自然言語処理 135−12
(2000. 1. 27)

配列アラインメントを用いた形態素レベルでの逸脱解消

中本 泰然　　目良 和也　　相沢 輝昭

広島市立大学 情報科学研究科
{uni, mera, aizawa}@nlp.its.hiroshima-cu.ac.jp

本研究では，チャットなどの書かれた話し言葉に対する日本語形態素解析において，配列アラインメントによる類似検索を適用した効果について報告する．IRC(Internet Relay Chat) の会話記録をもとに，1 ～ 3 文字程度の文字列の変形に対応するコストを人手により設定し，これを用いて動的計画法によるアラインメントを行なった．類似検索とコスト最小法を用いた形態素解析の結果，表記のゆらぎを含む試験文 296 の内，136 文 (45.95%) について正しい解析結果を得ることができ，入力文に対する解析誤りの割合は 54.05%，入力の全形態素数に対する解析誤りの割合は 7.05% となった．

Using Sequence Alignment to Improve the Morphological Analysis

Yasunari Nakamoto　　Kazuya Mera　　Teruaki Aizawa

Graduate School of Information Science, Hiroshima City University
{uni, mera, aizawa}@nlp.its.hiroshima-cu.ac.jp

In this paper, we describe effect of the approximate matching by sequence alignment for problem of Japanese morphological analysis of the written spoken language as chat etc. To calculate the gap cost in the dynamic programming algorithm, we measure the transformation rate of continuous character patterns, and the prototype system of the morphological analysis was made for trial purposes. We have evaluated the prototype system on the examination sentence from the conversation record of the IRC(Internet Relay Chat) and succeeded with 92.95% ratio.

1 はじめに

日本語の記述文を扱う自然言語処理システムにおいて，形態素解析による単語の分割は，最も基本的な処理であり，辞書検索を利用した形態素解析などでは，かなり処理精度の高いものも報告されている．しかし，これらのシステムは新聞記事などの書き言葉を対象としたものであるため，チャットなどの“書かれた話し言葉”に適用する場合には問題が生じる．話し言葉では縮約や冗長語・省略などといった形態素レベルでの逸脱が多く現れるため[1][2]，単純な辞書検索を用いた解析ではこれらの語に対応することが出来ず，単語の辞書引きの段階で解析精度の低下を招く．

本研究では，これらの問題に対応できるような“書かれた話し言葉”のための形態素解析システムを構築する．まず，実際のチャットにおいてどのような逸脱が生じているかを整理・分類した上で，配列アラインメントを用いた辞書の類似検索システムを作成し，これを元にした形態素解析の有効性について検討する．

以下，2章では書かれた話し言葉における単純な辞書検索の問題点を説明し，3章で配列アラインメントを用いた解決手法を提案，4章でプロトタイプを用いた実験内容について述べ，結果について考察する．5章は本論文のまとめである．

2 辞書検索における問題点

一般的な日本語形態素解析システムでは，まず単語辞書から得られる情報を元にして，文法規則などの適用が始まる．このため，辞書に存在していないものは未知語として特別な処理を施すことになる．

ところが，話し言葉の処理を考えた場合，単純なマッチングを用いた辞書検索では語の表記上のゆらぎを吸収することができないため，システムにとっての未知語が瀕出し，正しい解析が出来なくなる．また，書かれた話し言葉における表記上のゆらぎのパターンは非常に多岐に渡るため，これらを全て辞書に登録することは不可能である．

日本語形態素解析システム「茶筅」[5]を用いた解析例を図1に示す．

```
おお
 おお    オー    おお    感動詞
EOS
おー
 お      オ    お    接頭詞 - 名詞接続
 ー      ー    ー    名詞 - サ変接続
EOS
```

図 1: 茶筅による解析例

この例では，感動詞「おお」については正しく解析できているが，「おー」のように長音化した表記になると解析できない．そこで，このような書かれた話し言葉に現われやすいゆらぎのパターンを考慮した類似検索手法を導入する．これにより，形態素レベルの逸脱に対する柔軟な処理が可能になる．

2.1 形態素レベルでの逸脱の分類

ここで，書かれた話し言葉において実際にどのような逸脱がどの程度見られるのかを調査し，分類しておく．形態素レベルの逸脱は大きく分けて，変形に起因したものと，辞書に全く登録されていない未定義語の二つがある．それぞれについて実例を交えて示す．

2.1.1 変形に起因するもの

(a) 縮約・音便

縮約・音便とは，語が連接するときに発音しやすい別の音に変わることを指す．仮定縮約など，辞書によって登録されているものもある．ここでは，一般的な口語で用いられる縮約表現や撥音便，促音便などに加えて，一部の母音の長音化など何らかの音韻的な要素に起因した表記のゆらぎを指すものとする．表記上は変化していても，発音上は似通っているものが多い．

例) 「すげー」
「おめでとぉ～」

(b) 省略語

省略語とは，単語の一部を省略することにより短縮されたものである．辞書に登録されている名詞であっても，省略や短縮により未知語になる．

例) 「メアド」（メールアドレス）
「こばは」（こんばんは）

(c) 例外的カナ表記

例外的カナ表記とは，辞書に登録された見出し語と異なるカナ遣いを指す．漢字でしか登録されていない語をカナ表記したり，ひらがな・カタカナの独特な書き分けにより，計算機にとっての未知語が生じる．本来カタカナで記すべき単語をひらがなで書いたり，意図的に語尾をカタカナで書いたものが見られる．

例) 「こんぴゅーた」
「朝いちで映画館に向かうノダ！」

(d) タイプミス

タイプミスはコンピュータ上での文に特有のものであり，入力過程における人為的な原因によって生成された語のことを指す．キーボード入力の場合なら隣のキーを押す，キーの順序が入れ替わる，ローマ字入力でキーが抜けて英字がタイプされる，などのパターンが見られる．

例) 「恐怖症なんｄね」（なんだね）
「わりあ」（waria←warai）

(e) 言い淀み・言い直し

言い淀みとは，驚きを表現する際などに話しかけて途中で詰まる現象のことである．言い直しとは，言い誤りを訂正したり，より良い言い回しに変更するために話者自身が発話を中断してその部分を再度発話するものである．

例) 「そ、そんな」

2.1.2 全くの未定義語

(f) 冗長語

冗長語とは「えー」や「あのー」といった，意味を持たないとされる語である．これらはつなぎことば，遊びことばなどとも言われるものであり，情報的には不要とされることが多い．辞書に登録されている場合もあるが，あまりに多くの冗長語を登録するとかえって解析精度の低下を招く場合もある．

例) 「あー」
「むう」

(g) 擬音語・擬態語

擬音語は，自然の音響や動物などの音声を直接的に言語音に模倣したものである．擬態語は，擬音語の一種ともされるが，これは物事の状態や様子などを感覚的に音声化したものである．

例) 「ザーザー」
「ツブツブ」

(h) 固有名詞

固有名詞は非文法的現象では無いが，その多くは辞書に登録されていないために解析の妨げとなる．

例) 「五右衛門」
「ジョセフィーヌ」

(i) その他の未知語

書かれた話し言葉の中では，上記以外にフェイスマークなどの特殊な未知語が存在する．

例) 「(^-^;」
「（笑）」

2.2 出現頻度

チャットの一種であるIRC(Internet Relay Chat)における無目的な会話の記録から取り出した958文の発話に対し，前節で示したような現象の出現頻度について調査した．結果を表1に示す．

表1より，固有名詞や冗長語のような全くの未定義語が全体の発話の68.58%を占めている一方，縮約・音便などの変形に起因した未定義語も全体の48.12%の割合で存在している．本研究では，これらの形態素レベルでの逸脱のうち，縮約，音便，タイプミスなど，語の表記上の変形について解決することを考える．

表 1: 発話における形態素レベルでの逸脱数

種類	割合 (件数)
縮約・音便	36.85%(353)
省略語	4.80%(46)
例外的カナ表記	4.80%(46)
タイプミス	1.46%(14)
言い淀み・言い直し	0.21%(2)
固有名詞	28.71%(275)
冗長語	10.02%(96)
擬音語・擬態語	0.84%(8)
その他の未知語	29.02%(278)
問題なし	22.23%(213)

3 手法

前節までに述べたように, チャットなどの書かれた話し言葉で見られる逸脱を解消するための手法として, 配列アラインメントによる類似辞書検索を用いることを考える. まず, ゲノムサイエンスの分野における塩基配列のアラインメントについて触れ, 次に日本語におけるアラインメントの計算方法について考える.

3.1 塩基配列のアラインメント

突然変異による編集を受けたDNA塩基配列について, 同一の祖先を持つ複数の配列が突然変異によってどのような編集を受けたかを表すため, 配列への文字の挿入, または配列からの文字の削除のあった位置に"–" (ギャップ) を入れ, 配列の対応する位置を合わせる操作を行う. これをアラインメントと呼ぶ [3]. 表2に, アラインメントの例を示す.

表 2: アラインメントの例

S = ACTG → S' = ACTG–
T = AGGA → T' = AG–GA

配列 S の i 番目の文字を S[i] で表すとすると, この例では, S'[2] の C が T'[2] の G へ置換, T'[3] で T を削除, T'[5] に A を挿入する突然変異, あるいはその逆の突然変異が起こっている.

しかし, アラインメントによって得られる配列の数は膨大になるため, 多数のアラインメントの中から実際の突然変異を反映すると推定されるものを選び出す必要がある. このため, 各々のアラインメントに対し, それがどの程度よいものなのかをスコアで表す.

アラインメントのスコア S_{align} は (1) 式で表される.

$$S_{align} = f(n_{match}) - g(n_{gap}) \qquad (1)$$

n_{match} は文字の一致および置換の個数, n_{gap} は挿入されたギャップに応じたペナルティである. すなわち, 上の式では文字の一致または置換の個数が多いほどスコアが高くなり, ギャップが多いほどスコアが低くなる. これは, DNA塩基配列における塩基の挿入・欠失は非常にまれに起こることが知られているため, 挿入・欠失の回数が少ないアラインメントほどより確からしいと見なしていることによる.

また上の式の n_{match} は, 実際には分子構造が類似していたり, 科学的性質の近い塩基やアミノ酸どうしの置換は比較的頻度が高く, そうでない場合の置換は頻度が低いことを考慮して置換文字の組み合わせごとにスコアが設定するなどの補正が行われることが多い.

3.2 日本語のアラインメント

基本的な考え方は塩基配列の類似検索と同じく, 単語候補となる検索キーワード S の配列と辞書の見出し語 T を比較し, 文字の一致数から置換・挿入・削除といった操作によるギャップのコストを考慮して類似度を計算するものとする.

例として, 「こんちは～」と「こんにちは」の比較を表3に示す.

この場合, 「に」の削除ギャップ, 「～」の挿入ギャップに適切なコストが設定されていれば, 「こんちは～」という文字列が「こんにちは」の変形であることが検出可能となる.

塩基配列のアラインメントでは化学的性質や分子構造がギャップのコストに影響していた

表 3: 比較例

S = こんちは～　→　S' = こん－ちは～
T = こんにちは　→　T' = こんにちは－

ように, 日本語の場合にもコスト設定の指標を考える必要がある.

しかし, 前節で示したアラインメントの手法をそのまま日本語の処理に応用することを考えた場合, コストの設定において問題が生じる. 言語におけるギャップのコストは前後の文字の繋がりによっても変化すると考えられるため, 単純な文字対文字の類似スコアを用いることが出来ない.

例えば「い」と「あ」が長音に置換されるコストについて考えるとき, 「いいなあ」という文字列に対する「いーなー」, 「あいする」に対する「あーする」, 「ーいする」では, 異なった値を設定する必要がある.

本研究では, インターネット上で行われたチャットにおける実際の変形の例から, 人手によって 1 ～ 3 文字程度の文字列に対する変形コストを登録した類似度辞書を用意した. 辞書の記述例を表 4に示す.

表 4: 類似度辞書の記述例

原型	変形例	コスト
でしょ	っしょ	8
でえ	でー	2

類似度辞書は「原型」「変形例」「コスト」の三項目からなり, 1 ～ 3 文字の文字列に対しての置換コストが設定されている. 辞書に存在しない変換に対してのコストは, 漢字・かな・記号の文字種に応じた三種類のデフォルトコストを適用する.

3.3 計算モデル

二つの配列の対応づけを効率的に計算するため, 最適化問題を解くアルゴリズムとして動的計画法 (dynamic programming) を利用する. 長さの異なる二つの配列の対応づけを行なう場合, それぞれの文字に対して挿入, 削除, 置換の操作を行なうことが考えられる. 動的計画法を用いて類似度の最適解を求める場合, この三つの操作に対してコストを設定し, コスト合計が最小となる経路を選べば良い.

二つの配列の長さを L, M として, $L \times M$ の二次元配列 D を考える. D の各行及び各列は, 比較する文字列の各文字に対応するコストである. この D の各要素を, 次の漸化式を解くことによって決定する.

$$
\begin{aligned}
\mathrm{D(i,\,j)} = \min \{ \quad & \mathrm{D(i\text{-}1,\,j\text{-}1)} + \mathrm{s(i,\,j)}, \\
& \mathrm{D(i\text{-}1,\,j)} + \mathrm{del\_g(i)}, \\
& \mathrm{D(i,\,j\text{-}1)} + \mathrm{ins\_g(j)} \}
\end{aligned}
$$

ここで, $s(i,j)$ は i 番目の文字と j 番目の文字の置換コスト, $del_g(i)$ は i 番目の文字の削除ギャップコスト, $ins_g(j)$ は j 番目の文字の挿入ギャップコストである. また, min はこれらの内からコストが最小となるものを選択する操作である. 配列におけるこれらの関係を, 図 2に示す.

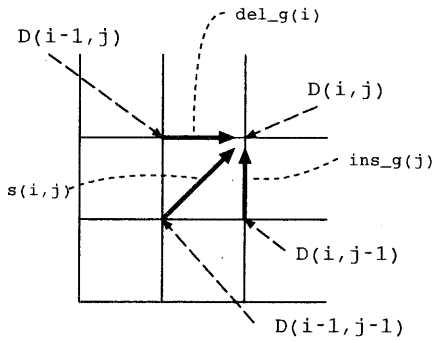

図 2: 動的計画法

配列中の点 D(i,j) において, 置換コスト・削除コスト・挿入コストの min を選択するということは, この点における局所的な最小パスを求めることになる.

置換コストである D(i-1, j-1) + s(i, j) が選択された場合には, i-1, j-1 を対応させた後に i, j を対応させることを意味する. 削除コスト D(i-1, j) + del_g(i) が選択された場合は, j は

既にある x (< i) と対応しており, i はギャップと並置される. 挿入コスト D(i,j-1) + in_g(j) が選択された場合は, i は既にある y (< j) と対応しており, j はギャップと並置される.

すなわち, この漸化式を解くことにより, 最もコストの合計が小さくなるような配列アラインメントを行なうことになる.

また, 前節で述べたように, 日本語のアラインメントの場合には置換・削除・挿入コストを計算する際, 前後の文字列との関連についても考慮する必要がある. 本研究では配列中の局所コストを計算する際, 類似度辞書を参照することにより, 後に続く文字列を含めた変形コストを算出する.

4 実験と評価

前節までに説明した配列アラインメントを用いて類似辞書検索を行ない, その結果を元にコスト最小法を用いて形態素解析を行なうシステムのプロトタイプを実装した.

プロトタイプシステムでは, まず入力された文字列に対し一形態素片づつずらしながら形態素辞書を引き, 可能性のある形態素を拾いあげる. 辞書引きの結果得られた形態素は, 単語コスト, 品詞情報, 置換・削除・挿入コストの総和 (ギャップコスト) を持っており, システムはそれらの中から, 単語コスト, ギャップコスト, 連接コストの総和が最小になるような解析結果を一つだけ出力する.

形態素辞書および連接コストの辞書としては, 形態素解析ツールキット LimaTK[4] に付属の, 変換済み茶筅辞書 (chadic 1.5) を使用した.

新たにインターネット上のチャットで採取した, 形態素レベルでの逸脱を含む 294 文を用いてプロトタイプの性能評価を行なった. 図3に入力文とその解析結果の例を示す.

評価の指標としてプロトタイプの出力した結果を人手による解析結果と比較し, 入力文の数に対する解析誤り文の数, 入力された全形態素数に対する解析誤りの個数について測定した. その結果, 296 文中の 136 文 (45.95%) について正しい解析結果を得ることができ, 入力文に対する解析誤りの割合は 54.05%, 入力の全形態素数に対する解析誤りの割合は 7.05% となった. また, 同一の辞書を用いて単純検索による解析を行なった結果との比較を表5に示す.

表 5: 実験結果の比較

		入力数	誤り個数	誤り率
類似	文	296	160	54.05%
	形態素	2453	173	7.05%
単純	文	296	271	91.55%
	形態素	2453	445	18.14%

解析誤りの要因について検討したところ, 複数の形態素にまたがった形での音便形に対応できていないためにミスにつながったもの, 未定義語を無理矢理既知の語に対応づけしてしまったもの, コスト設定上の問題などがあった.

複数の形態素にまたがる変形では, 特に「書 | い | て | い | る」→「書 | い | と | る」などのように, 2 つ以上の形態素がひとつの音に縮約される場合などに多く解析ミスが生じている.

未定義語の検出においては, 「ガメラ」のような辞書にない固有名詞を「カメラ」と認識するなど, コストの近い既知の語との区別において問題が生じている. また, 省略語についてはコストから原型を推定できているものは殆んど無かった.

コスト設定上の問題としては, 文字のデフォルトの置換コストが正しい解析における連接コストを下回るなどして, 誤った解析結果を導く原因となっていた. コストの数値設定によっては, 通常の文章の解析精度が低下することなどが考えられるため, この点についてはデフォルトのコストと連接コストの見直しを行ない, より多くのデータを採取して数値の調整を行なう必要がある.

なお, プロトタイプでは処理速度については度外視しているため, 一文の形態素解析に 15 秒から, 文字数によっては 240 秒を要している. 動的計画法では入力文字列の長さに比例した計算時間がかかるため, この点については今後の

お～、結構良いナァ

おお	[Y: おお POS: 感動詞]	10(+1) co:30
、	[Y:、 POS: 読点]	100(+0) co:30
結構	[Y: けっこう POS: 程度副詞]	99(+0) co:30
良	[Y: よ POS: 形容詞 - イ形容詞アウオ段 -X]	100(+0) co:0
い	[Y: い POS: 形容詞 - イ形容詞アウオ段 - 基本形]	0(+0) co:30
なあ	[Y: なあ POS: 終助詞 — なあ]	0(+10) co:30

「ずっと」ってとkろがいやん。

「	[Y: 「 POS: 括弧開]	100(+0) co:30
ずっと	[Y: ずっと POS: 時制相副詞]	99(+0) co:30
」	[Y:」 POS: 括弧閉]	100(+0) co:120
って	[Y:って POS: 副助詞 —って]	0(+0) co:30
ところ	[Y: ところ POS: 普通名詞]	100(+20) co:30
が	[Y: が POS: 格助詞 — が]	0(+0) co:30
いや	[Y: いや POS: 形容詞 - ナ形容詞 - 語幹]	100(+20) co:60
。	[Y:。 POS: 句点]	100(+0) co:15

図 3: 解析結果の例

表 6: 解析誤りの内訳

原因	件数	誤り率
未定義語の検出ミス	59	2.41%
連続した形態素の変形	49	2.00%
コスト設定の問題	34	1.39%
その他	31	1.26%

課題とする.

5 おわりに

チャットなどの書かれた話し言葉における形態素レベルでの逸脱を解消するための手段として, 配列アラインメントを用いた類似辞書検索を考えた.

これにより, 従来の単純辞書検索で対応することの出来なかった表記のゆらぎに対して92.95% の精度で解析することが出来た. ただし, 連接コストとギャップコストの兼ね合いにおいて, 通常の文章の解析精度が低下する可能性があることが解った.

今後の課題としては, コストの妥当性を検討した数値の調整, 未知語と変形・縮約の識別, 処理の高速化などが挙げられる.

参考文献

[1] 伝康晴 : 話し言葉における非文法的現象とその機械的処理, 人口知能学会研究会資料 言語・音声理解と対話処理研究会第 13 回チュートリアル講演 (1996)

[2] 丸山直子 : 話しことばの様相, 言語処理学会第 2 回年次大会チュートリアル資料 (1996)

[3] 松田秀雄 : 生命を構成する全遺伝子セットのコンピュータ解析, bit Vol.31, No.6 (June 1999)

[4] 形態素解析ツールキット LimaTK, http://cl.aist-nara.ac.jp/ tatuo-y/ma/

[5] 日本語形態素解析システム 茶筅 (ChaSen), http://cl.aist-nara.ac.jp/lab/nlt/chasen.html