解 説

Watson とロボットの音声対話機能

Watson and Spoken Interaction with Robots

立 花 隆 輝* *日本アイ・ビー・エム株式会社 東京基礎研究所
Ryuki Tachibana* *IBM Research—Tokyo

1. は じ め に

人工知能への期待感がビジネスにおいてもコンシューマにおいても非常に高い。長年解けなかった難問を人工知能が解くのではないかと、従来のロボットでは自動化できなかった作業を自動化するのではないかとといった期待を耳にすることが多い。今日の技術ではまだ解けない問題も残されている一方、クラウドや深層学習を背景としてブレイクスルーが得られた問題もある。IBM Watson もこのブームの契機の一つだったと言ってもよいだろう。本論文では音声対話ロボットに関連する部分を中心に、比較的新しい Watson の機能について紹介し、ロボットの認知機能の部品としてどのように利用できるかを解説する。特に 2016 年 11 月に発表されたプロジェクト Intu について紹介する。さらに、Watson を用いて音声対話ロボットを構成する際に考慮すべき点や、研究の必要がある点を明らかにする。

2. Watson

2.1 概要

当初 Watson は自然言語で記述された知識を幅広い事柄に関して蓄え、それに基づき、自然言語で記述された質問に回答するシステムとして開発された。特にアメリカのクイズ番組で人間の最高レベルの選手と競いその能力を証明することが開発の一つの目標であった[15]。その後 IBM は Watson の認知能力を様々な分野に拡張し、現在は自然言語だけでなく画像処理、音声認識、機械学習などの技術も含まれている。提供形態としてはクラウド上で Software as a Service (SaaS) 形式で提供しているサービスも、オンプレミスに提供しているソフトウェアもある。前者のための Platform as a Service (PaaS) クラウド・プラットフォームは Watson Developer Cloud (WDC) といい、Bluemix™ というシステムを利用しており、そこでは IBM やサードパーティーが様々なサービスを提供している。その一部が

Watson Cognitive Services (WCS) と呼ばれる言語処理や機械学習の一連のサービス群であり、音声対話ロボットに利用可能なサービスも多くがそこに含まれている。WCS と外部の通信は、決められた REST API や WebSocket API などによって行われる。この API で通信することが可能でさえあれば、PC はもちろんのこと、ロボットや組み込み機器やスマートフォンも Bluemix と通信してそのアプリケーションを利用することができる。

WCS の例としては Speech To Text (音声認識), Text To Speech (音声合成), Conversation (対話), Natural Language Classifier (テキスト分類), Retrieve & Rank (テキスト検索), Personality Insights (性格判断) などがある [7]。いくつかのサービスはその実現に深層学習 (Deep Learning) を駆使している。すでに日本語対応をしている自然言語関連サービスも多く、上に挙げたサービスはその一部である。挙げた以外にも多様なサービスがありそれぞれがビジネスの現場で様々な用途に実用されている。応用事例には例えばコールセンターでのリアルタイム・オペレータ支援システムがある。このシステムはオペレータと顧客の電話での会話音声で Speech To Text を用いてリアルタイムで音声認識し、オンライン・マニュアルや FAQ 文書のデータベースから、進行中の会話に関連するドキュメントを自動的に検索してオペレータの画面に表示することで、マニュアルを手作業で開く手間を省力化し、顧客対応を効率化・品質向上している。

WCS はそれぞれ API を開発者に公開しているので、個別の部品として開発者が使用することができる。多くのサービスが、開発者がアップロードするデータによって学習を行いそのデータに適応 (特化) できるという特徴を持っている。次節では特に音声対話ロボットの開発によく利用される Speech To Text と Conversation を例にして解説する。

2.2 Speech To Text (音声認識)

Speech To Text (以下 STT) は、アップロードされた音声の認識結果を返す不特定話者大語彙連続音声認識サービスである [5]。本稿執筆時点では日本語を含む合計 8ヶ国語をサポートしており、汎用の広帯域モデルと電話音声用の狭帯域モデルを提供している。モデルには音響モデル、

原稿受付 2017 年 1 月 4 日

キーワード: Human-robot Interaction, Spoken Dialog System, Watson, Artificial Intelligence

*〒 103-8510 中央区日本橋箱崎町 19-21

*Chuo-ku, Tokyo

図1 STTの利用の流れ

言語モデル、辞書が一体となって含まれている。日本語の場合、言語モデルおよび辞書は広範囲の入力文に対応できるように設計しており、ウェブ検索のための短いフレーズから長文や会話音声の認識も対象にして学習データを構成してある。また認識処理がリアルタイムで動作するようにチューニングを行っており、デモページ[6]でPCのマイクから音声を入力していくと、認識結果が発話の終了を待たずに先頭部分から逐次的に得られることがここで確認できる。APIの呼び出しにオプションを与えれば信頼度や対立仮説を出力させることも可能である。開発者はアプリケーションからAPIを直接利用することももちろん可能であるが、Swift[2]、Java[3]やPython[4]のなどの言語について公開されている開発者キット（SDK）を利用することも可能である。SDKを使用する利点は、例えばSDKを使わない場合には複数回のAPI呼び出しを正しい順番で行うプログラムを直接書かなければならないところを、一度のローカル関数呼び出しで同等の処理ができることである。2016年12月からは日本語についてもCustomization機能がβ公開された。これを利用すれば単語への辞書登録および、自分のコーパス（例文）を用いた言語モデルの追加学習ができ、例文に類似した文を認識しやすくし、また逆に関係ないような認識誤りを起きにくくできるので、ユーザーにとっての利便性が高まる。STTの利用の流れを簡単に図示したのが図1である。ほかのWCSも大まかには共通である。

2.3 Conversation（対話）

Conversation サービスは自然言語を用いた対話システムを作る際に中核となる対話エンジンであり、入力された自然言語テキストを解析する[1]。解析の結果として入力文の意図（インテント）の判別と、入力文中で言及された事物（エンティティ）の抽出を行って出力する。対話の状態はツリーで管理される。状態（ノード）の遷移はインテント、エンティティ、開発者が定義し設定した状態変数などの条件によって決められる。シンプルな対話アプリケーションでConversationを使う場合には、あらかじめノードに

紐づけておいた応答文をConversationから取り出し、これを対話アプリケーションの出力とする。そうではなく開発者がデータベースやキャッシングシステムなどのバックエンドシステムのフロントエンドとしてこの対話システムを利用したいならば、入力文を解析して得られたインテントやエンティティをそれらのバックエンドシステムのクエリに利用するのがよい。Conversationサービスはブラウザ上でGUIを持ったツールを提供しており、開発者はインテント、エンティティ、対話フローの設計をブラウザ上でインタラクティブに行うこともできる。

Conversationが主にターゲットとする対話は目的のない雑談ではなく、タスクを持った対話である。タスクを持った対話システムにおいては、ユーザーの意図や操作したい事物を正確に特定して、それに対して適切なアクションをとらなければならないが、それには手作業による対話フローの作り込みがある程度は避けられない。その作業効率が結果的に対話システムの完成度を左右するため、Conversationには効率化の仕組みがある。インテントの判別は、開発者が用意した例文に基づいて機械学習されたモデルを用いてなされる。これにより、想定した例文と正確に一致してはいない入力文であっても、ある程度の多様な文について同じインテントを持つと分類できる。エンティティについては同義語を定義できるだけでなく、ドメインに依存せず一般的に対話で頻繁に用いられる数値表現、金額表現、時刻表現などについてシステムエンティティを用意してある。開発者は候補の中から自分の対話システムで利用するシステムエンティティを選択して使用することができる。

3. ロボットの認知機能と Watson

前章で述べたように、自然言語の質問応答システムとして研究開発が始まったWatsonは、開発者が個別に取捨選択して使用できるような、画像や音声も含めた認知機能サービス群WCSへと拡張された。WCSの種類が揃ってくると、ロボットを一種のアプリケーションと捉えWCSをロボットの高次認知機能に利用しようという発想が自然な帰結として生まれ、IBM内外で様々な試みが行われるようになった。例えばイサナドットネットは、ソフトバンクロボティクス社のロボット（Pepper）とWatsonを連携するソフトウェアの開発を2016年7月に発表している[14]。独自のサービスロボットを開発しているタケロボは、ロボットの自然言語対話でのコミュニケーションを可能にして利便性を向上させるためにWatsonとの連携を同7月に発表している[16]。

Watsonとロボットの連携は、Watsonとロボットの双方の開発者にとって利点が多い。ロボット開発者にとってはクラウドの計算能力を活用する高次認知機能をすぐに利用できるということが利点になる。様々なロボットと様々な

ロボット以外のアプリケーションが、クラウドサービスの枠組みを共有し計算機資源を共通化して利用することになるので、個々のロボット向けに認知機能を作り込むのに比べて計算機資源の点でも研究開発の点でも効率が良い。一般ユーザにとっても、高機能なロボットを身近に利用できるようになる利点や、ロボットという目につきやすいアイコンがあるので Watson との対話を始めやすいという利点がある。

3.1 音声対話ロボットでの Watson の利用

高次の認知機能にのみ WCS を利用し、ロボットのハードウェアとその制御にはロボット開発者がすでに使用している仕組みを利用するというのが、ロボット開発者にとってまず取り組みやすい基本的な構成であろう。その場合の最も基本的な処理の流れは、次のようになる。(1) まず、マイクから取得した音声信号を STT に流し込み音声認識結果を得る。この際に STT の WebSocket インタフェースとストリーミング音声認識機能を用いることで、ユーザの発話が終わってから最終的な音声認識結果が得られるまでの遅延時間を短縮できる。さらに interim_results オプションを使うことで発話中の中間的な音声認識結果を使ってインタラクティブ性を高めることも可能である。(2) 次に音声認識結果を Conversation に送信し応答文を取得する。Conversation では対話ログが記録されるので、開発者は改善作業を繰り返し行いやすい。(3) 最後に応答文を Text To Speech (TTS) に送信し合成音声を得て再生する。日本語ではまだサポートされていないが他言語では timings オプションを使えば、単語ごとの時間位置を得ることができる。これを用いればロボットの表情やジェスチャーの動きと合成音声を同期できる。

Conversation 内でノードに紐づけたアウトプットとして用意しておくのは、音声合成用の応答文だけでなく、ロボットのアクションを指定するコマンドでもよい。例えばユーザが挨拶の言葉を言ったときに握手のために手を差し出すようなアクションをロボットにさせたいならば、挨拶の言葉で学習した『挨拶インテント』のアウトプットとして、手を差し出すジェスチャーを紐づけておき、それをトリガーにしてジェスチャーを呼び出せばよい。

このほかにロボットでの利用がしやすい WCS としては次のものがある。ロボットで取得したカメラ画像を Visual Recognition (画像認識) で分析することで、認識した物体の種類による反応をロボットにとらせたり、ユーザの年齢を推定した結果によって対応を変えさせたりすることができる。Personality Insights (性格判断) はテキストからそれを書いた人物の性格を判断できるので、ユーザの会話ログや SNS での過去の発言がある程度の分量利用可能であれば、判断した性格によってロボットの対応を変えることができる。

図2 音声対話ロボットでの Watson の利用

3.2 音声対話ロボットの技術課題

音声対話ロボットで重要な点の一つは、音声認識の頑健性である。特に、ロボット自身が再生する合成音声や、ユーザや周囲の人間の、ロボットに向けられていない発話などはロボットが反応すべきでない妨害音声であるが、基本的に STT そのものには対象音声と区別する方法がないため、誤動作の原因になりやすい。もう一つの重要な課題は音声応答の遅延時間である。前述のような順序的な処理を用いると、それぞれのステップに要する処理時間の加算が全体の遅延時間となるのは避けられない。データをクラウドとの間で送受信する通信時間もわずかながら遅延となる。このような阻害要因もある中で、ロボットとユーザの自然で速やかな音声対話を実現するにはロボットの組み込み処理系(エッジ)での反射的処理と、クラウド側での高次の認知機能との間での適切な処理の分担が重要である。

こういった課題に対応する例えば一つの方法として IBM 東京基礎研究所で研究開発を行っているのがマイクロフォンアレイを用いた beamformer である。ユーザの位置を特定の方向にあらかじめ想定しておき、その方向に音声を検出されたら即時にロボットの合成音声再生を停止するバザイン機能を実現したり [17]、その方向の音声のみを強調し他方向からの音声を抑圧したりする処理 [13] をエッジ側の処理として行い、STT には音声強調処理後の 1 チャンネル音声のみを送信する。もちろん STT はクラウドの計算能力を用いているので、エッジで音声認識までするのに比べて高精度な大語彙音声認識が可能になっている。このようにエッジ処理とクラウド処理を組み合わせることで、ユーザの発話のみに頑健に素早くロボットが対応できる。

3.3 Project Intu

3.1 節で処理の流れを説明したようにロボット含めユーザとインタラクションするアプリケーションや少し複雑なアプリケーションを構成するには複数の WCS を接続し連携させる必要がしばしば生じる。また、同一種の WCS であっても複数のインスタンスを並列して使う必要が生じ

ることがある。例えば日本語と英語の STT インスタンスを用いて多言語に対応する場合である。そのような状況においては、複数の WCS インスタンス間の連携をするためのミドルウェアが必要になる。この連携にはある程度決まったパターンがあるため、アプリケーションごとに新規に開発するのではなく枠組みを再利用するべきである。ミドルウェアを置く位置としてはクラウド上に置く選択肢と、エッジ側に置く選択肢が一般にはありうる。特にロボットの場合においては、前述の beamformer もそうだが、センサやアクチュエータなど組み込み側処理との連携も WCS 間の連携と同時に必要になるため、ミドルウェア自体はエッジ側に必要になる。

このようなニーズに応えることを一つの目的として IBM は 2016 年 11 月に Intu (イントゥー) というプロジェクトを発表した [9]。Intu はデバイスとしてロボット、ディスプレイ、部屋、Raspberry Pi のような組み込み機器、スマートフォン、PC など広い範囲を対象としている。このプロジェクトでは Self と呼ばれるアーキテクチャを開発者に試験的提供し始めた [8]。基本的に Self のターゲットはロボット OS よりもハイレベルなレイヤーにある。すでに公開されている SDK を用いれば開発者は、自身が開発しているハードウェアプラットフォームに対して、本稿でこれまで述べたような認知機能を容易に付加することができる。プロジェクト Intu では WCS だけでなく、サードパーティー製のソフトウェア・コンポーネントも Self 上で連携ができるように開発を呼びかけている。Self アーキテクチャは Minsky の心の社会 [12] の理論や Brooks [11] のサブサンブション・アーキテクチャを部分的に参考にして設計されている [10]。ただし、クラウド上の WCS やシンボルレベルの認知機能を前提にしている点などで、それらの参考理論と大きく違うところもある。今後より多くのソフトウェアコンポーネントがエッジとクラウドの双方に追加されていけば、より高度で低遅延な自律性をロボットに持たせることができるだろう。

4. ま と め

音声対話ロボットは、ビジネスにおいてはまだ実用が広がっているとは言いがたいと筆者は考える。それにはいくつかの理由が考えられるがその一つには、擬人のジレンマがあると筆者は思う。物理的な形状にしてもソフトウェア上でユーザに与える印象にしても、擬人的なインタフェースを備えると、ユーザは、これは言葉を理解し、知識を持ち、更新し続ける strong AI であるという過度に高い期待を容易に持ってしまう。一方で擬人的なインタフェースを提供することはユーザと授受する情報量を豊富にするためには鍵であり、避けては通れない。実用が進まないもう一つの

理由は、音声対話ロボットを構成する様々な技術要素の誤りが積算されシステム全体として対話を破綻させないことが非常に難しいということがある。ロボットや音声対話について進んだ研究がすでに数限りなくあるので、現時点で現実的に有用なアプリケーションを適切に選び、確かな価値をユーザに提供するのが音声対話ロボットの実用を広げる上で重要であろう。

参 考 文 献

- [1] Conversation, <https://www.ibm.com/watson/developercloud/conversation.html>
- [2] Github: Watson developer cloud ios sdk, <https://github.com/watson-developer-cloud/ios-sdk>
- [3] Github: Watson developer cloud java sdk, <https://github.com/watson-developer-cloud/java-sdk>
- [4] Github: Watson developer cloud python sdk, <https://github.com/watson-developer-cloud/python-sdk>
- [5] Speech to text, <https://www.ibm.com/watson/developercloud/speech-to-text.html>
- [6] Speech to text demo page, <https://speech-to-text-demo.mybluemix.net/>
- [7] Watson services, <http://www.ibm.com/watson/developercloud/services-catalog.html>
- [8] Github: Project intu, <https://github.com/watson-intu>, Nov. 2016.
- [9] Project intu, <https://www.ibm.com/watson/developercloud/project-intu.html>, Nov. 2016.
- [10] G. Booch: Grady booch discusses project intu, <https://youtu.be/ADnCUgrEt0U>, Nov. 2016.
- [11] R.A. Brooks: "Intelligence without representation," Artificial Intelligence, vol.47, pp.139-159, 1991.
- [12] Marvin Minsky: 心の社会。産業図書, 1990.
- [13] R. Tachibana, O. Ichikawa and T. Fukuda: "Effective speech suppression using a two-channel microphone array for privacy protection in face-to-face sales monitoring," Acoustical Science and Technology, vol.36, no.6, pp.507-515, 2015.
- [14] イサナドットネット株式会社: イサナドットネット, ソフトバンク株式会社と ibm watson エコシステムパートナー契約を締結しテクノロジーパートナーに選定, http://isana.net/news/ibm_watson/, July 2016.
- [15] スティーヴン・ベイカー: IBM 奇跡の「ワトソン」プロジェクト: 人工知能はクイズ王の夢をみる。早川書房, 2011.
- [16] タケロボ株式会社: Ibm watson エコシステムプログラムに参加に関するお知らせ, <http://www.takerobo.co.jp/news/news/2016/1188032.1822.html>, July 2016.
- [17] 清水周一, 井上忠宣, 高野光司: "コグニティブ・ロボット最前線 ibm 東京基礎研究所による「賢い」ロボットへの挑戦", ProVISION, no.90, pp.30-32, 2016.

立花隆輝 (Ryuki Tachibana)

1996 東京大学工学部航空宇宙工学科卒業。
1998 同大学大学院工学系研究科航空宇宙工学専攻修士課程修了。同年日本アイ・ビー・エム (株) 入社。以来、同社東京基礎研究所にて音楽電子透かし、音声言語情報処理、ロボットなどの研究に従事。2007 大阪大学大学院工学研究科電気電子情報専攻博士課程修了。2016 豊橋技術科学大学客員教授。電子情報通信学会、日本音響学会各会員。