情報検索対話におけるイニシアチブ移行制御*

徳久良子 † 寺嶌立太 † 脇田敏裕 † 乾健太郎 ‡

† (株) 豊田中央研究所 ‡ 奈良先端科学技術大学院大学 情報科学研究科

† {tokuhisa,ryuta,wakita}@mosk.tytlabs.co.jp ‡ inui@is.aist-nara.ac.jp

1 はじめに

対話システムが適切なタイミングで応答できないため，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，単なる対話の自然性だけでなく，タスク達成率向上の観点からも重視すべき課題と言える [2]．

本稿では，タスク指向型対話の中でも特に情報検索対話に限定し，システムが応答すべきタイミングとは何かを議論する．また，その応答タイミングが機械的に計算可能かどうかを検証する．

2 イニシアチブ移行のタイミング

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，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
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 イタリアーノが 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… (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

上記の具体例を用いて，どのようなシステム応答がタスク達成と関連が深いかを整理する．

情報検索対話において，タスクを達成するの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．対話例 2 のように，ユーザが検索要求を言い終えていないにも関らずシステムが応答すると，検索要求発話が中断されてしまうため，ユーザの要求を完全に認識することができない．このような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必要がある．

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**（結果の提示）	**応答**（追加情報の要求）
	発話未終了	**応答**（割込→結果の提示）	**不応答**（発話終了を待つ）

ユーザが検索要求を言い終えた時点で発話されるシステム発話や言えずに困った時点で発話されるシステム発話は，いずれもユーザからシステムへのイニシアチブの移行を伴う発話である．これらの発話のタイミングは，ユーザの検索要求を適切に認識できるかどうかに直接影響する．一方，システム応答には，イニシアチブの移行を伴わない相槌や確認発話（e.g. SYS2, SYS4）もあるが，この種の発話は検索要求発話をさえぎらないため，検索要求の認識の妨げにはなりにくい．そこで本稿では，システム応答の中でもユーザからシステムへのイニシアチブの移行を伴う発話のタイミングについて議論する．

ユーザからシステムへのイニシアチブ移行を決定する要因には，表 1 の (a) と (b) の 2 つが考えられる．(a) は，ユーザがイニシアチブを渡したいかどうかを表す．具体的には，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た（ USR 13 〜 USR 14 ）状態を指す．また (b) は，システムがイニシアチブを奪うべきだと判断するかどうかを表す．具体的には，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．

堂坂らは，応答タイミングを決定する指標としてデュアルコスト法を提案した [7]．これは，表 1 の (b) を用いてユーザからシステムへのイニシアチブの移行タイミングを決定するものである．この手法によると，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

*Initiative Control for Information-Seeking Dialogue

表 2: 収集したコーパスの規模

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

ない発話のやりとりでユーザはタスクを達成できる．ただしこれは，表 1 の「発話未終了- 情報十分」のみに相当し，ユーザ状態を考慮していないため，表 1 の「発話終了- 情報不十分」では応答できない．しかし，情報検索対話システムを利用する際，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，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が必要不可欠である．

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）とユーザ自身が考えたかどうかを機械的に判定する手法について述べる．

以下本稿では,『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）**と表現する．

3 検索要求単位タグ

3.1 検索要求単位タグの定義

本節では，検索要求単位（IRUS）タグを定義する．

まず，対話例 1 〜 3 のような所与の対話データに検索要求単位の情報を注釈付けする作業を考えてみよう．対話データの中でユーザからシステムへのイニシアチブ移行が観察できる点は，SYS6 や SYS19 のように『ユーザ自身が「自分は発話を終えた」と考えた点』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，本来の IRUS である『ユーザ自身が真に発話を終えた点』は，対話の発話のやりとりからは客観的に観察できない．

そこで今回は，IRUS を『ユーザの考える IRUS 』ではなく,『システムが判断した IRUS 』で近似した．以下これを，*S-IRUS* と略記する．また後述するように，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をしているとは限らない．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 IRUS 』にもタグを付与した．これを，*L-IRUS* と略記する．

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，ユーザ役の話者 100 名を用いて，飲食店を検索する対話を収集した（表 2）．以降,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》が対面して行う．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．

【S-IRUS のタグ付与基準】

S-1：《システム》が判断した IRUS 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 IRUS 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 IRUS 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にタグを付与する．タグづけの際，同じ箇所を何度聞いても構わない．

【対話例 4】

USR 22: <S-IRUS ID=008><L-IRUS ID=009 >えー 居酒屋で〜 そうか で なるべく〜
USR 23: 騒いでも いいような 所 </L-IRUS ID=009 > </S-IRUS ID=008>
SYS 24: はい では
USR 25: <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 26: 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 27: <S-IRUS ID=012 ><L-IRUS ID=013 >予算が
USR 28: 3000 円ぐらいが いいです </L-IRUS ID=013 >
SYS 29: はい
USR 30: <L-IRUS ID=014 >あと女性の 多い 店 </L-IRUS ID=014 ></S-IRUS ID=013 >
SYS 31: 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

上記の例を用いて，タグ付与基準で特に注意すべき点を説明する．

S-IRUS のタグ付与基準 S-3 について

上述の対話例 4 は，USR 23 の直後で《ユーザ》と《システム》の IRUS の判断に不一致が起きた例である．USR 23 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 24 を発話した．しかし同時に,《ユーザ》は USR 25 で検索要求の続きを発話した．つまり,《システム》は USR 23 の直後を IRUS の終端と判断したが，実は USR 23 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 IRUS が《ユーザ》自身が考える IRUS 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 IRUS と認識した箇所に 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 28 の直後でラベラーと《システム》の IRUS の判断に不一致が起きた例を示す．このように,《システム》が IRUS と判断しなかった箇所でもラベラーが IRUS に成り得ると判断した箇所には，L-IRUS タグを付与する．

3.2 検索要求単位タグ付与の信頼性評価

収集した対話コーパス 200 対話のうち話者の年代と性別が均等になるように選んだ 120 対話に対して，訓練されたラベラー 2 名で検索要求単位タグを付与した．

表 3 に，2 名のタグの一致率を示す．小規模な調査ではあるものの，S-IRUS と L-IRUS 共に，ユーザが検索要求を言い終えた箇所での一致率は高かった．このことから，検索要求を終えた箇所でのイニシアチブ移行は，ユーザ発話から得られる何らかの特徴に基づいた判定が可能と言える．一方，検索要求を言えずに

表 3: 再現可能性調査結果

		先頭と終端が一致	終端が一致	不一致	計
S-IRUS	(1) 検索要求を言い終えた	328 (94.8%)	333 (96.2%)	13 (3.8%)	346
	(2) 検索要求を言えずに困った	30 (76.9%)	33 (84.6%)	6 (15.4%)	39
	(1) と (2) の合計	358 (93.0%)	366 (95.1%)	19 (4.9%)	385
L-IRUS	(1) 検索要求を言い終えた	324 (92.6%)	334 (95.4%)	16 (4.6%)	350
	(2) 検索要求を言えずに困った	26 (57.8%)	30 (66.7%)	15 (33.3%)	45
	(1) と (2) の合計	350 (88.6%)	364 (92.2%)	31 (7.8%)	395

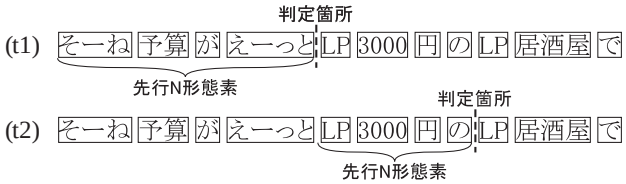

図 1: 提案手法の判定アルゴリズム

困った箇所では十分な一致が見られなかった.《ユーザ》が検索要求を言えずに困った状態かどうかの判断が人によって揺れたことは，このユーザ状態の判定が，ユーザ発話から認識できるある特徴に基づくものというよりもむしろ，次話者である《システム》個別の対話戦略に依存することを示唆する．

4 検索要求単位判定手法

IRUS であるか否かを判定する問題は 2 値の判別問題ととらえることができる．すなわち，発話区間中のある境界が IRUS である場合 $+1$, そうでない場合を -1 で表すとき，特徴量 x を用いてその符号を識別する問題と考えられる．そこで我々は SVM（Support Vetor Machine） を IRUS の境界判定課題に適用した．

一方，従来から行われている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，韻律や視線などのパラ言語情報や，検索スロットのような対話状態などを特徴量とする手法が提案されてきた [1, 4, 6]．

このような特徴量の内，今回我々は判別境界以前の発話に出現する情報を特徴量として用いた．これは我々が本手法を逐次的に応答が可能な音声対話システムに適用することを想定しているためである．具体的には，表 4 に示すような特徴量を用いた．

ここで，スロット状態とは，あらかじめ定めたスロットの項目が明示された場合に，その内容が発話されたかどうかを表す．例えば，図 1 の $t1$ では，予算という項目が明示されたものの内容が発話されていないため「埋まっていないスロットあり」，$t2$ では 「埋まっていないスロットなし」となる．

5 検索要求単位判定評価実験

5.1 実験の目的

提案手法の有効性を評価するための以下の 2 種類の実験を行った．

実験 1: 特徴量の有効性

実験 2: 音声対話システムに適用した場合の有効性

表 4: 判別に用いる特徴量

特徴量	特徴量説明
形態素	見出し語
	品詞，品詞細分類
	活用型
	活用形
文節	文節区切りかどうか
係り受け	現文節より前に係り先の決まっていない文節があるかどうか
スロット状態	埋まっていないスロットがあるかどうか
音の長さ	当該アクセント句の末尾以前の平均モーラ長 (子音部除く) と最終モーラ長 (子音部除く) の差

表 5: 特徴量の有効性：実験条件

条件 1:	形態素
条件 2:	形態素+文節+係り受け
条件 3:	形態素+音の長さ
条件 4:	形態素+音の長さ+スロットの状態
条件 5:	形態素+文節+係り受け+音の長さ+スロット状態

まず 実験 1 では，前節で述べた特徴量の内，表 5 に挙げる 5 通りの組み合わせにおいて，どれが判別に有効かを検証した．なお，特徴量の純粋な寄与を検証するため，形態素・文節・係り受け情報は人手で作成した正解を用いた．また，従来手法との比較のため，以下のふたつのシステムとの比較も行った．

比較システム 1（LP 検出システム）： 500msec 以上無音区間がある場合に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 [5].

次に実験 2 では，音声認識器の性能が IRUS 判定 に与える影響を調査した．実験 1 では与えられた言語情報は正しく認識できると仮定して実験を行った．しかし，音声対話システムへの応用を考える時，音声認識誤りが言語処理に与える影響は無視できない．そこで，今回は男性話者 50 名の発声から得られた誤認識を含む音声認識結果を用いて予備的な調査を行った．音声認識器は sonic[3] を用いた．また，認識誤りを含む場合にはアクセント句の特定が困難なため，表 4 の「音の長さ」を正確に計算できない．そこで,「音の長さ」は判別箇所直前の 2 形態素（最終モーラおよび子音部を除く）の平均モーラ長と最終モーラ長との差で近似した．

5.2 実験結果

5.2.1 特徴量の有効性

200 対話を 6 分割して交差検定を行った．瞬時の判断による振舞いを注釈付けした S-IRUS に比べて，後から対話を見直して付与した L-IRUS は一貫性が高い [8]．そこで実験には 1 名のラベラーが付与した L-IRUS を利用した．また，再現可能性評価で一致率が低かった「検索要求を言えずに困った箇所」は事例から除いた．

判定率を 表 6 と 図 2 に示す．図 2 の横軸は「ユーザが検索要求を述べる途中で応答を返すことなく検索要求を聞けるか」，縦軸は「ユーザが検索要求を言い終えた箇所で適切に応答できるか」を意味する．

まず，表 6 が示す通り，ベースラインのふたつの手法は IRUS 境界である点とない点を同時に高精度で判別できない．したがって，イニシアチブ判定手法としては不適切である．

表 6: ベースラインシステムの判定結果（再現率）

	イニシアチブ非移行	イニシアチブ移行
比較システム 1	0 %	100 %
比較システム 2	95.7 %	30.5 %

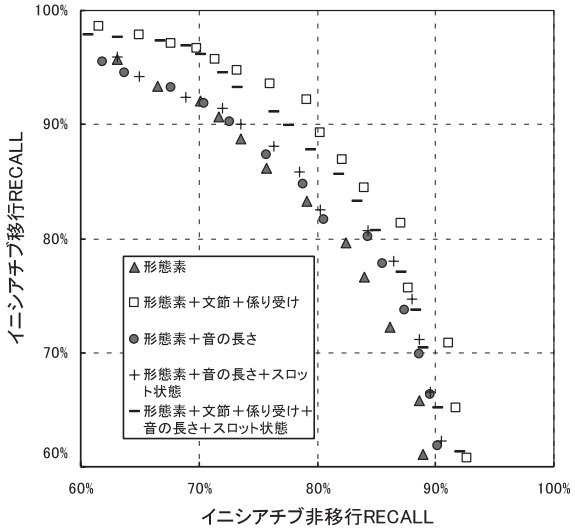

図 2: L-IRUS の判定結果

次に，提案手法でどの特徴量が判別に有効に働いたかを考察する．まず，条件 1 よりも 2 の方が精度が良いことから，文節や係り受けの情報が IRUS の判別に有効に働くと言える．また，これらの言語特徴量に音声やスロットの情報を加えるとさらに精度が向上した（条件 3，4）．ここで，形態素と音の長さのみを用いた条件 3 でも十分な精度が得られたことに注意したい．係り受け解析のような高級な言語処理を介さずに，音声認識結果から得られる特徴量のみで 判別が可能であったことから，IRUS 判定は逐次的な音声対話システムへも十分応用可能と考えられる．

5.2.2 音声対話システムに適用した場合の有効性

図 3 に，認識精度別に IRUS 判定精度を示す．図 3 中の白抜のグラフは形態素のみを判別の特徴量に用いたもので，黒塗のグラフは形態素と音声の長さを特徴量としたものを示す．

今回の実験では，音声認識率が低下すると IRUS 判定精度も低下するものの，音の長さを特徴量に加えることで誤認識による判定精度の低下が小さくなるとの結果を得た．形態素情報の一部が誤認識されても音の長さが正しく認識されていれば，IRUS 判定精度の低下を抑えることができると考えられる．韻律のような非言語的特徴を用いることで，音声認識誤りにある程度頑健な IRUS 判定が可能との見通しが得られた．

6 まとめと今後の課題

本稿では，システム応答の中でも特にイニシアチブの移行を伴う発話がタスク達成と関連が深いことに着目し，ユーザが発話を終えたという状態を機械的に判定する手法を提案した．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 IRUS を判定した結果，約 80 % の再現率 で IRUS を判別できた．また，IRUS

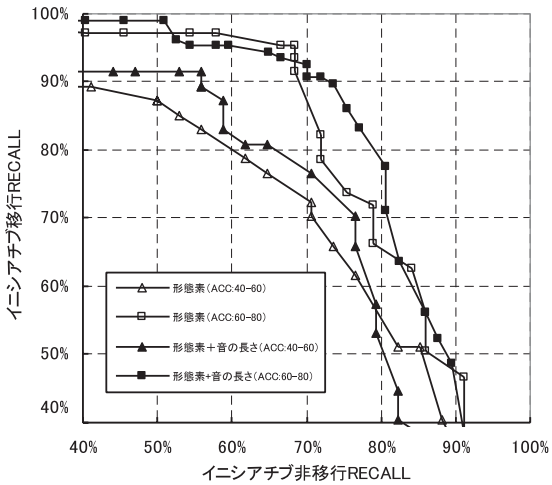

図 3: 検索要求単位判定精度と音声認識精度の関係

判定精度と音声認識精度との関係から，音声特徴量を加えることで，音声認識誤りによる IRUS 判定精度の低下を抑えられることを示した．

なお，今回は音声特徴量として音の長さを用いたが，音の長さ以外にも，音の高さ (f0) や強さ (パワ) を特徴量とすることが考えられる．現在の我々の試みでは，これらの音声情報は判別精度の向上に十分寄与しなかった．長さ以外の音声特徴量の利用に関しては，今後も検討を続ける予定である．

7 謝辞

東大の石崎助教授，千葉大の市川教授，伝助教授，堀内助教授には，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，奈良先端大の工藤氏には各種ツールの利用を始めとした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] L. Ferrer, E. Shriberg, and A. Stolcke. Is the speaker done yet? faster and more accurate end-of-utterance detection using prosody in human-computer dialog. *ICSLP-02*, pp. 2061–2064, 2002.

[2] Mikio Nakano, Noboru Miyazaki, Jun ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. Understanding unsegmented user utterances in real-time spoken dialogue systems. *ACL-99*, pp. 200–207, 1999.

[3] Bryan Pellom. Sonic: The university of colorado continuous speech recognizer. *TR-CSLR-2001-01*, 2001.

[4] Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. Learning decision trees to determine turn-taking by spoken dialogue systems. *ICSLP-2002*, pp. 861–864, 2002.

[5] 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.

[6] 大須賀智子, 堀内靖雄, 市川熹. 韻律からの文構造推定における局所的特徴の分析. *SIG-SLUD-A301-04*, pp. 1–6, 2003.

[7] 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.

[8] 徳久良子, 寺嶌立太, 脇田敏裕, 乾健太郎. 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. *03-SLP-49-42*, pp. 1–6, 2003.