みずから・私・我・余・自家・自我・自己・自身・当人・本人
  a（新語義）：自分の体

- 場合
  s1：物事の、その時に応じて分けて考えられる状態・事情。「―によっては」「万一の―」「問

題を―分けして扱う」
s2：とき。おり。「雨が降った―（には）中止する」▽概して、時（4）と同様に使う。法令文で、「場合」と「とき」とを重ねて用いるときは、前提とする条件の大きい方に「場合」を使うのが慣用。
a（新語義）：事例
b（新語義）：可能性

- 時間
s1：ある時刻と他の時刻との間（の長さ）。時のへだたりの量。「この仕事は―がかかる」「―が切れる」「―の問題」（その物事の結着まで長くはかかるまいという情況にまで立ち至ったこと）▽多くは一日より短い場合について使う。
s2：時間（1）の長さの単位。一時間は六十分。一日は二十四時間。
s3：空間と共に、物体界を成り立たせる基礎形式と考えるもの。普通、過去から未来に絶えず移り流れると考えている。▽↓とき（時）（1）。
s4：日常語で、時刻。「帰りの―がおそい」「お―（＝刻限）です」「日本では昼ですが現地―では午後十時です」＜関連＞時・時刻・タイム・年月・歳月・月日・日月・時日・光陰・星霜・春秋・多年・積年・短時日・片時・半時・一時・一刻・一瞬・瞬間・瞬時・刹那・寸陰・寸時・つかの間・たまゆら

- 意味
s1：その言葉の表す内容。意義。「辞書を引けば*意味*がわかる」
s2：表現や行為の意図・動機。「どういう*意味*でそんなことをしたのか」
s3：表現や行為のもつ価値。意義。「そんな事をしても*意味*がない」

- 電話
s1：電話機による通話。「*電話*をかける」「*電話*を切る」
s2：「電話機」の略。「*電話*ボックス」
a（新語義）：電話番号、電話回線

- 一緒
s1：ほかのもの・ことと併せて一つにする、または一つになるさま。「これも*一緒*に処理する」「君と*一緒*に行こう」「*一緒*になる」（一つに合わさる。特に、夫婦になる）
s2：同じであること。「趣味が*一緒*だ」▽正しくは「一所」だといわれる。

- 目
s1：生物の、物を見る働きをする器官。また、その様子・働き。▽「眼」とも書く。 眼球・視神経から成る器官。「―を皿のようにして見る」「鵜の―鷹の―」（熱心に捜し求める目つき・態度）「―を開く」（自覚したり、知識を得たりする意にも）「―をつぶる」（見て見ないふりをする、また、死ぬ意にも）「―がつぶれる」（視力を失う）「―がさえる」（眠るべき時に、眠くならない。また、物事の本質を見抜く力を発揮する。↓（ウ））「―がすわる」（怒りや酔いで、じっと一点を見詰め、目の玉を動かさない）「―が回る」（めまいがする。転じて、非常に忙しい）「―から鼻へ抜ける」（抜け目がなくすばやい、または知恵が速く働くことの形容）「―と鼻の間」（非常に近いことのたとえ）「―と鼻の先」（同上）「―が真っ暗になる」（突然の不幸などで、希望を失ったり手立てが見えず、どうしたらよいか分からない状態に陥る）「―に見えて」（見てわかるほどはっきり）「―に物を見せる」（思い知るようにしかける）「―を光らす」（よく注意する）「―を凝らす」「―を丸くする」（びっくりする）「―を剥く」（怒ったり驚いたりして目を大きく開く）「―を喜ばせる」（美しい物などを見

て喜び楽しむ)「―の保養」(同上)「―の正月」(同上)「―を細める」(喜ぶ意にも)「酒に―が無い(＝すっかり心を奪われる)」目(1)(ア)の様子。目つき。「―は口ほどに物を言う」「はたから変な―で見られる」「色っぽい―をする」見ること。見えること。また、視力。更に、注意(力)。洞察力。「―に触れる」「お―に掛ける」(お見せする)「お―に掛かる」(お会いする)「―を掛ける」(転じて、面倒を見て引き立ててやる)「―がくらむ」(めまいがする。また、心が奪われて正しく判断できなくなる)「―をさます」(迷いから脱却する意にも)「―に障る」「―のかたき」(目ざわりで憎くて仕方ないもの)「―に立つ」「―につく」「―にする」「―もあてられない」(あまりひどくて見ていられない)「―もくれない」(無視する)「長い―で見る」(現状だけで判断せず、将来を期待して気長に見る)「―を奪う」(すばらしさなどで見とれさせる)「―を引く」「―をつける」「―をとめる」「―を配る」「―が届く」(注意・監視がそこまで及ぶ)「―を離す」「―を盗む」(人に気づかれないように何かをする)「―をくらます」(ごまかして気づかれないようにする)「―を疑う」(意外な事に接し、これが事実なのか、見そこないではないかと驚く)「―を通す」(一通り見る。ざっと読む)「―が近い」(近眼)「―がいい」「―が高い」(ものを識別する力がある)「―が肥えている」(同上)「―を利かす」
s2：目(1)(ア)に見える姿・様子。「*見た目*ばかりよくて実質は悪い」
s3：ある物事に出会うこと。経験。体験。また、局面。「ひどい*目*を見る」「落第の*憂き目*」
s4：形が目(1)(ア)に似ているもの。「台風の*目*」「*うおのめ*」縦横に(線状に)並び交わったものによって囲まれてできた、すき間。「*目*粗い布地」「網の*目*」「碁盤の*目*」「*目*が詰んでいる」縦に並んだものの、すき間。「のこぎりの*め*」「櫛の*め*」さいころの面につけた点状のしるし。「六の*目*」「*目*が出る」(思いどおりの目が現れる。転じて、運がついて事がうまく運ぶ)量を示すしるしの刻み。目盛り。「秤の*め*」
s5：秤で量った重さ。「*め*がかかる」(相当の重みがある)「*目*減り」。また、「匁」の別称。「一貫二〇〇*目*」
s6：木材の表面にあらわれた模様。木目。「*目*が細かい板」「柾*目*」
s7：順序を表す時に添える語。「三番*目*の問題」「二つ*目*の角を曲がる」「十軒*目*」
s8：点または線のようになって、他と区別される状態になった所。「結び*目*」「割れ*目*」「境*目*」。物事の境となるような情況。「季節の変わり*目*」「彼の勢力も落ち*め*になる」「勝ち*め*がない」(勝つ見込みがない)
s9：《量・程度に関係する形容詞の語幹などに付いて》どちらかと言えば、その性質を持つ意を表す。「話を*短め*に打ち切る」「大き*め*な方を選ぶ」「細*め*の糸」「幾分早*め*に出掛ける」「控え*め*な人」

- 以前
s1：それから前。⇔以後。
s2：ずっと前(の時)。「彼は*以前*横浜に住んでいた」

- 代
s1：かわりになる。他にかわって仕事をする。「代理・代用・代作・代官・代診・代役・代弁・代数・代議士・代言人・名代・城代・所司代」
s2：用をするためのもの。「舌代・糊代」造・名かわりになるもの。商品や労力などを得たかわりに与えるもの。「お代を払う」「代価・代金・代償・地代・間代・足代・身代金」
s3：造・名家督または統治権を相続して当主である間。「代がかわる」「先祖代代・譜代・歴代・末代・初代・聖代・累代・世代」
s4：歴史上の長い時間の区分。「時代・上代・古代・近代・現代・当代」地質時代の最も大

きい区分。「古生代・新生代」
s5：年齢の範囲を示す語。「台」に同じ。「十代の少年」

- 顔
s1：頭部の前面。目や鼻や口がある所。「*顔*を洗う」「*顔*から火が出る」（非常に恥ずかしい思いをする）「*顔*にもみじを散らす」（婦人などが恥ずかしがって赤面した美しい様子の形容）「（人の）*顔*に泥を塗る」（他人の体面を傷つける。↓（3）（イ））「*顔*を合わせる」（会う。対面する）「*顔*がそろう」（一同が皆集まる）「大きな*顔*をする」（いばる）▽↓つら（面）s2：顔（1）の様子。顔つき。表情。「変な*顔*をする」容貌。「きれいな*顔*」《接尾語的に》…の様子。「得意*顔*」「*知らん顔*」「わけ知り*顔*」▽人の態度に言う。
s3：顔（1）は人を見分ける目立つ部分だから、次のようにも使う。人によく知られていること。「*顔*が広い」「彼はこの辺で*顔*だ」（顔を見せるだけでも、様様の便宜をはかってもらえるほどだ）「*顔*が売れる」（有名だ）「*顔*を利かす」（有名さや勢力を利用して特別扱いをさせる）面目。体面。「*顔*が立つ」「*顔*をつぶす」『*顔*がさす』人に見られては具合の悪い状態になる。＜関連＞おもて・つら・温顔・紅顔・厚顔・尊顔・童顔・笑顔・幼な顔・素顔・泣き顔・似顔・寝顔・真顔・横顔・赤ら顔・瓜ざね顔・恵比須顔・心得顔・したり顔・手柄顔・我が物顔・顔面・満面・面長・細面・顔立ち・しかめっ面・ふくれっ面・仏頂面・吠え面・横っ面・相貌・容貌・容色・美貌・人相・面相・形相・相好・プロフィール

- 系
s1：いとすじ。すじ。ひも。つながり。一つづきの関係をなすもの。「系図・系統・系譜・系列・世系・大系・体系・家系・直系・母系・傍系・太陽系」
s2：体系（としてとらえられるもの）。システム。「公理系・測定系・力学系・神経系・人間機械系」
s3：名〔数学〕ある定理から直ちに導かれる他の命題。▽ｃｏｒｏｌｌａｒｙの訳語。
s4：系統だった分類。またその部門。「文学系・結晶系・六方晶系」
s5：地質の時代区分に対応する地層。「ジュラ系・三畳系・第三紀系」

- 郵便
s1：書状・はがき・小包などを集配・送達する通信制度。また、それによって送られる書状・はがき等。「*郵便*が届かない」
s2：郵便局で取り扱うものに冠する語。「*郵便*貯金」「*郵便*年金」
a（新語義）：郵便局

- 反応
s1：生体が、刺激を受けた結果として起こす変化・活動など。「薬物*反応*」「*反応*がない」（手ごたえがない意にも）「冷たい*反応*」
s2：二種以上の物質の間に起こる化学的変化。「可逆*反応*」
a（新語義）：非生物による反応