Webと携帯端末向けの新聞記事からの文末表現の言い換え抽出

岩越 守孝† 増田 英孝† 中川 裕志‡

† 東京電機大学工学部 ‡ 東京大学情報基盤センター

1 はじめに

最近、種々の応用を睨んで言い換えの研究がさかんになっている [1]。要約も言い換えの応用分野として有力である。従来の文書要約は重要文の抽出が主体であった。しかし、抽出した文をさらに短縮することを目指す場合には言い換えが役立つ。

本研究では、文縮訳を目的とした文末における言い換え表現を抽出することを目的とした。そこで、言語の実際の使用例から自動収集するための言語資源としてWebに配信されている新聞記事と、これに対応した内容を携帯電話向けに発信している新聞記事に注目した。我々は、同じ内容が数十文字程度で構成された携帯端末向けの新聞記事 (以下、携帯記事) と数百文字程度で構成されている Web 新聞記事 (以下、Web 記事) が対応付けられた記事対応および文対応コーパスを有する。この研究で目的としている言い換えは「Web 記事の文 → 携帯端末向け記事の文」という方向性を持つ点である。以下では必要に応じて、言い換え操作の対象になる Web 記事の文からの抽出表現を「言い換え元表現」、対応する携帯端末向け新聞記事の文からの抽出表現を「言い換え先表現」と呼ぶ。

2 Web 記事と新聞記事の対応付けコーパスの利用

言い換え規則を機械的に取り出すためには同一内容の長短 2 文が大量に必要となる。そこで本研究では、大森らが Web から長期にわたって収集したコーパス [2] を佐藤らが文対応付け [3] したものを用いる。本節以下の実験では 2001 年 4 月 26 日から 2003 年 11 月 30 日までに収集した Web 記事と携帯記事から得た 48075 組の記事から抽出した合計 72203 組の対応文を対象に行った。携帯文は体言止めや文末が助詞で終わる文が多いという特徴が挙げられる。

3 文末言い換え表現の抽出

3.1 言い換え抽出の枠組

本節では携帯文と Web 文の対応文を用いて、言い換え先表現と言い換え元表現の対の抽出について述べる。例えば、

携帯文: コンピュータウィルス感染防止に有力な方法が 判明。

Web 文: コンピュータウィルス感染防止に有力な方法があることが、研究所の調査で 分かった。

という対応文があったとする。このとき文末に注目すると携帯文では 判明 で終わっているのに対して、Web 文では 分かった で文が終わっている。文の内容から要約に用いるには 分かった を言い換え元表現に、判明 を言い換え先表現に用いればよいことが、人間が見れば容易に判断することが出来る。

このような携帯文の文末にある言い換え先表現に対する言い換え元表現を Web 文から自動的に抽出する方法は、概略以下のようになる。

Step:1 対応文の携帯文文末から言い換え先表現を抽出する。この時、同じ言い換え先表現を文末持つ携帯文と Web 文で新たな集合を作成する。

Step:2 Step:1 で得られた Web 文集合の文末から言い換え元表現を抽出をする。

Step:3 抽出された言い換え元表現を語彙の分岐数、出現頻度、文字列長から正しい言い換え元表現が高得点になるような得点付けを行い、並び替えを行う。

Step:4 Step:3 の結果に対して、精度向上を目的とした言い換え元表現として不適切な表現の削除を行う。

すなわち、この処理では**言い換え先である携帯文文末の表現をまず決め、それに対応する複数の Web 文の言い換え元表現を推定する** という問題を解くことになる。

3.2 言い換え先表現の抽出および対応する Web 文集合の抽出

まず言い換え先表現を携帯文文末から抽出する。抽出のために茶筌 [4] を用いて形態素解析を行い、文末の 1 形態素を取り出す。しかしこれだけでは助詞や助動詞、動詞の場合は「も」や「た」や「示す」といった言い換え先表現として用いることが難しい表現や、

意味の範囲が広すぎるために言い換え元表現の抽出が困難な表現が抽出されてしまう。その問題はさらに1つ文頭方向の形態素を続けて抽出することにより解消できる。この方法により、言い換え先表現として4617個を抽出した。その中から頻度が上位20位までのものを表1に示す。

ここで抽出された言い換え先表現に対し、その言い換え先表現を文末に持つ携帯文に対応するWeb文を集めたものを言い換え先表現に対するWeb文集合と呼ぶ。Web文集合中の文の数を対応文数と呼ぶ。

表1: 言い換え先表現の例

抽出表現	頻度	抽出表現	頻度
高	1780	ている	505
安	1668	判明	422
発表	1118	方針	408
逮捕	967	見通し	402
」と	933	ため	399
会談	788	強調	378
表明	735	合意	341
死亡	629	開始	329
決定	538	検討	328
いた	513	批判	317

3.3 言い換え元表現の抽出

言い換え先表現に対応する言い換え元表現をWeb文集合内の各Web文の文末から抽出する。Web文を形態素解析し、文末から1形態素ずつ増やしながら表現を抽出する。言い換え元表現が含まれる長さとして十分な15形態素までを抽出する。

3.4 分岐数、頻度、文字列長に基づく言い換え元表現の得点付け

3.3節で抽出した言い換え元表現には言い換えとして正しい表現と正しくない表現が含まれている。そこで、抽出された言い換え元表現に対して正しい表現が上位に集中するような得点付けを行う。

得点付けを行うにあたってWeb文文末の特徴を説明する。図1に言い換え先表現が「発表」の場合の文末方向から文頭方向への分岐数と頻度の関係の例を示す。このグラフからは「発表した」までは分岐数が小さく、「を発表した」で分岐数が大きく、さらに「結果を発表した」となるとまた小さくなる。このように分岐数が大きい形態素から文末までの表現がよい言い換え元表現であると考えることができる。

この特徴を踏まえ、言い換え先表現に対応する言い換え元表現を抽出するための評価関数の構成要素として以下を用いる。

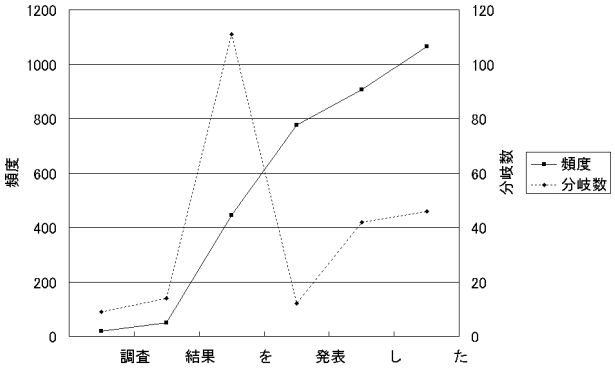

図1: 文末からみた分岐数と頻度の関係

分岐数: 先に述べた通り、分岐数の大小が言い換え表現句としての切れ目の可能性の大小を表すと考えられるので評価関数の構成要素として用いる。

頻度: よく使われる表現は安定していることを示すので、他の要因と組み合わせて用いることは有益である 。

文字列長: 言い換え元表現は短過ぎるならば言い回しにならず、長過ぎるならば文脈に依存した表現になってしまう。長過ぎもせず、短過ぎもせず、適度な長さの表現を抽出したい。そのため、評価関数では $\log($ 文字列長 $-1)$ として用いる。$\log$ により長い文字列に対して得点の抑制を、文字列長 -1 により1〜2文字の表現の排除をする効果がある。

上記の各要素を
$a =$ 分岐数，$b =$ 頻度，$c = \log($ 文字列長 $-1)$
とし、評価関数を $a, b, c, a \times b, a \times c, b \times c, a \times b \times c$ の7種類を用いて比較実験を行った。対応文数の多かった100位までの表現に対し、スコアが1位となる表現を人手で評価した。その結果からもっとも精度が良い評価関数は $a \times b \times c$ で71%、次いで $b \times c$ の65%という結果を得た。よって、評価関数 $a \times b \times c$ を用いた得点付けのデータを用いる。

さらに言い換え元表現として適切な表現が高順位になることを示すために、図2に言い換え先表現が「発表」の場合の正しい表現の分布を示す。低い順位にいくつか正しい言い換え元表現が表れているが、これは表現の頻度が少なかったために得点が低くなったことが原因である。

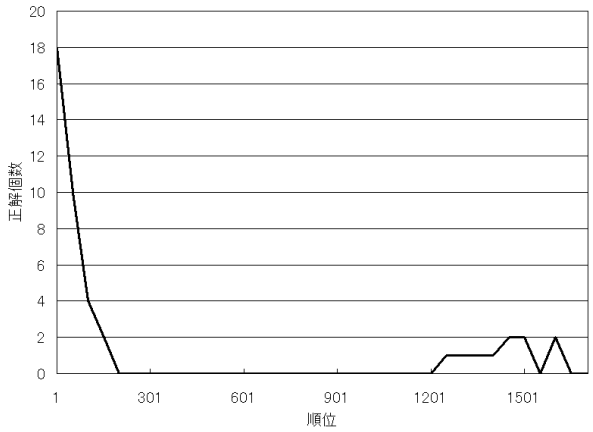

図 2: 言い換え先表現が「発表」の正解表現の分布

3.5 フィルタリングによる不適切な言い換え元表現の削除

言い換え元表現として抽出した表現の中には文の意味として必須な語まで含んでいるため、それを言い換えに用いた場合に意味が通らない文になってしまう可能性がある。得点付けによって順位が下位になる場合は採用されないが、収集した記事中でよく使われる表現であれば言い換え表現として正しくない表現も上位にきてしまう。そこでそのような表現を削除するためのフィルタリングを行う。その内容は次のようになる。

n を言い換え元表現の数とし、言い換え元表現の集合を $\{x_1, x_2, \cdots, x_n\}$，言い換え元表現を $x_i = W_{1i}W_{2i}\cdots W_{ni}$ (W_{ni} は形態素)，言い換え先表現を $y = W_{1y}W_{2y}\cdots W_{ly}$ (W_{ly} は形態素)，携帯文を $W_{1m}\cdots W_{jm}$ y，C を名詞とすると、

for(i = 1 , n)

if ($C \in \{W_{1m}, \cdots, W_{jm}\} \wedge C \in \{W_{1i}, \cdots, W_{ni}\} \wedge C \notin y$)

$then$ x_i を言い換え元表現集合から除く

具体例を図 3 に示す。ここで C は「声明」となり、削除対象となる言い換え元表現 x_i は「声明を発表した」となる。なお、y は「発表した」である。Web 文にも携帯文にも「声明」という語が含まれ、文の内容として必須の語であることがわかる。よって「声明」という意味を削除する「声明を発表した」は言い換え元表現として正しくないと考えられるため、言い換え元表現から削除する。

フィルタリングによって 3.4 節で得られたデータがどのように変化するかを、言い換え先表現が「発表」の場合を例にして表 2 に示す。表中で下線が引かれているものがフィルタリングによって削除された表現である。この表から、言い換え表現として用いるには不適切な表現が削除できていることが分かる。

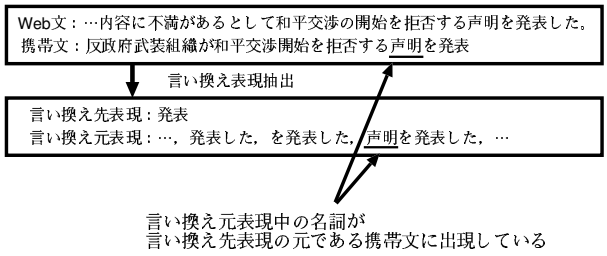

図 3: フィルタリング処理の具体例

表 2: フィルタリングによる「発表」の言い換え元表現の削除の例

を発表した	を明らかにした
と発表した	ことを明らかにした
発表した	たことを明らかにした
すると発表した	調査結果を発表した
たと発表した	明らかにした
したと発表した	策を発表した
結果を発表した	となった
声明を発表した	したことを明らかにした
計画を発表した	する声明を発表した

4 抽出された言い換え表現の評価

これまでの処理により抽出された言い換え表現についての評価する。表 3 に言い換え先表現の品詞ごとに抽出例をあげる。

次に言い換え表現の抽出精度について示す。そこでまず、精度の評価方法について説明する。精度は人手により評価を行っている。3 人が言い換え表現について正否を判定し、2 人以上が正しいと判断した場合を正解、それ以外は不正解としている。なお、グラフはそのままのデータでは見難いため 50 件で平均をとって表示をしている。図 4 は言い換え先表現が全品詞におけるフィルタ前後の精度の対数近似曲線のみを示したものである。ここでの順位とは、対応文数が多かった順である。全体的に精度が 12%向上し、特に対応文数が少ないところではかなりの精度向上していることが分かる。

最後に「精度」と「対応文数」の関係について図 5 に示す。用いるデータは対応文数の多い順に並べ、それぞれの言い換え元表現の得点付け順位が 1 位になった表現について人手で評価したものである。この図からは、言い換え先表現に対する対応文数が少ないと精度が低く、対応文数が増えるにつれて対数関数的に精度が向上していくことが分かる。つまり、この手法で

表 3: 言い換え表現の抽出例

会談 (名詞)	可能性も (助詞)	と語る (動詞)	述べた (助動詞)
と会談した	ている	を示した	と述べた
会談した	可能性がある	と語った	述べた
で会談した	可能性もある	と述べた	を示した
について意見交換した	している	語った	を述べた
と相次いで会談した	可能性が出てきた	考えを示した	を明らかにした

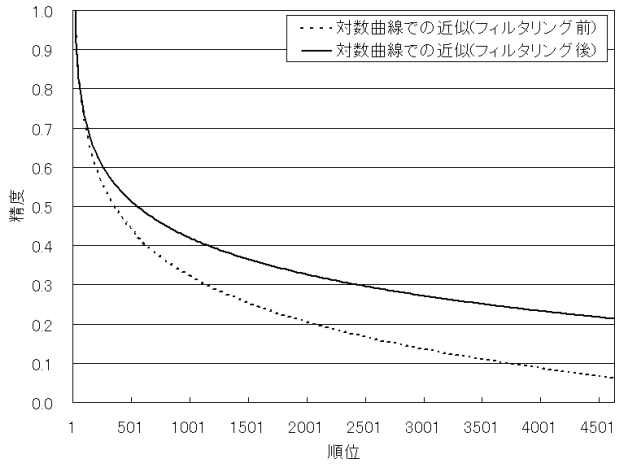

図 4: 全品詞時のフィルタリング前後の正解率の変化

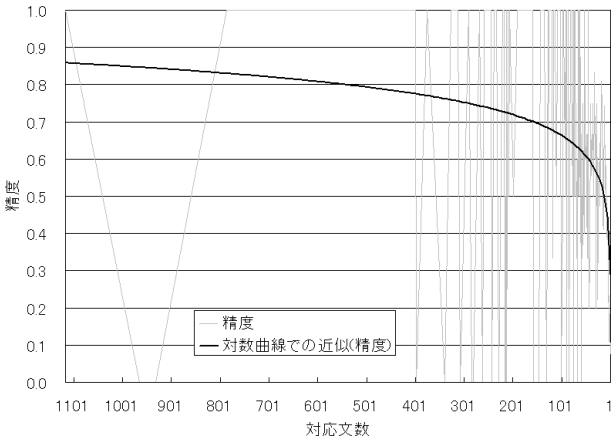

図 5: 対応文数と精度の関係

正解となる表現のより高い抽出精度を求めるならば、精度は対応文数に対し指数関数的な数のコーパスを集めなければならないということである。よって、今後は構文構造や意味内容を利用する精密な手法を用いることによる言い換え表現抽出を行う必要がある。

5 おわりに

本研究では携帯端末向け新聞記事と Web 新聞記事の対応付けコーパスから文末表現に関する言い換え表現の抽出方法を示した。まず、記事対応になっているデータから文単位での対応付けを行った。そして言い換え元表現の抽出を形態素単位で行い、分岐数、頻度、文字列長を用いた正しい言い換え元表現が高得点になるような得点付けと、精度向上を目的とした言い換え元表現のフィルタリングを行った。今後の課題としては、今回作成した携帯端末向け新聞記事と Web 新聞記事の対応付けコーパスを用いて、(1) 抽出された言い換え表現を用いた文縮約を試みることおよびその評価、(2) 精度向上を目的としたフィルタリング規則の追加、(3) 文末以外に現れる表現の言い換えの抽出実験、などがある。

参考文献

[1] 乾健太郎. 言語表現を言い換える技術. 言語処理学会第 8 回年次大会チュートリアル, pp. 1–22, 2002.

[2] 大森 岳史, 増田 英孝, 中川 裕志. Web 新聞記事の要約とその携帯端末向け記事による評価. 情報処理学会自然言語処理研究会, Vol. 153, pp. 1–8, 2003.

[3] 佐藤 大, 岩越 守孝, 増田 英孝, 中川 裕志. Web と携帯端末向けの新聞記事の対応コーパスからの言い換え抽出. 情報処理学会自然言語処理研究会, Vol. 159, pp. 193–200, 2004.

[4] 松本裕治, 北内啓, 平野善隆, 松田寛. 形態素解析システム「茶筌」version 2.2.9 使用説明書. 奈良先端科学技術大学院大学松本研究室, 2002.