多値論理研究ノート　　第26巻　第4号　　2003年9月
Note on Multiple-Valued Logic in Japan
Vol.26 No.4 September 2003

音声認識ブラウザーとその生成システムについて

Voice Browsers and Their Generation System

伊勢野 総[1]　井口 幸洋[1]　笹尾 勤[2,3]

Atsumu ISENO[1]　Yukihiro IGUCHI[1]　Tsutomu SASAO[2,3]

[1] 明治大学 理工学部 情報科学科
[2] 九州工業大学 情報工学部 電子情報工学科
[3] 九州工業大学マイクロ化総合技術センター
[1]Dept. of Computer Science, Meiji University
[2]Dept. of Electronics and Computer Science, Kyushu Institute of Technology
[3]Center for Microelectronic Systems, Kyushu Institute of Technology

概要　画面上に表示された単語リスト中の単語を音声認識し, 表示を変更するブラウザーの開発を行った. また, ブラウザーコンテンツの作成方法を考察し, ユーザーが簡単に観光案内やメニュー検索のアプリケーションを生成できるシステムを作成した.

1　はじめに

音声認識の技術レベルが発達し, 利用者の音声の特徴を学習させておけば, 明瞭な発声で読み上げた文章をほぼ完全に口述筆記することが可能になった. また, 自動車内の雑音環境でもカーナビゲーションの音声入力が可能になった. この技術を用いると, 手がふさがった状態で音声により電子機器を操作できる.　本稿は, 音声認識アプリケーション生成システムを開発したので報告する. 例えば, 観光案内アプリケーションが生成可能である.

大規模なアプリケーションや，追加修正が頻繁に行われるアプリケーションの場合，これを人手で処理するのは能率が悪い．本報告では，探索が容易な決定グラフの構築および，決定グラフの探索を能率よく実行できるように，リンクを張る作業を自動的に行うシステムついて述べる.

第 2 章で, 音声認識ブラウザーの構造, 第 3 章で, 音声認識アプリケーション生成システム, 第 4 章で, 国立公園案内の音声認識アプリケーションの作成方法を具体的に述べ, 第 5 章では, まとめと今後の課題について述べる.

2　音声認識ブラウザーの構造

本章では音声認識ブラウザーの仕様, 構造, 作成方法について概説する.

2.1　音声認識エンジン

現在, カーナビゲーションや携帯端末など, 様々な電子機器に音声入力制御が用いられている. 音声認識エンジンにはいくつかの種類があり, 用途に合わせて音声認識エンジンの選択を行う必要がある. まず, 音声認識エンジンは, 連続音声認識と単語認識に大きく分類できる. 連続音声認識エンジンは, 自然な発話をそのまま入力に使用できる. 自由度が高いため, 様々な応用ができる. しかし, 口語など, 文法の砕けた入力も考えられるため, 認識が難しく, 認識時間が長い. 一方, 単語認識では, 認識可能な単語を, 予めアプリケーションの開発時にリストに登録する. 連続音声認識に比べて認識精度が高く, 高速であり, 制御系のアプリケーションに有用である. カーナビゲーションでは単語認識のエンジンを用いているものが多い.

また, 音声認識エンジンは不特定話者対応, 特定話者対応でも分類できる. 特定話者対応のエンジンでは, 利用前に話者が, アプリケーションに対して数分〜数十分のトレーニング（エンローリング）をする必要がある. 話者の特徴を学習させることにより, 認識率が上がるが, 話者が変わる度に新たなエンローリング, もしくは設定の変更が必要である. 連続音声認識エンジンは, 音声認識の困難さのためか, 特定話者対応のものが多い. 一方, 単語認識エンジンは不特定話者対応のものが多い. 不特定話者対応のエンジンでは, エンローリングの必要はなく, 話者がただ単語を発声するだけでよい. また, 途中で話者が交代しても問題ない.

その他，音声認識手法として，統計確率的手法である

図 2.1: 音声認識ブラウザーの画面

HMM(Hidden Markov Model:隠れマルコフモデル）による方法と，人間の神経網の挙動の数学的モデルを用いたNN(Neural Network:ニューラルネットワーク) による手法が存在する．本音声認識ブラウザーでは，不特定話者対応，単語認識，HMM のエンジンを採用した．

2.2 音声認識ブラウザーの仕様

本音声認識ブラウザーは日本人，外国人両方に対応している．日本国内の観光案内や，飲食店のメニュー，百科事典等に使用することを目的に開発した．図 2.1 に作成した音声認識アプリケーションの実行画面を示す．図 2.1 の画面左側に表示された単語リストに基づいて発声することにより，画面表示と単語リストを順次入れ替え，目的とする情報を引き出す．ソフトウェアは VisualC++.NET を用いてWindows 用に開発した．

2.3 音声認識アプリケーションの作成方法

次に音声認識アプリケーションの作成方法を簡単に説明する．最初にアプリケーションの論理構造を示す多値決定グラフを作る．例として，”都道府県案内”の決定グラフを図 2.2 に示す．このグラフにおいて，順々に枝を辿ることにより，最終的に目的の都道府県に到達する．グラフの各節点には単語リストと 1 つのコンテンツファイルが対応する．各節点からの枝には単語リスト中の単語が対応する．下階層に向かって枝を持たない節点を終端節点と呼ぶ．本音声認識ブラウザーでは，次のファイルを用いる．

- コンテンツファイル

  画像とテキストから成るファイル．決定グラフの節点数だけ必要である．図 2.1 の画面右部分に対応する．HTML や JPEG，GIF など，Internet Explorer で表示可能なフォーマットで作成する．

表 2.1: 節点 0 の語彙ネットワーク

単語 ID	ラベル	発音
1	北海道	ほっかいどう
1	北海道	ほっかいどー
2	本州	ほんしゅう
2	本州	ほんしゅー
3	四国	しこく
4	九州	きゅうしゅう
4	九州	きゅーしゅー
5	沖縄	おきなわ

- 節点ネットワークファイル

  決定グラフの各節点における表示ファイルおよび，リンク先の接続関係を示すファイル．ブラウザーは，各節点で単語リスト内の単語を認識し，制御を接続先の節点へ移動する．さらに，表示を次のコンテンツに入れ替え，音声認識エンジンへ次の節点の語彙ネットワークを与える．この時，接続先情報を保存しているのが節点ネットワークファイルである．各節点は固有の節点番号をもつ。認識した単語の単語 ID を与えると，次に辿る節点の節点番号へ変換するための情報を貯える．単語 ID とは単語に整数を対応させたもので，詳しくは語彙ネットワークファイルの説明で述べる．図 2.3 に節点ネットワークファイルのデータ構造と，記述例を示す．99999 は行の終了コードである．

- 語彙ネットワークファイル

  単語 ID，ラベル，発音を対応付けるファイル．終端節点以外の各節点は語彙ネットワークファイルを持つ．表 2.1 に節点 0 の語彙ネットワークファイルを示す．音声認識後，同じ単語として処理すべき語には，単語 ID として同じ整数を割り当てる．このファイルは音声認識エンジンが使用する．語彙ネットワークファイルの作成には，音声認識エンジン付属の専用ツールを使用する．

- 単語リストファイル

  全節点の単語リストをまとめたファイル．データは，語彙ネットワークのラベル部分からなる．各節点毎に単語リスト中の単語を単語 ID 順に並べたもので，各リストの最後には終了コードの 99999 を付加している．図 2.4 に単語リストファイルのデータ構造と，記述例を示す．

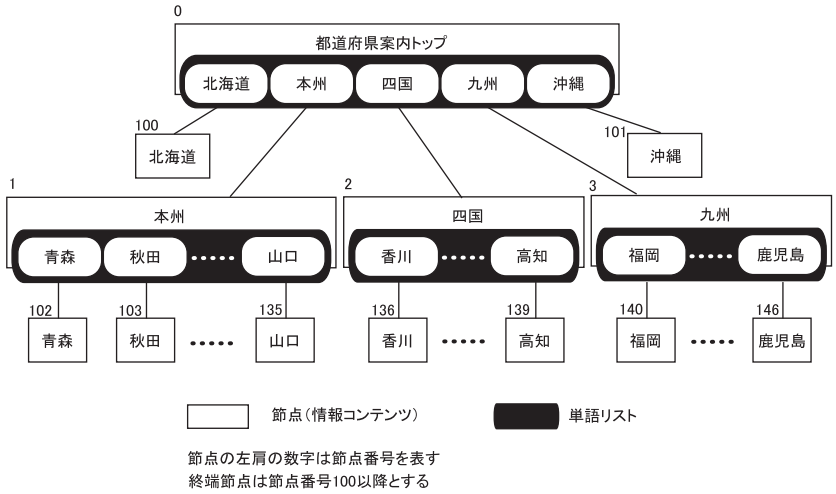

図 2.2: 都道府県案内の決定グラフ

2.4 英語入力への対応

本音声認識ブラウザでは外国人と日本人の両方のユーザーに対応させるため, 内部で, 英語と日本語の音声認識エンジンを並列に動作させている.

日本語音声認識エンジンは, ひらがな, またはカタカナで表現可能な任意の単語を認識可能である. 英語音声認識エンジンを使用する場合, 予め綴りと発音記号 (CMU Phoneme) の対を, **辞書ファイル**に登録する. いずれの場合でも, 辞書ファイル中に存在する単語のみ, 音声認識が可能である. 外国人向けに日本国内を観光案内するには, 英語と日本語の両方の音声認識エンジンによる並列認識が必要である. 例えば, ”阿蘇”の観光案内を行う場合, 単語リストには, ”猫岳”, ”火山”, ”カルデラ”, ”ビジターセンター”などの単語を登録する. この単語リストにて, 外国人の音声検索を実行する際, ”猫岳”, ”Volcano”, ”Caldera”, ”Visitor Center”の中から照合を行う必要がある. ”猫岳”は日本の地名なので, 英単語には存在しない. このような単語は, 英語音声認識エンジンでの認識が困難であり, 日本語音声認識エンジンで認識した方がよい. 日本語の単語を, 英語辞書に発音記号で登録する方法も可能だが, 発音やアクセントが人によって大きく異なるため認識は困難である. このように, 日本固有の単語は, 日本語として認識する必要がある. また, 日本語音声認識エンジンのみで英語の認識も試みた. 例えば, ”Volcano”を”ヴォルケーノ”というように, 仮名で表す. この場合, 日本人の発音は, 認識可能であったが, 外国人の発音はアクセントが強く, 認識困難であった.

これらの理由により, 本音声認識ブラウザでは, 英語と, 日本語の二つの音声認識エンジンを並列に動作させる方法を採用した.

［データ構造］

ネットワーク番号, 表示ファイル名, 単語ID(1)のリンク先のネットワーク番号,
単語ID(2)…単語ID(N), 99999

99999 ：行の終了コード（必須）

［記述例］

```
0, c:¥test¥page_00.html, 100, 1, 2, 3, 101, 99999
1, c:¥test¥page_01.html, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113,
        114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
        130, 131, 132, 133, 134, 135, 99999
2, c:¥test¥page_02.html, 136, 137, 138, 139, 99999
3, c:¥test¥page_03.html, 140, 141, 142, 143, 144, 145, 146, 99999
                    ⋮
100, c:¥test¥100.jpg, 99999
101, c:¥test¥101.jpg, 99999
102, c:¥test¥102.jpg, 99999
                    ⋮
145, c:¥test¥145.jpg, 99999
146, c:¥test¥146.jpg, 99999
```

図 2.3: 節点ネットワークファイルのデータ構造と記述例

3 音声認識アプリケーション生成システム

音声認識ブラウザの実行で用いるファイルの作成方法を 2 章で述べた. 目的とするアプリケーションの決定グラフが大きくなると, 必要となるファイルも増え, 管理が困難になる. そのため, これらを一括管理し, ファイルの生成を支援するシステムを開発した. 生成システムの実行画面を図 3.1 に示す.

この生成システムでは, 節点毎に単語リストおよび, 単語に対応するコンテンツが即座に確認できる. ファイルをテキスト・エディタなどで一つ一つ作成する場合に比べてバグが入り難い. 手作業で必要ファイルを作成する場合の手順を図 3.2(a), 生成システムを使用した場合の手順を図 3.2(b) に示す. 手作業で作成する場合, 節点ネットワークファイル,

［データ構造］

ネットワーク番号, 単語ID(1)のLabel, 単語ID(2)のLabel, …, 単語ID(N)のLabel, 99999

99999 ：行の終了コード（必須）

［記述例］

0, 北海道, 本州, 四国, 九州, 沖縄, 99999
1, 青森, 秋田, 岩手, 山形, 宮城, 福島, 新潟, 石川, 富山, 群馬, 茨城, 茨城, 栃木,
福井, 埼玉, 千葉, 東京, 神奈川, 山梨, 長野, 静岡, 岐阜, 愛知, 三重, 奈良,
和歌山, 滋賀, 京都, 大阪, 兵庫, 岡山, 広島, 島根, 鳥取, 山口, 99999
2, 高知, 香川, 愛媛, 徳島, 99999
3, 福岡, 大分, 宮崎, 長崎, 熊本, 鹿児島, 佐賀, 99999

図 2.4: 単語リストファイルのデータ構造と記述例

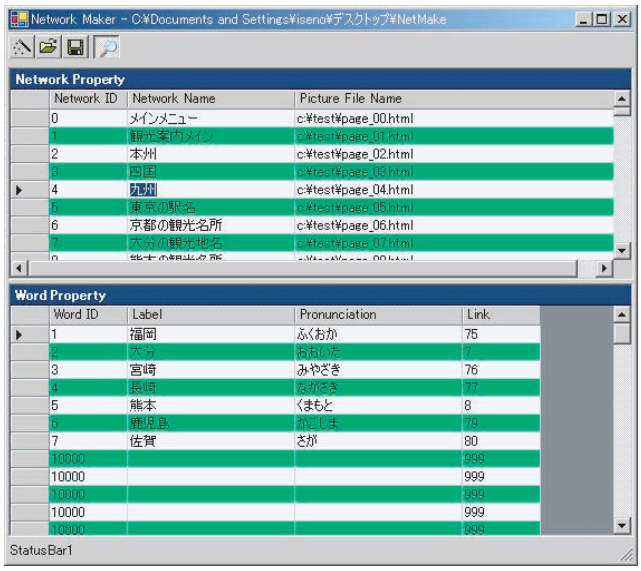

図 3.1: 生成システムの実行画面

語彙ネットワークファイル, 単語リストファイル, コンテンツファイルの接続に誤りが存在した場合, 誤りの箇所を特定するのが非常に困難である. 情報が各ファイルに分散しており, また各節点への接続がポインタ化されているため, 人手で追うのは大変な作業で時間がかかる. また, 全ファイルを作成後でないと動作の確認ができないため, バグが混入し易い. 一方, 生成システムを使用する場合, コンテンツの作成までは手作業の場合と同じであるが, 節点毎に単語リスト, 接続先などの必要な情報を支援ソフトウェア上で入力するだけで, 全ファイルを自動生成できる. 生成システムでは, リンク先のコンテンツを画面上で即座に確認できるため, 動作の検証を行いながらデータ入力できる. すべての情報を一括管理しているため, 入力も単純でバグの混入も少ない. この生成システムを使用することで, 音声認識アプリケーションを簡単に短時間で作成できる.

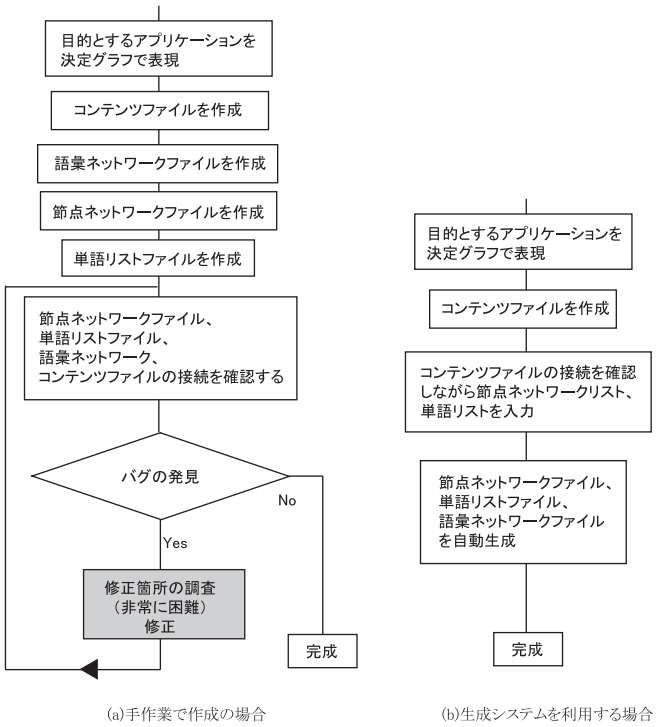

図 3.2: 手作業との比較

4 決定グラフの構成方法

音声認識ブラウザの実行で用いるファイルの作成を, 手動で行う場合でも, 生成システムで行う場合でも, コンテンツの作成は不可欠である. コンテンツを作成するとき, 最初に決定グラフを考える. 本章では, 決定グラフの構成方法について述べる.

決定グラフの各節点において, ユーザが遷移方向を指定する対話型データ検索を考える. 目的データを検索するため, ユーザがディスプレイに表示されたメニューを見ながら節点の遷移方向を決定する. この場合, 決定グラフの節点数が多くなると, ユーザが目的データにアクセスするまでの遷移回数が多くなる. また, 一つの親節点に対する子節点の数が多くなると, 子節点にアクセスするためのメニューが煩雑となり, メニューの中から所望の項目を選び出す効率が悪くなる. 特に, 音声入力によってメニューを選択する検索に適用する場合, メニューの項目数が増えると, 音声認識率が低下し, 検索効率が低下する.

木構造により階層化されたデータから目的のデータを効率よく検索できるようにするため,

1. 属性によって木構造を分割し, 一つの親節点に対する子節点の数を減らす. 非終端節点で属性テストを行い, 対象を絞ることで目的の終端節点に近づく.

2. 根節点を除く非終端節点においては, 根節点, 親節点, および子節点にリンクを張り, 終端節点においては, 根節点, 親節点, および兄弟節点にリンクを張る.

3. 属性テストの順序を変更することにより, テストの数

がなるべく少なくなるようにする．

4. それ以上の属性テストの追加が不可能で，一つの非終端節点に対する子節点の個数が多い場合，子節点の個数を一定以下に制限するため，子節点を幾つかのグループに分割し，其々のグループに対して部分木を構成し，部分木間を線形リンクで結合して，全ての探索を可能にする．

5. 以上の決定グラフを自動的に生成するシステムを用いる．

各節点から根節点にリンクを張ることによって，任意の節点から 1 ステップで初期状態に移行できるようにする．また，各終端節点から，兄弟節点にリンクを張ることによって，性質の似た要素間を容易に移動できるようにする．上記のような決定グラフを構成すれば，平均的にデータの検索時のパス長が短くなり，効率の高いデータ検索技術を提供できる．本音声認識ブラウザでは，根節点，親節点，兄弟節点へのパスは自動的に追加されるため，ユーザが意識する必要はない．

4.1 国立公園の紹介システム（図 4.1 参照）

全国 28 カ所の国立公園を音声認識ＷＥＢ形式で紹介する．地域別に公園を分割していくメニューを用いると，3〜4 ステップで特定の公園のページに到達できる．この例の場合，属性テストによって地域を絞る．スライド 1 は，根節点に対応する．このスライドから 4 つのリンク (北海道，本州，四国，九州・沖縄）が出ている．スライド 2 は，非終端節点（北海道）に対応する．この節点は，６つの子節点に対するリンク（利尻礼文サロベツ, 知床, 阿寒, 釧路湿原, 大雪山, 支笏洞爺）をもつ．また，親に戻るリンク（上に戻る）もある．スライド 3 は，終端節点（利尻礼文サロベツ）に対応する．この場合，兄弟に対するリンク（, 知床, 阿寒, 釧路湿原, 大雪山, 支笏洞爺），親に対するリンク（上に戻る），根に対するリンク（ホームに戻る）もある．

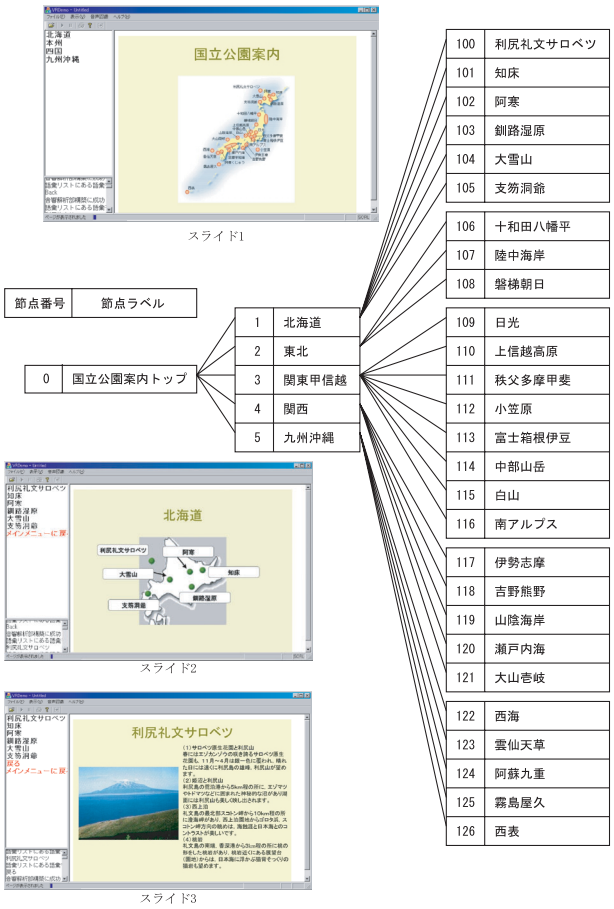

図 4.1: 国立公園案内の決定グラフ

4.2 東京レストランガイド

本アプリケーションは，東京のレストランを音声認識 WEB 形式で紹介する．表 4.1 に東京レストランガイドの情報を示す．この場合，3 個の属性テスト ($X1$:料理ジャンル, $X2$:地域, $X3$:目的) がある．$X1$ は，和食, 洋食, 中華, エスニック, 麺類の 5 つの値をとる．$X2$ は，新宿区, 渋谷区, 港区, の 3 つの値をとる．$X3$ は，デート, 接待, ファミリー, 一人の 4 つの値をとる．全部で, $5\times3\times4=60$ の組み合わせが存在するが, 対応するレストランが存在しない場合や, 対応するレストランが複数存在する場合がある．

この場合，3 つの変数は独立であり，どの順序で質問してもよい．決定木を構成する際, なるべく早く, 目的の要素

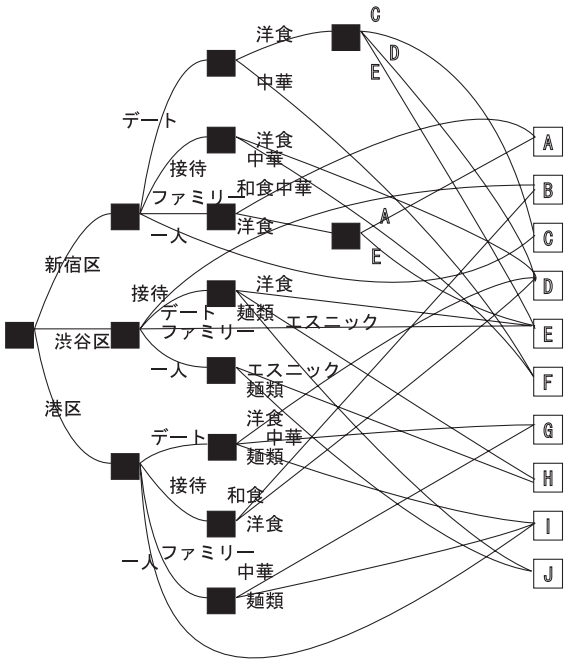

図 4.2: 東京レストランガイドの決定木 (地域-目的-料理ジャンル)

表 4.1: 東京レストランガイドの情報

店名	料理ジャンル					地域			目的			
	和	洋	中	亜	麺	新	渋	港	デート	接待	ファミリー	一人
A	○	○	○			○					○	
B	○						○	○		○		
C		○				○			○			○
D		○				○		○	○	○		
E		○				○	○		○		○	
F			○			○			○	○		
G			○					○	○		○	
H				○			○		○			○
I					○			○	○		○	○
J					○		○		○			○

和:和食 洋:洋食 中:中華 亜:エスニック 麺:麺類
新:新宿区 渋:渋谷区 港:港区

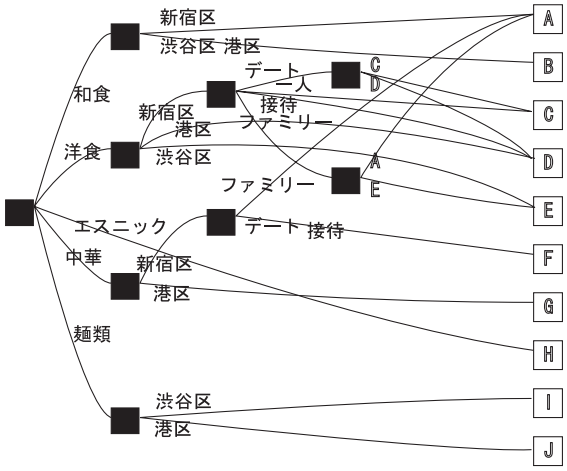

図 4.3: 東京レストランガイドの決定木 (料理ジャンル-地域-目的)

に到達するように属性テストを生成する．つまり, 重要なテストを最初に行う． レストランの案内では, 根節点に, 属性テストの順序を決めるメニューを入れておけば, 重要な属性から, 決定でき, あまり重要でないもの（例えば, 目的）が最後の属性テストとなる．図 4.2, 図 4.3 に属性テストの順番を変えた場合の決定木を示す. 図 4.2 は地域, 目的, 料理ジャンルの順に検索を行う. 図 4.3 は料理ジャンル, 地域, 目的の順に検索を行う．それぞれの平均パス長は 2.68 と 1.90 となっており, 後者のほうが検索効率がよい．この例では, 属性テストの順番はすべてのパスで固定だが, パスによって属性テストの順番を変えたほうが検索効率がよくなる場合もある.

また, 兄弟節点には, ユーザが興味のある要素がある可能性が高い. 兄弟節点に簡単に移動できれば, 所望の要素を能率よく選択可能となる．実際には, 東京の銀座だけでも, 日本料理屋が 200 近くあり，変数が三つでは，一つの節点に対応する要素数が多くなり過ぎる．そのため，地域をさらに細分するか，値段や，店の名前等で分割する必要がある．ただし，その作業が不可能な場合は，自動的に料理屋を指定された程度以下のグループに分割し，各画面で選択が容易なようにする.

5 国立公園案内用アプリケーション

本章では国立公園案内用アプリケーションを生成する手順について述べる.

1. 情報収集

   コンテンツファイル用の画像やテキスト情報の収集を行う.

2. 決定グラフの作成

   4 章で述べた構成法に基づき, 決定グラフを作成する. 図 4.1 に国立公園案内の決定グラフを示す. これは一般には多値決定グラフとなる.

3. コンテンツの作成

   html エディタ, ペイントソフト等を使用し, 節点数に等しいだけのコンテンツを作成する.

4. 単語リストを生成システムに入力

   図 4.1 の決定グラフから単語リストを作成し, 生成システムに入力する. 表 5.1 に単語リストを示す.

表 5.1: 国立公園案内の単語リスト

節点 0：国立公園案内トップ

単語 ID	ラベル	発音	リンク
1	北海道	ほっかいどう	1
2	東北	とうほく	2
3	関東甲信越	かんとうこうしんえつ	3
4	関西	かんさい	4
5	九州沖縄	きゅうしゅうおきなわ	5
1	北海道	ほっかいどー	1
5	九州沖縄	きゅーしゅーおきなわ	5

節点 1：北海道

単語 ID	ラベル	発音	リンク
1	利尻礼文サロベツ	りしりれぶんさろべつ	100
2	知床	しれとこ	101
3	阿寒	あかん	102
4	釧路湿原	くしろしつげん	103
5	大雪山	たいせつざん	104
6	支笏洞爺	しこつとうや	105

節点 2：東北

単語 ID	ラベル	発音	リンク
1	十和田八幡平	とわだはちまんたい	106
2	陸中海岸	りくちゅうかいがん	107
3	磐梯朝日	ばんだいあさひ	108

節点 3：関東甲信越

単語 ID	ラベル	発音	リンク
1	日光	にっこう	109
2	上信越高原	じょうしんえつこうげん	110
3	秩父多摩甲斐	ちちぶたまかい	111
4	小笠原	おがさわら	112
5	富士箱根伊豆	ふじはこねいず	113
6	中部山岳	ちゅうぶさんがく	114
7	白山	はくさん	115
8	南アルプス	みなみあるぷす	116

節点 4：関西

単語 ID	ラベル	発音	リンク
1	伊勢志摩	いせしま	117
2	吉野熊野	よしのくまの	118
3	山陰海岸	さんいんかいがん	119
4	瀬戸内海	せとないかい	120
5	大山壱岐	だいせんいき	121

節点 5：九州沖縄

単語 ID	ラベル	発音	リンク
1	西海	さいかい	122
2	雲仙天草	うんぜんあまくさ	123
3	阿蘇九重	あそくじゅう	124
4	霧島屋久	きりしまやく	125
5	西表	いりおもて	126

5. 生成システムによって節点ネットワークファイル, 語彙ネットワークファイル, 単語リストファイルを出力する.

6 まとめと今後の課題

本稿では, 音声認識アプリケーションの例と, コンテンツ作成の手順について示した. 本アプリケーションでは, 雑音に強い音声認識エンジンを用いたが, 環境や話者によって認識率はかなり変化した. 雑音や話者に対するパラメータの調整方法を検討する必要がある. また, 本稿では英語, 日本語の音声認識エンジンを並列動作させることで, 外国人に対応した. カナから, 発音記号を生成し, 辞書に登録する方法もある. この場合, アメリカ人とフランス人では同じローマ字表記に対する発音が異なるので, 考えられる全ての発音を記号化することが必要となる. また, 簡単な例を示して決定木の構成方法を述べたが, 実際の大規模な情報の中から, 効率良く目的を探し出すための決定木の構成方法なども検討していきたい [11, 12].

アプリケーションとしては, 国立公園案内の他に, 外国人に日本文化を教える百科辞典や, 外国人向けの日本観光案内などを, 開発している. 本システムは, 音声認識対話システム等における表示画面や, WEB による旅行・商品案内等のアプリケーションの開発に利用可能である.

謝辞

本研究は, 文部科学省・北九州地域・知的クラスタ創成事業の補助金による. 有益なご助言を賜った, 東京工業大学古井貞熙教授に感謝する. (株) マリックス北爪吉明氏, (株) 日立超 LSI システムズ, 旭化成 (株) の皆様には, 音声認識エンジンでお世話になった. コンテンツの作製には, Portland State University の Marek Perkowski 教授と九州工業大学大学院生の中原啓貴氏にお世話になった.

参考文献

[1] 駒谷, 河原, “音声認識結果の信頼度を用いた効率的な確認・誘導をおこなう対話管理, ” 情報処理学会論文誌, Vol.43, No.10, pp. 3078-3086, Oct. 2002.

[2] 古井, 音声情報処理, 森北出版, 1998.

[3] 鹿野他 (編著), 音声認識システム, オーム社, 2001.

[4] 河原達也, “ここまできた音声認識技術,” 情報処理, Vol.41, No.4, pp.436-439, 2000.

[5] 特許庁総務部技術調査課, 音声認識技術に関する特許出願技術動向調査報告, 2003-05

[6] 田熊 竜太, 岩野 公司, 古井 貞熙, “並列処理型計算機を用いた音声対話システムの検討,” 人工知能学会 研究報告, SLUD-A201-04, pp.21-26 (2002-6) .

[7] C. Kim, L. Lavagno, and A. Sangiovanni-Vincentelli, “Free MDD-based software optimization techniques for embedded systems,” In Proc. of the Conf. on Design Automation & Test in Europe, Mar. 2000.

[8] T. Sasao, M. Fujita, (eds.), Representations of Discrete Functions, Kluwer, 1996.

[9] T. Kam, T. Villa, R. Brayton, and A. Sangiovanni-Vincentelli, “Multi-valued decision diagrams: theory and applications,” Multiple-Valued Logic, Vol. 4(1-2), pp.9-62, 1998.

[10] B.M.E. Moret, “Decision trees and diagrams,” Computer Surveys, Vol. 14, No. 4, pp. 593-623, 1982.

[11] D. Popel and N. Hakeem,“Improving web database access using decision diagrams,” Proc. ACM/IEEE Int. Conf. on Computer Systems and Applications AICCSA’2001, 2001, pp. 519-525

[12] T. Sasao, Switching Theory for Logic Synthesis, Kluwer Academic Publishers, 1999.