平成 20 年度茨城大学工学部情報工学科卒業研究論文

シソーラスを利用した文書共クラスタリング

平成21年2月10日

工学部情報工学科

執筆者:豊川幸秀(05T4051R)

指導教員:新納浩幸 准教授

平成20年度　茨城大学工学部情報工学科　卒業研究

シソーラスを利用した文書共クラスタリング

著者：豊川 幸秀（05T4051R）
指導教員：新納浩幸 准教授

論文要旨

近年、インターネットの普及により、情報検索手法の重要性がますます高まっている。そこでは検索時間の短縮や、検索結果表示時の見易さなどを改善することが求められている。これを解決する一つの技術として、文書の集合を関連性のあるもの同士でグループ化する文書クラスタリングの研究が行われている。

クラスタリングの手法は多岐にわたるが、文書クラスタリングに対して特徴的なCo-clustering（共クラスタリング）という手法がある。通常の文書クラスタリングにおいて各文書データは、各次元に単語を対応させたベクトル空間モデルで表現され、文書間の類似度などによって分類されるのだが、既存の多くの手法では文書視点のみでしかクラスタリングできない。一方共クラスタリングは、文書集合をクラスタリングすることと、その文書集合で利用されている単語をクラスタリングすることが双対の関係になっていることを利用したものであり、文書と単語を同時にクラスタリングすることでクラスタリングの精度を高めるといった特徴がある。

ただし共クラスタリングにより文書と単語を同時にクラスタリングしても、精度の高いクラスタリング結果はなかなか得られない。一つの原因は単語集合のクラスタリングが不完全であるためである。しかし単語の集合をクラスタリングした結果は、シソーラスという形で既に存在している。このため共クラスタリングの処理の中で、単語のクラスタリングを既存のシソーラスから得て、それをもとに文書をクラスタリングすることで、全体としての精度向上が期待できる。ここでは既存のシソーラスとして分類語彙表を利用し本手法の有効性を確認する。

実験では、本手法とクラスタリングの標準手法である k-means との精度の比較を行った。その結果、オリジナルの共クラスタリングからは改善が見られたが、k-means と比較すると改善は見られなかった。その原因としては利用したシソーラスがこのタスクに対して適切でないことが考えられる。今回は分類語彙表を利用したが、これは人間用のシソーラスであり、機械処理には向いていない。文書クラスタリングに特化した機械処理用のシソーラスを構築することで、更なる精度改善が図れると考える。

目次

第１章　はじめに

１．１　背景

情報検索の分野において、書籍や新聞などの文書の集合を内容的に似通ったいくつかの集合に分けるための、文書クラスタリングの研究が長年試みられてきた。これを応用すると、関連性がある情報をグループ化しておくことにより、検索時間の短縮や、検索結果表示時に見やすくなるなど、有益な活用方法が見出せる。

文書クラスタリングを行う際に用いる手法には様々なものがあるが、その中の一つの手法としてCo-clusteringが挙げられる。後にも述べるが、通常、文書クラスタリングにおいて各文書データは、単語を重みとしたベクトルで表現され、文書間の類似度などによってクラスタリングされるのだが、Co-clusteringを用いることにより、単語と文書の両面から同時にクラスタリングすることができ、関連性が深いデータ同士をより正確にグループ化できることが望まれる。ただ現状では、前者のようなクラスタリング手法のほうが多いことが事実である。

そこで本実験では、シソーラスを利用してあらかじめ単語をクラスタリングしておき、そのクラスタを用いて各文書データをベクトル化し、文書クラスタリングすることにより、既存のクラスタリング手法を用いたCo-clusteringを可能とし、またより精度の高い結果を導いていこうとすることが目的である。

１．２　本論文の構成

第２章：クラスタリングとは何か、その中における文書クラスタリングの導出方法、また導出時に用いる各手法の簡単な紹介や、クラスタリング結果の評価方法についての説明をしていく

第３章：Co-clusteringの概念と、クラスタリング方法を2部グラフを用いてイメージとして説明する

第４章：シソーラスの詳細及び、それを使ったCo-clusteringにおける利点などを説明する

第５章：実際のデータを用いて、k-means、Co-clustering、シソーラスを用いた文書共クラスタリングの各手法におけるクラスタリング結果を導き出し、評価していく

第６章：第５章の結果を元にそれぞれを比較し、各手法のメリット・デメリットや、改善点などについて考察する

第７章：今回の研究の結果の活用方法や、今後の課題についてまとめる

第2章　文書クラスタリング

2．1　クラスタリングとは

クラスタリングとはデータ解析の一つの方法で，事前に基準を定義するのではなく，1つの多様な集団をいくつかの似ているものでグループ化する手法のことである。クラスタリングによってできたグループをクラスタと呼ぶ。似ているかどうかの指標には，類似度や距離を利用する。類似度とは，個体を2つ与えたときに，その対に対して決まる実数であり，その値が大きいほど2つの個体は似ていると仮定される。また，距離とは非類似度とも言われ，その値が小さいほど似ているとされる。距離が指標となった場合には，同じクラスタ内においては距離が小さく，異なるクラスタにおいては距離が大きくなければならない。クラスタリングが用いられる分野は，心理学，社会学，認知科学から，経営分析，マーケティングまで様々である。クラスタリング手法はそれらの分析や用途によって様々なものが提唱されているが，ここでは，図書や雑誌論文などの文書に対するクラスタリングに焦点をあてる。これが文書クラスタリングである。

2．2　文書クラスタリングの手順及び具体的手法

実際の文書クラスタリングの手順は、以下のようになる。

1．与えられた文書に対し、形態素解析を行い、語に分割する。

2．単語をリスト化し番号をつけ、文書を単語の重みからなるベクトルとして表現する。

3．文書間の類似度を定義する

4．各手法に従って、クラスタリングを行う

○形態素解析

先程の手順でも述べたように、文書クラスタリングでは文書を表現するために、文書を構成する語の重みからなるベクトルを用いることがほとんどである。語というのは、意味を持つ最小の単位である形態素のことを示しており、実際に分割する作業のことを形態素解析という。この作業は日本語にとっては重要な意味をもっている。というのも、日本語では単語ごとに区切らず続けて書くために、形態素ごとの分割が難しいためである。一方、英語などはもともと語と語に区切りがあるため，形態素解析の必要はない。

日本語における形態素解析には様々な問題が内包されているが、主に以下の4点のような問題が挙げられる。

1．単語の境界判別の問題：

ある一つの文が与えられた際、その単語の区切り方に複数通りの方法があるとすると、それぞれ意味が大きく変わってくる場合がある。このように複数の区切り方の候補がある場合には、経験的な知識による人間らしい判断が必要になることがあるが、コンピューターなどの計算機を用いた解析では、文法や単語の辞書的データを超えるようなそういった知識も導入したとすると必要な知識が膨大に用意される必要があるため、実現は非常に困難である。

2．品詞判別の問題：

この問題においては日本語よりもむしろ英語で顕著である。英語では同じ単語において複数の意味を持つことが多々あるが、その文における正確な意味を捉え損ねると、文法的構造や導かれる意味がまったく違うものになってしまう。品詞を見分けることは形態素解析の次の段階である構文解析にとって非常に重要であるが、英語では品詞の種類が文の構造と密接に関連しているため、これらを同時に行う方法も研究されている。

３．未知語の問題：

通常、形態素解析は辞書を用いておこなわれるが、解析対象に辞書に載っていない単語が出てくることもある。これを未知語と呼ぶが、日本語では漢字の列やカタカナの列はたとえ未知語であってもある程度単語として認識することができる。しかしそれが使えない場合、代表的な方法は「知っている単語が現れるまでよみとばす」というものだが、これは後の解析を狂わせてしまい、解析結果の精度低下につながってしまう。

４．ルーズな文法の問題：

解析対象の文書が、常に標準的な文法で書かれているとは限らない。例えば話し言葉や、電子メール等に用いられる略語やフェイスマーク、方言などが挙げられる。またこういった文は校正が不十分なため、書き手の誤りが入っている場合が多くある。このような文に対応するためには、正しい文が入力されるという前提の設計に基づく現在の形態素解析の手法を、誤りが含まれる様な文にも対処可能なように根本から見直す必要があるが、言語資源の不足のためあまり研究はされていない。

形態素解析には様々な手法があり，以上のような問題点を解決するために現在も研究が続けられている。主に挙げられる形態素解析のツールとしては，ＣｈａＳｅｎ(茶筌)、ＫＡＫＡＳＩ、Ｓｅｎ、Ｍｅｃａｂ(和布蕪)などがある。

次に具体的な手法について述べる。一般的に、クラスタリング手法は「階層的クラスタリング」もしくは「非階層的クラスタリング」の２つに大別できる。この２つの手法は、文書クラスタリングにおいても有効で、以下のような特徴がある。

○階層的クラスタリング

与えられたデータセット（ここでは単語の重みによりベクトル表現された文書セット）の各データが、それぞれ１つのクラスタとなっている状態を初期状態として、クラスタ間の距離や類似度に基づいて、各クラスタを逐次的に併合してゆく手法である。一般的に階層的クラスタリングは、文書量が増えるにつれ類似度計算が膨大になるケースが多い点から、文書クラスタリングには適さないとされているが、大規模文書に適するような階層的クラスタリングも研究されている。

具体的な手法としては、単連結法、完全連結法、群平均法、ウォード法、重心法、メディアン法、最短距離法、最長距離法などが挙げられる。

○非階層的クラスタリング

データ集合を、設定した評価関数に基づいて分割し、その評価関数に対する最適解(最適な分割)を求めていくことでクラスタリングを行う手法である。階層的クラスタリングに比べて、文書量が増えてもさほど類似度計算量が増えない点から、文書クラスタリングに適しているとされている。

具体的な手法としては、k-means、混合分布モデル、スペクトラルクラスタリング、pLSI、NMF、Fuzzy c-means などが挙げられ、k-means は本研究においても利用されるため、詳細を次ページに示す。

○k-means

非階層的手法における代表的手法であり、初期値として、分割後のクラスタ数Kをあらかじめ与える必要がある。k-meansのアルゴリズムは、以下のようになる。

Step 1：K個のクラスタの代表点$c_1, c_2, \cdots, c_K$を適当に設定する

Step 2：それぞれのデータxを、最も近い代表点の属するクラスタに割り当てる

Step 3：全て割り当て終わったら、クラスタごとに代表点を計算しなおす
ただしこの時、各データxのクラスタが変化していなければ終了
そうでなければ、Step 2へ戻る

結局のところこれは、以下の評価関数を最小化するようなデータのクラスタへの割り当てを求めることになる。

$$\sum_{i=1}^{K} \sum_{x \in C_i} \| x - c_i \|^2$$

この手法は、クラスタリング問題におけるもっともポピュラーな手法の一つとなっており、あるデータに対して何らかの手法でクラスタリングを行った場合、k-meansと比較してその精度を測るケースも多い。ただしk-meansは、その初期値によって得られる解が異なってくるため、初期値の選び方が重要となってくる。そのため、初期値を色々と変化させ解を複数得て、その中から最良の解を選び出す必要があることに注意しなければならない。

2．3　クラスタリング結果の評価

クラスタリング結果の評価方法は、正解集合の有無や、クラスタ数が既知かどうかによって変わってくる。今回の実験では、正解集合が存在しかつクラスタ数は既知であったので、それを前提に話を進めていく。この条件に当てはまるもので代表的な評価関数には、F値、エントロピー、純度、精度などがあり、これらの評価値を算出するにあたってクロス表というものを用いる。ここではクロス表及び、エントロピーと純度について述べる。

○クロス表

既知のクラスタ数をK、クラスタリング結果をC、得られた各クラスタをC_iとし、正解となるクラスタリング結果をAとしたとき、C、Aをそれぞれ、

$$C = \{C_1, C_2, \cdots, C_K\}$$

$$A = \{A_1, A_2, \cdots, A_K\}$$

と仮定する。このとき、クロス表は以下のようになる。

表2.1　クロス表

	A_1	A_2	…	A_j	…	A_K
C_1	x_{11}	x_{12}	…	x_{1j}	…	x_{1K}
C_2	x_{21}	x_{22}	…	x_{2j}	…	x_{2K}
…	…	…	…	…	…	…
C_i	x_{i1}	x_{i2}	…	x_{ij}	…	x_{iK}
…	…	…	…	…	…	…
C_K	x_{K1}	x_{K2}	…	x_{Kj}	…	x_{KK}

○エントロピー

最も標準的に用いられるクラスタリングの評価尺度。各クラスタC_iに対するエントロピーをE_iとすると、以下が評価関数となる。

$$\sum_{i=1}^{K}\frac{|C_i|}{N}E_i=\sum_{i=1}^{K}\frac{\sum_{j=1}^{K}x_{ij}}{N}E_i$$

Nは対象となるデータ数。この関数の値が低いほど、クラスタリング結果が良好であることを意味する。ちなみにE_iは、

$$E_i=-\sum_{h=1}^{K}P(A_h|C_i)\log P(A_h|C_i)$$

そして$P(A_h|C_i)$は、

$$\frac{|A_h\cap C_i|}{|C_i|}=\frac{x_{ih}}{\sum_{j=1}^{K}x_{ij}}$$

によって推定する。

○純度

エントロピー同様、標準的な評価尺度。クラスタC_iに対する純度P_iは、正解のクラスタのデータをどれだけ含むかというのを示す。以下が評価関数。

$$P_i = \frac{1}{|C_i|} \max_h |C_i \cap A_h|$$

よって、クラスタリング結果に対する純度は、

$$\sum_{i=1}^{K} \frac{|C_i|}{N} P_i = \frac{1}{N} \sum_{i=1}^{K} \max_h |C_i \cap A_h|$$

この関数は０～１までの値をとり、値が高いほどクラスタリング結果が良好であることを意味する。

第３章　Co-clustering

通常ほとんどのクラスタリングのアルゴリズムにおいて、文書データをクラスタリングする際、単語セットと文書セットを同時にクラスタリングすることはできないが、これを可能とする手法もいくつかある。こういった２つのデータを同時にクラスタリングする手法のことをCo-clusteringと呼び、文書データに対して行うCo-clusteringは、文書共クラスタリングと呼ぶ。ここでは、2部グラフという概念を用い、そこにスペクトル法などの方法を適用するCo-clusteringについて述べる。

以下に、2部グラフの概念を簡単に説明する。

○評価関数cutの定義

便宜上まず最初に、2部グラフを最適なクラスタに分割する際に用いる評価関数であるcutについて、その定義を述べておく。

グラフG=(Ｖ,Ｅ)が与えられたとする(Ｖ:{1,2,...,|Ｖ|}のデータセット、Ｅ:edge{i,j}とその重みE_{ij}のセット)

隣接行列Mは、

$$M_{ij} = \begin{cases} E_{ij} & \text{edge\{i,j\}があるとき} \\ 0 & \text{それ以外} \end{cases}$$

Ｖを２つのサブセットV_1とV_2に分割した場合のcutの定義

$$cut(V_1, V_2) = \sum_{i \in V_1, j \in V_2} M_{ij}$$

これをｋ個のサブセットに拡張

$$cut(V_1, V_2, \ldots, V_k) = \sum_{i<j} cut(V_i, V_j)$$

ここで、隣接行列とは、以下のグラフのようなもののことをいう

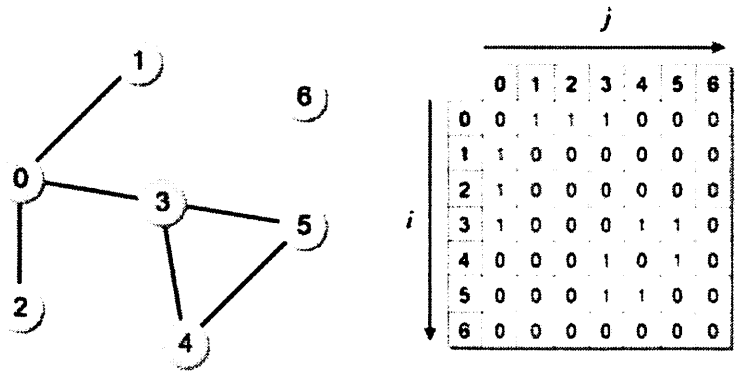

図3.1　隣接行列

○2部グラフモデル

まずは定義だが、以下のようにする。

グラフG = (D,W,E)

D = {$d_1,\ldots,d_n$}、W = {$w_1,\ldots,w_m$}

E = edge{{d_i,w_j} : $d_i \in D$,$w_j \in W$}

Dは文書のデータセット、Wは単語のデータセットである。また、Eはedgeのセットであり、w_jがd_i中にある場合edge{d_i、w_j}は存在するというものである。edge{d_i、w_j}の重みが大きいということは、その文書d_iと単語w_jの関連性が大きいということである。

次に、行が単語データ、列が文書データに対応するm×n行列Aについて、A_{ij}はE_{ij}と等しいと考えると、このグラフの隣接行列は、

$$M = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$$

のようになる。

次に、2部グラフモデルにおけるクラスタリングについて要約すると、

1．単語は、各文書クラスタとの関連性の大小によりクラスタリング可能（逆も同様）

2．各文書クラスタにおける、そのクラスタ内の全文書に対するedgeの重みの和＝単語との関連性の値

3．１及び２より、単語をクラスタリング

4．単語がクラスタリングされたことから、文書クラスタリングも引き起こされ、決定

5．２～５が再帰的に繰り返される

単語クラスタリング及び文書クラスタリングの式は、以下である。

$$W_m = \{w_i : \sum_{j \in D_m} A_{ij} \geq \sum_{j \in D_l} A_{ij}, \forall l = 1, \ldots, k\}$$

$$D_m = \{d_j : \sum_{i \in W_m} A_{ij} \geq \sum_{i \in W_l} A_{ij}, \forall l = 1, \ldots, k\}$$

また、最適なクラスタリングとなるのは、以下の式を満たす時である。

$$cut(W_1 \cup D_1, \ldots, W_k \cup D_k) = \min_{V_1, \ldots, V_k} cut(V_1, \ldots, V_k)$$

・・・※

以上の流れで、2部グラフモデルは文書と単語の最適な同時クラスタリングを可能としている。

最後に、2部グラフモデルについて簡単な図を作成してみる。

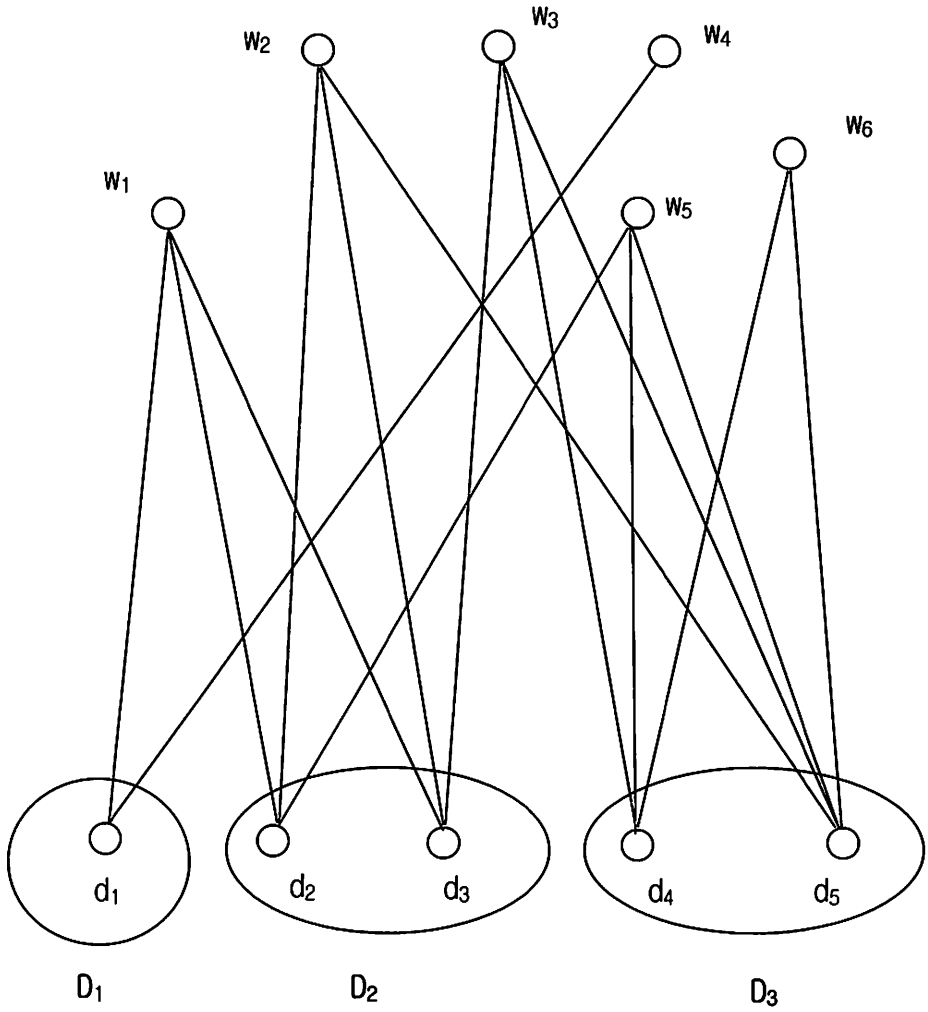

図3.2(a)　2部グラフモデル①

① 文書データを適当なクラスタに分割する。ただし今回は、各々の線(edge)の重みは等しく１と仮定する。

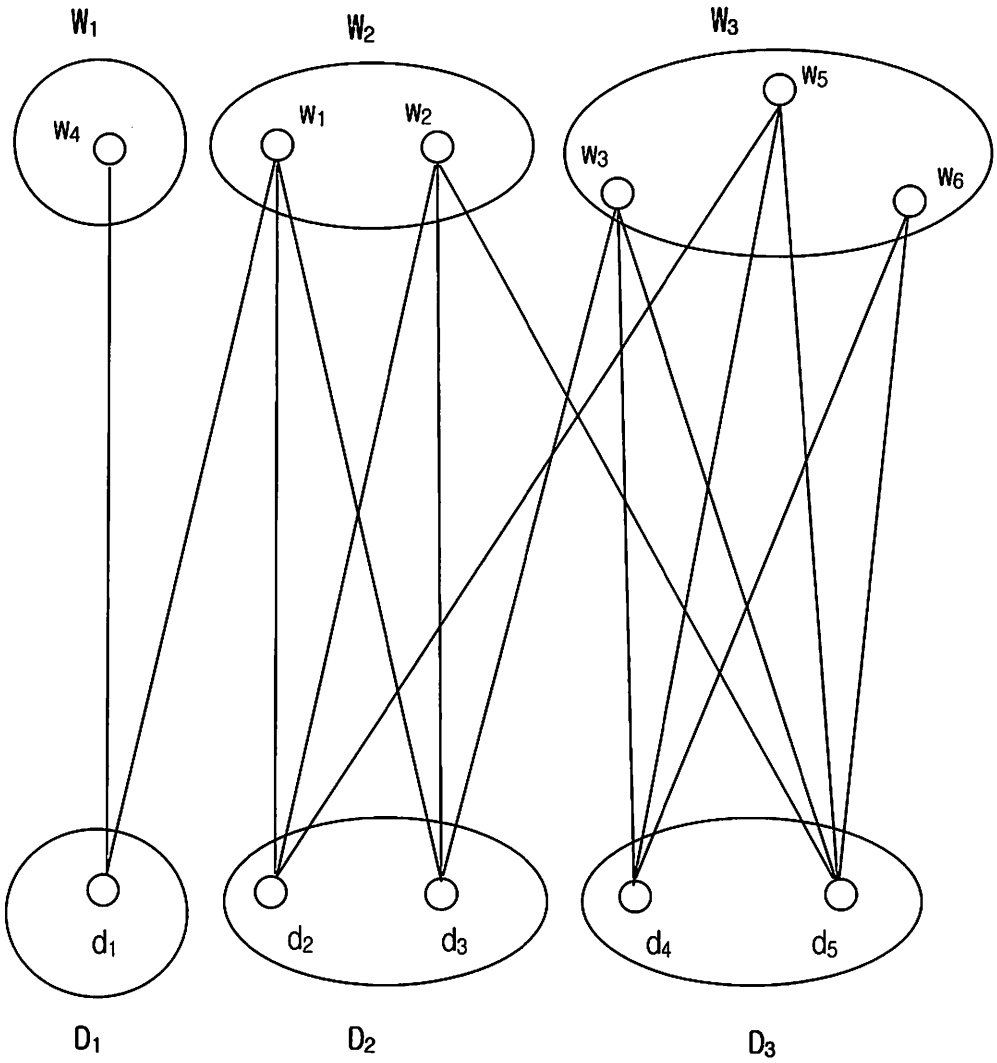

図3.2(b)　2部グラフモデル②

②　各単語データを、最も関連性がある文書クラスタと同じクラスタ番号に割り振る。

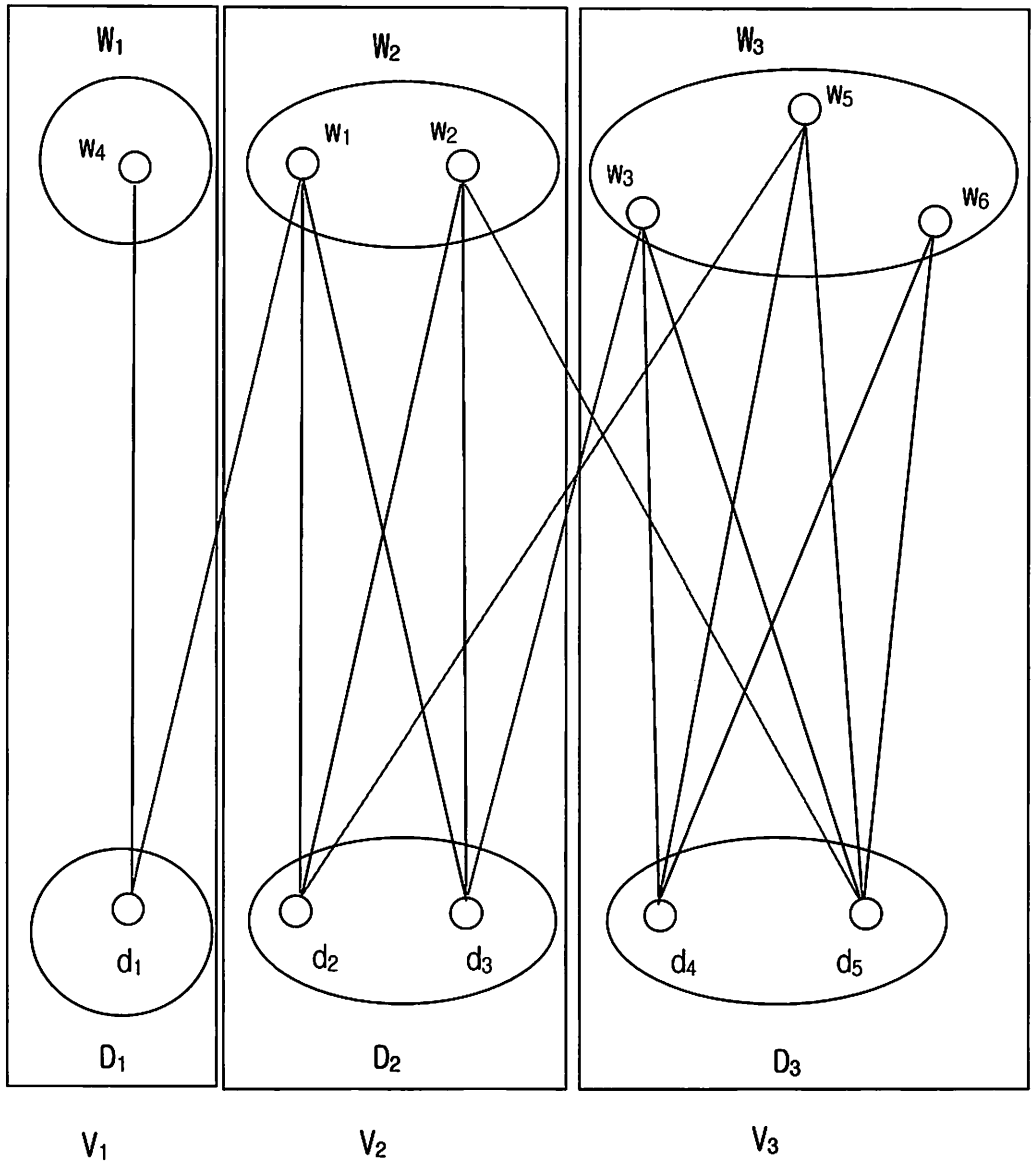

図3.2(c)　2部グラフモデル③

③　単語クラスタが決定したことにより、今度は文書データ視点で②を行う。

④　②と③を繰り返し、※の式を満たすまでクラスタリングを行う。(即ち、上記の赤線の重みの和が最小となる時である)

以上でクラスタリング完了である。

この2部グラフの概念にスペクトル法などの手法を用いることにより、実際の数値としてクラスタリング結果を得ることができる。ただ今回の実験では、ネット上で公開されている既存の Co-clustering のツールを使用したため、ここでは説明を省略する。

第４章　シソーラスを利用したCo-clustering

まずそもそもシソーラスというものは、言葉を単語の上位/下位関係、部分/全体関係、同義関係、類義関係などによって分類した辞書、あるいはデータベースのことである。一般的な辞書では50音順に項目立てがされているのに対し、シソーラスは語彙の持つ意味から、大分類－中分類と下っていき、目的の単語に達することができるようになっている。よって、ものを書いているときや、検索をするときにより適切な言葉を探そうとする際に有用であるといえる。

自然言語処理の分野においてもシソーラスは重要な位置を占めており、『分類語彙表』や『EDR概念体系辞書』、『デジタル類語辞典』のように電子データベース化されているものもある。これらは全文検索システムにも利用されており、あいまい検索もシソーラスを利用して行われている。

データベース化されたシソーラスは木構造、または表形式で成り立っているものが多い。以下に、シソーラスのイメージを簡単に図として示してみる。

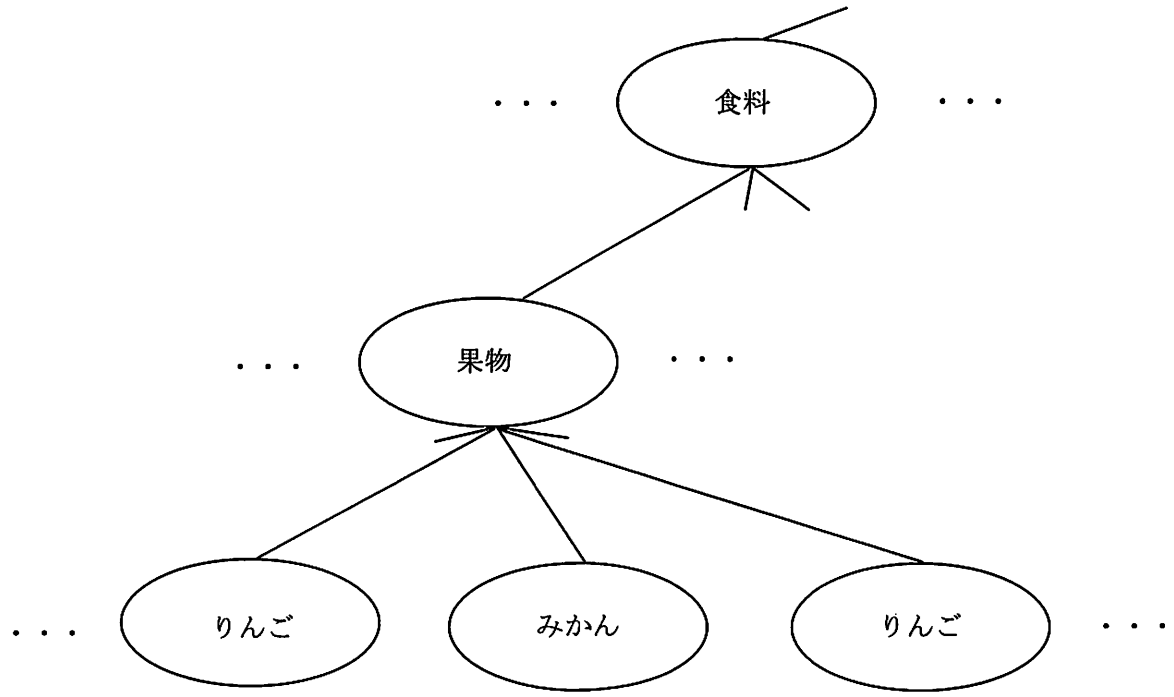

図4.1　シソーラスのイメージ

今回の実験では、数多くあるシソーラスの中から『分類語彙表』を利用した。シソーラス中の単語は、既になんらかの手法によってクラスタリングされたものであるため、このシソーラス中における各ノードを１つのクラスタとして捉え、ノードidをそのままクラスタのidとして扱う。これを参照することにより、実際に扱う各文書データを形態素解析した単語リストにクラスタ番号がつけることが可能となり、あらかじめ単語がクラスタリングされた状態として扱える。この状態のデータを文書クラスタリングすることにより、シソーラスを利用したCo-clusteringが可能となる。

この手法の利点を挙げると、まず第１に、既存の数多くの手法において文書共クラスタリングが可能となること。そして第２に、今後新語などが生まれた場合でも、シソーラスさえ更新してあれば問題なく対応できるという点が挙げられる。

本実験では、シソーラスを用いて変換したデータに対しk-meansを実行することにより比較を行っている。

第５章　実験

５．１　概要

第４章で示した手法の有用性を測るため、本研究ではこの手法に着目して実験を進めていく。したがって、k-means によるクラスタリング、Co-clustering によるクラスタリング、シソーラスを利用した Co-clustering(以下、本手法)の３つの手法を実行し、クラスタリングの正解率を比較することとなる。

今回の実験で使用するデータは、ネットニュースサイト(http://news.goo.ne.jp/)に掲載された、2003年11月25日から12月5日までの10日間のニュース記事とした。内容は、政治・経済・国際・社会・スポーツの5カテゴリで、総数は395文書。各文書には番号が割り振ってあり、各文書番号とそのカテゴリの種類は以下のようになる。

0～62	→	経済
63～115	→	国際
116～172	→	政治
173～354	→	社会
355～394	→	スポーツ

表5.1　文書データの番号とカテゴリの対応

それぞれの文書データは、あらかじめ形態素解析によって単語に分割され、頻度によって重み付けされている。

この文書データに対し各手法を実行していくわけだが、その際必要になるデータの変換には、ruby を利用したプログラムで行った(ソースリストを付録として載せる)。

また、k-means の実行及び、正解率の評価方法であるエントロピーと純度の数値算出には統計解析ソフトＲを用い、Co-clustering の実行に使用するツールは、http://www.cs.utexas.edu/users/dml/Software/cocluster.html で無料配布しているものを使用した。

5．2　k-means

まずはじめに、データに対してk-meansを実行するワケだが、用意したデータを以下の手順で変換した。

Step1:全ての文書データ中の単語をまとめ番号を振り、リスト化する

Step2:各文書データ中の単語データを、そのリスト化した番号と重みに変換し、ベクトル化する

Step3:全文書データをまとめた行列を作成する

Step4:MM形式のデータを作成する

Step4で作成したデータを用いて、k-meansを実行した。k-meansを行う際の繰り返し回数は10回・20回・30回・100回の4通り行い、それぞれについて5回ずつ試行して、そのエントロピー及び純度を算出した。ちなみにこの段階では、単語の種類は総数8995個であった。以下にその結果を表として示す。

試行回数 繰り 返し回数	1回目	2回目	3回目	4回目	5回目	平均
10回	0.83309	0.1955848	0.2678031	0.3280055	0.2519133	0.3752793
20回	0.8237311	0.2583964	0.2672690	0.1955848	0.2813583	0.3652679
30回	0.8237311	0.2118070	0.2260421	0.1978978	0.2050992	0.3329154
100回	0.8083835	0.1521214	0.2378584	0.2813583	0.2745592	0.3508562

表5.2(a)　k-means実行時のエントロピー

試行回数 繰り 返し回数	1回目	2回目	3回目	4回目	5回目	平均
10回	0.478481	0.9037975	0.8556962	0.8329114	0.878481	0.7898734
20回	0.5012658	0.8734177	0.8734177	0.9037975	0.8582278	0.8020253
30回	0.5012658	0.9012658	0.8936709	0.913924	0.9088608	0.8237975
100回	0.5189873	0.9316456	0.8886076	0.8582278	0.8582278	0.8111392

表5.2(b)　k-means 実行時の純度

上記結果をまとめたものを以下にグラフとして示す。

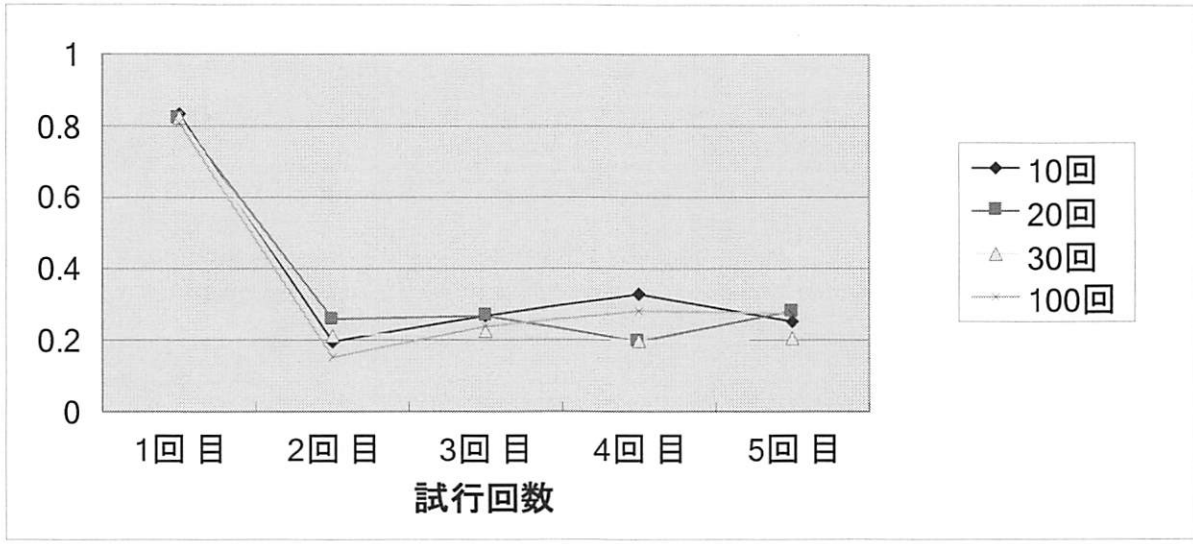

図5.1(a)　k-means 実行時のエントロピーのまとめグラフ

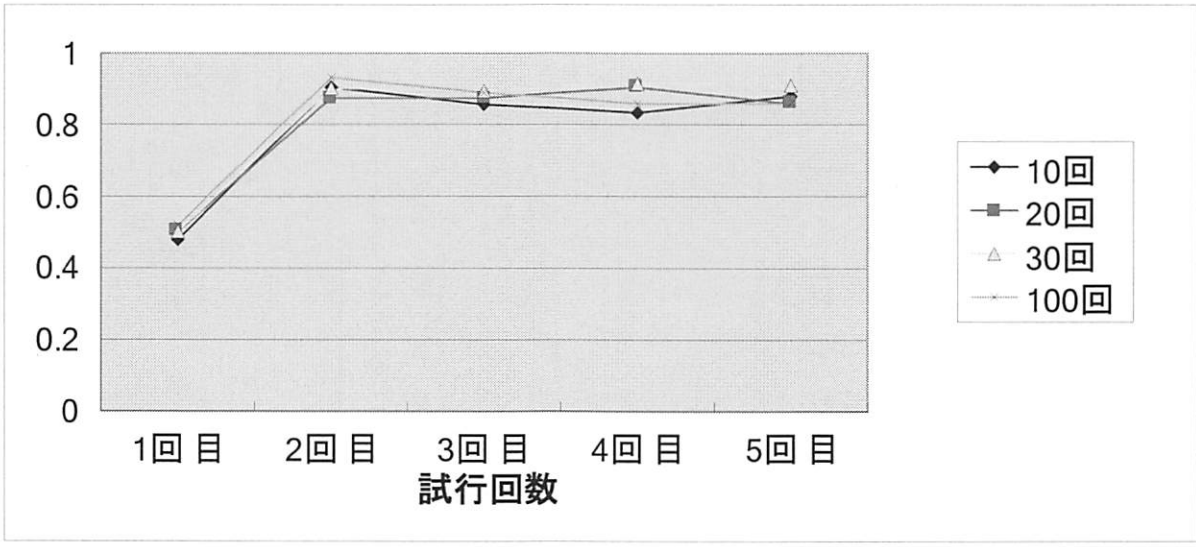

図5.1(b)　k-means 実行時の純度のまとめグラフ

5．3　Co-clustering

ここでは、5.2のStep3で作成した行列に対し、5.1でも述べたツールを利用してCo-clusteringを実行した。実行の際に設定する単語クラスタの数は、本データにおける単語の種類の最低個数である1個、及び最大個数である8995個、また、次節で述べるが、シソーラスを用いた際の単語クラスタ数4539個、あとは適当に10個、100個、1000個、この6通りに対して各5回ずつ試行を行い、出た結果に対してエントロピー及び純度の算出を行った。以下に結果を表として示す。

試行回数 単語 クラスタ数	1回目	2回目	3回目	4回目	5回目	平均
1個	0.8745415	0.8824855	0.8634206	0.8503453	0.8831973	0.8707980
10個	0.6085291	0.5248568	0.5970035	0.6445978	0.6118637	0.5973702
100個	0.736724	0.6068543	0.6110398	0.4435648	0.5103679	0.5817102
1000個	0.5150887	0.4720499	0.3529492	0.4502214	0.3576605	0.4295939
4539個	0.4887943	0.5419632	0.2890837	0.4196447	0.5620827	0.4603137
8995個	0.5407997	0.4441138	0.3108484	0.5252517	0.5175052	0.4677038

表5.3(a)　Co-clustering実行時のエントロピー

試行回数 単語 クラスタ数	1回目	2回目	3回目	4回目	5回目	平均
1個	0.4607595	0.4151899	0.4379747	0.4481013	0.4227848	0.4369620
10個	0.6278481	0.7063291	0.6227848	0.6101266	0.6506329	0.6435443
100個	0.556962	0.6253165	0.6531646	0.7721519	0.6835443	0.6582279
1000個	0.713924	0.7367089	0.835443	0.7670886	0.7848101	0.7675949
4539個	0.7316456	0.6810127	0.8455696	0.764557	0.6278481	0.7301266
8995個	0.7012658	0.7417722	0.8253165	0.6810127	0.6683544	0.7235443

表5.3(b)　Co-clustering 実行時の純度

上記結果の平均をまとめたものを以下にグラフとして示す。

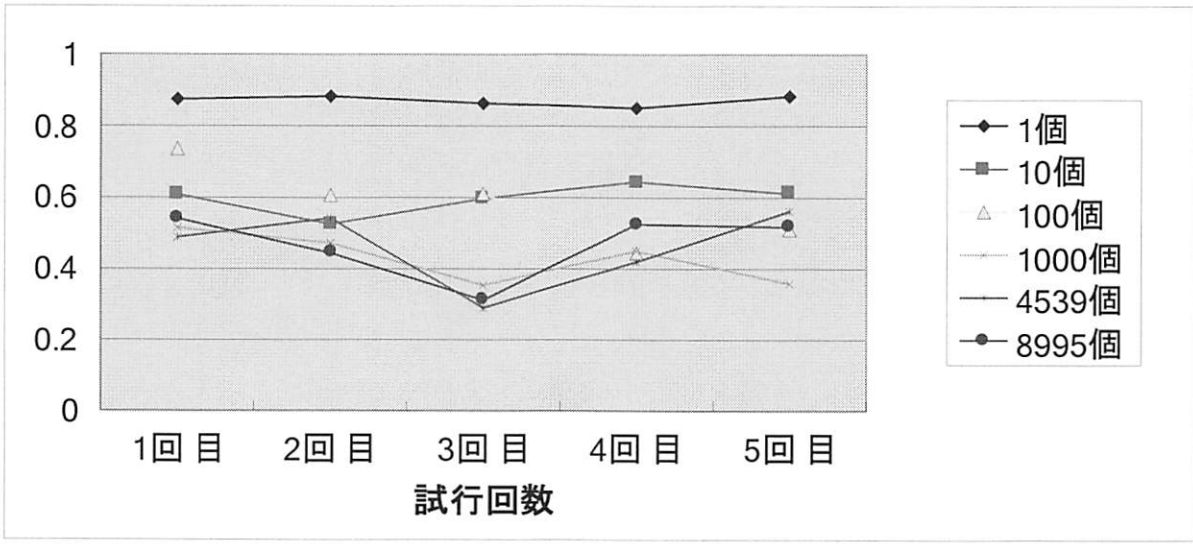

図5.2(a)　Co-clustering 実行時のエントロピーのまとめグラフ

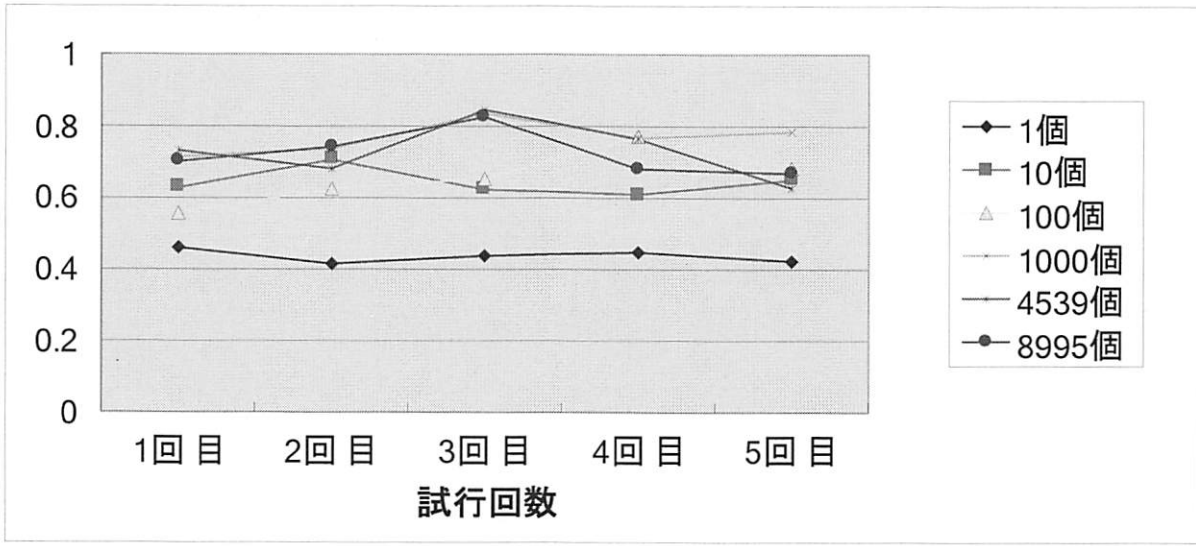

図5.2(b)　Co-clustering 実行時の純度の平均のまとめグラフ

５．４　シソーラスを利用したCo-clustering

ここでは、分類語彙表を利用して新たに単語をリスト化しなおしたデータに対し、5.2のStep1～Step4と同様の作業を行い、k-meansを実行した。k-meansを行う際の繰り返し回数は、これも5.2と同様10回・20回・30回・100回の4通り行い、それぞれについて5回ずつ試行して、そのエントロピー及び純度を算出した。尚、単語を分類語彙表によってリスト化しなおした結果、その種類は総数4539個となった。以下に結果を表として示す。

試行回数 繰り 返し回数	1回目	2回目	3回目	4回目	5回目	平均
10回	0.5448536	0.5739471	0.4221524	0.4864769	0.4773207	0.5009501
20回	0.5204929	0.5816594	0.5816594	0.4808252	0.5552639	0.5439802
30回	0.501426	0.4937711	0.5266418	0.5190874	0.5045949	0.5091042
100回	0.5215398	0.5190874	0.6147558	0.5002991	0.5021384	0.5315641

表5.4(a)　本手法実行時のエントロピー

試行回数 繰り 返し回数	1回目	2回目	3回目	4回目	5回目	平均
10回	0.6607595	0.6481013	0.721519	0.6962025	0.7113924	0.6875949
20回	0.6962025	0.6278481	0.6278481	0.6987342	0.6405063	0.6582278
30回	0.6810127	0.678481	0.678481	0.686076	0.6810127	0.6810127
100回	0.678481	0.686076	0.6126582	0.6911392	0.6962025	0.6729114

表5.4(b)　本手法実行時の純度

上記結果の平均をまとめたものを以下にグラフとして示す。

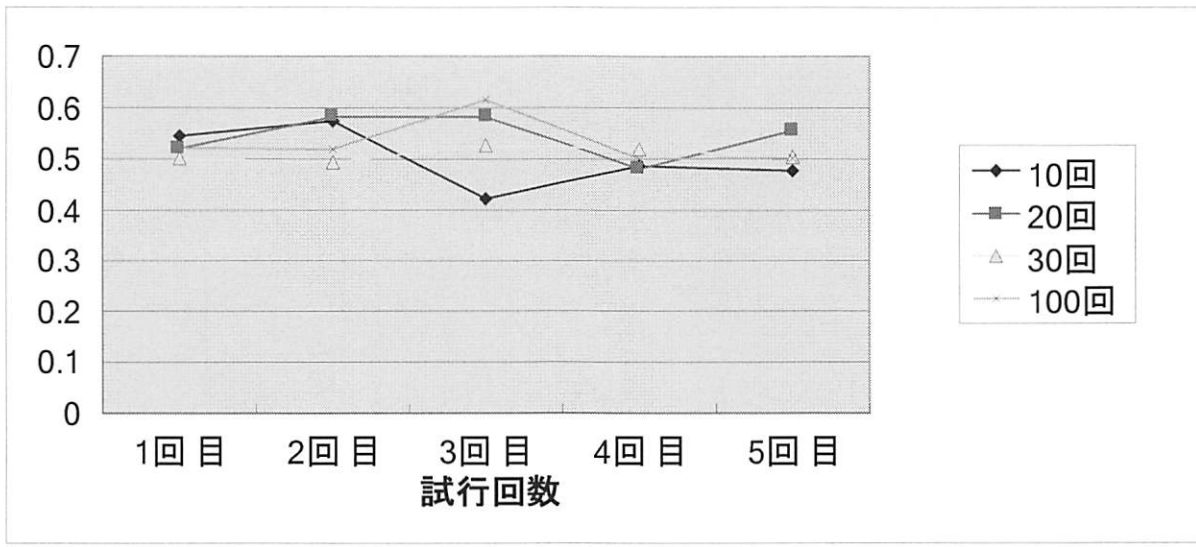

図5.3(a)　本手法実行時のエントロピーのまとめグラフ

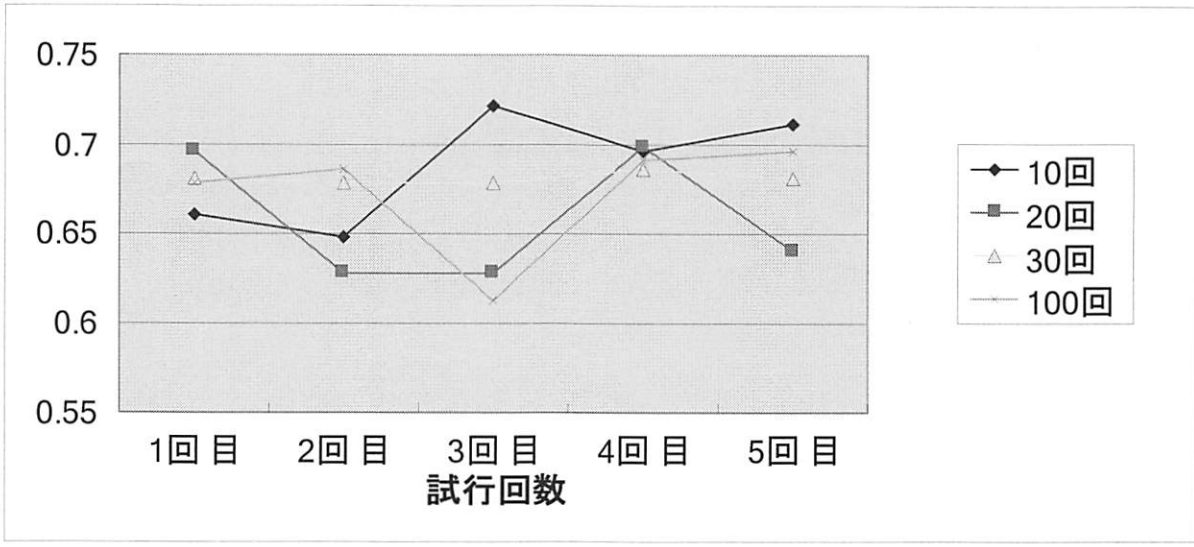

図5.3(b)　本手法実行時の純度のまとめグラフ

5．5　結果

まず最初に、実験を行った３つの手法の平均をまとめたグラフを以下に示す。

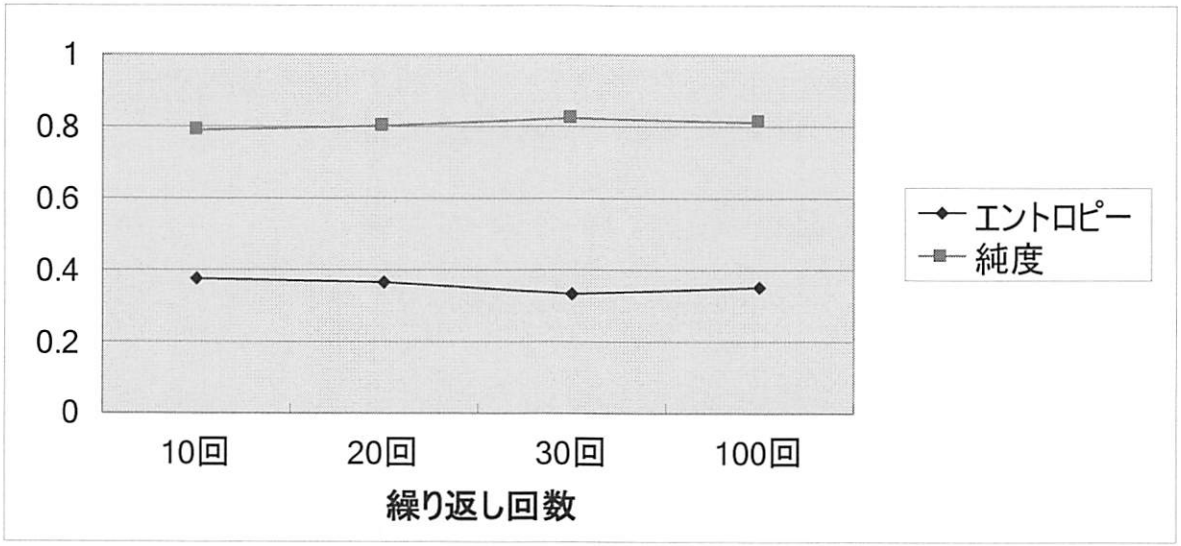

図5.4(a)　k-means 実行時のエントロピー・純度の平均のまとめグラフ

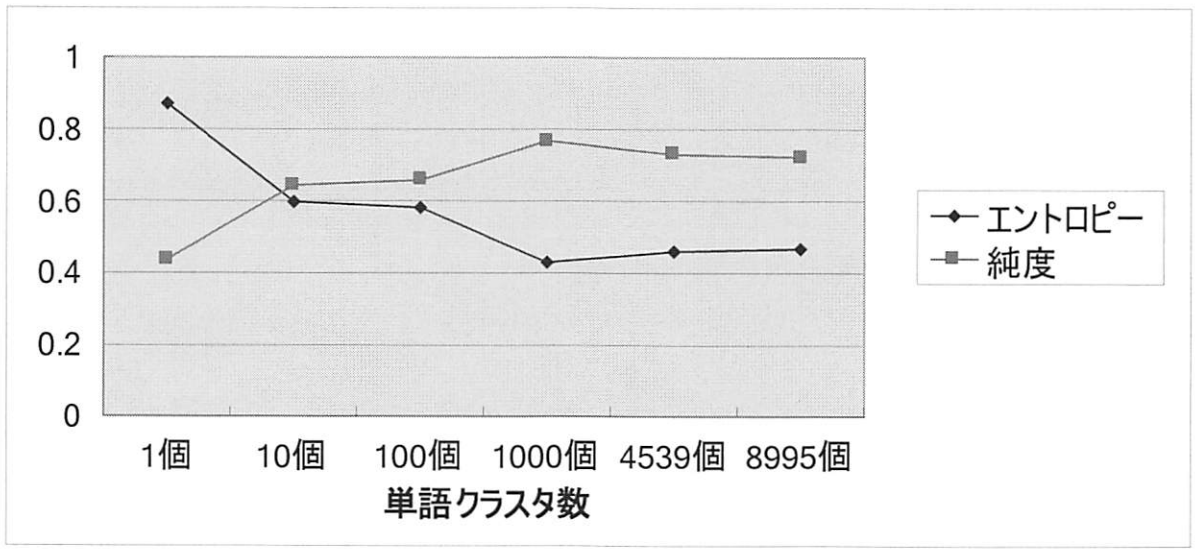

図5.4(b)　Co-clustering 実行時のエントロピー・純度の平均のまとめグラフ

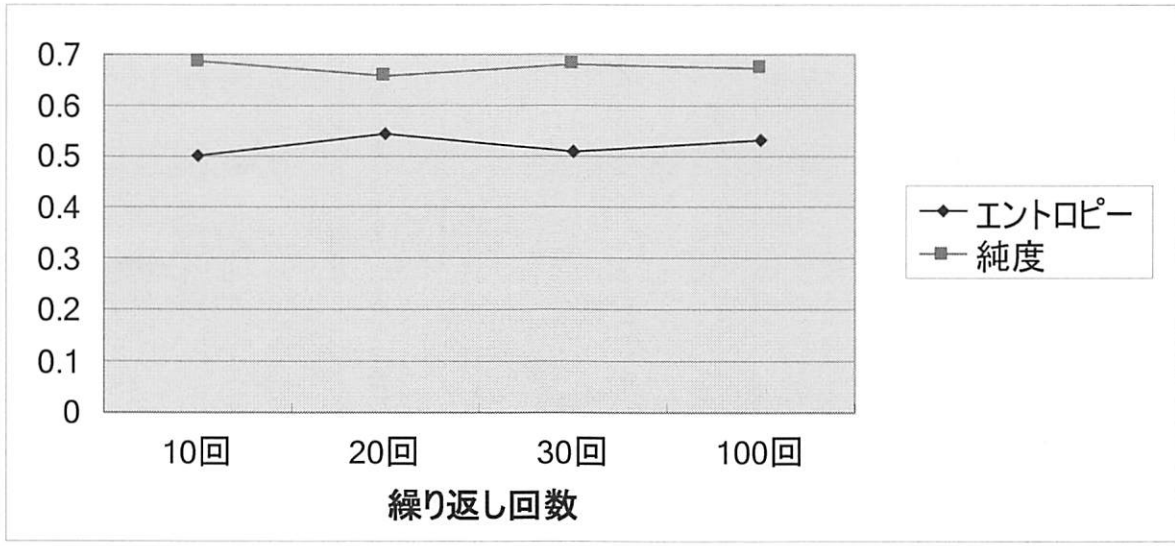

図5.4(c)　本手法実行時のエントロピー・純度の平均のまとめグラフ

以上のグラフを単純な結果として見ると、平均ではk-meansが一番良い値が出た。ただグラフを見て分かるとおり、k-meansは初期値をランダムに設定するため、毎回結果が変わってくる点からたまたま全体的に良い結果が出た可能性もある。Co-clusteringについて見ると、設定した単語クラスタ数によって結果にかなり差が出ている。本手法では、繰り返し回数にあまり依存しない、比較的安定した結果が得られた。

ここで着目すべき点は、Co-clusteringにおける単語クラスタ数４５３９個の結果と本手法の結果の比較である。Co-clusteringにおいて、単語クラスタ数４５３９個の時のエントロピー及び純度の結果は、試行の度に多少バラつきがあるものの、全体の中ではかなり良い結果となっている。また本手法では、あらかじめ単語を４５３９個のクラスタに分割していた結果、出た値が繰り返し回数にあまり依存しなかったことから、文書クラスタリングがかなり早い段階で完了しているのではないかと推測できる。これらの点から、文書だけでなく単語もクラスタリングしておくことの有用性が伺える。

ただ、単語クラスタ数４５３９個における結果だけを見比べると、通常のCo-clusteringの結果のほうが良好だった。

第6章　考察

実験の結果を通じて、単純な平均値のみに着目すると、本手法は他の２つの手法と比べてさほど改善されているようには見てとれない。しかし先にも述べたように、k-means と比較する場合は、ランダムな初期値による k-means の不確実性を考慮に入れなければならない。本手法においても、文書クラスタリングに k-means を利用してはいるが、どの試行においても安定した結果が得られている。つまりあらかじめ単語をクラスタリングしておくことにより、通常の k-means に比べて初期値に依存しづらくなっていると考えられる。

Co-clustering との比較においては、本手法における単語クラスタ数で設定した際に、Co-clustering で良好な結果が出たことなどから、本手法における単語クラスタリングがより「正しい」クラスタリングであったことが分かる。つまり、単語クラスタ数の設定ミスによる精度減少の可能性を低くできたと捉えられる。

ただしここで問題となってくるのは、単語クラスタリングがより「正しい」結果を導き出せるかどうかが、元となっているシソーラスに依存してくるということである。今回の実験では「分類語彙表」を利用したが、４章でも述べたようにシソーラスには様々な種類があるため、他のシソーラスを利用した場合も同様以上の出るかどうかは、調査の必要がある。更に今回は、政治・経済・国際・社会・スポーツの5カテゴリにおける文書データに限定して実験を行ったが、他のカテゴリに分類される文書データでもどうなのか、という点も同様に調べる必要がある。

以上の点を踏まえると、本手法の結果は安定しているという点で良好なものであったと考えられるが、調査すべき点も多く発見された。また、今回の実験では絶対的な試行回数が不足しているため、裏付けを取るために改めて調査しなおす必要もある。

第7章　おわりに

今回はk-meansの確認にのみ留まったが、本研究における目的である、既存のクラスタリング手法を文書共クラスタリングに活用するという点は達成できた。ただ、新たに浮き彫りになった課題も多く、今回確認しきれなかった点も含め、今後の更なる研究が必要となる。

その中でも、今回利用した「分類語彙表」以外のシソーラスに対しても同様の手法が実用可能であるかどうかについての調査は重要となる。これが可能となれば、扱うデータに最も適したシソーラスを選出することにより、更なるクラスタリング精度の向上が望めるからである。また同様に、k-means以外の既存のクラスタリング手法についても有用と確認できれば、扱えるデータの幅も更に広まると考えられる。

謝辞

本論文の作成にあたり、多大なご助言及び日頃よりのご指導を賜わりました新納浩幸先生、佐々木稔先生に心より感謝の意を表します。また、本研究を進めるにあたりご協力をいただいた、同研究室の茂木哲矢氏、及び田中洸一氏にも深く感謝致します。

参考文献

[1]新納浩幸："Rで学ぶクラスタ解析",オーム社,pp.34-41,81-83(2007)

[2]"文書クラスタリングの基礎"

<http://mikilab.doshisha.ac.jp/dia/research/report/2007/0913/004/report20070913004.html> (2009/2/5アクセス)

[3]"形態素解析-Wikipedia-"

<http://ja.wikipedia.org/wiki/%E5%BD%A2%E6%85%8B%E7%B4%A0%E8%A7%A3%E6%9E%90#.E3.83.95.E3.83.AA.E3.83.BC.E3.81.A7.E5.85.A5.E6.89.8B.E5.8F.AF.E8.83.BD.E3.81.AA.E3.82.82.E3.81.AE> (2009/2/5アクセス)

[4]"シソーラスとは(thesaurus)： - IT用語辞典バイナリ"

<http://www.sophia-it.com/content/%E3%82%B7%E3%82%BD%E3%83%BC%E3%83%A9%E3%82%B9> (2009/2/5アクセス)

付録

プログラムソースリスト(Ruby)

・Ｓｔｅｐ１－１

```
#!/usr/bin/ruby -Ke
s = 0
for s in 0..394
i = 1
h = Hash.new
b = Array.new
open("netnews/#{s}/nouns") {|file|
file.each{|line|
a = line.split(" ")
if h.key?(a[0]) == false then
h[a[0]]=i
b.push(a[0])
i += 1
end
}
}
open("LOG.txt",'a'){|file|
b.each{|t|
file.print t," ",h[t],"Wn"
}
}
end
```

・Ｓｔｅｐ１－２

```
#!/usr/bin/ruby -Ke

i = 1

h = Hash.new

b = Array.new

open("LOG.txt") {|file|

file.each{|line|

a = line.split(" ")

if h.key?(a[0]) == false then

h[a[0]]=i

b.push(a[0])

i += 1

end

}

}

open("List.txt",'a'){|file|

b.each{|t|

file.print t," ",h[t],"¥n"

}

}
```

• S t e p 2

```
#!/usr/bin/ruby -Ke

s = 0

for s in 0..394

i = 1

h = Hash.new

h2 = Hash.new

b = Array.new

open("List.txt") {|file|

file.each{|line|

a = line.split(" ")

h[a[0]]=a[1]

}

}


open("netnews/#{s}/nouns"){|file|

file.each{|line2|

z = line2.split(" ")

if h.key?(z[0]) == true then

h2[h[z[0]]]=z[1]

b.push(h[z[0]])

end

}
```

```
}

open("Vector/#{s}.txt",'a'){|file|
b.each{|t|
file.print t," ",h2[t],"¥n"
}
}
end
```

・S t e p 3

```
#!/usr/bin/ruby -Ke

s = 0

t = 0

for s in 0..394

h = Hash.new

open("Vector/#{s}.txt") {|file|

file.each{|line|

a = line.split(" ")

h["(#{s},#{a[0]})"]=a[1]

}

}


open("Matrix.txt",'a'){|file|

for t in 1..8995

if h.key?("(#{s},#{t})") == false then

file.print 0

else

file.print h["(#{s},#{t})"]

end

if t == 8995

break

end
```

```
file.print " "
end
file.print "Wn"
}
End
```

・S t e p 4

```
#!/usr/bin/ruby -Ke

s = 0

t = 0

for s in 0..394

h = Hash.new

open("Vector/#{s}.txt") {|file|

file.each{|line|

a = line.split(" ")

h[a[0]]=a[1]

}

}


open("MM.txt",'a'){|file|

for t in 1..8995

if h.key?("#{t}") == true then

file.print s+1," ",t," ",h["#{t}"],"Wn"

end

end

}

end
```