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

2004－NL－162 (17)
2004／7／16

複数の筆者の表記の違いを利用した同義表現抽出

村上 明子 那須川 哲哉
日本IBM東京基礎研究所
〒242-8502 大和市下鶴間 1623-14
akikom@jp.ibm.com, nasukawa@jp.ibm.com

概 要
大量のテキストを分析し傾向を捉えるテキストマイニングにおいて、分析の観点で同義表現とみなせる語を代表的な表現に集約することは、表層表現の出現頻度のみで分析をするよりも有効である。そのためには、一般的な同義表現のみならず、分析対象の文書と目的に特化した同義表現の辞書が必要になる。本稿では、筆者別に分けられた文書集合を、表現の一貫性が保たれた文書集合として用いることにより、同義表現抽出の精度を向上させる手法を示す。我々は同一の筆者であれば一つの対象を表現するために、常に同じ表現を使う傾向があると仮定した。この仮定によれば、筆者別に分けられた文書内で似た文脈をもつ語あるいは表現のほとんどが類義表現であっても同義表現ではないといえる。コールセンターの対応記録データを用いて実験を行った結果、この仮定と本手法の有効性が示された。

Mining Synonymous Expressions using Personal Stylistic Variations

Akiko Murakami Tetsuya Nasukawa
Tokyo Research Laboratory, IBM Japan
1623-14 Shimotsuruma Yamatoshi
Kanagawaken 242-8502 Japan
akikom@jp.ibm.com, nasukawa@jp.ibm.com

Abstract
We present a text mining method for finding synonymous expressions based on the distributional hypothesis in a set of coherent corpora. This paper proposes a new methodology to improve the accuracy of a term aggregation system using each author's text as a coherent corpus. Our approach is based on the idea that one person tends to use one expression for one meaning. According to our assumption, most of the words with similar context features in each author's corpus tend not to be synonymous expressions. Our proposed method improves the accuracy of our term aggregation system, showing that our approach is successful.

1 はじめに

インターネットや社内イントラネット上に多数のテキストが存在するようになり、その内容全体の傾向や時間変化、そこにおけるコミュニティの存在などを把握するためのテキストマイニングが注目を集めている。多くのテキストマイニング手法において、単語の出現頻度が重要な情報となりえるが、表現の揺らぎ、単語の同義性、多義性といった問題により、表層の一致だけで単語の頻度を捉えると正しく内容が結果に反映されない。これを解決するひとつの手段として、文書中の各単語を代表的な同義表現に置き換えた上で単語の頻度を捉えることが挙げられる。著者らが開発したテキストマイニングシステム TAKMI (Nasukawa, 2001) は、単語の出現頻度だけでなく、名詞ー動詞のような依存関係の組を用いることにより、特定のトピックに関する傾向などの重要なパターンやルールを見つけ出すことができる。以下に PC コールセンターの応対記録に現れる文章と、そこから抽出された依存関係の組を示す：

- *customer broke a tp*
  → `customer...break`,
  `break...tp`
- *end user broke a ThinkPad*
  → `end user...break`,
  `break...ThinkPad`

この例では、(*customer, end user*) と (*tp, ThinkPad*) はコールセンターの記録文書においてはそれぞれ同じ意味を持つ表現と考えられる。したがって、コールセンターにおけるお客様の声の分析という観点においては、これらの二つの文章はほぼ同じ意味を示すと捉えるべきである。しかし、従来の動詞ー名詞の組で意味を捉える方法では、これらの文章は違う意味を持つと判断されてしまう。これは語の表現の多様性に拠るものである。同義性をもつ単語の別の例を以下に示す:

customer = cu = cus = cust = end user = user = eu
Windows95 = Win95 = w95

この問題を解決するために、分析の目的に対して同義と見なせる様々な表現に対して、辞書を用いて代表的な表現に置き換えるという方法を考える。このような、特定の文書の分析において同義表現と見なせる語の置換を、我々は異なる表現を集合・置換するという意味で **term aggregation** と名づけ、通常の同義語抽出と区別する。上記の例で、“customer” と “end user” とは、一般的には同義語とは見なすことはできない。しかし、メーカーのコールセンターの記録文書を分析するという目的の下では、同じ “customer” を意味する表現といえる。

この term aggregation のためには辞書が必要である。分析の観点で表現を統一すべき語は一般的な同義語とは異なるが、文章中では同じ役割をもつから、その性質を利用した同義語抽出手法を用いて辞書の作成を半自動化することができる。

一般的な同義語の抽出手法として、単語の周りの文脈(単語の周囲に存在する単語、あるいは単語の文法的役割)の類似度を見る方法があげられる (Hindle, 1990)、(Lin, 1998)、(Gasperin, 2001) 。同義表現も反義表現も単語の出現する文脈は類似しているため、提案されてきた手法で同義語と反義語を区別することは難しい。しかし、表現の一貫性のあるコーパスを用いれば、この問題は解決する。表現の一貫性がある例として、会社によってある製品や性能を示すための表記が会社によって規定されている場合などが挙げられる。このような表現の一貫性のある文書では、語の周辺文脈が似ている場合でも同義語ではない可能性の方が高い。このような表現の一貫性のあるコーパスを**コヒーレント・コーパス**と呼ぶこととする。これらのコーパスを用いて、term aggregation のための同義表現抽出を行う。

図 1 にコヒーレント・コーパスを用いた同義表現抽出の概観図を示す。図の右上は、同義表現にまとめられた語の集合を示す。この語を取り出すために、まず、全体のコーパス（これはコヒーレント・コーパスではない）から文脈が類似している語の集合を取り出す。これを図の左に示す。表現の一貫性のないコーパスから得られる文脈が類似している語の集合は、同義表現・反義表現ともに含んでいる。対照的に、コヒーレント・コーパスから取り出された文脈が類似している語の集合は、図の右下に示されているように同義表現を含まないはずである。文脈は似ているにも拘らず同義表現ではない語の情報を用いて、同義表現抽出の精度を向上させる。

コヒーレント・コーパスとしては、上記に挙げた会社の文書の例以外に、場所や著者の違いなどに基づいて分割したものが考えられる。本稿では、筆者別に分けられた文書を、表現の一貫性があると仮定し、コヒーレント・コーパスとして用いる。

本手法は、(1) 同義表現の候補の抽出、(2) ノイズ候補の抽出、(3) 候補の再評価の三つの段階からなる。我々は本手法の性能を評価するために、term aggregation 辞書構築のための同義表現抽出実験を行なった。実験結果によると、本手法は基本的な同義語抽出の手法と比較して再現率が若干落ちるものの、より良い F 値が得られることがわかった。

本稿の構成は以下の通りである。はじめに第 2 節で筆者ごとに異なる表現の一貫性の違いについて述べ、第 3 節では本システムの概要を示す。第 4 節で実験結果と考察を、第 5 節では関連研究を示し、第 6 節で今後の研究課題につ

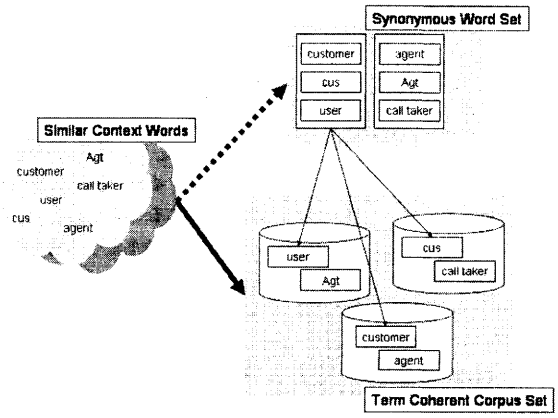

図 1: コヒーレント・コーパスを用いた同義表現辞書の構築の概要

いて言及した上で、本稿の結論を示す。

2 筆者別コーパス中での表現の特徴

我々の仮定においては、同一の筆者は複数存在する表現の中から 1 つの表現のみを用いる。この仮定を評価するため、あるコールセンターのコーパス中で筆者ごとの表現に関する調査を行った。このコーパスははパーソナルコンピュータのサービスコールセンター [1] において、コールテイカーが顧客対応の概要を記録したものである。

	筆者 A	筆者 B	筆者 C	筆者 D	筆者 E
cust	6	335	2	3	2
customer	31	62	32	31	286
eu[1]	345	89	179	402	62
user	5	20	2	3	13

1 eu は End User の略語

表 1: 5 人の筆者による “customer” の表記の頻度分布

表 1 はオペレーターが “customer” を表現するのに用いた語を示している。“customer” の意味を表現する表層表現の中で使用頻度の高い表現である “customer”、“cust”、“eu”、“user” など 4 種類について調査を行った。表 1 によると、各筆者は同じ意味の語に対して約 80%の割合で 1 種類の表現を利用する傾向にある。

この結果は我々の仮定がある程度は成立するものの、筆者別コーパスが完全なコヒーレント・コーパスではないことを示している。一つの意味に対しての表現は約 80%の割合で統一されているが、20%弱では別の表現が用いられている。この原因として、他の筆者が書いた文書をコピー・ペーストしたり、他の文章に影響を受けたりすることが考えられる。しかし、筆者別に分けられたコーパスは扱いに注意をすればコヒーレント・コーパスとして同義表現抽出の精度を向上させるために用いることができる。このことを以下の実験で示す。

[1] IBM PC ヘルプセンター

3 実験

3.1 実験に用いたデータ

実験では、コールセンターに蓄積された約 1 か月分、約 5 百万語からなるデータを用いた。データはすべて英語で記述されており、このデータから 29,961 種類の名詞、11,737 種類の動詞、3,350,200 組の名詞-動詞の依存関係の組を抽出した。また、表現の一貫性のあるコヒーレント・コーパスを取り出すため、この全体のデータの中からデータ量の多い順に 10 人のオペレーターによって書かれたデータを選んだ。このとき、各コーパスの名詞-動詞の依存関係の組の平均数は 37,454 である。

この実験の目的は term aggregation に用いるための同義表現集合を取り出すことにある。評価を行うために正解セットを人手で作成した。評価用正解セットは 10 の表現集合について作成され、一つの集合の表現の平均個数は 7.8 個である。

本提案手法では、ある入力単語に対して同義表現の候補を抽出する。このような同義表現抽出システムの性能は入

力となる語をどう選ぶかに依存する。本稿では、作成した正解セットの中で全コーパス中最も多く現れる語を選んだ。例をその集合の代表的表現とともに表 2 に示す。

代表的表現	表現集合
customer	customer, cu, cus, cust, end user, user, eu
HDD	harddisk, hdd drive, HD, HDD, hdds, harddrive, hd, H.D
battery	Battery, batteyr, battery, battary, batt, bat
screen	display, monitor, moniter, Monitor

表 2: 人手で作成した評価用データの例

今回実験に用いたコールセンターのデータは、オペレーターが電話を受けながら記録したものであり、略語、スペルミス、大文字・小文字の非統一といった問題が数多くある。そのため、通常のパーザーでは多くのエラーが発生する。

このように未知語や文法の間違いを多く含む文章に対し頑健性が重要であるため，浅い解析が最適である。そこで，我々はマルコフモデルに基づいた統計パーザーを使用した[2]。このパーザーの品詞付けの部分は (Charniak, 1993) に記述されているものと本質的に同じである。訓練コーパスとしては、人手によって正解を付与されている Penn Treebank corpus[3] の Wall Street Journal の記事が用いられた。この統計 Tagger は訓練コーパスに存在しなかった語はすべて名詞として扱うようになっている。次に、この統計 Tagger による品詞付けの結果を用い主辞決定規則 (Collins, 1999) に基づいて依存構造木を作成し、名詞-動詞の依存関係の組を抽出した。

3.2 筆者別コーパスの表現の一貫性を用いた同義表現抽出

本稿では、同義表現を取り出す対象を名詞（あるいは名詞が並んだ単純な名詞句）に限る。本手法は以下の三つの段階からなる：

- 名詞特徴ベクトルの生成
- 同義表現候補とノイズ候補の抽出
- 再評価

[2] このパーザーは CCAT と呼ばれており、IBM Watson Research Center で開発された。

[3] http://www.cis.upenn.edu/ treebank/

3.2.1 名詞特徴ベクトルの生成

単語の類似性に関する研究は今までに数多く行われており、その中でも主なアプローチとしては文書中での文脈を比較することが挙げられる。文書の文脈は主に 2 種類が考えられる。一つ目は、単語の文での役割を見る手法で、品詞に関する言語学的な情報や文法的な句構造や依存構造を素性とするものである。2 つ目は単語の周囲にある任意の数の語を素性とするものである。本稿で提案した term aggregation で行いたいのは、厳密な意味での同義語による置換ではなく、分析の観点から「役割」などが同じ単語による単語の表現の集約である。このとき、同じ役割を持つ単語は似た単語を修飾したり、似た単語に修飾されたりする (Hindle, 1990; Strzalkowski, 1992) という仮定に基づいて、文法的な関係を考慮する必要がある。その一方で、複数の筆者によって書かれた文は様々な表現（例：前置詞や冠詞の選択）を含んでおり、単語の周囲の語を素性とする方法は適切ではないと考えられる。そこで、本手法では依存構造による文法的特徴だけを素性として考える。ここで用いる依存関係の組は名詞と動詞、名詞動詞間の関係からなる。この名詞-動詞の組を (名詞, 動詞 (それらの関係)) のように表記する。

- (customer, boot↓)
- (customer, shut off↓)
- (tp, shut off↑).

↓は名詞が動詞を修飾することを、↑は動詞が名詞を修飾することを意味する。これらの抽出された依存関係の組を使うことによって名詞の特徴行列を作る。特徴行列は名詞の特徴ベクトルの集合からなり、この名詞の特徴ベクトルの類似度を比べることによってベクトル空間モデルと同様に名詞の類似度を測ることができる。これらの特徴行列・ベクトルは全体のコーパスと著者別コーパスの集合それぞれについて作られる。

3.2.2 同義表現候補とノイズ候補の抽出

本稿では二つの名詞の類似度を、互いの名詞特徴ベクトルの cosine の積で定義する。結果は、システムへ入力した名詞と抽出された名詞の類似度によって順序付けられた名詞のリストとして得られる。

全体のコーパスからは同義表現の候補、筆者別コーパスからはノイズの候補が取り出される。このときに比較対象となる名詞の特徴ベクトルは、全体のコーパスから作られた名詞の特徴行列の中にある、入力された名詞の特徴ベクトルである。全体のコーパス中の名詞と、この入力された名詞の特徴ベクトルを比較すると、取り出された表現は入

順位	同義語候補
1	batt
2	batterie
3	bat
4	cover
5	BTY
6	batterry
7	adapter
8	bezel
9	cheque
10	screw

表 3: "battery"の同義語候補 全体のデータからの結果

順位	同義語候補	順位	同義語候補
1	battery	1	batt
2	controller	2	form
3	Cover	3	protector
4	APM	4	DISKETTE
5	screw	5	Mwave
6	mark	6	adapter
7	cheque	7	mouse
8	diskette	8	cheque
9	checkmark	9	checkmark
10	boot	10	process

表 4: 筆者別のデータから抽出された同義表現とノイズ候補

力された名詞の同義表現候補である。これをベースラインとする。

次に、ベースラインの結果からノイズ候補を筆者別コーパスから抽出する。このとき、筆者別コーパスより作られた入力名詞の特徴ベクトルを比較対象のベクトルとはしない。それは、そのコーパスの筆者が入力名詞をコーパス中で用いていない場合、入力名詞の出現頻度が低く、名詞特徴ベクトルとして正しく値が入っていると信用することができないからである。したがって、筆者別コーパスにおいては、全体のコーパス中における名詞の特徴ベクトルと、各々の筆者別コーパスより作られた名詞の特徴ベクトルを比較し、語の同義表現/ノイズ候補を得る。

我々の仮定により、筆者別コーパスより取り出された入力名詞に類似している名詞は、類似度の最も高い名詞以外はたとえ文脈が似ていたとしても同じ意味を持たない。したがって、類似度が2位以下の名詞のリストをノイズの候補とする。これらのノイズ候補のリストは、N 人の筆者がいれば、N 個のリストとして抽出される。

3.2.3 再評価

前節で取り出された同義表現の候補及びノイズの候補を用いて、再評価を行う。まず、同義表現の候補の中から、単純にノイズ候補を削除するという方法が考えられる。しかし、第2節で示したように筆者別のコーパスの中には、主に使われている表現のほかに、少数ではあるが他の表現も含んでいる。例えば、表1において筆者Bは "cust" を一番多く用いているが、他の表現である "customer" や "eu" も用いている。従って、上記の方法で取り出されたノイズ候補の中に、同義表現であるものも含んでしまう可能性がある。したがって、単純にノイズ候補を同義表現の候補の仲から削除すると再現率の低下につながる。このような過剰な削除を避けるため、段階的に評価を行う。そのために、「完全同義表現候補」「同義表現候補」「ノイズ」の三つのラベルを設定する。

まず第一に、全体のコーパスから抽出された同義表現候補すべてに「同義表現候補」を割り当てる。第二に、筆者別コーパスから抽出された名詞のうち、一番入力名詞に類

似している名詞を取り出す。第一のステップで取り出された「同義表現候補」にその名詞があれば、その名詞を「完全同義表現候補」とする。第三に、筆者別コーパスから抽出された名詞のうち、第二ステップで取り出した一番入力名詞に類似している名詞以外を、ノイズ候補とする。そのノイズ候補と同じ名詞が「同義表現候補」にあれば、その名詞を「ノイズ」とする。このとき、「完全同義表現候補」は評価の対象にはならず、仮にノイズ候補と一致していたとしても「完全同義表現候補」のままである。最後に、「完全同義表現候補」と「同義語候補」と割り当てられた語を、この同義表現抽出の結果とする。つまり、筆者別コーパスからはたった一つだけの同義表現候補を取り出すことができ、その候補はたとえ他の筆者別コーパスからノイズ候補として抽出されても、同義表現候補から外されることはない。これにより、再現率の低下を防ぐことができる。

3.3 具体例

本節では、具体例を挙げ、上記の方法を説明する。以下に "battery" を入力名詞とした実際の例を示す。第一のステップでは、全体のコーパスから同義表現候補を抽出する(表 3)。この同義表現候補中には、" battery" の同義表現が多数含まれているが ("batt"、"batterie" など)、同時にノイズ ("cover"、"adapter" など) も含まれている。まず、これらの名詞全てが「同義語候補」となる。

第二のステップとして、筆者別コーパスから「完全同義表現候補」と「ノイズ候補」を取り出す。2 人の著者からの抽出リストを表 4 に示す。筆者別コーパス内でもっとも入力単語に近い語は "battery" と "batt" であり、同義表現候補内のこれらと同じ語には「完全同義表現候補」が割り当てられる。候補リストの残りの語、 "cover"、"adapter"、"cheque"、"screw" などはノイズ候補であり、「同義表現候補」内のこれらと同じ語には、「ノイズ」が割り当てられる。最終的には「完全同義表現候補」と「同義表現候補」が結果として出力される。表 5 にその結果を示す。

batt
batterie
bat
BTY
batterry
bazel

表 5: 再評価の結果

4 実験結果及び考察

評価のために、一般的な評価基準として再現率 (Precision)、適合率 (Recall)、F 値 (F-measure) を用いた。再現率、適合率、F 値の定義は次のようである。

$$\text{適合率 } (P) = \frac{\text{システムで得られた正解数}}{\text{システムが正解だと回答した数}}$$

$$\text{再現率 } (R) = \frac{\text{システムで得られた正解数}}{\text{得られるべき正解数}}$$

$$F \text{ 値 } (F - measure) = \frac{2 \times R \times P}{R + P}$$

システムの性能を測るため、ベースラインである全体からの同義表現抽出システムと、それを筆者別のコーパスからの情報を元に再評価したシステムにおける出力上位 N 個の名詞について、適合率と再現率を計算した。

4.1 予備実験 1：システムの抽出名詞数の決定

評価の指標として適合率と再現率を採用したが、これらは抽出する名詞の上位何位までを正解とするかによって値が変わる。この正解の判定基準を決めるため、予備実験を行なった。

本手法により、適合率は増加するが再現率は減少すると推察できる。そのため、適合率が十分に増加しきるところを抽出名詞数と定める。

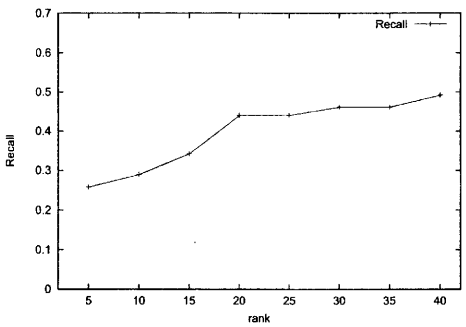

図 2: 同義表現抽出の抽出名詞数に対する再現率の変化

図 2 に、全体のコーパスにおける順位 N に対する適合率の変化を示す。この図より、抽出名詞数が 20 を超えると、再現率が横ばいであることがわかる。したがって、全体のコーパスから同義表現抽出する際には抽出名詞数を 20 に設定する。

同様に、筆者別コーパスよりノイズ候補を得るために、筆者別コーパスにおける抽出名詞数を決める必要がある。

ノイズ候補は、全体コーパスでの結果からノイズとして取り除かれる名詞であるので、抽出名詞数を多くする弊害として考えられるのは本来ならノイズとして取り除きたくない同義表現が入ってくる可能性があるということである。表1で示したように、数は少ないものの、筆者別コーパスにも同義表現が含まれている。抽出名詞が少なければ、消すことのできるノイズも減りシステム全体の適合率を上げることが困難になる。逆に抽出名詞を多くすると、ノイズ候補の中に同義表現が入ってくる可能性がある。本稿では、抽出名詞の大小のトレードオフを考慮し、筆者ごとのコーパスにおける抽出名詞数も同じ20と設定した。

4.2 予備実験2：筆者別コーパスのみを利用した同義表現抽出

我々の仮定により筆者別コーパス内で入力名詞に最も似た名詞は同義表現である。これを正解とする手法を、本稿で採用している筆者別コーパスよりノイズ候補を取り出す手法と比較するために、実験を行った。

筆者の人数が増えるとその筆者の数だけ同義表現が取り出せる。それを正解とし、筆者の数を増やした際に再現率がどのように変化するかを表6に示す。

	全体	筆者3人	筆者5人	筆者10人
再現率	0.624	0.114	0.114	0.143

表 6: 筆者別コーパスの1位のみを正解とした時の再現率

この表より、筆者の数を増やしても劇的な再現率の上昇は見られない。これは、筆者の数を増やしてもその筆者が用いている同義表現が他の筆者の同義表現と重なるからである。また、ベースラインと比較しても非常に再現率が低い。このため、筆者別コーパスのみを使用しても、ベースラインからの精度向上は見込めないことがわかる。

4.3 本実験：ノイズ除去手法

本稿の提案手法は、3．2節で提案した3ステップに基づくノイズ除去手法である。これを評価するため実験を行った。結果を再現率・適合率・F値の値と筆者別コーパス数のグラフとして図3に示す。図における筆者数0は全体コーパスの結果、つまりベースラインの結果を示している。

この図はベースラインより、ノイズ除去手法の方が適合率が上がり、再現率は若干下がっていることを示している。これは、筆者別のコーパスが完全なコヒーレント・コーパスではないために、実際は同義表現として扱うべき名詞まで除去してしまっているためであると考えられる。それにもかかわらず、F値は向上し、本手法によりF値は（10筆者の場合）37%向上している。

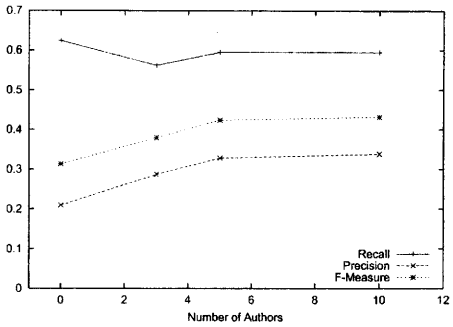

図 3: ノイズ除去手法における同義表現抽出の結果

5 関連研究

テキストから自動的に同義語・表現を抽出する手法には多くのアプローチがある。本稿で提案した手法は依存関係を単語の素性として用いている点で、語の文脈を単語の素性として用いている (Hindle, 1990)、(Lin, 1998)、(Gasperin, 2001) と似ている。他のアプローチとしては単語をクラスタリングするために、その単語の分布を用いるというもの (Pereira, 1993) がある。井上ら (Inoue, 1991) は単語の分布ではなく、パラレルコーパス内での日本語-英語の単語の組での分布の情報を多義語の問題を解決するのに適応している。

Wu (Wu, 2003) はフレーズレベルの同義表現を集めるために翻訳の情報を用いるアプローチを示した。今回、本稿では同義表現である語のみを考えたが、今後はフレーズレベルの同義表現も term aggregation の対象となるべきであろう。

省略語も、term aggregation を行う際に無視することのできない重要な問題である。Youngja (Youngja, 2001) は、自動的もしくは人手により生成されたパターンに基づいたルールを用いることによって、省略語とその定義をテキスト内から見つけるための手法を提案している。

本稿では筆者による表記の違いを、同義表現抽出の再評価に用いたが、これを用いる研究もいくつか行われている。最も有名なのは文書の著者同定である (Thisted, 1987)。この研究においても、筆者によって表現の一貫性が保たれていることが仮定されている。

6 まとめと今後の展望

本稿では、分析の観点では同義表現とみなせる表現を、代表的な表現に集約する term aggregation を行うにあたり、表現の一貫性が保たれたコヒーレント・コーパスを用いて同義表現抽出の精度を向上させる手法を提案した。コヒーレント・コーパスとしては、同一の筆者においては一つの対象を表現するのに常に同じ表現を使う傾向がある、という考えに基き、筆者別に分けられたコーパスを用いた。

筆者ごとの表現の違いが表現の種類を多くし、多様な表現を含む原因であったと考えられるが、本手法では、逆にこの表現の違いを用いて同義表現の抽出の精度を向上させたのである。ある特定の前置詞は、単語の類似度を量るための情報として重要である、と Gasperin (Gasperin, 2001) は示唆しているが、前置詞も筆者によって変わる可能性が在る。前置詞の種類によって、筆者ごとの特徴となるのか、名詞の同義性を測る特徴となるのかを考察することが今後の課題として考えられる。

本稿中では同義表現を中心に論じたが、同様の手法で省略語や頻出するスペルミスの抽出もできる。これらも term aggregation のターゲットとみなされるべきものである。本手法では単語の文中での役割に基づいた類似度だけ言及したが、省略語を抽出するためには文字列としての類似度も考慮に入れる必要があるだろう。

対象となる文書に関しては、本稿はにおいてはコールセンターの記録を用いたが、表現の統一されたコヒーレント・コーパスとしては他の領域におけるデータにも適応できると考えている。例えば、特許出願データは多様な表現を含んでいるにもかかわらず、表現は会社によって使用する専門用語が統一されている傾向が強いことから、適用対象になることが期待される。その一方で、e-mail などコミュニケーションの記録は、筆者がはっきりわかっているにも関わらず、使う表現は他の筆者の発言などに影響されるため、あまりこの手法にふさわしくないと考えられる。本研究では特定のデータのみを扱ったが、今後の課題として、文書の性質を調べて本手法にどのような影響を与えるか調査することが挙げられるだろう。

References

Michael Collins 1999. Head-Driven Statistical Modelsfor Natural Language Parsing *PhD Dissertation, University of Pennsylvania*

Charniak, E. 1993. *Statistical Language Learning.* MIT press.

Caroline Gasperin, Pablo Gamallo, Alexandre Agustini, Gabriel Lopes, and Vera de Lima 2001. Using Syntactic Contexts for Measuring Word Similarity *In the Workshop on Semantic Knowledge Acquisition & Categorisation (ESSLLI 2001)*

Donald Hindle 1990. Noun Classification From Predicate-Argument Structures. *Proceedings of the 28th Annual Meeting of ACL*, pp.268-275

Naomi Inoue 1991. AUTOMATIC NOUN CLASSIFICATION BY USING JAPANESE-ENGLISH WORD PAIRS. *Proceedings of the 29th Annual Meeting of ACL*, pp. 201-208

Dekang Lin 1998. Automatic Retrieval and Clustering of Similar Words *COLING - ACL*, pp768-774,

Nasukawa T. and Nagano, T. 2001. Text analysis and knowledge mining system. In *IBM Systems Journal*, Vol. 40, No. 4, pp. 967–984.

Youngja Park and Roy J. Byrd 2001. Hybrid text mining for finding abbreviations and their definitions. *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pp.126-133

Fernando Pereira and Naftali Tishby 1993. DISTRIBUTIONAL CLUSTERING OF ENGLISH WORDS *Proceedings of the 31th Annual Meeting of ACL*, pp. 183-190

Strzalkowski T. and Vauthey B. 1992. Information Retrieval Using Robust Natural Language Processing. *Proceedings of ACL-92*, pp.104-111.

B. Thisted and R. Efron. 1987. Did Shakespeare write a newly discovered poem?. *Biometrika*, pp. 445–455

Hua Wu and Ming Zhou 2003. Synonymous Collocation Extraction Using Translation Information *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pp.120-127