Vol.2013-SLP-98 No.7
2013/10/25

多様な雑音に頑健なクラウド音声認識サービス「SpeechRec Cloud」の紹介

大庭 隆伸[1,a)]　鎌土 記良[1,b)]　浅見 太一[1,c)]　青野 裕司[1,d)]　阪内 澄宇[1,e)]　高橋 敏[1,f)]

概要：NTT アイティ株式会社より提供が開始されたクラウド型音声認識サービス「SpeechRec Cloud」には，NTT 研究所で開発された技術が採用されている．本サービスはスマートフォン等のモバイル端末での利用も前提としており，そこでは従来のオンプレミス型のサービスとは異なる問題が生じる．本稿では，「SpeechRec Cloud」の概要を紹介するとともに，モバイル端末を通した利用上の問題点と，それを解決するための技術を紹介する．

キーワード：SpeechRec Cloud, 重み付き有限状態トランスデューサ, 雑音下音声区間検出

Noise-robust ASR Service: SpeechRec Cloud

Abstract: NTT-IT Corporation started to provide the automatic speech recognition (ASR) service, "SpeechRec Cloud", which enabled customers to use the ASR through a mobile device such as smart-phone. To achieve the cloud ASR service, there were problems to be solved. This paper outlines the "SpeechRec Cloud" service and describes the technologies that solves these problems, which were developed at NTT laboratories.

Keywords: SpeechRec Cloud, Weighted finite state transducer, Noise-robust voice activity detection

1. はじめに

NTT での音声認識の主要な導入先の１つにコールセンタがある．コールセンタでは１センターで１日平均に数千件もの通話が毎日のように行われている．これら大量の通話の中からお客様の要望の抽出や，オペレータの対応に関して，問題のあるもの，他のオペレータの手本になるものを見つけるなどいった情報抽出は，業務の品質向上や効率化の観点から重要である．これを行うためには，音声通話をテキスト化しておく必要性があり，音声認識はその重要な基盤技術となっている．

これまで，コールセンタをはじめ多くの場面において，音声認識はオンプレミス型のサービスとして提供されてきた．すなわち音声認識に必要な機器一式を使用場所に導入する形を取ってきた．しかし，初期導入のコストおよびシステム管理上のコスト，リスクが導入障壁となっていたのも事実である．この問題に対する１つのソリューションはクラウド対応である．機器を自社で持つことなく音声認識サービスを受けられるようになり，導入時および保守に掛かるコストを低く抑えられる．また，従量制のサービスを選択することで事業の伸張性に応じて柔軟にサービスの利用量を切り替えることができる．

クラウド化は音声認識事業者から見ても販路拡大の好機である．ネットワークに接続されたあらゆる端末に対して音声認識の提供が可能となり，利用者の大幅な増加が見込まれる．特に最近のスマートフォンの急速な普及は，クラウド型音声認識サービスにとって追い風となっている．キーボードを備えるコンピュータに比べ，圧倒的に文字入力に難があるスマートフォンでは，音声での入力が有効であり，また通信装置も標準的に搭載されているためである．

NTT アイティ株式会社では，2013 年 9 月 24 日に，スマー

[1] 日本電信電話株式会社メディアインテリジェンス研究所
Yokosuka, Kanagawa 239–0847, Japan
[a)] oba.takanobu@lab.ntt.co.jp
[b)] kamado.noriyoshi@lab.ntt.co.jp
[c)] asami.taichi@lab.ntt.co.jp
[d)] aono.yushi@lab.ntt.co.jp
[e)] sakauchi.sumitaka@lab.ntt.co.jp
[f)] takahashi.satoshi@lab.ntt.co.jp

トフォンやタブレット等で簡単・高精度な音声認識が利用できる「SpeechRec Cloud」サービスの提供を開始した [1]. これまでオンプレミス型で提供してきた「SpeechRec」のクラウド版とも位置付けられる本サービスを利用することで，従来より短期間かつ低コストで音声認識を導入・利用することができる．スマートフォンやタブレット等のモバイル端末用のインターフェースも用意されている．

一方，利用シーンの拡大は音声認識の研究者にとっても喜ばしいことであるが，同時に技術的な課題が生じる．特にスマートフォンでの利用に着目すると次のような課題があげられる．１つ目は，語彙と言語モデルに関する課題である．音声認識では，事前に登録されていない単語は認識できない．また，学習時と異なるタスクの認識率は著しく低下する場合がある．しかしながら，スマートフォンには様々なタイプのアプリが存在しており，音声認識もどのような用途で使用されるか定かではない．このような状況下では，語彙や言語モデルは極めて多数の単語と極めて広い話題をカバーしておく必要がある．

２つ目の課題は雑音である．スマートフォンは常に携帯されるので，どこで使用されるわからない．駅，人混み，社内，テレビの前など多種多様な音環境での利用が予想される．雑音下では，音声の検出さえ難しく，これが失敗すると音声認識を始めることさえできない．もちろん，複数マイクによるアレー処理などはデバイスの制約上導入できない．また雑音を事前に網羅的に観測しておくことができない．そのため雑音を推定・トラッキングしながらの音声の区間検出が必要となる．「SpeechRec Cloud」には，これらの課題に対して NTT 研究所で開発された技術が採用されている．本稿では，「SpeechRec Cloud」の概要を示すとともに，これら技術について紹介する．

2. クラウド型音声認識サービス「SpeechRec Cloud」

「SpeechRec Cloud」は，従来のオンプレミス型の「SpeechRec」を，スマホやタブレット等からより簡単に利用するできるよう開発されたクラウド型の音声認識サービスである．音声認識エンジンには NTT 研究所で開発された「VoiceRex」を搭載している．重み付き有限状態トランスデューサ (WFST) 技術を基軸に，高速・高精度な音声認識が可能である．

「SpeechRec Cloud」の概要を図 1 に示す．「SpeechRec Server」にインターネットを介して音声データを送ることで，音声認識結果を受け取ることができる．しかし実際に音声認識を利用するためには，周囲の雑音を抑圧する処理，端末のマイクの制御，音声データ送信のためのエンコード処理など複雑な処理が必要となる．「SpeechRec Cloud」では，これら基本的な機能を搭載したクライアントライブラリも提供する．これにより高性能な音声認識を利用する

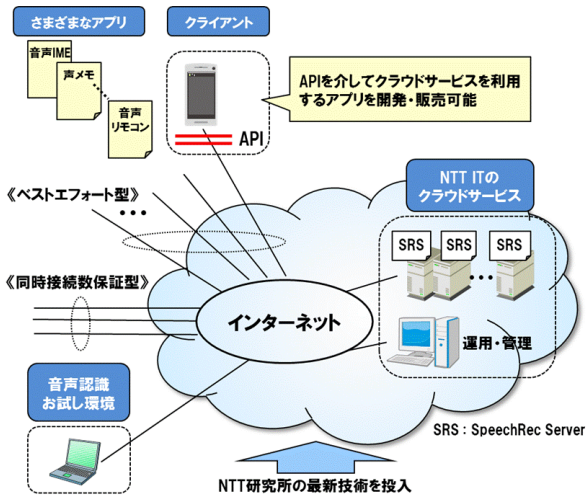

図 1 SpeechRec Cloud Service.

アプリケーション開発を，より簡単に行うことができる．また，独自の音声処理機能を搭載したい場合には，パケットの送受信など基本となるライブラリのみの使用も可能である．

クライアントとしては，スマートフォン，タブレット，パソコンの他，従来的なオンプレミス型のサービス提供時に使用してきた通録装置等との連携も対象となっている．また，マイクから直接収録された音声以外に，音声ファイルを認識することも可能である．多様な入力系をカバーすることで，様々な用途に音声認識を適用することができる．

多様な用途への対応という意味では，提供するサービスも柔軟に選択可能でなければならない．そこで「SpeechRec Cloud」ではリアルタイム性重視と利用頻度重視の観点から以下の 2 種類のサービス種別を提供している．

(1) 同時接続数保証型

「SpeechRec Server」のプロセス数（同時に認識処理を実行できる数）単位での契約で，リアルタイムの音声認識が必要で常時利用が見込める用途に適している．このような場合には，音声認識のモデルをタスクに応じて変更する必要性が生じる場合が多いと予想される．そこでオプションとしてモデル変更を行うこともできるようにしている．同時接続数保証型の利用例としては，スマートフォンやタブレットでの音声によるテキスト入力や，音声認識機能を具備したリモコンアプリといったものが挙げられる．

(2) ベストエフォート型

「SpeechRec Server」のプロセスを複数のユーザで共有するサービスで，ユーザ数単位での契約となる．リアルタイム性よりも，1 ユーザ単位で低コストに利用したい用途に適している．例えば，会話を録音したファイルを音声認識でテキスト化するといった用途での利用が想定される．

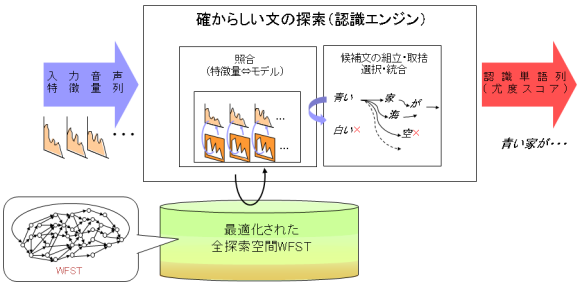

図 2 WFST Speech Recognition.

3. WFST と高速 on-the-fly 合成

「SpeechRec Cloud」で採用されている音声認識エンジン「VoiceRex」は，重み付き有限状態トランスデューサ (WFST) に基づく高精度かつ高速な音声認識デコーダである [2], [3]．現行の音声認識技術は，ある入力データに対して，基本の 3 つのモデル，音響モデル，発音辞書，言語モデルを参照し，トータルで尤もらしい文を返すという枠組みで動作する．これらモデルの複雑な連携のため，各モデルの参照のタイミングや，メモリの確保・解放のタイミングなどの設計方法によって，認識の精度，速度は大きく変化してしまう．従来は，ヒューリスティックに設計されていたため，十分な精度，速度を出すことができず，またプログラムも複雑となり保守の面でもコストの高いものとなっていた．

この問題を解決したのが WFST に基づく音声認識のフレームワークである．WFST とは，ネットワークの形状をとる，ある種の表現方式である．3 つのモデル，音響モデル，発音辞書，言語モデルは全て WFST のフォーマットで表現することができる．WFST には，合成，最適化といった演算が定義されている．合成は，2 つの WFST を接続する演算であり，最適化は冗長な部分を取り除くなどの操作を行う．

WFST に基づく音声認識では，まず事前に 3 つのモデルを合成・最適化した WFST を作る．認識時は，この合成・最適化された WFST のみを利用する．WFST はネットワークの形状をしているので，認識の処理としては，このネットワークを辿るだけでよい．これにより，3 つのモデルの複雑な連携は，WFST の演算により最適化され，プログラムもネットワークを辿るだけのシンプルなものになる．結果として，高精度かつ高速で，保守も比較的容易な音声認識デコーダを実現することができる (図 2 参照)．

しかし，WFST には合成により爆発的に巨大化するという問題がある．WFST の合成に耐える巨大なメモリ空間が必要となり，語彙や言語モデルのサイズに関して制限が出来てしまう．この問題に対して，NTT 研究所では高速 on-the-fly 合成法を開発した [4]．この方法では，音響モデル，発音辞書，言語モデルのユニグラム部分を合成・最適化した WFST と，言語モデルの残りのパートに関する WFST を分けて持っておき，認識時に必要な分だけ合成する．これにより WFST の巨大化を防ぐことができる．

この技術による効果の 1 つは，音声認識の超大語彙化である．音声認識は事前に登録された語彙しか認識できないが，WFST の巨大化のため通常は 10 万語程度での利用が標準的である．しかしながら，本技術を搭載した VoiceRex では，1000 万語彙の実時間音声認識も可能である．例えば，広辞苑の収録語数は 24 万語である．つまり，広辞苑アプリといったものも作ることができる．医療用語辞書，新語辞書など，様々な辞書を全て登録しても，1000 万語に到達するのは難しい．世界中の地名の認識や，日本人のあらゆる苗字 (姓)，および名を登録するといった使い方もできる．実際，VoiceRex の汎用モデルには膨大な数の単語が登録されており，この汎用モデルだけでも様々な用途に対応できるものになっている．

WFST の高速 on-the-fly 合成法のもう 1 つ利点は，サイズの大きな (トライグラムのエントリ数の多い) 言語モデルを利用できる点にある．WFST の巨大化を防ぐため，プルーニングされた小さな言語モデルを使用する必要はない．プルーニングは効果的で，精度の低下が少ないとされているが，それは学習時と評価時のタスクが一致している場合ある．「SpeechRec Cloud」における使用方法のように，事前にどのような用途に音声認識が使用されるかわからない状況にあって，広く話題，タスクをカバーすることは極めて重要である．タスク頑健性の観点において，様々なトライグラムエントリを保持することは有効である [5]．高速 on-the-fly 合成により，巨大な言語モデルを扱うことができ，結果として，より多くの話題，タスクをカバーするような言語モデルを利用することができる．

4. 雑音下音声区間検出

通常，スマートフォンで音声認識を用いる場合には，音声区間検出，雑音除去，音声認識の順で処理が行われる．音声区間検出で誤ると，その後の処理が全て失敗するため，高精度な検出が求められる．しかし，雑音下では，音声信号が雑音信号に埋もれ適切な検出が難しい．そこで NTT では，雑音を推定・除去しながら，音声区間検出を行う技術を開発した [6]．これにより，音声区間検出の精度に加え，雑音の除去精度も向上させることに成功した．

まず雑音推定について説明する．スマートフォンは様々な環境で使用されるため，あらゆる雑音を予め学習しておくことは困難である．そこで入力に沿って適応的に雑音に追従するモデルを採用した．これにより使用環境にあった雑音モデルを獲得することができ，かつ雑音の変化にも対応できる．

音声の検出については，予め音声のモデルと無音のモデルを学習しておく．そして，入力が与えられたときに，音声と雑音を合成したモデルと，無音と雑音を合成したモデ

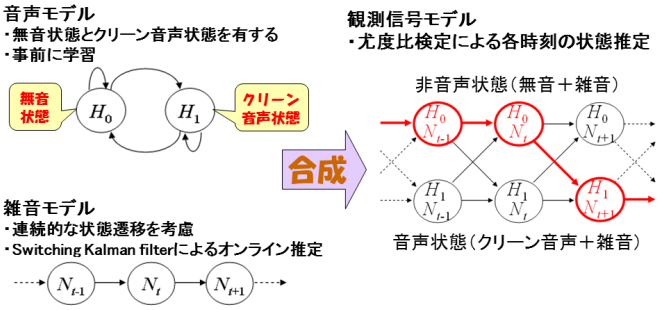

図 3 Voice Activity Detection with Noise Tracking.

ルのどちらによりマッチするかを計算し，前者のスコア (確率) が高ければ音声が入力されたものと判断する (図 3 参照).

そして，雑音除去処理ではまず，音声と無音に関する合成前後のモデルの差分から，音声を抽出するフィルタと，無音を抽出するフィルタを生成する．次に，両フィルタを音声検出処理時に算出したスコア (確率) の比で重ね合わせたフィルタを作る．このフィルタに入力音を通すことで，雑音成分が取り除かれた音声データを得ることができる．

これら一連の処理の演算量は極めて小さく，スマートフォン上でも容易に動作する．また処理はフレーム毎に逐次的に行われるため，入力に沿った音声区間検出および雑音除去が可能である．これにより音声データを逐次パケット化して送ることができるため，音声入力の末尾を待たずして音声認識処理を開始することができる．

5. まとめ

NTT アイティ株式会社から提供された「SpeechRec Cloud」と，そこに採用されている NTT 研究所で開発された技術を紹介した．「SpeechRec Cloud」はスマートフォン等からの利用も前提としており，語彙や言語モデル，雑音処理に関する課題を克服する必要があった．それらを解決する技術として，WFST の高速 on-the-fly 合成法や，雑音の推定・除去を行う音声区間検出法について概要を示した．前者は，1000 万語彙の超大語彙音声認識を可能にするものであり，様々な音声認識アプリを支える基盤となる技術である．また，タスクの違いによる言語モデルの頑健性を担保するためにも重要な技術である．後者は，高雑音下でも高精度な音声区間検出を実現するとともに，雑音除去技術としても高精度に動作する．また，任意の雑音への適応や，変化する雑音の追従が可能であり，使用場所を事前に特定できないスマートフォンでの利用に適した技術となっている．

参考文献

[1] http://www.ntt-it.co.jp/press/2013/0910/
[2] 堀 貴明，塚田元: 音声情報処理の最先端 3「重み付き有限状態トランスデューサによる音声認識, 情報処理学会誌「情報処理」45 巻 10 号, pp. 1020-1026, (2004.10).
[3] Takaaki Hori and Atsushi Nakamura: *Speech Recognition Algorithms Using Weighted Finite-State Transducers*, Morgan&Claypool Publishers, 2013.
[4] Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura: *Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition*, IEEE Transactions on Audio, Speech, and Language Processing, Vol. 15, No. 4, pp. 1352-1365, 2007.
[5] 増村亮, 政瀧浩和, 大庭隆伸, 吉岡理, 高橋敏: 生成型アプローチによる *Latent Words Language Model* の *N-gram* 近似, 情報処理学会研究報告. SLP, 音声言語情報処理 2013-SLP-97(5), pp. 1-8, 2013.
[6] Masakiyo Fujimoto, Kentaro Ishizuka, and Tomohiro Nakatani: *Study of integration of statistical model-based voice activity detection and noise suppression*, in Proceedings of Interspeech '08, pp. 2008-2011, Sept. 2008.