修 士 論 文

複数の特徴ベクトルを同時に考慮した語義識別

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

中西 隆一郎

2011 年 3 月

修 士 論 文

複数の特徴ベクトルを同時に考慮した語義識別

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　鶴岡 慶雅 准教授

北陸先端科学技術大学院大学
情報科学研究科情報科学専攻

0910041 中西 隆一郎

提出年月: 2011 年 2 月

概 要

本論文では、コーパスから新語義を発見する事を目標とし、そのための重要な要素技術である用例クラスタリング手法の新しい手法を提案する。一般に、語義の同一性は様々な観点から確認できる。九岡の研究では、対象となる用例を複数の特徴ベクトルで表現し、特徴ベクトルごとにクラスタリングを行い、最良のクラスタ集合を 1 つ選択する。これは単語によって単語の意味を特徴づけやすい観点が異なることに注目している。しかし、語義によっても特徴づけやすい観点が異なる。そのため、クラスタリングの行程において、複数の観点から語義の類似性を測ることで用例クラスタリングの性能の向上を狙う。

本研究で用いる用例の特徴ベクトルは、九岡の用いた隣接ベクトル, 文脈ベクトル, 連想ベクトル, トピックベクトルとほぼ同じものを用いた。ただし、本研究では隣接ベクトルを前後 2 語を素性とするように改良している。本研究では凝集型クラスタリングによって用例クラスタリングを行う。ただし、クラスタ間の類似度はそれぞれ 4 つの特徴ベクトルで計算されるコサイン類似度のうち最大のものと定義する。これは、4 つの特徴ベクトルのうちどれか 1 つでも類似度が高い場合、用例は同じ語義を持つという考えに基づく。また、特徴ベクトルによって類似度の平均値に大きなばらつきが生じていた。このような状況では、選択される特徴ベクトルに偏りができる。そこで、ベクトル間の類似度を正規化する 2 つの手法を提案し、特徴ベクトルの類似度を公平に比較できるように工夫した。さらに、複数の特徴ベクトルを同時に用いる際、生成されたクラスタがどのような観点で同一と認められたのかを把握するために、1 つのクラスタに複数の観点で併合された用例が混在しないという制約を設けた。

クラスタリングの結果を評価したところ、提案手法は九岡の手法よりも高い評価値を得たが、隣接ベクトルのみでクラスタリングを行ったものが全体での評価値が最も高かった。しかし、隣接ベクトルのみを用いる手法は、1 要素で構成されるクラスタを多く生成する。このようなクラスタは語義の判別には不向きである。そこで、2 つ以上の要素を含むクラスタについて、同じ語義を持つ用例が占める割合を調べたところ、提案手法は隣接ベクトルのみを用いる手法と比べてその割合が大きかった。また、類似度の正規化を行うことで、Purity などの評価値が向上した。以上の結果から、複数の特徴ベクトルを同時に考慮すること、その際に特徴ベクトルの類似度を正規化することが用例クラスタリングの性能の向上に有効であることがわかった。

目 次

図 目 次

表 目 次

第1章　はじめに

1.1　研究の背景

特定の文脈に出現する単語の語義を識別する語義曖昧性解消 (Word Sense Disambiguation;WSD) は、自然言語処理技術において重要な基礎技術の一つである。通常の語義曖昧性解消は、対象の単語に対して岩波国語辞典など既存の辞書に掲載されている語義の中から正しい語義を選択するが、単語が辞書に掲載されていない新しい意味として運用されている場合には、対象単語の正しい語義を選択することができないという問題がある。単語に新しい意味 (新語義) が生まれた場合にはその語義を辞書に追加する必要があるが、これを人手で行うためには、持続的なメンテナンスのコストが高いことと、新語義を網羅的に発見することが困難であるという問題がある。したがって、コーパスから新語義を自動的に発見することが出来れば、辞書の効率的かつ効果的な管理に貢献することが可能である。

新語義を判定する手法として九岡・田中の研究がある [11][12]。この手法は、用例のクラスタリングを行い、新語義の判定を行うものである。用例のクラスタリングとは、コーパスから対象単語の用例を抽出し、用例の集合に対してクラスタリングを行い、同じ語義を持つ用例をまとめたクラスタを作成する。さらに、作成されたクラスタと既存の語義との類似度を計算し、どの語義にも類似していないクラスタを新語義とみなす。上記処理のうち、本研究では用例のクラスタリングに着目する。

1.2　研究の目的

本研究の目的は、先に述べた新語義判定の手法の内、用例のクラスタリング手法を改良することにある。用例のクラスタリングは語義推定 (Word Sense Induction) あるいは語義識別 (Word Sense Discrimination) というタスクとみなせる。これは辞書を用いずに単語の意味を識別する技術で、辞書の情報に依存しないため、新語義発見のためには必要な技術である。先行研究の多くは、用例のクラスタリングをする際に、用例を一種類の特徴

ベクトルで表現し、ベクトル間の類似度をもとに語義の類似性を測る。しかし、語義の類似性は様々な観点から認められるものである。

例として「サービス」という単語について考える。

1. 前後の語から同じ意味と判断できるもの
   (a) あとのぶんは*サービス*残業...
   (b) いわゆる「*サービス*残業。...
2. 周辺文脈から同じ意味と判断できるもの
   (a) ケーキとシャンパンの*サービス*...
   (b) 値段と味と*サービス*のバランスが...
3. 特定のトピックの文書に出現することで同じ意味と判断できるもの
   (a) Apache*サービス*をインストール...
   (b) オラクルの*サービス*再起動方法...

1 の (a)(b) での「サービス」は、岩波国語辞典において、「奉仕」といった語義である。これは、「サービス」の後に「残業」という単語が出現していることから、つまり前後の単語から語義の同一性が認められる。一方、2 の (a)(b) は「客に対するもてなし、接客」という語義を持つ。この場合「サービス」の周辺には食べ物に関する表記があることから、周辺単語から語義の同一性が認められる、3 の (a)(b) は「Apatch」や「オラクル」などサーバに関する記述があり、コンピュータ関連のテキストに出現することから、同じ意味をもつものとわかる。つまり、文書のトピックより語義の同一性が認められる。なお、ここでの「サービス」とは計算機サーバの提供する「サービス」を指す語義であるが、この語義は岩波国語辞典には掲載されていない。つまり、この用法は新語義と認められる。このように、「サービス」の用例を調べると、それぞれの語義を特徴づける観点は異なる。このような結果は他の語についても同様に考えられる。

九岡らはインスタンスを複数の観点で特徴付けてクラスタリングを行う手法を提案している [11]。この研究では単語のインスタンスを 4 つの特徴ベクトルで表現しており、各特徴ベクトルを用いて合計 4 回クラスタリングを行う。そして、4 つのクラスタ集合から、最良のクラスタ集合を一つだけ選択する手法を採用している。これは単語ごとに語義識別に有効な特徴ベクトルが異なるという考えに基づいてはいる。しかしながら、先のサービ

スのように語義によっても特徴づけられやすい観点が異なる場面がある。したがって、単語のインスタンスをクラスタリングする際に、複数の観点を同時に考慮しながらクラスタリングを行うことで、クラスタリングの精度の向上が期待できる。

1.3　本論文の構成

本論文の構成は以下のとおりである。2 章ではクラスタリングや語義識別に関する関連研究を示し、本研究との差異について述べる。3 章では用例に対する特徴ベクトルの作成と、クラスタリング手法について述べる。4 章では提案手法を用いて用例をクラスタリングする実験を行い、評価と考察を行う。5 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

本章では、本論文の関連研究について述べる。また、本論文との違いについて論じる。

2.1　語義識別

語義識別 (Word Sense Discrimination) とは、岩波国語辞典のような既存の辞書を用いずに単語の意味を識別するタスクを指し、辞書を用いないことから新語義発見のために必要な技術である。このようなタスクは語義推定 (Word Sense Induction) と呼ばれることもある。語義識別の手法は、グラフに基づくものとクラスタリングに基づくもの、2つの手法に大別できる。

2.1.1　グラフに基づく手法

まず1つめの例としてグラフに基づく手法を示す。グラフベースの語義識別とは、周辺に出現する語をノードとするグラフを作成し、2単語の共起の強さを重みとする。その後、グラフを密なサブグラフに分割し、周辺語のグループが1つの語義に対応しているとみなす手法を指す。Agirre らは、HyperLex[9] と呼ばれる手法を拡張した語義識別の手法を提案している [1]。HyperLex は前述のような周辺に出現する単語をノードとし、互いの関連性を表すグラフを作成する。次に、周辺の単語との結びつきが強いハブと呼ばれるノードを見つけ、グラフをハブを中心としたサブグラフに分割する。分割されたサブグラフが語義の1つに対応する。より正確には、サブグラフに含まれる単語がある語義の周辺に出現しやすい単語として認識される。Agirre らはグラフからハブを発見する際に、HyperLex と PageRank[10] という2つ手法を実験的に比較した。さらに、それぞれの手法によるパラメータの最適化を試みている。また、Agirre らは、推定された語義 (サブグラフの1つを指す) と辞書の語義を対応付けることで、提案システムを語義曖昧性解消 (Word Sense Disambiguation:WSD) のタスクに適用している。彼らの WSD システムを Senseval-3 all word task のデータで評価したところ、同タスクの教師あり学習に基づく上位の参加システムと同等の精度が得られたと報告している。

2.1.2　クラスタリングに基づく手法

語義識別に関する手法として、グラフに基づく手法とは別にクラスタリングに基づく手法がある。これは、コーパスから対象となる単語のインスタンス (用例) を収集し、クラスタリングの手法を用いて同じ意味をまとめたクラスタ集合を作成する手法である。個々のクラスタが語の 1 つの意味に対応するとみなすことで語義を識別する。まずは代表的なクラスタリングアルゴリズムについて紹介する。

- 凝集型クラスタリング

  凝集型クラスタリングは、1 クラスタ 1 要素を初期状態とし、すべての組のクラスタに対して類似度を比較する。その中で類似度が最大となったクラスタの組を 1 つのクラスタとしてマージ (併合) する事を繰り返し、クラスタ集合を作成するといったクラスタリング方法である。

  具体的な手法を以下に示す。

  1. 1 要素 1 クラスタを初期状態とする。
  2. すべてのクラスタの組に対して、類似度の計算を行う。
  3. 類似度が最大となったクラスタの組を、1 つのクラスタにマージする。
  4. 停止条件を閾値を満たすまで 2,3 を繰り返す。停止条件は、マージ回数、クラスタの数、マージする際のクラスタ間の類似度、などによって設定される。

  1 要素 1 クラスタの初期状態から類似度の高いものをマージしていく手法であるため、クラスタ内の要素が類似した密なクラスタが生成されやすい。

- 分割型クラスタリング

  分割型クラスタリングとは、あらかじめ、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、各データに対してクラスタへの再割り当てを繰り返す手法を指す。データの再割り当ては、重心との類似度が最大となるようにクラスタのデータを割り当てなおすことによって行う。

  分割型クラスタリングの例として、k-means 法の具体的な手順を以下に示す。

  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. 各データが属してい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、先にも述べたように初期状態の作成はランダムに行われる。したがって、同じデータに対して同じクラスタ集合が常に得られるわけではない。

なお、凝集型クラスタリング、分割型クラスタリングの両者に言えることであるが、あらかじめ設定する停止条件や k-means 法における k の値などによって得られるクラスタ集合も変化する。したがって、正しいクラスタを作成するためには停止条件の際に用いるクラスタ間の類似度の閾値や k の値についての最適化が必要である。

クラスタリングに基づく手法を用いた語義識別の手法として、Schütze と九岡の 2 つの例を挙げる。

まず、Schütze の手法 [3] は、コーパスから単語の共起行列を学習し、それを基にして対象語と他の語との二次共起 (間接共起) の情報を用いた特徴ベクトルを作成し、階層的凝集型クラスタリングと正規混合分布についての EM アルゴリズムを組み合わせた Buckshot とよばれるアルゴリズムでクラスタリングを行っている。

九岡の手法 [11] は、用例を複数の特徴ベクトルで表現し、語義識別を行う。本研究で用いるベクトルは九岡の用いたベクトルと基本的に同一のものを用いる。九岡は、前後の要素、周辺の文脈、文書のトピックといったものに着目し、合計 4 つの特徴ベクトルを作成している (詳しくは 3.1 節にて述べる)。そして、作成されたベクトルごとにクラスタリングを行う。その後、4 つの特徴ベクトルで生成されるクラスタ集合の中から最良のクラスタ集合を 1 つ選択し、それを対象単語の用例におけるクラスタリングの結果とする手法である。九岡が用いたクラスタ集合の良さを評価する指標 rel_coh の求め方は以下の通りである。

まず、相対的クラスタ内類似度 rel_intra を求める。相対的クラスタ内類似度とはクラスタリングの結果 (C) に対して、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。

$$rel\_intra(C) = \sum_{\pi_j \in C} \frac{1}{N_j} \sum_{\vec{v_i} \in \pi_j} \frac{sim(\vec{g_j}, \vec{v_i})}{\max_{\vec{v_i}} sim(\vec{g_j}, \vec{v_i})} \tag{2.1}$$

なお、π_j は j 番目のクラスタを、$\vec{g_j}$ は π_i 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している。

また、クラスタの重心が互いに近いほど高い値をとる指標として、相対的クラスタ間類似度 *rel_inter* を定義する。

$$rel\_inter(C) = \sum_{\pi_j \in C} \frac{sim(\vec{G}, \vec{g_j})}{\max_{\vec{g_j}} sim(\vec{G}, \vec{g_j})} \tag{2.2}$$

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、G は $\vec{g_j}$ の各 π_j についての平均をそれぞれ表している。

式 (2.1)(2.2) から *rel_coh* は式 (2.3) で求められる。

$$rel\_coh(C) = \frac{rel\_intra(C)}{rel\_inter(C)} \tag{2.3}$$

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。

2.2 新語義の発見に関する手法

1 章でも述べたが、語義が日々変化していたとしても、新しい語義を自動的に抽出することができれば、辞書の作成や編集および管理に対して大きく貢献できる。しかし、新語義や希少語義と既存の語義との区別は一般に難しい。新語義発見の先行研究として、Richard らによる新語義の分類手法 [4]、田中の新語義発見手法 [12] の 2 つについて紹介する。

まず、Richard らの研究では複数の言語で記載されたパラレルコーパスを対象に、対象単語と対訳との共起ベクトルを作成し、k-means 法を用いてクラスタリングを行う。クラスタリングの結果から、クラスタ内の用例が持つ意味が一般的な語義かそうでない希少語義や新語義であるかの区別としている。しかし、この手法では、作成されたクラスタが既存のどの語義に該当するかまでは判定していない。

田中の新語義発見手法 [12] は、まず九岡の用いた特徴ベクトルと同じものを用いてクラスタリングを行う。作成されたクラスタと辞書に定義されている語義との類似度を求め、対象のクラスタがどのような語義に該当するのかを識別する手法である。既存語義と新語義との区別を行うに際して、田中はクラスタと辞書の語義 (既存語義) の集合との類似度を既存語義近接度と表している。この値は既存の語義に類似していれば類似しているほど、値が大きくなる指標である。クラスタを既存語義近接度を降順にソートし、クラスタ同士の既存語義近接度の差が相対的に大きな箇所を既存語義と新語義の境界とすることで、新語義の検出を行う。

2.3　本研究との関連

本論文も新語義の発見が最終的な目標である。Richard や田中の新語義発見の手法は、どちらも用例のクラスタリングを行い、得られた結果に対して新語義かどうかの判定を行っている。本研究では、上記の処理のうち用例のクラスタリングの精度向上を目的とする。

本研究では、グラフベースではなくクラスタリングに基づく手法で語義識別を行う。Schütze の手法をはじめとする先行研究の多くは、用例を 1 種類のベクトルで表現する。しかし、これでは多様な観点から語義の類似性をとらえることは難しい。一方、九岡・田中の手法は、4 つの特徴ベクトルについて、それぞれクラスタリングを行っている。算出された 4 つのクラスタ集合に対して、式 (2.3) によって最良と思われるクラスタ集合を 1 つ選択するという手法である。しかし、1 章で考察したように、語義によってもそれを特徴づけやすい観点が異なる場面があるということから、クラスタリングの段階において複数の観点を比較しながら用例のクラスタを作成する必要性がある。本研究では複数の特徴ベクトルを同時に考慮し、新語義発見に向けての語義識別の精度向上を目的とする。

この方式を採用することによって、クラスタごとに異なる観点で同じ意味を持つと判断された用例がまとめられることになる。例を図 2.1 に示す。図 2.1 において、記号の形状が語義、枠線の種類が注目した観点に該当している。つまり、同じ形状であれば同じ語義、同じ枠線の種類であれば同じ観点であることを意味している。

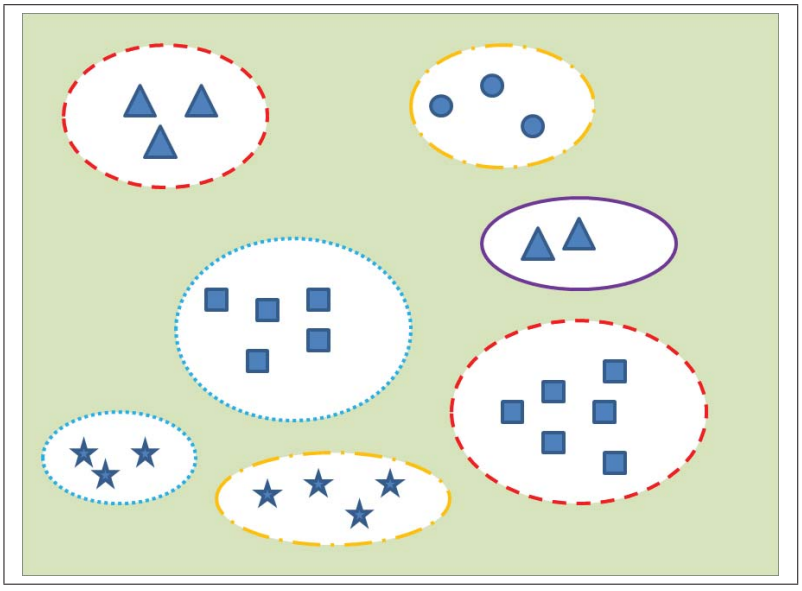

図 2.1: クラスタリング結果の例

一般に語義識別では以下 2 つの条件が要求される。

- 同じ語義を持つ用例をまとめたクラスタを作成すること。
- 語義の数を推定し、それと同じ数だけクラスタを作成すること。

しかし、本研究は新語義発見に向けた語義識別を行うため、同じ語義をまとめたクラスタを作成することが重要である。逆に、語義の数が推定出来なかったとしても、つまり図 2.1 のように同じ語義の用例が複数のクラスタに分割されてしまっていたとしても、同じ語義でまとめられた新語義のクラスタさえ作成されれば、新語義の検出は十分可能である。したがって、本研究では前者を優先し、同じ語義を持つ用例を 1 つのクラスタとしてまとめる事は目的としない。

また、先に述べたように九岡・田中の手法では分割型クラスタリングである k-means 法を用いて用例のクラスタリングを行っているが、初期状態から k 個のクラスタにランダムに割り当てられる段階で、クラスタリングの精度が低下してしまい、クラスタがどれだけ同じ語義を持つ要素でまとめられるか表わす純度という値について、常に高い値を示す保証がない。本研究では小さなクラスタであってもクラスタの純度が高い方が新語義発見において望ましい点と、複数の特徴ベクトルを同時に考慮する手法を実装しやすい点の 2 点から凝集型クラスタリングを用いる。

第3章　提案手法

本研究では、特徴ベクトルの作成とクラスタリングの二つを主な処理とする。それぞれ3.1節および3.2節において詳細を述べる。

3.1　特徴ベクトル

本研究で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)を表現している。この表現方法は、用例のクラスタリングを行った九岡・田中[11][12]の用いた4つの特徴ベクトルと基本的には同じ方法でベクトルを表現する。本節では、各特徴ベクトルの詳細と先行研究からの改良について述べる。

3.1.1　隣接ベクトル

対象単語 w について、前後2単語を特徴付けるベクトルを指す。w の前後2単語の出現形、及び品詞をベクトルの要素としている。なお、九岡・田中は前後1単語の出現形、並びに品詞を隣接ベクトルの要素として扱っていた。しかし、前後に出現する単語が完全に等しい場合でも、異なる語義のものが存在する場合がある。

「進める」の例を以下に示す。

1. ••• 主な流れとして筆を*進め*たいと思います。
2. ••• 検討を*進め*たい」と導入に意欲を示した。

これらの例は

1. 前方へ行かせる。 (語義ID:26839-0-1-1-1)
2. はかどらせる。　(語義ID:26839-0-1-1-2)

といった語義に対応している。

これらの用例に共通している事は、前後の単語を見た場合に「進め」の前後には助詞「を」と助動詞「たい」が付随している点である。このように、本来は違う語義であっても、前後の 1 単語を素性とした場合は、同じ語義と認識されてしまうことが多い。したがって、誤ったクラスタリングを抑止するためには、隣接ベクトルのウィンドウ幅を増加させる必要がある。そのため、本研究では前後 2 単語の出現形・品詞を素性として隣接ベクトルを作成する。このように、本研究における隣接ベクトルは九岡・田中の手法を改良したものである。

また、九岡は隣接ベクトル作成の際に単語と品詞を同じ重みとしている。しかし、前後の品詞のみが一致している場合にも高い類似度が計算されてしまう場面が多々あった。こういった問題に対応するため、単語の出現形の重みは 1.0、品詞の重みは出現形の単語の 1/2 である 0.5 として設定している。

3.1.2 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られるベクトルを指す。以降、連想ベクトルと呼ぶ。

連想ベクトルの作成には、以下の行程を前処理として行う。

1. 対象のコーパスから、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。また、この共起行列 A_c の要素 a_{ij} は、単語 c_k が文書 d_l に出現した回数とする。

2. 共起行列 A_c に対して LDA(Latent Dirichlet Allocation)[5] を適用し、トピックと単語の関連性を表すパラメータを学習する。

3. 各トピック z_m に対して、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。

ここからインスタンス w_i の文脈ベクトル $\vec{c_i}$ を以下のように定義する。

1. w_i の周辺に自立語 c_{ij} が出現した場合、$\vec{c_i}$ において c_{ij} の重みを 1 にする。

2. c_{ij} に重みが付与され、なおかつ c_{ij} があらかじめ作成された Z_m に含まれている場合、Z_m の残りの単語について重みを 0.5 として $\vec{c_i}$ の要素とする。

なお、ここでの周辺とは、インスタンスから前後 50 単語を示す。周辺に出現する語だけをベクトルの素性とするだけでは、一般にベクトルがスパースになり、語義の類似性を正しく測ることが出来ない。そこで、文脈ベクトルでは、関連語 Z_m に含まれる単語を素性として追加することで、ベクトルの過疎性を緩和している。

3.1.3 連想ベクトル

文脈ベクトルと同じく、対象単語 w の周辺に出現する単語で特徴付けられるベクトルを指す。文脈ベクトルとの差異は、コーパスにおいて出現頻度が上位 10000 語を行、コーパスにおける高頻度語 10000 語と岩波国語辞典の語釈文中の自立語の和集合を列として共起行列 A_a を作成する点にある。なお、A_a の要素 a_{ij} は出現頻度上位 10000 語の単語 c_i と上記の和集合に含まれる単語 c_j が同じ文書で共起した回数を指す。

連想ベクトル $\vec{a_i}$ は、w_i の周辺に出現する自立語 c_j に対する共起ベクトルの $\vec{o}(c_j)$ の和とする。$\vec{o}(c_j)$ とは共起行列の j 番目に対応するベクトルを指す。

$$\vec{a_i} = \sum_{c_j \in context} \vec{o}(c_j)$$

ここでの周辺とは文脈ベクトルと同じく対象のインスタンスの前後 50 単語を指す。連想ベクトルは Schütze の手法 [3] のように二次共起 (あるいは間接共起) の情報を用いることによって、文脈ベクトルとは異なる方法でベクトルのスパースネスに対応している。

3.1.4 トピックベクトル

トピックベクトルとは PLSI(Probabilistic Latent Semantic Indexing)[6] によって推定されるトピックから、対象単語 w を特徴付けるベクトルを指す。トピックベクトルの作成において、以下の前処理を行う。

1. 単語を行、文書を列とする共起行列 A_c を作成する。これは文脈ベクトル作成時と同じものである。

2. 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す確率パラメータを学習する。

3. 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文書 d_i に対して、トピック z_m が割り当てられる確率パラメータ $P(z_m|d_i)$ を推定する。

以上の行程で算出された $P(z_m|d_i)$ を用いて、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。

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

ここでの M は、PLSI の隠れ変数の数を表す。九岡・田中は M=50 としており、本研究でも同様に M=50 とする。

3.1.5 特徴ベクトルのまとめ

3.1.1〜3.1.4 項では特徴ベクトルの作成手法について述べた。本研究ではそれら 4 つの特徴ベクトルでインスタンスを特徴付けて表現する。ここで、1 章で例に挙げた「サービス」の用例を再度紹介する。

1. 前後の語から同じ意味と判断できるもの
   (a) あとのぶんは*サービス*残業...
   (b) いわゆる「*サービス*残業。...
2. 周辺文脈から同じ意味と判断できるもの
   (a) ケーキとシャンパンの*サービス*...
   (b) 値段と味と*サービス*のバランスが...
3. 特定のトピックの文書に出現することで同じ意味と判断できるもの
   (a) Apache*サービス*をインストール...
   (b) オラクルの*サービス*再起動方法...

上記の例のように、同じ単語であっても語義ごとに異なる観点で特徴づけられる場面がある。各特徴ベクトルはこの例で考察した語義の類似性を測るための観点と以下のように対応している。

1. 前後の単語で特徴づけられるもの :隣接ベクトル

2. 周辺の文脈で特徴づけられるもの :文脈ベクトル、連想ベクトル

   文脈ベクトルと連想ベクトルの違いは、ベクトルの過疎性を緩和させる方法が異なる点にある。

3. 文書のトピックで特徴づけられるもの :トピックベクトル

対象のインスタンス (用例) を特徴づける観点が異なれば、作成されるクラスタ集合も異なる。それらの観点を使い分ける事は、語義識別において非常に効果的であると考えられる。

3.2 クラスタリング

本節では、本研究で用いるクラスタリングの方法について述べる。

3.2.1 アルゴリズム

本研究の目的は、複数の特徴ベクトルを同時に考慮することで、語義識別の精度を向上させることにある。本研究で提案するアルゴリズムは凝集型クラスタリングを拡張したものである。凝集型クラスタリングの手順は以下の通りである。

1. 1 要素 1 クラスタを初期状態とする。
2. すべてのクラスタの組に対して、類似度の計算を行う。
3. 類似度が最大となったクラスタの組を、1 つのクラスタにマージする。
4. 停止条件を満たすまで 2,3 を繰り返す。

また、本実験では凝集型クラスタリングの停止条件として、式 (3.2) を設けた。

$$\begin{cases} \text{クラスタの数が } Tc \text{ 以下} \\ \text{最大のクラスタの要素数の全用例数に対する比が } Tr \text{ 以上} \end{cases} \tag{3.2}$$

式 (3.2) の条件について、前者については早い段階でのクラスタリングの停止を抑制するものである。後者は、新語義発見にはある程度の要素をまとめたクラスタが必要であることから条件として設けている。

式 (3.2) の条件をすべて満たすまでクラスタリングを継続する。本研究では、Tr の値は 1/5 として固定している。端数は切り上げることとした。ただし、本停止条件における閾値 Tc,Tr の最適化はしていない。これは今後の課題である。

本研究では、クラスタ間の類似度を計算する際に、用いる 4 つの特徴ベクトルすべてに対して類似度の計算を行う。求めた類似度を比較し、最大値のものをクラスタ間の類似度とすることで、複数の特徴ベクトルを同時に考慮する (式 (3.3))。なお、ベクトル間の類似度の計算にはコサイン類似度を用いることとする (式 (3.4))。

$$\begin{array}{l} sim(C_i, C_j) = \max_x sim(x, \vec{v_i}, \vec{v_j}) \\ x \in \{ \text{隣接, 文脈, 連想, トピック} \} \end{array} \tag{3.3}$$

$$sim(x, \vec{v_i}, \vec{v_j}) = \frac{\vec{v_i} \cdot \vec{v_j}}{\|\vec{v_i}\| \, \|\vec{v_j}\|} \quad (x \text{ は特徴ベクトルの種類を表す}) \tag{3.4}$$

式 (3.3) において C_i,C_j はクラスタの組を指し、$sim(x, \vec{v_i}, \vec{v_j})$ とは特徴ベクトル x によって計算されるクラスタの重心ベクトル $\vec{v_i}$,$\vec{v_j}$ の類似度である。4 つの特徴ベクトルのうち、最大の類似度をクラスタ間の類似度として定義しているのは、4 つの観点のうちどれか 1 つでも類似度が高ければ、2 つのクラスタの用例は同じ語義を持つ可能性が高いという考えに基づく。

さらに、本研究では、同じ種類の特徴ベクトルの類似度が高い用例しかマージしないという制約をつけて用例のクラスタリングを行う。これは、クラスタラベルという概念を導入して表現する。二つのクラスタがマージされた場合に、そのクラスタに対してクラスタラベル L を与える事にする。L はマージされた場合に注目された特徴ベクトルの種類、つまり式 (3.3) で最大の類似度を持つものとして選択された特徴ベクトルの種類を表す。そして、クラスタラベル L を用いた場合のクラスタ間類似度 $sim(C_i, C_j)$ は式 (3.5) によって定義される。

$$sim(C_i, C_j) = \begin{cases} \max_x \ sim(x, \vec{v_i}, \vec{v_j}) & \text{if } L(C_i) = L(C_j) = \text{未定} \\ sim(L(C_i), \vec{v_i}, \vec{v_j}) & \text{if } L(C_i) = L(C_j) \text{ or } L(C_j) = \text{未定} \\ sim(L(C_j), \vec{v_i}, \vec{v_j}) & \text{if } L(C_i) = L(C_j) \text{ or } L(C_i) = \text{未定} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

L の要素は特徴ベクトルの種類である { 隣接, 文脈, 連想, トピック } の中の 1 つである。式 (3.5) について、2 つのクラスタ C_i, C_j がクラスタラベルを持たない場合、クラスタ間の類

似度には、4 つの特徴ベクトルの類似度の中から最大のものが選択される (式 (3.5) の 1 行目)。また、クラスタ C_i または C_j のいずれかが一方がクラスタラベルを保有している場合、または C_i, C_j の両方がクラスタラベルを保有しており、$L(C_i)$ と $L(C_j)$ が同一であった場合には、未定でないクラスタラベル $L(C_i)$ または $L(C_j)$ と同じ特徴ベクトルでクラスタ間の類似度を計算する。クラスタラベル L は以下のように決定する。まず、初期状態のクラスタ (どの要素ともマージされていないクラスタ) のラベルは「未定」とする。新しくマージされたクラスタ (C_k) は、クラスタ間の類似度として選択された特徴ベクトルの種類をクラスタラベル $L(C_k)$ として記憶する。

このクラスタラベルを用いた制約を図 3.1 で説明する。図 3.1 では、4 つの図を例とし

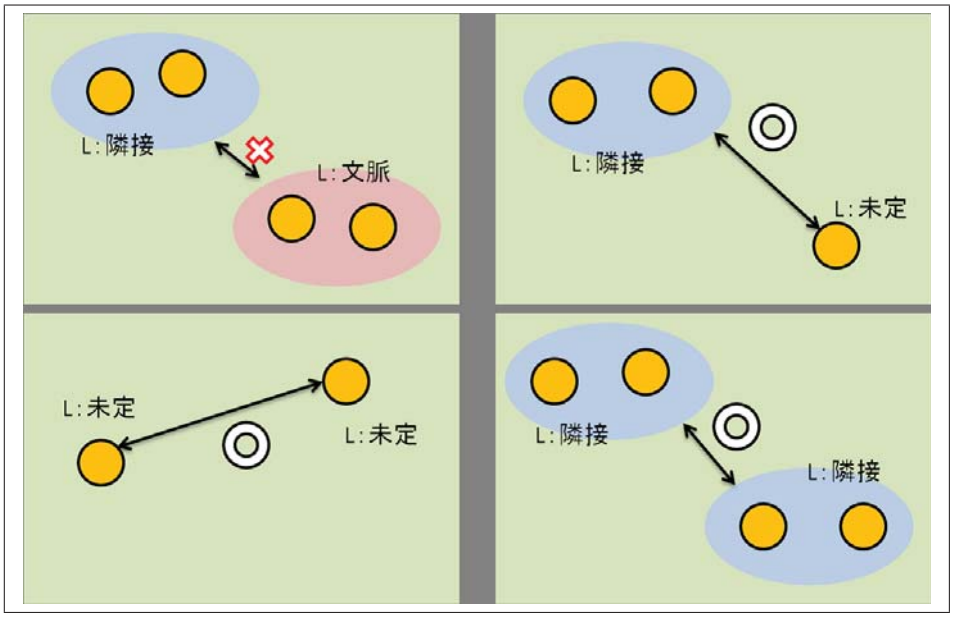

図 3.1: マージ可能な例と不可能な例

ている。例ではマージ不可能な例は L の種類が一致していない左上のもののみである。

このクラスタラベルを用いることによって、各クラスタには 1 つの特徴ベクトルに注目してマージされた用例が含まれる。これにより、生成されたクラスタがどのような観点で類似性が認められたのかを把握することが可能である。

3.2.2 類似度の正規化

予備実験として、各特徴ベクトルの類似度の値を調べたところ、平均値に大きな差がみられた。複数の特徴ベクトルから最大の類似度をもつベクトルを選択する際に常に 1 種類

の特徴ベクトルが選択されることが予想される。そのため、2 つの手法を用いて類似度の正規化を行う。

1 つ目の正規化として相対値を用いるものを示す。クラスタリングの前処理として以下を行う。

1. すべての用例の組について 4 種類の特徴ベクトルすべての類似度を計算する。
2. 1 の中から各特徴ベクトル毎に類似度の最大値 max_x と最小値 min_x を求める。なお、$x \in$ { 隣接, 文脈, 連想, トピック } とする。

上記の前処理を経て、式 (3.6) を用いてベクトル間類似度 $sim(x, \vec{v_i}, \vec{v_j})$ の正規化を行う。$sim_R(x, C_i, C_j)$ は正規化後のクラスタ間類似度である。

$$sim_R(x, \vec{v_i}, \vec{v_j}) = \frac{sim(x, \vec{v_i}, \vec{v_j}) - min_x}{max_x - min_x} \tag{3.6}$$

4 節の評価実験に用いたデータを用いて、17 単語、1 単語につき約 50 のインスタンスに対して、全ての用例間の組での正規化を行う前と正規化を行った後の類似度の平均を求めた。結果を表 3.1 に示す。

表 3.1: 正規化前と正規化後の類似度平均

ベクトルの種類	類似度平均	正規化後
隣接ベクトル	0.0320	0.0434
文脈ベクトル	0.4571	0.5152
連想ベクトル	0.8903	0.6393
トピックベクトル	0.2391	0.2693

この正規化により、類似度のバラつきは多少低減できた。なお、表 3.1 では隣接ベクトルとトピックベクトルの平均値は正規化を行った場合でも大きく変化しない。しかしながら、これらの特徴ベクトルは類似度の平均値は低いが標準偏差は高い値を持つ。つまり、類似度の平均が低い場合であっても、用例の組によっては類似度が大きいものも存在する。したがって、隣接ベクトルやトピックベクトルが全く選択されないという問題は起こりにくいと考えられる。

また、2 つ目の手法は偏差値による正規化である。この正規化でもクラスタリングの前処理として以下を行う。ここでの N とは計算に用いた用例の組の総数を指す。

1. すべての用例の組について 4 種類の特徴ベクトルすべての類似度を計算する。

2. 1 の中から各特徴ベクトル x 毎に類似度の平均値 μ_x を求める。

$$\mu_x = \frac{1}{N}\sqrt{\sum_{i,j}(sim(x, \vec{v_i}, \vec{v_j}))} \tag{3.7}$$

なお、相対値と同じく $x \in \{$ 隣接, 文脈, 連想, トピック $\}$ とする。

3. 2 で求めた類似度の平均値 μ_x を用いて標準偏差 σ_x を求める。

$$\sigma_x = \sqrt{\frac{1}{N}\sum_{i,j}(sim(x, \vec{v_i}, \vec{v_j}) - \mu_x)^2} \tag{3.8}$$

上記の前処理を経て、式 (3.9) によって正規化を行う。すなわち、$sim_{SD}(x, \vec{v_i}, \vec{v_j})$ とは、全ての用例の組についての類似度の標本における偏差値とする。

$$sim_{SD}(x, \vec{v_i}, \vec{v_j}) = \frac{10(sim(x, \vec{v_i}, \vec{v_j}) - \mu_x)}{\sigma_x} + 50 \tag{3.9}$$

また、μ_x, σ_x の計算において、類似度 0 の組は計算に用いないことする。この理由は以下の通りである。類似度 0 の組が多い特徴ベクトルは、平均値が下がり、類似度が 0 でないベクトルの組に対する偏差値が不当に大きく見積もられる。このとき、類似度 0 の組を多く含む特徴ベクトル (具体的には隣接ベクトル) の正規化後の類似度 (偏差値) が大きくなり、そればかりが選択されてしまう可能性が高い。このため μ_x, σ_x は類似度が 0 となる場合を除いて計算する。

偏差値を用いた正規化が相対値を用いた正規化と大きく異なる点は、最大値または最小値が大きく平均から突出した類似度を持つクラスタの組が存在した場合に、相対値での正規化では正規化後の類似度に大きく影響を与えてしまう傾向がある。そういった場合、正しく正規化が行われているとは言えない。したがって、各特徴ベクトルごとの平均 μ_x と標準偏差 σ_x をあらかじめ計測し、偏差値を用いて複数の特徴ベクトルの比較を行う。

これにより、表 3.1 と同じく 17 単語、1 単語につき約 50 のインスタンス、全ての用例間の組について、正規化前の類似度平均と偏差値による正規化後の類似度の平均を表 3.2 に示す。この表の「類似度の平均」の列は表 3.1 の再掲である。

正規化前では特徴ベクトルによって類似度の大きさにばらつきがあるのに対し、当然だが、偏差値による正規化後ではどの特徴ベクトルも類似度の平均は 50 である。したがっ

表 3.2: 正規化前と正規化後 (偏差値) の類似度平均

ベクトルの種類	類似度の平均	偏差値の平均
隣接ベクトル	0.0320	50
文脈ベクトル	0.4571	50
連想ベクトル	0.8903	50
トピックベクトル	0.2391	50

て、偏差値による正規化により、異なる特徴ベクトルの類似度をある程度公平に比較できるようになると考えられる。

第4章　評価

本章では、提案手法の評価実験について述べる。

4.1　実験データ

ここでは実験に用いたデータについて述べる。

4.1.1　Semeval-2 日本語タスク訓練データ

現代日本語書き言葉コーパス (Balanced Corpus of Contemporary Written Japanese :BCCWJ) とは、国立国語研究所で進められているプロジェクトによって提供されており、日本語研究の活性化を目指して構築されているコーパスである [13]。本研究では、BCCWJ を基にした SemEval-2 日本語タスク [8] の訓練データを対象にして実験を行う。SemEval-2 日本語タスク訓練データは BCCWJ から白書 (OW)、書籍 (PB) 及び新聞 (PN) の一部に品詞、語義、読み、の情報を付与し、xml 形式で表記したものである (カッコ内は、訓練データでのテキストジャンルの識別コードを指す)。

白書についてのコーパスデータの一例を図 4.1 に示す。なお、この例は以下の文に情報を付与している。

- 現行の円借款の供与条件では一部の環境案件、人材育成、中小企業育成、

また、図 4.1 において、多義語である「案件」という単語には岩波国語辞典 [14] に基づいて割り当てられた語義が付与されている。語義 ID の表記方法を図 4.2 に、岩波国語辞典における「出す」の語義の定義の一部を図 4.3 で示す。

岩波国語辞典では、大分類、中分類、小分類といった分け方がなされており、小分類は中分類に、中分類は大分類に、それぞれ属している。これらの 3 種類の分類を用いた語義は、図 4.2 のフォーマットで表記される。単語 ID とはどの単語を意味するかを示し、そのあとに大分類、中分類、小分類の順に数字を用いて語義が表記される。

```
- <sentence>
    <mor pos="名詞-普通名詞-一般" rd="ゲンコー">現行</mor>
    <mor pos="助詞-格助詞" rd="ノ">の</mor>
    <mor pos="名詞-普通名詞-一般" rd="エン">円</mor>
    <mor pos="名詞-普通名詞-一般" rd="シャッカン">借款</mor>
    <mor pos="助詞-格助詞" rd="ノ">の</mor>
    <mor pos="名詞-普通名詞-サ変可能" rd="キョーヨ">供与</mor>
    <mor pos="名詞-普通名詞-一般" rd="ジョーケン">条件</mor>
    <mor pos="助詞-格助詞" rd="デ">で</mor>
    <mor pos="助詞-係助詞" rd="ワ">は</mor>
    <mor pos="補助記号-読点" rd="、">、</mor>
    <mor pos="名詞-普通名詞-副詞可能" rd="イチブ">一部</mor>
    <mor pos="助詞-格助詞" rd="ノ">の</mor>
    <mor pos="名詞-普通名詞-一般" rd="カンキョー">環境</mor>
    <mor pos="名詞-普通名詞-一般" rd="アンケン" sense="1489-0-0-0-0">案件</mor>
    <mor pos="補助記号-読点" rd="、">、</mor>
    <mor pos="名詞-普通名詞-一般" rd="ジンザイ">人材</mor>
    <mor pos="名詞-普通名詞-サ変可能" rd="イクセー">育成</mor>
    <mor pos="補助記号-読点" rd="、">、</mor>
    <mor pos="接頭辞" rd="チュー">中</mor>
    <mor pos="接頭辞" rd="ショー">小</mor>
    <mor pos="名詞-普通名詞-一般" rd="キギョー">企業</mor>
    <mor pos="名詞-普通名詞-サ変可能" rd="イクセー">育成</mor>
    <mor pos="補助記号-読点" rd="、">、</mor>
```

図 4.1: コーパスの一例

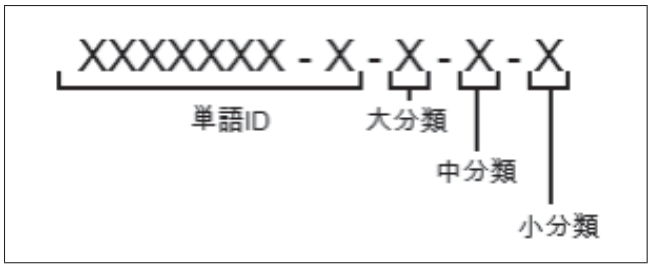

図 4.2: 岩波国語辞典における語義の表記方法

大分類	中分類	小分類	語義
一			
	①		内から外に移す
		ア	移して外部におく、また外方への動きを示す
		イ	出発・発車・出帆させる
		ウ	身近なところからはなしてよそに行かせる
		エ	(手紙など)を送る
	…	…	…
	②		他から姿が、または形として見えるようにする
		ア	現す
		イ	物事に取り掛かり、新たに開く
		ウ	(速度など)新たに加える
	…	…	…
二			・・・し始める

図 4.3: 岩波国語辞典における「出す」の語義の定義

また、図 4.3 では岩波国語辞典における「出す」という単語を例に、大分類、中分類、小分類の関係を示した。それらの従属関係については先ほど述べた。なお、大分類は漢数字の一から順に、中分類は数字の 1 から順に、小分類は片仮名のアに始まりアイウエオ順で、それぞれ順序づけられる。図 4.3 において、大語義 [一]、中語義 [1]、小語義 [ウ] に属する「身近なところからはなしてよそへ行かせる」という語義 ID は以下のように定められる。

1. 「出す」という単語の ID は 31472-0 である。
2. 大語義 [一] より、大語義の値は 1。
3. 中語義 [1] より、中語義の値は 1。
4. 小語義 [ウ] より、小語義の値は 3。
5. 1〜4 より、語義 ID は [31472-0-1-1-3] と表記される。

本実験で用いるコーパスには、岩波国語辞典の語義の 1 つが正しい語義として付与されている。この語義情報を正解とみなして、用例のクラスタリングの評価を行う。

4.2 評価実験

本実験の流れを以下に示す。

1. コーパスから対象単語の用例を抽出する。
2. 抽出された用例を表現する特徴ベクトルを作成する。
3. 作成された特徴ベクトルを用いてクラスタリングを行う。
4. クラスタリングの結果を評価する。

本節では、実験方法及び、実験結果について述べる。

4.2.1 実験方法

先にも述べたように、本実験はコーパスにから対象単語の用例を抽出し、各用例ごとに特徴ベクトルを作成する。

なお、本研究ではコーパスからインスタンス (用例) を抽出する際に、日本語表記の基本形を用いて抽出を行ったため、本来は抽出されるはずのデータが取り出されない場合がある。「入れる」というインスタンス集合のデータには「いれる」という単語が含まれているが、基本形の表記は異なるため、基本形をキーとした検索では抽出されないことが例として挙げられる。本研究の対象単語は SemEval-2 日本語タスクにおける対象単語 50 語を基にしており、これらの対象単語は用例数が 50 語と統一されている。しかし、上記のような表記ゆれから抽出される用例数が減少してしまう語が複数存在したため、本研究では対象単語 50 語の内、抽出された用例が 40 語以上 50 語以下の単語のみを対象単語として設定した。その結果、対象単語を図 4.4 に示す 40 語とした。

相手、与える、場所、文化、出す、電話、出る、現場、技術、早い、はじめ、
生きる、意味、大きい、教える、可能、考える、関係、経済、子供、時間、
市場、社会、情報、進める、する、高い、立つ、強い、手、乗る、始める、
開く、前、見える、認める、持つ、求める、良い

図 4.4: 本実験で用いる対象単語 40 語の基本形

対象単語のインスタンスから作成された特徴ベクトルに対して、提案手法である複数の特徴ベクトルを同時に考慮した手法、及び単独の特徴ベクトルを用いた手法によってクラスタリングを行う。

4.2.2 評価尺度について

本研究では生成されたクラスタを評価する際に以下の評価尺度を用いた。これらはクラスタリングを評価する際によく用いられる尺度である。特に、V-measure と Paired F-score は、SemEval-2 の英語の語義推定タスクにおいて評価指標として採用されている評価尺度である [7]。

- Purity , I-Purity , F-measure

- Homogeneity , Completeness , V-measure
- Paird Precision , Paired Recall , Paired F-score

以下、これらの評価尺度の定義について述べる。

Purity とは、クラスタの純度を示す。具体的には、1 つのクラスタ内にどれだけ同じ要素がマージ (併合) されているかを表現している。Purity は 1 を最大値としており、1 に近ければ近いほど、良い結果であることを表している。定義を式 (4.1) に示す。

$$Purity = \sum_{j=1}^{\Gamma} \frac{|P_j|}{N} \max_{L_i \in \Lambda} \frac{|L_i \cap P_j|}{|P_j|} \tag{4.1}$$

ここでは Γ がクラスタの数、Λ が全語義の数を表す。P_j は作成されたクラスタを表す。すなわち、用例集合は $P_1 \ldots P_\Gamma$ の部分集合で分割された状態にある。一方、L_i とは語義を表す。用例の集合は正解として付与された語義 ID に応じて $L_1 \ldots L_\Lambda$ の部分集合に分割される。Purity は、クラスタ P_j に含まれる最多数の語義に対し、それがどの程度クラスタ内を占めているかを見る評価尺度である。

I-Purity とは同じ語義を持つ要素がどれだけ同じクラスタにマージされているかを測る評価尺度を指す。I-Purity も Purity と同じく、1 に近ければ近いほど、良い値であることを表している。定義を式 (4.2) に示す。

$$I\text{-}Purity = \sum_{i=1}^{\Lambda} \frac{|L_i|}{N} \max_{P_j \in \Gamma} \frac{|L_i \cap P_j|}{|L_i|} \tag{4.2}$$

ここでも Purity と同様に P_j が作成されたクラスタ、L_i が語義を表す。ラベル L_i を持つ要素が 1 つのクラス P_j にどの程度まとめられているのかを見る評価尺度である。

Purity と I-Purity の値の調和平均が、F-measure という評価尺度である (式 (4.3))。

$$F\text{-}measure = \frac{(1+\beta^2) \cdot Purity \cdot I\text{-}Purity}{(\beta^2 \cdot Putiry) + I\text{-}Putiry} \tag{4.3}$$

なお、ここでの β は重み付けを表している。β が 1 よりも小さい場合には I-Purity が重視され、逆に β が 1 よりも大きい場合には Purity が重視される。本研究では一般的な値として $\beta = 1.0$ とした。

Homogeneity とは、同質性を意味しており、Purity と同じくクラスタ内にどれだけ同じ語義を持つ用例がマージされているかを表現する。Purity と大きく異なる点としては、エ

ントロピーを基にした評価尺度であり、評価値が語義の数と分布に依存しない点である。定義を式 (4.4) に示す。なお、Homogeneity は 1 が最大値であり、1 に近ければ近いほど良い結果であることを示す。

$$Homogeneity = \begin{cases} 1 & \text{語義が一つしか存在しないとき} \\ 1 - \frac{H(L|P)}{H(P)} & \text{else} \end{cases} \tag{4.4}$$

なお、$H(L|P), H(L)$ については、式 (4.5),(4.6) を用いて求められる。

$$H(L|P) = -\sum_{j=1}^{\Gamma}\sum_{i=1}^{\Lambda} \frac{|L_i \cap P_j|}{N} \log \frac{|L_i \cap P_j|}{|P_j|} \tag{4.5}$$

$$H(L) = -\sum_{i=1}^{\Lambda} \frac{|L_i|}{N} \log \frac{|L_i|}{N} \tag{4.6}$$

L_i は $\{L_1 \dots L_\Lambda\}$ に、P_j は $\{P_1 \dots P_\Gamma\}$ に、それぞれ属している。なお、Λ は語義の数を、Γ はクラスタの数をそれぞれ表している。Homogeneity は条件付きエントロピー $H(L|P)$(式 (4.5)) に対する、語義 L のエントロピー (式 (4.6)) 比と定義されている。H(L) が小さいとき、つまり語義の分布に大きな偏りがあるときには、$H(L|P)$ すなわちクラスタ内の語義の均質性も高く見積もられる。Homogeneity は $H(L|P)$ と $H(L)$ に対する比と定義されているので、語義の分布に依存しない評価が可能である。

Completeness は I-Purity と類似した評価尺度で、同じ語義を持つ要素が一つのクラスタにどれだけまとめられているかについてを評価する指標である。これは Homogeneity と同じく、エントロピーに基づく評価尺度であり、語義の数や分布に依存しない特徴を持つ。求め方を式 (4.7) に示す。なお、Homogeneity と同じく、1 に近ければ近いほど良い結果であることを示す。

$$Completeness = \begin{cases} 1 & \text{クラスタが 1 つしか存在しないとき} \\ 1 - \frac{H(P|L)}{H(P)} & \text{else} \end{cases} \tag{4.7}$$

Homogeneity と同様に、$H(P|L)$,$H(P)$ の求め方は式 (4.8),(4.9) とする。

$$H(P|L) = -\sum_{j=1}^{\Lambda}\sum_{j=1}^{\Gamma} \frac{|L_i \cap P_j|}{N} \log \frac{|L_i \cap P_j|}{|L_i|} \tag{4.8}$$

$$H(P) = -\sum_{j=1}^{\Gamma} \frac{|P_j|}{N} \log \frac{|P_j|}{N} \tag{4.9}$$

$H(P|L)$ はある語義 L_i を持つ要素が様々なクラスタに分配して配置されている状態に対するエントロピーであり、同じ語義を持つ要素が 1 つのクラスタにまとめられているほど低い値をとる (式 (4.8))。一方、H(P) はクラスタの要素数のばらつきをエントロピーで評価している (式 (4.9))。式 (4.8) と式 (4.9) の比をとることで Homogeneity と同じくクラスタの大きさの分布に依存しない評価が可能である。

V-measure は Homogeneity と Completeness の調和平均である (式 (4.10))。

$$V\text{-}measure = \frac{(1+\beta^2) \cdot Homogeneity \cdot Completeness}{(\beta^2 \cdot Homogeneity) + Completeness} \tag{4.10}$$

F-measure と同じく V-measure についても β は重み付けを表している。β が 1 よりも小さい場合には Completeness が重視され、逆に β が 1 よりも大きい場合には Homogeneity が重視される。本研究では、F-measure と同じく一般的な値として $\beta = 1.0$ とする。

Paired Precision とは、同じクラスタ内の要素に対してどれだけ同じ語義を持つ要素がまとまっているかを見る指標である。定義を式 (4.11) に示す。なお、以降では Paired Precision を PP と表記する。

$$PP = \frac{|F(K) \cap F(S)|}{|F(K)|} \tag{4.11}$$

式 (4.11) において、$F(K)$ は同じクラスタに属している全ての要素の組の集合を表し、$F(S)$ は同じ語義を持つ全ての要素の組の集合を指す。これらの二つの値から、同じクラスタに同じ語義を持つ要素がどの程度まとめられているのかを評価することが出来る。

Paired Recall とは、同じ語義を持つ要素が同じクラスタにどの程度まとめられているかを見る指標であり、式 (4.12) と定義される。なお、以降では Paired Recall を PR と表記する。

$$PR = \frac{|F(K) \cap F(S)|}{|F(S)|} \tag{4.12}$$

Paired F-score は F-measure や V-measure と同じく、Paired Precision と Paired Recall との調和平均と定義される。なお、定義式は式 (4.13) である。

$$Paired\ F\text{-}score = \frac{2 \cdot PP \cdot PR}{PP + PR} \tag{4.13}$$

本研究では新語義発見のためにクラスタリングの精度向上を目的としている。2 章でも述べたが、語義識別の一般的な目標は以下の 2 つである。

- クラスタの中に異なる語義を持つ用例を混在させず、同じ意味を持つ用例のみをまとめてクラスタを作成すること
- 同じ意味を持つ用例を 1 つのクラスタにまとめること。つまり、語義の数と同じ数のクラスタを作成する。語義の数を推定することとも言える。

同じ語義を持つ用例をまとめたクラスタが作成されれば、語義の特定は可能であるため、新語義の発見も可能である。したがって、本研究では前者を重視している。この評価に適した評価指標は Purity, Homogeneity, PP である。したがって本項で示した 9 つの評価指標のうち、今回の実験では、Purity, Homogeneity, PP に注目する。なお、本研究では語義の数を特定することは行わない。

4.2.3 予備実験

ここでは予備実験として、九岡らの隣接ベクトルと、提案手法の隣接ベクトルとの比較を述べる。対象単語 40 語については 4.2.1 項にてすでに述べた。ここでは対象単語を約半数の 17 語に限定して実験を行っている。これらの対象単語の用例を凝集型クラスタリングによってクラスタを作成する。

凝集型クラスタリングの停止条件を式 (4.14) にて再度示す。

$$\begin{cases} \text{クラスタ数が } Tc \text{ 以下} \\ \text{最大のクラスタの要素数の全用例数に対する比が全体の } Tr \text{ 以上} \end{cases} \tag{4.14}$$

Tc は 10 とした。Tr については、全ての用例数の 1/5(端数切り上げ) と定めることは 3.2.1 項にてすでに述べた。

それぞれのベクトルを用いて作成されたクラスタ集合の Homogeneity, Completeness, V-measure を表 4.1 にて示す。手法の表記は、「隣接ベクトル (九岡ら)」は先行研究の方法を示し、前後 1 語を素性とするベクトルを指している。「隣接ベクトル」は本研究で用いるもので、前後 2 語を素性とする特徴ベクトルを指す。

表 4.1: 対象単語 17 語について隣接ベクトルの差異

手法	Homogeneity	Completeness	V-measure
隣接ベクトル (九岡ら)	0.3870	0.1814	0.2225
隣接ベクトル	0.4136	0.2008	0.2387

表 4.1 から、隣接ベクトルの前後のウィンドウ幅を 2 とした方が、Homogeneity が高いことが分かった。したがって、本研究ではウィンドウ幅を 2 とした隣接ベクトルを用いている。

4.2.4 実験結果

本項では実験結果について述べる。2 章でも述べたように、凝集型クラスタリングは停止条件によって生成されるクラスタ集合が変化するため、本研究では異なる停止条件について 2 つの実験を行っている。すなわち、式 (4.14) において Tc を異なる 2 つの値 $Tc = 10, Tc = 15$ と定めた。対象単語 40 単語について実験を行い、既存の手法との比較行う。比較方法として、4.2.2 項で述べた 9 つの評価尺度をそれぞれ対象単語について求め、それらの平均を求める。$Tc = 10$ についての実験結果を表 4.2, 4.3, 4.4 に、$Tc = 15$ についての実験結果を表 4.5, 4.6, 4.7 にそれぞれ示す。また、手法欄の表記方法は以下の通りである。

- 提案手法 (正規化あり$_{SD}$)

  4 種類の特徴ベクトルを同時に用いる手法で、式 (3.9) にて定義した偏差値を用いて正規化を行う。

- 提案手法 (正規化あり$_{R}$)

  4 種類の特徴ベクトルを同時に用いる手法で、式 (3.6) にて定義した相対値を用いて正規化を行う。

- 提案手法 (正規化なし)

  4 種類の特徴ベクトルを同時に用いる手法で、正規化を行わないものを指す。

- 九岡ら

  4 回のクラスタリングを行い、$rel\_coh(C)$ を用いて最良のクラスタ集合を 1 つ選択する手法を指す [11]。

- 隣接ベクトル

  隣接ベクトルのみを用いてクラスタリングする手法を指す。

- 文脈ベクトル

  文脈ベクトルのみを用いてクラスタリングする手法を指す。

- 連想ベクトル

  連想ベクトルのみを用いてクラスタリングする手法を指す。

- トピックベクトル

  トピックベクトルのみを用いてクラスタリングする手法を指す。

- BL

  ベースラインを表す。このシステムは凝集型クラスタリングアルゴリズムで併合するクラスタの組をランダムに選択し、これを停止条件を満たすまで繰り返す手法である。なお、ベースラインはランダムにマージする要素を選択するため、常に同じクラスタ集合が得られるわけではない。そこで、クラスタリングを 10 回試行し、各評価指標の平均値を BL の評価結果とした。

表 4.2: Purity,I-Purity,F-measure での各手法の平均値 (Tc=10)

手法	Purity	I-Purity	F-measure
提案手法 (正規化あり$_{SD}$)	0.8000	0.3865	0.5071
提案手法 (正規化あり$_{R}$)	0.7711	0.6187	0.6731
提案手法 (正規化なし)	0.7618	0.7099	0.7222
九岡ら	0.7514	0.7534	0.7400
隣接ベクトル	0.8114	0.5549	0.6377
文脈ベクトル	0.7500	0.7620	0.7446
連想ベクトル	0.7492	0.7342	0.7281
トピックベクトル	0.7649	0.5236	0.6065
BL	0.7450	0.3102	0.4279

表 4.2 から 4.7 の結果を順に考察する。

- 表 4.2 では、提案手法の中では「提案手法 (正規化あり$_{SD}$)」が Purity においてもっとも高い値を出した。なお、3 つの提案手法はいずれも九岡の手法よりも purity が高かった。しかし、全ての手法で比較を行うと、隣接ベクトルが Purity において最良の結果であった。「提案手法 (正規化あり$_{SD}$)」は Purity,I-Purity,F-measure の全てにおいて隣接ベクトルより低い値をとった。

表 4.3: Homogeneity,Completeness,V-measure での各手法の平均値 (Tc=10)

手法	Homogeneity	Completeness	V-measure
提案手法 (正規化あり$_{SD}$)	0.4715	0.1795	0.2385
提案手法 (正規化あり$_{R}$)	0.3573	0.1836	0.2195
提案手法 (正規化なし)	0.3083	0.1816	0.2031
九岡ら	0.2939	0.1837	0.1986
隣接ベクトル	0.4873	0.2281	0.2780
文脈ベクトル	0.2823	0.1789	0.1919
連想ベクトル	0.2853	0.1836	0.1979
トピックベクトル	0.3736	0.1697	0.2128
BL	0.3270	0.1143	0.1541

表 4.4: PP, PR, Paired F-score での各手法の平均値 (Tc=10)

手法	PP	PR	Paired F-score
提案手法 (正規化あり$_{SD}$)	0.6483	0.2026	0.2953
提案手法 (正規化あり$_{R}$)	0.5917	0.3959	0.4458
提案手法 (正規化なし)	0.5870	0.5090	0.5155
九岡ら	0.5862	0.5756	0.5521
隣接ベクトル	0.6784	0.3758	0.4401
文脈ベクトル	0.5820	0.5835	0.5571
連想ベクトル	0.5787	0.5507	0.5342
トピックベクトル	0.6198	0.3296	0.4019
BL	0.5748	0.1329	0.2094

表 4.5: Purity,I-Purity,F-measure での各手法の平均値 (Tc=15)

手法	Purity	I-Purity	F-measure
提案手法 (正規化あり$_{SD}$)	0.8256	0.3454	0.4766
提案手法 (正規化あり$_{R}$)	0.8004	0.5472	0.6396
提案手法 (正規化なし)	0.7978	0.6142	0.6858
九岡ら	0.7933	0.6091	0.6768
隣接ベクトル	0.8525	0.4360	0.5582
文脈ベクトル	0.7878	0.6446	0.6990
連想ベクトル	0.7901	0.6300	0.6922
トピックベクトル	0.8025	0.4052	0.5309
BL	0.7486	0.2754	0.3912

表 4.6: Homogeneity,Completeness,V-measure での各手法の平均値 (Tc=15)

手法	Homogeneity	Completeness	V-measure
提案手法 (正規化あり$_{SD}$)	0.5590	0.1860	0.2582
提案手法 (正規化あり$_{R}$)	0.4475	0.1857	0.2041
提案手法 (正規化なし)	0.4297	0.1884	0.2382
九岡ら	0.4475	0.1843	0.2345
隣接ベクトル	0.6141	0.2235	0.3008
文脈ベクトル	0.4293	0.1829	0.2272
連想ベクトル	0.4033	0.1861	0.2331
トピックベクトル	0.5000	0.1731	0.2395
BL	0.3771	0.1216	0.1688

表 4.7: PP, PR, Paired F-score での各手法の平均値 (Tc=15)

手法	PP	PR	Paired F-score
提案手法 (正規化あり$_{SD}$)	0.6621	0.1532	0.2398
提案手法 (正規化あり$_{R}$)	0.5869	0.3044	0.3758
提案手法 (正規化なし)	0.5905	0.3753	0.4359
九岡ら	0.5936	0.3770	0.4325
隣接ベクトル	0.7067	0.2445	0.3323
文脈ベクトル	0.5833	0.4138	0.4559
連想ベクトル	0.5837	0.3995	0.4511
トピックベクトル	0.6415	0.1992	0.2928
BL	0.5661	0.1051	0.1707

- 表 4.3 について Homogeneity に着目すると、提案手法は全て九岡の値を上回っている。しかし、表 4.2 と同じく隣接ベクトルが Homogeneity において最も高い値をとっている。

- 表 4.4 について PP の値に着目すると、提案手法は全て九岡の値を上回っている。しかし、隣接ベクトルが PP においても最も高い値であった。

- 表 4.5 について Purity に着目した場合、隣接ベクトルを用いる手法が最大となり、「提案手法 (正規化あり$_{SD}$)」がそれに次ぐ。また、トピックベクトルの順位が $Tc = 10$ のときよりも高くなった。「提案手法 (正規化あり$_{SD}$)」に次ぐ 3 番目に高い値であり、「提案手法 (正規化あり$_{R}$)」よりもよい結果を示している。

- 表 4.6 について Homogeneity に注目した場合も、各手法の優劣は表 4.5 と同じ結果である。

- 表 4.7 について PP に注目した場合も、各手法の優劣は表 4.5 と同じ結果である。

これらの結果から、Purity, Homogeneity, Paired F-score について比較すると、提案手法の中では偏差値を用いて正規化を行ったもの (「提案手法 (正規化あり$_{SD}$)」) が最も良い結果を示している。また、「提案手法 (正規化あり$_{SD}$)」は先行研究で用いられた $rel\_coh(C)$ で 4 つの特徴ベクトルの中から 1 つ選択するといった九岡の手法よりも上回っている。また「提案手法 (正規化あり$_{R}$)」についても $Tc = 15$ の PP について比較したもの以外は、九岡の手法を上回っている。

提案手法の中において Purity や Homogeneity といったクラスタの同質性で着目したときには「提案手法 (正規化あり$_{SD}$)」は最も高い値をとっている。しかし、I-Purity や Completeness といったクラスタの完全性については、提案手法の中ではもっとも低い値であった。ただし、前述のように、本研究では Puriy, Homogeneity, PP を重視している。新語義発見をのためには、「提案手法 (正規化あり$_{SD}$)」が最も適しているといえる。

また、単独のベクトルを用いてクラスタリングを行った場合、要素を多く含む巨大なクラスタと、他の要素と 1 度もマージされずに要素を 1 つしか持たないようなクラスタで構成されたクラスタ集合が生成される傾向にあった。多くの要素を含む巨大なクラスタが I-Purity, Completeness, PR の向上に、1 つの要素しか持たないクラスタが Purity, Homogeneity, PP の向上に、それぞれ貢献している。しかし、多くの要素から構成される大きなクラスタは様々な語義の用例が混在している可能性が高く、また 1 つの要素からなるクラスタは明らかに語義の判定には有用でない。新語義判定には、同じ語義を持つ要素を 2 つ以上まとめたクラスタが多く存在する状況が望ましい。したがって、1 つの要素から構成されるクラスタを除外した場合の精度で提案手法と一種類のベクトルを用いる手法を比較した。$Tc = 10$,$Tc = 15$ の 2 つの条件についての比較の結果を表 4.8,4.9 に示す。

表 4.8,4.9 での $|C|$ とはクラスタリング結果におけるクラスタの数を指し、$|C_{\geq 2}|$ とは C の中で要素を 2 つ以上含むクラスタの数を示す。$R_{\geq 2}$ は要素数 2 以上のクラスタ数 ($|C_{\geq 2}|$) の全クラスタ数 ($|C|$) に対する比として、式 (4.15) によって定義される。$|C|$ の値は各手法によって差が大きい場合があるため、2 つ以上の要素を含むクラスタが占める割合 ($R_{\geq 2}$) で各手法を比較する。

$$R_{\geq 2} = \frac{|C_{\geq 2}|}{|C|} \tag{4.15}$$

また、AP は 2 つ以上の要素を含むクラスタについての最大適合率の平均を表す。ここでの最大適合率 (max_prec) とはクラスタの中で最多の語義が占める割合である。AP は式 (4.16) で定義される。

$$AP = \frac{1}{|C_{\geq 2}|} \sum_{C_i \in C_{\geq 2}} max\_prec(C_i) \tag{4.16}$$

表 4.8 の結果では、「提案手法 (正規化あり$_{SD}$)」と「提案手法 (正規化あり$_{R}$)」の場合は、単独のベクトルよりも $R_{\geq 2}$ が高い値を示している。これは、新語義判定に用いることのできない 1 要素で構成されるようなクラスタが少ない事を意味している。また、要素が 2 つ以上あるクラスタについての最大適合率を表す AP は、「提案手法 (正規化あり$_{SD}$)」、

表 4.8: 1 要素のクラスタを除外した場合の最大適合率 (Tc=10)

手法	$\|C\|$	$\|C_{\geq 2}\|$	$R_{\geq 2}$	AP
提案手法 (正規化あり$_{SD}$)	396	347	0.868	0.828
提案手法 (正規化あり$_{R}$)	400	258	0.645	0.857
提案手法 (正規化なし)	400	145	0.363	0.834
隣接ベクトル	400	211	0.528	0.819
文脈ベクトル	400	99	0.248	0.758
連想ベクトル	400	103	0.258	0.772
トピックベクトル	400	233	0.583	0.767

表 4.9: 1 要素のクラスタを除外した場合の最大適合率 (Tc=15)

手法	$\|C\|$	$\|C_{\geq 2}\|$	$R_{\geq 2}$	AP
提案手法 (正規化あり$_{SD}$)	548	396	0.723	0.780
提案手法 (正規化あり$_{R}$)	600	280	0.467	0.796
提案手法 (正規化なし)	600	156	0.260	0.760
隣接ベクトル	600	271	0.452	0.782
文脈ベクトル	600	120	0.200	0.732
連想ベクトル	600	118	0.197	0.725
トピックベクトル	600	285	0.483	0.734

「提案手法 (正規化あり$_R$)」ともに単独のベクトルを用いるものよりも高い値であった。また、提案手法 (正規化なし) は要素 2 以上のクラスタ数 $|C_{\geq 2}|$ が、単独のベクトルのものと大差なかった。これは正規化されていないがために、高い類似度平均を持つ連想ベクトルが多く選択され、また同じ種類のベクトルが選択されやすいがために、複数の特徴ベクトルを用いる効果が薄く、連想ベクトルのみを用いるものに近い結果が得られたためであると考えられる。しかし、表 4.9 の結果では、$Tc = 10$ の結果と $Tc = 15$ では、$R_{\geq 2}$ の値においてトピックベクトルが「提案手法 (正規化あり$_R$)」の値よりも上回っている。また、AP の値について比較すると、隣接ベクトルは「提案手法 (正規化あり$_{SD}$)」を上回っている。ただし、$R_{\geq 2}$ については「提案手法 (正規化あり$_{SD}$)」が、AP については「提案手法 (正規化あり$_R$)」がそれぞれ最大の値である。この傾向は $Tc = 10$,$Tc = 15$ の 2 つの条件について、ともに共通している。

4.2.5 特徴ベクトルの貢献度に対する考察

提案手法は、複数の観点でクラスタリングをすることを狙いとする。1 種類の特徴ベクトルばかりを選択しているのでは、この狙いは達成されているとは言えない。

本項では、クラスタリングの過程で 2 つのクラスタを併合して新しいクラスタを作成する際に、クラスタ間の類似度の計算に用いられた特徴ベクトルの回数を調べる。ここでは選択された回数が多いほどクラスタリングに対する貢献度が高いと考える。4 つの特徴ベクトルの貢献度が均一であれば、本論文での狙いが達成されていると考えられる。

3.2.2 項で述べた正規化の各手法と正規化を行わない手法について、ベクトルが選択された回数を表 4.10,4.11,4.12 に示す。表中の数値はクラスタリング時に特徴ベクトルが選択された回数を表す。なお、表 4.10, 4.11, 4.12 について、対象単語を名詞、動詞、形容詞の順にグループ分けを行い、表の最後に品詞別に見た特徴ベクトルの貢献度と、全ての単語に対する貢献度を示した。カッコ内の値は、全体に対する割合を示している。

これらの表から見てとれることは、多少の差異こそあれど、品詞によって選択されやすいベクトルが存在すると断言できない点である。逆に、提案手法は品詞によって影響を受けないことから、新語義発見に対して、品詞を選ばず効果を発揮することが出来ると予想できる。

また、相対値を用いて正規化を行った場合と、正規化を行わない場合は連想ベクトルが選択されやすい事が分かる。3 章の表 3.2 で示した通り、連想ベクトルは類似度平均が他のものよりも大きく上回っているためである。これに対し、偏差値を用いて正規化を行っ

表 4.10: 選択されたベクトルの種類の内訳 (組み合わせ正規化あり [偏差値])

品詞	単語	隣接	文脈	連想	トピック
名詞	相手	26	2	1	11
	場合	15	6	0	19
	場所	18	11	3	8
	文化	16	11	0	13
	電話	22	12	0	6
	現場	23	9	0	8
	技術	7	23	0	10
	はじめ	1	15	0	16
	意味	22	9	0	9
	可能	11	12	0	17
	関係	15	21	1	3
	経済	6	22	0	12
	子供	23	7	0	11
	時間	13	9	0	18
	市場	18	19	0	3
	社会	15	20	0	5
	情報	12	19	0	9
	手	24	11	0	6
	前	27	8	0	6
動詞	与える	24	0	0	15
	出す	23	1	0	14
	出る	14	1	0	16
	生きる	5	7	1	26
	教える	18	9	0	12
	考える	17	4	0	19
	進める	12	9	0	10
	する	15	13	1	11
	立つ	20	5	0	9
	乗る	14	8	0	8
	始める	10	8	0	13
	開く	23	11	0	4
	見える	9	4	0	22
	認める	21	15	0	4
	持つ	10	0	0	30
	求める	5	18	3	14
形容詞	早い	10	6	2	15
	大きい	18	15	0	6
	高い	12	10	0	17
	強い	19	8	0	12
	良い	15	6	2	12
	合計 (名詞)	314(0.416)	246(0.326)	5(0.007)	190(0.252)
	合計 (動詞)	240(0.410)	113(0.193)	5(0.009)	227(0.388)
	合計 (形容詞)	74(0.400)	45(0.243)	4(0.022)	62(0.335)
	総合計	628(0.412)	404(0.265)	14(0.009)	479(0.314)

表 4.11: 選択されたベクトルの種類の内訳 (組み合わせ正規化あり [相対値])

品詞	単語	隣接	文脈	連想	トピック
名詞	相手	1	6	31	2
	場合	5	1	28	6
	場所	3	2	28	7
	文化	3	2	31	4
	電話	7	1	31	1
	現場	7	1	26	6
	技術	4	0	31	5
	はじめ	4	0	26	2
	可能	3	1	24	12
	関係	3	1	32	4
	経済	3	1	24	12
	子供	2	2	34	2
	時間	1	5	31	3
	市場	1	0	36	3
	社会	2	2	31	5
	情報	3	1	34	2
	手	4	1	32	2
	前	1	0	37	2
動詞	与える	1	1	23	14
	出す	1	4	30	3
	出る	6	0	22	3
	生きる	2	0	27	10
	教える	1	1	31	6
	考える	3	0	33	4
	進める	11	0	19	1
	する	1	0	36	3
	立つ	1	0	31	2
	乗る	2	1	26	1
	始める	4	0	23	4
	開く	11	3	22	2
	見える	1	2	31	1
	認める	3	0	30	7
	持つ	17	2	19	2
	求める	18	0	11	11
形容詞	早い	1	0	22	10
	意味	1	0	32	7
	大きい	0	1	34	4
	高い	9	0	30	0
	強い	0	5	30	4
	良い	1	1	30	3
	合計 (名詞)	57(0.080)	27(0.024)	547(0.769)	80(0.070)
	合計 (動詞)	83(0.142)	14(0.024)	414(0.708)	74(0.126)
	合計 (形容詞)	12(0.053)	7(0.031)	178(0.781)	28(0.124)
	総合計	152(0.1000)	48(0.032)	1139(0.749)	182(0.118)

表 4.12: 選択されたベクトルの種類の内訳 (組み合わせ正規化なし)

品詞	単語	隣接	文脈	連想	トピック
名詞	相手	0	0	40	0
	場合	0	0	40	0
	場所	2	1	37	0
	文化	0	0	40	0
	電話	0	0	40	0
	現場	6	0	34	0
	技術	0	0	40	0
	はじめ	3	0	29	0
	意味	1	0	39	0
	可能	3	0	37	0
	関係	0	0	40	0
	経済	3	1	36	0
	子供	0	0	40	0
	時間	1	0	39	0
	市場	1	0	39	0
	社会	3	0	37	0
	情報	3	0	37	0
	手	0	0	39	0
	前	1	0	39	0
動詞	与える	2	0	37	0
	出す	1	0	37	0
	出る	1	1	29	0
	生きる	4	0	35	0
	教える	1	2	36	0
	考える	3	0	37	0
	進める	0	1	30	0
	する	1	0	39	0
	乗る	1	0	29	0
	始める	2	0	29	0
	開く	0	0	38	0
	見える	1	0	34	0
	認める	3	0	37	0
	持つ	0	0	40	0
	求める	18	0	22	0
形容詞	早い	0	0	33	0
	大きい	0	0	39	0
	高い	0	0	39	0
	立つ	1	0	33	0
	強い	1	0	38	0
	良い	0	0	35	0
	合計 (名詞)	27(0.036)	2(0.003)	722(0.961)	0(0.000)
	合計 (動詞)	38(0.069)	4(0.007)	509(0.924)	0(0.000)
	合計 (形容詞)	2(0.009)	0(0.000)	217(0.991)	0(0.000)
	総合計	67(0.044)	6(0.004)	1448(0.952)	0(0.000)

た場合、隣接ベクトル、文脈ベクトル、トピックベクトルが比較的に万遍なく選択されており、しかも連想ベクトルは選択されにくい傾向にある。これは、連想ベクトルの標準偏差がとても小さく、偏差値の値が大きくならないことが原因である。

特徴ベクトルの貢献度という面においては、偏差値を用いて正規化を行ったものは比較的に均整が取れている。また、相対値を用いたものは連想ベクトルが極端に選ばれる状況は回避できているが、連想ベクトルが選択される割合が約 75%と、連想ベクトルのクラスタリングに対する貢献度が非常に高い。これに対して、正規化を行わないものは、連想ベクトルの貢献度が非常に高く、連想ベクトルが選択された割合は約 95%であった。したがって、ベクトル間の類似度を正規化することにより、複数の特徴ベクトルを同時に用いるという本研究の狙いが達成されていると考えられる。

しかし、本実験の対象単語において、「生きる」、「市場」、「社会」といった単語では、連想ベクトルが Purity, Homogeneity などの同質性の面において、他の特徴ベクトルよりも最良のクラスタ集合を生成した。したがって、連想ベクトルが選択されることが有効な場面もある。また、このようなケースにおいては、相対値で正規化を行った手法が、偏差値の正規化を行った手法よりも高い評価結果を示した。したがって、連想ベクトルが一概に不要とは言い切れない。

参考のため、評価単語ごとに特徴ベクトルを一つ選択する九岡・田中の手法について、それぞれの特徴ベクトルが選択された回数から、同様に特徴ベクトルの貢献度を調べた。対象単語 40 語に対し、$Tc = 10$,$Tc = 15$ のそれぞれの場合について、特徴ベクトルが選択された回数を表 4.13 に示す。

表 4.13: $rel\_coh(C)$ で選択されたベクトルの種類の内訳

Tc	隣接	文脈	連想	トピック
10	2	32	0	6
15	1	34	0	5

表 4.13 から選択されるベクトルに偏りがあるということが分かる。特に文脈ベクトルが選択される事が多く、逆に連想ベクトルが一度も選択される事がなかった。

4.2.6　クラスタラベルの有効性に関する考察

提案手法では複数の特徴ベクトルを用いているが、1 つのクラスタは同じ種類の特徴ベクトルに注目して作成する制約を設けている。この制約は、クラスタラベルという概念を導入し、一度クラスタラベルが決まれば、以降は同じラベルを持つものかクラスタラベルの定まっていないクラスタとしかマージできないという処理で実現されている。クラスタラベル L は特徴ベクトルの種類 { 隣接, 文脈, 連想, トピック } を指しており、あるクラスタがどの特徴ベクトルの類似度が高い用例をまとめたものであるかを表している。しかし、マージに関する制約を設けずに、4 つの特徴ベクトルのうち、どれか 1 つでも類似度が大きいクラスタの組を優先的にマージし、同じクラスタの中に異なる観点で似ているとみなされた用例が混在しても、結果的に良いクラスタ集合が作成できるという考えもある。

そこで、本項ではクラスタラベルを適用する場合と、クラスタラベルを適用しない場合とを比較する実験を行った。その結果を表 4.14,4.15,4.16 にて示す。なお、第 2 列目の [Y][N] によってクラスタラベルが適用されているのか、適用されていないのかを区別している。

- Y:クラスタラベル適用あり
- N:クラスタラベル適用なし

表 4.14,4.15,4.16 より、クラスタラベルを用いるものと用いないものの間に大きな差は見当たらなかったが、ほとんどのケースでクラスタラベルを適用したものが高い評価値を出している。ただし、僅差でクラスタラベルを適用しないものがラベルを適用したものを上回ったケースが存在する。$Tc = 10$ の「正規化あり$_{SD}$」の手法では、Purity はクラスタラベルを適用しない方がクラスタラベルを用いる場合を上回る。しかし、その差は 0.005 と非常にわずかである。$Tc = 15$ の場合、$Tc = 10$ とは優劣が異なった場面がある。「正規化あり$_{SD}$」の手法で、Purity, Homogeneity を比較した場合、大きな差はないがラベルなしの手法が高い。しかし、PP はクラスタラベルありの方が評価値は高かった。また、I-Purity や Completeness といった完全性での評価尺度では、$Tc = 10$,$Tc = 15$ ともにクラスタラベルを適用しなかったものの方が高い傾向にあった。クラスタラベルが同じ場合のみクラスタをマージするといった制約を与えると、多くの要素を含む大きいクラスタが作成されにくくなる。逆に制約がない場合は、サイズの大きいクラスタが作成されやすく、I-Purity や Completeness といった値が高くなっていると考えられる。

クラスタラベルを適用しない方が、良いクラスタ集合を生成できた場合もある。たとえば、「文化」という評価単語に対し、「正規化あり$_{SD}$」の手法でクラスタリングを行った

表 4.14: クラスタラベルの有無についての比較 (Purity,I-Purity,F-measure)

$Tc = 10$	L	Purity	I-Purity	F-measure
提案手法 (正規化あり$_{SD}$)	Y	0.8000	0.3865	0.5071
提案手法 (正規化あり$_{SD}$)	N	0.8005	0.4027	0.5223
提案手法 (正規化あり$_{R}$)	Y	0.7711	0.6187	0.6731
提案手法 (正規化あり$_{R}$)	N	0.7582	0.7587	0.7005
提案手法 (正規化なし)	Y	0.7618	0.7099	0.7222
提案手法 (正規化なし)	N	0.7537	0.7375	0.7319
$Tc = 15$	L	Purity	I-Purity	F-measure
提案手法 (正規化あり$_{SD}$)	Y	0.8256	0.3454	0.4766
提案手法 (正規化あり$_{SD}$)	N	0.8336	0.3464	0.6374
提案手法 (正規化あり$_{R}$)	Y	0.8074	0.5273	0.6295
提案手法 (正規化あり$_{R}$)	N	0.7925	0.6224	0.6841
提案手法 (正規化なし)	Y	0.7978	0.6142	0.6858
提案手法 (正規化なし)	N	0.7909	0.6366	0.6947

表 4.15: クラスタラベルの有無についての比較 (Homogeneity,Completeness,V-measure)

$Tc = 10$	L	Homogeneity	Completeness	V-measure
提案手法 (正規化あり$_{SD}$)	Y	0.4715	0.1794	0.2385
提案手法 (正規化あり$_{SD}$)	N	0.4705	0.1834	0.2031
提案手法 (正規化あり$_{R}$)	Y	0.3573	0.1836	0.2195
提案手法 (正規化あり$_{R}$)	N	0.3219	0.2053	0.2197
提案手法 (正規化なし)	Y	0.3083	0.1816	0.2031
提案手法 (正規化なし)	N	0.2874	0.1880	0.2032
$Tc = 15$	L	Homogeneity	Completeness	V-measure
提案手法 (正規化あり$_{SD}$)	Y	0.5590	0.1860	0.2582
提案手法 (正規化あり$_{SD}$)	N	0.5729	0.1898	0.2650
提案手法 (正規化あり$_{R}$)	Y	0.4780	0.1870	0.2478
提案手法 (正規化あり$_{R}$)	N	0.4265	0.1942	0.2410
提案手法 (正規化なし)	Y	0.4297	0.1884	0.2382
提案手法 (正規化なし)	N	0.4065	0.1904	0.2364

表 4.16: クラスタラベルの有無についての比較 (PP,PR,Paired F-score)

$Tc = 10$	L	PP	PR	Paired F-score
提案手法 (正規化あり$_{SD}$)	Y	0.6483	0.2026	0.2953
提案手法 (正規化あり$_{SD}$)	N	0.6423	0.2176	0.3116
提案手法 (正規化あり$_{R}$)	Y	0.5917	0.3959	0.4458
提案手法 (正規化あり$_{R}$)	N	0.5914	0.5871	0.5588
提案手法 (正規化なし)	Y	0.5870	0.5090	0.5155
提案手法 (正規化なし)	N	0.5845	0.5583	0.5411
$Tc = 15$	L	PP	PR	Paired F-score
提案手法 (正規化あり$_{SD}$)	Y	0.6621	0.1532	0.2398
提案手法 (正規化あり$_{SD}$)	N	0.6611	0.1580	0.2477
提案手法 (正規化あり$_{R}$)	Y	0.5946	0.2845	0.3660
提案手法 (正規化あり$_{R}$)	N	0.5927	0.4937	0.4487
提案手法 (正規化なし)	Y	0.5905	0.3753	0.4359
提案手法 (正規化なし)	N	0.5661	0.4067	0.4535

ときの Homogeneity は、クラスタラベルを適用したときが 0.6410 であったのに対し、クラスタラベルを適用しなかったものは 1.000 という評価結果であった。しかし、全単語の平均を見ると、いくつかの例外的なケースはあるが、クラスタラベルを適用する方がしない手法よりも、Purity, Homogeneity, PP が高かった。

これらの考察から、クラスタラベルを使うことによって、クラスタリングの性能が劣る事はなかった。また、全体的にはクラスタラベルの制約を設けることで、作成された用例クラスタがどのような観点で似ている用例をまとめたものかが明確になる。したがって、クラスタラベルを適用する手法はクラスタリングの精度を向上させ、またクラスタが生成された理由 (どのような観点で似ている用例をまとめたクラスタであるのか) が理解しやすくなることから、有効であると言える。

第5章　おわりに

本論文では、同じ語義をまとめるため語義識別のタスクにおいて、クラスタリングの段階で複数の観点を同時に用いることで、クラスタリングの性能を向上させる手法を提案した。本章では、本研究で得た知見と今後の課題について述べる。

5.1　まとめ

提案手法を以下にまとめる。

1. 特徴ベクトルの作成
   (a) 九岡によって提案された 4 種類の特徴ベクトルを用いた。また、隣接ベクトルについては、前後の 2 語の単語と品詞をベクトルの要素とし、品詞の重みは単語の重みに比べて 1/2 となるように改良を行った。
2. クラスタリング
   (a) 複数の特徴ベクトルを同時に用いてクラスタリングを行った。
   (b) 特徴ベクトルの類似度を公平に比較するために、ベクトル間類似度の正規化を行った。
   (c) クラスタの数、クラスタのサイズの両方を考慮したクラスタリングの停止条件を設定した。

実験の結果、Purity, Homogeneity, Paired Precision の評価基準において、対象単語 40 語の平均で比較した場合、隣接ベクトルが最も高い値であった。その原因を調査したところ、隣接ベクトル単独によるクラスタリングは以下の問題点があることが分かった。

- 多くの要素を含む巨大なクラスタが一つ完成し、停止条件を満たす。
- 初期状態から一度もマージされなかったクラスタが多く残る。

これらの状態によって評価値が向上していた。クラスタ内で最多の語義が占める割合を最大適合率として、2つ以上の要素数を持つクラスタに対して最大適合率を求めた場合、提案手法は単独のベクトルを用いるよりも最大適合率が高かった。また、2つ以上の要素を持つクラスタの数は、提案手法と単独のベクトルとを比較した場合、提案手法の方が1.5倍程度多い。この点から、提案手法は新語義の判定の前処理としての用例クラスタリング手法として有効であると考えられる。

また、研究の目的でも述べたように、語義の類似性は様々な観点から認識されるが、複数の特徴ベクトルを同時に考慮するということは同じ語義を持つ用例をまとめてクラスタを作成するための有効な手段であると分かった。さらに、クラスタラベルを適用した場面と適用しない場面について、精度の違いは微々たるものであり、クラスタ数 $10(Tc = 10)$ のときはクラスタラベル適応する手法の方がクラスタラベルを適用しない手法をわずかに上回っている。クラスタ数 $15(Tc = 15)$ という実験設定において、偏差値を用いて正規化を行う手法についてはクラスタラベルを適用させない方が高い精度を出しているが、その差は小さい。I-Purity や Completeness といった完全性の尺度では、ラベルを適用しない方がラベルを適用させているものよりも、高い精度を出している。これは、一度ラベルが定まってしまったクラスタは、同じラベルを持つクラスタか、ラベルの定まっていないクラスタとしかマージすることが出来ないことから、クラスタラベルによって多くの要素を持つクラスタが作成されにくくなっていることが原因と考えられる。これらの結果から、クラスタラベルの制約によって一つのクラスタが大きくなりにくくなり、1つの要素しか持たないクラスタが生成されにくくなると考えられる。したがって、新語義や希少語義といったものの識別や判別にはクラスタラベルの適用が有効である。

5.2 今後の課題

本研究ではクラスタリングの精度を向上させる点に重点を置いており、新語義発見に有効であると思われるクラスタ集合を生成することができた。しかし、本研究は新語義発見というタスクについては取り組んでおらず、提案するクラスタリング手法によって新語義発見が可能であるとは言いきることが出来ない。したがって、今後の課題としては、作成されたクラスタについて辞書の語義との比較を行い、用例クラスタを辞書の語義に正しく対応付けることができるか、また新語義のクラスタを正しく検出できるかを確認する必要がある。田中 [12] や Richard ら [4] の研究のように新語義かどうかの判断基準を考案することが新語義の特定における課題であると考えられる。

また、本研究では複数の特徴ベクトルを同時に用いる際に、4つの特徴ベクトルの最大値を用例間の類似度としている。しかし、特徴ベクトルを組み合わせて使う方法は改善の余地があると考えられる。九岡は用例間の類似度を4つの特徴ベクトルの類似度の重み付け和で定義することによって4つの特徴ベクトルを同時に用いる方法を試した、しかし、本研究のように単独のベクトルでクラスタリングを行った結果に対して大きな差が出たわけではない [11]。本研究での正規化の手法別にベクトルの選択率 (貢献度) をみると、偏差値を用いた手法は標準偏差が大きい隣接ベクトルやトピックベクトルが選択されやすい。また、相対値を用いた正規化の手法や正規化を行わない手法では、全体の類似度平均が高かった連想ベクトルが選択されやすい。これらの結果から、正規化の手法によって選択される特徴ベクトルの傾向が異なることが確認できた。単語別で各手法の比較を行った場合、単語によって正規化手法の優劣が異なっている。したがって、現在はどちらの正規化が良いかということを断言出来ない。しかし、全ての特徴ベクトルが提案手法よりも万遍なく選択されるような正規化の手法、あるいは九岡のように、複数の正規化の手法によって作成された複数のクラスタ集合の中から最良のクラスタ集合を選択するといった手法を用いた場合には、語義識別の精度も一層向上すると予想できる。

さらに、本研究では、隣接ベクトルを除いて先行研究の特徴ベクトルをそのまま用いた。しかし、隣接ベクトルのウィンドウ幅の改良のように、特徴ベクトル自体の改良も必要である。特徴ベクトルの改良によって、インスタンスをより正確に特徴づけることができれば、語義識別の精度に向上すると考えられる。現在考案している手法は、連想ベクトルの改良である修正連想ベクトルである。連想ベクトルは単語の二次共起を用いて作成される特徴ベクトルであったが、対象単語と共起した単語との距離は考慮していない。式 (5.1) で定義される修正連想ベクトル $\vec{ad}_i$ は、対象単語と周辺語との距離を考慮するように連想ベクトルを改良したものである。

$$\vec{ad}_i = \sum_{c_j \in context} \frac{1}{d} \vec{o}(c_j) \tag{5.1}$$

ここでの c_j とは対象単語の周辺に出現した単語を指し、$\vec{o}(c_j)$ とはコーパスでの出現頻度上位 10000 語と対象単語との二次共起ベクトルである。そして、対象単語と c_j との距離を d と定義し、距離に反比例した重みづけを $\vec{o}(c_j)$ に与える。これにより、対象のインスタンスを九岡の考案した連想ベクトルよりもより正確に用例の特徴をベクトルとして表現できると考えている。

謝辞

本研究を進めるに当たって、白井清昭准教授、島津明教授、中村誠助教、Nguyen Minh Le 助教は数多くのご教示を頂きました。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頂きました。そして、4 年次編入入学での就学の際、中京大学田中穂積研究室・白井英俊研究室の皆様には数多くのご支援を頂きました。この場を借りて感謝申し上げます。

参考文献

[1] Eneko Agirre, David Martínez,Oier López de Lacalle and Aitor Soroa. Two graph-based algorithms for state-of-the-art WSD . EMNLP2006, pp.585-593, July 2006.

[2] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based extarnal cluster evaluation . Proceeding of the 2007 joint Conference on EMNLP, pp.410-420, June 2007.

[3] Hinrich Schütze. Automatic word sense discrimination , Computational linguistics Vol.24 No.1, pp.97-123, 1998.

[4] Richard Schwarz, Hinrich Schütze, Fabienne Martin, Achim Stein. Identification of Rare & Novel Senses Using Translations in a Parallel Corpus . LERC 2010, pp.2249-2252, June 2010.

[5] David M.Blei, Andrew Y.Ng, Michael I.Jordan . Latent Dirichlet Allocation . (2003) 993-1022 Journal of Machine Learning Research 3, pp.993-1022, 2003.

[6] Thomas Hofmann. Probabilistic Latent Semantic Indexing . In SIGER '99:Proceedingsof the 22nd annual international ACM SIGER conference on Research and development in information retrieval, pp.50-57, ACM press ,1999.

[7] Suresh Manandhar, Ioannis P.Klapaftis, Dmitry Dligach, Sameer S.Pradhan. SemEval-2010 Task 14: Word Sense Induction & Disambiguation . Proceeding of the IWSE,ACL 2010, pp.63-68, 15-16 July 2010.

[8] Manabu Okumura, Kiyoaki Shirai, Kanako Komiya and Hikaru Yokono. SemEval-2010 task: Japanese WSD. In Proceedings of SemEval-2010, pp.69-74, 2010.

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

[10] Sergey Brin and Lawrence Page. The anatomy of a largescale hypertextual web search engine. Computer Networks and ISDN Systems Volume 30, Issues 1-7, pp.107-117, 1998.

[11] 九岡佑介. 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情報処理専攻,3 2008.

[12] 田中博貴. 用例のクラスタリングに基づく単語の新語義の発見 . 修士論文,北陸先端科学技術大学院大学情報科学研究科情報処理専攻,3 2009.

[13] 前川喜久雄. 特定領域研究「日本語コーパス」のめざすもの . 「日本語コーパス」全体会議総括班報告,09.09 2006.

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