社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－NL－180 (13)
2007／7／24

半教師有りクラスタリングを用いた語義数の推定と語義別用例の収集

新納浩幸　　佐々木稔
茨城大学工学部情報工学科
{shinnou,msasaki}@mx.ibaraki.ac.jp

本論文では語義別用例の収集を目的とした半教師有りクラスタリング手法を提案する。このタスクではクラスタ数（語義数）の推定が重要であり、その点に特化した手法となっている。まず与えられた単語からその語義数（クラスタ数）を大目に見積もる。その見積もったクラスタ数を使って、用例集に対して通常のクラスタリングを行い、細かいクラスタに分割する。次にユーザからの制約を利用して、それらクラスタの統合を行う。本手法は少ない制約数で高い正解率を上げることができる。実験ではSENSEVAL2の日本語辞書タスクで利用された名詞12単語に対するデータを用いて、本手法の有効性を示した。用例間の類似度の測定方法の改良が今後の課題である。

キーワード：半教師有りクラスタリング、語義別用例、語義数、用例間の類似度、SENSEVAL2日本語辞書タスク

Collection of example sentences according to the meaning of a word and estimation of the number of meanings by using semi-supervised clustering

Hiroyuki Shinnou　　Minoru Sasaki
Department of Computer and Information Sciences, Ibaraki University
{shinnou,msasaki}@mx.ibaraki.ac.jp

In this paper, we propose a semi-supervised clustering method to collect sentences according to the meaning of a word. In this task, the estimation of the cluster number is most important. Our method concentrates on this problem. First we overestimate the number of meaning (the cluster number) for the target word. By using the number, we conduct the general clustering for data set to get many small clusters. Next using constrains given by the user, we integrate clusters. Our method performs the high precision with small constrains. In the experiment, we try our method for 12 Japanese noun words used in the SENSEVAL2 Japanese dictionary task. The experiment shows the effectiveness of our method. In future, we will improve the measurement of the similarity of sentences.

Keywords: semi-supervised clustering, example sentences, number of meanings, similarity between sentences, SENSEVAL2 Japanese dictionary task

1　はじめに

ある単語の用例を集め、それら用例をその単語の語義に基づいて分類するタスクに取り組んでいる。本論文では、このタスクを効果的に行うための半教師有りクラスタリング手法を提案する。

語義別の用例は本格的な意味解析にとって有用である。例えば、語義別の用例を訓練データとして利用することで語義の曖昧性を解消する分類器を学習することが可能となる[8]。また動詞の格フレームの自動構築[9]は、動詞の語義別の用例があれば容易に行える。またシソーラスの自動構築[5]においては、通常、名詞の多義性は無視されるが、語義別の用例があれば、語義を考慮したシソーラスの作成も容易である。辞書の編纂、語学学習においても、語義別の用例は有用であろう。

ある単語wの語義別の用例を収集するには、単語wを含む用例をコーパスから抽出し、その用例中の単

語 w の語義を識別できればよい。つまり語義別用例の収集は、語義の曖昧性解消のタスクとして扱える。そのために（半）教師有り学習を用いるアプローチによって解決できそうであるが、以下に示す２つの問題から、そのアプローチによる解決は困難である。第１の問題は、訓練データの作成コストである。教師有り学習では大量の訓練データを必要とし、対象となる単語が多い場合、その作成コストが大きすぎる。第２の問題は、語義の設定である。（半）教師有り学習で本タスクに挑む場合、単語 w の語義を予め設定しておかなければならない。単語 w が与えられたときに、w の語義を内省により列挙することは困難である。例えば、語義の粒度を均一に保つことは難しいし、マイナーな語義を見落としてしまう危険性もある。

（半）教師有り学習のアプローチは問題があるために、教師なし学習、つまりクラスタリングを行うことで、語義別の用例を収集することも考えられる。しかし k-means に代表される多くのクラスタリング手法は、クラスタ数、つまり語義数を陽に与える必要があり、それらの手法は使えない。適切なクラスタ数を推定してクラスタリングを行う手法もあるが、このタスクにおけるクラスタ数の推定は、語義の粒度を固定することに対応してしまう。語義の粒度は単語によって異なるので、語義の数を自動的に求めることには無理がある。

以上の理由から、語義別の用例を収集するタスクに対して、ここでは半教師有りクラスタリングを用いる。半教師有りクラスタリングでは、クラスタリングの対象のデータのいくつかに、クラスタに関する情報を与えて、クラスタリングを行う [4]。一般には、システムがデータのペアを複数個選び、それらをユーザに提示する。ユーザはそれらペアのデータに対して、ペアのデータが「同じクラスタに属する」(must-link) か「異なるクラスタに属する」(cannot-link) という制約を返す。ユーザはクラスタ数やその種類を設計する必要がないために、（半）教師有り学習よりもユーザの負荷が少ない。

本論文の半教師有りクラスタリングの特徴は２つある。

１つはクラスタ数を推定するために、ユーザから得られる制約情報を利用することである。従来の半教師有りクラスタリングの研究では、クラスタ数を予め与えるタイプが多い。クラスタ数をクラスタリング中で推定する枠組みになっていても、それらは単に通常のクラスタリング手法の拡張であり、ユーザからの制約情報を積極的に利用する形にはなっていない。このために前述した語義の粒度の問題に対応できない。本手法ではユーザから得られる制約を、主に、クラスタ数の推定に利用する。ただしクラスタ数が正しく推定できても、正しい分類が行えていなければ、実際は意味がない。そのため、ここではクラスタリングの正解率を高めるような、半教師有りクラスタリングの枠組みを提案する。

もう１つの特徴は本タスクの場合、クラスタ数が多いか少ないかがある程度見積もれることを利用している点である。従来の半教師有りクラスタリング手法でも、ユーザからの制約情報を利用してクラスタ数を推定するものはあるが、対象としているタスクが一般的であり、クラスタ数が多い場合も少ない場合も均一の処理になっている。本タスクでは、単語が与えられたときに、その単語に非常に多くの語義があるか、あるいはそれほど多くはないかは容易に推測できる。この概略のクラスタ数を利用している点が本手法の特徴である。

本論文で提案する半教師有りクラスタリングの手法の概略を述べる。まず対象の単語から判断して、ある程度大目の語義数（クラスタ数）を見積もる。この大目に設定されたクラスタ数を利用して、用例集に対して通常のクラスタリング手法によりクラスタリングを行う。次に各クラスタの代表点を求め、代表点どうしをペアにしてユーザに提示し、そのペアに制約を与える。must-link が与えられたら、代表点に対応するクラスタを統合し、cannot-link が与えられたら、それらクラスタは別クラスタと決定してゆくことで、クラスタリングが行える。ただしクラスタの統合が起こった場合でも代表点を変化させないことで、クラスタリングの正解率を高める。またこれによって、ユーザに提示するペアの順序は任意にできる。

実験では SENSEVAL2 の日本語辞書タスク [13] で用いられた名詞 12 単語を対象とした。SENSEVAL2 で提供されたそれら単語に対する訓練データとテストデータを合わせて用例のセットとし、本手法を適用した。ユーザに制約を付与してもらうペア数は、平均して 24.6 個であった。クラスタ数に対しても妥当な推定を行えた。またクラスタリングの正解率（0.757）は、通常のクラスタリングの正解率（0.592）と比較すると大きく改善された。

2 半教師有りクラスタリングによる用例の分類

本論文で提案する半教師有りクラスタリングのアルゴリズムを図 1 に示す。

Input w and $D=\{d_1,d_2,\cdots,d_N\}$
estimate k for w
cluster D to k clusters $C=\{C_1,C_2,\cdots,C_k\}$
$A=\{C_1\}$, $C=\{C_2,C_3,\cdots,C_k\}$
for(i =2; i < k+1; i++) {
  z is the center of C_i
  foreach C in A {
    x is the center of C
    give (x,z) to user
    get Const from user
    if (Const = msut-link) {
      $C \leftarrow C \cup C_i$; break
    }
  }
  if (Const = cannot-link) {
    $A \leftarrow A \cup \{C_i\}$
  }
}
Output A

図 1: 本手法のアルゴリズム

入力は対象単語 w とその用例の集合であるデータセット D である。まず w をみて、その語義数（クラスタ数）k を大目に見積もる[1]。D を既存クラスタリング手法を用いて k 個のクラスタに分割する。

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

図 1 の集合 A が最終的なクラスタリング結果を保持するクラスタの集合である。最初に $A = \{C_1\}$ とする。処理の概要としては C 中の C_i を順に、A の中のいずれかのクラスタと同じか、それとも A の中のどのクラスタとも異なるかをユーザから得られる制約に

[1] ここでは実際の語義数の 5 倍から 10 倍を想定している。

よって判断する。ただしユーザに与えるのはクラスタの代表点のペアである。得られた制約が must-link であれば統合し、A 内のどのクラスタとも cannot-link であれば、新たなクラスタとして A に加える。ただし統合した場合でも、クラスタの代表点は統合前ものを使う。これによって処理するクラスタの順序によらずに、一意の解が得られる。ユーザへの問いかけは最悪 $k(k-1)/2$ 回で済む。

以下、本章ではデータセット D のクラスタリングで必要となる用例間の類似度の設定、クラスタの代表点の求め方、代表点の継承について述べる。

2.1 用例間の類似度

用例内の対象単語の文脈から、その用例の素性リストを作成する。ここでは論文 [10] で示された以下の素性を利用する。

e1	直前の単語
e2	直後の単語
e3	前方の内容語２つまで
e4	後方の内容語２つまで
e5	e3 の分類語彙表の番号
e6	e5 の分類語彙表の番号

例を示す。対象の単語を「記録」として、以下の用例を考える（形態素解析され各単語は原型に戻されているとする）。

過去/最高/を/記録/する/た/。

この場合、「記録」の直前、直後の単語は「を」と「する」なので、‘e1=を’, ‘e2=する’ となる。次に、「記録」の前方の内容語は「過去」、「最高」なので、ここから「記録」に近い順に 2 つとり、‘e3=過去’、‘e3=最高’ が作られる。またここでは句読点も内容語に設定しているので、「記録」の後方の内容語は「する」と「。」となり、‘e4=する’、‘e4=。’ が作られる。次に「最高」の分類語彙表 [?] の番号を調べると、3.1920_4 である。ここでは分類語彙表の 4 桁目と 5 桁目までの数値をとることにした。つまり ‘e3=最高’ に対しては、‘e5=3192’ と ‘e5=31920’ が作られる。同様に「過去」の分類語彙表の番号 1.1642_1 から ‘e5=1164’ と ‘e5=11642’ が作られる。次は「する」の分類語彙表を調べるはずだが、ここでは平仮名だけで構成される単語の場合、分類語彙表の番号を調べないことにした。これは平仮名だけで構成される単語は多義性が高く、無意味な素性が増えるので，その問題を避けたためである。もしも分類語彙表上で多義になっていた場合には、それぞれの番号に対して並列にすべての素性を作成する。

結果として、上記の用例に対しては以下の 10 個の素性を要素とする素性リストが得られる。

```
( e1=を, e2=する, e3=最高,  e3=過去,
  e4=する, e4=。, e5=3192,  e5=31920,
  e5=1164,  e5=11642 )
```

語義の曖昧性解消の場合は、上記のような素性リストを定義するだけでよいが、本タスクにおいては、素性リスト間の類似度（あるいは距離）の定義が必要となる。ここでは素性リスト A と素性リスト B の類似度 $sim(A,B)$ を以下のように定義した。

まず素性リスト A と素性リスト B は以下の形になっている。

```
A = ( e1=a1, e2=a2,
      e3=a31, e3=a32, e4=a41, e4=a42,
      e5=a51, e5=a52, e5=a53, e5=a54,
      e6=a61, e6=a62, e6=a63, e6=a64  )

B = ( e1=b1, e2=b2,
      e3=b31, e3=b32, e4=b41, e4=b42,
      e5=b51, e5=b52, e5=b53, e5=b54,
      e6=b61, e6=b62, e6=b63, e6=b64  )
```

$sim(A,B)$ は 3 つの観点からの類似度の和として定義した。

```
sim(A,B) = 1/3 ( (直前単語からの類似度)
  + (直後単語からの類似度)
  + (直前直後の単語以外の文脈からの類似度) )
```

直前単語からの類似度 基本的に $a1 = b1$ であれば 1、それ以外は 0 とする。ただし $a1 = b1$ であっても、以下のように調整する。$a1$ が読点あるいは、長さ 1 のひらがなであれば 0.1、$a1$ が長さ 1 のひらがな以外あるいは、長さ 2 以上のひらがなであれば 0.5 とする。

直後単語からの類似度 基本的に $a2 = b2$ であれば 1、それ以外は 0 とする。ただし $a2 = b2$ であっても調整を行う。この調整は「直前単語からの類似度」と同様である。

直前直後の単語以外の文脈からの類似度 リスト A 中の $e3 = \sim$、$e4 = \sim$、$e5 = \sim$ とリスト B 中の $e3 = \sim$、$e4 = \sim$、$e5 = \sim$ で共通のものの個数を数える。その個数を c とする。リスト A 中の $e3 = \sim$、$e4 = \sim$、$e5 = \sim$ の全個数を a、またリスト B 中の $e3 = \sim$、$e4 = \sim$、$e5 = \sim$ の全個数を b とする。そして直前直後の単語以外の文脈からの類似度を $c/(a+b)$ とする。

以上より $sim(A,B)$ の値が 0〜1 の値として与えられる。

2.2 クラスタの代表点

クラスタ $C = \{x_1, x_2, \cdots, x_n\}$ の代表点 x_c を以下で定義する。

$$c = \arg\max_{i \in 1:n} \sum_{j \in 1:n} sim(x_i, x_j)$$

これはクラスタ C の中で、C 中の用例との類似度の和が、最も大きいものを意味する。

2.3 代表点の継承

本手法では細かくクラスタに分割した後に、各クラスタを順に、最終的なクラスタの集合 A に加えるか、A 内に既にあるクラスタに統合させるかの処理を行う。統合させる場合、クラスタの代表点を変化させず、継承させることが本手法の特徴である。

本手法における代表点は、クラス名と同じ機能を有している。データセット D を k 個のクラスタに分割したとき、各クラスタ C_i の代表点はラベルに相当する。このとき本タスクはこのラベルのクラスタリングとなる。さらにラベルどうしが同じか異なるかは、ユーザから与えられるので、ラベルのクラスタリングは単なる組み合わせ処理となる。つまりこの場合のラベルはクラス名と見なすことが可能である。

本タスクがクラス名（代表点）のクラスタリングであれば、本手法ではクラスタの統合が起こった場合でも、クラスタの代表点（クラス名）は変化させるべき

ではない。変化させてしまうと、正解率が下がる可能性が高い。この点を以下に示す。

今、データセット D を k 個のクラスタに分割したとき、各クラスタ C_i には、様々なクラスのデータが混在している。正解という観点でみる場合、C_i 中で最も頻度の高いクラスが C_i に対する正解のクラスとなり、その頻度に対応するものが正解数になる。極端な話、k がデータセットのデータ数であれば、各クラスタ C_i の正解率は 100% である。また $k=1$ とした場合でも、正解率は 0% になることはない。場合によっては、かなり高いこともある。一般に k が大きい数の場合は、各クラスタ内には同じクラスのデータが多く含まれているはずであり、各クラスタに対する正解率は高くなる。この正解率を p としておく。つまり最初に k 個のクラスタリングした際の各クラスタ C_i には $|C_i|p$ 個の正解が含まれている。本手法では、C_i の代表点を選ぶ際に $|C_i|p$ 個の中から選べれば、代表点どうしが同じクラスタになるかどうかは、ユーザから得られるので、各クラスタ C_i の正解数を保持して、最終的なクラスタリング結果が得られる。

クラスタ C_a とクラスタ C_b を統合させ、クラスタ H を作成した際に、H の代表点 h を新たに選ぶことを考えてみる。代表点 h と同じクラスのデータが H 内にどの程度あるかが問題である。本手法のように h を C_a あるいは C_b の代表点にとれば、H 内には $(|C_a|+|C_b|)p = |H|p$ 個の正解がある。h としてその他の代表点を取った場合に、$|H|p$ 以上の正解数が得られる可能性は低い。

3 実験

ここでは SENSEVAL2 の日本語辞書タスク [13] で課題とされた名詞 12 単語を用いて本手法の有効性を確認する。具体的には SENSEVAL2 で提供されたそれら単語に対する訓練データとテストデータを合わせて用例のセットとし、本手法を適用した。なお SENSEVAL2 の日本語辞書タスクでは名詞 50 単語が課題として与えられているが、訓練データとテストデータを合わせて 300 以上の用例を持つ単語だけを対象とした。その結果、対象単語は 12 単語となった。表 1 にその一覧を示す。

表 1: データセット

単語	用例数	語義数
もの	754	10
問題	636	4
代表	466	3
前	426	4
関係	414	3
午後	396	3
自分	362	2
時代	360	4
子供	354	2
現在	341	2
社会	339	6
今	329	4
平均	431.42	3.91

まず対象単語から判断して、クラスタ数を大目に見積もるが、ここでは一律に 20 に固定した。この点に関しては考察で述べる。

次に用例を素性リストで表現し、用例間の類似度を前述した方法によって計算することで、用例間の類似度行列を作成した。この類似度行列を入力として、クラスタリングツールの CLUTO[2] を利用してクラスタリングを行った。クラスタ数を 20 に設定し、オプションは何も付けずに default の設定でクラスタリングを実行した[3]。20 個のクラスタに関して、前述した手法でユーザから制約を得ることで、クラスタリング結果を得た。結果を表 2 に示す。

表 2 における「クラスタ数」の列は本手法により出力されたクラスタ数である。その横の括弧の数字は実際の語義数を示している。「制約数」の列は本手法において、ユーザに制約を付けてもらったペアの数である。「半教師」の列が本手法による結果である。値は正解率を示している。また「教師なし」の列は半教師有りの手法を用いずに、通常のクラスタリングを行った結果を示している。その際にはクラスタリングツール CLUTO を利用している。またこのときに指定するクラスタ数は、本手法により得られたクラスタ数とした。

表 2: 実験結果

単語	クラスタ数	制約数	半教師	教師なし
もの	4 (10)	66	0.391	0.309
問題	1 (4)	19	0.969	0.969
代表	3 (3)	35	0.858	0.667
前	3 (4)	24	0.855	0.371
関係	2 (3)	30	0.785	0.848
午後	3 (3)	27	0.634	0.444
自分	1 (2)	22	0.942	0.942
時代	2 (4)	28	0.653	0.550
子供	2 (2)	26	0.588	0.480
現在	2 (2)	26	0.974	0.707
社会	4 (6)	22	0.755	0.395
今	3 (4)	23	0.687	0.423
平均	3.25 (3.91)	24.6	0.757	0.592

表 2 をみると、「半教師」は「教師なし」よりもかなり正解率が高い。しかもユーザに要求した制約数も小さく、本手法の有効性が確認できる。

4 考察

4.1 初期のクラスタ数

本実験では対象単語のクラスタ数を大目に見積もる際に、その数を 20 に固定している。この値を 10 から 5 刻みで 100 まで変化させた結果を図 2 に示す。図 2 の横軸は初期のクラスタリングのクラスタ数を表し、縦軸は上図が 12 単語の正解率の平均、下図が 12 単語の制約数の平均を表す。

当然ではあるが、初期のクラスタ数を多くすれば、正解率は高くなる。ただしその場合、ユーザに課す制約数も多くなる。

つまりどの程度が妥当かは、対象の単語に依存する。対象の単語の語義数が多いと予想すれば、大きく取れば良いし、少ないと予想すれば、小さく取ればよい。この数を調整できることが、本手法の長所である。

[2] http://glaros.dtc.umn.edu/gkhome/views/cluto

[3] 起動するプログラムは scluster である。そこで使われているアルゴリズムはトップダウンにデータを 2 分割してゆく処理を、目的のクラスタ数が得られるまで再帰的に行う k-way clustering と呼ばれる手法である。

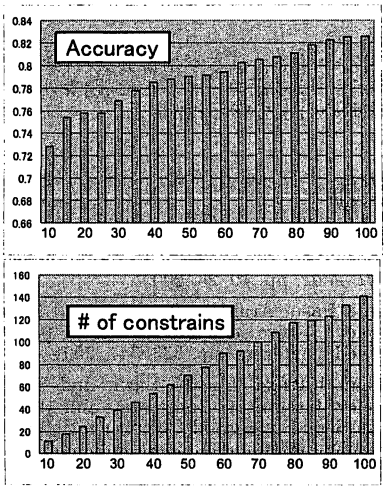

図 2: 初期のクラスタ数と制約数・正解率

4.2 用例間の類似度

本手法の精度には、用例間の類似度の定義が最も影響を及ぼす。本実験で定めた類似度はアドホックなものであり、更なる改良が必要である。特に今回は名詞だけを対象にしたので、簡易なもので済んだと思える。対象単語が動詞の場合の用例間の類似度を測るには、格の情報も影響するはずであり、構文解析も必要になるだろう。

また用例間の類似度を測る際には、シソーラスの情報が必須である。今回は分類語彙表を用いたが、更に広範囲かつ高精度なものを利用することで、類似度の精度が向上するだろう。

用例間の類似度の定義は今後の課題である。またその定義にはシソーラスが重要なので、本タスクに適したシソーラスを構築することも課題である。

4.3 クラスタリングの評価

本実験では正解率によってクラスタリングの評価を行ったが、タスクの目的から考えると、妥当ではない面もある。例えば、表 2 の実験結果では、「問題」や「自分」に対するクラスタ数が 1 である。それにも関わらず、正解率は 0.956 と 0.843 であり、かなり高い数値になっている。本タスクは語義別の用例を収集することが目的であり、メジャーな語義の用例を多数集めて、正解率を上げるよりも、むしろマイナーな語義の用例を発見できた方がよい。上記の例で言えば、クラスタをさらに分割して、「問題」や「自分」のもつマイナーな語義を発見できる方がシステムとしては有用である。

この点から考えて、各語義に対して等しい重みをつけ、クラスタリング結果を評価することを行ってみた。具体的には、対象単語の語義が m 種類存在し、入力となる用例のデータセットに i 番目の語義の用例が n_i 個あった場合、i 番目の語義を持つ用例の得点を $1/(mn_i)$ とする。クラスタリング結果では $1/(mn_i)$ の総和が最も高くなるように、各クラスタに語義の番号を付与する。このときの $1/(mn_i)$ の総和でクラスタリング結果を評価する。

表 2 のもとになったクラスタリング結果に対して、上記の評価値を調べた結果を表 3 に示す。

表 3: 別種の評価法

単語	半教師	教師なし	正解クラスタ数
もの	0.224	0.220	0.311
問題	0.250	0.250	0.422
代表	0.614	0.444	0.444
前	0.564	0.361	0.611
関係	0.565	0.586	0.625
午後	0.440	0.395	0.395
自分	0.500	0.500	0.608
時代	0.500	0.362	0.626
子供	0.564	0.559	0.559
現在	0.884	0.611	0.611
社会	0.404	0.556	0.568
今	0.321	0.443	0.477
平均	0.486	0.441	0.521

「半教師」の列が本手法を上記の評価法で評価した場合の値である。「教師なし」の列が通常のクラスタリングを行った結果に対して、上記の評価法で評価した場合の値である。「教師なし」と「半教師」との差は小さいが、「半教師」の方が優れている。参考までに、正しいクラスタ数を与えて、通常のクラスタリングを行い、その結果に対する評価値を表 3 の「正解クラスタ数」に示す。「正解クラスタ数」が最もよい値を示す。つまりここでの評価法を用いる場合、クラスタ数を正しく推定できることが重要である。本手法は正解率を上げる戦略をとったが、今後はここでの評価値を上げる戦略を検討する必要がある。その際には、クラスタ数を正しく推定することが鍵となる。

4.4 制約を得る順序

本手法では、制約を得る順序、つまりユーザに提示する代表点のペアの順番は任意でかまわない。しかしこの順序を変更することで、ユーザが制約を付与する回数を減らせる可能性がある。前述したアルゴリズムでは、現在の処理対象のクラスタの代表点 a と最終のクラスタリング結果を保持する集合 A 内のあるクラスタの代表点 b のペアがユーザに提示される。a と b が cannot-link である間は、集合 A 内のクラスタが順次試される。もしも a と b が must-link となれば、その時点で、対象のクラスタについての処理は終了する。そのため対象のクラスタと must-link になる集合 A 内のクラスタを、できるだけ早く見つけることができれば、ユーザが制約を付与する回数を減らせる。

単純には集合 A 内の各クラスタの代表点と処理対象のクラスタの代表点の類似度を調べ、類似度の高いものから順次ユーザに提示すればよい。

このような処理を行った実験も行ったが、結果的には制約数は若干増え、減らすことはできなかった。本手法の場合、代表点間の類似度はほとんどの場合 0 であり、ここでの工夫は効果がなかった。ただし、類似度の定義ももっと緻密なものにできれば、このような工夫も生かされるはずである。

4.5 関連研究

従来の半教師有りクラスタリングに対する本手法の位置づけを述べる。

従来の半教師有りクラスタリング手法は制約ベースの手法と距離ベースの手法に大別できる。制約ベースの手法とは通常のクラスタリングの目的関数に制約項を含めた新たな目的関数を定義し、与えられた制約を満足するようにクラスタリングを行う手法で

ある [1]。代表的な研究として Wagstaff らの提案した COP-kmeans (Constrined K-means) という手法がある [11]。そこではデータが制約を満たすように k-means でクラスタリングを行う。また距離ベースの手法とは、データ間の距離を、制約を考慮した形で再計算し、その距離を使って通常のクラスタリングを行う手法である [12]。代表的な研究として Klein らの提案した CCL (Constrined Complete-link) という手法がある [7]。そこでは must-link の制約を持つデータ間の距離を 0、cannot-link の制約を持つデータ間の距離を ∞ とし、更に must-link に関連したデータの距離を適切に修正する。最終的にデータ間の距離行列を作成して、その行列を使って階層的クラスタリング手法である complete-link [6] でクラスタリングを行う。またこれらの混合手法として Bilenko らは MPCK-means という手法を提案している [3]。これはデータの制約をクラスタリングの目的関数に含め、さらにその目的関数には素性の重みも加味されている。クラスタリングの繰り返し毎に素性の重みが学習されてゆく形になっている。

これらの手法は、本質的に、データのペアに制約が与えられた後に、そのデータの近傍のデータを調整する形になっている。このために制約を与えるデータのペアに対して柔軟に対応できる。しかしクラスタ数の推定を適切に行うことはできない。本手法では、制約を与えるデータが固定したクラスタの代表点なので、代表点の近傍のデータ、つまり固定したクラスタ内のデータの様子は変化しない。つまり初期のクラスタ内のデータの誤りは回復不可能となっている。このために本手法の精度は、初期のクラスタリングの精度に大きく依存する。ただし初期のクラスタリングで使うクラスタリング手法を問わないので、単語に応じたクラスタリング手法を使えるという長所もある。また初期のクラスタリングがある程度うまくいけば、必ず正しいクラスタ数が得られるという長所もある。

また半教師有りクラスタリングと能動学習的な試みを行った研究としては、上記の CCL と Basu らの研究がある [2]。CCL の能動学習ではクラスタを統合していく過程でクラスタ内の類似度が減少していくので、閾値に達したときにユーザからの制約を利用する形である。これはクラスタ数の推定を適切に行える保証がない。Basu らの研究では Explore と Consolidate という 2 段階の処理でクラスタリングを行うが、クラスタ数が未知の場合は、Explore のみを用いる。そこでは現在あるクラスタから最も遠い点を選んで、既存のクラスタとの代表点との制約をユーザから得ることで、クラスタリングが進行する。この手法は、本質的に、本手法と同様の処理を行っている。類似度行列が密である場合は、本手法と同様の結果が得られると考えられるが、本タスクのように類似度行列が疎である場合、最も遠い点を得るという処理ができない。また Explore の結果は最初に選ぶ代表点に依存する。

5 おわりに

本論文では語義別用例の収集を目的とした半教師有りクラスタリング手法を提案した。このタスクではクラスタ数の推定が重要であり、その点に特化した手法となっている。まず対象単語から大目の語義数（クラスタ数）を見積もり、そのクラスタ数を利用して通常のクラスタリングを行う。その結果、用例集が細かいクラスタに分割される。次にユーザからの制約を利用して、クラスタの統合を行う。本手法は少ない制約数で高い正解率を上げることができる。実験では名詞 12 単語に対して、その有効性を示した。精密な用例間の類似度の測定方法が今後の課題である。

謝辞

本研究の一部は、日本学術振興会 科学研究費補助金 特定研究「日本語コーパス」（課題番号 19011001）による補助のもとで行われた。

参考文献

[1] Sugato Basu, Arindam Banerjee, and Raymond Mooney. Semi-supervised Clustering by Seeding. In *ICML-2002*, pp. 19–26, 2002.

[2] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Active semi-supervision for pairwise constrained clustering. pp. 333–344, 2004.

[3] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. Integrating Constraints and Metric Learning in Semi-Supervised Clustering. In *ICML-2004*, pp. 81–88, 2004.

[4] David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised Clustering with User Feedback. Technical Report TR2003–1892, Cornell University, 2003.

[5] Donald Hindle. Noun classification from predicate argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics (ACL-90)*, pp. 268–275, 1990.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, Vol. 31, No. 3, 1999.

[7] Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. In *ICML-2002*, pp. 307–314, 2002.

[8] Masaki Murata and Masao Utiyama and Kiyotaka Uchimoto and Qing Ma and Hitoshi Isahara. Japanese word sense disambiguation using the simple Bayes and support vector machine methods. In *Proceedings of the SENSEVAL-2*, pp. 135–138, 2001.

[9] Resnik Philip. WordNet and Distributional Analysis: A Class-based Approach to Lexical Discovery. In *Proceedings of AAAI-92 Workshop on Statistically-Based NLP Techniques*, pp. 48–56, 1992.

[10] Hiroyuki Shinnou and Minoru Sasaki. Unsupervised learning of word sense disambiguation rules by estimating an optimum iteration number in the EM algorithm. In *Proceedings of Seventh Conference on Natural Language Learning (CoNLL-2003)*, pp. 41–48, 2003.

[11] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained K-means Clustering with Background Knowledge. In *Proceedings of ICML-2001*, pp. 577–584, 2001.

[12] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, pp. 505–512, 2003.

[13] 白井清昭. SENSEVAL-2 日本語辞書タスク. 自然言語処理, Vol. 10, No. 3, pp. 3–24, 2003.