

7M-5

対話システムにおけるプラン説明文の生成

小谷 亮 今村 誠 近藤 省造  
三菱電機株式会社 情報電子研究所

1 はじめに

コンサルテーションのための対話システムにおいては、システムが作成したプランをユーザに説明する文章を生成する必要がある。Mellish[1]は汎用のプランナーから生成されたプランの説明文生成の研究を行なっているが、生成される文章が退屈で冗長なものになってしまうという問題点があった。Mellishはこの問題をいくつかの原因にわけて論じているが、一つの大きな原因は知識の難易度について考慮しなかったことにあると考えられる。つまり、ユーザが容易にわかるようなことまで文章に盛り込んでいたため、噛んで含めるような文章になっていたわけである。発話内容(what to say)の選択は、理解しやすい文章の生成に本質的な役割をもつ。

対話における応答文では、文章中に必要な情報が簡潔に表現されていることが特に要求される。Chin[2]はUNIX OSに関するコンサルテーションシステムUCにおいて、ユーザのUNIXに関する習熟度および知識の難易度の指標を定義し、ユーザが知らない情報のみを含む応答文を生成することにより、発話内容の選択の問題をあつかった。ただし、Chinのあつかった応答文は主にシステム内にあらかじめ蓄積されている静的な知識をそのまま説明する文に限られ、動的に生成されたプランの説明文はあつかわなかった。

現在我々は、旅行者のトラブルに関するコンサルテーションシステムを試作中である。このシステムは、旅行者のトラブルを解決するプランを立案し、そのプランの説明文を生成する。この際に、次のようなアプローチをとった。

1. プランニングの結果を推論ネットというデータ構造で表現する。
2. システムのもつ静的な知識のそれぞれに難易度を定義しておく。また、ユーザの博識度を仮定する。推論ネットの各ノードがあらわす知識の難易度とユーザの博識度を比較することにより、文章に盛り込むべき知識を選択する。
3. 推論ネットの書き換えをおこない、FTS (Functional Text Structure)[3]に変換する。その際、各知識が文章中に出現すべき順序に関する制約(順序制約)を抽出する。

2により、ユーザが必要としている(と推定される)知識を選択し、必要十分な内容を文章に盛り込むことができる。また、3では、文章においてそれぞれの内容が現れるべき順序を、推論ネットの構造から抽出することができる。

2 プランニングと推論ネット

本研究では、システムは旅行者であるユーザのゴールの達成を支援するために次のような手順でユーザと対話する。

1. ユーザに質問を行なうことにより、ユーザのゴール及び現在ユーザがおかれている状況を把握する。
2. ユーザのゴールを達成するために、ドメインのプランニングを行なう。
3. 2で得られたドメインプランをユーザが行なうように説得する文章を生成する。

Generating Explanatory Texts of Plans in Dialogue Systems  
Akira KOTANI, Makoto IMAMURA, Shozo KONDO  
Mitsubishi Electric Corporation

本研究は、ICOTからの委託により第五世代コンピュータプロジェクトの一環として行なわれた。

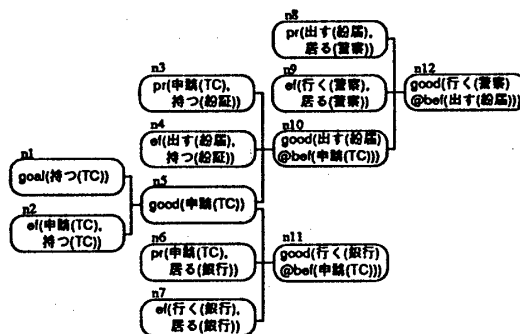


図1: 推論ネット

2の処理では、ユーザの意志を考慮せず、システムがユーザを意のままに動かせるものと仮定して、ドメインプランを生成する。このことによって得られたドメインプランをユーザが実行すればユーザのゴールは達成されるが、そのためにはユーザがドメインプランを行なうように説得する必要がある。そこで3では、ユーザがドメインプランを実行することがユーザのゴールの達成に益することをユーザに説明する。

以下では、TC(トラベラーズチェック)を失ったユーザがシステムに相談を持ちかけた場合を想定して説明を行なう。システムはユーザと次のような対話を行なうことにより、ユーザのゴール及び現在ユーザがおかれている状況を把握する。

- U1: TCを落としてしまったのですが。  
S1: TCの購入控えはもっていますか?  
U2: はい。  
S2: TCの紛失証明書はもっていますか?  
U3: いいえ。

この対話では、U1からシステムはユーザが「持つ(TC)」というゴールを持っていることを知り、U2から「TCの再発行を申請する」という行為が適用可能であることを知る。次に、システムはこのゴールを達成するためのプランを立案し、それと同時にプランニングの過程を推論ネットとして表現する。この場合に得られる推論ネットを図1に示す。

推論ネットはプランニングの途中で得られる構造であり、あらかじめシステムに蓄積されている静的な知識やプランニングの結果得られる動的な知識の間の依存関係をあらわしている。例えば、ノード n1 が表現している「TCを持つというゴールをユーザが持っている」という知識とノード n2 が表現している「TCの再発行を申請すれば、TCを持つことができる」という知識からノード n5 の「TCの再発行を申請することは、ユーザの現在のゴールに寄与する」ということが推論されたことが推論ネットに表現されている。

推論ネットのノードの内容は実際に文章化されるべき内容の候補をあらわしている。これには次のような種類がある。

1. ユーザの現在のゴール  
goal(Proposition)
2. システムに蓄積されている静的なプラン知識(行為の前提条件及び効果)  
pr(Action,Proposition), ef(Action,Proposition)
3. 推論の結果得られる動的な知識  
good(Action)

なお,  $pr(\text{Action}, \text{Proposition})$  は命題 Proposition が行為 Action の前提条件 (precondition) であることを,  $ef(\text{Action}, \text{Proposition})$  は命題 Proposition が行為 Action の効果 (effect) であることをそれぞれあらわす。また,  $good(\text{Action})$  は「ユーザ行為 Action を行なうことは, ユーザの現在のゴールの達成に寄与する」ということを意味している。ただし, 文章中では  $good(\text{Action})$  を「Action をして下さい」という依頼文の形で生成する。

ユーザのゴールが達成されるためには, システムが立案したプランを実際にユーザが実行する必要がある, それにはユーザがプラン中に含まれる各行為を実行する意図をもたなければならない。したがって, ユーザは3の  $good(\text{Action})$  という知識に関して必ず知る必要がある。ユーザを説得するためには基本的にはこれをユーザに依頼すれば良い。しかし, それがユーザのゴールに寄与する理由を説明する方がユーザの理解の助けになる。それぞれのノードの内容が推論された理由は推論ネットの上でリンクを辿ることにより得られる。ただし, その際ユーザがすでに知っていると考えられる内容は省いて構わない。したがって, 2のタイプのノードの内容については, ユーザがその知識をまだ知らないと予想される場合のみ説明に盛り込む。ユーザのゴールに関しては, 現在ユーザは必ず知っているものと仮定しており, 文章からは省かれる。

このような方法で文章化する内容を選択するために, ユーザの知識に関して, 次のように仮定する:

仮定 ユーザがある静的な知識を知っているかどうかは, ユーザの博識度とその知識の難易度の値の比較によって決まる。

プランニングおよび推論ネットを作る際に用いられる静的な知識のそれぞれに対して, あらかじめ知識の難易度の情報を定義しておく。例えば, 図1のノード n7 のあらわす「銀行へいけば, その効果として銀行にいくことになる」という知識は, ノード n3 のあらわす「TCの再発行を申請するためには, 紛失証明書を持っていなければならない」という知識と比較して自明であるから, 低い難易度を設定しておく。システムは, 推論ネット上において静的な知識を表現しているノードがもつ難易度の値と, ユーザに対して仮定されている博識度の値を比較することにより, どのノードの内容を文章に盛り込むかを決定する。

### 3 推論ネットから抽出される順序制約

図1のノード n5 およびノード n10 は, それぞれ次の内容をあらわしている。

- 「TCの再発行を申請することはユーザの現在のゴールに寄与する」
- 「TCの再発行を申請する前に紛失証明書の発行を申請することはユーザの現在のゴールに寄与する」

ユーザが2の内容を理解するためには, その前に1の内容を理解しなければならない。なぜなら, 2の内容は「ユーザがTCの再発行を申請する」ことを前提としているからである。したがって, これらの内容が文章中に現れるときには, 1,2の順序で出現しなければならない。このような表層文章における順序は, FTSの順序制約を用いることにより記述することができる。

### 4 生成処理

以下のような手順で推論ネットの内容を文章化する。

- 推論ネット中に含まれる各ノードのうち, その知識の難易度の値がユーザの博識度の値よりも小さいものを推論ネットから削除する。
- 推論ネットにおいて  $good(\text{Action})$  という形のノード同士を結んでいるアークを切ることにより, 推論ネットを分解する。そのとき, 同時に FTS の順序制約を得る。
- 可読性のため, 必要な場合にはノードの内容を他のノードの内容に埋めこむ。
- 以上の処理により書き換えられた推論ネットを FTS に変換する。

```
[thesis = n5:good([行く(銀行), 申請(TC)])],
[thesis = n10:good([行く(警察), 出す(紛届)]@bef(申請(TC))
reason = [thesis = n3:pr(申請(TC), 持つ(紛証))],
[thesis = n4:ef(出す(紛届), 持つ(紛証))]]]
n5 < n10
```

図2: 推論ネットから得られる FTS

銀行へ行って TC の再発行を申請して下さい。  
TC の再発行を申請するには紛失証明書を持っている  
必要があり, それは紛失届を出せば持つことができます。  
したがって, TC の再発行を申請する前に警察へいっ  
て紛失届を出して下さい。

図3: FTS から生成される文章

### 5. FTS から文章の接続表現を決定し, 表層文を生成する。

図1の推論ネットから作られる FTS を図2に示す。FTS は, 文章中におけるそれぞれの文の機能的な関係を表現したものである。FTS から生成可能な接続表現を評価し, 最も適当なものを選択して生成することができる。この FTS から生成される文章を図3に示す。この文章をみると, 細かい表層表現の点で不適切な部分もみられるが, 適切な内容が選択され, また内容の順序や接続表現も適切に生成されていることがわかる。

### 5 おわりに

コンサルテーションシステムにおいて, システムが生成したプランの説明文を生成するために, プランニングの結果を推論ネットというデータ構造で表現し, そこから文章に盛り込む内容を選択する方法, および文章中に各内容が現れるべき順序を抽出する方法について述べた。最近の文生成では, 生成される文が読者に与える効果を計算する方法が研究されつつあるが, 本研究はそのような方向を目指すものである。現在この方法によって応答文を生成する対話システムを試作中である。

以下に問題点と今後の課題を挙げる。

- この方法によれば, 各知識に対して難易度の情報をあらかじめ与えておく必要がある。これは人手によらなければならない, かなり面倒な作業となる。
- ここでは文章化する内容の選択を難易度という文脈に依存しない情報を用いて行なったが, 実際には選択すべき内容は文脈の影響を受けると考えられる。そこで, どのような影響がどれくらい強く働かかを評価する必要がある。
- 対話においてシステムが説明した内容に関し, ユーザがその理由を質問した場合 (follow-up question), 推論ネットの情報を用いて質問に答えることができると考えられる。この方法は, システムがある内容をユーザが知っているものと仮定して文章から省いたにもかかわらず, 実際にはユーザが知らなかった場合などに用いることができるであろう。

### 参考文献

- [1] Mellish, C. and Evans, R: *Natural Language Generation from Plans, Computational Linguistics* vol.15, No.4 (1989) 233-249.
- [2] D.N.Chin, *KNOME: Modeling What the User Knows in UC*, In: Kobsa, A. and Wahlster, W. eds.: *User Models in Dialog Systems*, Springer-Verlag (1989)
- [3] 小谷, 今村, 近藤: 文章の機能的構造からの接続表現の生成, 情報処理学会, 自然言語研究会 82-7 (1991)