

TR-IT-0339

テキストデータを使った音声認識誤り訂正

石川 開
Kai Ishikawa

隅田 英一郎
Eiichiro Sumita

2000 年 2 月 16 日

音声翻訳における認識誤りに起因する翻訳品質の劣化を解決するために、誤り訂正手法を提案しその有効性を実験的に検証した。すでに、認識結果の正解部分のみを翻訳する手法が提案されているが、不正解部分に対応する入力発声に存在した情報が失われたままであるという問題が存在する。そこで我々は、認識結果中の誤り部分を元に、音韻的に類似した用例をテキストデータから検索し、類似部分に基づき正解を推定することによって誤り訂正を試みる。提案する誤り訂正手法は次の手順からなる。(1) 構成素境界解析によって得られる意味的距離に従って、訂正の必要性を判断し、誤り部分を同定する。(2) 誤り部分を含む訂正対象部分に音韻的に近い断片をテキストデータ中から抽出し、差分の置換によって訂正候補を生成する。(3) 訂正候補の妥当性を、意味的距離を用いて判断する。提案手法の有効性を実験的に検証するために、旅行会話を対象とした日英、日韓、英日方向の翻訳システムを用いて評価を行った。日英翻訳結果では、単語誤り率の 2.3% の減少、翻訳率の 5.4% の改善が得られた。日韓、英日でも同様に誤り訂正の有効性が検証され、言語に依存しない有効性が確認された。

目次

1	はじめに	1
2	提案手法	2
2.1	処理の流れ	2
2.2	構文解析および意味的距離計算	5
2.3	テキストコーパスからの用例検索	5
2.4	音韻的距離の計算	5
2.5	訂正候補の妥当性判断	6
3	日英音声翻訳における誤り訂正実験	8
3.1	実験条件	8
3.2	訂正判断の精度	8
3.3	DP マッチングによる単語誤り率	9
3.4	訂正前後での単語誤り率	9
3.5	テストセット全体に対する単語誤り率	11
3.6	翻訳文の主観評価	13
3.7	翻訳結果の単語誤り率に基づく評価	14
4	日韓音声翻訳における誤り訂正実験	17
4.1	実験条件	17
4.2	訂正判断の精度	17
4.3	訂正前後での単語誤り率	17
4.4	翻訳結果の文字誤り率に基づく評価	19
5	英日音声翻訳における誤り訂正実験	21
5.1	実験条件	21
5.2	訂正判断の精度	21
5.3	訂正前後での単語誤り率	21
5.4	翻訳結果の文字誤り率に基づく評価	23
6	オープンデータベースでの日英音声翻訳実験	24
6.1	実験条件	24
6.2	訂正判断の精度	24
6.3	訂正前後での単語誤り率	24
6.4	翻訳文の主観評価	26

6.5	翻訳結果の単語誤り率に基づく評価	27
6.6	オープンデータベースにおける問題	28
6.7	データベースのオープン性	29
7	おわりに	30
	参考文献	31
	付録 A 音声認識誤り回復実験ツール	33
A.1	概要	33
A.2	ディレクトリ構成	33
A.3	設定方法	35
(A.3.1)	環境変数の設定	35
(A.3.2)	TDMT インターフェースの設定	35
(A.3.3)	コンフィギュレーションファイルの設定	36
A.4	操作方法	36
(A.4.1)	音声認識誤り回復処理クライアント	36
(A.4.2)	音声認識誤り回復処理サーバ	36
A.5	実装説明	37
(A.5.1)	音声認識誤り回復処理サーバ	37
(A.5.2)	音声認識誤り回復処理サーバのイベント仕様	37
(A.5.3)	外部処理	42
(A.5.4)	多言語対応	42
	付録 B 設定ファイル	44
B.1	回復処理サーバ側設定ファイル	44
(B.1.1)	解説	44
(B.1.2)	具体例	46
B.2	回復処理クライアント側設定ファイル	51
(B.2.1)	解説	51
(B.2.2)	具体例	53
	付録 C 辞書フォーマット	54
C.1	マスター辞書	54
C.2	音素ラベル辞書	54
C.3	形態素列バンク	55

付録 D 実験ツール使用例	57
D.1 回復処理サーバ起動	57
D.2 回復処理クライアント起動	61
付録 E デモシステム	63
E.1 概要	63
E.2 操作方法	63
(E.2.1) インストール	63
(E.2.2) 設定	63
(E.2.3) 起動および終了方法	65
(E.2.4) 操作方法	65
E.3 実装説明	66
(E.3.1) プログラムファイル	67
(E.3.2) 設定ファイル	67
(E.3.3) サブプロセスの呼び出し	67
付録 F 評価用ツール	69
F.1 目的	69
F.2 起動方法	69
F.3 関連ファイル	70
F.4 入出力ファイル仕様	70
F.5 入出力ファイル具体例	72

1 はじめに

近年の音声認識、および機械翻訳の性能向上に伴い、これらの統合である音声翻訳システムの実現を目指した研究活動が活発に行われている (Waibel 1996)(Sumita, Yamada, Yamamoto, Paul, Kashioka, Ishikawa, and Shirai 1999)。音声翻訳システムでは、話者の発声した内容を正しく認識、翻訳し利用者に伝える必要がある。このためには、音声認識、機械翻訳の各々の性能向上のみならず、システム全体としての性能の向上も重要な課題である。

音声認識結果において誤りを含む可能性は依然としてが高く、このような音声認識結果を適切に翻訳することは重要な研究課題の一つである。

音声認識と OCR の認識結果に基づいた言語処理には、共に認識誤りに対する頑健性という課題がある。OCR の分野では、すでに誤り訂正という方向での研究活動が活発に行なわれている (竹内 松本 1999)(新納 1999)(永田 1998)(Kukich 1992)。一方、音声認識では正解を含むのに必要な認識候補の探索空間は巨大であり、それゆえ音声認識における誤り訂正はより難しい問題である。誤り訂正に関する研究活動は OCR と比べてそれほど活発でない。音声認識中に統語的制約を適用することによって、誤り部分に対する品詞列の訂正結果を得る手法 (Tsukada, Yamamoto, Takezawa, and Sagisaka 1998) や、文字 N-gram と誤りパターンに基づく訂正手法 (Kaki, Sumita, and Iida 1998) などがある。

困難な誤り訂正まで行なわずに、認識結果に対する妥当性判断によってシステムの頑健性を高める手法も検討されている。例えば、統計的言語モデルから計算される Confidence Measure などに基づくものがある。また、構成素境界解析から計算される意味的距離に基づいて、認識結果の正しい部分のみを翻訳する手法が提案され頑健性を改善することが確認されている (脇田, 河井, 飯田 1998)。

我々は、人間が会話において、聞き取りがうまくいかなかった場合でも、話題に関する知識を元に発話の内容を推測して回復するように、コーパス中の用例から、誤りを含んだ認識結果と類似した表現を探し、誤り部分の訂正に生かすアプローチを検討してきた (Ishikawa, Sumita, and Iida 1998)(Ishikawa and Sumita 1999)。我々の手法は、訂正候補の妥当性を音韻と意味の両方の観点から判断するもので、評価実験によってその有効性が確認された。

本稿では、まず提案手法である誤り訂正の方法について説明する。続いて、日英、日韓、英日方向での音声翻訳実験の結果を示し、提案手法の有効性を検証する。さらに、入力に対してオープンなデータベースを用いた場合の有効性についても、同様に日英方向での音声翻訳実験の結果に基づいて検討する。

なお、巻末に付録として本稿の評価実験に用いた音声認識誤り訂正実験システムの手引を添付する。

2 提案手法

ここでは、まず提案手法の処理の流れを説明し、続いて各処理に関して詳細に説明する。

2.1 処理の流れ

図1では、誤りを含んだ音声認識結果の一例として“和室のほうですと18000円いずれの戦サービス料は別になります”に対する訂正処理の流れを示している。ここで、“いずれの戦サービス料”は“いずれも税サービス料”の誤りである。以下、処理の全体の流れを説明する。

I 認識結果の訂正必要性判断

I-1 構文解析および意味距離計算

入力である音声認識結果に対して構文解析を行なう(意味的距離計算を含む)。その結果、認識結果の構文解析木と、その意味的距離を得る。

図中、構文解析木の各ノードには意味的距離の値が示されており、これらの値の総和である“意味的距離の総和”は1.42である。

I-2 意味的距離による訂正必要性判断

意味的距離に対して適切な閾値 θ を設定する。意味的距離の総和がこの閾値以下の場合には“訂正不要”、閾値より大きい場合は“訂正必要”と判断する。“訂正不要”の場合、入力に対する判断“訂正不要”と共に元の音声認識結果を出力する。

本研究では意味的距離の閾値を $\theta = 1.00$ とした。図中の音声認識結果に対する意味的距離の総和1.42は閾値 θ より大きいため訂正必要と判断される。

I-3 訂正箇所の特定

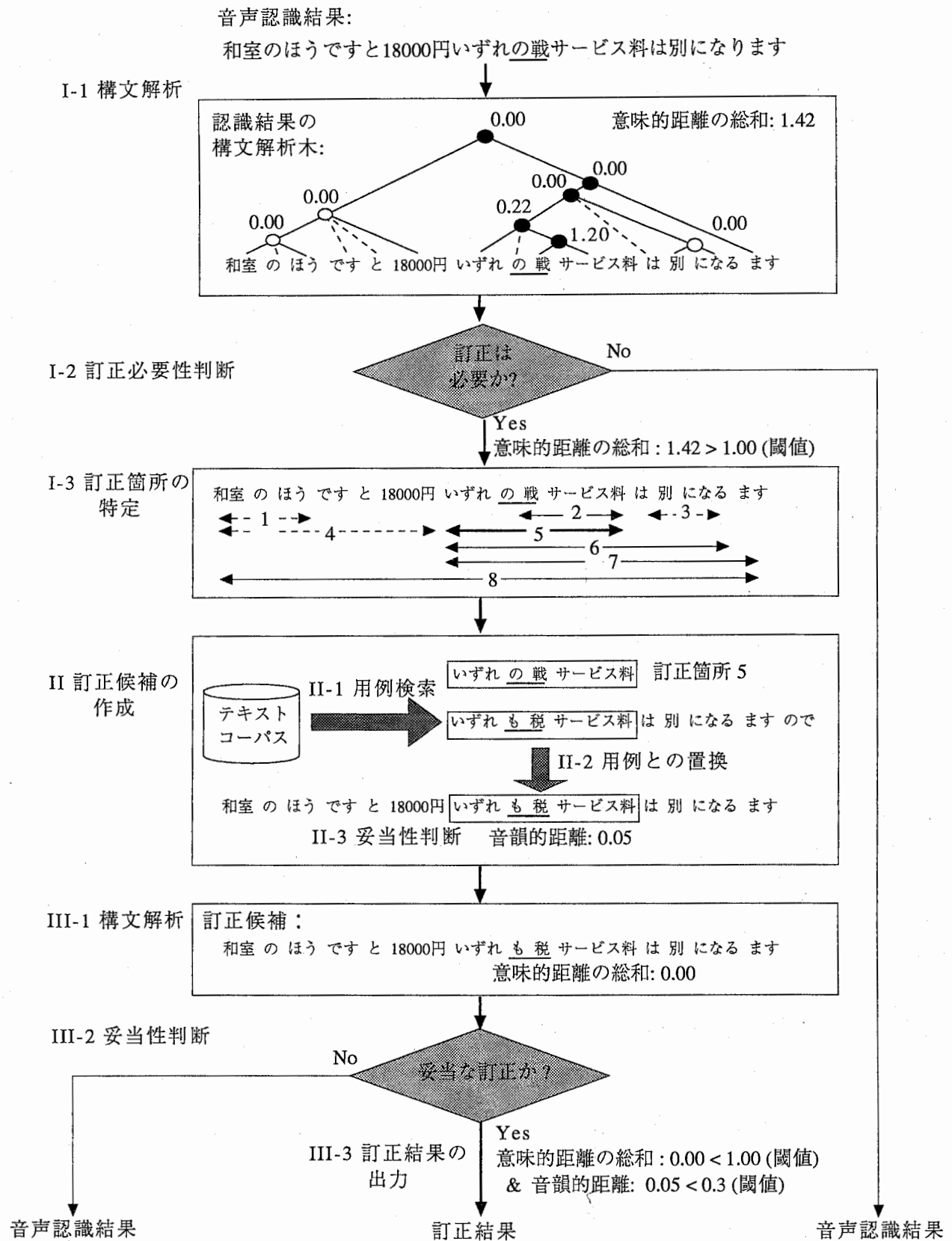
I-2で“訂正必要”と判断された場合、構文解析木と意味的距離の値から部分木の意味的距離の総和が閾値 θ をこえた場合、その葉に対応する部分を訂正箇所とする。

図中、構文解析木の部分構造に対応する形態素列を、形態素列“和室のほうですと18000円いずれの戦サービス料は別になるます”上に領域1~8として示している。これらの領域のうち、部分構造に対する意味的距離の総和が閾値 $\theta = 1.00$ を超える領域2,5,6,7,8を訂正箇所とする。

II 訂正候補の作成

II-1 テキストコーパスからの用例検索

I-3で求められた訂正箇所に基づいてテキストコーパスからの用例検索を行ない、訂正箇所と音韻的に類似した部分を含む用例を得る。



音声認識結果からの翻訳:

A Japanese-style room is 18,000 yen,
and anyway service charge for the match isn't included.

訂正結果からの翻訳:

A Japanese-style room is 18,000 yen,
and tax and service charge aren't included in either case.

図 1: 提案手法の処理の流れ

図中、訂正箇所5 “いずれの戦 サービス料” に対しては、類似した文字列を含む用例 “いずれも税 サービス料” は別 になる ます ので” が検索される。

II-2 訂正箇所の用例との置換

認識結果の訂正箇所を、用例文の対応箇所と置換し、訂正候補を得る。

図中、認識結果の訂正箇所5 “いずれの戦 サービス料” が、II-1 によって得られた用例の対応箇所 “いずれも税 サービス料” との置換により訂正され、訂正候補 “和室 の ほう です と 18000 円 いずれも税 サービス料” は別 になる ます” を得る。

II-3 音韻的距離による妥当性判断

音韻的距離に対して適当な閾値 δ を設定する。II-2 で得られた訂正候補と音声認識結果の間の音韻的距離を計算し、その値が閾値 δ を越える候補は妥当性が低いと判断し、排除する。本研究では音韻的距離の閾値を $\delta = 0.3$ とし、図中の訂正候補に対する音韻的距離は 0.05 であるため、妥当と判断される。

III 訂正候補の妥当性判断

III-1 構文解析および意味距離計算

II-3 で音韻的に妥当と判断された訂正候補に対して構文解析を行なう (意味距離計算を含む)。その結果、訂正候補の意味的距離を得る。

図中、訂正箇所5 に関して得られた訂正候補の “意味的距離の総和” 0.00 を得る。

III-2 意味的距離による妥当性判断

意味的距離の総和が閾値 θ を越える候補は妥当性が低いと判断し、排除する。訂正候補が全て排除された場合、入力に対する判断 “訂正不可” と共に音声認識結果を出力する。

図中、訂正箇所5 に関して得られた訂正候補の意味的距離の総和 0.00 は、意味的距離の閾値 $\theta = 1.00$ より小さいため、妥当と判断される。

III-3 訂正結果の出力

III-2 で妥当と判断された訂正候補が存在する場合、入力に対する判断 “訂正可能” と共に音韻的距離が最小のものを訂正結果として出力する。

図中、日英音声翻訳において、音声認識結果からの翻訳結果、および訂正結果 (訂正箇所5 に関して得られた訂正候補と同一) からの翻訳結果を示している。音声認識結果からの翻訳における下線部の誤りは、日本語認識結果における下線部の誤り “いずれの戦 サービス料” に起因しているが、訂正処理によって、“いずれも税 サービス料” と訂正されることにより、訂正結果からの翻訳では、日本語正解文の意味が適切に伝わる良好な翻訳結果を得る。

2.2 構文解析および意味的距離計算

I-1 および III-1 の構文解析および意味距離計算では、それぞれ認識結果、訂正候補に対して、構成素境界解析(古瀬, 山本, 山田 1999)によって構文解析を行なっている。

構成素境界解析は、入力文に対して変項と構成素境界よりなる表層パタンの照合を行なうことによって、ボトムアップに構文構造を構築する。また、得られる入力文の構文構造の候補中から、構文構造に対する意味的距離に基づいて最尤構造の選択を行なっている。この意味的距離は、各構成素境界パターンに関する意味的距離の総和である。それぞれのパターンに対する意味的距離(図1中の構文解析木中の各ノードに示した数値に対応)は、各変項についての単語間の意味的距離の合計値で定義されている。ここで、単語間にはシソーラス(大野 浜西 1981)上での意味属性の位置関係に基づき、0 ~ 1 の意味的距離が与えられている。この意味的距離によって、得られた構造または入力 of 妥当性を判定できる。

2.3 テキストコーパスからの用例検索

II-1 のテキストコーパスからの用例検索では、I-3 で求められた訂正箇所に対して音韻的に類似した部分を含む用例文の検索を行なう。

まず、テキストコーパスから、訂正候補の作成に有用な用例文を大まかに絞り込むために、訂正箇所に対して、文字列に関する類似度が一定以上の部分を含む用例文を抽出する(ここでは音韻的な類似度を文字列の類似度で近似的に扱う)。本研究では、Y. Lepage の方式(Lepage 1997)を用い、類似度は訂正箇所に対する部分文字列の文字に関する一致率を用いている。

さらに、II-2 において訂正箇所の用例との置換を行なうため、検索によって絞り込まれた用例文の形態素単位での対応付けが必要となる。

ここでは、用例文の形態素列中の可能な部分列のうち、訂正箇所に対して形態素の編集距離が最小となる形態素部分列を DP マッチングによって求め、訂正箇所への対応部分とする。

2.4 音韻的距離の計算

II-3 の音韻的距離による妥当性判断では、II-2 で得られた訂正候補の妥当性を、認識結果に対する音韻的距離に基づいて判断する。

認識結果と訂正候補の間の音韻的距離は、認識結果の音素列を s_1 、訂正候補の音素列を s_2 とすると、音素列 s_1 と s_2 の間の編集距離に基づいて次のように定義される。

$$\text{音韻的距離}(s_1, s_2) = \frac{\text{編集距離}(s_1, s_2)}{\text{全音素数}(s_1)}$$

ここで、編集距離(s_1, s_2)は、DP マッチングによって計算される音素列 s_2 の音素列 s_1 に対する編集距離である。

図中の訂正候補に対する音韻的距離の計算に関しては次のようになる。認識結果の形態素列“和室のほうですと18000円いづれの戦サービス料は別になるます”に対する認

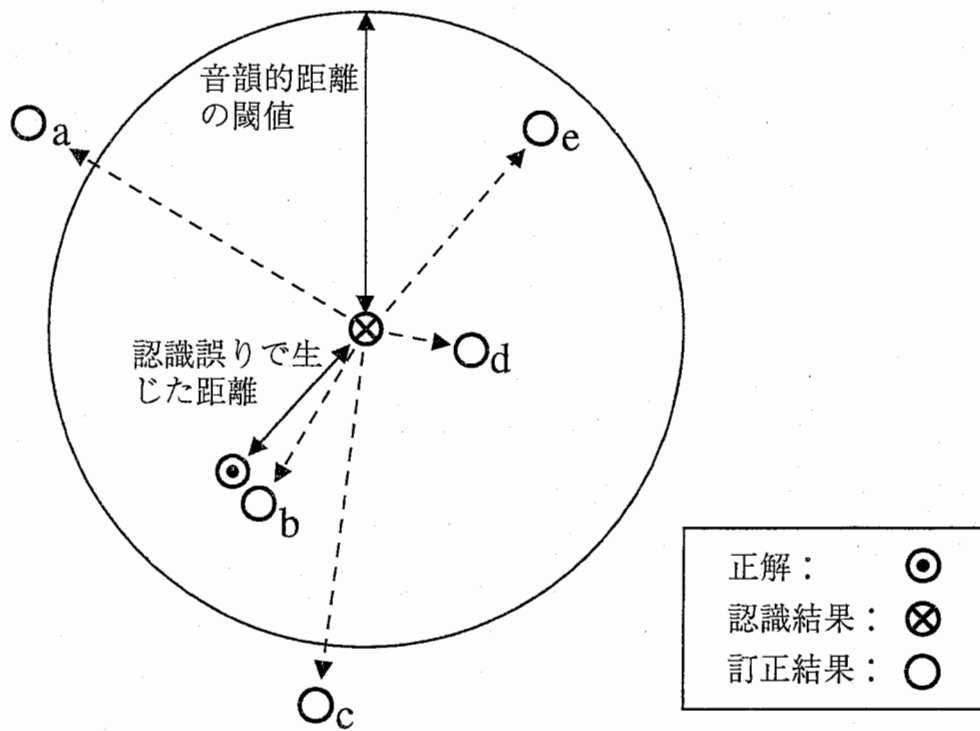


図 2: 正解、認識結果および訂正候補の間の音韻的な関係 (模式図)

認識結果の音素列 s_1 は “w_a_s_h_i_t_s_u_n_o_h_o_d_e_s_u_t_o_18000_e_n_g_i_z_u_r_e_n_o_s_e_n_g_s_a_a_b_i_s_u_r_j_o_o_w_a_b_e_t_s_u_n_i_n_a_r_u_m_a_s_u” であり、また訂正候補の形態素列 “和室のほうですと 18000 円いづれ も 税 サービス料は 別 になる ます” に対する訂正候補の音素列 s_2 は “w_a_s_h_i_t_s_u_n_o_h_o_d_e_s_u_t_o_18000_e_n_g_i_z_u_r_e_m_o_z_e_e_s_a_a_b_i_s_u_r_j_o_o_w_a_b_e_t_s_u_n_i_n_a_r_u_m_a_s_u” である。認識結果に対する訂正結果の音素編集は、挿入 : 0、削除 : 0、置換 : 3($n \rightarrow m, s \rightarrow z, ng \rightarrow e$) であるから、音韻的距離は $3/58 = 0.05$ となる。

2.5 訂正候補の妥当性判断

提案手法が用いている訂正候補の妥当性判断の考え方、および効果について説明する。図 2 は、正解、認識結果、訂正結果の音韻的距離における関係を模式的に表している。⊙ は正解、⊗ は認識結果、 $o(a, \dots, e)$ はそれぞれ訂正候補を表す。正解 ⊙ と認識結果 ⊗ の間の距離は、認識誤りによって生じた音韻的距離を表す。また、認識結果 ⊗ と訂正候補 $o(a, \dots, e)$ の間の距離は、訂正処理によって生じた音韻的距離を表す。訂正候補 $o(a, \dots, e)$ のうち、妥当なのは正解 ⊙ に最も近い b であると予想される。しかしながら、実際の音声翻訳において正解 ⊙ は未知である。

提案手法では、訂正候補の妥当性を次の仮定に基づいて判断する。

音韻的距離の仮定 ある値の誤り率を持つ認識結果に対して、その認識結果から一定の音韻的

距離以内に正解が含まれるような音韻的距離の閾値が存在する。

意味的距離の仮定 ある意味的に妥当な認識結果に対して、その認識結果の意味的距離が一定の値以下となるような意味的距離の閾値が存在する。

(1) まず音韻的距離の仮定に基づき、認識結果 ⊗ からの音韻的距離がある閾値を越える訂正候補は、正解には該当しないとして排除する。図の候補 a, c は、認識結果 ⊗ からある閾値以上 (円外) にあるため、排除される。

(2) 次に意味的距離の仮定に基づき、意味的距離がある閾値を越える訂正候補は、意味的に妥当でないとして排除する。図の候補 d は誤りが部分的、もしくは誤って訂正された不完全な訂正候補である。認識結果からの音韻的距離は近いが、意味的距離の値が閾値より大きいため意味的妥当性が十分には回復していないと判断して排除する。

(3) 最後に、(1)、(2) での条件を満たす訂正候補が複数ある場合、その内音韻的距離が最小のものを最終的な訂正結果として出力する。図では、音韻的、意味的に妥当な候補 b, e のうち、認識結果により近い候補 b を最終的な訂正結果として出力する。

このように、訂正候補に対して音韻的妥当性と意味的妥当性の異なる判断を組み合わせる事によって、妥当性判断における信頼性をより高める効果が得られる。

表 1: 音声翻訳における誤り訂正の実験条件 (日英)

日本語テキストデータ	旅行会話データ 618 会話 (異なり 15,265 発声)
日本語テストセット	旅行会話データ 467 発声 (機械翻訳の学習セット)
意味的距離の閾値	1.0
音韻的距離の閾値	0.3

表 2: テストセットに対する認識誤りと訂正判断 (日英)

入力 of 認識	訂正不要	訂正不可	訂正可能	合計
誤りなし	147	0	0	147
誤りあり	172	85	63	320
合計	319	85	63	467

3 日英音声翻訳における誤り訂正実験

3.1 実験条件

表 1 に示すような実験条件で、日本語音声認識結果に対する誤り訂正実験を行なった。テキストデータとして、ATR 旅行会話データ中の 618 会話 (異なり 15,265 発声) を使用した。テストセットは、テキストデータに含まれる日英機械翻訳の学習セット 467 発声を用いた。日本語音声認識の精度は、単語正解率: 68.0%、発声正解率: 31.5% であった。また、訂正処理では意味的距離の閾値は 1.0、音韻的距離の閾値は 0.3 を用いた。

3.2 訂正判断の精度

前章での説明のように、提案手法では入力の音声認識結果に対して、訂正不要 (誤りを含まないので訂正は不要)、訂正可能 (誤りを含むので訂正は必要であり、誤りに対する適切な訂正が可能)、訂正不可 (誤りを含むので訂正は必要だが、誤りに対する適切な訂正は不可能)、のいずれかの判断を下す。認識結果が正しい場合には、訂正不要と判断され、誤りが含まれている場合には、訂正可能もしくは訂正不可と判断されるのが理想的といえる。ここで、テストセット 467 発声の音声認識結果に対する訂正判断の結果と認識誤りの有無の関係を表 2 に示す。

誤りなしの認識結果に対してはすべて、訂正不要と判断されている。一方、訂正が必要と判断した発声 (訂正可能、訂正不可と判断された発声の和集合) において、誤りありの認識結果の再現率は 46.3% (148/320)、適合率は 100% (148/148) である。また、訂正が必要と判断した発声のうち、訂正可能であった発声の割合は 42.6% (63/148) である。

表 3: 発声の単語誤り率での分布 (日英)

単語誤り率 R	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R = 0\%$	169	0	0	169	13	182
$0\% < R \leq 20\%$	58	9	20	87	34	101
$20\% < R \leq 40\%$	31	17	34	82	9	57
$40\% < R$	61	59	9	129	7	127

もちろん、これらの数値の大小のみで単純に誤り訂正の有効性が結論されるわけではない。訂正が必要と判断されなかった誤りのある認識結果の中には、誤り訂正を行なわなくても発声の意味が十分理解できるような軽微な誤りのものが存在することが考えられ、また、訂正不可と判断された認識結果の中には、誤りがひどいために正解の推定が困難で、誤り訂正が本質的に不可能なものが存在することが考えられるからである。3.4以降ではさらに、認識結果における誤りのひどさと訂正判断の関係、訂正結果の有効性等について、詳細な検討を行なう。

3.3 DP マッチングによる単語誤り率

音声認識における単語誤りの訂正前後での増減から誤り訂正の有効性を検証する。誤りの尺度として単語誤り率を導入し、認識結果と訂正結果における誤り率の比較を行なう。単語誤り率を、ここでは DP マッチングを使って正解からの編集距離に基づいて定義する。単語の挿入 (Ins)、削除 (Del)、置換 (Sub) に関する誤りの数をそれぞれ N_{Ins} 、 N_{Del} 、 N_{Sub} 、正解の単語数を N_{Ans} とした時、単語誤り率 R は以下のように定義される。

$$\text{単語誤り率 } R = \frac{N_{Ins} + N_{Del} + N_{Sub}}{N_{Ans}}$$

3.4 訂正前後での単語誤り率

表 3では、単語誤り率 R の値を 4 段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) に分け、訂正前後での各訂正判断に対する発声の誤り率 R に関する分布を示した。訂正不要、訂正不可の発声に関しては訂正結果が出力されないので訂正前のみを示し、訂正後の結果は訂正可能の発声とテストセット全体 (“合計”) のみにについて示す。

まず、訂正不要の分布を見ると、その半数以上 (169/319) が誤り率 $R = 0\%$ となっている。¹しかし同時に、誤り率が 40% を越える発声が 61 発声もある。これらの多くは、表面的には文の体裁を備えていることに特徴がある。以下の実例に示すような、元の発声とは別の意味になっ

¹表 2と比較して誤りなしの発声が 22 発声増えている。DP マッチングの際に活用を標準形に直しているため、軽微な誤りが吸収されている。

てしまう“空耳”のような誤りや、発声の一部が欠落した“聞き落し”のような誤りなどが含まれる。このような認識結果に対して、本手法では誤りである可能性(訂正の必要性)を判断するのは困難であり、訂正不要と判断されている。

“空耳”、“聞き落し”のような誤り

正解:	いつのお泊りでしょうか
認識結果:	えその通りでしょうか
正解:	はいそうですそれではお待ちいたしております
認識結果:	それではお待ちいたしております

また、訂正不可の分布を見ると、その7割近くが誤り率40%以上であり、誤りのひどい認識結果が多くを占めていることがわかる。認識結果の誤りがひどく、正解との音韻的距離が閾値を越える場合、前章の訂正候補の妥当性の判断での説明ように、認識結果から音韻的に離れた訂正候補は、妥当でないと判断される。認識誤りの一例として以下に示した場合も、たとえ正解が訂正候補(仮説)として得られたとしても、音韻的に妥当でないと判断される。したがって、誤りのひどい認識結果に対しては、本手法では訂正不可と判断される。

誤りのひどい認識結果

発声(正解):	ただがんがんしてても起きてしばらくすると治るのですけど
認識結果:	あとでお電話したと思うのを着てしばらく都合の分ですよね

一方、訂正可能と判断された認識結果の誤り率は、20%あたりを中心に分布している。訂正結果には、誤り率20%以下の発声数が20から47へと増加しており、訂正によって誤り率が改善されていることが分かる。訂正可能と判断された認識結果では、訂正不可と判断された認識結果に比べて誤りが発声の一部に局所的にまとまる傾向が見られる。以下にそのような訂正の実例を示す。

訂正の実例

発声(正解):	シャトルバスは知っていますがタクシーを使いたいのです
認識結果:	シャトルバスをしていますタクシーを使いたいのです
訂正結果:	シャトルバスは知っていますがタクシーを使いたいのです
発声(正解):	えラスベガスからロサンジェルスまでの運賃はおいくらくらいですか
認識結果:	ラスベガスからロサンジェルスまでの音痴なおいくらくらいですか
訂正結果:	ラスベガスからロサンゼルスまでの運賃はおいくらくらいですか
発声(正解):	はい祇園の辺りですねそういたしましたら加茂川ホテルはいかがでしょうか
認識結果:	はい従来ですねそういたしましたら加茂川お寺はいかがでしょうか
訂正結果:	はい従来ですねそういたしましたら鴨川ホテルはいかがでしょうか

表 4: テストセット全体に対する単語誤り率 (日英)

訂正前				訂正後	
訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
26.1%	54.0%	30.5%	33.1%	18.1%	30.8%
(689/2636)	(531/984)	(255/835)	(1475/4455)	(151/835)	(1371/4455)

一番目の例では、訂正結果は正解と完全に一致している。二番目の例では、正解との間に表記上の微妙な違いが存在するが、意味的には完全に同等である。三番目の例では、認識結果に誤り箇所が複数あり、その一部が訂正されずに残留している。表 3 で、訂正可能な発声に関して訂正前後での誤り率の変化を見ると、完全に正解に一致する誤り率 0% の訂正結果だけでなく、正解に準じる 20% 以下程度の訂正結果も得られることが分かるが、二番目、三番目の例はその具体例にあたる。

3.5 テストセット全体に対する単語誤り率

ここでは、入力全体に関する誤り率の訂正前後での変化に基づいて、訂正処理の有効性を検証する。テストセットの音声認識結果に関する単語誤り率を表 4 に示す。テストセット全体に対する単語誤り率は訂正の前後で 33.1% から 30.8% へ変化しており、2.3% の改善が見られた。訂正可能な単語率では、30.5% から 18.1% へと変化しており、訂正の行なわれた発声に対しては、単語誤りをほぼ半減する効果が得られている。

さらに、誤りを種類別 (挿入、削除、置換) および品詞別に分け、それぞれの訂正判断での誤りの分布の傾向や、訂正の有効性について検証する。訂正前後での挿入、削除、置換誤りに関する誤り単語数を品詞別に表 5 に示す。ただし、品詞はそれぞれ、“挿入”では挿入後の認識結果の単語、“削除”では削除前の正解の単語、“置換”では置換前の正解の単語に対するものであり、“その他”には“感動詞”、“接頭辞”、“接尾辞”をまとめた。

まず、それぞれの訂正判断での誤りの種類の割合を見ると、訂正不要の発声では削除誤りが全体の半数以上を占めている。これに対して訂正不可、訂正可能では置換誤りが全体の半数以上を占めている。このことから、削除誤りに対する誤りの判断は難しく、見逃してしまう傾向が強いと考えられる。このことは、認識結果の実例で示したように、発声の一部が欠落した“聞き落し”誤りは訂正必要性の判断が困難であるという事実と一致している。

次に、訂正不可と訂正可能での誤りの種類の割合を比較すると、挿入、削除、置換の比率はどちらもほぼ 1:1:3 であり、特に大きな差は認められない。しかし、単語誤り率は両者間で大きく異なっており、単語誤り率 R が 40% を下回る発声の占める割合は、訂正不可の方が圧倒的に高い。このことから、認識結果の訂正可能性は、主に発声の単語誤り率によって決定さ

表 5: 挿入、削除、置換誤りの品詞別誤り単語数 (日英)

	誤り単語の品詞	訂正前			訂正後
		訂正不要	訂正不可	訂正可能	訂正可能
挿入	名詞	35	32	12	2
	動詞	6	20	6	2
	助動詞	12	18	7	0
	助詞	22	24	12	6
	形容詞	2	1	1	0
	副詞	2	3	1	1
	連体詞	0	0	0	0
	接続詞	1	0	0	0
	その他	8	4	1	0
	合計	88	102	40	11
削除	名詞	110	27	16	17
	動詞	49	18	4	0
	助動詞	77	21	8	8
	助詞	103	30	18	16
	形容詞	3	1	2	0
	副詞	8	1	0	0
	連体詞	2	0	0	0
	接続詞	6	0	1	1
	その他	29	5	10	9
	合計	387	103	59	51
置換	名詞	94	116	57	34
	動詞	24	50	23	10
	助動詞	26	36	15	8
	助詞	34	78	47	21
	形容詞	2	11	4	1
	副詞	3	11	0	1
	連体詞	2	5	3	1
	接続詞	1	2	2	4
	その他	18	14	5	9
	合計	214	326	156	89

表 6: 翻訳結果の評価基準

- A 入力文の正解と完全に同じ意味にとれる
- B 入力文の正解とほぼ同じ意味にとれる
- C 入力文の正解の主要情報が部分的には伝わる
- D 入力文の正解の主要情報が伝わらない／誤解が生じる

表 7: 翻訳結果の評価値 (日英)

評価値	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
A	190	1	2	193	36	227
B	35	5	10	50	11	51
C	18	6	16	40	6	30
D	75	73	35	183	10	158
翻訳率: $\frac{N_A+N_B+N_C}{N_A+N_B+N_C+N_D}$	76%	14%	44%	61%	84%	66%

れると結論できる。

さらに、訂正可能での誤り単語数に関して訂正前後で比較すると、削除誤りは、挿入、置換と比べて訂正の効果が比較的小さい。一方、挿入誤り、置換誤りに対する誤り単語数を訂正前後で比較すると、名詞、動詞、助動詞、助詞、形容詞など、品詞による偏りはそれほどなく全体的に減少していることが分かる。内容語、機能語を問わず、品詞によらず効果がある。

3.6 翻訳文の主観評価

ここでは、音声翻訳における訂正手法の有効性を検証するために、音声認識結果および訂正結果をもとに機械翻訳された日英翻訳の品質に対して主観評価を行なう。本研究では機械翻訳として TDMT (Sumita et al. 1999) を用いる。翻訳結果の主観評価では、翻訳結果から理解、伝達される発声の内容や情報が正しいかどうかを、入力文の正解 (原言語) と比較することによって判断する。評価者は、日英翻訳を専門とするアメリカ人ネイティブであり、各発声の認識結果および訂正結果に対する翻訳に関して、入力文の正解 (原言語) に基づく理解を基準として、表 6 に定義された評価値 (A, B, C, D) のいずれかを選択する方法で評価を行なった。

音声認識結果、およびその訂正結果を入力として翻訳システム (TDMT) を用いて得られた翻訳結果に関する主観評価を表 7 に示す。ここで、“翻訳率” は各訂正判断に対する発声数全体 $N_A + N_B + N_C + N_D$ に対する、評価値 A、B、C の発声数の合計 $N_A + N_B + N_C$ の割合である。

まず、テストセット全体に対する翻訳率は訂正前後で61%から66%へ増加しており、5%の改善が見られる。さらに、訂正可能と判断された発声に関しては、44%から84%へ増加している。訂正可能と判断された発声は訂正前で評価値Dが過半数を占め、翻訳結果の意味理解が不可能となるような、音声翻訳で致命的な誤りを含んだ音声認識結果が多く訂正の対象となっていることが分かる。

また、訂正後では逆に評価値Aが過半数を占めていることから、訂正結果からの翻訳で、発声の内容や情報が正しく回復されていることが分かる。訂正後の評価値Aの発声数は36であるが、これは表3での訂正後の誤り率 $R = 0\%$ の発声数13と比べて23発声多い。これは、訂正結果は正解に完全一致しないが、その翻訳結果が正解となるものが存在することを意味している。以下は、そのような実例である。

正解に完全には一致しないが、翻訳結果は正解となる訂正結果

正解:	恐れいりますがシングルルームは満室となっております
認識結果:	恐れいりますがシングルルームが増しとなっております
訂正結果:	恐れいりますがシングルルームが満室となっております
正解翻訳:	"I'm sorry , but single rooms are all occupied ."
認識翻訳:	"I'm sorry , but single rooms are the increase ."
訂正翻訳:	"I'm sorry , but single rooms are all occupied ."

認識結果では、(は → が)および(満室 → 増し)の2単語に対して置換誤りが起こっており、後者は正しく訂正されているが、前者は残っている。ところが、訂正結果からの翻訳は、正解からのものと完全に一致している。訂正前後での単語誤り率 R の変化は25.0% → 12.5%であるが、翻訳の評価値は、 $C \rightarrow A$ と完全に回復している。置換誤り(は → が)がこの場合、発声の理解や翻訳を妨げなかったため、誤りとは判断されなかったのである。

訂正不要と判断された認識結果の中にも、表層的には正解に完全一致しないが、その翻訳結果は正解となるものが同様に存在している。評価Aの発声数は190であるが、これは誤り率 $R = 0\%$ の発声数169や、表2での誤りなしの発声数147より多い。訂正が必要という判断の精度は、誤りありの認識結果に対して計算すると適合率100%、再現率42.3%であったが、ここでの評価値Aの発声以外を訂正が必要な発声として計算すれば、適合率98.0% (145/148)、再現率53.3%(145/273)、さらに評価値Bまでの発声を訂正不要として計算すれば、適合率87.8% (130/148)、再現率58.3% (130/223)となる。主観評価は翻訳結果における発声の内容理解、情報の伝達可能性という観点に基づいているので、これらの適合率、再現率は音声翻訳タスクに対してより実質的な値を表しているといえる。

3.7 翻訳結果の単語誤り率に基づく評価

ここでは、すでに導入された単語誤り率 R を翻訳結果の評価に用いる。このような、正解の翻訳と認識結果の翻訳の間での編集距離に基づく評価方法は、音声翻訳の自動評価方法として

表 8: 翻訳結果の単語誤り率 (日英)

誤り率 R	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R = 0\%$	190	1	1	192	31	222
$0\% < R \leq 20\%$	20	4	11	35	15	39
$20\% < R \leq 40\%$	27	6	12	45	7	40
$40\% < R$	82	74	39	195	10	166
$0\% \leq R \leq 40\%$ の割合	74%	13%	38%	58%	84%	64%

竹澤等 (Takezawa, Sugaya, Yokoo, and Yamamoto 1999) が提案している。同様に、訂正前後の認識結果からの翻訳の単語誤り率 R の比較により、翻訳結果における訂正の効果を議論することができる。

テストセットの翻訳結果に対する単語誤り率 R の値を4段階に分けて、各発声の誤り率 R に関する分布を表8に示す。主観評価の結果を示した表7と比較すると、興味深いことに、発声数の分布の傾向がよく一致しており、誤り率 R の各段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) と、主観評価の各評価値 (A,B,C,D) がそれぞれ対応しているように見える。テストセット全体に対する誤り率 $0\% \leq R \leq 40\%$ の発声の割合は、訂正前後で58%から64%へと改善しており、主観評価での翻訳率と同様に、翻訳結果の誤り率でも訂正の効果が確認できる。

また、翻訳結果での誤り率 $R = 0\%$ の発声数が、表3での認識結果での値に比べて増えていることは、主観評価での傾向と一致し、表層的には正解に完全一致しないが翻訳結果は正解となるような訂正結果が存在することを裏付けている。

このように、主観評価と単語誤り率 R の評価の比較から、各翻訳結果に対する評価値と単語誤り率 R の間には正の相関関係が存在することが予想される。テストセット全体に対する訂正後の各発声の翻訳結果について、その単語誤り率 R と、主観評価での評価値の間の関係を表9に示す。表からは、対角成分への分布の偏りが見られる。特に、評価値 A と $R = 0\%$ 、評価値 D と $40\% < R$ の間に強い相関が現れていることが分かる。²

ここで、評価値と誤り率の相関性を定量的にはかる尺度として、相関係数を導入する。2変量に関する n 個の観測値 $(x_i, y_i) (i = 1, 2, \dots, n)$ に対して、相関係数 r は以下のように定義できる。

$$r = \frac{\frac{1}{n} \sum_{i=1}^n \Delta x_i \Delta y_i}{\sqrt{\frac{1}{n} \sum_{i=1}^n \Delta x_i^2} \sqrt{\frac{1}{n} \sum_{i=1}^n \Delta y_i^2}}$$

²ここで、誤り率 R の段階の閾値は便宜的に決定したものであり、相関性を最大化するような最適化は行っていない。

表 9: 翻訳結果の評価値と単語誤り率の相関 (日英)

誤り率 R	評価値			
	A	B	C	D
$R = 0\%$	173	17	1	1
$0\% < R \leq 20\%$	9	13	11	2
$20\% < R \leq 40\%$	6	10	10	19
$40\% < R$	5	10	18	161

ここで、

$$\Delta x_i = x_i - \bar{x}, \quad \Delta y_i = y_i - \bar{y},$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

さらに、それぞれ評価値 (a,b,c,d) と単語誤り率 R の段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) が与えられたある翻訳結果を、以下の 2 変量上の観測点と解釈し直す。

$$x = \alpha_x l_x + \beta_x, \quad l_x = (0, 1, 2, 3),$$

$$y = \alpha_y l_y + \beta_y, \quad l_y = (0, 1, 2, 3)$$

l_x には、主観評価の各評価値 (a,b,c,d) に対してそれぞれ (0,1,2,3)、 l_y には、誤り率 R の各段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) に対してそれぞれ (0,1,2,3) を与える。 α_x 、 α_y 、 β_x 、 β_y は $\alpha_x \alpha_y > 0$ を満たす任意の実数であり、相関係数 r は、これらの値に対し不変である。

このような方法で計算された単語誤り率 R と主観評価での評価値の間での相関係数は 0.88 であり、両者の間には明らかに相関関係が成り立っていると結論できる。

表 10: 音声翻訳における誤り訂正の実験条件 (日韓)

日本語テキストデータ	旅行会話データ異なり 15,265 発声
日本語テストセット	旅行会話データ 169 発声 (機械翻訳の学習セット)
意味的距離の閾値	1.0
音韻的距離の閾値	0.3

4 日韓音声翻訳における誤り訂正実験

ここでは、多言語音声翻訳における提案手法の有効性を示すために、日韓翻訳における誤り訂正実験を行なう。日本語音声認識に対する誤り訂正である点では、前章 3 ですすでに議論された日英翻訳での実験と共通しているが、翻訳結果における誤り訂正の効果は翻訳言語対ごとに異なることが予想される。

4.1 実験条件

表 10 に示すような実験条件で、日本語音声認識結果に対する誤り訂正実験を行なった。テキストデータとして、日英翻訳における実験と同じ ATR 旅行会話データ中の 618 会話 (異なり 15,265 発声) を使用した。テストセットは、テキストデータに含まれる日韓機械翻訳の学習セットの 169 発声を用いた。このテストセットに対する音声認識の結果は、単語正解率: 77.9%、発声正解率: 45.0% であった。訂正処理のパラメーターは、日英翻訳における実験と同様、それぞれ意味的距離の閾値: 1.0、音韻的距離の閾値: 0.3 を用いた。

4.2 訂正判断の精度

テストセット 169 発声の音声認識結果に対する訂正判断の結果と認識誤りの有無の関係を表 11 に示す。訂正が必要と判断した発声 (訂正可能、訂正不可と判断された発声の和集合) において、誤りありの認識結果の再現率は 51.5%(47/93)、適合率は 100%(47/47) である。また、訂正が必要と判断した発声のうち、訂正可能であった発声の割合は 55.3%(26/47) である。いずれも日英翻訳での実験と比べて高い値となっているが、テストセットの音声認識の認識率の違いによると考えられる。

4.3 訂正前後での単語誤り率

訂正前後での各訂正判断に対する発声を、単語誤り率 R の値に基づいて 4 段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) に分け、表 12 に示した。訂正前のテストセット全体の分布は、3 での日英翻訳の場合と比べると誤り率が 40% 以上の発声の割合は低く、誤り

表 11: テストセットに対する認識誤りと訂正判断 (日韓)

入力認識	訂正判断			合計
	訂正不要	訂正不可	訂正可能	
誤りなし	76	0	0	76
誤りあり	46	21	26	93
合計	122	21	26	169

表 12: 認識結果の単語誤り率に関する分布 (日韓)

単語誤り率 R	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R = 0\%$	78	0	0	78	12	90
$0\% < R \leq 20\%$	22	2	14	38	12	36
$20\% < R \leq 40\%$	13	5	10	28	1	19
$40\% < R$	9	14	2	25	1	24

率が20%以下の発声の割合は高い。このテストセット全体の傾向を反映して、訂正不要、訂正可能と判断された発声では単語誤り率の低い発声の割合が高くなっている。

しかし一方で、訂正不可では、誤り率が40%以上の発声が7割近くを占めている。この傾向は日英の場合でも同様であり、テストセット全体の誤り率にあまり依存していない。この理由としては、認識結果の誤り率が40%以上の発声は、正解からの音韻的距離が閾値0.3を越えるために訂正不可と判断される可能性が高くなることが考えられる。すなわち、訂正不可の発声の誤り率は、主に音韻的距離の閾値によって決定されているといえる。

表13に示したテストセット全体に対する単語誤り率においても、同様の傾向が見られる。訂正前での誤り率を日英での表4と比較すると、訂正不可の発声に対する誤り率50.0%は日英での54.0%と比べて大きな差はないが、訂正不要、訂正可能の発声ではいずれも日英の値より10%程度低い。

テストセット全体に対する単語誤り率は訂正の前後で22.2%から19.2%へと変化し、3.0%の減少が見られる。訂正可能での単語誤り率では21.3%から7.6%へと減少し、単語誤りはほぼ3分の1に削減されている。日英との比較からは、誤り率の低い認識結果ほど適切な訂正が可能である傾向が読みとれる。

表 13: テストセット全体に対する単語誤り率 (日韓)

訂正前				訂正後	
訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
14.1%	50.0%	21.3%	22.2%	7.6%	19.2%
(129/917)	(141/283)	(73/342)	(343/1542)	(26/342)	(296/1542)

4.4 翻訳結果の文字誤り率に基づく評価

音声翻訳における誤り訂正の有効性を検証するために、機械翻訳 (TDMT)(古瀬他 1999) を用いた日韓翻訳結果の評価を行なう。日英の評価では、単語誤り率 R を用いて翻訳結果における訂正の効果を議論した。分かれ書きをしない韓国語に対して、ここでは評価の便宜上、文字誤り率 R_c を導入する。正解と認識結果の DP マッチングに基づく文字の挿入 (Ins)、削除 (Del)、置換 (Sub) に関する誤りの数をそれぞれ N_{Ins} 、 N_{Del} 、 N_{Sub} とし、正解の文字数を N_{Ans} とすると、文字誤り率 R_c は以下のように定義される。

$$\text{文字誤り率 } R_c = \frac{N_{Ins} + N_{Del} + N_{Sub}}{N_{Ans}}$$

表 14 に翻訳結果の文字誤り率 R_c に関する発声の分布を示す。テストセット全体の文字誤り率 R_c の分布からは、訂正によって誤り率 0% の発声数の増加が認められる。また、誤り率 40% 以下の発声の割合も 84% から 86% へと 2% 増加しており、日韓における誤り訂正の有効性を示している。

しかしながら、誤り率が 40% 以下の翻訳結果の割合における訂正の効果は、日英が日韓を上回っている (日韓では 2% の増加、日英では表 8 より 6% の増加)。この結果は、先に示した認識結果の単語誤り率に関する結果 (表 4、表 13 より、誤り訂正による誤り率の減少は日韓で 3.0%、日英で 2.3%) と一見相反する。

この結果は、日韓と日英での、訂正前における翻訳結果の誤り率が 40% 以下の発声の割合の違い (表 14、表 8 より、訂正可能の発声での割合は日韓で 85%、日英で 38%) によって理解される。誤り率が 40% を越える発声の割合が小さいほど、誤り訂正によって新たに誤り率が 40% 以下となる発声の数も少なくなるからである。訂正が行なわれる認識結果の誤り率が低い場合、認識率が改善されても翻訳率には訂正の効果があまり現れない可能性は同様に予想される。

さらに、すでに示した認識結果での誤り率で、訂正前における誤り率が 40% 以下の発声の割合において日韓と日英の間での大きな差が見られない (表 12、表 3 より、日韓で 86%、日英で 92%) ことを考慮すると、翻訳結果の誤り率における日韓と日英での違い (表 14、表 8 より、訂正可能の発声での割合は日韓で 85%、日英で 38%) は、認識誤りの翻訳結果への影響の日韓

表 14: 翻訳結果の文字誤り率に関する分布 (日韓)

文字誤り率 R_c	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R_c = 0\%$	85	0	2	87	17	102
$0\% < R_c \leq 20\%$	20	2	13	35	4	26
$20\% < R_c \leq 40\%$	8	5	7	20	4	17
$40\% < R_c$	9	14	4	27	1	24
$0\% \leq R_c \leq 40\%$ の割合	93%	33%	85%	84%	96%	86%

と日英での違いが反映された結果であり、翻訳言語対の言語間距離が大きいほど、認識誤りの影響が翻訳結果の全体に及ぶ性質によるものと考えられる。

表 15: 音声翻訳における誤り訂正の実験条件 (英日)

英語テキストデータ	旅行会話データ 異なり 15,630 発声
英語テストセット	旅行会話データ 311 発声 (機械翻訳の学習セット)
意味的距離の閾値	1.0
音韻的距離の閾値	0.3

5 英日音声翻訳における誤り訂正実験

多言語音声翻訳における提案手法の有効性を示すために、英日翻訳における誤り訂正実験を行なう。日英、日韓での実験では、日本語音声認識に対して誤り訂正を行なった。ここでは英語音声認識に対する誤り訂正においても、音韻的距離に基づく妥当性判断が有効であることを検証する。

5.1 実験条件

表 15に示すような実験条件で、英語音声認識結果に対する誤り訂正実験を行なった。テキストデータとして、ATR 旅行会話データ中の 618 会話 (異なり 15,630 発声) を使用した。テキストデータに含まれる英日機械翻訳の学習セットの 311 発声をテストセットに用いた。このテストセットに対する音声認識の結果は、単語正解率: 67.5%、発声正解率: 26.7% であった。訂正処理のパラメーターは、日英、日韓における実験と同様、それぞれ意味的距離の閾値: 1.0、音韻的距離の閾値: 0.3 を用いた。

5.2 訂正判断の精度

テストセット 311 発声の音声認識結果に対する訂正判断の結果と認識誤りの有無の関係を表 16に示す。訂正必要と判断された発声 (訂正可能、訂正不可と判断された発声の和集合) において、誤りのある認識結果の再現率は 75.4%(172/228)、適合率は 100%(172/172) である。また、訂正必要と判断された発声のうち、訂正可能であった発声の割合は 20.9%(36/172) である。日英、日韓と比べると再現率は明らかに高く、逆に訂正可能の発声の割合は低い。この傾向から、認識結果中には訂正の困難な誤りが多いが、訂正必要の判断で曖昧性を生じる誤りは少ないという英日のテストセットの性質が予想される。

5.3 訂正前後での単語誤り率

訂正前後での各訂正判断に対する発声を、単語誤り率 R の値に基づいて 4 段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) に分け、表 17に示した。テストセット全体 (合

表 16: テストセットに対する認識誤りと訂正判断 (英日)

入力 の認識	訂正判断			合計
	訂正不要	訂正不可	訂正可能	
誤りなし	83	0	0	83
誤りあり	56	136	36	228
合計	139	136	36	311

表 17: 認識結果の単語誤り率に関する分布 (英日)

単語誤り率 R	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R = 0\%$	86	0	0	86	13	99
$0\% < R \leq 20\%$	24	11	17	52	18	53
$20\% < R \leq 40\%$	18	44	16	78	3	65
$40\% < R$	11	81	3	95	2	94

計) に関しては、誤り率 R が 20% から 40% の発声を中心に訂正の効果が得られている。

英日で訂正前に誤り率 R が 40% を越える発声の割合は 30.5%(95/311) である。(表 3、表 12 より) 日韓での 14.8%(25/169) との差から、英日のテストセットでは誤りのひどい発声が多いことが分かる。一方、日英での 27.6%(129/467) とは差が認められない。

訂正不要での誤り率 R が 40% 以上の発声の割合は、日英: 19.1%(61/319)、日韓: 7.4%(9/122)、英日: 7.9%(11/139) である。これは、英日に比べて日英では誤りのひどい認識結果において訂正判断の曖昧性を生じる誤り (特に削除誤り) が多く含まれているためである。

英日と、日英、日韓との間のテストセット全体に対する訂正不可の発声数の割合における違い (表 3、表 12、表 17 より、訂正不要、訂正不可、訂正可能の発声数の比 日英: 68%, 18%, 14%、日韓: 71%, 13%, 16%、英日: 45%, 44%, 11%) が英日で異なっていることも、英日のテストセットでの誤りのひどい発声は多いが、訂正必要の判断で曖昧性を生じる誤りは少ないという性質から理解される。英日の訂正必要の判断が高い再現率を示すことも同様に説明される。

表 18 にテストセット全体に対する単語誤り率を示す。テストセット全体に対する単語誤り率は、訂正の前後で 32.2% から 30.0% へと変化し、2.2% の減少が見られる。訂正可能での単語誤り率では 24.3% から訂正不要での値と同程度の 10.1% まで減少している。訂正不要と判断された発声の単語誤り率は日英、日韓と比べて小さい (表 4、表 13、表 18 より、日英: 26.1%、日韓: 14.1%、英日: 10.8%) が、これも訂正必要の判断で曖昧性を生じる誤りが少ないという英

表 18: テストセット全体に対する単語誤り率 (英日)

訂正前				訂正後	
訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
10.8%	45.5%	24.0%	32.2%	10.1%	30.0%
(103/950)	(843/1854)	(126/524)	(1072/3328)	(53/524)	(999/3328)

表 19: 翻訳結果の文字誤り率に関する分布 (英日)

文字誤り率 R_c	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R_c = 0\%$	104	0	0	104	15	119
$0\% < R_c \leq 20\%$	9	4	9	22	11	24
$20\% < R_c \leq 40\%$	11	27	10	48	6	44
$40\% < R_c$	15	105	17	137	4	124
$0\% \leq R_c \leq 40\%$ の割合	89%	23%	53%	56%	89%	60%

日のテストセットの性質に起因している。

5.4 翻訳結果の文字誤り率に基づく評価

音声翻訳における誤り訂正の有効性を検証するために、機械翻訳 (TDMT) を用いた英日翻訳結果の評価を行なう。ここでも日韓の評価と同様に、翻訳結果の日本語に対して文字誤り率 R_c を適用する。表 19 に翻訳結果の文字誤り率 R_c に関する発声の分布を示す。テストセット全体の文字誤り率 R_c の分布からは、訂正によって誤り率 0% の発声数の増加が認められる。また、誤り率 40% 以下の発声の割合も 56% から 60% へと 4% 増加しており、英日における誤り訂正の有効性を示している。

表 20: 音声翻訳における誤り訂正の実験条件 (日英、オープンデータベース)

日本語テキストデータ	旅行会話データ 577 会話 (入力に対するオープンセット)
日本語テストセット	旅行会話データ 467 発話 (機械翻訳の学習セット)
意味的距離の閾値	1.0
音韻的距離の閾値	0.3

6 オープンデータベースでの日英音声翻訳実験

入力文に対してオープンなデータベースを用いた際の提案手法の性能を検証するために、先に行なった日英音声翻訳における誤り訂正実験に対して対照な、オープンなデータベースを用いた実験を行なった。

6.1 実験条件

表 20に示すような実験条件で、日本語音声認識結果に対する誤り訂正実験を行なった。入力とは、日英翻訳システムで学習済の 467 発話に対する 3 で用いたものと同じ認識結果を用いた。用例データベースとしては、入力に対してオープンになるように、ATR 旅行会話データ中の 618 会話から、入力 467 発話を含む 41 会話を除いた 577 会話 (異なり 14,405 発話) を使用した。訂正処理のパラメーターは、それぞれ意味的距離の閾値: 1.0、音韻的距離の閾値: 0.3 を用いた。

6.2 訂正判断の精度

テストセット 467 発声の音声認識結果に対する訂正判断の結果と認識誤りの有無の関係を表 21に示す。訂正不要の判断は認識結果の妥当性に基づいているため、データベースの違いに依存せず 3 での結果と一致するが、訂正不可、訂正可能の判断は訂正候補の妥当性に基づいているため、データベースの違いに依存して結果が異なっている。訂正処理が必要と判断された発声に対して実際に訂正の得られた発声の割合は 12.2%(18/148) であり、3 での割合 42.6%(63/148) と比べて大幅に小さい。訂正不要の発声は変わらないので、以降では訂正不可、訂正可能の発声に限って議論する。

6.3 訂正前後での単語誤り率

表 22では、単語誤り率 R の値を 4 段階 ($R = 0\%$, $0\% < R \leq 20\%$, $20\% < R \leq 40\%$, $40\% < R$) に分け、訂正前後での各訂正判断に対する発声の誤り率 R に関する分布を示した。テストセット全体の発声の分布からは、訂正の前後でほとんど変化が見られない。訂正可能の

表 21: テストセットに対する認識誤りと訂正判断 (日英、オープンデータベース)

入力の認識	訂正不要	訂正不可	訂正可能	合計
誤りなし	147	0	0	147
誤りあり	172	130	18	320
合計	319	130	18	467

表 22: 発声の単語誤り率での分布 (日英、オープンデータベース)

単語誤り率 R	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R = 0\%$	169	0	0	169	0	169
$0\% < R \leq 20\%$	58	23	6	87	5	86
$20\% < R \leq 40\%$	31	41	10	82	9	85
$40\% < R$	61	66	2	129	4	131

発声に関しても同様の傾向であり、オープンデータベースの実験では、訂正の対象となる発声の割合が小さく、訂正の効果が誤り率に現れない傾向にあると結論される。

また、表 23 に示したテストセット全体に対する単語誤り率も、訂正によって 33.1% から 33.3% へと、誤り率が訂正によってむしろわずかに増加しており、オープンデータベースによる訂正の有効性に関して否定的な結論が示唆される。しかしながら、実験で得られた訂正結果の中には、以下の例に示すように誤り率には訂正の効果は現れないが、意味的には正解と同等なものが得られている。

この例では、正解での表現“スケジュールが知りたい”に対して、意味的に近い表現“スケジュールを教えてください”が訂正で用いられており、意味的には誤りが回復されているといえる。しかし単語誤り率で見ると、訂正によって 23%(3/13) から 31%(4/13) へと値が大

表 23: テストセット全体に対する単語誤り率 (日英、オープンデータベース)

訂正前				訂正後	
訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
26.1%	45.5%	24.6%	33.1%	28.6%	33.3%
(689/2636)	(736/1616)	(50/203)	(1475/4455)	(58/203)	(1483/4455)

表 24: 翻訳結果の評価値 (日英、オープンデータベース)

評価値	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
A	190	2	1	193	5	197
B	35	12	3	50	3	50
C	18	17	4	39	4	39
D	75	99	10	184	6	180
翻訳率: $\frac{N_A+N_B+N_C}{N_A+N_B+N_C+N_D}$	76.4%	23.8%	44.4%	60.5%	66.7%	61.4%

きくなっている。このことから、誤り率は正解との表層的な照合に基づくため、表層に現れない意味レベルでの訂正の効果を測ることはできないことがわかる。したがって、訂正の効果の有無を単語誤り率のみによって結論するのは不十分であるといえる。

誤り率に現れないが意味的に有効な訂正 (1)

正解: そこからラスベガスまでのバスのスケジュールが知りたいのですが
 認識結果: そこからラスベガスまでのバスのスケジュールしたいのですが
 訂正結果: そこからラスベガスまでのバスのスケジュールを教えてくださいたいのです

6.4 翻訳文の主観評価

ここでは、オープンデータベースを用いた訂正手法の音声翻訳における有効性を検証するために、音声認識結果および訂正結果をもとに機械翻訳された日英翻訳結果に対して主観評価を行なう。比較のために、翻訳および評価方法については3と完全に同じ方法を用いた。音声認識結果、およびその訂正結果を入力として翻訳システム (TDMT) を用いて得られた翻訳結果に関する主観評価を表 24に示す。

まず、テストセット全体に対する翻訳率は訂正によって 60.5% から 61.4% へとわずかながら 0.9% の改善が見られる。さらに、訂正可能と判断された発声に関しては、44.4% から 66.7% へと増加している。この結果は、先に示した認識結果に対する単語誤り率における傾向と相反している。このことは、表層的な誤り率には現れないが翻訳結果を意味レベルで回復するような訂正結果が存在することによって説明される。以下にそのような実例を示す。

表 25: 翻訳結果の単語誤り率 (日英、オープンデータベース)

誤り率 R	訂正前				訂正後	
	訂正不要	訂正不可	訂正可能	合計	訂正可能	合計
$R = 0\%$	190	1	1	192	1	192
$0\% < R \leq 20\%$	20	11	4	35	5	36
$20\% < R \leq 40\%$	27	15	3	45	3	45
$40\% < R$	82	103	10	195	9	194
$0\% \leq R \leq 40\%$ の割合	74.3%	20.8%	44.4%	58.2%	50.0%	58.5%

誤り率に現れないが意味的に有効な訂正 (2)

正解:	えラスベガスからロサンジェルスまでの運賃はおいくらくらいですか
認識結果:	ラスベガスからロサンジェルスまでの音痴なおいくらくらいですか
訂正結果:	ラスベガスからロサンゼルスまでの料金はおいくらくらいですか
正解翻訳:	"Oh, how much is the fare from Las Vegas to Los Angeles?"
認識翻訳:	"Tone-deaf from Las Vegas to Los Angeles — how much is it?"
訂正翻訳:	"How much is the charge from Las Vegas to Los Angeles?"

例はすでに3に示した例と同じ認識結果であるが、オープンデータベース中には正解と一致する表現“運賃はおいくら”を含む用例が存在しないため、訂正では意味的に近い表現“料金はおいくら”が用いられている。認識結果の単語誤り率は、訂正によって23%(3/13)から31%(4/13)へと増加しているが、訂正結果の翻訳結果は正解と意味的にほぼ同等であり、主観評価ではdからaへ改善している。

6.5 翻訳結果の単語誤り率に基づく評価

テストセットの翻訳結果に対する単語誤り率 R の値を4段階に分けて、各発声の誤り率 R に関する分布を表25に示す。テストセット全体に対する誤り率 $0\% \leq R \leq 40\%$ の発声の割合は、訂正前後で58.3%から58.5%と変化が見られない。表24の主観評価の結果との比較から、翻訳結果に対する訂正の効果は主観評価では現れているが、誤り率では現れていないことが分かる。

次の例では、認識結果の“シングルは増し”に対して、訂正結果の“シングルが満室”が得られ、正解と意味的にほぼ同等な訂正結果および翻訳結果を得ている。しかし、認識結果の単語誤り率は変化せず、また削除誤りの生じた部分“お待たせいたしました”に対する翻訳“I'm sorry to have kept you waiting.”の部分は訂正によって回復しないため、翻訳の誤り率にも訂正の効果は現れない。このことから、オープンデータベースを用いた誤り訂正では、誤り率に効

果が現れにくい、意味的には正解とほぼ同等となるような訂正結果が得られる傾向があるといえる。

翻訳結果の誤り率に現れないが意味的に有効な訂正

正解:	お待たせいたしました申し訳ございませんシングルは満室となっております
認識結果:	申し訳ございませんシングルは増しとなっております
訂正結果:	申し訳ございませんシングルが満室となっております
正解翻訳:	"I'm sorry to have kept you waiting . I'm sorry , singles are all occupied ."
認識翻訳:	"I'm sorry , singles are the increase ."
訂正翻訳:	"I'm sorry , singles are all occupied ."

6.6 オープンデータベースにおける問題

オープンデータベースを用いた訂正では、入力 of 正解がデータベース中に用例として含まれることが保証されない。このような条件下では、(1) 入力の誤りを適切に訂正するための有用な用例がデータベース中に存在しないために訂正結果が得られない、もしくは、(2) 不適切な用例の適用により、正解と意味の異なった訂正結果を得る、といった問題に直面する。前者は、訂正必要と判断された発声に対する訂正可能の発声の割合に顕著に現れており、また後者は、以下に示す事例の通りである。

オープンデータベースによる不適切な訂正結果 (1)

正解:	はいよく撮っていただけると思います
認識結果:	はい約とっていただけると思います
訂正結果:	はい味わっていただけると思います
正解翻訳:	"Yes , I think you could take a photograph well ."
認識翻訳:	"Yes , I think could take about ."
訂正翻訳:	"Yes , I think you can taste ."

オープンデータベースによる不適切な訂正結果 (2)

正解:	すみませんできればバス付の部屋がいいのですが
認識結果:	すみませんできればバス付の部屋ないのですが
訂正結果:	すみませんバス付の部屋は無いのですが
正解翻訳:	"Excuse me , if possible , a room with a bath is good ."
認識翻訳:	"Excuse me , if possible , there isn't a room with a bath ."
訂正翻訳:	"Excuse me , there isn't a room with a bath ."

1 番目の例では、“よく撮って”に対して認識結果は“約とって”となっており、両者は音韻的に非常に近い。しかし、データベース中に正解の用例が含まれていないため、訂正結果では

別の用例からの“味わって”が用いられている。また、2番目の例では、“部屋がいい”に対して認識結果では“部屋ない”となっている。訂正結果ではこの誤り部分“部屋ない”を“部屋は無い”に置換している。これらの訂正結果は、正解とは意味が異なるが、音韻的距離と意味的距離の判断に基づき妥当な訂正結果と判断されている。このような(1)用例の不足による訂正の失敗、および(2)意味の異なる誤った訂正の出力、といったオープンデータベースで顕著な問題は、今後、用例の不足の解消や訂正候補の妥当性判断の高精度化などの課題として解決されて行かなければならない。

6.7 データベースのオープン性

実験では、入力に用いた会話をデータベースから排除したものをオープンデータベースとして用いている。ここで、我々は同一もしくは類似した用例がデータベース中の複数会話に出現することを仮定している。このデータベースにおける用例の冗長性の仮定が著しく破れていると仮定すると、オープンデータベースには入力と同一もしくは類似した用例が存在しないことになる。この場合、誤り訂正の前提であるデータベースにおける入力の誤り訂正に有用な用例の存在が保証されないため、誤り訂正実験において有効な結果は期待できない。

入力に対してデータベース中に同一、もしくは類似した用例がどの程度存在するかを調べる方法としては、データベース中のすべての用例の中から、入力に対する類似度の高いものを求めるような方法が考えられる。入力と用例の類似度としては、DP マッチングを用いて計算される編集距離や、(連続)一致単語列の長さなどを用いることが考えられる。データベースのオープン性の評価は検討中であり、誤り訂正の効果と使用するデータベースのオープン性との関係については、今後さらに検証を行っていく必要がある。

7 おわりに

本稿では、テキストコーパスを用いた誤り訂正手法の提案を行ない、音声翻訳における実験結果を示しながら提案手法の有効性について検証を行なった。実験では、誤りの品詞、内容語、機能語によらずに訂正の効果が得られたが、“削除”誤りに対する訂正の効果は“挿入”、“置換”誤りに対する効果に比べて低いという傾向が見られた。

また、認識結果が訂正可能かどうかは、その誤り率と密接に関係しており、本研究の実験条件では単語誤り率が20%程度の認識結果を中心に訂正結果が得られることが分かった。誤り率の低い認識結果に対しては訂正は容易であるが、同時に訂正不要と判断される場合も多い。また、翻訳結果における誤りの程度も軽いため、これらは訂正による翻訳率の改善にあまり寄与していない。

提案手法の言語によらない有効性は、日英、日韓、英日方向での音声翻訳における誤り訂正実験によって実証された。認識誤りの翻訳結果への影響は翻訳方向によって異なり、翻訳言語対の言語間距離が大きいほど、認識誤りの翻訳結果に与える影響も大きい。

入力に対してオープンなデータベースを用いた場合の有効性も、同様な実験によって確認された。オープンデータベースを用いた誤り訂正では、意味的に正解とほぼ同等となるような訂正結果が得られる傾向が見られた。このため、訂正の効果は誤り率には現れにくく、翻訳結果の主観評価において確認された。

今後、実時間自由発話の音声翻訳への適用に向けて、提案手法が現状で直面している問題として、(1) データベースに対してオープンな入力に対する訂正、(2) 翻訳トレーニングセットに対してオープンな入力に対する訂正、(3) 実時間音声翻訳に見合った動作速度の実現の三つが挙げられる。(1)、(2)に関しては、データベース中の用例のより効果的な利用方法、および訂正必要性の判断、妥当性の判断における信頼性の向上について検討を進める必要がある。後者に関しては、現在用いている意味的距離、音韻的距離にさらに、音響モデル、言語モデルでの尤度を組み合わせたり、発話状況、文脈、話題などの情報も考慮するなどの発展が考えられる。本稿では、(1)に関して提案手法の効果を示すための評価を行なったのみである。(2)、(3)に関しては今後の研究課題として検討中である。

謝辞

本研究を行なう機会を与えて頂いた ATR 音声翻訳通信研究所の山本誠一社長、第 3 研究室の白井諭室長に深く感謝いたします。また、有意義なコメントを頂きました ATR 音声翻訳通信研究所の皆様、角川類語新辞典を提供して頂きました角川書店に感謝いたします。

参考文献

- Ishikawa, K. and Sumita, E. (1999). "Error Correction Translation Using Text Corpora." In *Proceedings of EUROSPEECH'99*, Vol. 5, pp. 1995-1998.
- Ishikawa, K., Sumita, E., and Iida, H. (1998). "Example-Based Error Recovery Method for Speech Translation: Repairing Sub-Trees According to the Semantic Distance." In *Proceedings of the 5th ICSLP*, Vol. 4, pp. 1147-1150.
- Kaki, S., Sumita, E., and Iida, H. (1998). "A Method for Correcting Errors in Speech Recognition Using the Statistical Features of Character Co-occurrence." In *Proceedings of COLING'98*, pp. 653-657.
- Kukich, K. (1992). "Techniques for Automatically Correcting Words in Text." *ACM Computing Surveys*, 24 (4), 377-439.
- Lepage, Y. (1997). "String Approximate Pattern-Matting." 情報処理学会第 55 回全国大会, Vol. 3, pp. 139-140.
- Sumita, E., Yamada, S., Yamamoto, K., Paul, M., Kashioka, H., Ishikawa, K., and Shirai, S. (1999). "Solutions to Problems Inherent in Spoken-language Translation: The ATR-MATRIX Approach." In *Proceedings of MT-SUMMIT'99*, Vol. 1, pp. 1-4.
- Takezawa, T., Sugaya, F., Yokoo, A., and Yamamoto, S. (1999). "A New Evaluation Method for Speech Translation Systems and a Case Study on ATR-MATRIX from Japanese to English." In *Proceedings of MT-SUMMIT'99*, pp. 299-307.
- Tsukada, H., Yamamoto, H., Takezawa, T., and Sagisaka, Y. (1998). "Gramatical Word Graph Re-Generation for Spontaneous Speech Recognition." In *Proceedings of ICSLP98*, Vol. 5.
- Waibel, Alex, e. (1996). "JANUS-II Translation of Spontaneous Conversational Speech." In *Proceedings of ICASSP*.

- 竹内孔一 松本裕治 (1999). “統計的言語モデルを用いた OCR 誤り訂正システムの構築.” 情報処理学会論文誌, **40** (6), 2679–2689.
- 新納浩幸 (1999). “平仮名 N-gram による平仮名列の誤り検出とその修正.” 情報処理学会論文誌, **40** (6), 2690–2698.
- 永田昌明 (1998). “文字類似度と統計的言語モデルを用いた日本語文字認識誤り訂正手法.” 電子情報通信学会論文誌, **J81-D-II** (11), 2624–2634.
- 大野晋 浜西正人 (1981). 角川類語新辞典. 角川書店.
- 古瀬蔵, 山本和英, 山田節夫 (1999). “構成素境界解析を用いた多言語話し言葉翻訳.” 自然言語処理, **6** (5), 63–91.
- 脇田由美, 河井淳, 飯田仁 (1998). “意味的類似性を用いた音声認識正解部分の特定法.” 自然言語処理, **5** (4), 111–125.

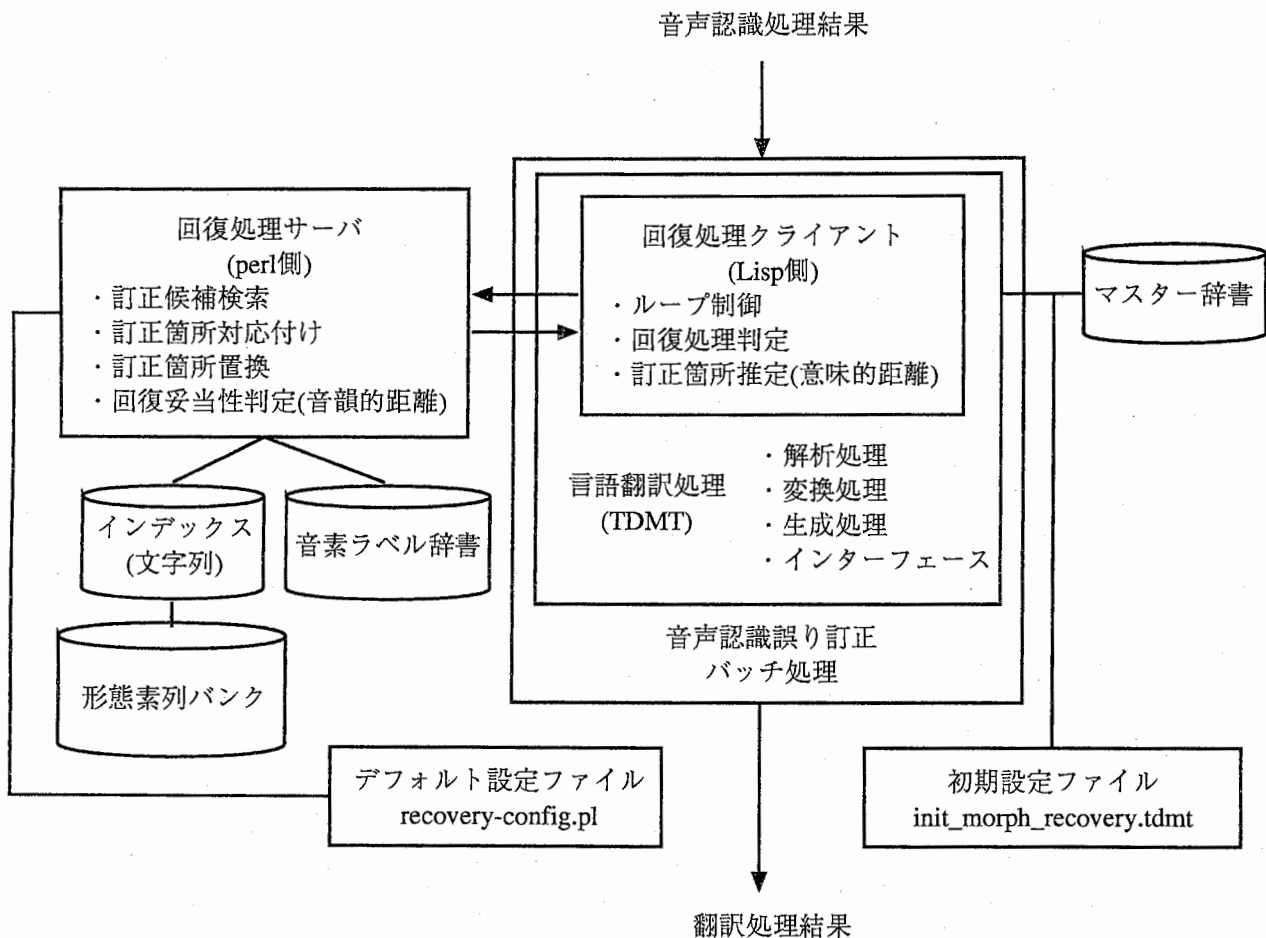


図 3: システム構成図

付録 A 音声認識誤り回復実験ツール

A.1 概要

本ツールは音声認識結果に対して誤り回復処理を施した場合の効果を確かめるためのツールである。システムの構成を図 3 に示す。

具体的には、(1) 正解部分特定処理、(2) 特定された正解部分に基づく誤り部分の翻訳知識を用いた修正、を行なうことにより、音声認識結果の誤り回復処理を実現している。音声認識結果の誤り回復処理は、予め用意された正解文に関する TDMT の原言語形態素列バンクを検索することによって、誤りを含む原言語形態素列を部分的に修正することにより行なわれる。ここでは、この手法を検証するための実験ツールについて説明する。

A.2 ディレクトリ構成

\$IF_DIR TDMT インターフェイスのトップディレクトリ

環境変数 **IF_DIR** または、`bin/RECOVERY/recovery-config.pl` に定義された **\$IF_DIR**

に指定されたもの。(ただし、環境変数を優先。) RECOVER 処理のために、この下にある次のツールを利用している。以前、A-filter の一部として提供されていたマスター辞書引きの機能は入力フィルタに置き換えられている。

tdmt_input_filter.csh	入力フィルタ (マスター辞書引き)
tdmt_input_filter.pl	入力フィルタ (マスター辞書引き)
tdmt_output_filter.csh	出力フィルタ (internal から EUC へ変換)
tdmt_output_filter.pl	出力フィルタ (internal から EUC へ変換)
tdmt_of_internal.csh	出力フィルタ (internal のまま無変換)
TDMT_MULTI.SunOS (or TDMT_MULTI.OSF1)	TDMT 実行イメージ
TDMT_MULTI.csh	TDMT 実行イメージの wrapper

\$CHK_SHEET_DIR CHK_SHEET ツールのトップディレクトリ (\$IF_DIR/chk_sheet)

RECOVER 処理のために、この下にある次のツールを利用している。

tdmt_internal.csh
bat_morph_recovery_tdmt.csh リカバリのバッチ処理

これには次の設定ファイルが必要。

init_morph_recovery_tdmt 初期設定ファイル
partial_off_tdmt 部分翻訳モードオフコマンド
separate_on_tdmt 分割翻訳モードオンコマンド
quit_tdmt バッチ終了コマンド

\$IF_DIR/code MORPH_RECOVERY ツールのディレクトリ

RECOVER 処理のために、この下にある次のツールを利用している。これらが Lisp 側で TDMT とのインターフェースを確立する。

morph-recovery.lisp(.fasl)
morph-recovery-interface.lisp(.fasl)

ここにあるその他の TDMT インターフェイス用ファイルも必要。

\$RECOVER_DIR 音声認識誤り回復処理のトップディレクトリ (\$IF_DIR/bin/RECOVERY)

DIC	辞書、テーブル用ディレクトリ
recovery.pl	リカバリサーバのトップレベル Perl スクリプト
recovery-config.pl	リカバリサーバの設定ファイル
util.pl	ユーティリティ (汎用リストなど)
source-word.pl	形態素クラス
morph-db.pl	形態素バンクのクラス
recovery-mode.pl	リカバリモードの管理クラス
recovery-replace.pl	対応置換モジュール
phone-lex.pl	音素ラベル辞書クラス

`$IF_DIR/bin/agrep_tool` 外部ツール

Agrep	曖昧検索ツール (Sun 用)
Agrep.alpha	曖昧検索ツール (Alpha 用)

`$IF_DIR/bin/dpm_tool` 外部ツール

dpm	DP マッチングツール (Sun 用)
dpm.alpha	DP マッチングツール (Alpha 用)

A.3 設定方法

(A.3.1) 環境変数の設定

環境変数 `IF_DIR` を TDMT インターフェースのトップディレクトリとして設定することにより、相対的に音声認識誤り回復処理のトップディレクトリが指定される。もし、環境変数 `IF_DIR` が設定されていない場合は、コンフィギュレーションファイル (`recovery-config.pl`) に設定された値がデフォルト値として使用される。

(A.3.2) TDMT インターフェースの設定

本ツールは TDMT インターフェイスと協調して動作するので、あらかじめ TDMT インターフェイスがインストールされていなければならない。また、本ツールは TDMT 体系の形態素入力を仮定しているので、TDMT インターフェイスは、TDMT 体系の設定になっている必要がある。なお、TDMT のバージョンは 1999/02/01 のものを使用することとする。インターフェイスの設定は、以下の `morph-recovery` ツールに対して行なう。

morph-recovery.lisp	(ソースコード)
morph-recovery.fasl	(Sun 用バイナリーコード)
morph-recovery.afasl	(Alpha 用バイナリーコード)
morph-recovery-interface.lisp	(ソースコード)
morph-recovery-interface.fasl	(Sun 用バイナリーコード)
morph-recovery-interface.afasl	(Alpha 用バイナリーコード)

これらは \$IF_DIR/code/ の下になければならない。

(A.3.3) コンフィギュレーションファイルの設定

ツールや辞書のディレクトリなどのデフォルト設定は、コンフィギュレーションファイル (recovery-config.pl) に設定する。詳細についてはコンフィギュレーションファイル内のコメントを参照。また、起動後に TDMT 側から送信される初期設定は、init_morph_recovery.tdmt に TDMTCommand 形式で設定する。詳細は init_morph_recovery.tdmt の設定例を参照。

A.4 操作方法

(A.4.1) 音声認識誤り回復処理クライアント

- NAME

bat_morph_recovery.tdmt.pl - Batch to recover the morphological seq.

- SYNOPSIS

bat_morph_recovery.tdmt.csh INPUT_FILE

- DESCRIPTION

初期設定 ファイル (init_morph_recovery.tdmt) に設定された実験環境の元で音声認識誤り回復処理を実行する。音声認識結果である INPUT_FILE の内容を順次読み込んで誤り回復処理機能付き TDMT を実行する。現在の実装では、誤り回復処理は外部でサーバ化されており、サーバと TDMT が名前付きパイプを通じて通信する仕組みになっている。そのため、この処理を実行するためには、誤り訂正サーバ (recovery.pl) が起動されている必要がある。(起動する順序はどちらが先でもよい。)

(A.4.2) 音声認識誤り回復処理サーバ

- NAME

recovery.pl - Batch to recovery the morphological seq. as server

- SYNOPSIS

recovery.pl

● DESCRIPTION

コンフィグレーションファイル (recovery-config.pl) に設定された環境で音声認識誤り回復処理を実行するサーバ。クライアントとは名前付きパイプで通信を行なう。クライアントの指示により一部環境を動的に変更することが可能である。これを利用して翻訳方向を切替えたり、音韻距離の閾値を変更することができる。

A.5 実装説明

(A.5.1) 音声認識誤り回復処理サーバ

perl で記述されている。音声認識誤り回復処理実験に必要な基本的手順をバッチ処理としてまとめたもの。クライアント部分とサーバ部分からなる。コンフィギュレーションファイル、および初期設定ファイルに指定された設定に従って処理を行う。

(A.5.2) 音声認識誤り回復処理サーバのイベント仕様

回復処理サーバは入力用の名前付きパイプから、"EVENT=VALUE"形式の文字列を受け取るとそれをイベントとして認識し、処理を実行する。実行結果は出力用の名前付きパイプからクライアント側へ返される。ただし、先頭が"#"で始まる行は無視される。

1. 音韻距離閾値の設定

イベント名: SET_TH_PHONE

内部関数名: setThPhone

処理: 音韻距離を動的に再設定する。

入力値: float

出力値: 再設定後の音韻距離閾値 (= 入力値)

2. 翻訳方向の切替え

イベント名: SWITCH_MODE

内部関数名: switchMode

処理: 翻訳方向を動的に切替える。指定された言語対がロードされていない場合は切替えない。

入力値: 翻訳方向ラベル (':J-E', ':E-J', ':J-K', ':J-C')

出力値: 切替え後の翻訳方向ラベル

3. 発話番号情報

イベント名: UTTERANCE

内部関数名: setUtterance

処理: 発話番号情報を受け取り、ログに出力する。

入力値: 発話番号文字列

出力値: 発話番号文字列 (= 入力値)

4. オリジナル形態素列の設定

イベント名: ORG_MORPH

内部関数名: setOrgMorph

処理: オリジナル形態素列情報を受け取り、内部に記憶する。

入力値: 形態素列情報

出力値: (空文字列)

5. 形態素列の回復要求

イベント名: PRE_MORPH

内部関数名: recoverMorph

処理: 形態素列を受け取り、回復処理を行なう。

入力値: 形態素列、訂正箇所推定リストをまとめたもの。

○フォーマット

```
-----
PRE_MORPH := ([MORPH] [WRONG])
MORPH := (:MORPH ([WORD_1] [WORD_2] ...))
WORD_x := (:WORD [WORD_STR]
           :REG-EXP [REG_EXP_STR]
           :POS [POS_SYM]
           :ATTRIBUTE [ATTRIBUTE_LIST]
           :SEM-CODE [SEM_CODE_LIST]
           :COMPOUND [COMPOUND_LIST])
WORD_STR := string
REG-EXP := string
POS := symbol
ATTRIBUTE_LIST := list(attribute information)
```

```

SEM_CODE_LIST := list(sem-code information)
COMPOUND := list(compound information)
WRONG := (:WRONG ([RANGE_1] [RANGE_2] ...))
RANGE := ([START] [END] [SEM_DIST])
START := int
END := int
SEM_DIST := float

```

○具体例

```

PRE_MORPH=(((:MORPH ((:WORD "到着" :REG-EXP "到着" :POS サ変名
詞 :ATTRIBUTE NIL :SEM-CODE (386 314) :COMPOUND NIL) (:WORD "が"
:REG-EXP "が" :POS 格助詞 :ATTRIBUTE NIL :SEM-CODE
NIL :COMPOUND NIL) (:WORD "遅く" :REG-EXP "遅い" :POS 形容
詞 :ATTRIBUTE ((:CONJ-FORM . 連用)) :SEM-CODE (156
155) :COMPOUND NIL) (:WORD "なり" :REG-EXP "なる" :POS 本動
詞 :ATTRIBUTE ((:CONJ-FORM . 連用)) :SEM-CODE (283 273 260
250) :COMPOUND NIL) (:WORD "そう" :REG-EXP "そう" :POS 助動
詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND
NIL) (:WORD "な" :REG-EXP "な" :POS 連体助詞 :ATTRIBUTE NIL :SEM-
CODE NIL :COMPOUND NIL) (:WORD "ので" :REG-EXP "ので" :POS 接続助
詞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL) (:WORD "フランク
" :REG-EXP "フランク" :POS 形容名詞 :ATTRIBUTE NIL :SEM-CODE (822
643) :COMPOUND NIL) (:WORD "に" :REG-EXP "に" :POS 格助
詞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL) (:WORD "し
" :REG-EXP "する" :POS 本動詞 :ATTRIBUTE ((:CONJ-FORM . た)) :SEM-
CODE (400 361 250) :COMPOUND NIL) (:WORD "ていただきたい" :REG-EXP
"ていただきたい" :POS 助動詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-
CODE NIL :COMPOUND NIL) (:WORD "のです" :REG-EXP "のです" :POS 助
動詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND
NIL) (:WORD "けれども" :REG-EXP "けれども" :POS 接続助詞 :ATTRIBUTE
NIL :SEM-CODE NIL :COMPOUND NIL))) (:WRONG ((7 10 1.8666668) (7 11
2.227778) (7 12 2.227783) (0 12 2.6444547))))

```

出力値： 回復後の形態素列、音韻距離

複数値を返し、エンドマークとして、POST_MORPH=:END を出力する。

○フォーマット

```
POST_MORPH := ([MORPH] :PH-DIST [PH_DIST])
```

```
MORPH := (:MORPH ([WORD_1] [WORD_2] ...))
```

```
WORD_x := (:WORD [WORD_STR]
```

```
      :REG-EXP [REG_EXP_STR]
```

```
      :POS [POS_SYM]
```

```
      :ATTRIBUTE [ATTRIBUTE_LIST]
```

```
      :SEM-CODE [SEM_CODE_LIST]
```

```
      :COMPOUND [COMPOUND_LIST])
```

```
WORD_STR := string
```

```
REG-EXP := string
```

```
POS := symbol
```

```
ATTRIBUTE_LIST := list(attribute information)
```

```
SEM_CODE_LIST := list(sem-code information)
```

```
COMPOUND := list(compound information)
```

```
PH_DIST := float(phonetic distance)
```

○具体例

```
POST_MORPH=((:MORPH ((:WORD "到着" :REG-EXP "到着" :POS サ変名
```

```
詞 :ATTRIBUTE NIL :SEM-CODE (386 314) :COMPOUND NIL) (:WORD "が
```

```
" :REG-EXP "が" :POS 格助詞 :ATTRIBUTE NIL :SEM-CODE
```

```
NIL :COMPOUND NIL) (:WORD "遅く" :REG-EXP "遅い" :POS 形容
```

```
詞 :ATTRIBUTE ((:CONJ-FORM . 連用)) :SEM-CODE (156
```

```
155) :COMPOUND NIL) (:WORD "なり" :REG-EXP "なる" :POS 本動
```

```
詞 :ATTRIBUTE ((:CONJ-FORM . 連用)) :SEM-CODE (283 273 260
```

```
250) :COMPOUND NIL) (:WORD "そう" :REG-EXP "そう" :POS 助動
```

```
詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND
```

```
NIL) (:WORD "な" :REG-EXP "な" :POS 連体助詞 :ATTRIBUTE NIL :SEM-
```

```
CODE NIL :COMPOUND NIL) (:WORD "ので" :REG-EXP "ので" :POS 接続助
```

```
詞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL) (:WORD "に
```

```
" :REG-EXP "に" :POS 格助詞 :ATTRIBUTE NIL :SEM-CODE
NIL :COMPOUND NIL) (:WORD "来" :REG-EXP "来る" :POS 本動
詞 :ATTRIBUTE ((:CONJ-FORM . た)) :SEM-CODE (312 283) :COMPOUND
NIL) (:WORD "ていただきたい" :REG-EXP "ていただきたい" :POS 助動
詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND
NIL) (:WORD "のです" :REG-EXP "のです" :POS 助動詞 :ATTRIBUTE
((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND NIL) (:WORD "けれど
も" :REG-EXP "けれども" :POS 接続助詞 :ATTRIBUTE NIL :SEM-CODE
NIL :COMPOUND NIL)) :PH-DIST 0.169492)
POST_MORPH=( :MORPH ((:WORD "到着" :REG-EXP "到着" :POS サ変名
詞 :ATTRIBUTE NIL :SEM-CODE (386 314) :COMPOUND NIL) (:WORD "が
" :REG-EXP "が" :POS 格助詞 :ATTRIBUTE NIL :SEM-CODE
NIL :COMPOUND NIL) (:WORD "遅く" :REG-EXP "遅い" :POS 形容
詞 :ATTRIBUTE ((:CONJ-FORM . 連用)) :SEM-CODE (156
155) :COMPOUND NIL) (:WORD "なり" :REG-EXP "なる" :POS 本動
詞 :ATTRIBUTE ((:CONJ-FORM . 連用)) :SEM-CODE (283 273 260
250) :COMPOUND NIL) (:WORD "そう" :REG-EXP "そう" :POS 助動
詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND
NIL) (:WORD "な" :REG-EXP "な" :POS 連体助詞 :ATTRIBUTE NIL :SEM-
CODE NIL :COMPOUND NIL) (:WORD "ので" :REG-EXP "ので" :POS 接続助
詞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL) (:WORD "ギャラン
ティにし" :REG-EXP "ギャランティにする" :POS 本動詞 :ATTRIBUTE
((:CONJ-FORM . た)) :SEM-CODE (448) :COMPOUND NIL) (:WORD "ていた
だきたい" :REG-EXP "ていただきたい" :POS 助動詞 :ATTRIBUTE ((:CONJ-
FORM . 基本)) :SEM-CODE NIL :COMPOUND NIL) (:WORD "のです" :REG-
EXP "のです" :POS 助動詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE
NIL :COMPOUND NIL) (:WORD "けれども" :REG-EXP "けれども" :POS 接続
助詞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL)) :PH-DIST
0.102941)
```

6. 未定義イベント

イベント名： 未定義のイベントすべて

内部関数名： undefinedFunc

処理： サポートしないイベントの処理を行なう。実際には何も処理を行わずに、イベントをスルーする。

入力値： (無視)

出力値： (= 入力値)

(A.5.3) 外部処理

このツールでは次の2つの外部処理を呼出している。

1. 曖昧検索ツール (Agrep)
2. DP マッチング (dpm)

これらは共に C 言語で書かれたツールであり、独立したプロセスとして実行されるようになっている。これらはバイナリがマシンアーキテクチャに依存するので、動作環境によってバイナリを切替える必要がある。動作環境は SunOS と OSF1 を識別して、自動的に切替えるようになっている。

(A.5.4) 多言語対応

多言語の翻訳に対応するためには、次の項目を考慮する必要がある。

1. 翻訳方向の動的切替え
2. マスター辞書、ラベル辞書、形態素バンクの切替え
3. 処理過程での文字コードのエンコーディング対応

以上を実現するために、RecoveryMode という管理用のクラスを用いている。回復処理は基本的に原言語側の形態素列に対して行なわれるため、一見、原言語側が同じであれば切替える必要はないように思われる。しかし、実際には翻訳知識が言語対に依存しているため、形態素列もそれに適した変換をうけるようになっている。従って、回復処理の管理クラスも言語対ごとにインスタンスを生成するように実装している。そして、各言語対のインスタンスを切替えることで、翻訳方向を動的に切替えることができる。例えば、:J-E と :J-K はそれぞれ別のインスタンスにより管理される。こうした場合、現在の実装では、辞書などが共通であったとしても、現在は独立にロードされメモリ空間を圧迫することになる。(この問題は将来、重複ロードをさける仕組みを入れることで回避できる。)

TDMT は内部で MULE の INTERNAL コードを使用しているが、誤り訂正処理サーバ内では、MULE の INTERNAL コードでなく、通常の EUC-JAPAN などを使用する。これはマスター辞書や形態素列バンクで INTERNAL コードを使用しない方が一般の UNIX ツールにとっ

ては文字化けなどの危険が少なく都合が良いためである。従って、処理の過程では、これらの文字コードのエンコーディングをサポートする必要がある。この切替えも言語対の管理クラスがサポートする。現在、サポート、またはサポートする準備のできている言語対は次のとおりである。()内は MULE の INTERNAL コードで使われる国番号。ただし、英語は通常の ASCII なので存在せず、ドイツ語ではウムラウトなど特殊文字のために ISO-8859-1 で規定された文字のみで使用する。

- | | | |
|------|-----------------------|--------------------------|
| :J-E | 日本語 (0x92/EUC-JAPAN) | → 英語 (-) |
| :J-K | 日本語 (0x92/EUC-JAPAN) | → 韓国語 (0x93/EUC-KOREAN) |
| :J-G | 日本語 (0x92/EUC-JAPAN) | → ドイツ語 (0x81/ISO-8859-1) |
| :J-C | 日本語 (0x92/EUC-JAPAN) | → 中国語 (0x91/EUC-CHINESE) |
| :E-J | 英語 (-) | → 日本語 (0x92/EUC-JAPAN) |
| :K-J | 韓国語 (0x93/EUC-KOREAN) | → 日本語 (0x92/EUC-JAPAN) |

付録 B 設定ファイル

B.1 回復処理サーバ側設定ファイル

recovery-config.pl

回復処理サーバのデフォルトの設定を行なう。一部の設定は、後でクライアントからのコマンド発行で動的に変更できる。

(B.1.1) 解説

- 実験環境

\$IF_DIR ツールのトップディレクトリの指定。

\$THRESHOLD_SEM_DIST 意味距離閾値の設定。現在は Perl 側では意味距離値は使用しないのでこの値は無視される。

\$THRESHOLD_PHONE_DIST 音韻距離閾値の設定。制御コマンドにより、後から動的に変更できる。

\$THRESHOLD_REPLACE_DIST 置換範囲制約の設定。対応範囲を自動識別するときに許される曖昧性を指定する。

\$TH_PH_EXP_GLOBAL, \$TH_PH_EXP_LOCAL 音素展開閾値の設定。各形態素を音素展開したときの最大数 (LOCAL) と、形態素列全体を音素展開したときの最大数 (GLOBAL) を指定することにより、音素展開の爆発を抑制する。
\$TH_PH_EXP_GLOBAL が優位に働く。

\$TH_AGREP_RATIO, \$TH_AGREP_DIFF インデックス検索閾値の設定。検索クエリーの長さに対する曖昧性の比率 (RATIO) と、曖昧性として許される文字列の絶対的なバイト数 (DIFF) を用いて、曖昧検索ツールに対する制約を与える。
\$TH_AGREP_DIFF が優位に働く。

- RECOVER 処理ツールのパス設定

\$CHK_SHEET_DIR CHK_SHEET 処理のトップディレクトリの設定。

\$RECOVER_DIR RECOVER 処理のトップディレクトリの設定。

\$RECOVER_BIN_DIR RECOVER 処理ツールのトップディレクトリの設定。

\$RECOVER_DIC_DIR RECOVER 処理ツールの辞書ディレクトリの設定。

\$RECOVER_TMP_DIR RECOVER 処理ツールの TMP ディレクトリの設定。

\$RECOG_DIR 音声認識データのディレクトリの設定。

\$RS_DIR RECOVER 処理結果用のディレクトリの設定。

\$RECOVER_EVAL_DIR RECOVER 処理ツールの EVAL ディレクトリの設定。

- 入出力 I/O

\$PRE_DATA_FILE サーバ入力用の名前付きパイプの設定。

\$POST_DATA_FILE サーバ出力用の名前付きパイプの設定。

- テストセット

\$RECOG_FILE, \$SPREC_FILE 音声認識データのファイル名の設定。現在は未使用。

- ロードする言語セット

LANG_PAIRS サーバ起動時にロードする言語対セットの設定。ここでロードしたものから翻訳方向を選択切替えできる。デフォルトでは最後にロードしたものがカレントの翻訳方向になる。

○設定例

```
@LANG_PAIRS = (':J-E', ':J-K', ':J-G', ':J-C', ':E-J', ':K-J');
```

\$LEX_FILELANG_PAIR ラベル辞書の設定。LANG_PAIR(':J-E' など) で指定された言語セットに対して指定する。

\$CHAR_LABELS_FILELANG_PAIR 文字ラベル辞書の設定。LANG_PAIR(':J-E' など) で指定された言語セットに対して指定する。

\$MORPH_BANK_FILELANG_PAIR 単独ファイルの場合のファイル指定。

LANG_PAIR(':J-E' など) で指定された言語セットに対して指定する。

\$MORPH_BANK_INDEX_FILELANG_PAIR 単独ファイルの場合のインデックスファイル指定。LANG_PAIR(':J-E' など) で指定された言語セットに対して指定する。

\$PHONE_PATCH_FILELANG_PAIR 音素ラベル変換のパッチファイルの設定。

LANG_PAIR(':J-E' など) で指定された言語セットに対して指定する。

- 外部ツール

\$OS_TYPE 自動識別された OS の種類がセットされる。

\$DPM_COM dpm ツールのパスが設定される。\$OS_TYPE により自動設定される。

\$AGREP_COM Agrep ツールのパスが設定される。\$OS_TYPE により自動設定される。

(B.1.2) 具体例

```
-----
#!/usr/local/bin/perl
# -*- Mode: perl -*-

package RecoveryConfig;

#####
# 実験環境 #
#####
### TDMT のインターフェースのトップディレクトリ ###
# $IF_DIR/setenv_if.csh で定義される環境変数 IF_DIR
# と一致していなければならない。
# 外部で定義された環境変数を優先して使用する。
if($ENV{"IF_DIR"} eq ""){
#   $IF_DIR = "/usr1/kawai/ATR/data/tdmt.980216";
#   $IF_DIR = "/data/as06/kawai/tdmt.980216";
    $IF_DIR = "/home/as41/kai_ishi/tdmt.980216.fb";
} else {
    $IF_DIR = $ENV{"IF_DIR"};
}

### 意味距離閾値 ###
$THRESHOLD_SEM_DIST = 0.3;

### 音韻距離閾値 ###
$THRESHOLD_PHONE_DIST = 0.3;

### 置換範囲制約 ###
$THRESHOLD_REPLACE_DIST = 0.4;

### 音素展開閾値 ###
```

```
$TH_PH_EXP_GLOBAL = 100;
$TH_PH_EXP_LOCAL = 10;

### インデックス検索閾値 ###
$TH_AGREP_RATIO = 0.3;
$TH_AGREP_DIFF = 8;

#####
# RECOVER 処理ツールのパス設定 #
#####
### CHK_SHEET 処理のトップディレクトリ ###
$CHK_SHEET_DIR = $IF_DIR . "/" . "chk_sheet";

### RECOVER 処理のトップディレクトリ ###
$RECOVER_DIR = $IF_DIR . "/" . "bin/RECOVERY";

### RECOVER 処理ツールのトップディレクトリ ###
$RECOVER_BIN_DIR = $RECOVER_DIR . "/" . ".";

### RECOVER 処理ツールの辞書ディレクトリ ###
$RECOVER_DIC_DIR = $RECOVER_DIR . "/" . "DIC";

### RECOVER 処理ツールの TMP ディレクトリ ###
$RECOVER_TMP_DIR = $RECOVER_DIR . "/" . "demo98_tmp";

### 音声認識データのディレクトリ ###
$RECOG_DIR = $RECOVER_DIR . "/" . "tmp";

### RECOVER 処理結果用のディレクトリ ###
$RS_DIR = $RECOVER_DIR . "/" . "tmp";

### RECOVER 処理ツールの EVAL ディレクトリ ###
$RECOVER_EVAL_DIR = $RECOVER_DIR . "/" . "eval";
```

```
### 入出力 I/O ###
# named pipes
$PRE_DATA_FILE = $RECOVER_DIR . "/" . "tmp/=pre-data=";
$POST_DATA_FILE = $RECOVER_DIR . "/" . "tmp/=post-data=";

#####
# テストセット #
#####
### 音声認識データのファイル名 ###
$RECOG_FILE = $RECOG_DIR . "/" . "tmp.res";
$SPREC_FILE = $RECOG_DIR . "/" . "tmp.sprec";

#####
# ロードする言語セット #
#####
# J-E Single Direction
@LANG_PAIRS = (':J-E');
# J-E Bidirection
# @LANG_PAIRS = (':J-E', ':E-J');
# J-Multi Directions
# @LANG_PAIRS = (':J-E', ':J-K', ':J-G', ':J-C');
# ALL Directions
# @LANG_PAIRS = (':J-E', ':J-K', ':J-G', ':J-C', ':E-J', ':K-J');

#####
# 辞書テーブル類 #
#####
### ラベル辞書 ###
$LEX_FILE{':J-E'} = $RECOVER_DIC_DIR . "/" . "LEX.W";
$LEX_FILE{':J-K'} = $RECOVER_DIC_DIR . "/" . "LEX.W";
$LEX_FILE{':J-G'} = $RECOVER_DIC_DIR . "/" . "LEX.W";
$LEX_FILE{':J-C'} = $RECOVER_DIC_DIR . "/" . "LEX.W";
$LEX_FILE{':E-J'} = $RECOVER_DIC_DIR . "/" . "LEX.W";
```

```
$LEX_FILE{' :K-J'} = $RECOVER_DIC_DIR . "/" . "LEX.W";
```

文字ラベル辞書

```
$CHAR_LABELS_FILE{' :J-E'} = $RECOVER_DIC_DIR . "/" . "char-label.txt";  
$CHAR_LABELS_FILE{' :J-K'} = $RECOVER_DIC_DIR . "/" . "char-label.txt";  
$CHAR_LABELS_FILE{' :J-G'} = $RECOVER_DIC_DIR . "/" . "char-label.txt";  
$CHAR_LABELS_FILE{' :J-C'} = $RECOVER_DIC_DIR . "/" . "char-label.txt";  
$CHAR_LABELS_FILE{' :E-J'} = $RECOVER_DIC_DIR . "/" . "char-label.txt";  
$CHAR_LABELS_FILE{' :K-J'} = $RECOVER_DIC_DIR . "/" . "char-label.txt";
```

単独ファイルの場合のファイル指定

```
# $MORPH_BANK_FILE{' :J-E'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
sldb.txt";  
# $MORPH_BANK_FILE{' :J-E'} =  
"/home/ishikawa/work/recover/morph_bank/" . "morph-bank-sldb.txt";  
$MORPH_BANK_FILE{' :J-E'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
9conv.txt";  
$MORPH_BANK_FILE{' :J-K'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
9conv.txt";  
$MORPH_BANK_FILE{' :J-G'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
9conv.txt";  
$MORPH_BANK_FILE{' :J-C'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
9conv.txt";  
$MORPH_BANK_FILE{' :E-J'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
9conv.txt";  
$MORPH_BANK_FILE{' :K-J'} = $RECOVER_DIC_DIR . "/" . "morph-bank-  
9conv.txt";
```

単独ファイルの場合のインデックスファイル指定

```
# $MORPH_BANK_INDEX_FILE{' :J-E'} = $RECOVER_DIC_DIR . "/" .  
"morph-bank-sldb-index.txt";  
#$MORPH_BANK_INDEX_FILE{' :J-E'} =  
"/home/ishikawa/work/recover/morph_bank/" . "morph-bank-sldb-index.txt";  
$MORPH_BANK_INDEX_FILE{' :J-E'} = $RECOVER_DIC_DIR . "/" . "morph-  
bank-9conv-index.txt";
```

```
$MORPH_BANK_INDEX_FILE{'':J-K'} = $RECOVER_DIC_DIR . "/" . "morph-  
bank-9conv-index.txt";  
$MORPH_BANK_INDEX_FILE{'':J-G'} = $RECOVER_DIC_DIR . "/" . "morph-  
bank-9conv-index.txt";  
$MORPH_BANK_INDEX_FILE{'':J-C'} = $RECOVER_DIC_DIR . "/" . "morph-  
bank-9conv-index.txt";  
$MORPH_BANK_INDEX_FILE{'':E-J'} = $RECOVER_DIC_DIR . "/" . "morph-  
bank-9conv-index.txt";  
$MORPH_BANK_INDEX_FILE{'':K-J'} = $RECOVER_DIC_DIR . "/" . "morph-  
bank-9conv-index.txt";
```

音素ラベル変換のパッチファイル

```
$PHONE_PATCH_FILE{'':J-E'} = $RECOVER_DIC_DIR . "/" . "phone-patch-  
j.txt";  
$PHONE_PATCH_FILE{'':J-K'} = $RECOVER_DIC_DIR . "/" . "phone-patch-  
j.txt";  
$PHONE_PATCH_FILE{'':J-G'} = $RECOVER_DIC_DIR . "/" . "phone-patch-  
j.txt";  
$PHONE_PATCH_FILE{'':J-C'} = $RECOVER_DIC_DIR . "/" . "phone-patch-  
j.txt";  
$PHONE_PATCH_FILE{'':E-J'} = $RECOVER_DIC_DIR . "/" . "phone-patch-  
j.txt";  
$PHONE_PATCH_FILE{'':K-J'} = $RECOVER_DIC_DIR . "/" . "phone-patch-  
j.txt";
```

#####

外部ツール

#####

コード変換

内部処理をしているため不要になった

\$I2JKGC_COM = "perl ".\$IF_DIR."/". "i2jkgc.perl";

音素ラベル変換のパッチ

パッチファイルで対応しているため不要になった

\$PHONE_COM = "perl ".\$RECOVER_BIN_DIR."/". "phone.pl";

```
### OS を自動識別 ###
open(COM_TMP, "uname -s |");
$OS_TYPE = <COM_TMP>;
close(COM_TMP);
print "OS_TYPE=$OS_TYPE\n";

if($OS_TYPE =~ /SunOS/){
    # for SUN
    $DPM_COM = $IF_DIR."/bin/dpm_tool/". "dpm";
    $AGREP_COM = $IF_DIR."/bin/agrep_tool/". "Agrep";
}else{
    # for ALPHA
    $DPM_COM = $IF_DIR."/bin/dpm_tool/". "dpm_alpha";
    $AGREP_COM = $IF_DIR."/bin/agrep_tool/". "Agrep_alpha";
}

# as memo
#$AGREP_COM = $IF_DIR."/bin/agrep_tool/". "Agrep_alpha -7 -D1 -I1 -S1";
#ALPHA

1;
### END OF FILE ###
```

B.2 回復処理クライアント側設定ファイル

init_morph_recovery.tdmt

バッチ処理の初期設定を行なう。TDMT サーバ、回復処理サーバに対して設定変更のコマンドを発行する。回復処理サーバへの制御コマンドも TDMT サーバを経由して送信される。この設定はバッチ処理起動時に使用されるので、結果的にサーバ側のデフォルト設定よりも優先される。

(B.2.1) 解説

- TDMTCommand="0 (load (format nil "~a~a\" PUBLIC::tdmt-interface-dir \"code/morph-recovery \"))"

TDMT に対して、morph-recovery に必要なパッチを読み込むことを指示している。

- `TDMTCommand="0 (load (format nil "~a~a\" PUBLIC::*tdmt-interface-dir* \ "code/morph-recovery-interface \ "))"`

TDMT に対して、morph-recovery に必要なパッチを読み込むことを指示している。

- `TDMTCommand="0 (switch-mode :J-E)"`

翻訳方向の切替え。TDMT だけでなく、回復処理サーバに対しても同時に指示を行なう。

- `TDMTCommand="0 (set-nbest-number 1)"`

TDMT の出力 NBEST 数の指定。

- `TDMTCommand="0 (switch-translation-method :separate)"`

翻訳方式の指定。morph-recovery による回復処理を行なう場合は、:separate でなければならない。

- `TDMTCommand="0 (switch-output-function 'output-result-for-morph-recovery)"`

出力関数の指定。morph-recovery による回復処理を行ない、専用の出力を得るための関数を指定する。

- `TDMTCommand="1 (setq *print-pretty* nil)"`

LISP の出力形式に関する設定。

- `TDMTCommand="31 (setf (TRANSFER::rm-loop-num (TRANSFER::current-rm)) 2)"`

morph-recovery による回復処理のループ回数の設定。

- `TDMTCommand="31 (TRANSFER::switch-rm-condition t)"`

morph-recovery による回復処理を切替える。t のときオン。nil のときオフ。

- `TDMTCommand="31 (setf (TRANSFER::rm-sem-th (TRANSFER::current-rm)) .3)"`

morph-recovery による回復処理の意味距離閾値の設定。

- `TDMTCommand="31 (set-th-phone .3)"`

morph-recovery による回復処理の音韻距離閾値の設定。TDMT だけでなく、回復処理サーバに対しても同時に指示を行なう。

(B.2.2) 具体例

```
-----  
TDMTCommand="0 (load (format nil "~a~a\" PUBLIC:*tdmt-interface-dir*  
\"code/morph-recovery\"))"  
TDMTCommand="0 (load (format nil "~a~a\" PUBLIC:*tdmt-interface-dir*  
\"code/morph-recovery-interface\"))"  
TDMTCommand="0 (switch-mode :J-E)"  
TDMTCommand="0 (set-nbest-number 1)"  
# TDMTCommand="0 (switch-translation-method :partial)"  
TDMTCommand="0 (switch-output-function 'output-result-for-morph-  
recovery)"  
TDMTCommand="0 (set-output-tree nil)"  
TDMTCommand="1 (setq *print-pretty* nil)"  
TDMTCommand="21 (set-partial-max-dist .2)"  
TDMTCommand="21 (set-partial-min-len 2)"  
### morph-recovery setting ###  
TDMTCommand="31 (setf (TRANSFER::rm-loop-num (TRANSFER::current-  
rm)) 2)"  
TDMTCommand="31 (TRANSFER::switch-rm-condition t)"  
TDMTCommand="31 (setf (TRANSFER::rm-sem-th (TRANSFER::current-  
rm)) .3)"  
# TDMTCommand="31 (setf (TRANSFER::rm-marker-flag  
(TRANSFER::current-rm)) t)"  
TDMTCommand="31 (set-th-phone .3)"  
-----
```

付録 C 辞書フォーマット

C.1 マスター辞書

以下に示すのは日本語用である。各言語により形態素表現の仕様は異なる。

bin/RECOVERY/DIC/MASTER_LEXICON_TDMT_98NOV21

```
-----
31223 受付者名 | ウケツケシャメイ | 受付者名 | 普通名詞 ||||
31224 承っ | タマワッ | 承る | 本動詞 | 五段ラ | た ||
31225 支払え | シハラエ | 支払える | 本動詞 | 一段 | 連用 ||
31226 風呂場 | ブロバ | 風呂場 | 普通名詞 ||||
31227 番組欄 | バングミラン | 番組欄 | 普通名詞 ||||
31228 当フロント | トウフロント | 当フロント | 普通名詞 ||||
31229 サービス代 | サービスダイ | サービス代 | 普通名詞 ||||
-----
```

C.2 音素ラベル辞書

● 音素ラベル辞書

bin/RECOVERY/DIC/LEX.W

```
-----
5 [UTT-START] - # UTT-START
6 [UTT-END] - # UTT-END
10000 [ありがとうございます] a r i g a t o o g o z a i m a s u {|-}
# ありがとうございます | アリガトウゴザイマス | ありがとうございます |
感動詞 ||||
10001 [ニューヨークシティホテル] n j u u j o o k u s h i t i {i|-}
h o t e r u {|-} # ニューヨークシティホテル | ニューヨークシティホテル |
ニューヨークシティホテル | 普通名詞 ||||
10002 [で] d e {|-} # で | デ | だ | 判定詞 | 形容動詞 | 連用 ||
10003 [ございます] g o z a i m a s u {|-} # ございます | ゴザイマス |
ございます | 補助動詞 | 特殊サ | 基本 ||
10005 [もしもし] m o s h i m o s h i {|-}
# もしもし | モシモシ | もしもし | 感動詞 ||||
10006 [わたし] w a t a s h i {|-} # わたし | ワタシ | わたし | 代名詞 ||||
10007 [田中] t a n a k a {|-} # 田中 | タナカ | 田中 | 普通名詞 ||||
```

```
10008 [弘子] h i r o k o {-} # 弘子|ヒロコ|弘子|普通名詞| | | |
10009 [と] t o {-} # と|ト|と|格助詞| | | |
```

- 音素ラベルサブ辞書

```
bin/RECOVERY/DIC/char-label.txt
```

```
# 平仮名
```

```
00000 [あ] a
```

```
00000 [い] i
```

```
00000 [う] u
```

```
00000 [え] e
```

```
00000 [お] o
```

- 音素ラベルパッチ辞書

```
bin/RECOVERY/DIC/phone-patch-j.txt
```

```
# special
```

```
e ー |e e
```

```
b i ー |b i i|
```

```
sh i ー |sh i i|
```

```
k a ー |k a a|
```

```
ch i ゃ |ch o|
```

```
持 ch i |m o ch i |
```

```
待 ch i |m a ch i |
```

```
比 b e |k u r a b e|
```

C.3 形態素列バンク

- 形態素列バンク

```
bin/RECOVERY/DIC/morph-bank-9conv.txt
```

```
(:MORPH ((:WORD "週末" :REG-EXP "週末" :POS 普通名詞 :ATTRIBUTE
NIL :SEM-CODE (18) :COMPOUND NIL) (:WORD "の" :REG-EXP "の" :POS
連体助詞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL) (:WORD "宿
泊" :REG-EXP "宿泊" :POS サ変名詞 :ATTRIBUTE NIL :SEM-CODE
(352) :COMPOUND NIL) (:WORD "割引" :REG-EXP "割引" :POS サ変名
詞 :ATTRIBUTE NIL :SEM-CODE (263) :COMPOUND NIL) (:WORD "につい
て" :REG-EXP "について" :POS 格助詞 :ATTRIBUTE NIL :SEM-CODE
NIL :COMPOUND NIL) (:WORD "お" :REG-EXP "お" :POS 接頭
辞 :ATTRIBUTE NIL :SEM-CODE NIL :COMPOUND NIL) (:WORD "教え
" :REG-EXP "教える" :POS 本動詞 :ATTRIBUTE ((:CONJ-FORM . た)) :SEM-
CODE NIL :COMPOUND NIL) (:WORD "ていただきたい" :REG-EXP "ていただ
きたい" :POS 助動詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE
NIL :COMPOUND NIL) (:WORD "のです" :REG-EXP "のです" :POS 助動
詞 :ATTRIBUTE ((:CONJ-FORM . 基本)) :SEM-CODE NIL :COMPOUND
NIL) (:WORD "が" :REG-EXP "が" :POS 接続助詞 :ATTRIBUTE NIL :SEM-
CODE NIL :COMPOUND NIL)))
```

● 形態素列バンクインデックス

先頭の数字は対応する形態素列バンク上の位置を示す。

bin/RECOVERY/DIC/morph-bank-9conv-index.txt

```
0 "週末の宿泊割引についてお教えいただきたいのですが"
960 "食事付で 300 ドルぐらいのものがいいのですけれども"
2077 "そうですか"
2192 "29 日をお願いしたいのですけれども"
2785 "はいお願いします"
3006 "鈴木直子と申します"
3489 "2 人でお願いします"
```

付録 D 実験ツール使用例

D.1 回復処理サーバ起動

外部化された回復処理サーバを起動する。デフォルト設定は recovery-config.pl に記述する。

- 実行コマンド (csh 上)

```
-----
cd bin/RECOVERY
perl recovery.pl >! ../../chk_sheet/smp/bar.log
-----
```

- 実行結果 (bar.log)

```
-----
OS_TYPE=SunOS

TH_SEM=0.3
TH_PHONE=0.3
#####
# RM :J-E #
#####
# Lexicon: 'Word-Label Dictionary' PhoneLex=HASH(0x15d1b8)
#   filePath:
# '/usr1/kawai/ATR/data/tdmt.980216/bin/RECOVERY/DIC/LEX.W'
#   total: 7017 words
#   expanded: 142 words
#   patchFilePath:
# '/usr1/kawai/ATR/data/tdmt.980216/bin/RECOVERY/DIC/phone-patch-j.txt'
#   total patches: 99 lines
#   warning level: 0
#   sub lexicon: 'Char-Label Dcitionary' PhoneLex=HASH(0x5e33a4)
# Lexicon: 'Char-Label Dcitionary' PhoneLex=HASH(0x5e33a4)
#   filePath: '/usr1/kawai/ATR/data/tdmt.980216/bin/RECOVERY/DIC/char-
label.txt'
#   total: 161 words
```

```
#    expanded: 0 words
#    patchFilePath: ''
#    total patches: 0 lines
#    warning level: 0
#    sub lexicon: no
Open recovery session
Recovery(0): event=SWITCH_MODE, data size=4, method=switchMode()
SWITCH_MODE=:J-E
outDataSize=16
Recovery(1): event=SET_TH_PHONE, data size=3, method=setThPhone()
SET_TH_PHONE=0.3
outDataSize=16
Recovery(2): event=UTTERANCE, data size=1, method=setUtterance()
UTTERANCE=2
outDataSize=11
Recovery(3): event=ORG_MORPH, data size=1391, method=setOrgMorph()
outDataSize=0
Recovery(4): event=PRE_MORPH, data size=1391, method=recoverMorph()
qList->depthStr(1)=(A A A A)
AGREP COM: /usr1/kawai/ATR/data/tdmt.980216/bin/agrep_tool/Agrep -7 -D1
-I1 -S1 フランクにしていだきたい </usr1/kawai/ATR/data/tdmt.980216/
bin/RECOVERY/DIC/morph-bank-9conv-index.txt|
AGREP COM: /usr1/kawai/ATR/data/tdmt.980216/bin/agrep_tool/Agrep -8 -D1
-I1 -S1 フランクにしていだきたいのです </usr1/kawai/ATR/data/
tdmt.980216/bin/RECOVERY/DIC/morph-bank-9conv-index.txt|
AGREP COM: /usr1/kawai/ATR/data/tdmt.980216/bin/agrep_tool/Agrep -8 -D1
-I1 -S1 フランクにしていだきたいのですけれども </usr1/kawai/ATR/data/
tdmt.980216/bin/RECOVERY/DIC/morph-bank-9conv-index.txt|
(フランク に していだきたい のです けれども)=>(に 来 ていだきたい
のです けれども), 0.278543
AGREP COM: /usr1/kawai/ATR/data/tdmt.980216/bin/agrep_tool/Agrep -8 -D1
-I1 -S1 到着が遅くなりそうなのでフランクにしていだきたいのですけれど
も </usr1/kawai/ATR/data/tdmt.980216/bin/RECOVERY/DIC/morph-bank-9conv-
index.txt|
(到着 が 遅く なり そう な ので フランク に し ていだきたい のです けれども
```

)=>(到着 が 遅く なり そう な ので ギャランティにし ていただきたい のです けれども), 0.167132

rList->depthStr(1)=(() (L) (L))

mRList->depthStr(1)=(() (L) (L))

wordList=(到着, が, 遅く, なり, そう, な, ので, に, 来, ていただきたい, のです, けれども)

objectPhoneStrList=

t o o c h j a k u g a o s o k u n a r i s o o n a n o d e n i k o t e i
t a d a k i - t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o o n a n o d e n i k o t e i
t a d a k i t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o o n a n o d e n i k i t e i
t a d a k i - t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o o n a n o d e n i k i t e i
t a d a k i t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o u n a n o d e n i k o t e i
t a d a k i - t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o u n a n o d e n i k o t e i
t a d a k i t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o u n a n o d e n i k i t e i
t a d a k i - t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o u n a n o d e n i k i t e i
t a d a k i t a i n o d e s u k e r e d o m o

phDist=0.169492

baseLabels=t o o c h j a k u g a o s o k u n a r i s o o n a n o d e h
u r a n g k u n i s h i t e i t a d a k i - t a i n o d e s u k e r e d
o m o

objectLabels=t o o c h j a k u g a o s o k u n a r i s o u n a n o d e
n i k o t e i t a d a k i - t a i n o d e s u k e r e d o m o

wordList=(到着, が, 遅く, なり, そう, な, ので, ギャランティにし, ていただきたい,
のです, けれども)

objectPhoneStrList=

t o o c h j a k u g a o s o k u n a r i s o o n a n o d e g i y a r a n g
t e i n i s h i t e i t a d a k i - t a i n o d e s u k e r e d o m o

t o o c h j a k u g a o s o k u n a r i s o o n a n o d e g i y a r a n g

```

te ī n i s h i t e i t a d a k i t a i n o d e s u k e r e d o m o
t o o c h j a k u g a o s o k u n a r i s o u n a n o d e g i y r a n g
te ī n i s h i t e i t a d a k i - t a i n o d e s u k e r e d o m o
t o o c h j a k u g a o s o k u n a r i s o u n a n o d e g i y r a n g
te ī n i s h i t e i t a d a k i t a i n o d e s u k e r e d o m o
phDist=0.102941
baseLabels=t o o c h j a k u g a o s o k u n a r i s o o n a n o d e h
u r a n g k u n i s h i t e i t a d a k i - t a i n o d e s u k e r e d
o m o
objectLabels=t o o c h j a k u g a o s o k u n a r i s o u n a n o d e
g i y r a n g t e ī n i s h i t e i t a d a k i - t a i n o d e s u k
e r e d o m o
phList->depthStr(1)=(( ) ( ) (L) (L))
POST_MORPH=:END
outDataSize=2654
Recovery(5): event=PRE_MORPH, data size=1250, method=recoverMorph()
qList->depthStr(1)=(A A)
AGREP COM: /usr1/kawai/ATR/data/tdmt.980216/bin/agrep_tool/Agrep -7 -D1
-I1 -S1 到着が遅くなりそうなのでに
</usr1/kawai/ATR/data/tdmt.980216/bin/RECOVERY/DIC/morph-bank-9conv-
index.txt|
(到着 が 遅 く な り そ う な の で に)=>(到着 が 遅 く な り そ う な の で),
0.138675
AGREP COM: /usr1/kawai/ATR/data/tdmt.980216/bin/agrep_tool/Agrep -8 -D1
-I1 -S1 到着が遅くなりそうなのでに来ていただきたいのですけれども
</usr1/kawai/ATR/data/tdmt.980216/bin/RECOVERY/DIC/morph-bank-9conv-
index.txt|
rList->depthStr(1)=((L) ( ))
mRList->depthStr(1)=((L) ( ))
wordList=(到着, が, 遅く, なり, そう, な, ので, 来, ていただきたい, のです, けれど
も)
objectPhoneStrList=
t o o c h j a k u g a o s o k u n a r i s o o n a n o d e k o t e i t a
d a k i - t a i n o d e s u k e r e d o m o
t o o c h j a k u g a o s o k u n a r i s o o n a n o d e k o t e i t a

```



```
dakitainodesukeredomo
toochjakugaosokunarisoonanodekiteita
daki-tainodesukeredomo
toochjakugaosokunarisoonanodekiteita
dakитайnodesukeredomo
toochjakugaosokunarisounanodekoteita
daki-tainodesukeredomo
toochjakugaosokunarisounanodekoteita
dakитайnodesukeredomo
toochjakugaosokunarisounanodekiteita
daki-tainodesukeredomo
toochjakugaosokunarisounanodekiteita
dakитайnodesukeredomo
phDist=0.175439
baseLabels=toochjakugaosokunarisoonanodehu
rangkuni shiteitadaki-tainodesukeredom
o
objectLabels=toochjakugaosokunarisounanodek
iteitadaki-tainodesukeredomo
phList->depthStr(1)=((L) ())
POST_MORPH=:END
outDataSize=1272
~略~
```

D.2 回復処理クライアント起動

```
WORDS=UTT-START/ 到着 / が / 遅く / なり / そう / な / ので / フランク / に / し / ていただ
き / たい / ん / です / けれども / UTT-END
wordids=5/10053/10104/10448/10332/10647/10164/10065/21648/10051/10085/10086/
10019/10069/10021/10316/6
SOURCE_SENT="到着が遅くなりそうなのでフランクにしていだきたいんですけれども "
tdmt_nbest=1
tdmt_order=1
TARGET_SENT="The arrival will be late , so could you please hold the room ?"
```

tdmt_recovery=RecoverySuccess

tdmt_recovered_morph=(到着 が 遅く なり そう な ので ギャランティにし
ていただきたい のです けれども)

tdmt_rss_ph_dist=0.102941

tdmt_rss_loop_count=1

tdmt_score=1.0e-5

tdmt_max_words=0

tdmt_min_words=1000

vars=1/1/1/1/1/1/1/1/1/1/1/1/1

divs=-/t/o/o/ch/j/a/k/u/g/a/o/s/o/k/u/n/a/r/i/s/o/o/n/a/n/o/d/e/-/h/u/r/a/
ng/k/u/n/i/sh/i/t/e/i/t/a/d/a/k/i/t/a/i/ng/d/e/s/u/k/e/r/e/d/o/m/o/-/-

times=0.000000/0.230000/0.690000/0.830000/1.180000/1.430000/1.630000/1.76
0000/2.580000/2.980000/3.180000/3.780000/4.910000

score=-2631109.000000

acoustic=4580891.000000

ngram=-7212000.000000

comment=END-OF-UTTERANCE-2

～略～

付録 E デモシステム

E.1 概要

本デモシステムでは、音声認識結果を翻訳処理に渡し、音声認識正解部分の特定処理によって部分的に翻訳される様子を見る（あるいは示す）ことができる。なお、検討課題である音声認識正解部分の特定処理は TDMT の一部を変更する形で試験的に組み込まれている。本デモシステムとして提供される範囲は以下のとおりである。

- デモ画面
- TDMT の変更部分（正解部分特定処理）
- TDMT とのインターフェイス

なお、本デモシステムが動作するためには少なくとも以下の動作環境が必要である。

- csh, perl(ver4.0), tcl/tk(ver4.0), X window

さらに、本デモシステムが動作するためには以下のものを別途用意する必要がある。

- TDMT(tdmt-multi-19990201 版), allegro-common-lisp(ver4.3)

E.2 操作方法

本デモシステムでは、X window による GUI でユーザインターフェイスを構成しているため、主にマウスによって操作するように設計されている。

(E.2.1) インストール

プログラム及び、データは (8mm テープなどの) UNIX の tar 形式のアーカイブにより供給される。起動は `tdmt.990201/tdmt-recover-99.csh` から行なうが、`tdmt.990201/demo-recover-99/` 以下のディレクトリに本体がある。

(E.2.2) 設定

前述のとおり、別途用意した各プログラムと連携させなければならないため、これらを指定する必要がある。また、ユーザのマシン環境に合わせてカスタマイズすることもできる。変更可能な項目は、

`tdmt.990201/demo-recover-99/config.txt` のファイルに記述されているので、エディタなどで変更することができる。以下にその項目を解説する。具体例は付録を参照。

- 全般の設定

Top_Dir デモシステム本体のインストール先のディレクトリ IF_DIR から起動時に設定される。

Trans_Com_List_File 翻訳プロセスへのコマンドファイル

Wave_List_J_File 入力波形ファイル (日本語) を指定するリストのファイル

Wave_List_E_File 入力波形ファイル (英語) を指定するリストのファイル

Label_List_File デモ画面のラベルを表わすリストのファイル

Label_List_E_File 英語版のラベルを表わすリストのファイル

Label_List_J_File 日本語版のラベルを表わすリストのファイル

Color_List_File 表示色を指定するリストのファイル

Default_Font デフォルトのフォント指定

Default_KanjiFont デフォルトのフォント指定 (日本語)

Text_Font テキストのフォント指定

Text_KanjiFont テキストのフォント指定 (日本語)

Frame_Size デモ画面のサイズ (LARGE: 大きい、SMALL: 小さい)

● 音声入力プロセスの設定

Wave_Run_Mode 実プロセスを実行するかどうか (REAL: する、DUMMY: しない)

Wave_In_REAL 実プロセスを実行する場合のプロセス起動

Wave_In_DUMMY 実プロセスを実行しない場合の処理 (特になければ、とする。)

● 翻訳プロセスの設定

Trans_Run_Mode 実プロセスを実行するかどうか (REAL: する、DUMMY: しない)

Trans_In_REAL 実プロセスを実行する場合のプロセス起動 (入力側)

Trans_Out_REAL 実プロセスを実行する場合のプロセス起動 (出力側)

Trans_Process_REAL 実プロセスを実行する場合のプロセス起動 (翻訳サーバー起動)

Trans_In_DUMMY 実プロセスの代わりに代替プロセス (入力側)

Trans_Out_DUMMY 実プロセスの代わりに代替プロセス (出力側)

Trans_Process_DUMMY 実プロセスの代わりに代替プロセス (翻訳サーバー起動)

(E.2.3) 起動および終了方法

プログラムは次のシェルコマンドで起動できる。内部的にいくつかのプロセスを起動するため、完全に起動するには少し時間がかかる。プロセスが完全に起動し終わっていてもマウスからボタンを押すなどしてコマンドを発行することはできる。この場合、起動され次第、コマンドが実行される。コマンドが実行される様子は、起動したシェルに出力される。

```
user@host[131]: cd tdmt.990201
user@host[132]: csh -f demo-recover-99.csh
```

また、別ターミナルから、次のコマンドを実行しておく。(バックグラウンドでも良い) これにより訂正処理が起動される。

```
user@host[131]: cd tdmt.990201/bin/RECOVERY
user@host[132]: perl recovery.pl
```

メニューの「ファイル」から「終了」を選択するか、同じシェルで Ctrl+C とすることによりデモシステムを終了することができる。通常、この操作ですべてのサブプロセスを終了できるが、ごくまれに終了できない場合がある。その場合は、サブプロセスを kill コマンドにより終了させる必要がある。主なサブプロセスの ID は次のようにして知ることができる。これらを kill すればその他のものは自然に消滅する。

```
user@host[134]: ps auxw | grep TDMT
```

(E.2.4) 操作方法

- メニュー

ファイル 「終了」 デモシステムを終了する。

日本語文 日本語文から入力（波形ファイル）を選択する。

英語文 英語文から入力（波形ファイル）を選択する。

オプション 現在、「音声入力」「音声認識処理」「翻訳処理」「部分翻訳処理」は機能無し。

「日英」「英日」 翻訳方向切替え（入力文を選択したときに自動切替）

「音声認識実行」 現在は無効

「音声認識結果」 本バージョンでは常に認識結果を使用

「ラベル」 デモシステムのボタンなどのラベルを変更

- ボタン

音声入力 メニューから「選択」された音声入力をスピーカで確認する。WAV ファイルの再生。

音声認識 音声入力の波形ファイルを音声認識処理に入力し、認識結果を表示する。

翻訳処理 音声認識結果を翻訳処理に入力し、翻訳結果を表示する。

訂正処理 訂正処理を実行し、翻訳結果を表示する。「訂正が必要かどうか」「訂正が可能かどうか」の情報が示され、訂正された原言語文は「訂正結果」、それを翻訳したものが「訂正翻訳」として表示される。訂正部分がハイライトされて強調表示される。

解説 別ウィンドウを開き、訂正翻訳処理の一連の流れをフローチャートと実例のアニメーションで示しながら解説するための画面を表示する。

「Reset」	ウィンドウを初期化する。
「Current」	カレントのステップを再表示する。
「Next」	次のステップを表示する。
「Stop」	自動デモを停止する。
「Auto」	自動デモを開始する。
「Close」	ウィンドウを閉じる。

● パラメータ設定

最大距離 用例距離計算に対する制限を与えるための閾値。

最小単語数 短い単語列に対して採否を決めるための閾値。（詳細は「ガーベージ区間の判定」参照）

● 音声合成処理

音声認識結果の確認 音声認識結果の文字列上をマウスでクリックすると、音声合成されてスピーカで確認することができる。

部分翻訳結果の確認 部分翻訳結果の文字列上をマウスでクリックすると、音声合成されてスピーカで確認することができる。

E.3 実装説明

本デモシステムは、本体が Tcl/Tk により書かれており、トップレベルは csh のスクリプトから呼ばれるようになっている。音声波形出力、音声認識、言語翻訳、音声合成の各処理はサブプロセスとして外部化されており、パイプによりデモシステム本体と接続される。この方式のため、適当なダミーファイルを用意してパイプに接続することにより、部分的なテストを容

易に行うことができる。また、各プロセスはそれぞれ C 言語、Common Lisp 言語を始めとして様々なスクリプト言語を利用して書かれているが、これに関係なく標準入出力を利用してデモシステム本体と接続している。デモシステム本体では、基本的にデータの流れを制御し、表示に必要な情報を得ているだけであり、各プロセス間に渡される情報の詳細については関知しない。(ただし、今回のバージョンでは音声認識、音声合成は使用していない。)

(E.3.1) プログラムファイル

demo-recover-99.csh から demo-recover-99/demo-recover-99.tcl を呼び出す。さらに「解説」機能のために demo-recover-99/demo-presentation.tcl を呼び出す。

(E.3.2) 設定ファイル

config.txt	マスター設定ファイル
color_list.txt	テキスト色設定ファイル
label_list.txt	画面上のラベルの指定ファイル
label_list_j.txt	画面上の日本語ラベルの指定ファイル
label_list_e.txt	画面上の英語ラベルの指定ファイル
trans_com_list.txt	翻訳処理へのコマンドの指定ファイル
wave_list_j.txt	入力波形の指定ファイル (日本語)
wave_list_e.txt	入力波形の指定ファイル (英語)

(E.3.3) サブプロセスの呼び出し

本バージョンでは音声認識処理、音声合成処理は使用しない。

- 音声入力プロセス

音声波形ファイルを入力としてスピーカから音声を出力する。

デフォルトでは ../bin/WAVE/Talk3 を使用している。このプロセスは毎回起動され、処理が終われば終了する。

- 音声認識処理プロセス

音声波形ファイルを入力として音声認識プログラムを実行する。

デフォルトでは ../bin/ATRlattice/ の WaveIn.csh, ATRexec, sp2tr.perl が使用される。これらは、初めに一度起動されると、その入力パイプに受けたデータを処理し、順次出力パイプに送るようになっている。そのためデモシステム本体は、read/write モードでこのプロセス群と接続して入出力を行う。

- 翻訳処理プロセス

音声認識処理の結果を入力として言語翻訳プログラムを実行する。

デフォルトでは af.csh を入力用プロセスとし、tdmt_of.csh を出力用プロセスとしている。翻訳処理はこれらの入出力プロセスと名前付きパイプ (named pipe) で接続されている

TDMT-MULTI.csh が行う。これらのサブプロセス群は初めに一度起動されると、受けたデータを順次処理するようになっている。デモシステム本体は入出力プロセスを通じてデータをやりとりする。

- 音声合成処理プロセス

文字列を入力として音声合成プログラムを実行し、スピーカから出力する。

デフォルトでは../bin/CHATR/j2r.csh, /DB/PI/bin/chatr-0.9 が使用される。CHATR へ渡すコマンドはデモシステム本体が内部で生成している。また、漢字かな混じり文からローマ字への変換フィルタ (j2r.csh) を使用している。変換フィルタのプロセスは毎回起動されるが、CHATR は初めに一度起動されてコマンド待ち状態に入る。

付録 F 評価用ツール

F.1 目的

誤り訂正処理を行なう場合の効果を調べるための比較実験を行なうためのバッチ処理である。比較は、正解文、音声認識文に対して行ない、音声認識文についてはさらに訂正処理をする場合としない場合について翻訳結果を比較する。

F.2 起動方法

コマンド:

```
bat_comparison_tdma.csh  RCGANS_FILE  >!  TRANS_FILE  
perl  chfmt-outputs.pl  TRANS_FILE  >!  COMP_FILE
```

入力ファイル (RCGANS_FILE) を inputs/small-sample.rcgans とし、
中間ファイル (TRANS_FILE) を inputs/small-sample.rcgans.trans
出力ファイル (COMP_FILE) を inputs/small-sample.rcgans.trans.comp
としたとき、次のように起動する。

訂正サーバ起動

```
-----  
kawai@ssten01[39] cd bin/RECOVERY  
kawai@ssten01[40] recovery.pl  
-----
```

別ターミナルで、

```
-----  
kawai@ssten01[172] bat_comparison_tdma.csh inputs/small-sample.rcgans >!  
inputs/small-sample.rcgans.trans  
#  
tdmt_MLEX_J=../bin/RECOVERY/DIC/MASTER_LEXICON_TDMT_98NOV2  
1  
### MLEX_J Loading ...  
### MLEX_J Total: 20182 words  
### MLEX_J Pseudo: 9 words  
### MLEX_J Redundant: 427 words  
  
# tdmt_MLEX_E=../bin/RECOVERY/DIC/E_MSTR_LEX_99JUL08
```

```
### MLEX_E Loading ...
```

```
### MLEX_E Total: 11626 words
```

```
### MLEX_E Pseudo: 9 words
```

```
### MLEX_E Redundant: 9 words
```

```
kawai@ssten01[173] perl chfmt-outputs.pl inputs/small-sample.rcgans.trans >!  
inputs/small-sample.rcgans.trans.comp  
-----
```

F.3 関連ファイル

- bf バッチ処理：
bat_comparison_tdmr.csh
chfmt-inputs.pl
chfmt-outputs.pl
- bf 初期設定ファイル：
init_morph_recovery.tdmr
- bf サンプル入出力ファイル：
inputs/small-sample.rcgans
inputs/small-sample.rcgans.trans
inputs/small-sample.rcgans.trans.comp

F.4 入出力ファイル仕様

- bf 入力ファイル (RCGANS_FILE):
inputs/small-sample.rcgans

*ここに'#@ Funcall=showExperimentInfo();'の行を入れておくと
ヘッダーとして、実験環境の情報が出力される。

*ここに'TDMTCommand="0 (switch-mode :E-J)'"の行を入れておくと
翻訳モードが動的に変更できる。

FILE := (CONT \n)+

CONT := COMMENT | TDMT_COMMAND | UTTERANCE_SET |

NULL_LINE

COMMENT := '#@'で始まる行。'Funcall=xxx();'が有れば、
chfmt-outpus.pl内で実行される。

TDMT_COMMAND := 'TDMTCommand='で始まる行。

TDMTで解釈され実行される。

UTTERANCE_SET := UID ANS_WIDS ANS_WORDS RCG_WIDS

RCG_WORDS U_END

UID := 'UTTERANCE='で始まる行。

ANS_WIDS := 'ans-wordids'で始まる行。

ANS_WORDS := 'ans-WORDS'で始まる行。

RCG_WIDS := 'rcg-wordids'で始まる行。

RCG_WORDS := 'rcg-WORDS'で始まる行。

U_END := 'comment=END-OF-UTTERANCE-'で始まる行。

NULL_LINE := 空行

• bf 中間ファイル (TRANS_FILE):

inputs/small-sample.rcgans.trans

FILE := (CONT \n)+

CONT := COMMENT | TDMT_COMMAND | UTTERANCE_SET |

NULL_LINE

COMMENT := '#'で始まる行。但し、'# output'で始まる行を除く

TDMT_COMMAND := 'TDMTCommand='で始まる行。

TDMTで解釈され実行される。

UTTERANCE_SET := ANSWER RECOG RECOV

ANSWER := ANSWERの翻訳結果。'# output ANSWER #'の行で始まる。

RECOG := RECOGの翻訳結果。'# output RECOG #'の行で始まる。

RECOV := RECOGの訂正翻訳結果。'# output RECOV #'の行で始まる。

NULL_LINE := 空行

- inputs/small-sample.rcgans.trans.comp

NULL_LINE := 空行

inputs/small-sample.rcgans

```
#@ Funcall=showExperimentInfo():
```

```
#0000000000000000
```

```
#0 Mode: E-J 0
```

```
#0000000000000000
```

```
TDMTCommand="0 (switch-mode :E-J)"
```

```
UTTERANCE=TAS12008.B.1
```

```
ans-wordids=5/11019/10090/13544/10385/6
```

```
rcg-wordids=5/10057/10090/14355/6
```

```
rcg-WORDS=UTT-START/if/the/eve/UTT-END
```

```
comment=END-OF-UTTERANCE-TAS12008.B.1
```

```
UTTERANCE=TAS12008.B.2
```

```
ans-wordids=5/10025/11356/11355/11364/6
```

```
rcg-wordids=5/10025/10102/11355/11364/6
```

```
rcg-WORDS=UTT-START/yes/miss/suzuki/how_may_i_help_you/UTT-END
```

```
comment=END-OF-UTTERANCE-TAS12008.B.2
```

```
#0000 END OF FILE 0000
```

```
-----
```

- 中間ファイル (TRANS_FILE):

```
inputs/small-sample.rcgans.trans
```

```
-----
```

```
#tdmt: TDMT start ...
```

```
#tdmt: in-file:  "../tmp/=TDMT_IN=.11906"
```

```
#tdmt: out-file:  "../tmp/=TDMT_OUT=.11906"
```

```
#tdmt: if-level: :MORPH
```

```
#tdmt: output-function: OUTPUT-RESULT
```

```
#tdmt: TDMT-MULTI: READY
```

```
### TDMT_INPUT_FILTER: Cond=READY
```

```
TDMTCommand="0 (load (format nil "~a~a\" PUBLIC:*tdmt-interface-dir*  
\"code/morph-recovery.lisp\"))"
```

```
TDMTResult="0 T"
```

```
TDMTCommand="0 (load (format nil "~a~a\" PUBLIC:*tdmt-interface-dir*  
\"code/morph-recovery-interface.lisp\"))"
```

```

TDMTResult="0 T"
TDMTCommand="0 (switch-mode :E-J)"
TDMTResult="0 :E-J"

```

～略～

```

#####
#@ Sample Experiment      @
#@ By: Jun Kawai         @
#@ Date: 1999/11/17      @
#@ Comment: ---          @
#####
#@ Funcall=showExperimentInfo();
#####
#@ Mode: E-J @
#####
TDMTCommand="0 (switch-mode :E-J)"
TDMTResult="0 :E-J"
### TDMT_INPUT_FILTER: Language=E

##### BEGIN of UTTERANCE #####
# output ANSWER #
TDMTCommand="130 (TRANSFER::switch-rm-condition nil)"
TDMTResult="130 NIL"

UTTERANCE=TAS12008.B.1
WORDS=
wordids=5/11019/10090/13544/10385/6
tdmt_pmorph=UTT-START

SOURCE_SENT="good afternoon the peninsula may i help you"
tdmt_nbest=1
tdmt_order=1
TARGET_SENT="こんにちは、ペニンシュラホテルです。どうしましたか。"
tdmt_recovery=Original

```

～略～

inputs/small-sample.rcgans.trans.comp

UTTERANCE=TAS12008.B.1

RcvJudge: RecoveryFail

AnsSent: "good afternoon the peninsula may i help you"

RcgSent: "if the eve"

RcvSent: "if the eve "

AnsMorph: ((good afternoon INTERJ) (the DET) (peninsula PROPN) (may i help you INTERJ))

RcgMorph: ((if CONJ) (the DET) (eve CN))

RcvMorph: ((if CONJ) (the DET) (eve CN))

AnsTrans: "こんにちは、ペニンシュラホテルです。どうしましたか。 "

RcgTrans: " . . . ____ 前日です。 "

RcvTrans: " . . . ____ 前日です。 "

AnsScore: 0.000015

RcgScore: 5.000005

RcvScore: 5.000005

comment=END-OF-UTTERANCE-TAS12008.B.1

END OF FILE
