OS-17

知的対話システムの形式的モデルとその実装

Formal Model of Intelligent Dialog System and it's Implementation

金盛　克俊[*1]　　大和田　勇人[*1]
KATSUTOSHI KANAMORI　HAYATO OHWADA

[*1]東京理科大学理工学部経営工学科
Tokyo University of Science, Faculty of Science and Technology

The purpose of our work is to achieve intelligent dialog system. The intelligent dialog system contains various integrant features. For example, intention analysis, utterance generation, and others. But, studies about them often has been investigated independently. A part of this reason is that there is no consistent model as interaction model We propose a formal model of intelligent dialog system, for this problem. The model is defined as a mathematical system constructed of few sets and functions. And we show a implementation example of this model to declare availability.

1. はじめに

音声や自然言語を用いた知的な対話システムの実現に向けて，古くから多くの試みが行われてきた．その実現には，意図同定や意味解釈，感情推定や発話文生成など，多くの困難かつ重要な要素技術が必要である。これらの対話システムに必要な要素技術はそれぞれ独立して研究されていることも多く，それが対話システムのために研究されているとしても，それらを単純に組み合わせて対話システムを構築することは容易ではない．各要素技術をどのように組み合わせて対話システムを構築すべきかという問題に対しては十分な議論がなされていない．それぞれの要素技術が成すべき役割や利用するリソース等が整理されていないことが一つの大きな原因である．このような問題を解決するため，本論文では知的対話システムの形式的なモデルを提案する．それぞれの要素技術を共通のモデルに基づいて研究開発することにより，これまで煩雑であった対話システムに関わる技術を整理し，対話システムを要素技術の組み合わせで開発できるようにするのが目的である．

本論文では，システムが持つべきリソースや知識を集合で表現し，それぞれの機能を関数で表現する形式的な対話システムフレームワークモデルを提案する．提案モデルにおいては，各意図同定や意味解釈などの機能は関数として定式化され，ユーザの発話文を入力として応答文を出力する一連の機能は，各機能の関数を用いて表現される．このようなモデルを想定することで，各機能の果たすべき役割と全体における立ち位置が明確になる．本モデルに基づく対話システムのための各要素技術の研究開発は，モデルにおける関数を求めることに他ならないため，実際に対話システムを構築する場合にはそれらの組み合わせで求めることが期待される

本論文では提案モデルの定義とともに，このモデルに基づいて実際に全ての知識体系と機能を実装した例を紹介する．本論文で示す実装例は，それぞれが単純な仕組みしか持たないが，そのような単純な機能の組み合わせでも，対話システムが構築可能であることを示している．

連絡先: 金盛克俊，東京理科大学理工学部経営工学科，千葉県野田市山崎 2641，katsu@rs.tus.ac.jp

2. 提案モデル概要

本モデルは自然言語で与えられた入力文に対して，応答文を返す対話システムの形式的なモデルである．音声対話処理に適用するには，当然のことながら音声入力を自然言語文に変換する必要がある．

提案モデルは，システムの持つ知識やそれを表すために必要な概念の集合を持ち，各機能を表す関数を持つ体系として定義される．

それぞれ関数は本モデルにおいて対話処理に必要な各機能に対応しており，入力文から応答文を返す応答文生成関数は，これらの関数を用いて定義される．このモデルに基づいて実際に対話システムを実装する場合には，それぞれの関数について具体的な機能を実装すればよい．意図解釈や意味解析など，対話システムを想定した要素技術の研究開発においても，このようなモデルを想定しておくことで，それぞれの機能が満たすべき役割と利用すべきリソース等が明確になる．

2.1 実装例概要

提案モデルの有用性を示すため，簡単な実装例を紹介する．この例は，極めて単純な機能の組み合わせで構成されるが，それでも簡単な対話システムが実装できることを示す．

実装例は日本語の単文による対話を対象として，名詞同士の is-a 関係による階層構造を持つ知識をもとに受け答えを行う，質問応答システムである．このシステムは，ユーザの発話文を受け取ると，それが質問なのか教示なのかを判断し，それに応じて適切な応答文を返す．いわゆる What 型の質問に対応しており，例えば”犬とは何ですか？”という入力文に対しては”動物です”というような上位概念を答えるような応答文や，”好きな果物は何ですか？”という入力文に対して”リンゴです”と答えるような具体例を答えるような応答文を返すことを目標としている．また，”バナナは果物だよ”というような教示の入力文が与えられたときには，応答文を返すだけではなく，システムの知識として新たに学習する機構も備える．

3. モデルを構成する概念と集合

3.1 意図

対話システムにおいて，ユーザの発話文の意図を解析することは重要である．実用化されている音声対話システムしゃべっ

てコンシェル [?, docomo] おいても，意図は呼び出すべき携帯端末の機能と結びついており，対話により機能を呼び出すために重要な役割を担っている．対話における意図同定の研究では，対話における一般的な意図の階層的な分類方法等も提案されている [J. Allen 97].

主に発話文の意図を表すために，発話文のある種の性質を意図と呼ぶこととし，意図集合 I を意図を要素として持つ集合と定義する．

意図という言葉を使っているが，ここでは理解を助けるため便宜上意図という名前を付けているだけで，必ずしもそのような意図である必要はない。モデルにおける意図の形式的な役割に注目し，実装するシステムにおいて必要な性質を意図集合として求めるのが重要である．

通常はシステムが想定すべき有限個の意図を用意すれば十分であると考えられるが，無限集合であっても，動的に意図集合が更新されていくようなものであってもよい．

3.1.1 実装例

What 型の質問に答える質問応答システムにおいて想定されるユーザの発話文に対して判別すべき意図と，システムが構成する応答文の種類を区別するために必要な意図を用意する．

対象物の上位概念を求める質問文の意図を q_{wh}，下位概念を求める質問文の意図を q_{ex}，yes か no で答えるべき質問文の意図を q_{yn} とし，それぞれに答える応答文の意図を a_{wh}，a_{ex}，a_y，a_n とする．また，質問に答えられないときの意図として a_{un} を与える．さらに，ユーザがシステムに新たな知識を与えるための発話文の意図として t，それに対する応答文の意図として t_{re}，これら以外の文の意図として $else$ を与え，意図集合 I を次のような有限集合とする．

$$I = \{q_{wh}, q_{ex}, q_{yn}, a_{wh}, a_{ex}, a_{yn}, a_{un}, t, t_{re}, else\}$$

3.2 意味と知識

意味は理想的には言語によらない，発話文が持つある性質であり，環境の状態や知識と結びつく重要な性質である．意味集合 M はそのような取りうる全ての性質の集合として定義される．意味集合は一般には無限集合と考えるべきである．

対話システムにおいて，意味や知識をどのように表現すべきかという問題は重要である．論理式や，意味ネットワーク，オントロジーなど，システムの目的によって適切な知識表現は異なり，様々な表現が考えられるが，ここでは，単に，ユーザの発話や状況に対して，応答文の生成に必要な意味を出力することができる体系であるととらえることにする．すると知識は，意味集合の部分集合を内在する何らかの体系であると考えることができるが，ここでは知識 K を単純に意味集合の部分集合と定義し，知識体系が持つべき関数等は別に定義することにする．

3.2.1 実装例

意味は名詞の組あるいはその集合で表され，名詞の組は 2 つの名詞の is-a 関係を表す．例えば，(犬, 動物) は犬と動物が is-a 関係を持つことを表す．意味集合 M は，名詞の組の集合として定義される．

知識はシステムが知っている意味の集合であり，M の有限部分集合である．例えば知識 K は以下のように定義される．

$$K = \{(犬, 動物), (猫, 動物), (苺, 果物)\}$$

実装例においては，定義通り知識は意味の集合であるが，知識表現が意味の集合でなければならないわけではないことに注意されたい．このモデルで重要なことは，意味を引数として知識から新たな意味を出力する関数が実装可能なことである．

3.3 感情

本モデルにおいて，システムの応答文は意図や意味によって構成されるが，ユーザの発話に対して応答文として適切な意図と意味が複数ある場合，または意図と意味が決定した場合においても，複数の応答文候補が存在する場合には，それらを絞り込み、最終的には一つの応答文を生成しなくてはならない。このような役割を担うのが感情である．意図と同様に，理解を助けるため感情という言葉を用いているが，実装する際には必ずしも人間の感情に基づいた実装をなさなければならないものではない．ここでは感情集合 E は距離 d を持つ距離空間として定義される．

システムは現在の感情状態として e_{sys} を持ち，これと発話文の感情を用いて複数の候補は絞り込まれることになる．例えば，応答文候補の感情を求めることにより，現在のシステムの感情状態と距離 d において最も近いものを選択することができるようになる．

3.3.1 実装例

快，不快を表す簡単な感情の実装を考える．感情空間 E として，一次元の数直線上の区間 $[-1, 1]$ を考える．

$$E = [-1, 1]$$

E 上の距離 d は 2 点の数直線上の距離と定義する．

また，この実装例においては，知識に含まれる意味に現れる名詞には感情が付与されている．

4. モデルを構成する関数

4.1 意図同定

ユーザの発話文の意図を解釈するための関数である．意図同定関数は，文字列から意図集合への写像である．文字列の集合を S，意図集合を I とすると，意図同定関数 a_i は

$$a_i : S \to I$$

と定義される．

4.1.1 実装例

語尾や名詞を含む文節同士のかかり受け関係により，発話文から意図を同定する．例えば，文中に代名詞”何”や”なに”を含む文節があり，それが他の名詞を持つ文節にかかっていれば意図は q_{wh}，ただしその場合他の名詞を持つ文節に係る形容詞か形容動詞が存在すれば q_{ex} となる．

$$a_i(\text{”犬って何？”}) = q_{wh}$$

$$a_i(\text{”好きな動物は何ですか”}) = q_{ex}$$

全てを紹介する余裕はないが，このような単純なヒューリスティクスにより意図は同定される．

4.1.2 感情推定

ユーザの発話文の感情を推定する関数である．感情推定関数は，文字列から感情集合への写像である．文字列の集合を S，感情集合を E とすると，感情推定関数 a_m は

$$a_e : S \to E$$

と定義される．

4.1.3 実装例

発話文に出現する形容動詞"好きだ"（あるいはその語幹）の個数を p，形容動詞"嫌いだ"（あるいはその語幹）の個数を n とするとき，発話文の感情 e_s は次の式で求められる．

$$e_s = (p - n)/(p + n)$$

この式に基づき，発話文の感情を求めた例が以下である．

$$a_e(\text{"好きな動物は何ですか"}) = 1$$

$$a_e(\text{"果物は嫌いです"}) = -1$$

4.2 意味解釈

ユーザの発話文の意味を解釈する関数である．意味解釈関数は，文字列から意味集合への写像である．文字列の集合を S，意味集合 M とすると，意味解釈関数 a_m は

$$a_m : S \to M$$

と定義される．

4.2.1 実装例

係り受け解析により，名詞を含む文節同士の係り受け関係を調べ，文節 ph_1 に係る ph_2 に対して，それぞれに現れる名詞を no_1, no_2 とすると，意味 (no_1, no_2) を返す．ただし，no_2 が代名詞"何"である場合には，no_1 に係る形容詞あるいは形容動詞が存在する場合には，意味 (no_2, no_1) を返す．例えば，

"犬って何？"

の意味は (犬, 何) であり，

"好きな動物って何？"

の意味は (何, 動物) となる．

4.3 発話文生成

システムの応答文は，意図と意味の組から構成される．その構成の仕方にはシステムの感情やユーザの発話の感情にも影響されるべきであるので，発話文生成関数は，意図，意味，システムの感情，ユーザ発話の感情の４つ組から文字列への写像と定義される．

$$g_s : I \times M \times E \times E \to S$$

4.3.1 実装例

名詞が代入可能な変数を含む文字列を，発話文のテンプレートとして利用することを考える．テンプレートには，意図が対応付けられている．例えば，以下のようなものになる．

"X_1は X_2だよ" : 意図 $= t$

"X_1が好きだよ" : 意図 $= a_{ex}$

"X_1です" : 意図 $= a_{wh}$

このようなテンプレートの集合を T とする．テンプレートは，感情推定関数により感情を計算することができる．また，テンプレートは通常の発話の名詞部分を変数に置き換えて得られるものであり，文節の係り受け関係は失われておらず，意味解析が可能であるとする．

引数で与えられた意図に対して，意図が一致するテンプレートの中から，システムの感情に最も近いものを選ぶ．引数で与えられた意味をもとに，テンプレートの変数を具体的な適切な名詞に置き換える．

例えば，

$$g_s(a_{wh}, (\text{苺}, \text{果物}), 0, 0)$$

に対して，テンプレート

"X_1です" : 意図 $= a_{wh}$

が得られたときには，意図が a_{wh} であるので，上位概念である 果物 を X_1 に代入し，発話文

"果物です"

が得られる．

4.4 応答意図決定

応答文を構成する意図を決定する関数である．応答意図は，ユーザの発話文の意図と意味に大きく関係する．また，応答意図には複数の候補が存在する場合が考えられるため，応答意図決定関数 d_i は，意図，意味，システムの感情，ユーザ発話の感情の４つ組から意図への写像と定義される．

$$d_i : I \times M \times E \times E \to I$$

4.4.1 実装例

システムは応答意図決定のために，意図対応表を持つ．意図対応表は入力意図と引数で与えられた意図を知識に含むかどうかに対して，適切な応答意図を持つ表である．表 1 はその一部である．

入力意図	意味を含む	応答意図
q_{wh}	true	a_{wh}
q_{wh}	false	a_{un}
q_{ex}	true	a_{ex}
q_{ex}	false	a_{un}

表 1: 意図対応表

ただし，知識 K が意味を含むとは，意味 $(no_1, no_2) \in K$ であるか，no_1, no_2 どちらかが代名詞"何"でもう片方を同じ側に持つ意味が少なくとも一つ K に含まれるかどうかを表す．例えば入力意味 (犬, 何) である場合には，(犬, 動物) $\in K$ であれば意味を含むという．

なお，この実装例においては意図対応テーブルにおける入力意図と応答意図は 1 対 1 対応であり，感情による影響は受けない．

4.5 応答意味決定

応答文生成のために，意味を決定する関数である．ユーザの発話の意味と意図によって知識から抽出すべき意味を決定するが，場合によっては複数の候補が存在することが考えられるため，応答意味決定関数 d_m は，意図，意味，システムの感情，ユーザ発話の感情の４つ組から意味への写像と定義される．

$$d_m : I \times M \times E \times E \to M$$

4.5.1 実装例

応答意図決定関数と同様に，入力意味を知識に含んでいるかどうかで挙動は異なる．入力意図が q_{wh} である場合には，知識から意味の第一要素が一致する意味を候補として抽出する．入力意図が q_{ex} である場合には，知識から意味の第二要素が一致する意味を候補として抽出する．複数の候補がある場合には，答えるべき名詞の感情とユーザの発話の感情が最も近いものを選ぶ．それ以外の意図については，入力意味と一致する意味が知識に含まれれば入力意味をそのまま返す．入力意味が含まれない場合には，必ず変数を持たないテンプレートが使用されるため，意味を使用する必要はなく，NULL が返される．

例えば，

$$d_m(q_{ex}, (\text{何}, \text{果物}), 0, 1)$$

に対して，複数の候補 (苺, 果物), (リンゴ, 果物) があったとき，苺, リンゴ のうち感情が 1 に近いほうが選ばれる．

4.6 応答文生成

ユーザの発話文を入力とし，システムの応答文を返す関数で，これまでに紹介した関数を用いて定義される．応答文生成関数 F は，文字列から文字列への写像で，入力文 s が与えられたとき，以下の式で定義される．

$$F = g_s(d_i(a_i(s), a_m(s), e_{sys}, a_e(s)),$$
$$d_m(a_i(s), a_m(s), e_{sys}, a_e(s)), e_{sys}, a_e(s))$$

5. 学習

対話システムにおいて，対話の中から知識を学習するようなシステムについては十分な研究がなされていないが，ここでは，実装例を用いて本モデルに基づく単純な学習機構が実現できることを示す．本論文で紹介してきた実装例が持つ知識は，初期状態ではごく少数の知識しか持っていないとしても，ユーザとの対話により知識を獲得していく学習機構を持つ．

対話による知識獲得は，ユーザの全ての発話が学習の対象となるとは考えにくいため，ここではユーザの意図がシステムに新たな知識を与えようとしているかどうかを判断し，そのような意図である場合にのみ，その発話の意味を知識に追加（格納）していく．

実装例では，ユーザの発話文の意図が t であるとき，その発話文の意味を知識に追加していく機構を持つ．

6. 実装例による対話の例

以下は実装例における対話の例である．What 型の質問に答えられていることと，学習により新たな知識を獲得していることが確認できる．

ユーザ	犬って何？
システム	犬は動物だよ
ユーザ	トカゲって何？
システム	知りません
ユーザ	トカゲは爬虫類だよ
システム	そうでしたか
ユーザ	トカゲって何？
システム	トカゲは爬虫類だよ

表 2: 対話例

7. モデルの拡張

本論文では知的対話システムを実現するための最小限と考えられるモデルを提案したが，実装するシステムによってはこれだけでは不十分な場合もある．このような場合にも，モデルを拡張することにより対応可能であることを例を用いて紹介する．

7.1 状態を用いた拡張

このモデルで用いた知識は恒久的な情報を前提としている。質問応答システム等，システムとの対話が 1 度で終了するようなものはこれで十分だが、対話が連続して発生する対話システムでは，文脈を考慮することは重要である．また，対話エージェントが置かれている状況によって，生成すべき発話が異なる場合も考えられる．

このような場合には，モデルに状態集合を新たに用意して，文脈による状態を考慮したモデルの拡張が考えられる．文脈の推移や感情の推移を状態として定義し，状態によって適切な応答文が生成されるような拡張を行えばよい．

8. まとめ

知的対話システムを構築するための形式的なモデルを提案し，その簡単な実装例を紹介した．それぞれの機能は単純であっても，本モデルに従うことにより一貫した対話システムが構築できることを示した．今後は，このモデルを用いた具体的なシステムの開発や，このモデルにおける各関数の研究が期待される．

参考文献

[吉村 12] 吉村 健: しゃべってコンシェルと言語処理, 情報処理学会研究報告, Vol.2012-SLP-93 No.4(2012).

[J. Allen 97] J. Allen and M. Core : Draft of DAMSL: Dialog act markup in several layers, 1st Discourse Tagging Workshop (1997).