

情報検索対話におけるユーザ発話の検索要求単位への分割

徳久良子[†] 寺寫立太[†]
脇田敏裕[†] 乾健太郎^{††}

従来の情報検索対話システムの多くは、システム応答のタイミングが適切でないためにシステムがユーザの検索要求を正確に理解できないという問題があった。そこで我々は、ユーザからシステムにイニシアチブが移行するタイミングに注目し、検索要求単位タグつきコーパスを作成した。作成したコーパスに基づいて機械学習によりユーザ発話を検索要求単位にチャンキングしたところ、形態素情報のみを用いた場合でも 81.1% が正しく判定できることが分かった。また、音声対話システムへの応用に向けて音声認識精度と検索要求単位判定精度との関係を調べたところ、70% 以上の音声認識率があれば認識誤りが全くない場合とほぼ同等の精度で検索要求単位を判定できることが分かった。

User Utterance Segmentation for Information-Seeking Dialogue

TOKUHISA Ryoko,[†] TERASHIMA Ryuta,[†]
WAKITA Toshihiro[†] and INUI Kentaro^{††}

One of the old but unsolved issues in designing mixed-initiative spoken dialogue systems is how to decide the timing of system utterances. This paper addresses the issue restricting the target domain to information-seeking dialogues. We first claim that it is critically important for a system to decide when it should take the dialogue initiative, and define the notion of *information-requesting utterance segments (IRUSs)*, which gives useful clues for turn/initiative taking. We then report the present results of our experiments in which we took a machine learning-based approach achieving 81.1% accuracy in segmenting user utterances into IRUSs. We also examined how speech recognition errors influenced the performance of our model, finding that it worked tolerantly to at most 30% word recognition errors.

1. はじめに

音声対話システムの研究では、システムがどのタイミングで応答すべきかという問題が古くから議論されてきた [3, 4, 20]。対話システムが適切なタイミングで応答できないと、ユーザが発話意図を伝え終わる前にシステムが応答することがしばしば起こる。タスク指向型対話ではこれが原因でタスクが適切に達成されない場合がある。したがって、システムの適切な応答タイミングを判定することは、単に発話のやりとりの自然性向上だけでなく、タスク達成率向上という観点からも重視すべき課題と言える [8]。

システム応答のタイミングといっても、初めからあらゆる領域の対話を対象として論じるのは容易でない。そこで我々は、タスク指向型対話の中でも情報検索対話に限定して、適切な応答タイミングとは何かを考える。情報検索対話に限定するとはいえ、QA システムや乗り継ぎ案内システムなど応用アプリケーションは広い。したがって、情報検索対話におけるシステム応答の適切なタイミングについて論じることは有益だと考える。以上より、本稿では、情報検索対話においてシステムがどのタイミングで応答するのが適切かを明確にすると共に、応答タイミングを機械的に判定する手法につい

て議論する。

以下、次章ではシステム応答のタイミングについて論じ、3章で作成したコーパスについて説明する。さらに4章では応答のタイミングを判定するためのアルゴリズムを提案し、5章で評価実験の結果を述べる。最後に6章でまとめと今後の課題を述べる。

2. システム応答のタイミングとは

2.1 情報検索対話の構造とシステム応答の種類

情報検索対話のようなタスク指向型対話では、システム応答のタイミングが適切でないことが原因でタスクが達成されない場合がある [8]。システム応答の中で、発話タイミングがタスク達成に特に影響を与えるのはどのような応答かを明らかにするために、情報検索対話の構造を整理する。

以下に情報検索対話の例を示す。対話例中のUSRはユーザ発話を、SYSはシステム発話を表す。また、改行は一定長 (500msec) 以上の無音区間を表し、下線部はユーザとシステムが同時に発話したことを表す。

【対話例 1】

USR 1: えーっと 駅の近くで～
SYS 2: ええ
USR 3: デザートも～出して くれるような
SYS 4: デザート はい
USR 5: イタリア料理の 店って あるかな
SYS 6: 東京駅周辺で よろしいですか
USR 7: 東京駅で はい
SYS 8: はい 東京駅周辺の デザートが おいしい イタリア料理の 店を 検索します
USR 9: うん
SYS 10: 該当する店は 8 件見つかりました

[†] (株) 豊田中央研究所
Toyota Central R&D Labs., INC.
Email: {tokuhisa, ryuta, wakita}@mosk.tytlabs.co.jp

^{††} 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
Email: inui@is.aist-nara.ac.jp

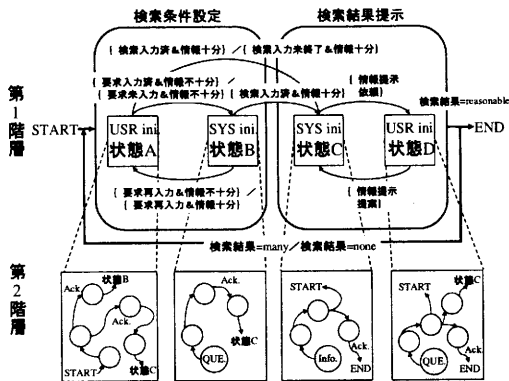


図1 情報検索対話の構造

USR 11: ランチがあるところがいいんだけど
 SYS 12: はい 1件 レストラン イタリアノが 見つかりました
【対話例 2】
 USR 13: えーっと
 USR 14: デザートも〜 出してくれるような
 SYS 15: デザートで 検索します 喫茶店が 3件 見つかりました
 USR 16: イタリア料理… (まだ話している途中なのに…)
【対話例 3】
 USR 17: うんと じゃ そーね
 USR 18: えっとー
 SYS 19: 食べたいものや 予算の ご希望は ございますか
 USR 20: あ じゃ そーね 餃子が 食べたいんだけど
 SYS 21: はい では 餃子という キーワードで 検索します

上記のような情報検索対話を談話セグメント [19] で分割すると、(a) ユーザが検索条件を設定する対話と、(b) システムが検索結果を提示する対話のふたつに分けられる。また、各セグメントはユーザとシステムのどちらがイニシアチブを持つかによってさらに細分できる。ここで、イニシアチブとは対話の主導権をどちらの話者が持つかを表すものである [1, 14]。イニシアチブには複数の捉え方がある [2] が、ここでは、単純に発話者がイニシアチブを持つと考えるのではなく、対話の中で局所的な話題を決定した話者がイニシアチブを持つと考える。対話例 1 では USR 1 から USR 5 が、検索条件を設定する対話でイニシアチブがユーザにある状態 (図 1 状態 A)、SYS 6 から USR 7 が 検索条件を設定する対話でイニシアチブがシステムにある状態 (状態 B)、SYS 8 から SYS 10 が 検索結果を提示する対話でイニシアチブがシステムにある状態 (状態 C) である。

ここで、システム応答には、イニシアチブの移行を伴わない発話とイニシアチブの移行を伴う発話があることに注意したい。イニシアチブの移行を伴わない発話とは、グラウンディングの過程で行われる相槌や確認発話を指す (e.g. SYS 2, SYS 4)。これは、図 1 の状態 A ~ 状態 D の全ての対話状態において発話されるもので、第 2 階層の同状態における推移はこの発話のやり取りを指す。またグラウンディングとは、複数の話者が共有知識を形成することを言う。グラウンディングのための応答のタイミングについてはこれまでに複数の報告がある [13, 17, 21]。一方、イニシアチブの移行を伴う発話は対話状態が移行する際のシステム応答で、例えば、状態 A (USR 1 ~ USR 5) から状態 B (SYS 6 ~ USR 7) に移行する際の 発話 SYS 6 がこれに相当する。

さて、以上で述べた対話構造を踏まえて、応答タイミングの相違がタスク達成に特に影響を与えるのはどのようなシステム応答かを議論する。情報検索対話において、タスクの

表 1 システムの応答を決定する要素

		(b) システムが検索に十分な情報を得たか	
		情報十分	情報不十分
(a) ユーザが発話を終えたか	発話終了	応答 (検索結果の提示)	応答 (追加情報の要求)
	発話未終了	応答 (割込→検索結果の提示)	不応答 (発話終了を待つ)

達成に必要な要件のひとつは、ユーザの検索要求を適切に理解することである。仮に、ユーザが「デザートが食べられるイタリア料理の店が知りたい」という要求を持っていたとする。この時、対話例 2 のようにユーザが検索要求を述べる途中であるにも関わらずシステムが応答すると、ユーザの検索要求を十分に反映した検索が行えない。このような検索要求の認識誤りを避けるためには、システムはユーザがどの時点で検索要求を言い終えたかを的確に判断して応答する必要がある。また、対話例 3 のように、ユーザがどのように検索要求を述べて良いか分からない場合には、システムはユーザが検索要求を言えずに困っている状態を的確に判断して検索要求発話を促すような応答をする必要がある。ユーザが検索要求を言い終えたり言えずに困った時点で発話されるシステム応答とは、前述のイニシアチブの移行を伴う発話に相当する。そこで本稿では、ユーザからシステムにイニシアチブが移行するタイミングについて議論する。

以下本稿では、『イニシアチブの移行を伴うシステム応答』を、単にシステム応答と表現する。また、『ユーザが発話を終える (検索条件を言い終える/言えずに困る) までのユーザ発話のまとまり』のことを 検索要求単位 (information-requesting utterance segments; IRUS) と呼ぶ。ここで検索要求単位の終端はシステム応答のタイミングと一致する。例えば対話例 1 では「えーっと 駅の近くで〜 デザートとかも〜 出してくれるような イタリア料理の 店ってあるかな」が検索要求単位で、その直後にシステムが応答するのが望ましい。検索要求単位の定義の詳細は 3 章で述べる。

2.2 イニシアチブ移行のタイミング

我々は、情報検索対話におけるシステム応答のタイミングを決定する要因には、以下の 2 つがあると考えられる。

- (a) ユーザが発話を終えたかどうか
- (b) システムが検索に十分な情報を得たかどうか

(a) は、ユーザがシステムにイニシアチブを渡したいと考えるかどうかを表す。ユーザが発話を終えるというのは、具体的には、ユーザが検索要求を言い終えたり (USR 1 ~ USR 5)、検索要求を言えずに困っている (USR 17 ~ USR 18) 状態を指す。一方 (b) は、システムがユーザからイニシアチブを奪うべきと判断するかどうかを表す。システムが検索に十分な情報を得たというのは、システムが持つ検索スロットが全て埋まったり、検索結果が十分絞り込まれた状態を指す。表 1 に、(a) と (b)、および我々が適切と考えるシステム応答との関係を示す。

堂坂らは、確認コストと情報伝達コストという 2 軸を評価軸としてシステム応答のタイミングを決定するデュアルコスト法を提案した [20]。これは、上記の (b) を用いて応答タイミングを決定するものである。彼らのシステムは、発話内容がうまく認識できなかった際に内容を確認したり、ユーザが検索要求を発話中でも検索結果が見つからなかったり一

表 2 収集したコーバスの規模

被験者	ユーザ役の話者数	100 名
	システム役の話者数	1 名
タスク	1 人あたりのタスク数	2 タスク
	タスクの内容	お好みの居酒屋 サラリーマンで賑わう居酒屋
規模	対話数合計	200 対話
	ユーザ話者平均ターン数	12.9 ターン
	システム話者平均ターン数	12.6 ターン

意に決まった場合に検索結果を提示するための応答をする。この手法は、ユーザが発話中でもシステムが割り込むことで結果的に少ない発話のやりとりでタスクが達成されるという点で、表 1 の「発話未終了-情報十分」で有効な手法と言える。しかし、ユーザの状態を考慮していないため、表 1 の「発話終了-情報不十分」では応答することができないという問題がある。

情報検索対話システムを利用する際、ユーザがシステムの持つスロットの数やデータベースの内容をあらかじめ知ることは難しい。したがって、ユーザが常にシステムにとって十分な検索条件を述べることは期待できない。そのため情報検索対話システムにとって、システム状態だけでなくユーザ状態を考慮しながら対話を進めることは必要不可欠である。しかし、ユーザが発話を終えたかどうかというユーザ状態を機械的に推定すること自身が可能かどうか、また、可能であるとするどのような特徴量が判別に有効かは自明でない。そこで以下では、ユーザが発話を終えた（検索要求を言い終えた／言えずに困った）かどうかを機械的に判定する手法について述べる。

3. 検索要求単位タグつき対話コーパス

本章では、検索要求単位の定義の詳細と、検索要求単位タグつき対話コーパスについて説明をする。

3.1 コーパス収集

検索要求単位タグつき対話コーパス作成のため、システム役の話者 1 名と、ユーザ役の一般被験者 100 名を用いて、情報検索対話を収集した。以下、便宜的にシステム役の話者を《システム》、ユーザ役の話者を《ユーザ》と呼ぶ。

対話は《システム》と《ユーザ》の 2 名が対面して行う。対話収録の際、《ユーザ》にはあらかじめ飲食店を検索するタスクを与えた。まず、《ユーザ》は検索条件を自由に考えて《システム》に伝える。次に、《システム》は《ユーザ》の検索要求に基づいてデータベースから店を検索する。《ユーザ》が満足する店が 1 件に絞られた時点で対話は終了する。なお、《システム》には、対話をしながら検索タスクを遂行できるようにあらかじめ訓練した話者を用いた。したがって、本コーパスは検索キーワードの入力など対話以外のタスクが対話タスクに与える影響を極力少なくした上で収録されている。収集した対話コーパスの規模を表 2 に示す。表中の平均ターン数は、500 msec 以上の無音区間で区切られた発話を 1 ターンと見なした場合のターン数の平均を表す。

3.2 検索要求単位タグの定義

検索要求単位 (IRUS) とは『ユーザが発話を終える (検索要求を言い終える／言えずに困る) までのユーザ発話のまとまり』を言う。検索要求単位の終端は、《ユーザ》から《システム》にイニシアチブが移行するタイミングと一致する。

ここで、対話コーパスにおけるイニシアチブ移行のタイミングについて整理する。コーパスから観察されるイニシアチ

ブ移行のタイミングは、厳密には『《ユーザ》がイニシアチブを渡したいと考えたタイミング』ではなく『《システム》が『《ユーザ》はイニシアチブを渡したいだろう』と考えたタイミング』である。言い換えると、対話コーパスにおいて『《ユーザ》から《システム》にイニシアチブが移行した点は、『《ユーザ》自身が真に発話を終えた点』ではなく、あくまで『《システム》が『《ユーザ》は発話を終えただろう』と判断した点』である。つまり、真の検索要求単位 (IRUS) である『《ユーザ》自身が真に発話を終えた点』は、対話コーパスの発話のやりとりからは客観的に特定することができない。

そこで今回は、検索要求単位 (IRUS) を『《ユーザ》の考える真の検索要求単位』ではなく、『《システム》が判断した検索要求単位』で近似することにした。以下これを、*S-IRUS* と略記する。*S-IRUS* は、対話コーパスにおける《ユーザ》と《システム》との発話のやりとりを観察することによって付与できる。また、対話中に《ユーザ》が発話を終えたかどうかに関して《システム》が常に最良の判断ができていないとは限らない。そこで『ラベラーが後から対話を見直した場合に判断できる検索要求単位』に対してもタグを付与することにした。これを以下、*L-IRUS* と略記する。

3.3 検索要求単位タグつき対話コーパスの作成

本節では、*S-IRUS* と *L-IRUS* のタグ付与基準について述べる。

【S-IRUS のタグ付与基準】

S-1: 《システム》が判断した検索要求単位に対して *S-IRUS* タグを付与する。

S-2: 《システム》の応答に準じてタグを付与する。

S-3: 《ユーザ》と《システム》の間で検索要求単位の認識が異なる場合でも《システム》の認識を優先する。

【L-IRUS のタグ付与基準】

L-1: ラベラーが、収録されたユーザ発話を聞いた際に判断する検索要求単位に対して *L-IRUS* タグを付与する。

L-2: 《システム》の応答とは独立にタグを付与する。

L-3: 《ユーザ》の発話を最後まで聞き、ラベラーが最良と思う箇所タグを付与する。タグつけの際、同じ箇所を何度聞いても構わない。

以下に、検索要求単位タグつきコーパスの具体例を示す。また、この例を用いて、タグ付与基準で特に注意すべき点を説明する。

【対話例 4】

USR 22: <S-IRUS ID=008><L-IRUS ID=009 >えー居酒屋で～
 USR 23: そうか なるべく～
 USR 24: 願いでもいいような所 </L-IRUS ID=009 >
 </S-IRUS ID=008>
 SYS 25: はいでは
 USR 26: <S-IRUS ID=010 ><L-IRUS ID=011 >静かじゃ
 ないってことね</L-IRUS ID=011 >
 </S-IRUS ID=010 >
 SYS 27: あ はいでは 静かではない 居酒屋 ということで
 えー 2 件 見つかりました

【対話例 5】

USR 28: <S-IRUS ID=012 ><L-IRUS ID=013 >予算が
 えーっと
 USR 29: 3000 円ぐらいがいいです </L-IRUS ID=013 >
 SYS 30: はい
 USR 31: <L-IRUS ID=014 >あと女性の 多い 店

表 3 S-IRUS と L-IRUS の一致率

	L-IRUS タグあり	L-IRUS タグなし
S-IRUS タグあり	614 (98.2%)	0 (0%)
S-IRUS タグなし	11 (1.8%)	—

</L-IRUS ID=014 ></S-IRUS ID=013 >
SYS 32: はいでは予算 3000 円 キーワード 女性で 3 件
見つかりました

S-IRUS のタグ付与基準 S-3 について

上述の対話例 4 は、USR 24 の直後で《ユーザ》と《システム》の検索要求単位の判断に不一致が起きた例である。USR 24 の直後で《システム》は『《ユーザ》は検索要求を言い終えた』と判断し SYS 25 を発話した。しかし同時に、《ユーザ》は USR 26 で検索要求の続きを発話した。つまり、《システム》は USR 24 の直後を検索要求単位の終端と判断したが、実は USR 24 の時点では《ユーザ》は検索要求を言い終えていなかった。このように、明らかに《システム》の認識した検索要求単位が《ユーザ》自身が考える真の検索要求単位と食い違う場合でも、<S-IRUS ID=008>が示すように、対話進行中に《システム》が検索要求単位と認識した箇所には S-IRUS タグを付与する。

L-IRUS のタグ付与基準 L-2 について

対話例 5 は USR 29 の直後でラベラーと《システム》の検索要求単位の判断に不一致が起きた例を示している。このように、《システム》が検索要求単位と判断しなかった箇所でもラベラーが検索要求単位に成り得ると判断した箇所には、L-IRUS タグを付与する。

3.4 検索要求単位タグ付き対話コーパスの分析

収集した対話コーパス 200 対話に対して、訓練されたラベラー 1 名で検索要求単位タグを付与した。その結果、S-IRUS タグが 625 個、L-IRUS タグが 614 個付与された。表 3 に 2 種類のタグの一致率を示す。

表 3 が示す通り、S-IRUS タグが付与された箇所にはすべて L-IRUS タグが付与された。これは、対話中に《システム》がイニシアチブが移行する点だと判断した箇所は、ラベラーが後から対話を複数回聞いた場合でもイニシアチブの移行が可能だと判断できることを意味する。また、S-IRUS タグは付与されていないが、L-IRUS タグが付与された箇所が 11 箇所 (1.8%) あった。これは、対話例 5 <L-IRUS ID=013> のように、《システム》は応答しなかったがラベラーはイニシアチブの移行が可能だと判断した箇所の個数を示す。この割合が全体の 2% 未満であったという結果は、検索キーワード入力などの検索タスクを遂行しながらの対話であったにも関わらず、《システム》は十分に高い精度で応答タイミングを判断し、実際に応答できていたことを示している。以上より、今回収集した対話コーパスにおける《システム》応答のタイミングは十分に安定したものであり、応答タイミングの学習のための訓練データとして使用に耐えるものであると考えてよい。

なお、今回は、1 名のラベラーでタグづけ作業を行ったため、S-IRUS タグ および L-IRUS タグ自身の信頼性については評価できなかった。タグの評価は今後の課題である。

4. 検索要求単位判定手法

4.1 検索要求単位判定アルゴリズム

図 2 に想定する音声対話システムの構成を示す。まず、検

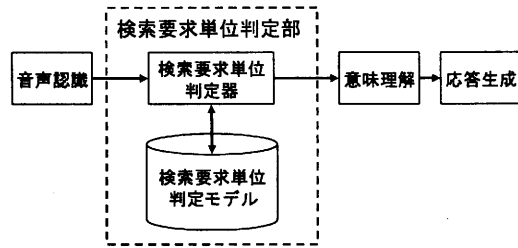


図 2 検索要求単位判定部

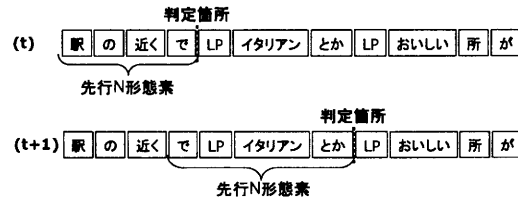


図 3 提案手法の判定アルゴリズム

索要求単位判定器には、音声認識部から音声認識結果が入力される。次に、単位判定器は認識文字列を検索要求単位判定モデルと照合して検索要求単位にチャンクする。最後に、検索要求単位にチャンキングした単語列を意味理解部に出力する。

発話やテキストのセグメントする手法については、これまでにいくつか報告がある [7, 10, 18]。竹沢らは、翻訳処理に適切な単位に発話文を分割することを目的として、分割候補の前後数個の形態素を用いてある形態素間が翻訳処理単位の境界かどうかを判定する手法を提案した。また小倉らは、タスク指向型の対話を対象に、接続詞などの局所的な手がかりと単語分布などの大局的な手がかりを用いて対話のセグメント境界を判定する手法を提案している。

検索要求単位の判定を音声対話システムに応用するためには、検索要求単位の判定結果を逐次的に獲得できる枠組みを用意することが望ましい。しかし、上記で紹介した研究はいずれも、逐次性という観点で十分でなかった。そこで我々は、局所的な発話列の情報を用いて漸進的に検索要求単位を判定する手法を開発した。

図 3 を用いて、提案手法のアルゴリズムを説明する。ここでは、一定以上のポーズごとに音声認識結果が検索要求単位判定器に入力されると仮定する。検索要求単位判定部は、認識単位ごとに認識文字列の終端が検索要求単位かどうかを判定する。判別には、判定箇所に先行する N 形態素の言語情報を用いる。例えば、図 3 の時刻 (t) において、形態素「で」の直後が検索要求単位境界であるかどうかを判定する場合には、「で」に先行する任意個の形態素からなる単語列の言語情報を用いる。図 3 は、先行する 4 個の形態素を用いた場合を示している。さらに、時刻 (t+1) では、その時刻に音声認識された単位を入力として同様に判定を行う。この時、一定時間以上の無音区間 (図 3 中 LP) は 1 形態素と見なし、他の形態素と同等に取り扱う。

以上で述べた通り、我々の提案手法は、音声認識部から認識単語列を受け取りながら逐次的に検索要求単位が判定することが可能である。そのため、音声対話システムのように逐次的に判定結果を要求されるシステムへの応用に向く。

表 4 判別に用いる特徴量

ID	特徴概要	特徴量	説明
1	形態素	見出し語	発話された語
2		品詞, 品詞細分類	
3		活用型	
4		活用形	
5	文節	文節区切りかどうか	文節の終端で { ある/ない }
6	係り受け	直後に係るかどうか	直後の形態素に { 係る/係らない }
7		係り元の個数	先行文節の何個が係り受けるか

4.2 検索要求単位判定のための特徴量

発話のセグメンテーションの研究では、形態素情報のような言語情報、韻律や視線情報のようなパラ言語情報、検索スロット情報のような対話状態などの特徴量を使う手法が提案されてきた [5, 8, 10, 18]。いわゆる言語情報だけでは曖昧性があるため、パラ言語情報などを複合的に用いて検索要求単位を判定することが有効であることは容易に想像される。しかし、今回は予備的に、言語情報のみを特徴量として検索要求単位を判定した。

表 4 に、判定に用いる特徴量の一覧を示す。なお、文節終端にない形態素の場合には、係り受け情報「直後に係るかどうか」は常に「直後に係る」と見なした。例えば、図 3 「で」の特徴量は、特徴量 ID 順に { で, 格助詞一般, *, *, 文節の終端である, 直後に係らない, 1 個 } となる。

5. 検索要求単位判定評価実験

5.1 実験の目的と方法

我々の提案手法が検索要求単位の判定に有効かどうかを調べるために評価実験を行った。まず、提案手法との比較のために、以下の 2 種類のシステムを用意した。

比較システム 1 (LP 検出システム): 500msec 以上の無音区間がある場合は全て検索要求単位境界と判定する。

比較システム 2 (文末表現抽出システム): 入力文字列の終端が文末表現であれば検索要求単位境界と判定する。文末表現には終助詞/助動詞 (特殊・ダ, 特殊・デス, 特殊・マス) を採用した [15]。

また、我々の手法において表 4 のどの特徴量が有効かを明らかにするために、以下の 5 通りの特徴量の組み合わせを用意して実験を行った。

- (1) 形態素情報のみ
- (2) 形態素と文節情報 (人手で作成した正解を利用)
- (3) 形態素と文節情報 (解析器の出力を利用)
- (4) 形態素, 文節 (人手で作成した正解), 係り受け情報 (人手で作成した正解)
- (5) 形態素, 文節 (解析器の出力を利用), 係り受け情報 (解析器の出力を利用)

ここで、形態素・文節・係り受けの正解は人手で作成した。タグの修正には、東京大学で開発されたツールを用いた [16]。また、自動的な文節の解析には YamCha [11] を、係り受け解析には CaboCha を用いた [12]。なお、漸進的な処理を実現するために CaboCha を一部改造して用いた。検索要求単位の判定には SVM (Support Vector Machine) を用いる。SVM の学習と判別には Tiny SVM を利用した^{††}。

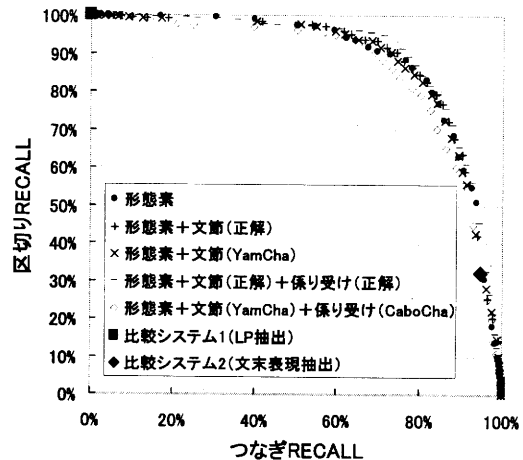


図 4 L-IRUS の判定精度

3章で作成した検索要求単位タグつきコーパス 200 対話を用いて実験を行う。3.4 節で述べた通り、S-IRUS と L-IRUS という 2 種類の検索要求単位は一致率が高い。しかし、S-IRUS には検索タスク遂行などの影響によってノイズが含まれる可能性がある。そこで、実験には L-IRUS タグを利用した。なお今回は、《ユーザ》が検索条件を言えずに困ってイニシアチブが移行した 39 事例 (対話例 3 USR 17 ~ USR 18) はコーパスから除き、ユーザが検索要求を言い終えたか言い終えていないかを判定可能か調べた。

5.2 実験結果

200 対話を 6 分割してクロスバリデーションテストを行った。500msec 以上の無音区間がある箇所までをひとつの入力単位とし、無音区間の直前が検索要求単位の境界となるかどうかを判定した。L-IRUS の判定精度を図 4 に示す。

図 4 の横軸は検索要求単位境界でない箇所の再現率を、縦軸は検索要求単位境界である箇所の再現率を示す。以下、検索要求単位の境界でない箇所のことをつなぎ、検索要求単位境界である箇所のことを区切りと表現する。ここで、つなぎの再現率は「ユーザが検索要求を述べている途中で応答を返すことなく検索要求を聞けるか」を、区切りの再現率は「ユーザが検索要求を言い終えた箇所適切に応答が返せるか」を評価していることに相当する。

グラフから読み取れる通り、比較システム 1 は区切りの再現率が高いがつなぎの再現率が低い。一方、比較システム 2 はつなぎの再現率が高いが区切りの再現率が低い。それに比べて我々の手法では、つなぎと区切りの再現率が共に 80% 程度と高い精度で検索要求単位を判定することができた。このことから、提案手法は比較システムよりも正確に検索要求単位を判定することが可能だと言える。

次に、提案手法においてどの特徴量が判別に有効に働いたかを考察する。実験では、人手で作成した形態素・文節・係り受けの正解を特徴量として利用するのが最も精度が良く、解析器によって求めた形態素・文節・係り受けを特徴量とする場合が最も精度が悪かった。このことから、係り受け情報は検索要求単位の判定に有効に働くものの、現状の係り受け解析の精度では解析誤りが逆に検索要求単位判定精度を下げてしまうことが分かる。また、形態素のみを用いる場合と形態素・文節を用いる場合には精度に差が見られなかった。判

^{††} YamCha, CaboCha, Tiny SVM は、工藤氏の HP よりダウンロードできる。 <http://cl.aist-nara.ac.jp/iaku-ku/>

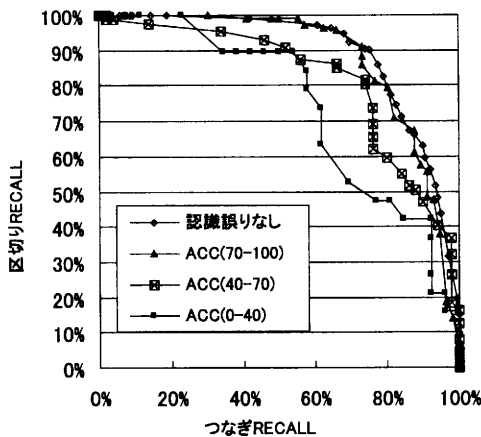


図5 検索要求単位判定精度と音声認識精度の関係

定精度や処理時間を考慮すると、検索要求単位の判定を音声対話システムに適用する場合には、形態素情報のみを特徴量とするのが良いと言える。

5.3 音声認識精度と検索要求単位判定精度の関係

検索要求単位判定を音声対話システムに適用する場合、音声認識誤りの問題を無視することはできない。そこで予備調査として、検索要求単位の判定精度と音声認識精度との関係を調べた。音声認識器にはコロラド大学で開発された sonic を用いた [9]。また、検索要求単位判定の特徴量には形態素情報のみを用いた。

図5に、話者別の認識精度と検索要求単位判定精度の関係を示す。図5から、音声認識精度が低下すると検索要求単位判定精度も下がるものの、音声認識率が70%以上あれば、理想的な音声認識が実現できた場合とほとんど変わらない精度で検索要求単位を判定できることが読み取れる。70%という音声認識精度は、現状の音声認識技術では無理のない数値である。したがって、一般的な音声認識装置を用いても十分な精度で検索要求単位の判定が可能であると考えられる。

6. まとめと今後の課題

我々は、情報検索対話におけるシステム応答のタイミングを決定する要因には、ユーザの状態とシステムの状態という2つの要素があると考えた。本稿では、ユーザが発話を終えたというユーザ状態が機械的に判定可能かどうかを調べた。

まず、ユーザからシステムにイニシアチブが移行するタイミングに着目し、検索要求単位タグつきコーパスを作成した。作成した対話コーパス200対話に基づいて、機械学習により検索要求単位を判定した結果、約80%の再現率で検索要求単位が判定できた。また、音声認識精度と検索要求単位判定精度との関係を調べたところ、認識精度が70%以上あれば、認識誤りが全くない場合とほとんど変わらない精度で検索要求単位が判定できることが分かった。

本稿で報告した実験では、ユーザが検索要求を言えずに困ってイニシアチブが移行した事例はあらかじめテストセットから除いた。しかし、検索要求を言えずに困っている状態を推定することは、情報検索対話を円滑に進める上で必要な要件のひとつである。このような事例も含めてユーザ状態を判定することは今後の課題である。

また、対話システムを総合的に評価した場合 [6] に、システム応答のタイミングの決定にユーザ状態とシステム状態がそれぞれどの程度寄与するかは自明でない。今後、これを実験的に明らかにしていく必要がある。

さらに今回、話者別の認識精度を用いて音声認識精度と検索要求単位判定精度との関係を調べた。しかし、収集したコーパスの量が十分でないため、認識精度と検索要求単位判定精度との関係を精緻に調べることができなかった。対象とするシステムにとってどの程度の音声認識精度が要求されるかを明らかにするためには、コーパスの量に寄らず音声認識精度とシステムの精度との関係を正確に評価できる技術の開発を進める必要がある。

7. 謝 辞

東京大学の石崎雅人助教授、千葉大学の市川薫教授、伝康晴助教授、堀内靖雄助教授には、システム応答タイミングの定義に関して示唆に富む助言をいただきました。また、奈良先端科学技術大学院大学の工藤拓氏、東京大学の河原大輔氏には各種ツールの利用に関して多くの助言をいただきました。ここに深謝致します。

参 考 文 献

- [1] Chu-Carroll, J. and Brown, M. K.: Initiative in Collaborative Interactions - Its Cues and Effects, pp. 16-22 (1997).
- [2] Cohen, R., Allaby, C., Cumbaa, C., Fitzgerald, M., Ho, K., Hui, B., Latulipe, C., Lu, F., Moussa, N., Pooley, D., Qian, A. and Siddiqi, S.: What is Initiative?, *User Modeling and User-Adapted Interaction*, Vol. 8, No. 3-4, pp. 171-214 (1998).
- [3] Heisterkamp, P. and McGlashan, S.: Units of Dialogue Management: An Example, Vol. 1, pp. 200-203 (1996).
- [4] Jonsson, A.: Dialogue Actions for Natural Language Interfaces, pp. 1405-1413 (1995).
- [5] Jurafsky, D. and Martin, J. H.: *Speech and Language Processing*, Prentice Hall (2000).
- [6] Marilyn Walker, Diane Litman, C. K. and Abella, A.: PARADISE: A Framework for Evaluating Spoken Dialogue Agents, *ACL-97* (1997).
- [7] 竹沢寿幸, 森元暹: 発話単位の分割または接合による言語処理単位への変換手法, *自然言語処理*, Vol. 6, No. 2, pp. 83-95 (1999).
- [8] Nakano, M., Miyazaki, N., ichi Hirasawa, J., Dohsaka, K. and Kawabata, T.: Understanding Unsegmented User Utterances in Real-Time Spoken Dialogue Systems, *ACL-99*, pp. 200-207 (1999).
- [9] Pellom, B.: SONIC: The University of Colorado Continuous Speech Recognizer, *TR-CSLR-2001-01* (2001).
- [10] Sato, R., Higashinaka, R., Tamoto, M., Nakano, M. and Aikawa, K.: Learning Decision Trees to Determine Turn-Taking by Spoken Dialogue Systems., *ICSLP-2002*, pp. 861-864 (2002).
- [11] 工藤拓, 松本裕治: Support Vector Machineを用いた Chunk 同定, *自然言語処理*, Vol. 9, No. 5, pp. 3-21 (2002).
- [12] 工藤拓, 松本裕治: チャンキングの段階適用による係り受け解析, *情報処理学会論文誌*, Vol. 43, No. 6, pp. 1834-1842 (2002).
- [13] Traum, D. R. and Heeman, P. A.: Utterance Units And Grounding In Spoken Dialogue, *ICSLP-96* (1996).
- [14] Walker, M. A. and Whittaker, S.: Mixed Initiative in Dialogue: An Investigation into Discourse Segmentation, pp. 70-78 (1990).
- [15] 益岡隆志, 田窪行則: *基礎日本語文法 - 改定版 -*, くろしお出版 (1992).
- [16] 河原大輔, 黒橋慎夫, 橋田浩一: 「関係」タグ付きコーパスの作成, pp. 495-498 (2002).
- [17] 小嶋花絵, 野口広彰, 伝康晴, 福田泰子: 韻律情報に基づいたあいづち挿入箇所の推定, *言語処理学会 第4回年次大会 発表論文集*, pp. 484-487 (1998).
- [18] 小倉加奈代, 須藤由加, 松永政幸, 石崎雅人: 多レベルの知識を利用した課題遂行対話のためのセグメンテーション分割, *SIG-SLUD-A103-6*, pp. 33-38 (2002).
- [19] 人工知能学会談話・対話研究におけるコーパス利用研究グループ: 様々な応用研究に向けた談話タグ付き音声対話コーパス, *SIG-SLUD-9903-4*, pp. 19-24 (2000).
- [20] 堂坂浩二, 相川清明: 対話コスト最小化原理に基づく対話制御, *言語処理学会 第7回年次大会 発表論文集*, pp. 518-521 (2001).
- [21] 野口広彰, 片桐恭弘, 伝康晴: 尤度付きあいづち生起文脈コーパスの提案, *SIG-SLUD-A101-6*, pp. 25-32 (2001).