音声言語情報処理 22－1
（1998．7．24）

発話同時理解に基づくマルチモーダル図形エディタ

河口信夫† 松原茂樹‡ 外山勝彦† 稲垣康善†

† 名古屋大学大学院工学研究科 計算理工学専攻
‡ 名古屋大学 言語文化部

マルチモーダルインタフェースのモダリティの一つとして音声言語を用いる場合，従来の一文を単位とする音声対話処理の枠組では，自然なインタラクションを実現することは困難であった．本稿では，音声言語を発話と同時に理解し，応答を返すマルチモーダルシステムについて，その有効性と問題点について述べる．また，マルチモーダル図形エディタ Sync/Draw の実現について報告する．Sync/Draw は，日本語音声言語情報とポインティング情報を入力にとり，単純な書換え規則を用いた意味構造の遷移によって実現された漸進的な言語解釈，描画処理により，入力と同時進行的に作図結果の適切なフィードバックを行う．また，図形や操作に対する照応が可能である．

Multimodal Drawing Editor Based on Simultaneous Speech Understanding

Nobuo Kawaguchi† Shigeki Matsubara‡ Katsuhiko Toyama† Yasuyoshi Inagaki†

†Department of Computational Science and Engineering,
Graduate School of Engineering, Nagoya University
‡ Faculty of Language and Culture, Nagoya University

It is difficult to make a natural speech dialogue system based on the conventional sentence-based language processing. In this paper, we propose a simultaneous speech understanding framework to develop a spontaneous speech dialogue system. We describe the effectiveness of the framework for the drawing task domain and clarify promlems to implement the system for Japanese. We also report a prototype mulimodal drawing editor Sync/Draw based on the framework.

1 はじめに

人と計算機のより自然で，より滑らかなインタフェースの構築のため，近年，マルチモーダルインタフェースの研究が盛んである [1, 2, 4, 5]．特に，モダリティの一つとして自然言語の音声入力を用いた場合は，初心者にとっては複雑な操作方法やコマンド等を覚える必要がなく，また，熟練者にとっても複雑な指示を容易に行うことができるという点において，効率的なインタラクションを実現することが期待される．音声インタフェースにおいて人と計算機の自然な対話を実現するには，自由な発話に対し，適切な反応をタイミング良く返すことが特に重要となる．しかし，従来の多くの音声対話システムは，一文入力後に解釈処理を行うため，文入力中に反応を返すことが困難であった．また，キーワードスポッティングを用いたシステム [5] では，キーワード入力に対し，素早い応答が可能であるが，複雑な自然言語文への対応は困難である．

一方，自然言語の処理手法として漸進的解釈が提案されている [3]．これは，入力を語順に従い，できる限り語単位で解釈する枠組である．この枠組を用い，音声入力を発話と同時に理解することにより，適切な応答を行うシステムの構築が期待できる．

本稿では，発話と同時にその意味を理解し，適切な反応をタイミング良く返すシステムの構築を目指す．特に自然言語処理の立場から，その有効性と日本語を用いる場合の問題点を明らかにし，タスクとしてユーザと計算機とのインタラクションが密である作図作業を対象とする．

また，プロトタイプシステムとして開発したマルチモーダル作図エディタ Sync/Draw について報告する [6, 7]．Sync/Draw は，入力と同時進行的に出力を行うことができ，細やかな反応を可能としている．

表 1: 音声インタフェースの解釈単位

システム名	音声認識単位	反応単位	入力の多様性	応答タイミング
インテリアデザイン支援 [1]	文	文	○	×
TOSBURG II[5]	キーワード	文	△	△
S-tgif[4]	単語	単語	×	○
Put-That-There[2]	単語	単語	△	○
Sync/Draw[7]	単語	単語	○	○

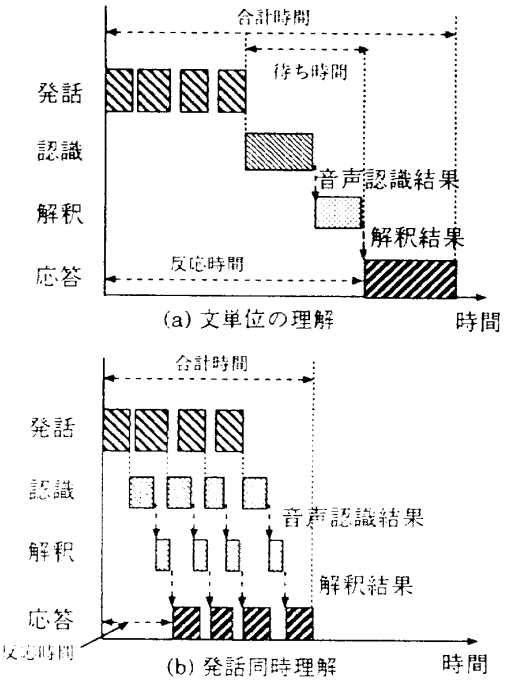


図 1: 処理タイミングの違い

2 発話同時理解の必要性

2.1 従来の音声インタフェース

従来の音声インタフェースについて反応のタイミング，及び入力の多様性[1]に注目し，評価した結果を表 1に示す．インテリアデザイン支援システム[2] [1] は，入力の多様性は高いが，発話が終るまでシステムが入力に反応できないため，応答タイミングは悪い．また，キーワードスポッティングを使って，単語単位で解釈を行う TOSBURG II[5] では，キーワードにより入力が制限されるため入力の多様性はやや低く，基本的には文入力の終端を判断してから応答を行うため，十分細やかな応答は行われない．一方，単語単位で解釈し応答を返す S-tgif[4] は応答のタイミングは適切であるが，入力の多様性は低い．また，古典的なマルチモーダルシステムである Put-That-There[2] では動詞が入力の先頭に来るという英語の特徴を利用し，動詞ごとに処理を定めている．そのため，入力の多様性は低いが，反応は適切なタイミングで行えている．

表 1が示すように，従来の枠組では，入力の多様性と応答タイミングの両立は，発話と同時に解釈を行う枠組を用いていないため困難であった．

2.2 発話同時理解に基づくインタフェース

我々の開発した発話同時理解に基づく図形エディタ Sync/Draw は，単語ごとに反応を返し，語順の入れ換えや照応に対応しており，ある程度の多様性と，素早い応答を実現している．また，畑崎らは発話同時理解に基づく新幹線チケット購入システム [8, 9] を構築し，その評価を行い，意味理解率が向上し，タスクの達成時間が短縮されたことを報告している．図 1に従来の文単位で理解を行うシステムと，発話同時理解に基づくシステムとの処理タイミングの違いを示す．この図から明らかなように，文単位のシステムでは，発話が終了するまで認識，解釈が行われないため，ユーザの待ち時間が生じる．一方，発話同時理解に基づくシステムでは，発話の小さな単位毎に認識，解釈が行われ，できる限り早い段階で応答が行われる．そのため，ユーザの待ち時間はなくなり，応答時間も短縮される．さらに，認識誤りが生じた場合には，ユーザによる即時訂正が可能となる．

しかし，これまでの音声対話のための音声認識システムでは，フレーズスポッティング [11] などの結果に対し，文脈自由文法や格に基づく文法 [10] などの一文単位での制約を用いることにより，連続音声認識率の向上を計ってきた．一方，発話同時理解の枠組では，一文単位ではなく，より小さな認識単位が必要となるため，音声認識技術の向上が望まれ

[1] ここでは「多様性」を，一つの指示に対応する語順の入換えや照応等によるいいまわしの多さという意味で用いている．

[2] 文発声のシステムを評価の対象とした．

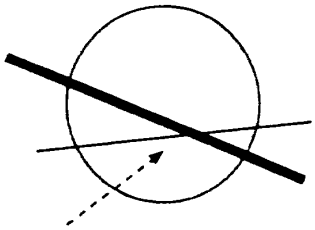


図 2: 音声によるポインティング指示の支援

る．また，発話途中の解釈結果を認識側へフィードバックすることにより，動的な探索範囲の変更等の認識率向上への寄与が期待できる．

2.3 作図タスクにおける発話同時理解

発話途中の意味を解釈し，以降の発話予測をできる限り正確に行うためには，タスクを限定することが望ましい．我々は，発話同時理解を評価するためのタスクとして，ユーザの意図が明確であり，インタラクションが密である作図タスクを対象とした．作図タスクは，図形を描画することが明確な意図として存在するため，予測が行い易い．作図タスクの特徴を以下に挙げる．

1. 作図はユーザとの対話により逐次的に進む．
2. 作図途中の図形が存在するため，常に画面上に反応を必要とする．
3. モードの概念がある（線，円，移動等）．
4. 照応は，画面上の図形に対して起こる．

さらに音声の利用による利点が多く存在する．S-tgif においても，多数のコマンドを音声コマンドにより階層メニューを辿らずに実行できることが利点として挙げられているが，我々の考える作図システムでは，より高度な表現が可能となる．例えば，図形に対する複雑な指示でも，「▽[3]この四角の内接円を描く」や「▽ これと ▽ この四角を左端で揃える」と，容易に発話可能である．また，図 2 に示すように，複数の図形が重なっている時，ポインティングデバイスのみ用いる従来のシステムでは，重なった図形に対する指示は困難であったが，音声システムでは例えば次のように容易に行える．

U:「▽ これを」
S: 円を選択状態として表示する
U:「あっ，それじゃなくて線を」
S: 細い線を選択状態にする
U:「太い線を」
S: 太い線を選択状態にする
U:「消して」

[3]▽ はポインティングを表す．

このように，システムの応答に対し修正を行うことが可能である．また同様の指示を「▽ この太い線を消して」と発話することも可能である．さらに，操作に対する照応も可能であり「さっき描いた円と同じ円を ▽ ここに描く」といった入力により，最近に描画した円と同じ大きさや色等の属性を持った円の描画を簡単に行える．このように，発話同時理解に基づくシステムは作図タスクに対して高い有効性を持つといえる．

しかし，日本語を入力とするシステムを実現するためには様々な問題点が存在する．第 1 に，日本語は語順が比較的自由な膠着言語であることが挙げられる．すなわち，日本語では語順よりも助詞によって定まる格が重要であり，これは助詞を利用しないキーワードスポッティングが自由度の高い文を扱えない理由でもある．例えば一本の線を描画する場合，「▽ ここから ▽ ここまで 線 を 描く」や「 線 を ▽ ここから ▽ ここまで 描く」など単純な線の描画でさえ「▽ ここから」「▽ ここまで」「線を」の 3 つの文節の組み合わせが 6 通りも存在する．

第 2 に，述語が文の最後に現れるため，発話途中では動作が確定しないことが挙げられる．すなわち，「描く」という動詞が出現するまで「線を」や「▽ ここから」の意味は確定しない．

第 3 に，ポインティングに対応する指示語や助詞の省略が挙げられる．例えば「▽ ここから ▽ ここまで 線 を 描く」という入力は，時には「▽ ▽ 線」のみで済ませる場合もありえる．

第 4 に作図タスクがモードを持ち，一般に作図システムはメニューによって駆動されているため，従来のメニューシステムとの整合性をとる必要がある．S-tgif のように発話が単語のみであれば，メニューとの整合は容易であるが，発話が省略された文や，文の一部であり，その後をメニューによる操作で補った場合の対応が問題となる．

これらの問題は，独立ではなく互いに関係しており，発話同時理解を行うためには，高度な解決法を必要とする．

3 図形エディタ Sync/Draw

我々は発話同時理解に基づくプロトタイプシステムとして Sync/Draw を実現した．Sync/Draw は，前節の第 1, 第 2 の問題解決を目的としている．複数の入力手段に対して統一的に意味を表現するためにオブジェクトと呼ばれる構造を用意し，語順の入換えには書換え規則による変更により対応した．

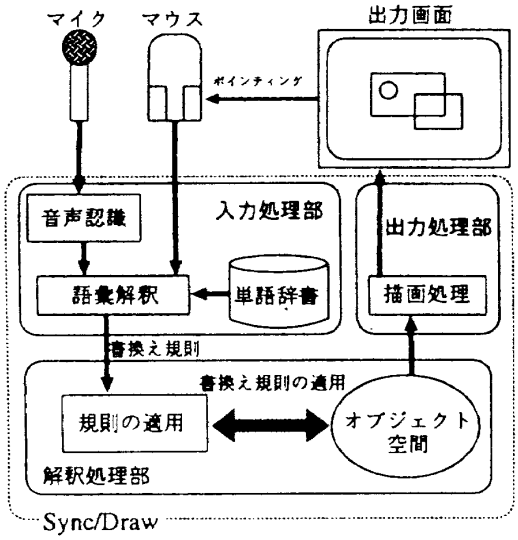


図 3: Sync/Draw の構成

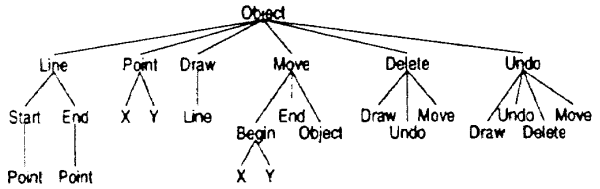


図 4: 属性間の階層構造

述語は，その意味を予測によって補うこととした．作図ドメインを分析することにより，述語が持つべき意味構造をオブジェクトの属性間の階層構造として定めた．本来は「描く」という語が入力されなければ「線を」と「▽ ここから」は互いに意味をなさないが，線と始点の関係づけにより，途中の段階での作図が可能となった．これは，作図ドメインでは，ほとんどの入力の意味が描画「描く」であるためである．

3.1 実現

Sync/Draw は，図 3に示すように，入力処理部，解釈処理部，出力処理部から構成されている．入力処理部は音声認識[4]，語彙解釈を行い，入力に対して書換え規則を対応づける．解釈処理部は，オブジェクト空間に対し，書換え規則の適用を行う．出力処理部は，オブジェクト空間の内容を画面に描画する．以下では，オブジェクト，書換え規則，描画処理，Sync/Draw の機能について述べる．

[4]音声認識は「Wizard of Oz 法」によるものである．

3.1.1 オブジェクト

入力の意味を図形情報に近い形で表すために，オブジェクトと呼ぶ構造を用いる．オブジェクトは定数か，もしくは属性 $attr$ とオブジェクトのリスト O_l の対 $attr{:}O_l$ である (例 `Point:[X:20,Y:30]`)[5]．O_l に含まれるオブジェクトを $attr{:}O_l$ の要素と呼ぶ．属性の集合上には要素に成り得ることを表す関係 (図 4[6]) が与えられている．この関係は図形と操作に関する知識構造を表している．Sync/Draw は，入力の意味としての図形や操作をオブジェクトとして表し，そのリストをオブジェクト空間と呼ぶ．操作に対する照応を可能にするため，オブジェクトは操作結果を表すのではなく，操作の履歴を表している．

3.1.2 書換え規則

意味構造は，語やポインティング情報の入力毎に更新される．更新は，それまでの入力に対する意味全てに対して行われるわけではなく，その一部を変更したり，新たな意味を追加させることとみなせる．Sync/Draw では，オブジェクト空間を入力ごとに遷移させるために，書換え規則を用いている．書換え規則の採用により，新たなオブジェクトの導入や，左辺にマッチングしたオブジェクトのみを右辺の構造に変更するといった，部分的な変化を単純な形式で表すことが可能になった．

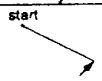
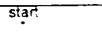
書換え規則の種類は新たなオブジェクトを導入する追加規則 ($\Rightarrow \beta$) と，すでに存在するオブジェクトを変更する変更規則 ($\alpha \Rightarrow \beta$) に分類できる．ポインティングに対する書換え規則は，ポインティングの座標に対応した追加規則を用いる．また，単語に対しては，各単語ごとに書換え規則が定められ，一般に，名詞に対しては追加規則を，助詞や動詞に対しては変更規則を用いる．これは，名詞は新たな概念を導入し，助詞はすでに存在する概念を格づけし，動詞は概念に対する操作を行うといった日本語の解釈に自然に対応している．

書換え規則は，適用の結果得られたオブジェクトが属性間の関係を満たす場合のみに適用される．追加規則の場合，新たなオブジェクト β をオブジェクト空間に追加し，他に存在するオブジェクトを β に含めることが可能ならば，それを β に挿入する．変更規則の場合，α にマッチングしたオブジェク

[5]リストの要素が一つのときはカッコを省略する．

[6]属性間の関係は巡回グラフで表されるが，ここでは簡単のため木構造で表している．

表 2: 例 1,4,5 の作図過程

入力	書換え規則 (ϕ は変数)	オブジェクト空間	画面出力
線	$\Rightarrow Line : [\,]$	$Line : []$	
を	$\phi \Rightarrow Object : [\phi]$	$Object : Line : []$	
$\nabla_{(65,38)}$	$\Rightarrow X : 65,\ \Rightarrow Y : 38$	$X : 65,\ Y : 38,\ Object : Line : []$	
ここ	$\Rightarrow Point : [\,]$	$Point : [X : 65, Y : 38]$ (以下では$Pt(65,38)$), $Object : Line : []$	
から	$\phi \Rightarrow Start : [\phi]$	$Object : Line : Start : Pt(65,38)$	
$\nabla_{(198,106)}$	$\Rightarrow X : 198,\ \Rightarrow Y : 106$	$X : 198, Y : 106$, $Object : Draw : Start : Pt(65,38)$	
ここ	$\Rightarrow Point : [\,]$	$Pt(198.106), Object : Line : Start : Pt(65,38)$	
まで	$\phi \Rightarrow End : [\phi]$	$Object : Line : [End : Pt(198,106), Start : Pt(65,38)]$	
描く	$Object : [\phi] \Rightarrow Draw : [\phi]$	$Draw : Line : [End : Pt(198,106), Start : Pt(65,38)]$ (以下では$Line(65,38) - (198,106)$)	
それ	$\phi \Rightarrow \phi$	$Line(65,38) - (198,106)$	
を	$\phi \Rightarrow Object : [\phi]$	$Object : Line(65,38) - (198,106)$	
消す	$Object : [\phi] \Rightarrow Delete : [\phi]$	$Delete : Line(65,38) - (198,106)$	
今の	$\phi \Rightarrow \phi$	$Delete : Line(65,38) - (198,106)$	
を	$\phi \Rightarrow Object : [\phi]$	$Object : Delete : Line(65,38) - (198,106)$	
取り消す	$Object : [\phi] \Rightarrow Undo : [\phi]$	$Undo : Delete : Line(65,38) - (198,106)$	

トを β に書換える．また，書換えた直後，変更されたオブジェクトが他のオブジェクトの下位階層に属する場合，それを上位のオブジェクトへ挿入する．

3.1.3 オブジェクト空間の描画

オブジェクト空間は，それまでの入力の意味，すなわち作図された図形，および作図途中の図形を表している．図形を画面に出力するため，各オブジェクトは具体的な座標を持つ線分や矩形といった図形として解釈される．始点のみを持つ線分や，中心のみを持つ円といった作図途中の図形を表すオブジェクトも，それぞれの意味を表す形で描画する．また，移動や削除等の操作の履歴を持つオブジェクトは，操作を順に行った結果の図形として描画する．

3.2 機能

Sync/Draw の機能としては，直線，矩形，円等の描画．図形の移動・消去の指示があり，これらは音声及びポインティングのみで行う．また，図形や操作への照応が可能である．従来の言語入力を持たない作図システムでは，作図の手順として，作図モードを選んでから図形を描くのが一般的である．これに対して Sync/Draw は，日本語の語順の自由度を利用することにより，以下に示す 2 通りの指示手順が可能である．

1. 書くものを決めてから位置を指示する．
   例 1: 線 を ∇ ここ から ∇ ここ まで 描く
2. 位置を指示してから何を書くかを指示する．
   例 2: ∇ ここ から ∇ ここ まで 線 を 描く

また，すでに描画した図形や行った操作に対し，以下に示す 2 種類の照応が可能である．

1. ポインティング及び言語による照応
   例 3：∇ これ を ∇ ここ まで 動かす
   (図形に対する照応)
2. 言語のみによる照応
   例 4：それ を 消す (図形に対する照応)
   例 5：今の (操作) を 取り消す (操作に対する照応)

発話同時理解により，Sync/Draw は入力に対し細やかな反応を返す．例 1,4,5 を順に行った作図過程を表 2に示す．画面出力が単語入力毎に更新される

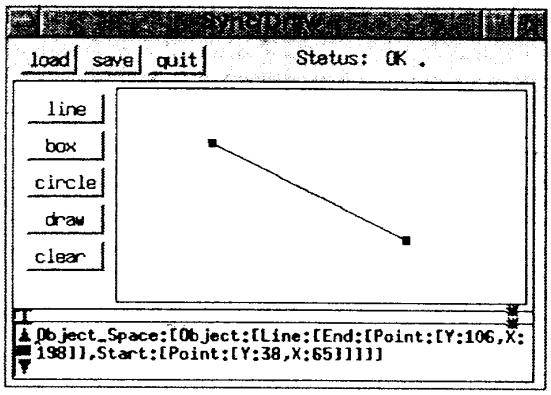


図 5: 例 1 において「まで」の入力後の状態

点に注目されたい．例 1 においては，「から」の入力後には，線を描画することがわかるため，線分が始点から描画される．また，例 4,5 では図形や操作に対する照応が行われている[7]．また図 5に例 1 に対する Sync/Draw の実行中の画面を示す．

4 おわりに

発話同時理解の枠組を提案し，その有効性を述べ，日本語を用いた場合の問題点について述べた．また，プロトタイプシステム Sync/Draw では，統一的な意味構造を採用し，単純な書換え規則を用いることにより，語やポインティング情報の入力毎の解釈処理，および同時進行的な描画処理を可能にした．Sync/Draw は，統一的な意味構造を持ち，単純な書換え規則を用い，語やポインティング情報の入力毎の解釈処理により，入力に対して細やかな反応を返すシステムである．

Sync/Draw により明らかになった問題点としては，構文解析を基本的に行っていないため，意味の無い入力があった場合，意味がとれないようなオブジェクトが生成されてしまう点が挙げられる．例えば「線を描く」といった入力に対して生成されるオブジェクトは`Draw:Line:[]` となり，図形として解釈できない．また，書換え規則は名詞，動詞，助詞などには対応できたが，「この」「赤い」などの連体詞に対応できていない．今後の課題としては，2.3 節における第 3,4 の問題の解決，意味の無い入力のキャンセルや，頑健性の向上，意味の曖昧性への対応，複文や図形の知識を必要とするようなより高度な指示に対する処理を可能にすることが挙げられる．

[7] オブジェクトの挿入や書換えは，オブジェクト空間のリストの先頭から行われるため，言語による照応はもっとも上位にあるものを対象とし，ポインティングによる照応は，その点に近いオブジェクトをリストの上位に移して処理している．

参考文献

[1] 安藤ハル，北原義典，畑岡信夫: インテリアデザイン支援システムを対象としたマルチモーダルインタフェースの評価，信学論，Vol. J77-D-II, No. 8(1994), pp. 1465–1474.

[2] Bolt, R. A.: Put-that-there: Voice and Gesture at the Graphics Interface, *ACM Computer Graphics*, Vol. 14, No. 3(1980), pp. 262–270.

[3] Inagaki, Y. and Matsubara, S.: Models for Incremental Interpretation of Natural Language, NLP95(1995), pp. 51–60.

[4] 西本卓也，志田修利，小林哲則，白井克彦: マルチモーダル入力環境下における音声の協調的利用 – 音声作図システム S-tgif の設計と評価 –，信学論，Vol. J79-D-II, No. 12(1996), pp. 2176–2183.

[5] 竹林洋一: 音声対話システム TOSBURGII – ユーザ中心のマルチモーダルインタフェースの実現に向けて –，信学論，Vol. J77-D-II, No. 8(1994), pp. 1417–1428.

[6] 山本博之，松原茂樹，河口信夫，稲垣康善: 自然言語の漸進的解釈に基づくマルチモーダル対話システム - 図形エディタ Sync/Draw-，インタラクティブシステムとソフトウェア IV，近代科学社 (1996), pp. 121–130.

[7] Shigeki Matsubara, Hiroyuki Yamamoto, Nobuo Kawaguchi, Yasuyoshi Inagaki, Katsuhiko Toyama, An Interactive Multimodal Drawing System based on Incremental Interpretation, IJCAI97 Workshop: Intelligent Multimodal Systems(1997), pp. 55–62.

[8] Farzad Ehsani, Kaichiro Hatazaki, Jun Noguchi and Takao Watanabe: Interactive Speech Dialogue System Using Simultaneous Understanding, ICSLP94(1994), pp. 879–882.

[9] 畑崎香一郎：発話同時理解に基づく音声対話，信学会大会講論集 (1996), pp. 335-336.

[10] 甲斐充彦，中川聖一：冗長語・言い直し等を含む発話のための未知語処理を用いた音声認識システムの比較評価，信学論 (D-II)，Vol.J80-D-II, No.10 (1997), pp. 2615–2625.

[11] 河原達也，北岡教英，堂下修司：A探索に基づいたフレーズスポッティングによる頑健な音声理解，信学論 (D-II)，Vol.J79-D-II, No.7(1996), pp. 1187–1194.