クラウドロボティクス基盤 rospeex の長期実証実験と大規模ロボット対話データの解析

○杉浦孔明，是津耕司（情報通信研究機構）

1. はじめに

スマートフォンを始めとする種々のデバイスに音声インタフェースが導入され，広く一般に認知されるようになってきた [1,2]．また，音声認識や対話機能を利用可能な種々のクラウドサービスも存在している（例えば，[3,4]）．音声認識および音声合成機能をクラウド化することで，音響モデルや言語モデルなどの大規模な資源をロボット上に搭載する必要がなくなり，ハードウェアを簡略化することでコストを低減できる．一方，クラウドサービスの提供者にとっては，サービスのログとして収集した大規模データを用いた性能向上など新たな研究テーマを創出できるというメリットがある [5]．

コミュニケーションロボットの開発において，開発者は既存の音声検索等のクラウドサービスを用いることもできる．しかし，ひとたび非ロボティクス用途のクラウドサービスにロボティクス用途の発話が集積されると，その発話ログは他のログと混じってしまい，ロボティクスに特化したコーパスを抽出することはコストの面から現実的ではない[1]．よって，ロボティクスに特化した音声言語処理の性能向上を目指すためには，ロボット向けクラウドサービスを構築することが合理的であると考えられる．

しかし，クラウドロボティクス基盤の構築においては，音声認識・合成・対話についての基礎技術から，基盤の安定性・スケーラビリティに至るまでの包括的な基盤構築が必要とされ，簡単な課題ではない．また，ロボットのユースケースに対する深い知識を持つことも重要である．

このような問題背景から，我々はクラウドロボティクス基盤 “rospeex” を構築・公開し，サービスを長期間にわたり実運用している．クラウド型音声認識・合成サービスを通じて，NICT で開発されたエンジン [2] をユーザは利用可能である．また，他のクラウド型音声認識・合成サービスに切り替えて利用することも可能である．現在，4 か国語（日英中韓）の音声認識・合成に対応しており，学術研究目的に限り無償・登録不要で公開している．rospeex は `http://rospeex.org` からダウンロード可能である．

クラウドロボティクスやクラウドネットワークロボティクスなどの分野では，物体認識，知識共有，機械学習などのためにクラウドコンピューティングを用いるアプローチが提案されている（例えば [6]）．本研究はこれらと関連するが，ロボットの音声コミュニケーションに主眼を置く点が異なる．また，HARK [7] などミドルウェアに対応した音声コミュニケーションツールでは，内部的にスタンドアロン型のエンジンを用いている．これらのエンジンは機能的には複数言語の音声認識・合成が可能であるが，言語モデルの入れ替えなどをロボット開発者自身が行う必要がある．一方，提案手法では，次節で説明するように言語やボイスフォントの変更を簡単に行うことができる．

本研究の独自性は以下の 2 点である．

- ロボット向けクラウドサービスを構築・公開し，長期間の実証実験を行った．
- 大規模ユーザのログを対象として，音声認識・合成のドメイン依存性と実用性について解析した．

2. クラウドロボティクス基盤 rospeex

本節では，rospeex におけるサーバとクライアントモジュールの構成の概要について述べる．機能の詳細については，[8] を参照されたい．図 1 および図 2 に，それぞれ rospeex を用いた音声対話の例および標準的な構成を示す．

以下，本論文では，rospeex の 2 種類の使用者を区別するため，以下のように定義する．

- ユーザ: rospeex を使用する開発者．
- 話者: ロボットとの対話者．厳密には話し手と聞き手になる場合があるが，説明の都合上，話者と表記することにする．本論文では，参与構造 [9] の解析が必要な多人数会話を扱わない．

2.1 クラウド型音声認識・合成

rospeex は複数のクラウド型音声サービスに接続可能であり，それらを切り替えて使用できる．本節では，

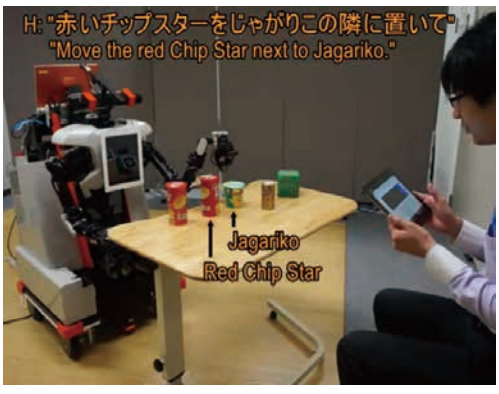

図 1 rospeex を用いたサービスロボットとの対話例．rospeex は多言語の大語彙連続音声認識機能を提供している．つまり，「チップスター」や「じゃがりこ」などの固有名詞を開発者が文法に登録する必要はない．

[1] この問題はクラウドロボティクスにおける普遍的な問題であり，画像など音声以外のクラウドサービスについても当てはまる．

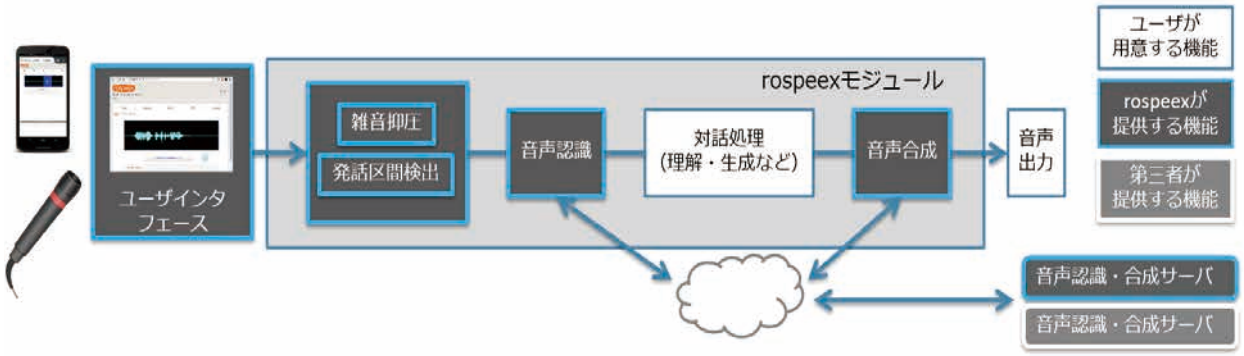

図 2　rospeex におけるサーバとクライアントモジュールの標準的構成

NICT が提供する音声認識・合成サービスについて説明する．

ロボットとの音声対話において特徴的な難しさは，push-to-talk 方式を前提とできないことである．スマートフォン型音声インタフェースなどでは，音声の入力前にボタン等のインタフェース（「OK，XXX」のように音声の場合もある）を用いることが多い．これに対し，例外はあるものの，ロボットではそのようなインタフェースを前提とすることはできない．したがって，発話区間検出（Voice Activity Detection; VAD）が重要になる．さらに，マイクと話者の距離が大きいことが多いため，雑音抑圧が必要である．ロボカップ@ホームのような高騒音環境では，雑音抑圧を使用しなければ正確な発話区間検出はほぼ不可能である．

rospeex では，雑音抑圧と発話区間検出はネットワーク上サーバで行わない設計としている．これらをサーバで処理するとネットワーク由来の遅延によりリアルタイム性の確保が難しくなるためである．また，一般的に発話区間検出の精度はそれほど高くないため，後段の処理でロボット名を含む発話のみ受け付けるなどの工夫が必要である．

ロボット対話における自然性を向上させるため，クラウドサービスへの通信時間を短縮することは重要な課題である．そのため，音声認識において分割送信方式に対応している．分割送信とは，VAD 終了後に一括で送信するのではなく，音声入力中に前もって送信する方法を指す．分割送信自体は新しい技術ではなく，ロボット用途以外のクラウドサービスでは一般的に用いられている．分割数を多くすれば処理時間を減少させられるものの，多くしすぎても性能は頭打ちになる．本実験では，サーバの負荷とのバランスから 3.52kB を単位として分割を行った．ATR503 文であれば，平均的に 50 ほどに分割されサーバに送信される．

一括送信でも分割送信でもサーバ上での音声の処理時間には大きく違いがあるものではない．これまでの実験により RTF=0.7 程度であるので，仮に 5 秒程度の発話であれば，双方とも 3.5 秒程度の時間が認識に必要である．双方とも同じ文を用いているので，送信するバイト数に大きな違いはない．違いは VAD 終了後に一括で送信するか，音声入力中に前もって送信するかの違いである．

日本語の音声合成としては非モノローグ HMM 音声合成 [10] に対応し，ロボットとの対話に特化して開発

図 3 PC およびスマートフォンから利用可能なユーザインタフェース．下部には音声認識結果や音声合成リクエストが表示される．

されたボイスフォントを利用可能である．ロボットのコミュニケーション機能の開発においては自然な音声合成が求められているが，一般的な音声合成器は人-ロボット対話に最適化されている訳ではない．そのため，ロボットの合成音声は自然さや親しみやすさに欠け，抑揚が適切でないため話者が質問されたことに気づかないなどの事態も起こり得る．我々はサービスロボットタスクとの対話タスクにおいて非モノローグ HMM 音声合成の評価を行い，理論的上限にせまる MOS 値を持つことを示している．性能評価の詳細については，[10] を参照されたい．

なお，ROS を経由せずに音声認識または合成単体としても利用可能であり，その場合は JSON ファイルをインタフェースとする．ユーザが用いるプログラミング言語には依存しないため，C++や Python など各種のプログラミング言語を利用可能である．

2.2　マルチプラットフォームユーザインタフェース

図 3 に rospeex のユーザインタフェースを示す．図中の青で示す部分は発話区間として検出された部分である．このように話者に波形をリアルタイムにフィードバックすることで，声が小さいなどの問題を話者自身が気づくことができる．一方，波形表示が提示されない場合，ロボットの反応から誤りの原因（発話区間検出の失敗か音声認識の誤認識か，など）を推定することは難しい．現時点では，rospeex のユーザは開発者であることが多いため，このようなインタフェースが有

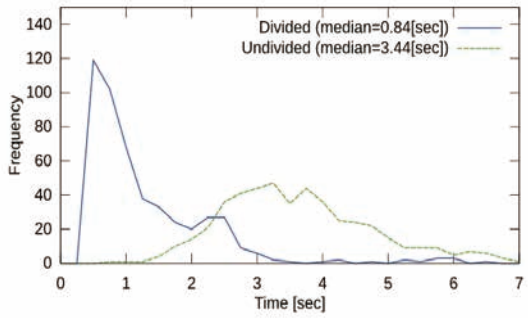

図 4　レスポンス時間のヒストグラム（音声認識）

効であると考えられる．図のインタフェースでは，必要ない場合に発話区間が検出されないよう，発話区間検出機能を無効にすることも可能である．

rospeex のユーザインタフェースはブラウザ上で提供されており，マルチプラットフォームに対応している．そのため，PC のオンボードマイクや USB 接続の指向性マイクロフォン，さらにスマートフォンを入力デバイスとして用いることができる．対応ブラウザは，Google Chrome（Windows, Linux）および Firefox（Windows, Linux, Android）である．ただし，Android を OS とするハードウェアは無数に存在するため，全ての機種で動作を保証するものではない．

3.　実証実験および考察

3.1　実験設定

本節では，実利用におけるロボットとの音声対話について，通信速度の解析およびサーバ上のログの解析を行う．実験に用いたサーバの CPU は Intel 製 X5690（12 コア，3.47GHz），メモリは 200GB であった．

3.2　音声認識における分割送信の解析

本実験の目的は，代表的な使用環境における分割送信と一括送信の時間差の調査である．代表的な使用環境を模擬するため，無線 LAN 環境を用い，rospeex サーバとは異なるネットワーク上からサービスを使用した．なお，本実験ではあらゆる帯域および使用状況におけるクラウドサーバとの通信速度を計測することを目的としていない．

コーパスとして ATR503 文を用い，声優 1 名による収録を行った．そのうち発話区間検出に問題がない 495 文についてレスポンス時間を求めた．レスポンス時間とは，クラウドサービスからのレスポンスが得られた時刻から VAD 終了時刻を引いたものである．

実験の結果を図 4 に示す．一括送信でのレスポンス時間の中央値は 3.44 秒であったが，分割送信では 0.84 秒に短縮されている．我々の経験では，ロボットとのインタラクションにおいて 1 秒以内にレスポンスを返せない場合，話者が再度発話を行ってしまい，それが誤認識されるなど，望ましくない挙動に陥りやすい．本実験より，分割送信によって待ち時間を減らせることが示唆された．

3.3　音声認識ログの分析

2014/1/1 から 2014/11/28 までのアクセス記録をもとに，実際の利用における音声認識ログを解析した．本論文では rospeex の適用対象をホームロボットであると仮定し，ホームロボットに関連したカテゴリに発話を分類する．ログに含まれる発話の音声認識結果の総数は，44960 であった．ただし，無音など明らかに発話が含まれないものは取り除いた．このうち，頻度が 3 以上のものを分析対象として発話を分類した結果を表 1 に示す．ただし，本実験ではカテゴリを以下のように定義する．

1. 挨拶・雑談: 日常会話
   例：こんにちは，君は誰，バイバイ
2. 1 問 1 答型質問: 対話履歴を必要としない情報源への問い合わせ
   例：今何時，今日の予定を教えて，天気を教えて
3. 行動指示（移動・把持）: 移動や把持に関連する動作指示発話
   止まれ，右に折れて，本棚まで行って
4. 行動指示（家電操作）: 音声リモコンのように家電を操作する発話
   例：テレビを消して，電気をつけて
5. 行動指示（認識・学習）: センサ入力の学習または認識を指示する発話
   例：ここはどこ，あれ見て
6. 行動指示（その他）: 3-5 以外でロボットの行動を指示する発話
   例：手を上げろ，終わり
7. その他（検索・回答，判別不能）: 1-6 以外の発話．主に，質問への応答，音声認識誤りまたは判別不能な発話を含む．
   例：富士山，ちょっと

表 1 より，挨拶や雑談に比べ行動指示発話が少ないことがわかる．これは，主なユーザが開発者であり，ロボットに未実装の機能を指示しないためであると考えられる．また，約半数の発話は挨拶・雑談や 1 問 1 答型の質問である．これらの発話に対しては，一般的に提供されている質問応答や雑談対話のクラウド型 API を用いることが有効であると考えられる．

一方，行動指示はロボットごとに機能を実装する必要があり，一般的に解決は簡単ではない．音声認識誤りのため,「その他」カテゴリに分類された発話も多いと考えられるため，音声認識精度の向上は今後の課題である．さらに，対話履歴の解析を行うとともに，行動にグラウンドした対話管理が必要になるであろう．

3.4　音声合成に関する個人依存性

本実験では，音声合成におけるユーザごとのテキストの多様性を調べることが目的である．本システムではキャッシュ機能を組み込んでいないが，多様性が少な

表 1　発話の分類

カテゴリ	発話数	割合 [%]
挨拶・雑談	1894	31.70
1 問 1 答型質問	1153	19.30
行動指示（移動・把持）	258	4.31
行動指示（家電操作）	229	3.83
行動指示（認識・学習）	215	3.59
行動指示（その他）	41	0.68
その他（検索・回答，判別不能）	2205	36.91
合計	5973	100

ければ，ローカルにキャッシュを持つことで体感的に応答速度を改善できる可能性がある．また，実ユーザの文の多様性を調査することは，ロボティクス分野における知識としても重要である．これまでロボティクスにおいてはスタンドアロンに閉じた開発がなされてきたため，ロボット開発者が求める音声処理機能の性質について，多様かつ多人数のユーザを扱った定量的な解析が困難であった．クラウドロボティクス基盤を構築することで，ロボット対話開発における個人依存性（およびドメイン依存性）を調査することが初めて可能になる．

いま，あるユーザ u_i の各音声合成リクエストを文とし，u_i の音声合成リクエスト履歴全体を文集合 S_i とする．音声合成では，S_i の結果を全てキャッシュしておき，完全一致する文については，すでにキャッシュした文を再生することでクラウドサービスへの通信時間を省略することができる．

S_i に含まれる文の種類を N_{uniq} とすると，S_i 中でキャッシュを利用できるリクエスト数 N_c は，$N_c = ||S_i|| - N_{uniq}$ と表すことができる．ここで，擬似キャッシュヒット率 r_p を以下のように定義する．

$$r_p = \frac{N_c}{||S_i||} \tag{1}$$

r_p が高いユーザは，同じ言い回しを繰り返し使っていると考えられる．

2014/1/1 から 2015/5/31 までの rospeex のユーザのうち，10 回以上合成リクエストを行ったユーザを抽出した．本論文では，同一 IP アドレスを有するアクセスを 1 ユーザと定義する．技術的には各マシンごとにカウントすることも可能であるが，同じマシンを複数人で使用する場合など，いずれにせよユーザ数の完全な把握は困難である．上記のユーザのうち，明らかに自動テストと考えられるアクセスを除外し，残りの 88 ユーザについて解析を行った．

そのうち，上位 15 ユーザについて解析した結果を図 5 に示す．図において，(a) は文の種類 N_{uniq}，(b) は N_c を表す．N_{uniq} は N_c に比べて小さく，上位ユーザは 100〜200 種類の文を繰り返し使っていることがわかる．

上位 88 ユーザについて同様の調査を行ったところ，r_p のユーザ平均は $\bar{r}_p = 0.504$ であった．つまり，ユーザのリクエストのおよそ 50.4%はそれまでの履歴に含まれる文である．このことは，キャッシュを持つことで，クラウドサービスに接続すべきリクエストを半分に削減できることを示唆している．

キャッシュを利用する際に検討すべき課題はストレージ容量であるが，平均的なロボット対話において問題となることは少ないと考えられる．図 5 より上位 88 ユーザのうち，もっとも大きい N_{uniq} でも 200 を超えることはなかった．多くのロボット対話では 1 発話のファイル容量が 100kB 程度であることを考えると，20MB 程度の領域があれば解決可能である．通常の環境ではこれは問題とならず，メリットが勝ると考えられる．

4. おわりに

ロボット対話の開発においてはスタンドアロンの音声処理を用いるものが多かったため，ロボット開発者

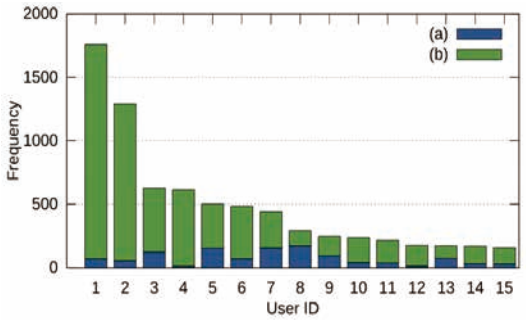

図 5 上位 15 ユーザの合成リクエスト内訳．(a) ユニーク文数 N_{uniq}，(b)2 度目以降の出現 N_c．上位 88 ユーザについて平均 $\bar{r}_p = 0.504$ であった．

が求める音声処理機能の性質について，多様かつ多人数のユーザを扱った定量的な解析が困難であった．我々はクラウドロボティクス基盤 rospeex を構築・公開し，サービスを長期間にわたり運用してきた．本研究では，大規模ユーザのログを対象として，音声認識・合成のドメイン依存性と実用性について解析した．音声以外に対してもサービスロボット向けにクラウドコンピューティングの導入が進みつつあり，今後は蓄積されたデータの利活用が期待される．

本研究に関する動画は `http://rospeex.org/` を参照されたい．

謝辞

本研究の一部は，立石科学技術振興財団研究助成および JSPS 科研費 15K16074 の助成を受けて実施されたものである．

参 考 文 献

[1] 河原達也，“音声対話システムの進化と淘汰–歴史と最近の技術動向–,” 人工知能学会誌，vol.28，no.1，pp.45–51，2013.

[2] 松田繁樹，林輝昭，葦苅豊，志賀芳則，柏岡秀紀，安田圭志，大熊英男，内山将夫，隅田英一郎，河井恒，中村哲，“多言語音声翻訳システム“ VoiceTra ”の構築と実運用による大規模実証実験,” 電子情報通信学会論文誌，vol.J96-D，no.10，pp.2549–2561，2013.

[3] K. Haverlock and S. Sudarsan, “Creating cognitive applications powered by ibm watson: Getting started with the api,” Technical report, IBM, 2013.

[4] K. Tsujino, Y. Nakashima, S. Iizuka, and Y. Isoda, “Speech recognition and spoken language understanding for mobile personal assistants: A case study of “shabette concier”,” Proc. Workshop on Field Speech and Data Management, pp.225–228, 2013.

[5] S. Matsuda, X. Hu, Y. Shiga, H. Kashioka, C. Hori, K. Yasuda, H. Okuma, M. Uchiyama, E. Sumita, H. Kawai, and S. Nakamura, “Multilingual Speech-to-Speech Translation System ”VoiceTra”,” Proc. Workshop on Field Speech and Mobile Data, pp.229–233, 2013.

[6] K. Kamei, S. Nishio, N. Hagita, and M. Sato, “Cloud Networked Robotics,” Network, IEEE, vol.26, no.3, pp.28–34, 2012.

[7] K. Nakadai, T. Takahashi, H.G. Okuno, H. Nakajima, Y. Hasegawa, and H. Tsujino, “Design and Implementation of Robot Audition System’HARK’—Open Source Software for Listening to Three Simultaneous Speakers,” Advanced Robotics，vol.24，no.5-6，pp.739–761，2010.

[8] 杉浦孔明，堀　智織，是津耕司，“音声対話向けクラウドロボティクス基盤 rospeex の構築と長期実証実験,” 第 32 回ロボット学会学術講演会資料，pp.RSJ2014AC3I2–05，2014.

[9] D. Traum, “Issues in Multiparty Dialogues,” Advances in Agent Communication, pp.201–211, Springer, 2004.

[10] K. Sugiura, Y. Shiga, H. Kawai, T. Misu, and C. Hori, “Non-Monologue HMM-Based Speech Synthesis for Service Robots: A Cloud Robotics Approach,” Proc. ICRA, pp.2237–2242, 2014.