

コーパスに基づく発話理解・対話制御・発話生成 -コーパスベース音声対話システムの構築-

林 啓太† 入江 友紀‡ 山口 由紀子§ 松原 茂樹†§ 河口 信夫†§

†名古屋大学工学部電気電子情報工学科 ‡名古屋大学大学院情報科学研究科
‡名古屋大学統合音響情報研究拠点 (CIAIR) §名古屋大学情報連携基盤センター
keita@inagaki.nuie.nagoya-u.ac.jp

1 はじめに

近年、音声処理技術の向上により、音声対話システムの研究が盛んに行われてきている。コーパスベースシステムでは、個別の発話に対応したルールを作成する必要がなく、発話の多様性にロバストに対応可能なことが期待できる。S.Young は、各要素技術に対し確率的なアプローチを採用することにより、低コストでかつロバストなシステム構築を提案している [3]。我々はこれに対して、大規模な事例データを利用したアプローチを採用する。

本稿では、大規模な意図タグつき音声対話コーパスを利用した音声対話システムの構築について述べる。本手法では、発話理解、対話制御、応答生成の音声対話システムの主要な構成要素のすべてをコーパスベースで作成した。評価実験の結果、コーパスベース音声対話システム特有の傾向が明らかになり、事例の追加による性能向上について確認した。

2 意図タグつきコーパス

本研究では、名古屋大学統合音響情報研究拠点 (CIAIR) が収録した車内音声対話コーパス [1] のレストラン検索対話を利用した。レストラン検索対話はドライバとオペレータとの間で交わされたドライバ発話、オペレータ発話からなる。これらの発話にはすでに、事例を用いた対話処理のために設計されたレストラン検索タスクに依存した意図タグが付与されている [2]。原則として意図タグは、一発話に対して一つ付与されている。

3 コーパスベース音声対話システム

意図タグつきコーパスを利用してレストラン検索音声対話システムを構築した。本手法による対話システムの構成は図 1 のようになる。各モジュール間での情報のやり取りに意図タグを用いることにより、対話全体の流れを意図タグの流れとして捉えることができる。すなわち、意図タグという抽象化した情報により対話全体を処理できる。例として、実際の対話の流れとその処理に用いる情報で図 2 に示す。

入力発話 U1「どこかにおいしいイタリア料理の店はない」に対し、まず、発話理解によりその意図が「依頼+検索+店」と推定する。推定された意図と入力文

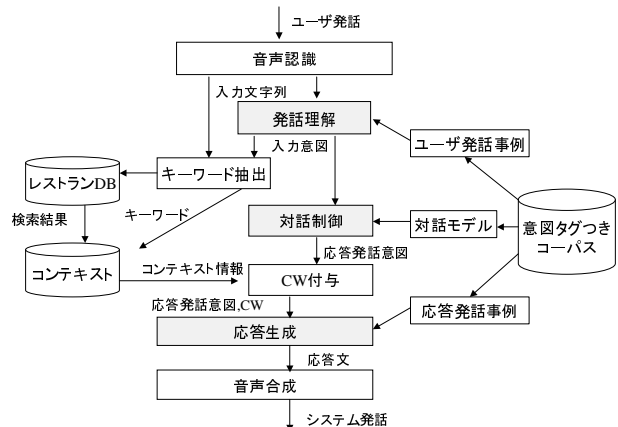


図 1: コーパスベース音声対話システムの構成

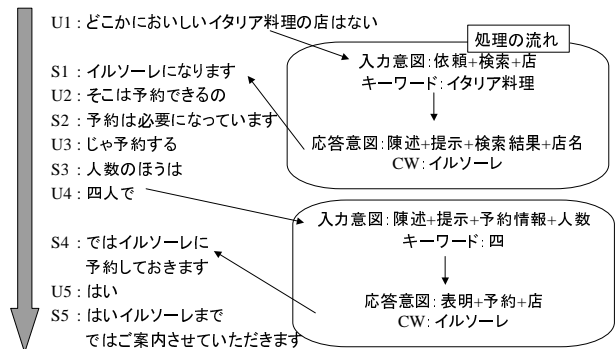


図 2: システムの動作例

から「イタリア料理」という名詞を検索用キーワードとして抽出し、データベースを検索し、「イルソーレ」という店の情報をコンテキスト情報として保持する。もし入力発話の意図が U4 の「陳述+提示+予約情報+人数」のように具体的な情報を含むものを示す意図であるならば「四」という数をキーワードとして抽出する。これらのキーワードは、入力発話を形態素解析した結果から、該当する品詞の形態素を取り出すことにより獲得する。

次に、対話制御により入力発話意図「依頼+検索+店」に対する応答発話意図として「陳述+提示+検索結果+店名」を決定する。応答文生成には、発話意図に加えて具体的な情報 (Contents Words, CW) を与える必要がある。CW は 1 個以上の名詞からなり、コンテキスト情報から出力される意図に沿ったものや、現在の対話の状況にあったものをルールによって選び出す。U1 に対するの応答意図は「陳述+提示+検索結果+店名」であるため、CW として「イルソーレ」という店名を与える。

表 1: 実験に利用した事例

	実験 1		実験 2	
発話理解	188 名	3,810 発話	483 名	5,629 発話
対話制御	188 名	7,113 発話	483 名	12,743 発話
発話生成	72 名 2,546 発話			

最後に、応答意図と CW によって応答生成で応答文 S1 を生成する。

3.1 発話理解

コーパスベースの発話理解では、直前の発話の意図タグが一致する事例文と入力文との類似度を計算し、類似度が最も高い事例文の意図を入力文の意図とする [2]。類似度計算には、形態素 (固有名詞に対しては単語クラス情報) の一致度を利用する。

3.2 対話制御

システム応答発話意図を得るために、意図タグつきコーパスから作成した意図タグの 3-gram による対話モデルを利用する。発話理解から得た入力発話の意図とその直前に行われた発話の意図が一致したもの中から、尤度の最も高いものを選ぶ。失敗した場合は 2-gram を用いて応答発話意図を決定する。

3.3 発話生成

本手法では、コーパス中のオペレータ発話についての発話意図、係り受け構造、単語の意味属性情報を付与した事例データベースを利用する。

まず、事例データベースからシステム応答発話意図に対する事例を抽出し、事例中の単語を CW に置き換え、係り受け構造を複数生成する。次に、生成した係り受け構造に対し、係り受けの非交差性、後方修飾性を満たしたあらゆる語順で表層文を生成する。

生成された応答文には、不自然な表現になっているものや、詳細な意図が異なってしまうものもある。そこで、前述の発話理解と同様の手法で応答発話意図と合致する発話文を選び出すようにフィルタリングを行い、その中の一つを出力として選び出す。

4 評価実験

4.1 実験の概要

本手法を WindowsXP 上に言語として Java、形態素解析を Chasen[4]、データベースを MySQL[5] を用いて実装し、評価実験を行った。評価実験はデータ量が異なる 2 つの事例を用いて行った (表 1)。テストセットは、事例データとは別の 100 名分の対話より抜粋した 100 タスク 483 発話のドライバ発話をテストセットとして利用した。

4.2 実験結果

各構成要素における正解率を表 2 に示す。評価基準として、発話理解と対話制御はテストセットの意図タグとの客観評価で行い、発話生成については、生成された発話文を入力文と照らし合わせて主観評価で行った。タスク達成率は入力された一連のタスクにおける主観評価により測定した。

得られた正解率、タスク達成率ともに必ずしも満足のいくものではなかった。また、事例を増やした場合でも、発話理解、対話制御の正解率については大きな効果が得られなかったが、発話生成の正解率及び、タスク達成率は向上した。これは、事例を増やしたことにより、

表 2: 実験結果 (100 タスク 483 発話)

学習データ	実験 1	実験 2
発話理解の正解率	60.04%(290)	60.87%(294)
対話制御の正解率	36.23%(175)	37.27%(180)
発話生成の正解率	61.08%(295)	64.80%(313)
タスク達成率	17.00%(17)	26.00%(26)

表 3: 高頻度意図の応答文正解率

入力文の正解意図	出現数	実験 1	実験 2
依頼+検索+店	106	89.62%	83.96%
陳述+選択+店	85	61.18%	65.88%
依頼+再検索+店	48	77.08%	79.17%
陳述+選択+ジャンル	48	85.42%	83.33%
陳述+提示+意思内容+肯定	43	34.88%	62.79%
依頼+案内+店	25	32.00%	32.00%
陳述+提示+意思内容+了解	24	37.50%	50.00%

タスクの最後に余分な応答をしてしまう問題が改善されたことによる。

テストセット中の出現頻度上位 7 種類の発話意図について、発話生成の評価結果を表 3 に示す。

表から、発話意図によって正解率に大きな違いが出てくることが確認できた。対話のはじめに出現する「依頼+検索+店」を入力発話意図としてもつ発話に対しては高い正解率が得られた。しかし、対話の終わりのほうに出現する「依頼+案内+店」の発話は、生成したシステムの応答発話意図が必ずしもテストセットのものとは一致していないため、途中で対話が破綻し失敗することが多かった。また、「はい」といった「陳述+提示+意思内容+肯定」や「陳述+提示+意思内容+了解」の発話に対しても正解率は低くなった。これは、表層的には両者の区別がつかないことや、その発話自体に特別な情報を含まないことから、次に発話すべき表現が見つからないためであると考えられる。これに関しては、事例を増やすことによってある程度改善されることを確認した。

5 おわりに

本稿では、コーパスベースの音声対話システムの構築について述べた。本手法によって構築したシステムで、100 タスク 483 発話のドライバ発話に対して評価実験を行った結果、扱うには困難な入力発話の種類が存在するというコーパスベース音声対話システムが持つ特有の傾向が明らかになった。また、利用する事例の追加による性能向上を確認した。

なお、テストセットを用いた評価実験では、途中で対話が破綻するという現象があったため、今後は、被験者を用いた評価実験を行う予定である。

参考文献

- [1] 河口, 松原, 山口, 武田, 板倉: CIAIR 実走行車内音声データベース, 情処研報, SIG-SLP-49, pp.139-144 (2003).
- [2] 入江, 松原, 河口, 山口, 稲垣: 意図タグつきコーパスを用いた発話意図推定手法, 人工知能学会研資, SIG-SLUD-A301-03, pp.7-12 (2003).
- [3] S.Young: Talking to machines (statistically speaking), Proc. of ICSLP-2002, pp.9-16 (2002).
- [4] Chasen, <http://chasen.aist-nara.ac.jp/>
- [5] MySQL, <http://www.mysql.com/>