TR-IT-0231

生成駆動音声認識方式 (GD-SR)

Generation Driven Speech Recognition (GD-SR)

高橋 一裕 Takahashi Kazuhiro
中村 篤 Nakamura Atsushi

1997.08.28

概要

本報告では、音声翻訳の際に問題となる認識誤り（脱落、湧出）を効果的に抑制し、適切に言語翻訳処理可能な結果を出力する音声認識手法を提案する。提案手法（生成駆動音声認識 “Generaion Driven Speech Recognition(GD-SR)”）は１度目の音声認識結果を基に文仮説を生成し、生成した仮説を言語制約として再度音声認識を行なうという手法である。実験を行ない、従来のN-gramモデルに比較して単語正解率が６％向上し、又、認識結果の翻訳の成功率は２１％向上する、という結果を得た。

In thie paper, we proposes a spontaneous speech recognition method which effectively suppresses false alarms and deletions that are serious problems for speech translation. Our method (“Generation Driven Speech Recognition (GD-SR)”) generates sentence hypotheses from speech recognition results and executes speech recognition again using these hypotheses as new language constraints. In the experiment, the proposed method improved 6% in word accuracy and 21% in Japanese-English translation quality in comparison with a conventional N-gram based speech recognizer.

目次

1 序論 iv

1 研究の背景と目的 iv

1.1 Feedback Verification Procedure(FVP) v

1.2 提案手法 v

2 生成駆動音声認識方式 (GD-SR) vi

1 生成駆動音声認識方式 (GD-SR) vi

2 GD-SR の実装 vii

3 試作システム構成 ix

3.1 First Speech Recognition Module ix

3.2 Semantic Analysis Module ix

3.3 Sentence Hypotheses Generation Module ix

3.4 Verification: Second Speech Recognition Module x

3 実験 xiii

1 実験 xiii

1.1 例文集 xiii

1.2 単語群 xiii

1.3 テスト用音響データ xvi

1.4 音響処理 xvi

1.5 実験結果 xvi

2 考察 xx

4 まとめと今後の課題 xxi

1 まとめと今後の課題 xxi

5 謝辞 xxii

参考文献 xxiii

図目次

2.1 従来手法 vi
2.2 GD-SR vi
2.3 The processing flow in GD-SR vii
2.4 first speech recognition result xi
2.5 semantic analysis results xi
2.6 sentence hypotheses xi

表目次

3.1 音響処理条件 xvi
3.2 WORD ACCURACY xix
3.3 LANGUAGE TRANSLATION RESULTS xix

第 1 章

序論

1　研究の背景と目的

音声認識を利用したシステムを構築する場合、認識誤りは重大な問題となる。

これに対処するのには大きく２つの方向性がある。１つは認識精度自体を向上させる方向、もう１つは言語処理部の頑健性を向上させて誤認識混じりであっても処理できるようにする方向、である。これら両者に対して、多くの研究がなされている。しかし、どちらの方向に関してもその研究の多くは、音声認識部／言語処理部の一方の性能の向上を目指す物であり、必ずしもシステム全体の能力向上を目指すものにはなっていなかった。

単に音声認識部と言語処理部とを結合しただけのシステムでは、認識はできても言語処理部は認識結果を処理できない、と言ったミスマッチが起こりがちである。例えば、その単語がないと言語処理できなくなるような単語を平気で脱落させる音声認識システムや、音声認識システムが起こしもしない認識誤りへの対処ばかりしている言語処理システムなども、ないとは言い切れない。

このようなミスマッチを解消する方法の一つは、システム全体を考慮した上でトップダウンに音声認識・言語処理の役割を設定することである。TOSBERG[2] はそのようなトップダウン設計の例である。TOSBERG では、音声認識対象単語を少数のキーワードに限定している。対象語彙数を少なくすることで誤認識を低減する一方、意味内容を推定するに足るだけの情報を得られるようにキーワードを選んでいる。少数の誤認識に対しては、言語処理部においてキーワードの組合せの可否を判定し、不適切な単語を除去することで対処する。

この手法が有効なのは、少数のキーワードから意味内容の推定が可能であるようなドメイン、（ファーストフードの注文など）を対象にしたときであり、つまり、すべてのドメインにとって有効というわけではない。例えば、ＡＴＲの目指す音声翻訳の場合、扱う意味内容がより詳細なものになるため必要となるキーワードが増加してしまう。その増加したキーワードの中には音響的に短い機能語も多く含まれており、それらは音声認識精度の低下をもたらす。増加した誤認識に対応するための言語処理は複雑なものになり、結果として通常のロバストパーザが抱える問題をそのまま抱えることになってしまう。

ミスマッチを解消するもう一つの方法は、音声認識の際に、強い言語制約を利用する方法である。例えば、認識の途中結果に対して文法的に接続可能であるような単語のみを探索の対象として認識処理を継続する手法である。合文法的な単語のみを探索対象としていくことにより、最終的な認識結果も合文法的なものであることが保証できる。

この手法は、発声が合文法的であり、かつ、文頭から現在処理中の部分までに誤認識が無ければ、有効に機能する。言い替えれば、その前提が満たされない場面では利用できない。例えば、「東京から大阪まで急行券を１枚下さい」と発話しようとしたが、言い誤り／誤認識に

より冒頭部分を「特急が」ととってしまったとする。この場合、強い言語制約がかかっているが故にその後の認識もうまくいかない可能性が高い。だからと言って、発話全体の処理が終るまで全ての仮説を保持していては、処理量が爆発的に増加してしまう。

1.1 Feedback Verification Procedure(FVP)

これらを解決する手法のひとつとして、“Feedback Verification Procedure (FVP)”[1] が提案されている。FVP は音声認識部と言語処理部との情報を交換しあうことで適切な音声認識結果を得ようとするものである。その手順は、

1. 1度目の音声認識を行ない、単語ラティスを得る。認識対象単語は内容語のみ。ワードスポッティングを行なう。

2. ラティス内の各単語に対して、文法的に接続可能な機能語類（助詞、助動詞、接辞、など）を付加し、単語グラフを生成する。

3. 生成した単語グラフを言語制約として2度目の音声認識を行ない、最終的な認識結果を得る。

言語処理部にとって必須である機能語類があった場合、それは1度目の認識では検出されないが、ステップ2で生成された強い言語制約を用いることによって2度目の音声認識の際に検出することが期待できる。FVP は音声認識部とそれに続く言語処理部とをより強く結合することによってシステム全体のパフォーマンスの向上に成功している。

1.2 提案手法

本報告では、この FVP を発展させたものとして生成駆動音声認識方式（“Generation Driven Speech Recognition (GD-SR)”）を提案する。

FVP は phrase 内の文法的制約を用いて強い言語制約を生成することにより効果的に機能語の検出を行なう。しかしながら、1度目の認識で脱落した内容語は救済されない。一方、提案手法 (GD-SR) では、機能語類だけでなく内容語をも生成対象にしている。仮説生成の際に意味的内容を併用し、それにより、機能語類だけでなく、1度目の認識で脱落した内容語の救済も可能にした。

以下、2章では GD-SR の概要を、3章では実験システムとして試作した GD-SR を、4章では実験結果をそれぞれ説明する。その後、まとめと考察を述べる。

第 2 章

生成駆動音声認識方式 (GD-SR)

1　生成駆動音声認識方式 (GD-SR)

生成駆動音声認識方式（“Sentence Generation Driven Speech Recognition (GD-SR)”）と従来の認識方式との違いは、図 2.1、図 2.2のように表すことができる。

従来手法（図 2.1）においては、言語制約はあらかじめ与えられた静的なものであり音声認識部と言語翻訳部とは独立している。認識結果は一方的に言語翻訳部に送られ、そこで処理される。

一方、GD-SR（図 2.2）においては、言語制約は言語翻訳部から動的に提供される。情報交換の後、翻訳に必要充分な認識結果が得られたら翻訳処理が行なわれる。

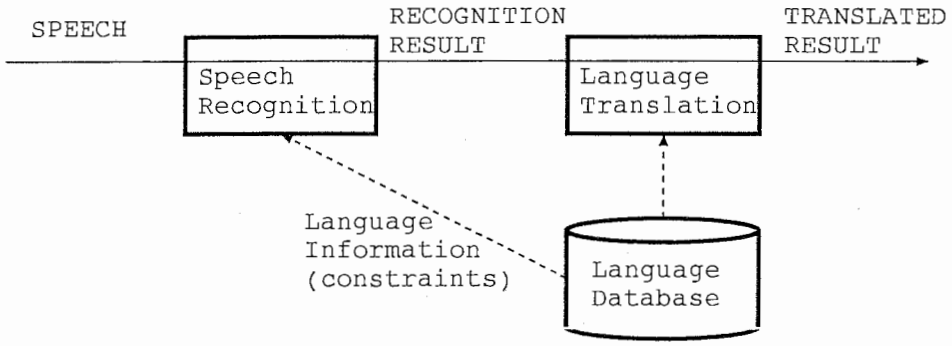

図 2.1: 従来手法

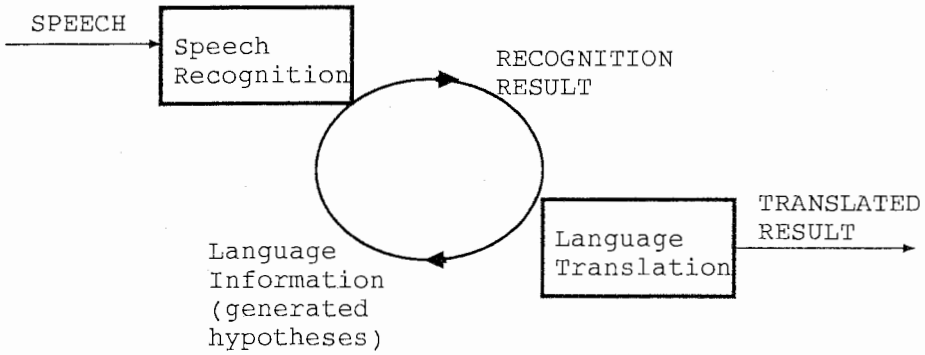

図 2.2: GD-SR

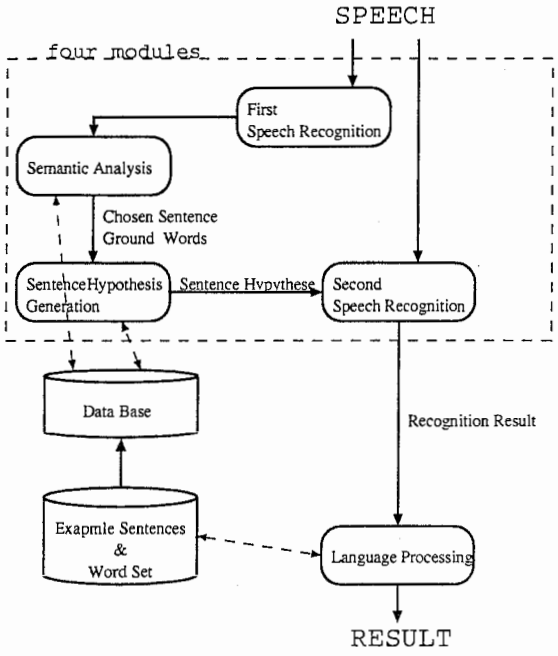

図 2.3: The processing flow in GD-SR

2 GD-SR の実装

GD-SR の実現方法には多くの方法が考えられるが、本報告においては図 2.3に示すように、4 つのモジュールから構成するように実装した。4 つのモジュールとは “First Speech Recognition”、“Semantic Analysis”、“Sentence Hypothesis Generation”、“Verification”、の 4 つである。

各モジュールは以下のような処理を行なう。

1. First Speech Recognition Module：入力音声に対して音声認識処理を行なう。一般的な音声認識であり、特殊なことはしていない。

2. Semantic Analysis Module：認識結果を解析し、発話された可能性のある意味内容を列挙する。

3. Sentence Hypothesis Generation Module：列挙された意味内容と 1 度目の認識結果とを基に、適切に言語処理可能であるような文仮説を生成する。

4. Verification Module：生成された仮説を言語制約として音声認識を再実行し、最終的な認識結果を出力する。

4 つのモジュールの中で最も重要なモジュールは、“Sentence Hypothesis Generation Module” である。言語処理部に必要十分な情報（ある特定の単語の有無の確定や曖昧性解消にとって不可欠な機能語類の情報など）を含んでいて、かつ言語処理不能となるような現象が含まれていないような強い言語仮説の生成が GD-SR のキーポイントとなるからである。

従来の音声認識システムでは、あまり強い言語制約はかけないのが一般的であった。例えば、“TOSBERG” では音声認識部には事実上言語制約は適用されていないし、その他の多くの統計的言語モデルにおいても “flooring” や “smoothing” などによって言語制約を緩める処

理が為されるのが普通である。強い言語制約を避ける理由は大きく 2 つあると考えられる。1 つは、話言葉、特に spontaneous speech を扱おうとした時、制約が強過ぎてなんの候補も残らなくなってしまう危険や、更には前述のように処理途中での誤認識がそのまま全体の誤認識につながってしまう危険などがあるからである。もう 1 つは、強い言語制約は言語処理部においてかけるものであると考えられているから、である。

それに対して提案手法である GD-SR では、意味表現から文仮説を生成し、それを用いて音声認識を再実行する。1 度目の認識を通常の緩い言語制約で行なうことにより致命的な誤認識の危険を回避する一方、2 度目の認識は文仮説を言語制約として用いて行なうことにより、言語処理部が適切に処理可能であるような認識結果を出すことが可能となる。

3 試作システム構成

各モジュールの実現方法には多くの方法が考えられるが、今回は有効性の検証が目的であるため簡易に実現できる方法を選んだ。

3.1 First Speech Recognition Module

N-gram 言語モデルを用いて連続音声認識処理を行ない単語グラフ [4] を獲得する。単語グラフは有向ネットワークであり、各エッジに単語情報とスコアが各ノードに時刻情報が、それぞれ格納されている。単語グラフは連続音声認識と言語処理とを結合するための有効なインタフェースであるとされている。

具体的には、N-gram として１９９７年春に作成された標準言語モデルを用いた ATRlattice[10] で構成した。今回の実験に現れない単語は認識対象語彙から除去した。語彙の変更に伴う N-gram の再学習は行なっていないが、そのことによる大きな変動は無いことが経験的に知られている。

3.2 Semantic Analysis Module

意味の取扱は重要かつ難しい問題である。本報告では、あらかじめ用意された対象ドメイン用の例文集の例文の１つ１つが１つの意味カテゴリーを形成すると仮定した。更に変形規則と交換可能な単語群をまとめた単語リストとを用意し、各例文のカバー範囲を広げている。

この仮定は精密なものではない。しかし、

- 対象となるドメインが決まれば意味カテゴリーも決まる。
- 各意味カテゴリーに対して、少数の代表的な例文を想定することができる。
- 各例文と単語リストと変形ルールとによって、対象ドメインのかなりの部分をカバーすることができる。

という想定は現実的なものであり、よって例文の１つ１つを１つの意味カテゴリーに対応させるのは妥当な近似であると言える。

以上の想定を基に、本報告では、
意味解析：あらかじめ用意された例文集の中の例文の１つを選ぶこと
解析結果：選択された例文と選択の根拠となった単語（群）
と定義した。

意味解析（例文選択）は１度目の音声認識処理によって検出された単語群と各例文とのマッチングによって行なう [11]。

この時点ではすべての曖昧性解消が済んでいる必要はない。つまり、複数の解析結果が残っていても構わない。但し、それぞれの解析結果は適切に処理可能な曖昧性のないものでなければならない。

3.3 Sentence Hypotheses Generation Module

Sentence Hypothese Generation Module は解析結果（選択された文＋根拠となった単語）を基に文仮説を生成する。

生成手順は以下の通りである。

1. 各解析結果（選択文＋根拠単語）に対して、２～５を繰り返す。
2. 選択文に含まれ得る文節をリストアップする。
3. 各文節を有限状態オートマトン（ＦＳＡ）で表現する。これを文節ＦＳＡとする。
4. 根拠単語を持つ文節ＦＳＡと検出されたFillerWord（間投詞、感動詞など）を結合する。この際、根拠単語の出現順序は守る。FillerWordも出現順序を守って結合するが、スキップ可能にしておく。
5. 根拠単語を持たない文節ＦＳＡを全ての結合点に埋め込む。これを文ＦＳＡとする。
6. 全ての文ＦＳＡを１つのＦＳＡにまとめあげる。これを文仮説とする。

この手順により１度目の認識結果に似ており、かつ意味的・文法的に整合性のある仮説が生成できる。この仮説をＦＳＡで表現することにより、全体をよりコンパクトに表現するようにしている。

3.4 Verification: Second Speech Recognition Module

生成された文仮説を言語制約として音声認識処理を再度実行する。具体的には、生成された文仮説（ＦＳＡ）をATRmk_fsa_bin[10]を用いてバイナリー変換し、それをATRlattice実行時の制約として使用する。

検定モジュールでは、１度目の音声認識時に検出できなかった単語を検出できるようにするため音響制約が緩められる。一般に、音響制約の緩和は認識誤り（湧出誤り）を増加させることにつながるが、ここでは非常に強い言語制約（文仮説）が課されているので、これらの認識誤に対する抑制効果が期待できる。

本実験においては、検定モジュールで用いる発音辞書に対して典型的な発声変形の結果を追加するということによって音響制約の緩和を行なっている。

発生変形のルールは以下の通りである。

- 助詞における子音の脱落
- 拗音 (“j”) の脱落
- ９個より多い音素から構成される単語において、単語頭・単語末の音素の脱落
- 長母音・二重母音の単母音化

以上のルールにより、単語の読みを拡大している。

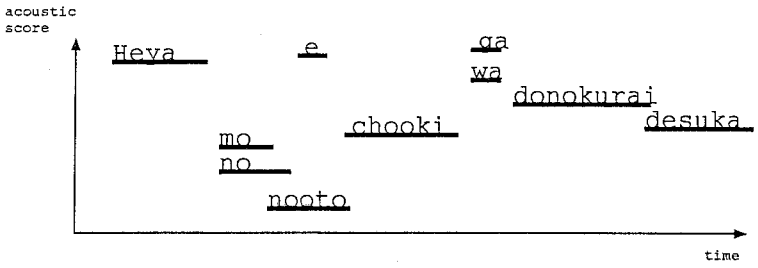

図 2.4: first speech recognition result

RESULT 1

chosen sentence	ground words
Asking a room rate	
[ROOM] no	heya
[CHARGE] wa	
donokurai desuka	donokurai desuka

RESULT 2

chosen sentence	ground words
Asking room size	
[ROOM] no	heya
[SPACE] wa	
donokurai desuka	donokurai desuka

図 2.5: semantic analysis results

提案手法を「部屋の料金はどのくらいですか」という発話を例にして説明する。

1度目の認識結果として得られた単語グラフを図2.4に示す。単語グラフを見ると、『の』が最尤候補になっていない、湧出がある、『料金』が脱落している、などの認識誤りがあり、適切な認識結果とは言い難いものである。

解析結果の例を図2.5に示す。2つの解析結果が得られている。1つ目は部屋の料金を尋ねるもの、2つ目は部屋の広さを尋ねるものである。

根拠単語の出現順序（『部屋』、『どのくらい』、『ですか』）を守って文ＦＳＡを生成し、それらをまとめあげて1つの文仮説を生成する。最終的な文仮説を図2.6に示す。

生成された文仮説を言語制約として検証（2度目の認識処理）が行なわれる。そして、最終的な認識結果として「部屋の料金はどのくらいですか」が得られる。検証処理時には、「料

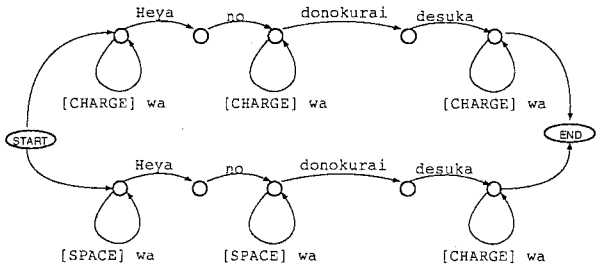

図 2.6: sentence hypotheses

金（りょうきん）」の発音として語頭の“r”の音が脱落した“youkin”や 2 重母音を単母音化した“ryokin”が加えられている。

第 3 章

実験

1　実験

GD-SR の有効性を検証するため実験を行なった。統計言語モデル [5] を用いた認識結果を比較対象とする。比較は認識結果の最尤パスの単語正解率および最尤パスを TDMT 機械翻訳システム [9] を用いて日英翻訳したときの翻訳成功率で行なった。

1.1　例文集

選択用例文集の基データとしてＡＴＲ旅行会話データベース [6][7] の中から２２会話分のデータを採用した。具体的には、

TAS12001.JMOR	TAS12007.JMOR	TAS12014.JMOR	TAS12019.JMOR	TAS12025.JMOR
TAS12002.JMOR	TAS12008.JMOR	TAS12015.JMOR	TAS12020.JMOR	TAS12026.JMOR
TAS12003.JMOR	TAS12009.JMOR	TAS12016.JMOR	TAS12022.JMOR	
TAS12004.JMOR	TAS12010.JMOR	TAS12017.JMOR	TAS12023.JMOR	
TAS12006.JMOR	TAS12013.JMOR	TAS12018.JMOR	TAS12024.JMOR	

の２２会話である。各例文は音声収録された会話を文字化したものであり、形態素分割され、品詞ラベルが付与されている。

Section 3.2 で述べた理由により、例文は言語処理システムによって適切に処理できなくてはならない。又、処理効率を考えると無意味語や重複などの無駄は少ないことが望まれる。しかし、各例文は会話文であるため間投詞なども多く、そういった文は効率的な例文とは言い難い。又、例文全体を見るとほとんど同じパターンの文が複数回出現するなど、これも効率的な例文集とは言い難い。したがって、これらの例文（集）から生成される仮説は、冗長に過ぎる可能性が高い。

しかし、冗長な単語があったとしても、１度目の認識結果中に根拠となる単語がなければ文仮説中では「あってもいい」文節として扱われるだけである。ほとんど同じ文が解析結果中に含まれていると文仮説のサイズが増大するが、直ちに提案手法を破綻させるようなものではない。

提案手法が例文集の品質に対して頑健であることを示す目的もあり、今回の実験では例文集を細かく調整することは行なわないものとする。

例文は全体で６４５文。異なり単語数は６３１である。

1.2　単語群

例文集のカバー範囲を広げるために交換可能な単語群を用いる。

単語群の定義は「お互いに交換可能であるような単語又は句をひとまとめにしたもの」である。各単語群には適当なラベル名が付けられるが、ラベル名は単語群のＩＤとして扱われるだけであり、ラベル名自体が意味処理の対象になるわけではない。

例えば、「はい、こちらは田中です」「私は佐藤です」という２つの例文だけでは、狭い部分しかカバーできない。しかし、「LABEL1: 佐藤、田中、フィリップ」という単語群があれば、２つの例文のカバーする範囲は大きく広がる。

これらの単語群は本来はシソーラスなどを基に作成するべきだが、今回の実験では、手で分類を行なった。但し、高度な意味的判断などは用いていない。単純に交換可能なものを集めただけである。どの単語群にも属さない単語もあるし、複数の単語群に属する単語もある。

本実験では、３６の単語群が設定された。単語群のラベルを次に示す。

: 人名
: ホテル名
: 当方
: 人敬称
: ですので
: 繋ぎ副詞
: 詫び副詞
: 短い感動詞系
: 長い感動詞系
: 同意返事
: 時間
: 日付
: 数字
: 人数
: 曜日
: 月
: 金額
: 人単位
: 宿泊単位
: 予約オプション
: 部屋種類
: カード
: 風呂
: 時間問い
: 到着尋ねる
: 依頼
: 受諾
: 確認
: 宿泊
: 無部屋詫び
: 希望
: 推測伝達
: 問い合わせ
: 理由
: 限定
: ある

表 3.1: 音響処理条件

Analysis conditions	
Sampling rate	12 kHz
Window type	Hamming
Frame shift	10 msec
Acoustic parameters	16 order LPC cepstrum, log power, 16 order Δ LPC cepstrum, Δ log power
state-shared context-dependant HMM [8]	
Speaker independent	
400 states, 1 mixture + 1 state, 10-mixture pause model	

1.3　テスト用音響データ

例文集として用いた ATR 旅行会話データベースの収録音声をテスト用の音響データとして用いた。

実験をオープンテストとして行なうため、ある音響データを処理する際には対応する例文を例文集から外して処理を行なうようにした。

1.4　音響処理

音声認識モジュールとして、不特定話者連続音声認識システム [3] を用いた。

音響処理条件を Table 3.1 に示す。

1.5　実験結果

音声認識部が１００％に近い性能を出しているのならそのまま言語処理すれば良い。音声認識部が極めて低い性能しか出せないのであれば、GD-SR が正しく機能することは期待できない。GD-SR が最も有効に機能すると思われるのは、ある程度認識できているが、言語処理部が期待するほどの精度は出ていない、という状況である。

そこで、実験対象となるテストセットを以下のように選択した。

まず、既存手法による統計言語モデル [5] を用いて音声認識を行ない、認識結果を得る。

その中から、most-likely path の単語正解率、best-accurate path の単語正解率、のいずれもが５０％から８０％の間になるような文を選択した。

これらの文は、

- 言語処理部にとっては不充分な認識結果であり、
- 並べ替えによっても正解は得られない

ものである。

この条件を満たす文は全部で９６文あり、これをテストセットとした。テストセットは以下の通りである。

```
TAS12001.0010   TAS12001.0060   TAS12001.0070   TAS12001.0110
TAS12001.0130   TAS12001.0140   TAS12001.0150   TAS12001.0160
TAS12001.0290   TAS12001.0340   TAS12001.0430
TAS12002.0020   TAS12002.0070   TAS12002.0110   TAS12002.0140
TAS12002.0220   TAS12002.0230   TAS12002.0360   TAS12002.0420
TAS12003.0030   TAS12003.0060   TAS12003.0140   TAS12003.0170
TAS12003.0230   TAS12003.0240   TAS12003.0260   TAS12003.0340
TAS12004.0060   TAS12004.0070   TAS12004.0230   TAS12004.0290
TAS12004.0300   TAS12004.0340   TAS12004.0370   TAS12004.0390
TAS12006.0030   TAS12006.0040   TAS12006.0150   TAS12007.0010
TAS12007.0120
TAS12008.0020   TAS12008.0030   TAS12008.0040   TAS12008.0050
TAS12008.0170
TAS12009.0090   TAS12009.0210   TAS12009.0220
TAS12010.0110
TAS12013.0110   TAS12013.0170   TAS12013.0250   TAS12013.0260
TAS12014.0030   TAS12014.0090   TAS12014.0140   TAS12014.0150
TAS12014.0370
TAS12015.0190
TAS12016.0020   TAS12016.0040   TAS12016.0060   TAS12016.0100
TAS12016.0120   TAS12016.0170
TAS12017.0010   TAS12017.0050   TAS12017.0190
TAS12018.0100   TAS12018.0190   TAS12018.0280   TAS12018.0370
TAS12019.0050   TAS12019.0070   TAS12019.0080   TAS12019.0090
TAS12019.0220
TAS12020.0020   TAS12020.0030   TAS12020.0050   TAS12020.0170
TAS12020.0260   TAS12020.0280   TAS12020.0390
TAS12022.0170   TAS12022.0220
TAS12023.0110   TAS12023.0140
TAS12024.0030   TAS12024.0060
TAS12025.0030   TAS12025.0190   TAS12025.0220
TAS12026.0020   TAS12026.0260   TAS12026.0290
```

表 3.2: WORD ACCURACY

	accuracy	INS	DEL	SUB
Baseline	66.33%	0.68	0.70	2.88
GD-SR	72.50%	0.46	1.05	1.97

表 3.3: LANGUAGE TRANSLATION RESULTS

	No	Bad	Acceptable	Good
Baseline	39%	37%	13%	11%
	76%		24%	
GD-SR	31%	18%	12%	39%
	49%		51%	

９６文のうち４４文に対して検証モジュールからの出力が得られたので、この４４文については、この出力を GD-SR の結果とした。残りの５２文については、検証モジュールからの出力は得られなかったので、１度目の認識の結果を GD-SR の結果とした。

実験結果を Table 3.2、 Table 3.3に示す。

Table 3.2の中の “accuracy”, “INS”, ”DEL”, “SUB” はそれぞれ、単語正解率、１文中の平均湧出誤り数、１文中の平均脱落誤り数、１文中の平均置換誤り数、を示している。

GD-SR は湧出・置換誤りを効果的に抑制し、単語正解率も６％向上している。

又、さらに重要なのは翻訳成功率である。Table 3.3 は認識結果の翻訳結果の評価をまとめたものである。Table 3.3 の中の “No” 、 “Bad” 、 “Acceptable” 、 “Good” はそれぞれ、翻訳できない、意味的におかしい、多少おかしいが意味内容は把握できる、意味的・文法的に適切、な結果であることをそれぞれ示している。前２者を失敗、後２者を成功と、それぞれとらえることができる。

N-gram を用いた場合、全体の４分の１しか翻訳できないのに対し、 GD-SR では、全体の半分以上が翻訳できている

適切に言語処理可能な認識結果を出す、という我々の目的に対して、 GD-SR が有効に機能していることを、この実験結果は示している。

2 考察

実験結果は GD-SR の有効性を示している。湧出・置換誤りは減少し、単語正解率も向上し、そして一番の目的である、言語処理（翻訳）の成功率も向上している。

しかし、実験により、幾つかの問題点も明らかになった。

1つ目の問題点は検証モジュールが何の結果も出せないことが良くあるということである。今回の実験では、全体の半数近くに対して結果を出せていない。具体的に言えば、ATRlattice 実行中に workarea 不足により処理が中断してしまうことが多い。このことは、GD-SR は 統計言語モデルに比べて頑健性で劣るということを意味している。

主な原因は文仮説生成能力の不足である。未知のパターンの文は、今回の実験のような例文ベースの方式では生成できない。しかし、この問題は、例文を追加していくことによって克服可能と考えている。又、間投詞・感動詞など例文に現れない単語（本実験では filler word と称している。）の検出に失敗した場合も仮説生成がうまくいかないことが多い（短い単語であれば強引にスキップすることも可能であるが、現実にはうまくいかないことが多い）。これに関しては、1度目の認識において脱落を極力減らす工夫がさらに必要である。

2つ目の問題点は脱落誤りが増加していることである。主な原因は文の最後の文節の脱落である。通常の N-gram を用いたような認識であれば、最後の1単語の音響尤度が著しく低くて脱落しても、それは1単語の脱落にしかならない。しかし、GD-SR では、最後の1単語の脱落はそのまま最後の1文節の脱落になり、結果として複数単語の脱落になってしまう。なんらかのサーチ上の工夫が必要であると考えている。

3つ目の問題は必ずしも全ての認識誤りが解決できるわけではないことである。主な原因は、文仮説生成手法が必ずしも正解を生成しないことである。今回の実験でも、複数の不正解が生成されている。例えば、「部屋お願いします」という発声に対して「部屋をお願いします」という仮説だけしか生成できないことがある。発声（正解）では助詞の「を」が省略されていても、「を」を省略した例文が無ければ、そういう文仮説は生成できない。これは湧出誤りとして扱われることになる。

しかし、実験結果を見てみる限り、この種の誤りがシステム全体のパフォーマンスに対して致命的な悪影響を与えているようには見えない。こうした湧出が起きるのは、前後からその単語が容易かつ一意に推定できる場合に限られているからであると考えている。

4つ目の問題は、処理効率の問題である。他の目的で作られたツールを流用しているという理由もあるものの、従来法と比較してかなり多くの処理時間とメモリ容量とが必要（平均で約10倍）となっている。実用上の観点からは、この問題の解決は必須である。

第 4 章

まとめと今後の課題

1　まとめと今後の課題

本報告では、生成駆動音声認識方式 “Sentence Generation Based Speech Recognition (GD-SR)” を提案した。GD-SR は言語制約を動的に生成することによって、言語処理部にとって望ましい音声認識結果を得るようにした音声認識手法である。

実験の結果、GD-SR はより正確な単語認識率を達成できた。又、従来難しかった内容語の脱落誤りの救済もある程度実現できた。それにより、音声翻訳の結果も改善することができた。

内容語の脱落を救済できるようになったのは、意味的な情報を取り込んだことが大きな理由である。今回の実験では、単語群設定や変形ルールは手書きの簡単なものを用いたが、これらは大規模データや世界知識などを用いることによりより正確・機械的に行なうことが可能なはずである。

今後、より適切な意味表現、文生成手法を用いることが可能となれば、GD-SR の意味解析モジュール、文仮説生成モジュールをそれらに対応させることも考えられる。しかし、依然として GD-SR の枠組は有効であろうと期待している。

第 5 章

謝辞

日頃御指導頂き、又本発表の機会を与えて下さったＡＴＲ音声翻訳通信研究所第１研究室の匂坂室長に感謝致します。

参考文献

[1] Paolo et. al., "Improving speech understanding performance through feedback verification," Speech Communication 11 pp. 289-297 (1992)

[2] Takebayashi et. al., "A REAL-TIME SPEECH DIALOGUE SYSTEM USING SPONTANEOUS SPEECH UNDERSTANDING," Proc. of ICSLP '92, pp. 651-654 (1992)

[3] Shimizu et. al., "Spontaneous dialogue speech recognition using cross-word context constrained word graphs," Proc. of ICASSP '96, pp. 145-148 (1996)

[4] Martin Oerder, "WORD GRAPHS: AN EFFICIENT INTERFACE BETWEEN CONTINUOUS-SPEECH RECOGNITION AND LANGUAGE UNDERSTANDING," Proc. ICAASP 93, II-119 (1993)

[5] Masataki et. al., "VARIABLE-ORDER N-GRAM GENERATION BY WORD-CLASS SPLITTING AND CONSECUTIVE WORD GROUPING," Proc. of ICASSP '96, pp. 188-191 (1996)

[6] Morimoto et. al., "Speech and language database for speech translation research," Proc. of ICSLP '94, pp. 1792-1794 (1994)

[7] Nakamura et. al., "Japanese Speech Database for Robust Speech Recognition," Proc. ICSLP '96, pp. 2199-2202 (1996)

[8] Singer et. al., "Maximum likelihood successive state spliting," Proc. ICASSP-96, pp. 601-604 (1996)

[9] O.Furuse, H.Iida, "Constituent Boundary Parsing for Example-Based Machine Translation," Proc. COLING'94, pp.105-111 (1994)

[10] "http://www.itl.atr.co.jp/ singer/software/SPRECDOC/release.html"

[11] 高橋、中村," 単語グラフから例文集へのマッピング," 情報処理学会平成 9 年春季全国大会, 2-41 (1997)