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

2004－NL－159　(15)
2004／1／14

言い直しを含む日本語発話の解析手法に関する考察

原野　司※　　佐川　雄二※※　　杉江　昇※※
※ ※※ 名城大学大学院理工学研究科　〒468-8502 名古屋市天白区塩釜口 1-501
※ c302j020@ccmailg.meijo-u.ac.jp ，※※ {sagawa ，sugie}@ccmfs.meijo-u.ac.jp

あらまし　言葉を用いた自然な対話には，誤りを始めとする多種多様の不適格性が数多く現れる．しかし，対話システムを始めとする自然言語システムは，不適格性に対して非常に弱く，人間のような柔軟さはない．そこで本研究では，不適格性の一つである言い直しを含んでいても構文解析できるシステムを構築することを目指す．本稿では，言い直しを含む日本語発話の特徴に基づいて，（1）文法規則を追加することにより言い直しを解析する方法，（2）発話中の類似文字列を抽出することにより言い直しを検出する方法，を併用する解析手法を提案した．そして各手法を対話コーパスに適用し，評価実験を行った．

キーワード　言い直し，日本語発話，構文解析，形態素解析

Consideration about Parsing Japanese Utterances with Self-Repairs

Tsukasa HARANO※　Yuji SAGAWA※※　and　Noboru SUGIE※※
※ ※※ Graduate school of Science and Technology, Meijo University
1-501 Shiogamaguchi, Tempaku-ku, Nagoya, 468-8502 Japan
※ c302j020@ccmailg.meijo-u.ac.jp ，※※ {sagawa ，sugie}@ccmfs.meijo-u.ac.jp

Abstract　In spontaneous speech, speakers make many kinds of mistakes. However, the previous natural language processing systems including the spoken dialogue systems cannot cope with ill-formed utterances, and they have no flexibility like human. We aim to construct a parsing system that can cope with self-repairs, one of typical and frequent ill-formedness. In this paper, we proposed an analytical technique that combines two techniques based on the feature of the Japanese utterance including self-repair. And, we applied each technique to the conversation corpus, and did the evaluation experiment.

Keyword　Self-Repair, Japanese Utterance, Parsing, Morphological Analysis

1. はじめに

言葉を用いた自然な対話には，誤りを始めとする多種多様の不適格性が数多く現れる．しかし人間の聞き手は，不適格な発話でも，そこから話者の意図した意味を推測することが可能である．一方，対話システムを始めとする自然言語システムは，不適格性に対して非常に弱く，人間のような柔

軟さはない．その原因の一つとして，従来の構文解析技術は文法等が不適格性を考慮していない点が考えられる．音声発話において不適格文は非常に多く，音声による入出力を備えたシステムを入力する際に，不適格文は大きな障害になると思われる[1].

近年の音声認識技術の進歩により，話し言葉の解析は自然言語処理の中心的なテーマの一つになりつつある．しかし，今日の音声認識システムでは，一部言い淀みを扱うことができるソフトはあるものの，完全に不適格性を除去することは不可能である.

従来の構文解析技術でできることは，テキストに書き起こされた話し言葉の解析に限られており，不適格な表現が含まれていると，文を解析することですら不可能なことや, 言い直し以外の表現として解析され, 対処できないことが多い.

そこで，本研究では不適格性の一つである言い直しに着目した．本研究の目標は言い直しを含んでいても構文解析できるシステムを構築することである．本稿では，言い直しを含む日本語発話の特徴に基づいて,（1）文法規則を追加することにより言い直しを解析する方法,（2）発話中の類似文字列を抽出することにより言い直しを検出する方法，を併用する解析手法を提案する.

そして各手法を対話コーパスに適用し，評価実験を行う.

ただし，以下を条件とする.

- 書き起こした文を対象とし，韻律情報は用いない.
- 言い淀みは対象外とする.
- 文法は単一化文法とする.

第 2 章では（1）の方法について，第 3 章では(2)の方法についてそれぞれまとめ, その有効性を評価する．第 4 章ではそれらを併用する解析手法を評価する．第 5 章では総括を述べる.

2. 文法規則を追加することにより言い直しを解析する方法

2.1. 文法規則の提案

言い直しを含む発話の例を以下に示す. 但し，[ ]xを言い直される部分,[ ]zを言い直す部分とする.

（例1）：[私は]x[彼は]z買い物に行きます.

（例2）：私は[食べない]x[食べます]z.

（例3）：今日は学校[が]x[に]z行った.

これらの言い直しを含む文には，共通して次のことが言える.

- 言い直す部分は, 言い直される部分の直後に現れることが多い.
- 言い直す部分は, 言い直される部分と文法的に同一カテゴリであることが多い.
- 言い直される部分を削除した文は, 文法的，意味的に正しい文であることが多い.

これらの特徴を用いて，通常の日本語句構造文法に図 1 の規則を追加することにより，図 2 のように言い直しを含む文（例 1）を構文解析することができると考えられる.

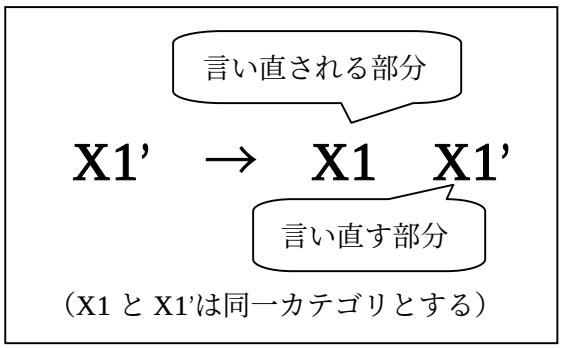

図 1　提案した文法規則

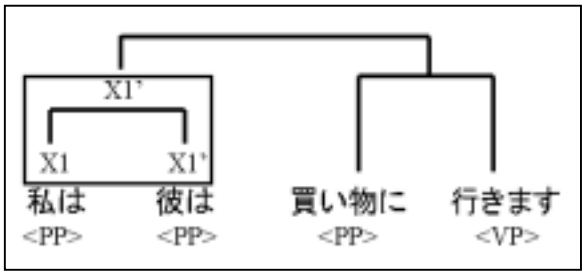

図 2　（例 1）の構文解析木

ここで言い直しを含まない場合について説明する．自然発話中には次のような言い直しを含まない（X1 X1’） が存在する．

（例4）：私は彼らは好きです．

（例5）：私は今日は買い物に行きます．

（例6）：私はその講義は受けます．

例えば（例 4）に先ほどの規則を適用すると図 3[上]のような結果になってしまい，図 3[下]のような望ましい結果にならない．

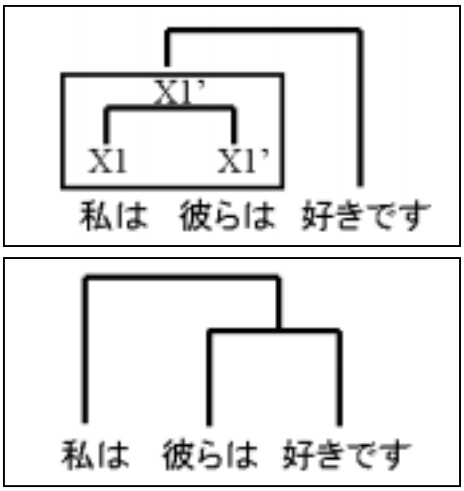

図 3　規則の適用結果[上]と
望ましい結果[下]

これは助詞の多義性を考慮していないからである．図 4 に示すように，（例 1）では“私は”，“彼は”はともに動作主として動詞句に掛かるが，（例 5）では“私は”は動作主，“今日は”は動作時制として動詞句に掛かる．このように“は”が文章中に複数存在する場合でも，“は”の役割によって言い直しを含むか含まないが違ってくる．

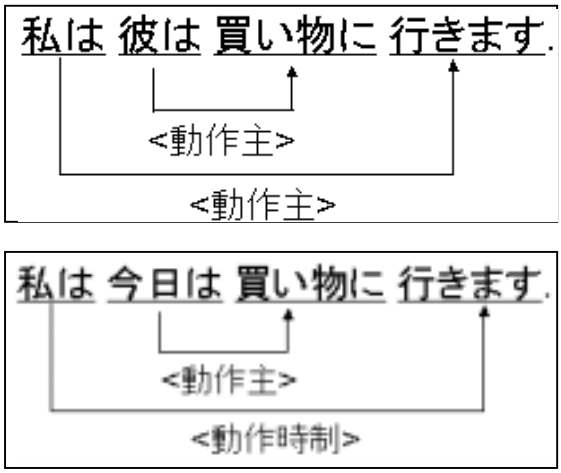

図 4　（例 1）[上]と（例 5）[下]の
句のかかり

（X1 X1’）が言い直しを含まないと判定する方法例を以下に挙げる．

- （X1 X1’）の一方に人称の単語（彼，太郎など）を含む場合・・・（例4）
- （X1 X1’）の一方に時間の単語（今日，来年など）を含む場合・・・（例5）
- X1’が知覚動詞の単語（知る，好きなど）である場合・・・（例 6）

これらの場合には，言い直しを含まない文と判断し，規則を適用しない．

2.2.　PC-PATR を用いた実験

規則の有効性を示すために，構文解析システム PC-PATR[2]を使用し，作成した辞書ファイルと文法ファイルを用いて実験を行った．その結果，言い直しを含む場合（例 1）～（例 3）と言い直しを含まない場合（例 4）～（例 6）を正しく解析することができた．（例 1）を構文解析した結果を図 5，（例 5）を構文解析した結果を図 6 に示す．

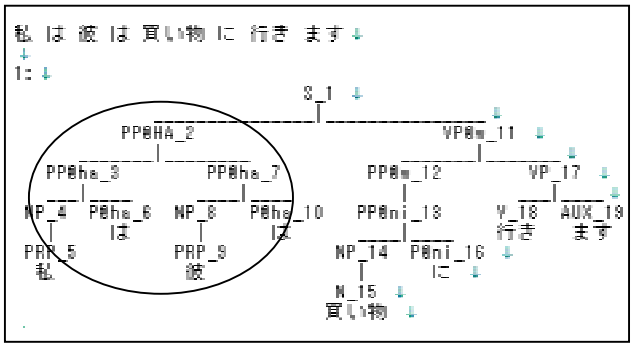

図 5　（例 1）の構文解析結果

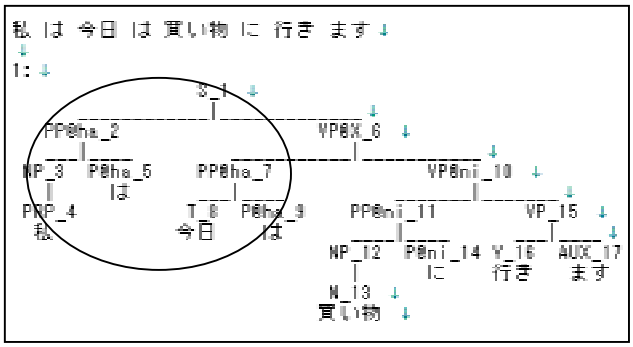

図 6　（例 5）の構文解析結果

2.3.　評価実験

次に本規則が実際の言い直しを含む文にどの程度有効であるかを調査した．しかし，構文解析の前段である形態素解析が言い直しを解析できなくては，本規則を適用することができない．そこで，構文解析だけではなく，形態素解析を含めた評価実験を行った．手順としては，まずコーパス（言い直しを含む日本語文）の形態素解析を行い，その結果（区切り，品詞）を用いて構文解析（規則の適用）を行った．コーパスには，（株）エイ・ティ・アール自動翻訳電話研究所作成の対話データベース ADD[3]を使用した．今回はその一部である，“ 国際会議の申し込みに関する参加者と事務局の電話による対話 ” 計 761 文より，言い直しを含む日本語文（111 文）を使用した．文中の言い直しを分類すると，以下の場合に分けられる．

- 言い直しが語の境界にある場合（39文）
  （例：[アメリカ]x[米国]zにおける・・）
  （例：お尋ね[申したいんですが]x
  [申し上げたいんですが]z，）
- 言い直しが語の途中にある場合（72文）
  （例：こういう[国]x[国際会議]zには・・）
  （例：チケット取るのにかかるかも
  [ご]x[わかりません]z.）

形態素解析器には，日本語形態素解析システム “ 茶筌（ChaSen）” [4]を使用した．

構文解析器には，PC-PATR を使用した．

規則の適用結果を表 1 に示す．

表 1　規則の適用結果

	○	×	△
語の境界（39文）	27	7	5
語の途中（72文）	27	6	39

○は正しく構文解析できた場合，×は正しく構文解析できなかった場合（形態素解析は成功），△は正しく構文解析できなかったが，前段である形態素解析が失敗している（思案と異なる）場合である．

誤の境界の場合では，7 割弱の文を正しく構文解析することができた．しかし，語の途中の場合では，正しく構文解析できたのは 4 割弱であり，△が半数を占めた．

△は言い直される部分が未知語や結果的に他の単語と置き換わってしまうことが原因であり，通常の形態素解析器では扱うことが難しいと思われる．その為，他の解析手法が必要となってくる．

3. 発話中の類似文字列を抽出することにより言い直しを検出する方法

3.1. 検出方法の提案

言い直しを含む発話の例を以下に示す．但し，[ ]x を言い直される部分，[ ]z を言い直す部分とする．

（例 7）：[大阪城]x[大阪城]z が目印に・・・

（例 8）：また[べっ]x[別途]z にですね，

（例 9）：[ど]x[どういう]z 会議なのか，

これらの例より，言い直される部分の文字列と言い直す部分の文字列は同じまたは近いことがわかる．そこで，言い直される部分と言い直す部分を比較し，文字列が一致または近ければ，言い直される部分を言い直しと判断することができると考えた．

そこで，次のような言い直し検出方法を提案した．

＜言い直し検出方法＞

言い直しを含む文の形態素解析を行い，ある形態素（以後，前部分）とその直後の形態素（以後，後部分）を比較する．そこで，比較した文字列同士が近ければ，前部分を言い直しとして検出する．これを全ての組み合わせで行う．ただし，形態素は複数の形態素から構成してもよいものとする．

ここで，文字列が近いことをどのように判断するかという問題がある．いろいろな方法が考えられるが，本稿では編集最小距離[5]の概念を利用する．これは平たく言えば，ある文字列から別の文字列に変形するのに必要な編集操作（挿入，削除，置換）の最小回数を示したものである．例えば，“ あお ”と“ あか ”の編集最小距離は 1 であり，“ りんご ”と“ なし ”では 3 になる．

以下に本手法で用いる，文字列の近さの定義を述べる．

＜文字列の近さの定義＞

前部分と後部分の形態素を先頭から一文字ずつ比較し，置換の必要が無ければ（同じ文字であれば）count に+1 する．これを前部分の文字数（以後，number）だけ比較し，count / number が一定値（以後，rate）より大きければ，比較した形態素同士の文字列は近いと判断する．

検出方法の具体例を図 7 に示す．

1. 形態素解析器“ 茶筅 ”を用いて言い直しを含む文の形態素解析を行う．
2. 解析結果を全てひらがなにして, 形態素ごとに分ける．
3. 前部分と後部分と取り出し, 文字列の近さを調べる．ただし, 比較する文字数は前部分の文字数とする．例えば,“ みな ”と“ みなと ”の場合では，先頭から 2 文字だけ比較する．
4. 一致した文字数 / 比較した文字数　の値（rate）を求める．例えば,“ みな ”と“ みなと ”の場合では，2 / 2 = 1.00 となる．
5. 3～4 の作業を考えられる組み合わせだけ行う．図 7 の場合では 4 つの組み合わせが考えられる．
6. rate の数値を決めて，言い直しを検出する．例えば, rate = 0.50 とすると,“ みな ”を言い直しとして検出する．

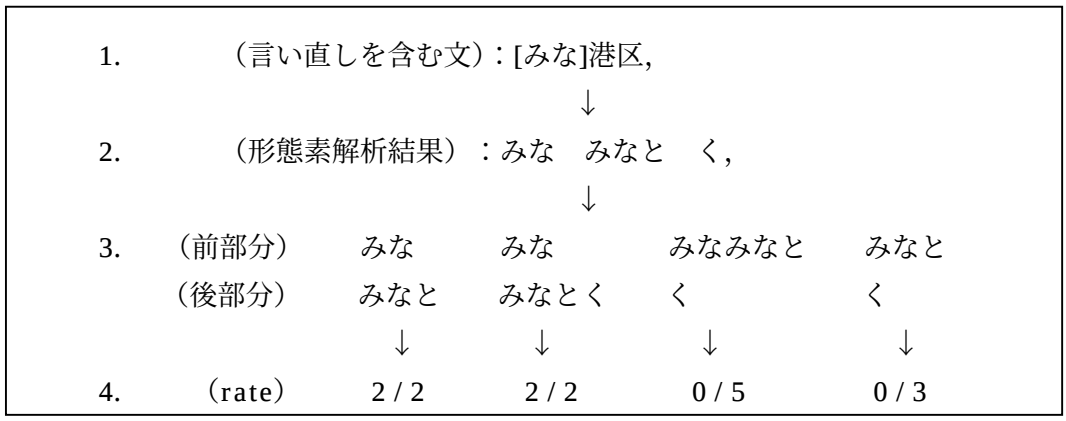

図 7　検出方法の具体例

3.2.　評価実験

次に本方法が実際の言い直しを含む文にどの程度有効であるかを調査した．コーパスには，2.3 節と同じく，言い直しを含む日本語文（111 文）を使用した．本稿では有効性の評価実験として，rate を変化させた場合，言い直しの検出（検出されたものが言い直しだった場合）・誤検出（検出されたものが言い直し以外だった場合）がどの程度になるかを調査した. 結果を表 2 に示す.

rate が高くなるにしたがって検出数が減るものの，誤検出数も減っており，検出数を重視する場合は rate = 0.50，誤検出数を重視する場合は rate = 1.00 が良いことがわかる.

表 2　言い直しの検出数・誤検出数

rate	総検出数	検出数	誤検出数
0.50	80	41 (41)	38 (30)
0.75	41	30 (30)	11 (11)
1.00	28	24 (24)	4 (4)

※ ( )は文単位換算時
※ コーパス中の言い直しは 111 個
※ コーパス中の形態素（言い直し候補）は 911 個

結果だけを見れば，rate = 0.50 の場合でも言い直しを検出できたのは 4 割弱であり，rate = 1.00 の場合では 2 割強しか検出できていない．しかし，本方法では 2 章で紹介した方法では検出できなかった言い直しを多数検出することができた.

そこで，2 章で述べた方法と本章で述べた方法を併用すれば，さらに言い直しを検出することができると考えた.

4. 二つの方法を併用する解析手法

4.1.　併用の手順

二つの方法を併用する解析手法の手順を図 8 に示す.

1. 言い直しを含む文の形態素解析を行う.
2. 3 章で述べた, 発話中の類似文字列を抽出することにより言い直しを検出する方法を行う. ただし rate = 1.00 とする. 本方法では誤検出が少ないことから, ここで言い直しを検出された文は言い直しの検出ができたものとして(誤検出はないものとして), 次の処理は行わない.
3. 2 章で述べた, 文法規則を追加することにより言い直しを解析する方法を行う.

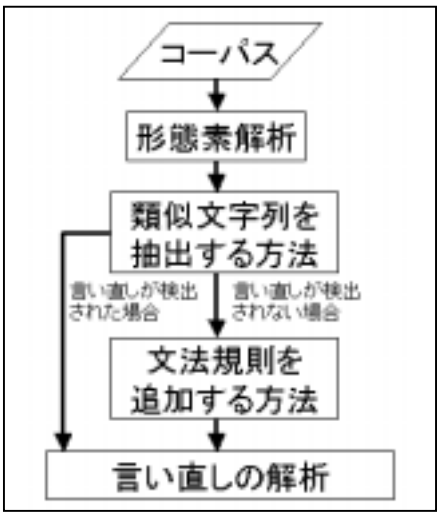

図 8　解析手法の手順

4.2.　評価実験

次に本方法が実際の言い直しを含む文にどの程度有効であるかを調査した．コーパスには，2.3 節，3.2 節と同じく，言い直しを含む日本語文（111 文）を使用した．結果を表 3 に示す．

表 3　解析手法の適用結果

	○	×
語の境界（39文）	28	11
語の途中（72文）	34	38

○は言い直しの解析ができたもの，×は言い直しの解析ができなかったものである．

言い直しの解析数は表 1 と比べると増えたものの，それほど多くならなかった．この原因の一つとしては，発話を形態素解析の対象としたため，その解析精度があまり高くなかったことが挙げられる．言い直し部分で形態素解析の区切りが間違っていたものが 16 文あり（これらの文から言い直しを検出することができない），他にも，品詞や読みの間違いなどがあった．よって，形態素解析後に何らかの処理を行い，形態素解析の精度を向上させれば，本手法の精度も上がると考えられる．

5. まとめ

本稿では，言い直しを含む日本語発話の特徴に基づいた解析手法を提案した．そして対話コーパスに適用し，その有効性を確認した．

今後の課題としては, 以下が挙げられる．

- 言い直しを含まない場合をさらに考え，機能を拡張していくこと
- 形態素解析の精度を向上させること
- さらに多くのコーパスを用いて評価すること

＜参考文献＞

[1] Yuji Sagawa, Noboru Ohnishi and Noboru Sugie：A Parser Coping with Self-Repaired Japanese Utterances and Large Corpus-based Evaluation , COLING-94, pp. 593-597,（1994）.

[2] Summer Institute of Linguistics, Int：PC-PATR Reference Manual,（2000）.

[3] 江原, 井ノ上, 幸山, 長谷川, 庄山, 森：ATR 対話データベースの内容, ATR テクニカルレポート，TR-I-0186,（1990）.

[4] 松本祐治 他：日本語形態素解析システム 茶筌（ChaSen）version 2.0 for Windows, 奈良先端科学技術大学院大学,（1999）.

[5] 田中穂積:自然言語処理－基礎と応用－電子情報通信学会, pp. 230-231,（1990）.