TR-IT-0283

文字連鎖の特徴を利用した誤り検出と訂正

Study on Detecting and Correcting Errors in Speech Recognition Using the Statistical Features of Character Co-occurrence

垣　　智
Satoshi Kaki

1998 年 10 月

内容概要

文字連鎖の統計的特徴を用いた訂正手法についてその内容と改良経過などを取りまとめた。また、新認識結果や人工的な誤りデータを用いて訂正手法の再評価を行った。さらに、これまでの改良結果を集大成してプログラム化した訂正サーバーの構築を行った。

ＡＴＲ音声翻訳通信研究所
ATR Interpreting Telecommunications Research Laboratories

目　次

はじめに

音声翻訳システムの性能を向上する上で、音声認識結果に含まれる誤りを訂正する機能を実現することは重要である。その機能実現のため、我々は文字連鎖の統計的特徴を用いた手法を提案してきた。提案手法には EPC と SSC があり、初期バージョンの発表以後、改良を重ねていった。本レポートは、これら訂正手法についてその内容と改良経過などを取りまとめたものである。

まず、第１章で初期バージョンについて述べ、第２章は初期バージョン以後の改良内容を改良経過に沿ってまとめている。第３章は改良の検討課題であったコーパス文の語順並び替えについて述べている。第４、５章では新認識結果や人工的な誤りデータを用いて訂正手法の再評価を行った。第６章ではこれまでの改良結果を集大成してプログラム化した訂正サーバーについて述べている。最後の第７章では、文字連鎖を利用した誤り検出の応用として機械的訳文評価について検討した結果をまとめている。

第１章　誤り検出・訂正

１．１　経緯

音声翻訳システムの性能を向上する上で、音声認識結果に含まれる誤りを訂正する機能を実現することは重要である。

脇田等[1]は誤りを含む音声認識結果を翻訳するために、用例に基づいた意味的距離から決定された依存関係を用いて、音声認識結果の中の正解部分を特定し、正解部分のみを翻訳することで高い翻訳率が得られたと報告している。また、塚田等[2]は n-gram に基づく統計的言語モデルと文法制約の両方を、文法的逸脱を許容しながら適用することで、信頼性の高い発話断片を得ている。

しかしながら、これら手法は音声認識結果の中の正しい部分を特定するだけで、含まれる誤りの訂正は行っていない。そこで、本稿では誤りや表現の傾向を利用した訂正手法を提案し、さらに、その評価について報告する。

１．２　訂正手法

提案する手法は二つの訂正処理から構成される。まず、音声認識結果は前段の訂正処理に入力され、その処理結果は後段の訂正処理の入力となる。後段では、前段で見逃された誤りの訂正処理を行なう。

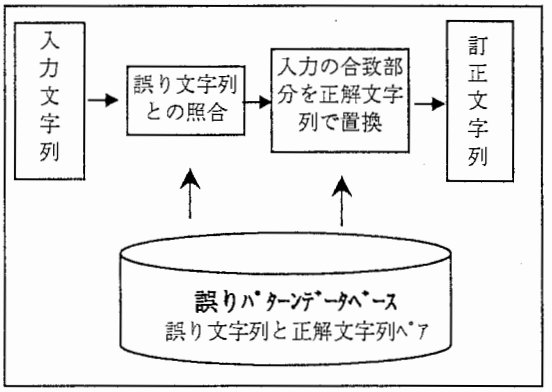

図 1-1 誤りパターン訂正のブロック図

前段の訂正処理は、誤りを含む音声認識結果と対応する正解文から抽出した文字列対を利用する。後段の訂正処理は、コーパスから抽出した文字列集合から、誤りを含む文字列をキーとして類似検索された文字列を利用する。それぞれ、「誤りﾊﾟﾀｰﾝ訂正」（EPC）、「類似文字列訂正」（SSC）と呼び、この順に二つの訂正処理を合わせて適用したものを EPC+SSC と書くことにする。

1.2.1　誤りﾊﾟﾀｰﾝ訂正（EPC）

音声認識誤りを眺めてみると、その誤りは全くランダムではなく、ある一定の傾向があることに気付く。本手法は、そのような誤り傾向（誤りﾊﾟﾀｰﾝと呼ぶ）を、誤りを含む音声認識結果と対応する正解文を用いてとらえ、誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽとして保存する。誤りﾊﾟﾀｰﾝは誤りを含む文字列とそれに対応する正解文字列のペアである（表 1-1 参照）。入力された音声認識結果に誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽ内の誤り文字列と同じ文字列があれば、該当部分を正解文字列で置き換えることにより訂正を行う（図 1-1 参照）。

表 1-1　　誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽ

＜正解文字列＞	＜誤り文字列＞
は何名様	はな名様
ますでしょうか	ますえしょうか
して頂きますので	してできますので
失礼いたします	していたします
お客様	お件様
ご希望	ご気後
支払い方法	支払いを方法
では失礼いたします	ですねいたします
ております	てります
和室の方	ますの方

1.2.1.1　誤りﾊﾟﾀｰﾝの抽出

誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽは、音声認識結果と対応する正解文から機械的に作成する。

誤りﾊﾟﾀｰﾝによる訂正は、誤りの検出と訂正がﾊﾟﾀｰﾝﾏｯﾁのみで行う単純な方式であるため、無制限の適用は誤訂正を招いてしまう。そこで本手法では以下のような条件をすべて満足した候補を誤りﾊﾟﾀｰﾝとして使用している。

①高頻度条件：候補の内、出現頻度が与えられた閾値（実験では２）以上のものを選ぶ。
②適格性条件：正解文と誤り文字列とのパターンマッチを行い、マッチするものは候補から除外する。
③包含条件１：マッチングに使用する文字列が長いほどより信頼できると仮定し、２つの誤りﾊﾟﾀｰﾝ候補の誤り部分の文字列において、一方が他方を包含し、かつ、出現頻度が同じならば、包含関係において大きい方の候補を残す。
④包含条件２：異なる発話から得られた候補で、互いに共通する部分があれば、その共通部分を取り出す。２つの誤りﾊﾟﾀｰﾝ候補で一方が他方を包含し、かつ、出現頻度が異なるならば（異なる発話から得られて候補とみなせる）、包含関係において小さい方の候補を残す。

図 1-2 に誤りﾊﾟﾀｰﾝの抽出例を示す。

（A）誤りﾊﾟﾀｰﾝの抽出は、誤認識部と対応する正解部が同じ事例を集めることから始める。図中の例は、<てお>を<た>と誤認識した事例を集めたものである。

（B）誤りﾊﾟﾀｰﾝ候補として、誤り部及び正解部を中心に部分文字列を切り出し、出現頻度が２以上の候補を生成する。

（C）適格性条件で誤り文字列が正解文に含まれる候補を除外する。ここでは、誤り文字列「た」、「たり」などの候補が正解と一致するので除外される。

（D）包含条件１によって、誤り文字列「お待ちしたります」に包含される「待ちしたります」、「したります」「お待ちしたり」などを除外する。

（E）包含条件２によって、誤り文字列「たりま」を含む候補、「お待ちしたります」、「となったります」などを除外する。

（F）以上の条件をすべて満足して、誤りﾊﾟﾀｰﾝとして最終的に残るのは、誤り文字列が「たりま」、「たきま」の二つのﾊﾟﾀｰﾝである。

(A) <てお>を<た>に誤認識した例

正解文	→	認識結果
一台押さえ<てお>きましょう	→	一でおさえ<た>きましょう
はご変更し<てお>きます失礼	→	はこれごし<た>きます失礼
では変更し<てお>きます	→	では変更し<た>きます
では変更し<てお>きますほか	→	では変更し<た>きますああ
頃お待ちし<てお>ります	→	がお待ちし<た>ります
たお待ちし<てお>ります	→	たお待ちし<た>ります
ゃお待ちし<てお>りますあり	→	つお待ちし<た>りますあり
はお待ちし<てお>りますあり	→	でお待ちし<た>りますあり
はお待ちし<てお>りますわた	→	はお待ちし<た>ります亜は
をお待ちし<てお>りますお電	→	用お待ちし<た>りますおで
間をつぶし<てお>りますので	→	とおつぶし<た>りますので
をお待ちし<てお>ります	→	うでのまし<た>ります
日お待ちし<てお>ります	→	ん日をまし<た>ります
番街に面し<てお>りますので	→	番街に面し<た>りますので
お安くなっ<てお>りますがイ	→	お安くなっ<た>りますがい
からとなっ<てお>りますがご	→	からとなっ<た>りますがお
インとなっ<てお>りますがご	→	インとなっ<た>りますがご
千円となっ<てお>りますがよ	→	に円となっ<た>りますがあ
千円となっ<てお>ります	→	五千となっ<た>ります
インになっ<てお>りますが	→	インになっ<た>りますが
料金になっ<てお>りまして和	→	料金になっ<た>りまして和
千円になっ<てお>ります	→	四千になっ<た>ります
千円になっ<てお>ります	→	子千になっ<た>ります

<>内は誤認識部と対応する正解部、太字及び下線付き太字は抽出された誤りﾊﾟﾀｰﾝを示す。

(B) 高頻度条件を満たす候補の生成例

<てお>りま → <た>りま
お待ちし<てお>ります → お待ちし<た>ります
し<てお>ります → し<た>ります

(C) 適格性条件の適用例

＜除外されるﾊﾟﾀｰﾝ＞
<てお>り → <た>り
<てお> → <た>

(D) 包含条件1の適用例

＜残るﾊﾟﾀｰﾝ＞
お待ちし<てお>ります → お待ちし<た>ります
＜除外されるﾊﾟﾀｰﾝ＞
待ちし<てお>ります → 待ちし<た>ります
し<てお>ります → し<た>ります
お待ちし<てお>り → お待ちし<た>り

(E) 包含条件2の適用例

＜残るﾊﾟﾀｰﾝ＞
<てお>りま → <た>りま
＜除外されるﾊﾟﾀｰﾝ＞
お待ちし<てお>ります → お待ちし<た>ります
となっ<てお>ります → となっ<た>ります
になっ<てお>りま → になっ<た>りま
し<てお>ります → し<た>ります

(F) 最終的に残るもの

<てお>りま → <た>りま
<てお>きま → <た>きま

図 1-2　誤りﾊﾟﾀｰﾝの抽出例

1.2.2 類似文字列訂正（SSC）

人間は、文中にある誤りに対して、誤り前後の文字の並びから、正しい表記を推測することできる。これは無意識に、誤り前後の文字列に類似した正しい表現をあてはめているためと考えられる。本手法は、このような類似表現を文字列ﾃﾞｰﾀﾍﾞｰｽから検索して、訂正に活用する手法である。文字列ﾃﾞｰﾀﾍﾞｰｽは正しい文に出現する文字列を集めたものである。

この手法では、最初に誤り検出[1] を行い、次に検出した誤りを含む文字列に類似する文字列を文字列ﾃﾞｰﾀﾍﾞｰｽから検索する。そして、最後に二つの文字列の差分に従って訂正を行う（図1-3 参照）。

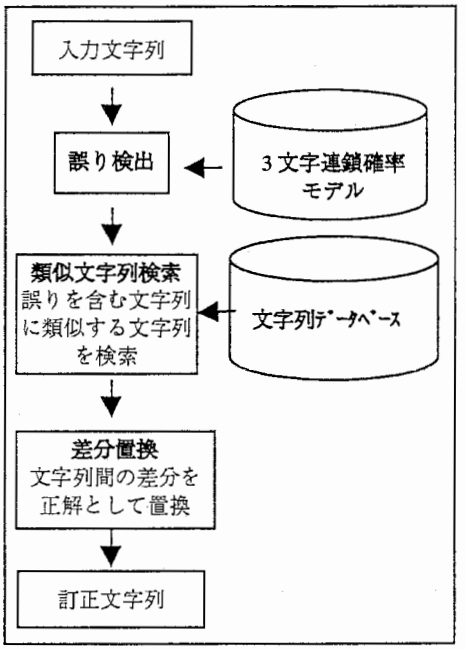

図 1-3 類似文字列訂正のブロック図

1.2.2.1 訂正手順

訂正手順を次の入力文字列を例に説明する。

入力文字列：「九月十四から十六までの二泊ですね五人背は何名様ですか」

誤り検出：入力文字列に 3 文字の文字連鎖確率モデルを適用すると、誤り「五人背」が検出される。

類似文字列検索：この「五人背」に前後M文字（ここではM＝５）を付け加え、文字列「二泊ですね五人背は何名様で」を作成する。この文字列をキーとして、文字列ﾃﾞｰﾀﾍﾞｰｽの中でもっとも類似し、かつ、与えられた閾値以上の類似条件を満たす文字列を検索する。その結果、文字列「ですね人数は何名様で」が最終的に選ばれる（以下、類似文字列と呼ぶ）。

類似文字列：「ですね人数は何名様で」

文字列データベースが大規模になった場合、類似文字列検索の処理速度が問題となるが、著者等は Lepage[7]の文字列近似照合アルゴリズムに基づいた高速検索プログラムを用いて対処している。

---

[1] 誤り検出に関しては、文字 3-gram の連鎖確率に基づく手法を用いた。この手法は、入力文字列の前方から一文字ごとに順次その連鎖確率を計算し、連鎖確率値が与えられた閾値以下である部分を誤りと見なすものである。予備実験の結果、検出精度は適合率 80%以上、再現率 70%以上であった（付録１参照）。

差分置換：次に、誤り「五人背」外の前後K文字（ここではK＝2）が類似文字列に含まれるかを調べる。下記の例では誤り「五人背」外の前後文字「です」と「何名」が類似文字列に含まれるので、その間に挟まれた「ね五人背は」を「ね人数は」で置き換え、訂正を行う。

検出誤り部と前後文字列：

[です]{ね<五人背>は}[何名]

類似文字列：[です]{ね人数は}[何名]様で

ここで、<>内は検出誤り部、[]内は前後文字列、{}内は置換文字列

訂正文字列：「九月十四から十六までの二泊ですね人数は何名様ですか」

1．3　評価実験

1.3.1 実験データ

音声認識結果データ：旅行会話ﾃﾞｰﾀﾍﾞｰｽ[6]の 4806 発話に対する音声認識結果を用いた。認識装置は、音素ＨＭＭと可変長Ｎ-gram 言語モデル[4]を使い、マルチパス探索でワードグラフを出力する連続音声認識方式[5]に基づくもので、認識装置から出力された尤度第一位の結果を用いている。表 1-2 にデータ諸元を示す。

表 1-2　評価に用いた音声認識結果諸元

発話数	認識率(%)（文字単位）	誤り数 挿入	脱落	置換	合計
4806	74.73	2642	1702	8087	12431

この音声認識結果 4806 発話の内、4321 発話を誤りﾊﾟﾀｰﾝ作成用に、残り 485 発話を評価用に使用した。

誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽ：上記音声認識結果 4321 発話より作成し、誤りﾊﾟﾀｰﾝの出現頻度は 2 回以上のものを用いた。 抽出された誤りﾊﾟﾀｰﾝ数は 629 個であった。

文字列ﾃﾞｰﾀﾍﾞｰｽと検出用 n-gram：文字列ﾃﾞｰﾀﾍﾞｰｽと検出用 n-gram の元となる発話は、旅行会話ﾃﾞｰﾀﾍﾞｰｽから上述した音声認識結果とは異なる会話セットを利用して作成した。文字列ﾃﾞｰﾀﾍﾞｰｽの文字列の長さは 10 文字で、出現頻度が 3 回以上のものを用いた。 抽出された件数は 16655 件であった。

表 1-3　文字列ﾃﾞｰﾀﾍﾞｰｽのデータ諸元

発話数	延べ文字数	異なり文字数
20176	570306	1396

1.3.2　評価方法

評価は次の 2 方法で行った。

機械的評価：訂正前後での誤り個数の変化を機械的に計数する。

理解度評価：理解度評価は訂正前後の認識結果と対応する正解発話文を比べ、主に情報伝達の観点から理解度を評価

表 1-4　理解度ﾗﾝｸ基準

理解度ﾗﾝｸ	評価基準
A	情報伝達、表現ともまったく問題なし
B	情報伝達としてはまったく問題ないが不自然な表現である
C	少し情報が欠けている
D	かなり情報が欠けている
E	正解発話の情報が想像もできない

している。日本人の被験者２名が訂正前後の発話文に対して以下の５段階の理解度評価を行い、そのうち、より厳しい評価者の評価を採用した。

表 1-5 理解度ﾗﾝｸ別の認識結果例

理解度ﾗﾝｸ	認識結果例
B	認識結果：チェックインはだいたい何時ごろうご予定されておりますか 正解発話：チェックインはだいたい何時ごろご予定されておりますか
C	認識結果：一万七千いいのか一万九千のお部屋をご用意できますが 正解発話：一万七千円か一万九千円のお部屋をご用意できますが
D	認識結果：れしくお願いしますしていたします 正解発話：よろしくお願いします失礼いたします
E	認識結果：はいえお司会のいましたらえーで窓呼びいたしましょうか 正解発話：はいお時間になりましたらえー電話でお呼びいたしましょうか

１．４　実験結果および考察

1.4.1　訂正前後の誤り個数の変化

表 1-6 に訂正前後での誤り個数の変化を示す。

表 1-6　訂正前後の誤り個数変化

	挿入	脱落	置換	合計
訂正前	264	206	891	1361
EPC	226(-14.4)	190(-7.8)	853(-4.3)	1269(-6.8)
SSC	251(-4.9)	214(+3.9)	870(-2.4)	1335(-1.9)
EPC+SSC	216(-18.2)	198(-3.9)	831(-7.9)	1245(-8.5)

（１）**EPC+SSC では、8.5％の誤り個数の減少が見られた。**誤り種類別には挿入、置換、脱落の順に減少度合が大きい。

（２）EPC、SSC、それぞれ単独での誤り個数減少率は、EPC が 6.8%、SSC が 1.9%と EPC が多い。

（３）SSC 単独では訂正後に脱落誤りが上昇している。これは、SSC では、下記の例に示すような置換誤り部分を削除した結果、脱落誤りになるケースが多いためである。機械的評価では誤訂正を生起しているように見えるが、誤り部のノイズがなくなるせいで了解性が上がり、また、後段の機械翻訳にとっても処理可能なものになるので、実質的には改善したことになっている。

正解発話：はいありがとうございます京都観光ホテル**予約係で**ございます
認識結果：あはいありがとうございますえ京都観光ホテル**や日間**ございます
訂正結果：あはいありがとうございますえ京都観光ホテルでございます

1.4.2　理解度ﾗﾝｸの変化

訂正前後での理解度ランクの変化結果を表 1-7～8 に示す。また、理解度ランクに変化のあった発話で変化に寄与したと考えられる部分例を表 1-9 に示す。

これらの結果から次のことが分かる。

（１）Ａランクの発話数上昇とＥ、Ｃランクの発話数減少が目立つ。

（２）訂正前後での発話ごとの評価ランクの変化を見ると、評価が上がったものが全体の７％、変化なしが約９２％であった。また、逆に評価が下がったものが約１％（４例）あ

った。

（３）理解度ランクに変化のあった発話で変化に寄与した部分例をみると、**ランクが改善されたものは内容語が回復したものが多かった。**

表 1-7　訂正前後のランク別発話数

ﾗﾝｸ	訂正前	EPC	SSC	EPC+SSC
A	117	126(9)	126(9)	137(20)
C	26	24(-2)	23(-3)	18(-8)
D	89	91(2)	87(-2)	91(2)
E	229	223(-6)	226(-3)	219(-10)

()内は訂正前との差

表 1-8　訂正前後での理解度ﾗﾝｸ変化

	EPC	SSC	EPC+SSC
評価が上がる	18(3.7)	15(3.1)	34(7.0)
同じ	466(96.1)	467(96.3)	447(92.2)
下がる	1(0.2)	3(0.6)	4(0.8)

() 内は評価対象文に対する割合（%）

（４）一方、ランクが悪くなったものは、下記の例のように認識結果はほぼ正しいが、文字３-gram による誤り検出によって誤りありと見なされ（例の下線部）、類似文字訂正によって高頻度で出現する文字列に置換されてしまうものが３例（例１～３）と、誤りパターンによるものが１例（例４）であった。誤りパターン作成時の適格性条件で除外できなかった誤りﾊﾟﾀｰﾝが原因である。

表 1-9　評価ランクに変化があった例

区分	件数	具体例（訂正前　→　訂正後）
挿入誤りの回復	5	きょうからあ二泊お願い → きょうから二泊お願い／かしこまりましたすそうしましたら → かしこまりましたそうしましたら
内容語回復	24	二百七号しの森山 → 二百七号室の森山／返金で → 現金で／いますがい降ろしでしょうか → いますがよろしいでしょうか／な名様 → 何名様／ご約 → ご予約／確に → 確認／五内 → ご案内／そうでお出ます → そうでございます／していたします → 失礼いたします／用具がいします → お伺いします
機能語回復	1	何時ごろうご予定 → 何時ごろをご予定
言い回し改善	10	客様の → お客様の／安くなのでしょうか → 安くなるのでしょうか／お待ちいてます → お待ちしてます／ありがとうございまする → ありがとうございます／それ者のおよろしく → ではよろしく

（例１）
正解発話：お越しをお待ちしております
認識結果：お越<u>しお</u>待ちしております
訂正結果：ではお待ちしております
（例２）
正解発話：はい入っています
認識結果：はい<u>入っ</u>ています
訂正結果：はいすぐ伺います
（例３）
正解発話：はい入りました

認識結果：はい入りました
訂正結果：はい分かりました
（例４）
正解発話：はいなっておりますので
認識結果：はいなっていますので
訂正結果：はになっていますので

「誤りパターン：<に>なって　→　<い>なって」が適用された。この誤りパターンは次のような認識結果から抽出されたものである。
割り増し<い>なっても　→　割り増し<に>なっても
方でお待ち<い>なっていて　→　方でお待ち<に>なっていて

1.4.3　正しい発話文への影響

訂正前後の理解度ランク変化を取り出したものが表 1-10 である。

（１）正しい発話であるＡランクに対しては訂正処理がほとんど実施されていない。
（２）ランクが高いＢ、Ｃランクに対しては、訂正処理が実施されたものの６割以上が評価の上がる方向に訂正が行われている。
（３）一方、ランクが低いＤ、Ｅに対しては、訂正が実施される割合は高いものの、評価が変わる割合は小さい。

表 1-10　訂正前後でのランク別評価ランク変化（EPC+SSC）

評価ランク	A	B	C	D	E	全体
発話数	117	24	26	89	229	485
訂正を実施したもの(%)	1.7	37.5	50.0	48.3	43.2	34.2
評価が上がる(%)	0.0	25.0	34.6	7.9	5.2	7.0
評価が同じ(%)	98.3	66.7	65.4	92.1	94.8	92.2
処理あり(%)	0.0	4.2	15.4	40.4	38.0	26.4
処理なし(%)	98.3	62.5	50.0	51.7	56.8	65.8
評価が下がる(%)	1.7	8.3	0.0	0.0	0.0	0.8

(%)は各ランクに属する総発話に対する割合

以上のことから、提案する EPC、SSC 訂正は**正しい発話に対してはほとんど副作用がなく、比較的理解度ﾗﾝｸが高いもの（情報の欠落が少ない）に対しては特に有効である**ことがわかる。

1.4.4　誤り程度の訂正への影響

誤り程度別に訂正前後の理解度ﾗﾝｸ変化を調べたものが表 1-11 である（誤り程度は、認識結果を対応する正解発話に変換するのに必要な文字単位の編集操作（挿入、削除、置換）回数を用いている）。

理解度ﾗﾝｸが上昇したものは、誤り程度が７個以内の評価発話にほぼ集中しており、この手法が**誤りの多くないものに対して特に有効である**ことを示している。

表 1-11　誤り程度別の理解度ﾗﾝｸ変化（EPC+SSC）

誤り程度	発話数	理解度ﾗﾝｸの変化（%）		
		上昇	同じ	下降
0	102	0.0	98.0	2.0
1	30	16.7	80.0	3.3
2	21	28.6	66.7	4.8
3	26	19.2	80.8	0.0
4	40	12.5	87.5	0.0
5	27	14.8	85.2	0.0
6	24	12.5	87.5	0.0
7	21	9.5	90.5	0.0
8	17	0.0	100.0	0.0
9	20	5.0	95.0	0.0
10	29	0.0	100.0	0.0
11	22	0.0	100.0	0.0
12 以上	106	2.8	97.2	0.0
全体	485	7.0	92.2	0.8

１．５　提案手法の特徴

提案手法には次のような特徴がある。

（１）訂正単位が任意の文字列であるため、単語単位では扱えない訂正が可能である。

例えば、表 1-1 に示す誤り文字列「支払いを方法」にある挿入誤り「を」は、助詞「を」が正しい単語として存在するため、従来手法の単語単位の誤り辞書、あるいは前後の単語の接続可能性による判定等では扱うことができないが、本手法では「を」の前後にある文字列を考慮することで訂正可能となっている。

（２）長い文字列を用いて誤りや表現の傾向を学習するため、文字の連鎖確率だけでは候補の絞り込みが難しい誤りも訂正可能である。

例えば、表 1-1 の誤り文字列「しててきますので」で、誤り文字「て」に置換可能な候補は連鎖確率では「い」、「お」、「頂」の順に高くなるため、正しい文字「頂」を選択するのは難しいが、本手法では「て」の前後にある文字列を考慮することで訂正可能となっている。

（３）この手法で用いる訂正用ﾃﾞｰﾀﾍﾞｰｽは機械的に作成するため、認識装置が更新されても短期間で対応することができる。

参考文献

[1]Y. Wakita et al., 1997. *Correct parts extraction from speech recognition results using semantic distance calculation, and its application to speech translation.* ACL/EACL Workshop Spoken Language Translation, pp. 24-31, 1997-7.

[2]H. Tsukada et al., 1997. *Integration of grammar and statistical language constraints for partial word-sequence recognition.* In Proc. of 5th European Conference on Speech Communication and Technology (EuroSpeech '97), 1997.

[3]Y. Lepage 、1997：String approximate pattern-matching（文字列近似照合)、情報処理学会第 55 回全国大会 6N-1, 1997.

[4]H. Masataki et al., 1996. *Variable-order n-gram generation by word-class splitting and consecutive word grouping*. In Proc. of ICASSP, 1996.

[5]T. Shimizu et al., 1996. *Spontaneous Dialogue Speech Recognition using Cross-word Context Constrained Word Graphs*. ICASSP '96, pp. 145-148, 1996.

[6]T. Morimoto et al., 1994: *A Speech and language database for speech translation research.* Proc. of ICSLP '94, pp. 1791-1794, 1994.

[7] 脇田等、1997：単語 bi-gram を用いた連続音声認識への状態系列の誤認識特性の利用. 日本音響学会平成 9 年度春季研究発表会講演論文集

第２章　訂正手法の改良

誤り訂正手法、EPC と SSC は初期バージョンの発表以後、表 2-1、2 に示す改良を重ねてきた。本章では改良経過に沿ってその改良内容、評価結果などをまとめた。

表 2-1　ＳＳＣのバージョン

略号	改良内容
Ｖ０	初期バージョン
Ｖ１	抽象化、差分置換方法の改良
Ｖ２	高頻度文字列の利用
Ｖ３	形態素対応
Ｖ４	高速化

表 2-2　ＥＰＣのバージョン

略号	改良内容
Ｖ０	初期バージョン
Ｖ１	形態素対応、包含条件の変更

２．１　SSC 改良（Ｖ１バージョン）

SSC の初期バージョンに対して次の３つの改良を行った。

2.1.1　改良内容

（１）差分置換アルゴリズム改良

類似文字列と誤り文字列の差分置換方法を変更した。初期バージョンでは、共通文字列の位置と文字数が誤り文字列において固定であったが、この改良では、２文字以上であればその位置は自由とした。その結果、図 2-1 に示すように、これまで訂正が成立しなかっ

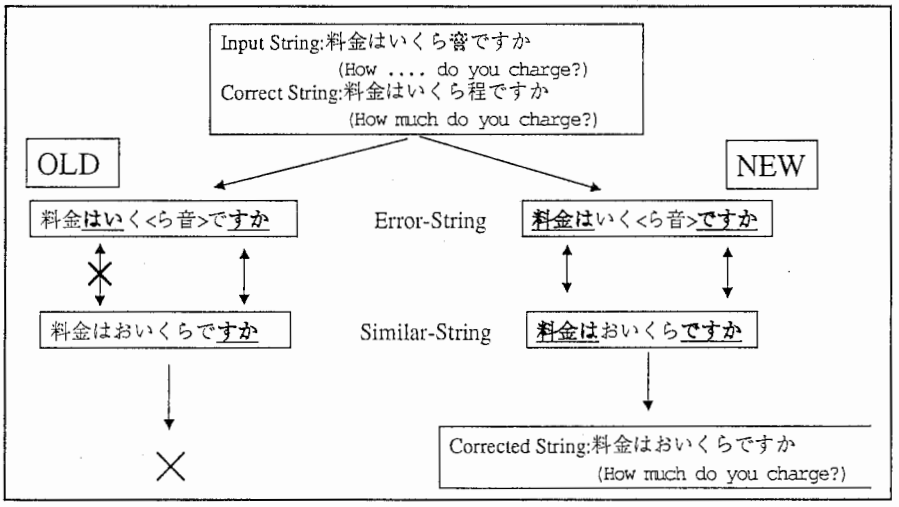

図 2-1　差分置換アルゴリズムの改良

たものがかなり救われた。

（２）間投詞除去

これは、「ああ」とか「えー」などの自然な発話ではどこでも出現する間投詞を除くことで、ノイズを減らし、訂正の精度を高めようというものである。

（3）抽象化

日付、曜日、固有名詞、数字を記号で置き換えて類似文字列検索などを行うようにした。この改良によって、図 2-2 に示す例のように、番号と名前が異なる為にこれまでは訂正が成立しないケースに対応できるようになった。

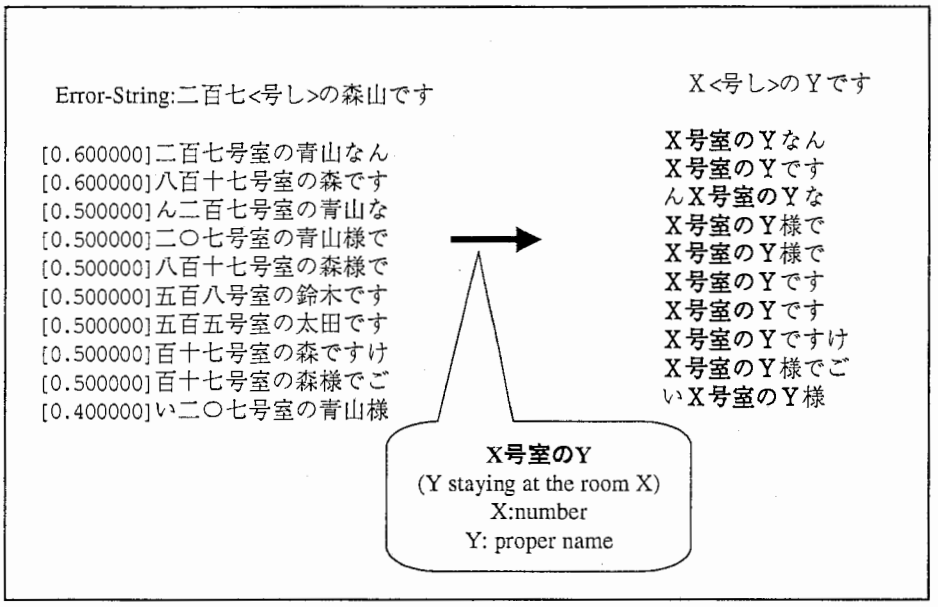

図 2-2　抽象化

2.1.2　ＳＳＣ（Ｖ１）評価実験データ

実験に用いたﾃﾞｰﾀは、初期バージョン（Ｖ０）の評価に使用したものとほぼ同じである。

評価は訂正前後での誤り個数の変化を用いて行った。また、ｺｰﾊﾟｽｻｲｽﾞを変化さして訂正に与える影響を調べる実験も行った。下記に用いたﾃﾞｰﾀの概要を示す。

（1）音声認識結果

音声認識結果データを表 2-3 に示す。表中で、既存というのは初期バージョンと同じく間投詞を含むものである（1．3節参照)。

表 2-3　音声認識結果

（全認識結果４８０６発話）

	発話数	認識率(%)（文字単位）	誤り数			
			挿入	脱落	置換	合計
既存	4806	74.73	2642	1702	8087	12431
間投詞除去	4806	75.23	1792	1558	7592	10942

（訂正評価に用いた４８５発話）

	発話数	認識率(%)（文字単位）	誤り数			
			挿入	脱落	置換	合計
既存	485	72.4	264	206	891	1361
間投詞除去	485	72.7	187	187	842	1216

（2）コーパス

用いたコーパスを表 2-3-1 に示す。コーパスの種類は次のとおりである。

既存：初期バージョンの評価に用いたものと同じもの
間投詞除去：「既存」から間投詞を除去したもの
抽象化：間投詞除去後、抽象化したもの
増量：間投詞を除去したもので、発話数が「既存」の約２倍のｺｰﾊﾟｽ
減量：間投詞を除去したもので、発話数が「既存」の約半分のｺｰﾊﾟｽ
ｸﾛｰｽﾞ(485)：評価した認識結果（４８５発話）の正解発話で構成されるｺｰﾊﾟｽ

表 2-3-1　コーパス

ｺｰﾊﾟｽ種類	タスク数	発話数	延べ文字数	異なり文字数
既存	894	20176	570306	1396
間投詞除去	894	20199	554689	1395
抽象化	894	20199	531732	1271
増量	1643	40895	1026008	1554
減量	451	9609	259061	1044
ｸﾛｰｽﾞ(485)	44	485	12018	514

2.1.3　誤り検出精度

各種ｺｰﾊﾟｽを用いて作成した文字 3-NGRAM による誤り検出精度を求めた。

表 2-4　誤り検出精度

	誤り個数	予測個所数	予測が正しい数	見逃し数	適合率	再現率
既存	12431	11109	9362	3495	84.27	71.89
間投詞除去	10942	10367	8726	2618	84.17	76.07
抽象化	10942	10053	8610	2745	85.65	74.91
増量	10942	10130	8648	2870	85.37	73.77
減量	10942	10751	8654	2270	80.50	79.25

注）４８０６発話（閾値は全て-3.7）

（1）適合率はほとんど変わらないが、間投詞除去によって再現率が上昇している。恐らく、発話のどこにでも入る間投詞がなくなったため、文字連鎖種類が制限されるためと思われる。

（2）抽象化しても検出精度にあまり変化はなかった。数字に関係する部分が若干良くなる程度である（付録２参照）。

（3）コーパスを二倍にしても、改善は見られなかった。「既存」ｺｰﾊﾟｽの発話でほとんどの表現が出尽くしていたためと思われる。

2.1.4 SSC 訂正精度

（1）間投詞の影響

表 2-5 に間投詞を除去した場合の訂正精度を示す。

表 2-5 間投詞を除去しての訂正精度

	SSC		EPC	SSC+EPC	
	V0	V1	V0	V0	V1
間投詞あり	-1.91	-3.31	-6.76	-8.45	-9.63
間投詞除去	-2.88	-4.44	-2.80	-5.02	-6.25
変化率（%）	50.79	34.14	-58.58	-40.59	-35.10

注）ここで、SSC の閾値は 0.6、EPC の頻度制限は 2 である。

訂正全体としては精度が落ちている。原因は間投詞除去によって除去前に EPC で訂正できていた誤りが減ったことで EPC の訂正率が下がったためである。

しかし、SSC の訂正率は上昇している。間投詞による雑音がなくなるため、訂正がより有効に働くようになった為と思われる。

（2）文字列ﾃﾞｰﾀﾍﾞｰｽの比較

文字列ﾃﾞｰﾀﾍﾞｰｽのデータ量による訂正精度の比較結果を表 2-6 に示す。

1）文字列ﾃﾞｰﾀﾍﾞｰｽ作成の際の出現頻度制限を 3 から 1 に変えることによる精度向上がもっとも高い。しかし、ﾃﾞｰﾀﾍﾞｰｽの件数比で比較すると、ｺｰﾊﾟｽを二倍にする方が精度向上効率は高い。

2）ｸﾛｰｽﾞにした場合、訂正精度は大幅にアップする。これは、ｺｰﾊﾟｽに訂正対象発話と良く似たものをもってくれば、まだまだ訂正精度を高めることができることを示している。

3）抽象化は、訂正精度を下げている。ただ、訂正条件のパラメータを変化さすことで同じかそれ以上になる。

表 2-6 文字列ﾃﾞｰﾀﾍﾞｰｽの比較

ｺｰﾊﾟｽ種類	文字列出現頻度の制限値	件数		V0		V1	
		件数	比率	精度	増減率	精度	増減率
間投詞除去	3	17318	1.0	-2.88	0.00	-4.44	0.00
間投詞除去	1	337185	19.5	-3.29	14.24	-5.10	14.86
増量	3	32032	1.8	-3.21	11.46	-4.77	7.43
減量	3	9452	0.6	-2.80	-2.78		
抽象化	3	16625	1.0	-2.63	-8.68	-4.19	-5.63
抽象化	1	330686	19.1	-3.13	8.68	-5.43	22.30
ｸﾛｰｽﾞ	1	9254	0.5	-14.88	416.67		

（3）文字列ﾃﾞｰﾀﾍﾞｰｽと誤り位置検出の効果

表 2-7 に文字列ﾃﾞｰﾀﾍﾞｰｽおよび誤り位置をｸﾛｰｽﾞの条件で行った訂正精度を示す。

文字列ﾃﾞｰﾀﾍﾞｰｽの効果が誤り検出に比べ大きいことがわかる。

表 2-7 文字列ﾃﾞｰﾀﾍﾞｰｽと誤り位置の効果

文字列 DB	誤り位置検出	V0	V1＊
既存 3	真	-6.00	-7.81
ｸﾛｰｽﾞ	既存	-14.88	-20.81
ｸﾛｰｽﾞ	真	-26.23	-35.77

ここで、V1＊は V1 への改良途中のモデルである。両者は、ほぼ同じアルゴリズムで、性能は V1 の方が高い。

（4）SSC のバージョン比較

SSC 各バージョンの訂正精度を表 2-8 に示す。

表 2-8　バージョン比較

ｺｰﾊﾟｽ	Ｖ0	Ｖ1＊		Ｖ1	
	基準	精度	率(%)	精度	率(%)
間投詞あり	-1.91	-3.31	73.30		
間投詞なし	-2.88	-4.44	54.17	-4.44	54.17
抽象	-2.63	-4.11	56.27	-4.19	59.32
ｺｰﾊﾟｽ倍増	-3.21	-4.77	48.60	-4.77	48.60
誤り位置真	-6.00	-7.81	30.17		
ｸﾛｰｽﾞ	-14.88	-20.81	39.85		
ｸﾛｰｽﾞ位置真	-26.23	-35.77	36.37		

1)Ｖ１＊で平均４８%、Ｖ１で５４%の改善が見られた。

2)類似度制限のパラメータを変化さすと、Ｖ１では表 2-9 のような結果になる。

（5）抽象化の効果について

1)表 2-9 を見る限り、あまり有効であるとは思えない。

2)付録 3 に間投詞除去と抽象化の差分を掲載している。二つの差分は非常に異なり、正確な評価は**人間による評価判断が必要である。**

3)抽象化することで、精度が落ちる場合があるが、それは誤り検出の違いや、類似検索文字列の類似度の相違などが原因である。抽象化の長所には、人名などを過剰訂正しなくなるなどがある。

表 2-9　類似制限の変化

ｺｰﾊﾟｽ	閾値	Ｖ1
間投詞除去	0.6	-4.44
	0.5	-6.17
	0.4	-6.83
	0.3	-5.51
抽象	0.6	-4.19
	0.5	-6.41
	0.4	-6.83
	0.3	-6.00

（6）最高精度

ｺｰﾊﾟｽが「既存」しかない条件でもっともいい条件を表 2-10 にまとめた。モデルはＶ１である。

表 2-10　最高精度（Ｖ１）

閾値	間投詞除去		抽象	
	3	1	3	1
0.6	-4.44	-5.10	-4.19	-5.43
0.5	-6.17	-7.15	-6.41	-7.40
0.4	-6.83	-6.74	-6.83	-6.66
0.3	-5.51	-6.91	-6.00	

2.1.5　ＥＰＣ訂正精度

1)間投詞除去によって全体に誤りが減少することで、抽出されたﾊﾟﾀｰﾝ数も減少し、訂正精度も低くなる。

2)抽出条件頻度を低めることで、訂正精度はある程度改善できた。

表 2-11　EPC 精度

		挿入	置換	削除	計	EPC
間投詞あり	頻度 2	281	263	85	629	-6.76
間投詞除去	頻度 2	87	202	72	361	-2.80
	頻度 1	1259	5830	669	5830	-4.11

2．2　ＳＳＣ改良（Ｖ２）　「高頻度文字列の利用」

SSC（V1ﾊﾞｰｼﾞｮﾝ）に次のような改良を加えた。

（１）類似文字列検索において文字列長さを可変にする。

これまでは、類似文字列検索において文字列ﾃﾞｰﾀﾍﾞｰｽや誤り文字列は一定長さの文字列を利用して行っていたのを、誤り部を中心に最も良く類似する部分をｺｰﾊﾟｽ発話から任意の長さで検索するようにした。

（２）共通文字列に高頻度文字列を用いる。

差分置換の目印となる共通文字列をコーパス中に高頻度で出現する部分文字列とした。類似文字列検索では、共通文字列を含む発話をｺｰﾊﾟｽから検索する。

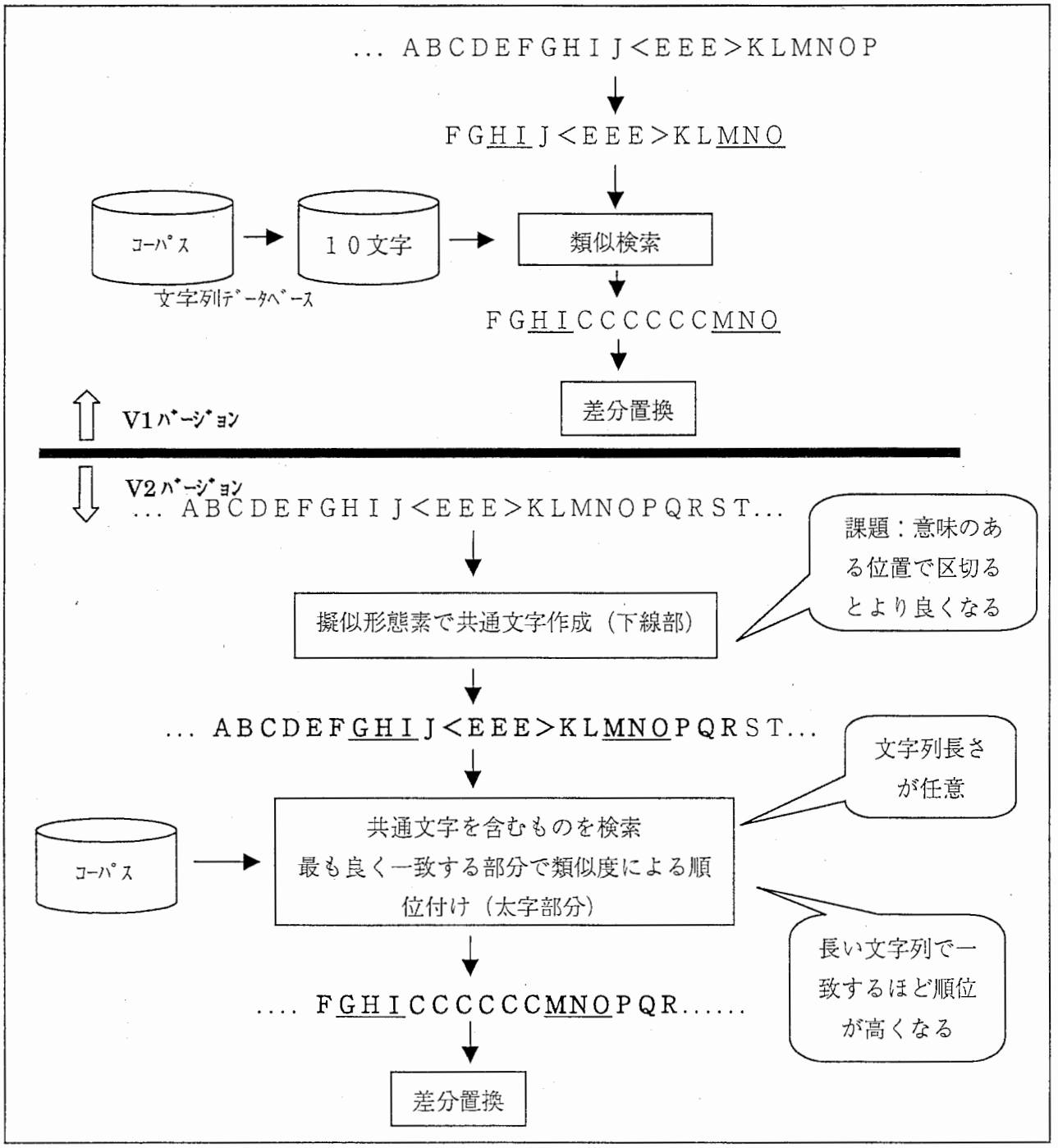

図 2-3　V1ﾊﾞｰｼﾞｮﾝと V2ﾊﾞｰｼﾞｮﾝの比較

2.2.1 実験条件

実験条件はこれまでと同じ認識結果、ｺｰﾊﾟｽを用いた（2.1.2 節参照）。

2.2.2 訂正精度

表 2-12 に訂正精度を示す。

Ｖ１での最大値（-7.40%）を超える訂正精度（-8.63%）が得られた。

Ｖ１と異なり、脱落が訂正前より増えるというようなことはなく、誤り種類に関わりなく減少している。これは、Ｖ２がより長いスパンで類似するものを検索することによるものと思われる。

付録４にＶ１との差異例を載せる。

表 2-12　SSC（V2）訂正精度（４８５発話）

	文字 DB	照合	擬似形態素	閾値	挿入	置換	脱落	全体
Ｖ１	既存３＊	＊		0.6	-10.70	-4.04	1.60	-4.19
				0.5	-13.37	-6.41	0.53	-6.41
				0.4	-16.04	-6.65	1.60	-6.83
				0.3	-13.90	-6.41	3.74	-6.00
	既存１＊			0.6	-18.18	-5.23	6.42	-5.43
				0.5	-21.39	-7.13	5.35	-7.40
				0.4	-22.99	-7.13	11.76	-6.66
Ｖ２	既存Ｒ	＋	1000	0.8	-9.63	-4.75	-11.76	-6.58
				0.7	-10.16	-5.58	-15.51	-7.81
				0.6	-6.95	-5.82	-22.99	-8.63
				0.5	-8.02	-3.44	-22.99	-7.15
			10000	0.7	-10.16	-5.23	-12.83	-7.15
				0.6	-3.21	-5.70	-16.04	-6.91
		＊	1000	0.8	-9.63	-4.87	-10.70	-6.50
				0.7	-10.70	-5.82	-13.90	-7.81
				0.6	-7.49	-6.06	-21.39	-8.63
				0.5	-8.56	-3.92	-20.32	-7.15
			10000	0.7	-10.16	-5.58	-11.76	-7.24

2．3　形態素単位（ＳＳＣ：Ｖ３、ＥＰＣ：Ｖ１）

これまで述べてきた訂正手法を認識と翻訳の間で用いるためには、訂正結果にも形態素情報が必要である。そこで、訂正が形態素単位で実行できるように改善を行った。

改善の主な点は次のとおりである。

2.3.1　誤りﾊﾟﾀｰﾝの形態素対応（Ｖ１）

（１）誤りﾊﾟﾀｰﾝ抽出時に形態素単位で抽出を行う。また、誤りﾊﾟﾀｰﾝに形態素番号の情報も付加する。

具体的には図 2-4 に示すように、誤りﾊﾟﾀｰﾝ候補の生成段階で、取り出される部分文字列が形態素で区切られるようにして取り出している。

（２）誤りﾊﾟﾀｰﾝ候補の選択条件のうち、包含条件を次のように変更する。

文字単位：
ＡＢＣＤＥＦ＜ＥＥＥ＞ＬＭＮＯＰＱＲ
Ｆ＜ＥＥＥ＞
ＥＦ＜ＥＥＥ＞
ＤＥＦ＜ＥＥＥ＞
形態素単位：
／ＡＢＣＤ／ＥＦ＜ＥＥ／Ｅ＞Ｌ／ＭＮＯ／ＰＱＲ／
／ＥＦ＜ＥＥ／Ｅ＞Ｌ／
／ＡＢＣＤ／ＥＦ＜ＥＥ／Ｅ＞Ｌ／

図 2-4　形態素単位での誤りﾊﾟﾀｰﾝ生成

改善前：

頻度が同じで包含関係にある候補は、大きい方を採用する。

頻度が異なり包含関係にある候補は、小さい方を採用する。

改善後：

包含関係にある候補は、小さい方を採用する。

「お客」を「用件」と誤認識した結果から誤りﾊﾟﾀｰﾝを抽出した場合、これまでは、例１のような結果が抽出された。Ｖ１バージョンでは例２に示すようなﾊﾟﾀｰﾝが抽出される。

例１）

まりました<お客>様カードの:まりました<用件>様だカード
まりました<お客>様のお名前:まりました<用件>様のお名前

誤りﾊﾟﾀｰﾝ：まりました<お客>様:まりました<用件>様

例２）

かしこまりました/<お/客>/様/の/お/名前/は:かしこまりました/<用件>/様/の/お/名前/は
かしこまりました/<お/客>/様/カード/の/方/に:かしこまりました/<用件>/様/だ/カード/の/方

誤りﾊﾟﾀｰﾝ：かしこまりました/お/客:かしこまりました/用件
お/客/様:用件/様

図 2-5　誤りﾊﾟﾀｰﾝ抽出時の差異

2.3.2 ＳＳＣの形態素対応（Ｖ３）

差分置換条件をエラーブロック前後の共通文字列が形態素単位で行うようにした。具体的な違いは次のとおりである。

認識結果=かまいません忘れは寝いたします
正解発話=構いません和室でお願いいたします

【文字単位】

誤り文字列：「かまいませ<ん忘れは寝>いたします」
類似文字列：
候補１：「いませ<んそれでは失礼>いたします」
候補２：「かまいませ<んお願い>いたします」

文字列の一致でしかみないので、候補１から次のように訂正する。

訂正文字列=かまいませんそれでは失礼いたします

【形態素単位】

差分置換の位置決めに形態素単位でうまく区切れる候補を探す。この例では、候補１はうまく形態素単位で区切れない。そこで、うまく区切れる候補２を利用して訂正を行う。

認識結果=かま/い/ま/せ/<ん/忘れ/は/寝>/いた/し/ま/す
候補１=とんでもござ「いませ<ん/それでは/失礼>いたします」
候補２=はい/それ/で/「かま/い/ま/せ/ん/お/願/い/いた/し/ま/す」

訂正文字列=かまいませんお願いいたします

2.3.3 訂正精度

EPC は格段に良くなった。原因は選択条件の変更にある。文字単位でも同様の条件をつけるべきであろう。

SSC は若干良くなっているが、人間の評価が必要である。文字単位との差分を付録５に掲載した。

表 2-13 訂正精度（ＳＳＣ：Ｖ３、ＥＰＣ：Ｖ１）（４８５発話）

処理単位	モデル	訂正精度	備考
文字単位	SSC(V2)	-7.81	擬似形態素 1000、類似度 0.7
	EPC(V0)	-2.80	頻度 2
形態素単位	SSC(V3)	-8.39	擬似形態素 1000、類似度 0.7
	EPC(V1)	-5.43	頻度 2
	EPC(V1)+SSC(V3)	-11.68	

注）ここでの条件は、間投詞なし、抽象化を行ったものである。

2．4　高速化

SSC の類似文字列検索処理は処理速度が遅く大規模なコーパスでは実用に耐えないため高速化を検討した。高速化する方法として、図 2-6 に示すように誤り文字列に含まれる前方キー（PRE）と後方キー（POST）を含むコーパス文を高速に取り出せるようにインデックス化した。前方キー、後方キーはコーパスに出現する高頻度文字列を用いた。また、インデックスによって検索された多数の候補をさらに絞り込むことを検討した。それら検討結果を表 2-14 に示すとおりである。最終的に「ｲﾝﾃﾞｯｸｽ化」と「DP-match ﾃﾞｰﾀの絞り込み 2」の方法を採用することにした。

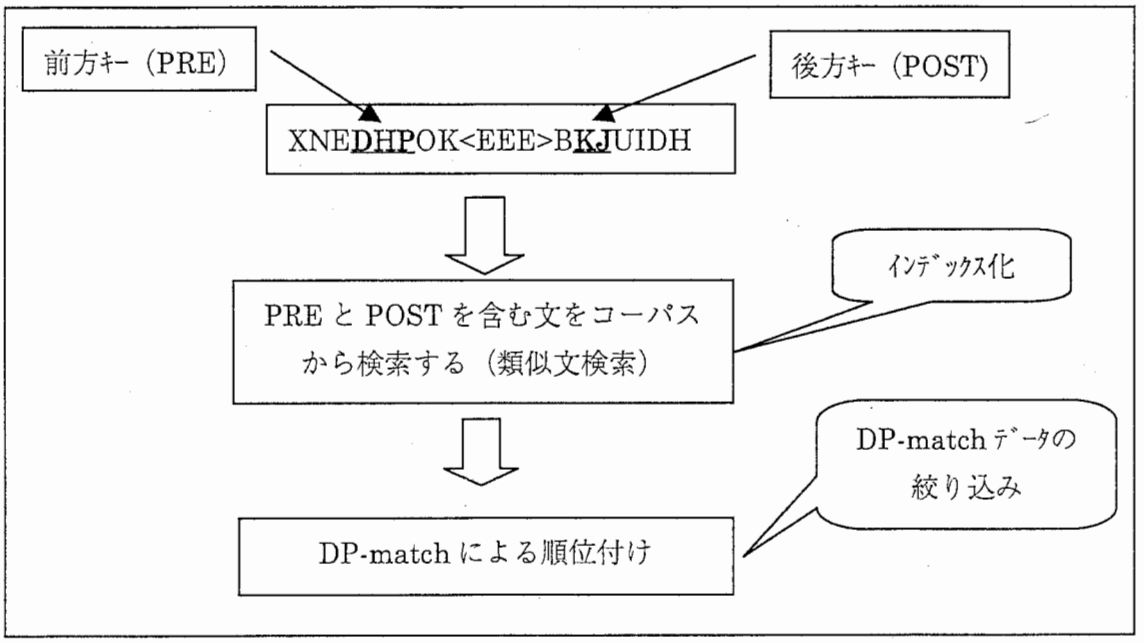

図 2-6　高速化位置

表 2-14　高速化検討結果

項目	概要	効果	課題
ｲﾝﾃﾞｯｸｽ化	コーパス文について高頻度文字列（ｎ以上）でｲﾝﾃﾞｯｸｽを作成し、類似文検索のキー文字列でｲﾝﾃﾞｯｸｽを参照してコーパス文の検索範囲を絞りこむ。	あり ｲﾝﾃﾞｯｸｽなし　1 ｲﾝﾃﾞｯｸｽ利用　4～42≒25	キー文字列がｲﾝﾃﾞｯｸｽにない場合、全文検索になる。
DP-match ｲﾝﾀｰﾌｪｰｽ改良	DP-match の高速化を目的に、パイプで接続していた部分を Perl 組み込み関数に変更。SWIG パッケージを用いて作成	ほとんど、効果なし	
DP-match ﾃﾞｰﾀの絞りこみ 1	DP-match の件数を減らすため、誤り文字列に文頭、文末情報を付加して絞りこみを行う。 （ｲﾝﾃﾞｯｸｽに PRE-POST が共にある場合以外に適用）	あり	複文の場合に正解ﾃﾞｰﾀを見つけられない可能性あり
DP-match ﾃﾞｰﾀの絞りこみ 2	DP-match の件数を減らすため、誤り文字列と類似文検索された文でﾊﾟﾀｰﾝﾏｯﾁする部分文字列の長さを比較して、その文字列長さの差が±ｎ文字以内の類似文のみ採用 （ｲﾝﾃﾞｯｸｽに PRE-POST が共にある場合以外に適用）	あり ｲﾝﾃﾞｯｸｽ利用　1 本手法　　1.2～1.6	正解データを見つけられない可能性あり
DP-match ﾃﾞｰﾀの絞りこみ 3	類似文検索と DP-match の件数の絞りこみに Lepage さん作成の Agrep を利用。 ｲﾝﾃﾞｯｸｽに PRE-POST が共にある場合以外に適用）	なし Agrep のための前処理に時間がかかり、絞り込み効果が相殺される。	

第３章　語順並び替えによるコーパスの拡張

３．１　目的

SSC の訂正性能は誤り文字列と類似する文字列がｺｰﾊﾟｽに存在するかどうかに強く依存している。数字、固有名詞などの抽象化によってある程度は類似条件の許容度を広げることができた。しかし、内容的には同じ文でも語順が異なるため類似検索にひっかからないという現象もみられた。そこで、日本語における特徴、「特定の語については、語順を入れ替えても文の内容が変わらない」に注目して、ｺｰﾊﾟｽ発話の語順並び替えによる誤り検出・訂正精度に対する効果の検討を行った。

３．２　並び替えアルゴリズム

語順の並び替えが可能なものには副詞、後置詞句などがあるが、今回は後置詞句に注目した。具体的には構文解析木付きの発話（JTREE）から後置詞句を取り出し、並び替えを行った。

例えば「９月２０日に家族でラスベガスへの旅行を計画しているんですが」の場合、「９月２０日に」、「家族で」、「ラスベガスへの旅行を」の３つの後置詞句があり、これらの語順を任意に並び替えて次の６発話を生成する。

「９月２０日に家族でラスベガスへの旅行を計画しているんですが」
「９月２０日にラスベガスへの旅行を家族で計画しているんですが」
「家族で９月２０日にラスベガスへの旅行を計画しているんですが」
「家族でラスベガスへの旅行を９月２０日に計画しているんですが」
「ラスベガスへの旅行を９月２０日に家族で計画しているんですが」
「ラスベガスへの旅行を家族で９月２０日に計画しているんですが」

３．３　評価実験

3.3.1　ｺｰﾊﾟｽについて

ＳＬＤＢの JTREE を利用して並び替えを行った。表 3-1 にその結果を示す。

並び替えが可能であった発話は４０５７発話であり、全体の約２割を占めていた。また、並び替えによって発話数は約二倍に増加した。

表 3-1　コーパス

	タスク数	発話数	延べ文字数	異なり文字数
並替え前	611	21211	408572	1296
並替え後	611	40917	1260814	1296
並替え前・抽象化	611	21211	387067	1218
並替え後・抽象化	611	40917	1198162	1218
間投詞除去	894	20199	554689	1395

注）最終行はこれまでの実験で用いたｺｰﾊﾟｽ（2.1.2 節参照）

3.3.2 誤り検出精度

認識結果４８０６発話に対して誤り検出精度の測定を行った。結果は表 3-2 のとおりである。並び替え前後で検出精度にほとんど差は見られなかった。ただ、これまでのｺｰﾊﾟｽ 2.1.2 節参照）と比較すると適合率が約１０ポイント近く悪くなっている。

表 3-2 誤り検出精度

	適合率	再現率
並替え前	74.18	78.58
並替え後	74.05	78.25
並替え前・抽象化	77.53	77.49
並替え前・抽象化	77.27	77.20
間投詞除去	84.17	76.07
抽象化	85.65	74.91
増量	85.37	73.77
減量	80.50	79.25

注）４８０６発話（閾値は全て-3.7）

3.3.3 訂正精度

訂正精度の実験を SSC のバージョンＶ１モデルを用いて行った。文字列ﾃﾞｰﾀﾍﾞｰｽに採用した文字列は、頻度閾値を１以上で抽出したものを用いている。結果は表 3-3 のとおりである。モデルのパラメータによって精度のばらつきはあるが、同一条件では並び替え後の方が精度が上がっている。

そこで、抽象化を行い類似条件閾値０．５の結果について並び替え前後で訂正結果に差が出た発話について人間による評価を行った。評価は、「並び替えで良くなった」、「どちらともいえない」、「悪くなった」発話数をそれぞれカウントした。結果を表 3-4 に示す。

表 3-3 訂正精度

文字 DB		閾値	挿入	置換	脱落	全体
並替え前	抽象化なし	0.5	-8.02	2.14	8.56	1.56
	抽象化あり	0.5	-10.70	0.83	6.95	0.00
		0.6	-6.95	-0.71	2.67	-1.15
並替え後	抽象化なし	0.5	-9.09	0.36	9.63	0.33
	抽象化あり	0.5	-13.37	-0.24	6.42	-1.23
		0.6	-6.42	-1.19	3.74	-1.23

注）テストブロック１（４８５発話）

表 3-4 差分評価結果

種類	発話数
並び替えで良くなった	17
どちらともいえない	35
悪くなった	6
合計	58

良くなった発話数が悪くなった発話数を上回っている。具体例を付録６にまとめてある。

第４章　新認識結果への適用

新しい認識プログラムが出力した認識結果を用いて、訂正精度実験を行った。比較のため旧認識結果（１～３章まで用いた認識結果）への適用値も併記する。

４．１　実験データ

新認識結果は「TR-IT-0265」（音声翻訳ｼｽﾃﾑのための日本語音声認識言語モデル）における研究用言語モデルの結果を用いた。また、コーパスも認識と同じ学習セットを用いた。

（１）認識結果

新認識結果の概要を表 4-1 に示す。これらの値は間投詞を除いて算出したものである。

旧認識結果と比較すると、認識率で１０ポイント近く良くなっている。一発話あたりの誤り個数は、旧認識結果で 2.3 個、新認識結果では 1.3 個であり、新認識結果はかなり誤り個数も少なくなっている。

表 4-1　認識結果の概要

	新認識	旧認識（評価文）	旧認識（全文）
発話数	881	485	4806
文字認識率	81.20	72.75	75.23
単語認識率	76.49	60.13	63.73
挿入誤り個数	140	187	1792
削除誤り個数	224	187	1558
置換誤り個数	775	842	7592
全誤り個数	1139	1216	10942

（２）コーパス及び誤り検出精度

コーパスは間投詞を除き、抽象化を行って使用した。コーパスの概要を表 4-2 に示す。また、誤り検出精度も合わせて掲載した。

誤り検出は、種類が「新」の場合は新しいコーパスで新しい認識結果全文について行ったものである。「旧」の場合は間投詞抜きの「既存コーパス」を用いて旧認識結果全文に対して行った結果である。

誤り検出精度は、「旧」と比較すると、適合率で１０ポイント、再現率で３０ポイント近く悪くなっている。原因は新認識結果の誤りが、表記のゆれによる置換誤りや、誤り文でも日本語として正しい表現になる場合が多いためと考えられる。ちなみに表記のゆれによる誤りは約１０％であった。

表 4-2　コーパス及び誤り検出精度

種類	タスク数	発話数	延文字数	異なり文字数	適合率	再現率	備考
新	2473	28603	753214	1326	72.787	45.215	3-gram
					85.106	19.140	2-gram
旧	894	20199	531732	1271	85.646	74.913	3-gram

4．2　訂正精度

表 4-3 に訂正精度を示す。モデルは SSC（V3）、EPC（V1）を用いている。EPC については、認識タスクファイルごとに、残り６４タスクファイルの認識結果から抽出した誤りﾊﾟﾀｰﾝを用いた。

表 4-3　訂正精度

新旧	SSC 条件	誤り検出 n-gram	EPC 条件	文字認識率	単語認識率	挿入	削除	置換	全体
新	1000,0.7	3-gram	----	80.21	75.27	9.29	10.27	0.77	3.69
	1000,0.7	2-gram	----	80.99	76.30	0.71	2.23	-0.65	0.09
	1000,0.7	true-position	----	80.74	75.88	5.00	-3.57	-6.58	-4.57
	-----		2	81.52	76.81	-19.29	0.45	-0.13	-2.37
旧	1000,0.7	3-gram	----	73.32	62.41	-12.30	-13.37	-6.41	-8.39
	----		2	73.80	62.25	-14.44	-11.76	-2.02	-5.43
	1000,0.7	3-gram	2	74.12	64.09	-21.93	-18.72	-11.68	-11.68

旧結果と比較すると極端に悪くなっている。その原因として次のことが考えられる。

（1）EPC

抽出した誤りﾊﾟﾀｰﾝの数が少ない。使用した発話あたりで抽出した誤りﾊﾟﾀｰﾝ数を比較すると、新認識結果の方が三分の一程度になっている（表 4-4 参照)。

表 4-4　誤りﾊﾟﾀｰﾝの違い

新旧	使用した発話数	誤りﾊﾟﾀｰﾝ件数	発話あたりの誤りﾊﾟﾀｰﾝ件数
旧	4321	398	0.092
新	~867	~25	0.028

（2）SSC

誤り検出精度が悪いことが挙げられる。表 4-3 に示すように誤り検出の適合率が良い 2-GRAM、あるいは真の誤り位置で行うと訂正精度が上がることからそのことがわかる。

4．3　新旧の認識結果で共通する発話を取り出しての比較

SSC の訂正精度が悪い原因を詳細に調べるため、新旧の認識結果で共通する発話に注目した。共通する発話は４８６発話ありそれらの特徴を表 4-5 に示す。

認識率、誤り個数とも旧認識結果の方が悪い。ただ、削除誤りの数は新認識結果の方が多い。

表 4-5　新旧で共通する発話の特徴

	発話数	正解発話数	文正解率	認識率（文字）	認識率単語	挿入	削除	置換	計
新	486	192	39.51	80.18	75.52	85	120	386	591
旧	486	167	34.36	75.47	65.27	122	99	521	742

（1）テキストを眺めて

表 4-6 に正解発話の分布を示す。その内、旧認識結果が正解で新認識結果が誤りを含む場合を表 4-7 に示す。

これを見ると、認識装置の定型的な誤りの「ですです」、あるいは「そうですね」のように、文字単位では正確に認識しているのだが、形態素としては正解と異なるために誤りとなったものが多かった。

また、助詞「を」が抜けているのだが、日本語としては問題ないものもある。

表 4-6　新旧の正解発話の分布

旧	新	発話数
○	×	55
×	○	80
○	○	112
×	×	239

○は正解であったもの、×は誤りを表す

表 4-7　旧認識結果が正解で新認識結果が誤りのケース

頻度	内容	具体例
19	定型誤り	ANS=はいそうです OLD=はいそうです NEW=はいそうですです
8	TDMT 体系の形態素の扱いの違い	ANS=そうですね OLD=そうですね NEW= WORDS=/あっ/そう/です/ね/end wordids=/20006/10057/10021/10056/end WORDS=/あ/そうですね/end wordids=/20001/20060/end
3	助詞の欠落	ANS=和食をお願いします OLD=和食をお願いします NEW=和食お願いします
4	表記のゆれ	
2	助詞の置換	
19	明らかな誤り	ANS=いいよ OLD=いいよ NEW=イーオー OLD=よろしくお願いします NEW=義美をします OLD=十月の二十七日と二十八日の二泊なんですけれど NEW=十月の二十七日と二八日の二泊なんですけど OLD=はいよろしくお願いします NEW=よろしくねします OLD=はいそうです NEW=はい都です

表 4-8 に新旧の認識結果ともに誤っている例を示す。新認識結果は文単体でみれば正しい日本語になっているものが多いが、旧結果は明らかに日本語からはずれた表現になっている。

表 4-8　誤り文の新旧比較
(ANS が正解、OLD が旧認識結果、NEW が新認識結果である)

ANS=アメリカにファックスを送りたいんですがどうしたら良いでしょうか
OLD=雨に対にファックス送りたいんですがどうしたらワイでしょうか
NEW=アメリカにファックス送りたいんですがどうしたらいいでしょうか

ANS=もしもし九月十四日の夜そちらに泊まりたいんですが
OLD=もしもしいってなと十四日の夜そちらに泊まりたいんですが
NEW=もしもし九月十四日にそちらに泊まりたいんですが

ANS=はい六時頃そちらに着く予定なんですけれども
OLD=はい六時ごろ市をツイン着く予定なんですけれども
NEW=はい六時ごろそちらに着く予定なんですけれども

ANS=ツインだといくらぐらいからの料金ですか
OLD=ツインだとおうねいいくらぐらいからんので空きんですか
NEW=ツインだといくらぐらいからご用件ですか

ANS=二〇一号室の西川と申します
OLD=二二〇一号室の西カードもします
NEW=二〇一号室の石川と申します

(2) SSC 精度の違い

表 4-9 に SSC 訂正精度を示す。比較として、誤り位置、ｺｰﾊﾟｽそれぞれが真である場合も掲載している。

どのケースについても「旧」が「新」を上回っている。

表 4-9　SSC 訂正精度

位置	ｺｰﾊﾟｽ	旧	新
ー	ー	-2.02	3.55
○	ー	-8.49	-3.21
ー	○	-26.15	-16.24
○	○	-46.50	-40.78

ここで、○は真の値を使用したことを表す。

４．４　人間による訂正

新旧認識結果それぞれ３００発話程度を取り出し人間による訂正可能性の評価を行った。この評価は、まず発話単体を眺めて日本語として意味をなすか否かを判断する（表中では「日本語としてＯＫ」「ＮＯ」）。さらに意味をなすものについては、正解発話と比べて意図が同じかどうかを判断した。一方、「日本語としてＮＯ」と判定されたものは発話そのものから元の発話を推定できるか否かを評価し（表中では「推定ＯＫ」「ＮＯ」）、推定できるものは推定発話を正解発話と比較した。結果を表 4-10、11 に示す。

新旧の大きな違いは日本語として意味をなすものの割合である。「新」は 46%、「旧」は 17%であり、新認識結果が日本語としてもっともらしい結果を出力しているのが分かる。一方、旧認識結果は人間が見ても正解発話を推測さえできない割合が高かった。推測が可能で、かつ、その推測結果が正解である割合は新旧でそう違いはない。

表 4-10　新認識結果の人間による評価

			発話数	割合
	正解と比べてＯＫ		81	27
	正解と比べてＮＯ		56	19
		正解と比べてＯＫ	66	22
		正解と比べてＮＯ	19	6
	推測ＮＯ		78	25
合計			300	

表 4-11　旧認識結果の人間による評価

			発話数	割合
	正解と比べてＯＫ		38	12
	正解と比べてＮＯ		17	5
		正解と比べてＯＫ	74	24
		正解と比べてＮＯ	30	10
	推測ＮＯ		153	49
合計			312	

４．５　今後の課題

現訂正手法において訂正が見込めるのは表 4-10 における、正しい推測が行える認識結果である。そこで、この認識結果の訂正履歴をもとに現手法の課題を分析した。

4.5.1　誤り位置予測

位置予測が正しくても訂正できるかどうかはわからないが正確な誤り位置予測は必須である。位置予測の改善が見込めるものとして次のような例がある。

【単語共起の利用】

正解発話：ＰＭ六時の予定です
認識結果：クレーム六時の予定です

この例では、３文字連鎖による予測で下記の誤り予測位置を示す。

誤り予測位置：クレーム<@0 時>の予定です

この場合、「六時」「クレーム」「予定」「です」の単語の共起確率を考えれば、「クレーム」が怪しいというのが検出できる可能性がある。同様の例として次のものがある。

正解発話：ツインだといくらぐらいからの料金ですか
認識結果：ツインだといくらぐらいからご用件ですか

【定型表現の利用】

正解発話：八月十三日から十五日までお願いしたいんですが
認識結果：八月十三日から十五時までお願いしたいんですが

この例では、「ＸからＹまで」という表現で出現する可能性の高いものをあらかじめ用意しておくことで、誤りを検出できる。この場合、下記の二つの定型表現にマッチするのだが、文頭の「八月」を考慮すれば（１）の定型表現の可能性が高くなり、誤り位置を正確に予測できる可能性がある。

（１）@0 日から@0 日まで
（２）@0 時から@0 時まで

【数字チェック】

これは出現する数字について一般常識を適用することで誤りを指摘するものである。

例えば、下記の例１のように「二八日」というのは日にちとしては存在しないことから明らかに誤りと指摘可能である。例２でも数字の意味が推測できれば誤りを検出できる可能性がある。

ただ、数字チェックは誤り検出だけで正しい数字の推測は難しい。

（例１）
正解発話：十月の二十七日と二十八日の二泊なんですけれど
認識結果：十月の二十七日と二八日の二泊なんですけど
（例２）
正解発話：十七十八と友人が来ますのでその時だけ友人と部屋を一緒にしたいのですが
認識結果：五十七十八と友人が行きますのでその時だけ友人とお部屋を一緒にしたいのですが

4.5.2 類似用例検索

誤り位置予測が正しくても訂正できない場合を分析すると、類似検索のキーとなる擬似形態素がうまくとれないために失敗している例が多く見られた。

擬似形態素の取り方がうまくいかない原因は二つある。一つは、誤り文字列から擬似形態素が選ばれてしまうということ、一つは人間が区切るだろう位置で擬似形態素がうまく区切られていないことである。

例えば、例１では、擬似形態素が「ビス」「には」になり、「には」などは誤り文字列から選ばれたものである。その結果検索された類似用例は的外れなものが選ばれている。

（例１）
正解発話：サービス料はどのようになっておりますか
認識結果：サービス料うなどにはなっていますか

誤り予測位置：サービス<料うな>どにはなっていますか
検索用擬似形態素：[ビス]<料うなど>[には]
検索結果：
(サー)[ビス]<はそちらのホテル>[には](な)
(サー)[ビス]<の明細はちょっと今こちら>[には](な)

一方、例２などは、人間の感覚では「人数」「増え」などで区切るのだが、高頻度文字列の上位１０００個を利用すると的外れな「と@0」が擬似形態素として選ばれてしまっている。

（例２）
正解発話：十七十八と一人人数が増えちゃうんですが
認識結果：十七十八と一人人数が増え違うんですが
誤り位置：@0@0@0@0 と@0 人人数が増<え違う>んですが
検索用擬似形態素：[と@0]<人人数が増え違う>[んですが]

上記問題の対策として次のような方法が考えられる。

【認識結果に含まれる単語による類似検索】

誤り近傍の擬似形態素だけに頼るだけではなく、認識結果に含まれる単語を利用して類似用例を検索するとうまくいく場合が多い。

例1でも、「サービス料」「なっています」などをもとにコーパスを検索すると次のような正解を含む発話を抽出することができる。

抽出発話：そうですかところで税金サービス料のほうはどうなっていますか
はい飲物がビールが@0@0 円それに税金とサービス料は別になっています

このような例はたくさんあり、単純に名詞だけを用いて検索しても正解に近い用例を検索できる場合も多い。例えば、例3などは「タクシー」と「予約」というキーワードだけでたくさんの用例が検索でき、その中に正解になりそうなものもあった。

（例3）

正解発話：はい分かりましたでタクシーは予約しておいた方がよろしいんでしょうか

認識結果：はい分かりましたタクシーが予約しておいてもよろしいでしょうか

grep タクシー * | grep 予約
@0 時@0@0@0 分ですねタクシーは予約しておいた方がいいでしょうか

ここまではしなくて、擬似形態素として高頻度文字列の上位N位を使用しているが、それ以外に名詞を含めるだけでもうまくいく可能性がある。また、擬似形態素ではなく認識結果の形態素をそのまま利用するという方法も考えられる。

第５章　人工誤りによる訂正性能評価

５．１　目的

訂正手法の評価の為に、様々な性能の音声認識結果を用いて実験を行ってきたが、評価結果は用いたデータに依存してかなりのばらつきがあった。また、実際の認識結果の誤りはいろりろな要素がランダムに絡み合っているため、訂正手法の長短を見極める解析が難しい。そのため、音声認識装置が出力する誤りを含む認識結果の代わりに、誤りの特徴を調整した人工的誤りを生成して、訂正手法の評価を行った。

５．２　人工誤り生成法

誤りの特徴として以下のものを人工的に区別して生成できる方法を用いた。

1)言語モデル（言語モデルに従がう、従がわない）
2)誤り種類（挿入、削除、置換、混合）
3)誤り品詞ｸﾞﾙｰﾌﾟ（内用語、機能語（助詞、助動詞）、混合）
4)誤り個数

（１）言語ﾓﾃﾞﾙ（LM）

言語ﾓﾃﾞﾙは誤り生成後に誤り単語を含む前後の単語連鎖が言語ﾓﾃﾞﾙと矛盾しないかどうかをﾁｪｯｸするものである。「言語モデルの重み（LM)」を用いて次のように実現している。

言語ﾓﾃﾞﾙの重みは０から１０の範囲の値をとり、０が言語モデルのﾁｪｯｸなし、１０が全ての人工誤りが言語モデルに従がうというものである。重みがＸ（０＜Ｘ＜１０）の場合、任意の生成誤りに対して、言語ﾓﾃﾞﾙに「従がう」「従がわない」の選択をＸの重みで確率的に行う。

言語モデルは品詞ｶﾃｺﾞﾘｰ 3-gram を用いた。

（２）　誤り種類（ES）

挿入、削除、置換、混合の４種類とする。混合の場合は等確率で３種類の中から選ぶ。

（３）誤り品詞ｸﾞﾙｰﾌﾟ（EP）

誤る単語の品詞が機能語（助詞、助動詞）か内容語（助詞、助動詞以外）を区別する。混合の場合は等確率で選択する。

（４）誤り個数（QT）

一文中に QT 個の誤りを再帰的に作成する。QT が２以上の場合、既に生成した誤り単語を再度書き換える場合もある。

図 5-1 に人工誤り生成のアルゴリズムを示す。また、表 5-1 に品詞ｶﾃｺﾞﾘｰ分類を示す。

図 5-1　人工誤り生成のアルゴリズム

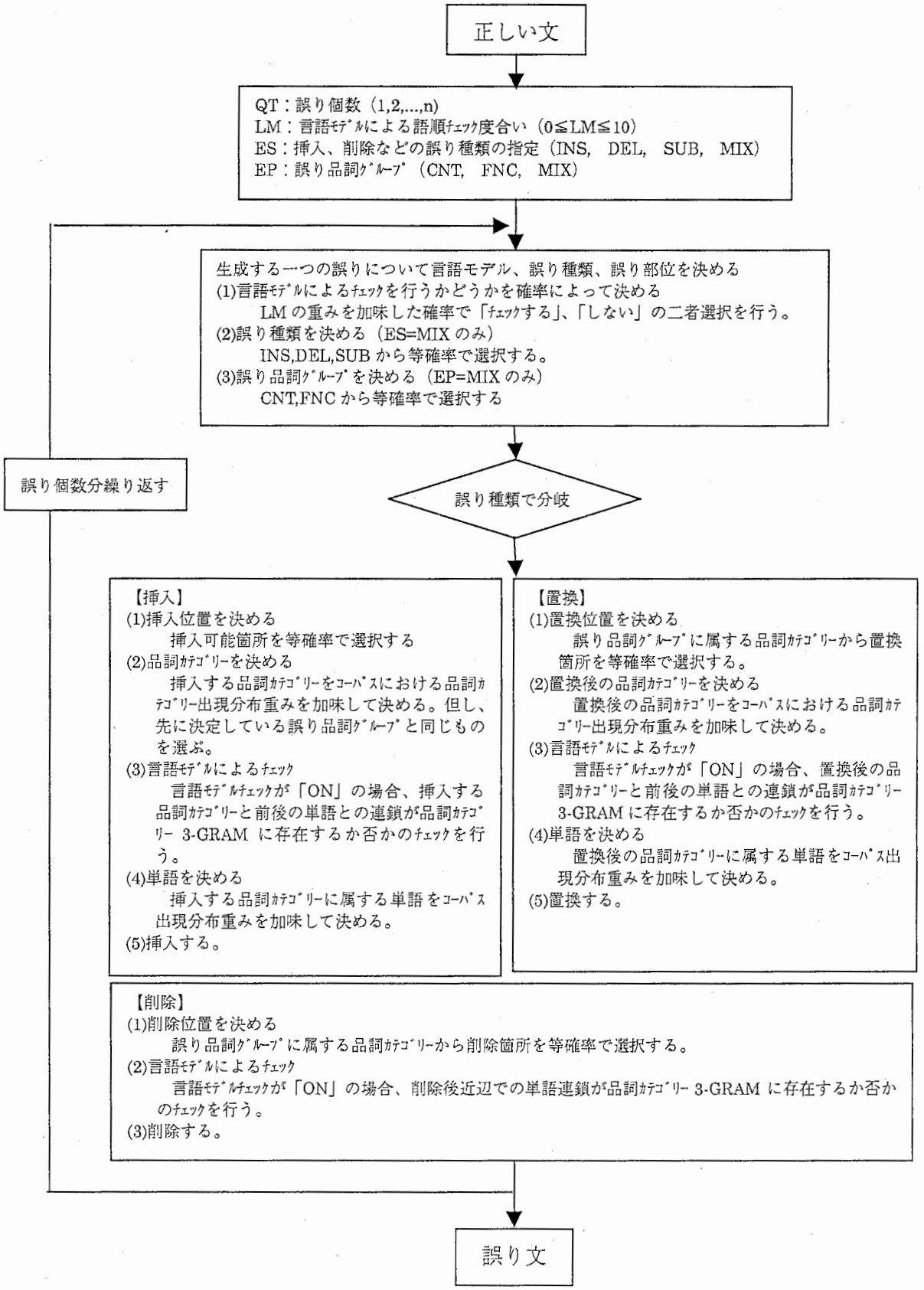

表 5-1　品詞ｶﾃｺﾞﾘｰと誤り品詞ｸﾞﾙｰﾌﾟ分類

品詞ｶﾃｺﾞﾘｰ	ｺｰﾊﾟｽ出現頻度	誤り品詞ｸﾞﾙｰﾌﾟ	品詞ｶﾃｺﾞﾘｰ	ｺｰﾊﾟｽ出現頻度	誤り品詞ｸﾞﾙｰﾌﾟ
サ変形容名詞	383	CNT	本動詞:う	61	CNT
サ変名詞	10862	CNT	本動詞:ず	18	CNT
ローマ字	507	CNT	本動詞:せる	561	CNT
格助詞	34237	FNC	本動詞:た	7463	CNT
感動詞	14481	CNT	本動詞:だ	296	CNT
係助詞	9912	FNC	本動詞:ない	599	CNT
形式名詞	2950	CNT	本動詞:ば	511	CNT
形容詞:すぎる	110	CNT	本動詞:べき	1	CNT
形容詞:そう	85	CNT	本動詞:れる	369	CNT
形容詞:た	83	CNT	本動詞:基本	4102	CNT
形容詞:ば	115	CNT	本動詞:命令	9	CNT
形容詞:基本	3626	CNT	本動詞:連用	16968	CNT
形容詞:連用	1663	CNT	連体形容詞	61	CNT
形容名詞	3147	CNT	連体詞	1885	CNT
終助詞	13552	FNC	連体助詞	17972	FNC
準数詞	3	CNT	普通名詞	70600	CNT
準体助詞	6749	FNC	副詞	8710	CNT
準体助動詞:基本	4611	FNC	副助詞	3034	FNC
準体助動詞:連用	5	FNC	並立助詞	1512	FNC
助動詞:う	333	FNC	補助動詞:う	88	FNC
助動詞:ず	1	FNC	補助動詞:ず	1	FNC
助動詞:せる	4	FNC	補助動詞:せる	457	FNC
助動詞:そう	3	FNC	補助動詞:た	2078	FNC
助動詞:た	5251	FNC	補助動詞:ない	161	FNC
助動詞:ない	267	FNC	補助動詞:ば	139	FNC
助動詞:ば	168	FNC	補助動詞:べき	1	FNC
助動詞:れる	28	FNC	補助動詞:れる	276	FNC
助動詞:基本	24850	FNC	補助動詞:基本	3476	FNC
助動詞:命令	873	FNC	補助動詞:命令	445	FNC
助動詞:連用	5573	FNC	補助動詞:連用	6936	FNC
人称接尾辞	3216	CNT	接尾辞	223	CNT
数詞	29314	CNT	代名詞	5281	CNT
接続詞	6667	CNT	判定詞:た	665	CNT
接続助詞	16487	FNC	判定詞:基本	13627	CNT
接続副詞	415	CNT	判定詞:連用	3453	CNT
接頭辞	16474	CNT			

5．3　実験データ

（1）訂正元ﾃﾞｰﾀ

これまでの新旧認識結果における新旧で同じ発話約３９５発話を用いる（重複する発話はマージした)。

（2）コーパス

新の copus04 を用いる。

表 5-2　コーパス

タスク数	発話数	種類	のべ単位数	異なり単位数
2473	28603	文字	753214	1326
		品詞	389044	71

5．4　結果・考察

実験結果を表 5-3-1～5-3-4、及び図 5-2-1～5-2-3 に示す。表中の値は誤り個数の減少率である（正は訂正後に減少、負は増加を表す）。結果は次のとおりであった。

言語ﾓﾃﾞﾙの影響：言語ﾓﾃﾞﾙによる制限がゆるくなるほど訂正性能は上がる傾向がある。
誤り品詞ｸﾞﾙｰﾌﾟの影響：内容語と比較して機能語の誤りに対しては高い訂正精度を示す。
誤り種類の影響：挿入誤りは訂正精度が高く、他の２種類の誤りはそれと比べて低く、両者とも同程度の精度であった。
誤り個数の影響：はっきりした傾向は読み取れなかった。

表 5-4-1～5-4-2 に新旧認識結果の誤り傾向を形態素単位でまとめた結果を示す（対象発話は 4-3 節と同じ新旧で共通する発話である）。

挿入誤りの割合、及び機能語の割合が「旧」では「新」より多い。したがって、表 4-9 に示す SSC 訂正精度の新旧の違いは、人工誤りによる訂正精度の結果から、これら誤り分布の違いによるものと思われる。

表 5-3-1　言語ﾓﾃﾞﾙの影響、誤り種類＝混合、誤り個数＝１

誤り品詞ｸﾞﾙｰﾌﾟ	言語ﾓﾃﾞﾙ										
	10	9	8	7	6	5	4	3	2	1	0
機能語	4.03	3.52	4.05	7.56	6.05	5.28	7.32	4.52	11.59	8.31	11.03
内容語	-2.44	-1.76	4.28	-0.50	-2.00	3.54	0.00	0.50	3.29	1.26	2.26
混合	2.26	2.76	-1.25	0.00	5.81	7.52	0.76	6.53	1.75	9.75	5.82

表 5-3-2　誤り個数の影響、誤り種類＝混合

誤り品詞ｸﾞﾙｰﾌﾟ	言語ﾓﾃﾞﾙ	誤り個数									
		1	2	3	4	5	6	7	8	9	10
機能語	１０	3.02	11.44	10.51	8.60	9.56	11.43	10.23	9.82	9.74	9.98
	0	7.87	14.46	13.24	15.60	14.08	13.47	11.34	11.56	10.17	11.31
内容語	１０	-2.02	4.23	6.04	5.21	5.07	5.15	6.03	6.48	5.63	6.47
	0	4.25	5.94	9.44	8.17	9.59	9.25	9.24	9.42	8.01	8.79
混合	１０	-3.00	5.99	7.00	7.77	6.60	6.96	6.58	6.82	6.31	4.86

表 5-3-3　誤り種類の影響、誤り品詞ｸﾞﾙｰﾌﾟ＝混合、言語ﾓﾃﾞﾙ＝１０

誤り種類	誤り個数									
	1	2	3	4	5	6	7	8	9	10
挿入	9.62	16.88	12.74	14.32	13.37	10.21	9.21	7.91	9.16	9.05
削除	-6.19	1.67	2.50	3.77	4.93	4.51	3.31	4.04	3.66	3.38
置換	1.51	1.10	3.68	4.04	2.79	3.50	4.31	3.71	3.84	2.02

表 5-3-4　誤り種類の影響、誤り品詞ｸﾞﾙｰﾌﾟ＝混合、誤り個数＝１

誤り種類	言語ﾓﾃﾞﾙ										
	10	9	8	7	6	5	4	3	2	1	0
挿入	12.41		9.11		11.14		16.20		12.91		21.01
削除	-8.25		-8.95		-5.12		-5.12		-6.65		-9.72
置換	-1.99		6.95		-2.24		-2.22		5.01		1.24

表 5-4-1　新旧認識結果の誤り傾向（形態素単位）

誤り種類

種類		挿入	削除	置換	合計
新	個数	71	120	778	969
	割合	7.33	12.38	80.29	100
旧	個数	143	76	1057	1276
	割合	11.21	5.96	82.84	100

表 5-4-2　新旧認識結果の誤り傾向（形態素単位）

誤り品詞ｸﾞﾙｰﾌﾟ

種類		内容語	機能語	合計
新	個数	632	337	969
	割合	65.22	34.78	100
旧	個数	745	531	1276
	割合	58.39	41.61	100

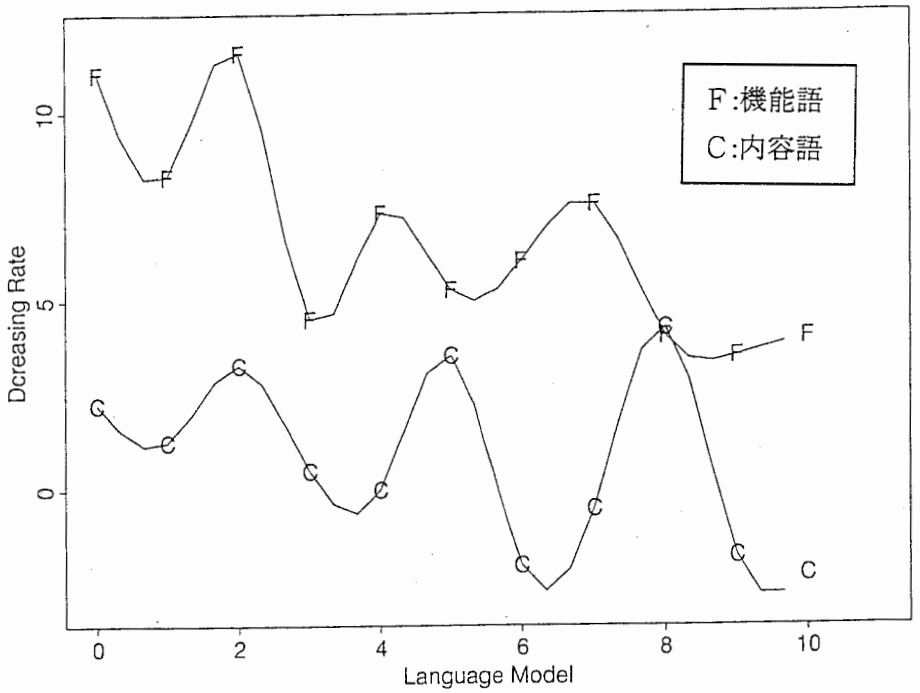

図 5-2-1　言語ﾓﾃﾞﾙの影響
（誤り種類＝混合、誤り個数＝１）

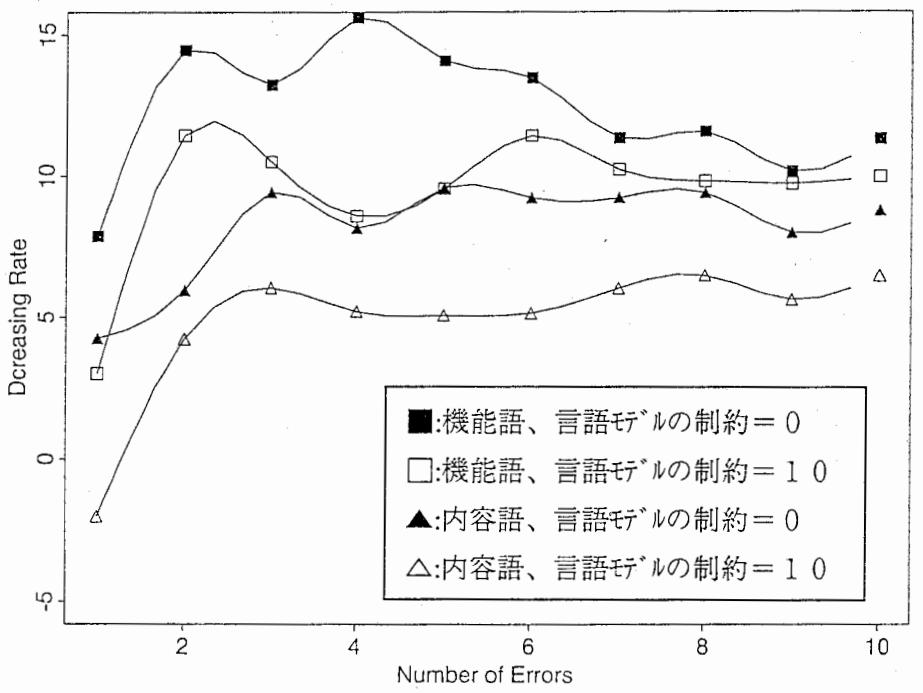

図 5-2-2　誤り個数の影響
（誤り種類＝混合）

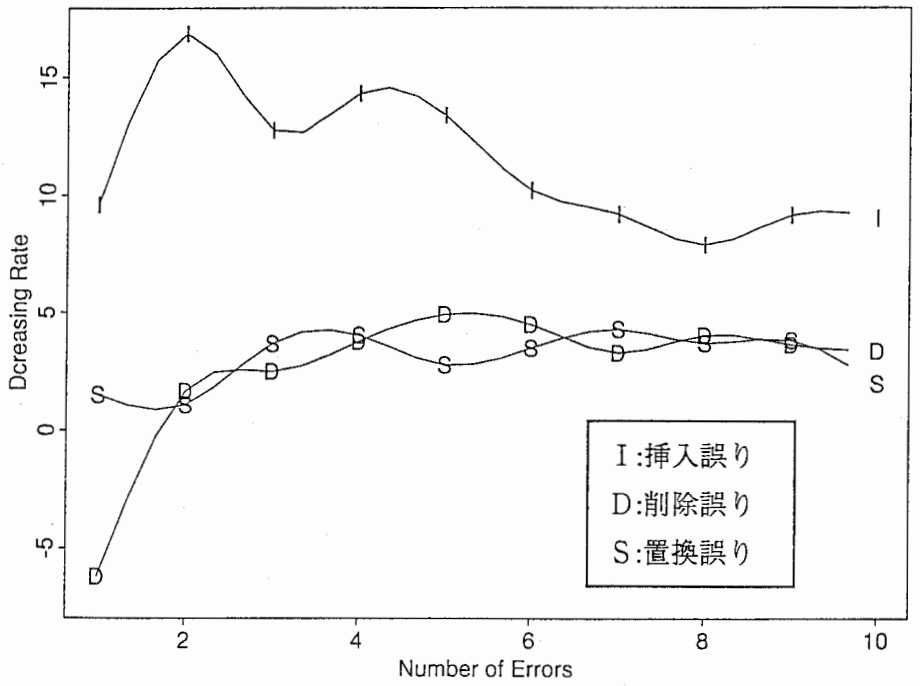

図 5-2-3　誤り種類の影響
（言語ﾓﾃﾞﾙの制約＝１０、誤り品詞ｸﾞﾙｰﾌﾟ＝混合）

第６章　訂正サーバー

EPC、SSC 手法に基づく日本語訂正サーバーを作成した。６．１節に概要を、６．２節以降に必要なデータに関して述べる。

６．１　概要

訂正プログラムは、クライアント、サーバーシステムで構成される。訂正の本体は訂正サーバーである。訂正サーバーへの入力は、日本語文を形態素分割した形で与える。訂正結果は入力と同様のフォーマットで出力される。訂正に際し入力に含まれる間投詞は除去される。

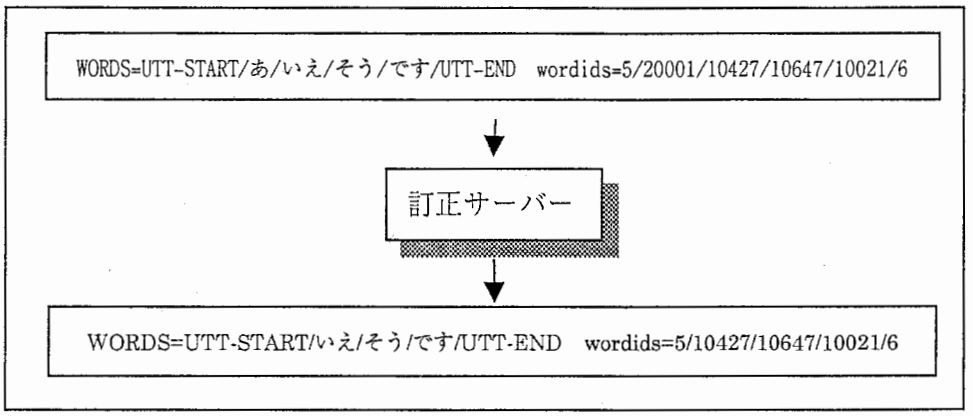

図 6-1　訂正サーバー

（１）起動

訂正に関するデータ及びパラメータを記述したファイル（config_file）を指定して起動する。

cor_main.pl config_file

（２）停止

「CNTL-C」で終了する。

（３）config_file

表 6-1 に示すパラメータがある。表中の「□、■」で示した項目は高速化バージョンに関わるパラメーターである。「■」は高速化バージョンのみに必要なもの、一方、「□」は高速化によって指定するものが変わるパラメーターである。

（４）クライアントプログラム

サーバーへの入力は、日本語文を音声認識結果の WORDS、wordids フォーマットと同様の形式で、スペースで区切って与える。両者がないものは入力がそのままの形でサーバー側から出力される。

サーバーからの出力も入力と同じ形式である。但し、訂正はなくても間投詞は除去されている。

表 6-1　起動パラメーター

	パラメータ	機能	選択肢
	dic_type	形態素体系を指定する	TDMT SLDB
	sw_model	訂正の種類を指定する	EPC .... EPC のみ SSC .... SSC のみ BOTH .... EPC+SSC
	sw_mor	訂正単位の指定 （注意）今のところ形態素単位のみしかできない。	MOR .... 形態素単位 CHAR .... 文字単位
	SSC_ngram	SSC の誤り検出の N-GRAM 種類	TRI .... 3-ngram BI 2-ngram
	ngram_file	N-GRAM ファイル名	
	MACHIN	実行マシン	ALPH .... alpha SUN .... sun
	abs_dic1 abs_dic2 abs_dic3	形態素辞書 曜日辞書 固有名詞、人名辞書	
□	hreq_word_file	高頻度文字列ファイルの指定 （高速化バージョンの場合はインデックスファイルを指定する）	
	copus_dir copus_ids_dir	ｺｰﾊﾟｽ関係のファイル指定	
	epc_file	誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽファイルの指定	
	port	ソケットポート番号	
	hreq_limit	擬似形態素に用いる高頻度文字の使用数	
	DP_limit	ＳＳＣの類似度（０～１）	
	ngram_limit	誤り検出の閾値（０から－６．０）	
	match_condition	類似文字列と誤り文字列のﾏｯﾁﾝｸﾞ条件（０、１）	通常は 0
	cand_max	類似文字列の候補数上限（１以上）	
■	copus_file	文ー形態素ファイルを指定	copus-sent_mor-1000
■	TH	誤りパターンと比較する文の長さの閾値(ー TH<文の長さ<＋TH)	

＜config_file 例＞

```
dic_type = TDMT
abs_dic1 = /data/as37/skaki/sreg/LAST/dic/recg02.dic
abs_dic2 = /data/as37/skaki/sreg/LAST/dic/recg01.dic.youbi
abs_dic3 = /data/as37/skaki/sreg/LAST/dic/recg02.dic.diff.2
ngram_file = /data/as37/skaki/sreg/LAST/copus04/ngram/char.ngram.3.val
hreq_word_file = /data/as37/skaki/sreg/LAST/copus04/ngram/hreq.sort
copus_dir = /data/as37/skaki/sreg/LAST/copus04/abs_wd/
copus_ids_dir = /data/as37/skaki/sreg/LAST/copus04/abs_id/
epc_file = /data/as37/skaki/sreg/LAST/recg02/epc-mor/epc.blk42.lm2
port = 11113
SSC_ngram = TRI
sw_model = EPC
sw_mor = MOR
hreq_limit = 1000
DP_limit = 0.7
ngram_limit = -4.5
match_condition = 0
cand_max = 10
MACHIN = ALPH
```

6．2　辞書

間投詞の除去、一部単語の抽象化などの処理のため、表 6-2 に示す辞書を使用している。

表 6-2　使用辞書

abs_dic1	形態素辞書		【TDMT 体系】 /data/as37/skaki/sreg/LAST/dic/recg02.dic （/DB/SHARE/MASTER_LEX/TDMT/MASTER_LEXICON に形態素を追加したもの） 【SLDB 体系】 /DB/SHARE/MASTER_LEX/SLDB/MASTER_LEXICON
abs_dic2	固有名詞、人名の抽象化用辞書	固有名詞、人名の抽象化に利用 （TDMT 体系のみ必要）	/data/as37/skaki/sreg/LAST/dic/recg02.dic.diff.2
abs_dic3	曜日抽象化用辞書	曜日の抽象化のために利用 （TDMT 体系のみ必要）	/data/as37/skaki/sreg/LAST/dic/recg01.dic.youbi

6．3　EPC 用データ作成

EPC で用いる誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽを表 6-3 に示す手順で作成する。

表 6-3　誤りﾊﾟﾀｰﾝﾃﾞｰﾀﾍﾞｰｽ作成手順

処理	概要	参考プログラム[1]
認識結果と対応する正解発話ﾃﾞｰﾀの作成	認識結果から WORDS,wordids を取出す。また、対応する正解の WORDS,wordids を作成する。	last_30_1_1.pl
誤り傾向をまとめる（形態素ベース）	認識結果と正解発話から誤り傾向を取り出す。	last_35_1.pl
誤りパターン抽出（形態素ベース）	誤りﾊﾟﾀｰﾝの抽出を行う。	last_35_2.pl

6．4　SSC 用データ作成

SSC で用いるデータ作成手順を表 6-4 に示す。

表 6-4　SSC 用データ作成手順

処理	概要	参考プログラム
ｺｰﾊﾟｽ基礎データ作成	コーパスファイルから WORDS,wordids を取出す。	last_32_1.pl last_32_2.pl
ｺｰﾊﾟｽの抽象化	間投詞抜きで抽象化したコーパスと、形態素情報を作成する。	last_33.pl
N-GRAM 作成	表 6-5 参照	
高頻度文字列の作成	（1）コーパスから高頻度文字列の抽出 （2）出現頻度順にソートする。	hreq_01.pl
インデックス作成	高速化のためのｲﾝﾃﾞｯｸｽを作成する	mkindex.prl mkindex_resource.prl

表 6-5　N-GRAM の作成手順

処理内容	参考プログラム	入力データ	出力データ
文字のコード化とコードファイル作成	mong_04_01.pl	間投詞抜きで抽象化したコーパスデータ	dir/char.code dir/uniq.char dir/dic.char
N-gram 辞書テーブルを作成する	mong_04_02a.csh （alpha 版）	dir/char.code	dir/char.tbl
N-gram を作成する	mong_04_03.csh N N: N-gram の N	dir/char.code dir/char.tbl	dir/char.ngram.N
2-,3-gram の確率を計算する	mong_04_04.pl	dir/char.ngram.1 dir/char.ngram.2 dir/char.ngram.3	dir.char.ngram.2.val dir.char.ngram.3.val

[1] 参考ﾌﾟﾛｸﾞﾗﾑは全て” /home/as37/skaki/src/ngram” ﾃﾞｨﾚｸﾄﾘｰにある。

第７章　翻訳文の機械的評価

７．１　概要

翻訳文の機械的評価の可能性を検討した。検討に用いた尺度は文字３-GRAM と翻訳時の尤度である。

７．２　実験データ

実験に用いたデータは以下のとおりである。

7.2.1　人間評価付き翻訳文

TDMT による英語から日本語への翻訳結果と、訳文に対する人間による評価付きデータである。人間評価はＡ、Ｂ、Ｃ、Ｄの４ﾗﾝｸがあるが、本実験では評価Ａとそれ以外の二つに分けて実験を行った。評価Ａの訳文は「正訳文」、それ以外のＢ、Ｃ、Ｄの訳文を「誤訳文」と呼ぶことにする（表7-1参照）。

表7-1　人間評価付き翻訳文

人間評価ﾗﾝｸ	正訳文	誤訳文			計
	A	B	C	D	
文数	663	357	128	193	1341

7.2.2　翻訳尤度

TDMT の翻訳尤度を求めるため、上記翻訳データの英文を再度 TDMT を用いて日本語への翻訳実験を行った。翻訳条件は次のとおりである。

TDMT の翻訳条件

```
/usr/local/TDMT/tdmt-multi-6-eval-970902/build/setup-ej.lisp
(setq *disable-sem-code* nil)
(setq *all-targets* nil)
(setq *generate-one* t)
(translate-method :separate)
```

この翻訳実験結果で出力された日本語訳文のうち６２文は人間評価付き訳文と一致しなかったため、翻訳尤度を用いる実験対象から除外した（表7-2参照）。

表7-2　人間評価付き翻訳文（62文除外）

人間評価Ａ（正訳文）	Ａ以外（誤訳文）	計
650	629	1279

7.2.3　文字 3-GRAM

コーパスは旧認識結果の誤り訂正に用いたものと同じである（但し、間投詞の除去は行っていない）。

表7-3　コーパス

タスク数	発話数	延べ文字数	異なり文字数
894	20176	570306	1396

7．3　判別精度実験

説明変数として表 7-4 に示す 4 変数を用いた。

誤り検出は、文字 3-GRAM の連鎖確率が閾値以下の部分を誤りと見做した。その際、訳文中の句読点は除いた状態で各文字位置の連鎖確率を計算している。

表 7-4　説明変数

略称	内容
誤り個数	文字 3-gram による誤り検出個数
尤度閾値個数	TDMT の sub 尤度で閾値以上（0.5）の個数
トータル尤度	TDMT のトータル尤度
規格尤度	TDMT のトータル尤度を sub 尤度個数で規格化したもの

7.3.1　説明変数相互間の相関係数

表 7-5 に説明変数間の相関係数を示す。トータル尤度と規格尤度が最も相関が高く、次いでトータル尤度と尤度閾値個数である。誤り個数と他の説明変数との相関はかなり低い。

表 7-5　相関係数

	誤り個数	尤度閾値個数	トータル尤度	規格尤度
誤り個数	1.0000001	0.7488205	0.5498410	0.3926217
尤度閾値個数	0.7488205	1.0000001	0.8058799	0.6491476
トータル尤度	0.5498410	0.8058799	1.0000000	0.8507769
規格尤度	0.3926217	0.6491476	0.8507769	1.0000000

これは、尤度関連の 3 変数は TDMT の翻訳に際し、訳文対象が学習した用例にあったか否かを現す尺度であり、一方、文字 3-GRAM は TDMT の学習した用例とは関係なく、訳文の表現が文字 3-GRAM 作成に用いたｺｰﾊﾟｽに出現する表現に適合するかどうかを見ているためである。

7.3.2　相関比（級間分散／分散）

各説明変数が翻訳文を「正訳文」か「誤訳文」に判別する性能の良否尺度である相関比を求めた。結果はいづれの説明変数も判別にはよくない結果であった（表 7-6 参照）。

表 7-6　相関比

誤り個数	0.1407135
尤度閾値個数	0.1627453
トータル尤度	0.1119328
規格尤度	0.1010122

相関比の参考値
0.8～1.0：非常によい
0.5～0.8：ややよい
0.5 未満：よくない

7.3.3　マハラノビスの汎距離による判別分析

マハラノビスの汎距離による判別分析を行った。また、説明変数の「(有無)」はその変数の値をそのまま用いるのではなく、値が 0（無）か 1 以上（有）かで判定したものである。

結果を表 7-7 に示す。いずれの判別精度も良くない。変数を組み合わせても判別的中率は単独よりも下がる傾向にある。

表 7-7　マハラノビスの汎距離による判別分析結果

説明変数	判別的中率	正訳文予測	
		適合率	再現率
誤り個数	66.22316	64.6	74.3
誤り個数（有無）	64.97263	73.1	49.2
尤度閾値個数	69.89836	69.9	71.5
尤度閾値個数（有無）	69.89836	69.9	71.5
トータル尤度	70.75841	66.9	84.0
規格尤度	69.1165	66.4	79.5
トータル尤度＋誤り個数	69.03831	67.8	69.8
トータル尤度＋尤度閾値個数	69.1165		
トータル尤度＋規格尤度	69.35106	68.0	77.4

判別的中率の参考値
90～100：非常によい
75～90：ややよい
50～75：よくない

7．4　文字3-GRAMによる判別結果の分析

7.4.1　正訳文を誤訳文と判別した原因とその頻度

文字3-GRAMによって正訳文中に誤りを検出した原因を表7-8にまとめた。

これらの内、Ｃ１やＣ３のようにｺｰﾊﾟｽを改良して解決するものが約３０％、そして、訳文の表現がｺｰﾊﾟｽに比較して練れていないために起こったものが約２０％であった。残り約４０％は訳文中の表現がｺｰﾊﾟｽに一度も出現していないためであった。

図7-1に訳文表現が練れいない代表例を示す。

表7-8　誤り判別の原因

原因	頻度
C1：漢字／ひらがな表記の違いによるもの	24
C2：判別が正しく、人間評価に誤りがある	8
C3：数値の抽象化で解決	12
C4：コーパスに類似用例があるが、訳文と若干異なる（訳文がいまひとつ）	14
C5：区切りを句読点で表現するか助詞をいれるかの違い	1
C6：若干の変更でOKになる（訳文がいまひとつ）	5
C7：コーパス表現にならって語順などを入れ替えるとよくなる	4
C8：語順などを入れ替えると若干よくなる	2
C98：コーパスに表現なし（複文も含む）	47
C99：その他	2
計	119

	頻度	割合（%）	原因
C2	8	6.7	人間評価ミス
C4+C6+C7+C8	25	21.0	訳文の生成の問題
C1+C3	38	31.9	コーパス改良
C98	47	39.5	コーパスに表現なし

図7-1　訳文表現が練れていない例

下記の例はいずれも訳文は人間評価でＡにﾗﾝｸされている。「＊」は文字3-GRAMで検出した誤り位置を示す。

はい鈴木様いつ御予約が御希望ですか
　　****　****
はい鈴木様いつの御予約が御希望ですか
　　****
はい鈴木様いつの御予約がご希望ですか

二十九日に何時に到着しますか鈴木様です
　****
二十九日の何時に到着しますか鈴木様です

以下の例は「有効期限」を含む表現をコーパスから取り出したもので、訳文と比較すると文章の完成度は格段の差がある。

E:"I will need the card number and the expiration date"
J:"カード番号と有効期限が必要です。"
N=カード番号と有効期限が必要です
L=　　****

「有効期限」を含むコーパス文
はいそれではカード番号と有効期限の方お伺いできますか
ではカード番号と有効期限を教えていただけますか
ではカード番号と有効期限を教えていただけますか
ではカードナンバーと有効期限をお願いいたします
分かりました鈴木様それからカードの有効期限を教えていただけませんか
カードの有効期限はいつですか

7.4.2　誤訳文を指摘できなかった原因

誤訳文であるにも関わらず、文字 3-GRAM による誤り検出で指摘できなかったものは、ほとんどが短い慣用句で状況（文脈）によって訳し別けが必要なものであった。

7．5　TDMT のトータル尤度による判別結果の分析

7.5.1　誤訳文を正訳文と判別したケース

誤訳文を正訳文と判別したもので、頻度が2以上のものを表 7-9 に示す。頻度が2以上という条件をつけたためか、短い文が多い。ほとんどが、訳文の表現が固いか、文脈による訳仕分けがうまくできていないものであった。特に、同じ英語文の翻訳結果に対して、人間による評価が別れるものは「訳仕分け」の問題を端的に示している。

7.5.2　人間評価が別れるケース

同一訳文で人間評価の判定が別れるものを表 7-10 にまとめた。これらは訳文の評価が文脈に依存したケースと思われる。

表 7-9　誤訳文を正訳文と判別したケース

頻度	評価が別れるもの（○）	英語入力文→日本語訳文
4	○	All right -> 結構です。
2	○	Certainly sir -> もちろんです。
8	○	Certainly -> もちろんです。
2	○	Fine -> 結構です。
2	○	Good -> 良いです。
2		Great -> 素晴らしいです。
2	○	I'm sorry -> 申し訳ないです。
2	○	Is that right -> 正しいですか。
2	○	Of course -> もちろんです。
3		Okay great -> はい、素晴らしいです。
29	○	Okay -> よろしいです。
3		Right -> 正しいです。
6		Sure -> 確かです。
4		That sounds fine -> 良さそうです。
5		That sounds good -> 良さそうです。
2	○	That sounds great -> 良さそうです。
2		That'd be great -> 素晴らしいです。
4	○	That's fine -> 良いです。
2		Very well -> とても良いです。
2		Will that be all right -> 大丈夫ですか。
2	○	Yes certainly -> はい、もちろんです。
2	○	Yes of course -> はい、もちろんです。
7		Yes please -> はい、どうぞ。
2		Yes that's correct -> はい、正しいです。
16	○	Yes -> はい。

表 7-10　人間評価で判定が別れるケース

正訳文	誤訳文	英語入力文→日本語訳文
10	4	All right -> 結構です。
1	2	Certainly sir -> もちろんです。
1	8	Certainly -> もちろんです。
1	1	Do you have any other suggestions -> 他の提案がありますか。
1	2	Fine -> 結構です。
1	2	Good -> 良いです。
35	1	I see -> 分かりました。
1	2	I'm sorry -> 申し訳ないです。
3	2	Is that right -> 正しいですか。
1	1	No -> いいえ。
1	2	Of course -> もちろんです。
9	29	Okay -> よろしいです。
21	1	Thank you very much -> どうもありがとうございます。
1	2	That sounds great -> 良さそうです。
2	4	That's fine -> 良いです。
6	1	That's right -> そうです。
1	2	Yes certainly -> はい、もちろんです。
2	2	Yes of course -> はい、もちろんです。
5	1	Yes that's fine -> はい、良いです。
22	16	Yes -> はい。
2	1	You're very welcome -> どういたしまして。
127	86	

7．6　文脈による訳仕分け部分を除いた判別精度に関する実験

翻訳に対する機械的評価手法の利用が科学文献などの書き言葉翻訳であれば、上述した訳仕分けが必要な短いフレーズが出現することは少ないと思われる。そこで、これまでの訳文データから訳仕分けが必要な発話を除外して判別精度の再評価を行った。

表 7-12　評価翻訳文内訳

正訳文	誤訳文	計
523	505	1028

7.6.1　実験条件

評価に際して、訳仕分けが必要なフレーズ（今回の評価から除くもの）は次のものである。

1）人間評価が同一文で正訳文と誤訳文に別れるもの。

2）トータル尤度を説明変数とする判別分析で誤訳文を正訳文と誤り判別し、かつ頻度が2以上のもの

表 7-11 に再評価において除外した英文を示す。

その結果、実験に用いた翻訳文は表 7-12 のとおりである。

表 7-11　除外したフレーズ

"Yes certainly"	"Very well"
"I'm sorry"	"No"
"Fine"	"Yes that's fine"
"Thank you very much"	"Right"
"Of course"	"Sure"
"I see"	"That sounds great"
"Will that be all right"	"Good"
"Okay great"	"That's right"
"Is that right"	"Yes please"
"Certainly"	"That's fine"
"You're very welcome"	"Yes that's correct"
"That sounds fine"	"Yes"
"Okay"	"Great"
"That'd be great"	"That sounds good"
"All right"	"Yes of course"
"Certainly sir"	"Do you have any other suggestions"

7.6.2 実験結果

判別精度は表 7-13 に示すとおりである。

表 7-13 判別精度

説明変数	ケース	判別的中率	正訳文予測	
			適合率	再現率
トータル尤度	文脈文除去	74.12451	70.6	84.3
	除去前	70.75841	66.9	84.0
誤り個数（有無）	文脈文除去	65.95331	86.2	39.4
	除去前	64.97263	73.1	49.2

7.3.3 考察

（1）判別の精度はどのケースも上昇している。
（2）トータルの性能としては、トータル尤度が優れているが、訳文評価結果に基づいて人間による編集処理を行うということになると適合率の高い誤り個数（有無）がよい。
（3）誤り個数(有無)、トータル尤度ともに、誤訳文を正訳文と誤判別するケースを表 7-14 にしめす。
文脈と生成によるものがほとんどである。翻訳の性質上、原因がそこに集中するのはしかたないことである。
（4）これ以上の判別精度の向上には、文脈などを考慮することになり、判別機能としては役目を超えたものになる。

7．8 今後課題

今後の課題として次のことが考えられる。

1）翻訳文の良否判別を尤度、あるいは文字 N-GRAM でやるには限界がある。
　　n-gram はコーパスや抽象化などで若干の精度向上が見込まれる可能性はある。
2）文脈に依存した翻訳が必要である。
3）訳文の表現が練れていない。
4）文字 3-GRAM が指摘する誤りに対して、コーパスに頻繁に出現する言い回しを提示してやれば、人間が編集するための補助として有効である。

表 7-14　誤訳文を正訳文と誤判別するケース

英文	訳文	評価者コメント	分類
A twin	ツインです。	[b(5.0e-6)]不自然	文脈
And may I have a telephone number where we can contact you Ms. Suzuki	それから、連絡することができる電話番号を何ってもよろしいですか、鈴木様。	[b(1.00002)]「連絡することができる電話番号」は「連絡先」でないと不自然。	生成
And we'll have it sent up	それから、あります。	[d(0.8333584)]何があるのか不明	?
Dear	まあ。	[b(0.0)]女性言葉なので、話者が男性の場合はおかしく聞こえる。	話者
Do you think you will take the tour	ツアーに参加すると思いますか。	[b(2.0e-5)]不自然な表現	生成
I prefer to call myself	電話したいです。	[b(1.0000049)]「preferto」が「～したい」と訳されていることと、前の文とのつながりが不自然。日本語は間違っていないが、「電話は自分でする」という話者の意図を反映していない出力。	文脈
I'm afraid not	そう思いません。	[b(5.0e-6)]「思う」は不適切	文脈、生成
It was very good	とても良いです。	[b(1.4999999e-5)]食べ物の「～wasgood」だから「おいしかった」とするべき。	文脈
Just a moment	少しだけです。	[d(0.0)]リクエストになっていない。	生成
Let's see then	では、そうですね。	[d(5.0e-6)]意味不明	生成
Let's see	そうですね。	[d(0.0)]何に対するあいずちか分からない。	生成
No I don't think so	いいえ、そう思いません。	[c(2.0e-5)]否定の表現が英語的で、英語の知識のない人にはわかりにくいのではないかと思う。	生成
Okay I understand	はい、分かります。	[b(1.0e-5)]未完了形で表現すると、「承知した」という意味ではなく、「言っている意味が分かる」と解釈される。原文の英語「Iunderstand」も少々不自然だと思う。	文脈、生成
Okay let's see	はい、そうですね。	[d(0.6666667)]関係のない発話になっている。	文脈
Okay thanks for waiting	はい、お待たせしてありがとうございます。	[b(1.0e-5)]「お待たせしてすみません」となるべき。	生成
Okay then	では、よろしいです。	[b(5.0e-6)]「では」がちぐはぐな印象を与える。	生成
Sure that'd be no problem	はい、大丈夫です。	[b(0.6666817)]「大丈夫」は変。	生成
Thank you Mr. Suzuki	ありがとうございます、鈴木様。	[d(5.0e-6)]ここで礼を言うのはおかしい。	文脈
Thank you very much for waiting Mr. Suzuki	お待たせしてどうもありがとうございます、鈴木様。	[b(0.6666767)]「ありがとうございます」の直訳は変。日本語では「すみません」とすべき。	生成
Very well then	では、とても良いです。	[b(5.0e-6)]不自然な応答	文脈、生成
We're staying at the Kyoto Kanko Hotel room two fifteen	京都観光ホテルに滞在しています、二、十五号室です。	[b(1.000025)]3 桁の数字が英語読みになっているため、部屋番号がはっきりしない。	解析
Would that be all right	それで大丈夫ですか。	[b(0.0)]「大丈夫」が不適切	文脈、生成
Yes I am	はい、います。	[d(1.0e-5)]Be 動詞の用法が間違って訳されたため、質問に対する答えになっていない。	解析
Yes I can	はい、できます。	[d(1.0e-5)]質問に答えていない。	文脈
Yes it is	はい、あります。	[d(1.4999999e-5)]質問に対する答えになっていない。	文脈
Yes it's a dollar ninety five	はい、九十五ドルです。	[d(0.33335838)]値段の誤訳	解析
Yes that's it	はい、それです。	[d(0.7333484)]質問に答えていない。	文脈
Yes they are	はい、あります。	[d(1.4999999e-5)]異なる動詞を使っているので、質問の答えになっていない。	文脈
You're welcome Ms. Suzuki	どういたしまして。	[b(1.000005)]不自然だが、障害にならない。	文脈

付録１　音声認識誤りの検出手法とその評価

１　検出の手法

誤り検出は、予め大量の会話データベースから文字もしくは品詞・単語混合連鎖確率モデルを作成しておき、このモデルを用いて音声認識結果の連鎖単位ごとに連鎖確率を計算し、連鎖確率が与えられた閾値以下になる部分を基に誤り部分を特定するものである。

ここで、品詞・単語混合連鎖確率モデルとは連鎖の単位として自立語は品詞で分類してまとめたカテゴリーとして扱い、付属語（助詞および助動詞）は各単語独立したカテゴリーとして扱ったものである。

検出手順は以下のとおりである。

【連鎖確率の計算】

まず、以下の手順で連鎖確率を計算する。

手順１）入力発話文を連鎖確率モデルと同じ単位に分割する。
品詞・単語混合の場合は、形態素解析を行う。

手順２）分割された単位ごとに、前方から順次、その単位の連鎖確率を求める。
連鎖確率はデータベースから得られた n-gram（連鎖確率モデル）に基づいて、次式によって計算する。

$n-1$ 連鎖（C_{j-n+1}、…、C_{j-2}、C_{j-1}）の次に X_j が出現する連鎖確率　＝

$$P(X_j \mid C_{j-n+1}、\ldots、C_{j-2}、C_{j-1}) =$$

log（X_j を n 番目とする n-gram 頻度　÷　直前の(n-1)-gram 頻度）

【誤り区間の推定】

次に、以下の手順で連鎖確率が与えられた閾値以下の部分から誤り区間を推定する。（ここでは３単位の連鎖確率モデルに述べており、２単位の連鎖確率モデルと異なるところは{}内に記述している）。

手順１）連鎖確率が与えられた閾値より低い単位（文字、あるいは単語）が連続するものを誤りブロックとする。誤りブロックの始点を Pos1、終点を Pos2 とする。

手順２）**終点の調整**

誤りブロックの単位数を Len として、

Len が３単位以上｛２単位以上｝の長さの場合

誤りブロックの後ろから２単位｛１単位｝を誤りブロックから取り除く。

Pos2 ← Pos2 − 2

｛Pos2 ← Pos2 − 1｝

Len が３単位未満｛２単位未満｝の場合

そのまま。

とする。

手順３）**始点の調整**

誤りブロックの始点が文の先頭でなければ、誤りブロックの始点を先頭方向へ１単位ずらす（この手順は文字の連鎖確率モデルのみ）。

Pos1 ← Pos1 − 1

2　評価実験

2．1　連鎖確率モデルの作成

文字列データベース作成（本文 1.3.1 節参照）に用いたデータを利用して作成した。なお、形態素解析には、形態素解析ツール JUMAN3.1（以下 JUMAN と呼称）を用い、JUMAN の標準辞書１０万語に旅行会話で出現する固有名詞５０００語を追加した。

連鎖確率モデル作成のデータ諸元

発話数	延べ形態素数	異なり形態素数	延べ文字数	異なり文字数
20176	270955	8089	570306	1396

2．2　評価用データ

本文 1.3.1 節に述べた音声認識結果４８０６発話を用いた。

2．3　閾値

誤りの有無に用いた連鎖確率の閾値は、少量データによる予備的な実験を踏まえて、経験的に決定した。

3　実験結果

【誤り検出精度】

各手法の検出精度を下表に示す。なお、再現率については誤り種類別にも計算している。

誤り検出精度

手法	適合率	再現率			
		全体	挿入	脱落	置換
a)　文字　3-gram	84.27	71.89	61.13	53.47	79.28
b)　文字　2-gram	91.36	39.82	22.60	21.56	49.29
c)　品詞・単語混合　3-gram	62.58	30.30	23.13	28.09	33.10
d)　品詞・単語混合　2-gram	71.81	12.43	9.50	11.16	13.65

【正解発話の選別精度】

誤りの有無によって、音声認識結果から正しい文（正解発話と同じ）を選別する精度を下表に示す。

正解発話選別精度

手法	適合率	再現率
a)　文字　3-gram	80.61	86.37
b)　文字　2-gram	51.65	92.37
c)　品詞・単語混合　3-gram	73.52	59.30
d)　品詞・単語混合　2-gram	56.13	65.49

4　考察

以上の実験結果から次のことが明らかになった。

（1）誤りの検出精度は文字の連鎖確率モデルが有効で、適合率と再現率のバランスを考えると文字の3文字連鎖確率モデルがもっとも優れており、適合率80％以上、再現率70以上であった。

（2）また、誤りの有無による音声認識結果から正しい文（正解発話と同じ）を選別する点でも、やはり文字の3文字連鎖確率モデルがもっとも精度が良かった。

（3）品詞・単語混合連鎖確率モデルは形態素解析の過程で誤り文字列が別の単語に解釈されたり、連鎖確率が低くなる位置が誤り位置よりずれて発生するなどして、検出精度はよくない。

付録２　間投詞除去と抽象化の誤り検出位置の違い

間投詞除去と抽象化で誤り検出位置の違う例をまとめた。ANS は正解発話、REZ が認識結果、TPS が真の誤り位置、POS が抽象化なしでの誤り予測位置、OPS が抽象化しての誤り予測位置である。ABS は REZ を抽象化した発話である。

```
ANS=大人二名様と子供二名様で和室を二部屋
REZ=大人二名様と子供を二名様でバスと二部屋
TPS=                  ^^          ^^^^^^
POS=            ****  ****      ****
OPS=            ****              ****
ABS=大人@0 名様と子供を@0 名様でバスと@0 部屋

ANS=十六十七の二泊はツインのお部屋空いておりますが
REZ=十六十七の二泊はツインのおや空いておりますが
TPS=                              ^^
POS=          ****                ****
OPS=                              ****
ABS=@0@0@0@0 の@0 泊はツインのおや空いておりますが

ANS=到着予定は二時なんですが二時には入れないんでしょうか
REZ=到着予定は二一なんですが二時におあ晴れないんでしょうか
TPS=              ^^                    ^^^^^^
POS=**        **********              ******
OPS=**        ****                    ******
ABS=到着予定は@0@0 なんですが@4 時におあ晴れないんでしょうか

ANS=そうですかはい分かりました後それから朝食は和食と洋食とどっちなんでしょうか
REZ=そう通すかわわかりました後おそれから朝食を当和食と洋食と当地なんでしょうか
TPS=    ^^    ^^^^            ^^              ^^^^            ^^^^
POS=  ****    ******          ****            ****            ******
OPS=  ****    ******          ****            ****            ******
ABS=そう通すかわわかりました後おそれから朝食を当和食と洋食と当地なんでしょうか

ANS=はい九月十四日からの四泊五日ですねそれでは何名様でしょうか
REZ=はい九月十四日からの四二泊五日ですねそれではほね様でしょうか
TPS=                        ^^                      ^^^^
POS=                      ****                    ****
OPS=                                              ****
ABS=はい@2 月@3 日からの@0@0 泊@0 日ですねそれではほね様でしょうか

ANS=九月の十四日から二泊大人二名和室が希望なんですが
REZ=九月の十四日から二八区大人二め和室が器具をなんですが
TPS=                    ^^^^      ^^      ^^^^^^
POS=                  ******    ****    ********
OPS=                    ****    ****    ********
ABS=@2 月の@3 日から@0@0 区大人@0 め和室が器具をなんですが

ANS=ではやはり二時に行きますので荷物をお願いします
REZ=ではやはり二時に来ますので荷物をお願いします
TPS=                  ^^
POS=          ******
OPS=
ABS=ではやはり@4 時に来ますので荷物をお願いします

＃字面ではいいのだが、単語単位で誤っている。
ANS=失礼ですがもう一方のお名前をおわかりになりますか
REZ=失礼ですがもう一方のお名前をおお借りなりますか
TPS=                                ^^^^^^^^
POS=                              ************
OPS=                ******        ************
ABS=失礼ですがもう@0 方のお名前をおお借りなりますか
```

付録３　抽象化の有無による訂正の差異例

抽象化の有無による差異をモデルＶ１＊で比較を行った。下記の例で、Ｖ１が間投詞除去での結果、Ｖ２が間投詞除去後に抽象化した結果である。

EXE[2]:いえきょうから二泊お願いしたいんですけれども
ANS[2]:いえきょうから二泊お願いしたいんですけれども
V1 [2]:いえきょうから二泊お願いしたいんですけれども
V2 [2]:きょうから二泊お願いしたいんですけれども

query_word_pat [い<えきょ>うから二泊]
[0.571429]きょうから一泊
[0.500000]きょうから一泊で
---
query_word_pat [い<えきょ>うから@0 泊]
[0.714286]きょうから@0 泊
[0.625000]きょうから@0 泊で

EXE[9]:二百七号しの森山ですけれどもへ滞在の延長お願いしたいのですが
ANS[9]:二百七号室の森山ですけれども滞在の延長をお願いしたいのですが
V1 [9]:二百七号しの森山ですけれどもへ滞在の延長お願いしたいのですが
V2 [9]:二百七号室の森山ですけれどもへ滞在の延長お願いしたいのですが

query_word_pat [二百七<号し>の森山です]
[0.600000]二百七号室の青山なん
[0.600000]八百十七号室の森です
---
query_word_pat [@0 百@0<号し>の@h ですけ]
[0.900000]@0 百@0 号室の@h ですけ
[0.800000]@0@0@0 号室の@h ですけ
[0.800000]@0 百@0 号室の@h ですが
[0.700000]@0@0@0 号室の@h ですが

EXE[26]:六時枚ぐらいには着くと思うんのでビーでお願いします
ANS[26]:六時前ぐらいには着くと思うので六時でお願いします
V1 [26]:六時枚ぐらいには着くと思うんのでビーでお願いします
V2 [26]:六時ぐらいには着くと思うんのでビーでお願いします

query_word_pat [六<時枚>ぐらいには]
[0.375000]いくらぐらいかか
---
query_word_pat [@4<時枚>ぐらいには]
[0.714286]@4 時ぐらいに出
[0.500000]夜の@4 時ぐらいに

EXE[93]:マスターカードで五二七九三九二零二万六九零零九八てす
ANS[93]:マスターカードで五二七九三九二零二四六九零零九八です
V1 [93]:マスターカードで五二七九三九二零二四六九零零九八です
V2 [93]:マスターカードで五二七九三九二零二万六九零零九八です

query_word_pat [九三九二零<二万>六九零零九]
[0.700000]三九二零二四六九零零
[0.700000]九二零二四六九零零九
[0.700000]九三九二零二四六九零

誤り検出せず

EXE[150]:すいません五だ五号室の青山なんですけど嬢からやつかまで四五度お願いしてあるんですけど十七時と十八一の二泊お友達が来るんですけどその日がツインに行いできませんか
ANS[150]:すいません五百五号室の青山なんですけどきょうから二十日までシングルでお願いしてあるんですけど十七日と十八日の二泊友達が来るんですけどその日だけツインにできませんか
V1 [150]:すいません五だ五号室の青山なんですけど嬢からやつかまで四五度お願いしてあるんですけど十七日と十八日の二泊お友達が来るんですけどその日がツインに行いできませんか

V2 [150]:すいません五百五号室の青山なんですけど嬢からやつかまで四五度お願いしてあるんですけど十七時と十八一の二泊お友達が来るんですけどその日がツインに行いできませんか

query_word_pat [すいません<五だ>五号室の青]
[0.500000]いません二百七号室の
[0.500000]ません二百七号室の青
---
query_word_pat [すいません<@0 だ>@0 号室の@h]
[0.700000]いません@0 百@0 号室の
[0.700000]ません@0 百@0 号室の@h
[0.700000]すいません@0@0@0 号室

query_word_pat [十七時と十<八一>の二泊お友]
[0.600000]十七日と十八日の二泊
---
query_word_pat [すけど@4 時<と@0>@0@0 の@0 泊]

EXE[165]:夜の九時くりございますねもしもおその時間よりかなり遅れすうようでしたらあったホテルの方でいご連絡入れていますでしょうか
ANS[165]:夜の九時ごろでございますねもしもそのお時間よりかなり遅れるようでしたらまたホテルの方にご連絡いただけますでしょうか
V1 [165]:夜の九時くりございますねもしもおその時間よりかなり遅れすうようでしたらまたホテルの方でいご連絡入れていますでしょうか
V2 [165]:夜の九時でございますねもしもおその時間よりかなり遅れすうようでしたらまたホテルの方でいご連絡入れていますでしょうか

query_word_pat [夜の九<時くり>ございます]
[0.500000]あいにくでございます
[0.500000]クのリサでございます
---
query_word_pat [夜の@4 時<くり>ございます]
[0.600000]午後@4 時でございます

EXE[175]:ツインルームが一万五千がとります
ANS[175]:ツインルームは一万五千円になっております
V1 [175]:ツインルームが一万五千がとります
V2 [175]:ツインルームが一万五千円となります

query_word_pat [ムが一万五<千がと>ります]
[0.500000]が一万五千円となって
[0.500000]一万五千円になります
---
query_word_pat [ムが@0 万@0<千がと>ります]
[0.600000]@0 万@0 千円となります

EXE[192]:はい住所は京都府ぐ相楽郡精華町光台二の二です電話番号は零う七七四九の五の一三零一です
ANS[192]:はい住所は京都府相楽郡精華町光台二の二です電話番号は零七七四九の五の一三零一です
V1 [192]:はい住所は京都府相楽郡精華町光台二の二です電話番号は零七七四九の五の一三零一です
V2 [192]:はい住所は京都府相楽郡精華町光台二の二です電話番号七四九の五の一三零一です

query_word_pat [電話番号は<零う>七七四九の]
[0.800000]電話番号は零七七四九
[0.800000]話番号は零七七四九の
---
query_word_pat [電話番号は<@0 う@0>@0@0@0 の@0]
[0.700000]電話番号@0@0@0@0@0@0
[0.700000]電話番号は@0@0@0@0@0
[0.700000]電話番号は@0@0@0 の@0
[0.700000]話番号は@0@0@0@0@0@0
[0.700000]話番号は@0@0@0@0@0 の
[0.700000]番号は@0@0@0@0@0 の@0
[0.600000]電話番号が@0@0@0@0@0
[0.600000]電話番号が@0@0@0 の@0
[0.600000]電話番号は@0@0 の@0@0
[0.600000]話番号が@0@0@0@0@0@0
MID Loop=[0], flg=[1], parts=[] :BF=電話番号(5), AF=@0@0@0(8)

WP=はい住所は京都府相楽郡精華町光台@0 の@0 です電話番号@0@0@0 の@0 の@0@0@0@0 です

EXE[197]:きょうから十八日までの九三泊のじょうねがいしたいのですが
ANS[197]:きょうから十八日までの三泊の延長お願いしたいのですが
V1 [197]:きょうから十八日までの三泊のじょうねがいしたいのですが
V2 [197]:きょうから十八日までの九三泊のじょうねがいしたいのですが

query_word_pat [十八日まで<の九>三泊のじょ]
[0.600000]十八日までの三泊でご

誤り検出せず

EXE[208]:かしこまりましたすそれでは本日をご一泊和室でご予約をけどりますをちょうどお部屋の移動願え出しますのですよろしくお願いいたします
ANS[208]:かしこまりましたそれでは本日のご一泊和室でご予約承ります後ほどお部屋の移動をお願いいたしますのでよろしくお願いいたします
V1 [208]:かしこまりましたそれでは本日をご一泊和室でご予約をけどりますをちょうどお部屋の移動願え出しますのですよろしくお願いいたします
V2 [208]:かしこまりましたそれでは本日から一泊和室でご予約をけどりますをちょうどお部屋の移動願え出しますのですよろしくお願いいたします

query_word_pat [それでは本<日を>ご一泊和室]
[0.500000]それでは本日から二泊
---
query_word_pat [それでは本<日を>ご@0 泊和室]
[0.600000]それでは本日から@0 泊

EXE[214]:はいえわたくしとものホテルツインルームはえ一泊二万五千となっております
ANS[214]:はいわたくしどものホテルツインルームは一泊二万五千円となっております
V1 [214]:はいわたくしどものホテルツインルームはえ一泊二万五千となっております
V2 [214]:はいわたくしどものホテルツインルームが二万円となっております

query_word_pat [インルーム<はえ一>泊二万五千]
[0.500000]インルームが一万四千
---
query_word_pat [インルーム<はえ@0>泊@0 万@0 千]
[0.600000]インルームが@0 万@0 千

EXE[235]:はい大阪観光ホテルと土に取でございます
ANS[235]:はい大阪観光ホテルフロントでございます
V1 [235]:はい大阪観光ホテルでございます
V2 [235]:はい大阪観光ホテルと土に取でございます

query_word_pat [観光ホテル<と土に取>でございま]
[0.600000]観光ホテルでございま
---
query_word_pat [はい@k<と@1 に取>でございま]
[0.500000]い@k 予約係でございま
[0.500000]はい@k@k でございます
[0.500000]はい@k 予約係でござい

EXE[280]:そうある一四号室のをおただと申しますけど
ANS[280]:三一四号室の大竹と申しますけども
V1 [280]:そうある一四号室の鈴木と申しますけど
V2 [280]:そうある一四号室のをおただと申しますけど

query_word_pat [号室のをお<ただ>と申します]
[0.600000]号室の鈴木と申します
MID Loop=[0], flg=[1], parts=[鈴木] :BF=号室の(4), AF=と申します(7)
---
query_word_pat [号室のをお<ただ>と申します]
[0.600000]号室の@h@h と申します

MID Loop=[0], flg=[1], parts=[@h@h] :BF=号室の(4), AF=と申します(7)

EXE[314]:横須賀市上町六の十二の二十六様らです
ANS[314]:横須賀市上町六の十二の二十六田村です
V1 [314]:横須賀市上町六の十二の二十六です
V2 [314]:横須賀市上町六の十二の二十六様らです

query_word_pat [十二の二十<六様>らです]
[0.600000]六の十二の二十六です
---
query_word_pat [@0@0 の@0 十<@0 様>らです]
[0.700000]@0@0 の@0@0@0@0 番です
[0.600000]@0@0@0@0@0@0@0@0 です
[0.600000]@0@0@0@0@0@0@0 番です
[0.600000]@0@0@0 の@0@0@0@0 です
[0.600000]@0@0 の@0@0@0@0 ですね
[0.600000]@0 の@0@0 の@0 十@0 です
[0.600000]@0 の@0 の@0@0@0@0 です
[0.500000]が゜@0@0@0@0@0@0@0 です
[0.500000]が゜@0 千@0 百@0 十円です
[0.500000]@0@0@0@0@0@0@0 ですね
MID Loop=[0], flg=[1], parts=[@0@0@0 番] :BF=@0@0 の@0(5), AF=です(8)

EXE[317]:二分で結構ですので
ANS[317]:普通郵便で結構ですので
V1 [317]:片道で結構ですので
V2 [317]:二分で結構ですので

query_word_pat [<二分>で結構です]
[0.714286]片道で結構です

誤り検出せず

EXE[324]:そういたしますと八て九十六号室熱すアメリカ人を男性が一人でまてるはずなんですが
ANS[324]:そういたしますと八百十六号室アメリカ人の男性が一人で泊まってるはずなんですが
V1 [324]:そういたしますと八百十六号室熱すアメリカ人を男性が一人でまてるはずなんですが
V2 [324]:そういたしますと八て九十六号室熱すアメリカ人を男性が一人でまてるはずなんですが

query_word_pat [たしますと<八て九>十六号室熱]
[0.600000]しますと八百十六号室
---
query_word_pat [たしますと<@0 て@0>@0@0 号室熱]
[0.600000]しますと@0 百@0@0 号室
[0.600000]たしますと@0 百@0@0 号
[0.500000]たしますと@0 万@0 千@0
[0.400000]いたしますと@0 百@0@0
MID Loop=[0], flg=[1], parts=[@0 百] :BF=しますと(6), AF=@0@0 号室(8)
MID Loop=[1], flg=[1], parts=[@0 百] :BF=たしますと(6), AF=@0@0 号(8)

EXE[362]:ありがとうざいますうフロントで下でございまそれ
ANS[362]:ありがとうございますフロントの石川でございます
V1 [362]:ありがとうございますフロントでございますが
V2 [362]:ありがとうございますフロントでございます

EXE[425]:四百五号室の諸岡案様ですねかしこまりました税ではすな様でしれておきますので
ANS[425]:四百五号室の村岡様ですねかしこまりましたそれではあすの朝までに仕上げておきますので
V1 [425]:四百五号室の諸岡案様ですねかしこまりましたではすな様でしれておきますので
V2 [425]:四百五号室の諸岡様ですねかしこまりましたではすな様でしれておきますので

query_word_pat [四百五号室<の諸岡案>様ですねか]
---
query_word_pat [百@0 号室の<@h 案>様ですねか]
[0.800000]百@0 号室の@h 様ですね

```
EXE[466]:すいません客様のお部屋番号と名前を教えてください
ANS[466]:すいませんお客様のお部屋番号とお名前教えて下さい
V1 [466]:すいませんお客様のお部屋番号と名前を教えてください
V2 [466]:すいません客様のお部屋番号と名前を教えてください

query_word_pat [すいませ<ん客様>のお部屋番]
[0.600000]いませんお客様のお部
------------------------------------
query_word_pat [すいませ<ん客様>のお部屋番]
[0.700000]いません@h 様のお部屋
[0.600000]ざいません@h 様のお部
[0.600000]いませんお客様のお部
[0.600000]ませんお客様のお部屋
[0.600000]ません@h 様のお部屋に
[0.600000]すいませんお客様のお
[0.600000]すいません隣の部屋が
[0.500000]いませんお客様のお名
[0.500000]すぐお客様のお部屋に
[0.400000]ございません@h 様のお
MID Loop=[0], flg=[1], parts=[ん@h 様] :BF=いませ(5), AF=のお部屋(7)
```

付録4　SSC、V1バージョンとV2バージョンとの差異例

##v1=/data/as37/skaki/sreg/mong/abst1/MC.02.V1.SSC
##v2=/data/as37/skaki/sreg/mong/hreq/result/test04.1000.L6

EXE[334]:はい八百十な申しのほう山様で五様すねそれはたいえ申し訳ございませんそれでは今から抱えの者がそちらにお変わいますのでそのもおもの街ください
ANS[334]:はい八百十七号室の青山様でございますねそれは大変申し訳ございませんそれでは今から係の者がそちらに伺いますのでそのままでお待ちください
V1 [334]:はい八百十な申しのほう山様ですねそれは大変申し訳ございませんそれでは今から係の者がそちらに伺いますのでそのもおもの街ください
V2 [334]:はい八百十な申しのほう山様ですねそれは大変申し訳ございませんそれでは今から係の者がそちらにお伺いいたしますのでそのままでお待ちください

query_word_pat [でそのもお<もの街>ください]
[0.400000]でもお電話くださいお
[0.400000]つでもお電話ください
ーーーーーーーーーーーーー
query word pat [はい@0 百@0 な申しのほう山様ですねそれは大変申し訳ございませんそれでは今から係の者がそちらにお伺いいたしますのでそのもお<もの街>ください] ...
pre=はい/@0 百@0/な/申し/のほう/山/様で/すね/それは/大/変/申し訳ございません/それでは/今/から/係/の/者/が/そちら/にお/伺い/いたします/ので/その/も/お
post=ください
before pos1=60, pos2=62
pre=その||はい@0 百@0 な申しのほう山様ですねそれは大変申し訳ございませんそれでは今から係の者がそちらにお伺いいたしますので
post=ください||
err_block=もおもの街
(39)[0.882353](0)|1|=(それは大変申し訳ございませんそれでは今から係の者がそちらにお伺いいたしますので)[その]<ままでお待ち>[ください]()
(21)[0.818182](0)|1|=(今から係の者がそちらにお伺いいたしますので)[その]<ままでお待ち>[ください]()
(7)[0.722222](0)|3|=(いたしますので)[その]<ままお待ち>[ください]()
(7)[0.684211](0)|1|=(いたしますので)[その]<ままでお待ち>[ください]()
(4)[0.666667](0)|3|=(ますので)[その]<ままお待ち>[ください]()
(4)[0.625000](0)|5|=(ますので)[その]<ままでお待ち>[ください]()
(4)[0.625000](0)|2|=(ますので)[その]<ようにお届け>[ください]()
(4)[0.588235](0)|1|=(ますので)[その]<看板の前に来て>[ください]()
(4)[0.588235](0)|1|=(ますので)[その]<営業所でお支払>[ください]()
(0)[0.583333](0)|1|=()[その]<つもりでいて>[ください]()
TRY cand No = [0]
HF=はい@0 百@0 な申しのほう山様ですねそれは大変申し訳ございませんそれでは今から係の者がそちらにお伺いいたしますのでそのままでお待ちください

EXE[164]:チェックインはだいたい何時ごろうご予定されておりますか
ANS[164]:チェックインはだいたい何時ごろご予定されておりますか
V1 [164]:チェックインは大体何時ごろご予定されておりますか
V2 [164]:チェックインしていたい何時ごろをご予定されておりますか

query_word_pat [エックイン<はだ>いたい何時]
[0.600000]エックインは大体何時
[0.500000]のほうはだいたい何時
[0.500000]エックインといいます
[0.500000]エックインいたします
[0.500000]エックインですが何時
[0.500000]エックインしたいと思
[0.500000]エックインしたいんで
[0.500000]エックインしたいので
[0.500000]エックインしていただ
[0.500000]エックインはできない
MID Loop=[0], flg=[1], parts=[は大体] :BF=エックイン(6), AF=何時(10)
WP=チェックインは大体何時ごろうご予定されておりますか

ーーーーーーーーーーーーーーーーーー
query word pat [チェックイン<はだ>いたい何時ごろうご予定されておりますか] ...
pre=チェックイン

post=いた/い/何時/ご/ろ/うご/予/定/さ/れて/おります/か
before pos1=7, pos2=8
pre=チェックイン||
post=いた||い何時ごろうご予定されておりますか
err_block=はだ
(0)[0.800000](0) {2}=()[チェックイン]<して>[いた]()
(0)[0.750000](0) {1}=()[チェックイン]<>[いた]()
(0)[0.727273](0) {1}=()[チェックイン]<をして>[いた]()
(0)[0.727273](0) {5}=()[チェックイン]<させて>[いた]()
(0)[0.714286](5) {1}=()[チェックイン]<のご予定時間はだ>[いた](い何時ごろ)
(0)[0.615385](0) {1}=()[チェックイン]<を早くして>[いた]()
(0)[0.608696](5) {2}=()[チェックイン]<のお時間なんですがだ>[いた](い何時ごろ)
(0)[0.571429](0) {1}=()[チェックイン]<の時間教えて>[いた]()
(0)[0.538462](5) {1}=()[チェックイン]<のお時間なんですけれどもだ>[いた](い何時ごろ)
(0)[0.533333](0) {3}=()[チェックイン]<の時間を教えて>[いた]()
TRY cand No = [0]
HF=チェックインしていたい何時ごろうご予定されておりますか

●誤りの検出精度
●意味のある区切り文字

==========
<（S+）と（S*）の差の具体例>
6文の違い
良くなった 2例
どっちもどっち 3例
悪くなった 1例（誤りではなく正解により近いということ）

##v1=/data/as37/skaki/sreg/mong/hreq/result/test03.1000.L6
##v2=/data/as37/skaki/sreg/mong/hreq/result/test04.1000.L6
##DP range=0 <--> 100
EXE[46]:おたせいたしましたってえ十四日十五日ご二泊は一千の方一ございますね
ANS[46]:お待たせいたしました十四日十五日二泊はいツインの方空きがございますが
V1 [46]:お待たせいたしましたところ十四日十五日ご二泊一名様でございますね
V2 [46]:お待たせいたしました十四日十五日ご二泊一名様でございますね

query word pat [お待たせいたしましたっ<てえ>@3 日@3 日ご@0 泊は@0 千の方@0 ございますね] ...
pre=お待/た/せ/いたしました/っ
post=@3 日/@3 日/ご/@0 泊/は@0/千/の方/@0/ございますね
before pos1=12, pos2=13
pre=いたしました||お待たせ
post=@3 日||@3 日ご@0 泊は@0 千の方@0 ございますね
err_block=ってえ
(4)[0.800000](0) {1}=(お待たせ)[いたしました]<@3 日と>[@3 日]()
(4)[0.785714](0) {1}=(お待たせ)[いたしました]<@3 日>[@3 日]()
(4)[0.785714](0) {5}=(お待たせ)[いたしました]<@2 月>[@3 日]()
(4)[0.750000](0) {1}=(お待たせ)[いたしました]<@3 日から>[@3 日]()
(0)[0.727273](0) {1}=()[いたしました]<ところ>[@3 日]()
(4)[0.705882](0) {1}=(お待たせ)[いたしました]<@h@h 様あす>[@3 日]()
(0)[0.700000](0) {1}=()[いたしました]<@2 月>[@3 日]()
(4)[0.666667](0) {1}=(お待たせ)[いたしました]<それでは@2 月>[@3 日]()
(4)[0.666667](0) {1}=(お待たせ)[いたしました]<こちらが@2 月>[@3 日]()
(4)[0.631579](0) {1}=(お待たせ)[いたしました]<あいにくですが>[@3 日]()
TRY cand No = [0]
TRY cand No = [1]
TRY cand No = [2]
TRY cand No = [3]
TRY cand No = [4]
HF=お待たせいたしましたところ@3 日@3 日ご@0 泊は@0 千の方@0 ございますね
----
query word pat [お待たせいたしましたっ<てえ>@3 日@3 日ご@0 泊は@0 千の方@0 ございますね] ...
pre=お待/た/せ/いたしました/っ
post=@3 日/@3 日/ご/@0 泊/は@0/千/の方/@0/ございますね
before pos1=12, pos2=13
pre=いたしました||お待たせ
post=@3 日||@3 日ご@0 泊は@0 千の方@0 ございますね

```
err_block=ってえ
(4)[0.800000](0)|1|=(お待たせ)[いたしました]<@3 日と>[@3 日]()
(4)[0.785714](0)|1|=(お待たせ)[いたしました]<@3 日>[@3 日]()
(4)[0.785714](0)|5|=(お待たせ)[いたしました]<@2 月>[@3 日]()
(4)[0.750000](0)|2|=(お待たせ)[いたしました]<>[@3 日]()
(4)[0.750000](0)|1|=(お待たせ)[いたしました]<@3 日から>[@3 日]()
(0)[0.727273](0)|1|=()[いたしました]<ところ>[@3 日]()
(4)[0.705882](0)|1|=(お待たせ)[いたしました]<@h@h 様あす>[@3 日]()
(0)[0.700000](0)|1|=()[いたしました]<@2 月>[@3 日]()
(4)[0.666667](0)|1|=(お待たせ)[いたしました]<それでは@2 月>[@3 日]()
(4)[0.666667](0)|1|=(お待たせ)[いたしました]<こちらが@2 月>[@3 日]()
TRY cand No = [0]
TRY cand No = [1]
TRY cand No = [2]
TRY cand No = [3]
HF=お待たせいたしました@3 日@3 日ご@0 泊は@0 千の方@0 ございますね
```

付録5　SSCの形態素化による影響

下記の例でV1が文字単位、V2が形態素単位での訂正結果である。

```
##v1=/data/as37/skaki/sreg/mong/hreq/result/test04.1000.L7
##v2=/data/as37/skaki/sreg/mong/hreq/result/test06.1000.L7
##DP range=0 <--> 60
EXE[10]:きょうかですね十八日までの三日間隔の延長お願いしたいのですが
ANS[10]:きょうからですね十八日までの三泊の延長をお願いしたいのですが
V1 [10]:きょうぶですね十八日までの三日間隔の延長お願いしたいのですが
V2 [10]:きょうかですね十八日までの三日間隔の延長お願いしたいのですが

EXE[42]:はいありがとうではすれいたしいますのは一なりますでしょうか
ANS[42]:はいありがとうございますいらっしゃいますのはいつになりますでしょうか
V1 [42]:はいありがとうでは失礼いたしいますのは一なりますでしょうか
V2 [42]:はいありがとうでは確認いたしますのは一なりますでしょうか

EXE[80]:かまいません忘れは寝いたします
ANS[80]:構いません和室でお願いいたします
V1 [80]:かまいませんそれでは失礼いたします
V2 [80]:かまいませんお願いいたします

EXE[105]:二万二製の方はデラックスタイプとなっておりまして少々部屋が広めなとります
ANS[105]:二万一千円の方はデラックスタイプとなっておりまして少々お部屋が広めになっております
V1 [105]:二万二製の方はファックスタイプとなっておりまして少々部屋が広めなとります
V2 [105]:二万二製の方はデラックスタイプとなっておりまして少々部屋が広めなとります

EXE[148]:もうしゃございませ朝のいい日英ツインルーム枚室など折れまして会いご用意できるのあいシングルルームだけなりますが
ANS[148]:申し訳ございませんそのお日にちツインルーム満室になっておりましてご用意できるのはシングルルームだけになりますが
V1 [148]:もうございますお願いい日英ツインルーム枚室など折れましてご用意できるのあいシングルルームだけなりますが
V2 [148]:もうございませ朝のいいですツインルーム枚室など折れましてご用意できるのあいシングルルームだけなりますが

EXE[157]:ただいま空室状況調べしますところふ十四日のご一泊和室がワンルームは空きが有れだけです二部屋のご予約行きませは十五でしに関しては和室の方はまったくお取りできない条件でお取ります
ANS[157]:ただいま空室状況をお調べしましたところ十四日のご一泊和室がワンルーム空きがあるだけで二部屋のご用意ができません十五日に関しては和室の方はまったくお取りできない状況になっております
V1 [157]:ただいま空室状況調べしますところ十四日のご一泊和室がワンルームは空きが有れだけですよ部屋のご予約行きませは十五でしに関しては和室の方はまったらお取りできない条件でお取ります
V2 [157]:ただいま空室状況調べしますところ十四日のご一泊和室がワンルームは空きが有れだけですよ部屋のご予約行きませは十五でしに関しては和室の方はまったくお取りできない条件でお取ります

EXE[159]:お部屋をお二部屋別々にお取りするということでしたがすシングルルーム二部屋ということでよろしいでしょうか
ANS[159]:お部屋を二部屋別々にお取りするということでしたらシングルルーム二部屋ということでよろしいでしょうか
V1 [159]:お部屋をお二部屋別々にお取りするということでしたらシングルルーム二部屋ということでよろしいでしょうか
V2 [159]:お部屋をお二部屋別々にお取りするということでしたがすシングルルーム二部屋ということでよろしいでしょうか

EXE[160]:かしこまりましたそうしましたら十四日から二泊すツインルー吸いませシングルルームおおくあ二部屋ずつ二泊お取りいたしますお客様のお名前お願いいたします
ANS[160]:かしこまりましたそうしましたら十四日から二泊すいませんシングルルームを二部屋ずつ二泊お取りいたしますお客様のお名前をお願いいたします
V1 [160]:かしこまりましたそうしましたら十四日から二泊はツインルー吸いまシングルルームおおくあ二部屋ずつお取りいたしますお客様のお名前お願いいたします
V2 [160]:かしこまりましたそうしましたら十四日から二泊すツインルー吸いまシングルルームおおくあ二部屋ずつお取りいたしますお客様のお名前お願いいたします

EXE[164]:チェックインはだいたい何時ごろうご予定されておりますか
ANS[164]:チェックインはだいたい何時ごろご予定されておりますか
V1 [164]:チェックインしていたい何時ごろをご予定されておりますか
V2 [164]:チェックインのご予定時間はだいたい何時ごろをご予定されておりますか
```

EXE[173]:本日からの三泊の追加ですねご希望はツインルームだんでしょうか
ANS[173]:本日からの三泊の追加ですねご希望はツインルームなんでしょうか
V1 [173]:本日からの三泊の追加ですねお部屋はツインルームがご希望でしょうか
V2 [173]:本日からの三泊の追加ですねご希望はツインルームがご希望でしょうか

EXE[174]:大事申し訳ないんですがす混んでてはツインルームが毎週がとりますみょうにち料後に時十六日十七日のをお二泊に関してはツインルームでも取りできますが
ANS[174]:大変申し訳ないんですが本日はツインルームが満室になっておりますみょうにち明後日十六日十七日の二泊に関してはツインルームでもお取りできますが
V1 [174]:大変申し訳ないんですがそれではツインルームが毎週がとりますみょうにち料後に時十六日十七日のご二泊に関してはツインルームでも取りできますが
V2 [174]:大変申し訳ないんですがす混んですがツインルームが毎週がとりますみょうにち料後に時十六日十七日のご二泊に関してはツインルームでも取りできますが

EXE[197]:きょうから十八日までの九三泊のじょうねがいしたいのですが
ANS[197]:きょうから十八日までの三泊の延長お願いしたいのですが
V1 [197]:きょうから十八日までの九三泊のじょうだいしたいのですが
V2 [197]:きょうから十八日までの九三泊のじょうねがいしたいのですが

EXE[214]:はいえわたくしとものホテルツインルームはえ一泊二万五千となっております
ANS[214]:はいわたくしどものホテルツインルームは一泊二万五千円となっております
V1 [214]:はいわたくしどものホテルのツインルームは一泊二万五千円となっております
V2 [214]:はいわたくしどものホテルツインルームは一泊二万五千円となっております

EXE[226]:フロントに荷物を混んで最寄りに行ったんですけれどもに入ったソースなくなってたんですが
ANS[226]:フロントに荷物を運んでもらうように言ったんですけれども部屋に入ったら一つ無くなってたんですが
V1 [226]:フロントに荷物を混んで最寄りに行ったんですけれどもたったソースにはなってたんですが
V2 [226]:フロントに荷物を混んで最寄りに行ったんですけれどもに入ったソースにはなってたんですが

EXE[237]:もしもし五されませんがあすエイチ一区祝いいしいくらになっ取りましてお急いきでしょうか
ANS[237]:申し訳ございませんがチェックインは一時からになっておりましてお急ぎでしょうか
V1 [237]:もしもしれませんがあすエイチ一区祝いいしいくらになっておりましてお急ぎでしょうか
V2 [237]:もしもし五されませんがあすエイチ一区祝いいしいくらになっておりましてお急ぎでしょうか

EXE[267]:大至急ねがいします
ANS[267]:じゃ大至急お願いします
V1 [267]:大至急ねがいいたします
V2 [267]:大至急ねがいします

EXE[360]:大変申し訳ありませんが薬の方は医者も処方せんがなければお出しできないんですねホテルには医務室がりましてそこに嬢りますのでそちらえ借りてはどうでしょうか
ANS[360]:大変申し訳ありませんが薬の方は医者の処方せんがなければお出しできないんですね当ホテルには医務室がありましてそこに医者がおりますのでそちらへ行かれてはどうでしょうか
V1 [360]:大変申し訳ありませんがお薬の方は医者の処方せんがなければお出しできないんですねホテルには医務室がありましてそこに嬢りますのでそちらえ借りてはどうでしょうか
V2 [360]:大変申し訳ありませんこちらの方は医者の処方せんがなければお出しできないんですねホテルには医務室がありましてそこに嬢りますのでそちらえ借りてはどうでしょうか

EXE[364]:はい二才の方法でれべた降りていてでたところすでにビジネストコーナーて今ございますうそちらに行って痛ければ分かりなると思いますが
ANS[364]:はいっ二階の方のエレベーター下りて頂いたところすぐにビジネストコーナーというものがございますそちらに行っていただければお分かりになると思いますが
V1 [364]:はい二才の方法でれべた降りていてでたところすでにビジネストコーナーもございますがそちらに行って痛ければ分かりになると思いますが
V2 [364]:はい二才の方法でれべた降りていてでたところすでにビジネストコーナーて今ございますがそちらに行って痛ければ分かりになると思いますが

EXE[376]:分かりましたお世話さまです
ANS[376]:わかりましたお世話様です
V1 [376]:分かりましたどうもお世話さまです
V2 [376]:分かりましたお世話さまです

EXE[409]:はいかしこまりましたするですでお部屋の番号と生頂けませしょうか
ANS[409]:はいかしこまりました失礼ですがお部屋の番号とお名前いただけますでしょうか
V1 [409]:はいかしこまりましたカードですがお部屋の番号とお名前を頂けますでしょうか
V2 [409]:はいかしこまりましたカードですかお部屋の番号の方頂けますでしょうか

付録６　語順並び替えによる効果の具体例

（１）良くなった例

認識結果：二百七号室の森山様でございますねくいたかまいちょごじごということなんですけれどもお部屋のタイプのご希望はございますでしょうか
正解発話：二百七号室の森山様でございますねご一泊の延長ご希望ということなんですけれどもお部屋のタイプのご希望はございますでしょうか
並替え前：二百七号室の森山様でございますねくいたかまいちょごじごということなんですけれども部屋のタイプの希望はございますでしょうか
並替え後：二百七号室の森山様でございますねくいたかまいちょごじごということなんですけれどもお部屋のタイプの希望はございますでしょうか

◆誤り検出の精度の問題
　　並替え前：query_word_pat［すけれども<お部>屋のタイプ］
　　並替え後：誤り検出なし

認識結果：かしこまりましたツインの部屋中かつとかはいかまきございます
正解発話：かしこまりましたツインのお部屋十月十日はい空きがございます
並替え前：かしこまりましたので部屋中かつとかはいかまきございます
並替え後：かしこまりましたツインの部屋中かつとかはいかまきございます

◆誤り検出の精度の若干の差で、条件に適合する類似文字列を検索できたかどうかの違い
　　並替え前：
　　query_word_pat［まりました<ツイ>ンの部屋中］
　　[0.500000]まりましたので部屋で
　　[0.400000]決まりましたので部屋
　　並替え後：
　　query_word_pat［こまりまし<たツイ>ンの部屋中］

認識結果：見つかりしたいわたしの方で連絡してください
正解発話：見つかりしたいわたしのほうに連絡してください
並替え前：見つかりしたいわたしの方につけてください
並替え後：見つかりしたいわたしの方まで連絡してください

◆並び替えによる効果
　　並替え前：
　　query_word_pat［わたしの方<で連>絡してくだ］
　　[0.500000]でしたら連絡してくだ
　　[0.500000]わたしの方につけてく
　　[0.500000]たしの方につけてくだ
　　並替え後：
　　query_word_pat［わたしの方<で連>絡してくだ］
　　[0.600000]しの方まで連絡してく
　　[0.600000]わたしの方まで連絡し
　　[0.600000]たしの方まで連絡して

　　＜関係する並び替え発話＞
　　では彼女が部屋にわたしの方まで電話で連絡してくれるように伝えてください
　　では彼女が部屋に電話でわたしの方まで連絡してくれるように伝えてください
　　では彼女がわたしの方まで部屋に電話で連絡してくれるように伝えてください
　　では彼女がわたしの方まで電話で部屋に連絡してくれるように伝えてください
　　では彼女が電話で部屋にわたしの方まで連絡してくれるように伝えてください
　　では彼女が電話でわたしの方まで部屋に連絡してくれるように伝えてください
　　では部屋に彼女がわたしの方まで電話で連絡してくれるように伝えてください
　　では部屋に彼女が電話でわたしの方まで連絡してくれるように伝えてください
　　では部屋にわたしの方まで彼女が電話で連絡してくれるように伝えてください
　　では部屋にわたしの方まで電話で彼女が連絡してくれるように伝えてください
　　では部屋に電話で彼女がわたしの方まで連絡してくれるように伝えてください

では部屋に電話でわたしの方まで彼女が連絡してくれるように伝えてください
ではわたしの方まで彼女が部屋に電話で連絡してくれるように伝えてください
ではわたしの方まで彼女が電話で部屋に連絡してくれるように伝えてください
ではわたしの方まで部屋に彼女が電話で連絡してくれるように伝えてください
ではわたしの方まで部屋に電話で彼女が連絡してくれるように伝えてください
ではわたしの方まで電話で彼女が部屋に連絡してくれるように伝えてください
ではわたしの方まで電話で部屋に彼女が連絡してくれるように伝えてください
では電話で彼女が部屋にわたしの方まで連絡してくれるように伝えてください
では電話で彼女がわたしの方まで部屋に連絡してくれるように伝えてください
では電話で部屋に彼女がわたしの方まで連絡してくれるように伝えてください
では電話で部屋にわたしの方まで彼女が連絡してくれるように伝えてください
では電話でわたしの方まで彼女が部屋に連絡してくれるように伝えてください
では電話でわたしの方まで部屋に彼女が連絡してくれるように伝えてください

認識結果：十時な便がなんで八時までには空港について点ですホテルをおいつで出たら間に合でしょうか
正解発話：十時の便なので八時までには空港に着いていたいのですホテルをいつ出たら間に合うでしょうか
並替え前：十時な便がなんで八時までには空港について点ですホテルをおいつで出たらいいでしょうか
並替え後：十時な便がなんで八時までには空港について点ですホテルを何時に出れば間に合うでしょうか

並替え前：

query_word_pat［すホテルを<おいつ>で出たら間］

並替え後：

query_word_pat［ホテルをお<いつ>で出たら間］
[0.500000]ホテルを何時に出たら
[0.300000]ホテルを出たらいいか
[0.300000]にホテルを出たらいい
[0.300000]時にホテルを出たらい
[0.200000]何時にホテルを出たら
MID Loop=[0], flg=[1], parts=[何時に] :BF=ホテルを(5), AF=出たら(8)
WP=@4 時な便がなんで@4 時までには空港について点ですホテルを何時に出たら間に合でしょうか
query_word_pat［@4<時な便>がなんで@4］
query_word_pat［な便がなん<で@4>時までには］
query_word_pat［でには空港<につ>いて点です］
query_word_pat［空港につい<て点で>すホテルを］
query_word_pat［何時に出た<ら間>に合でしょ］
[0.500000]何時に出れば間に合い
[0.400000]を何時に出れば間に合
MID Loop=[0], flg=[1], parts=[れば間] :BF=何時に出(5), AF=に合(7)
WP=@4 時な便がなんで@4 時までには空港について点ですホテルを何時に出れば間に合でしょうか

＜関連する並び替え発話＞

フライトにちゃんと間に合うように何時にホテルを出たらいいかわからないんです
フライトにちゃんと間に合うようにホテルを何時に出たらいいかわからないんです

（2） 悪くなった例

認識結果：大変申し訳ありませんが薬の方は医者も処方せんがなければお出しできないんですねホテルには医務室がりましてそこに嬢りますのでそちらえ借りてはどうでしょうか
正解発話：大変申し訳ありませんが薬の方は医者の処方せんがなければお出しできないんですね当ホテルには医務室がありましてそこに医者がおりますのでそちらへ行かれてはどうでしょうか
並替え前：大変申し訳ありませんが薬の方は医者も処方せんがなければお出しできないんですねホテルには医務室がりまして左側に在りますのでそちらえ借りてはどうでしょうか
並替え後：大変申し訳ありませんが宿泊の方は医者も処方せんがなければお出しできないんですねホテルには医務室がりまして作りますのでそちらえ借りてはどうでしょうか

並替え前：

query_word_pat［ありません<が薬>の方は医者］

[0.600000]ありませんがその便は
[0.500000]ありませんがご利用は
[0.500000]ありませんが@k は窓を
[0.500000]ありませんがそのよう
[0.500000]ありませんが足のサイ
[0.500000]りませんがその便は満
[0.400000]訳ありませんが@k は窓
[0.400000]訳ありませんがそのよ
[0.400000]訳ありませんがその便
[0.400000]訳ありませんが足のサ

並替え後：

query_word_pat [ありません<が薬>の方は医者]
[0.600000]ありませんがその便は
[0.500000]ありませんがご利用は
[0.500000]ありませんが@k は窓を
[0.500000]ありませんがそのよう
[0.500000]ありませんが宿泊の方
[0.500000]ありませんが足のサイ
[0.500000]りませんがその便は満
[0.400000]訳ありませんが@k は窓
[0.400000]訳ありませんがそのよ
[0.400000]訳ありませんがその便
MID Loop=[4], flg=[1], parts=[が宿泊] :BF=ありません(6), AF=の方(7)
WP=大変申し訳ありませんが宿泊の方は医者も処方せんがなければお出しできないんですねホテルには医務室がりましてそこに嬢りますのでそちらえ借りてはどうでしょうか

＜関連する並び替え発話＞

申し訳ありませんがサウナとジムの方は宿泊の方に限らせていただいております
申し訳ありませんが宿泊の方にサウナとジムの方は限らせていただいております