電話音声による情報検索を行う基盤ソフトウェア

Free Software for Voice-Input Information Services on the Phone

住吉貴志
Takashi SUMIYOSHI
E-mail: sumiyosi@kuis.kyoto-u.ac.jp

京都大学情報学研究科知能情報学専攻 修士課程 1 回
〒606-8501 京都市左京区吉田本町 (工学部 10 号館)

Abstract These days we can access various information with cellular phones. In such cases, voice input has great advantages for specifying queries. In this project, we developed a speech recognition engine which implements Microsoft Speech API (SAPI) based on open software 'Julius' and 'Julian', that had been developed in Kyoto University. And we also made acoustic models for the telephone speech and a VoiceXML interpreter. These are free software and expected to be used for information services on the phone.

1 背景

近年種々の情報が Web 上に蓄積され、種々の情報検索や予約などの取引がオンライン上でなされている。また、携帯端末・携帯電話によるアクセスも爆発的に普及している。一方で、音声（電話）による問い合わせも依然としてかなりの割合を占めているが、アクセスする情報はほとんど Web のものと一元化されつつあるのが実状である。したがって、ブラウザを使うか音声を使うかというのは、インタフェースの違いと考えてよく、両者の特質を考慮してサービスが設計されるべきである。

情報検索に音声を用いることの主な利点のひとつは、その自然さである。我々人間どうしのコミュニケーションの大部分を占める音声による意思伝達は、ユーザに多くのスキルを要求しないため、音声はストレスのない、よいインターフェースであるといえる。また、例えばレストランを検索する場合、Web 上では「京都市」ー「四条河原町エリア」ー「居酒屋」といったようにメニューをたどっていくが、音声では「四条河原町にいるのですけど、このあたりで今開いている居酒屋を教えて下さい」といったショートカット的な問い合わせが可能である。

音声認識を用いた電話による情報検索・システムはいくつか実用化が行われているが、基本的には単語ベースの認識であり、自然言語を対象としたものは少ない。より自然で使いやすいサービスの提供には、自然言語ベースの認識が不可欠である。

音声認識技術は進歩しているが、どうしても認識誤りは避けられない。これは人間でも起こりうるが、人間の場合は認識・理解できたキーフレーズを基に解釈を構成し、曖昧な点を対話によって詳細化するというプロセスによって、最終的に要求を満たす情報に到達しているものと考えられる。そのため、認識エンジンの認識誤りを対話によって回復させる研究も盛んに行われている。

残念ながら人間程度のレベルの人工知能をもつコンピュータはまだ実現できていないため、対話のプロセスはタスクごとに設計しなければいけない。その際、対話を簡潔に記述する枠組みが必要であるため、近年 VoiceXML という規格が提唱されているが、その実装は少なく、普及しているとはいいがたい。

2 目的

本研究開発では、自然な音声で情報検索を行えるシステムを実現することを目的とする。前章に挙げた問題のソリューションとして以下を行った。

1. 電話音響モデルの構築
2. フリーで汎用的な音声認識ソフトウェア (Julius for SAPI) の開発
3. 標準的な仕様記述言語の処理系 (VoiceXML インタプリタ) の開発

以降の章では、それぞれの詳細な説明を行う。

3 電話音響モデル

電話音声は帯域が制限されており、また雑音や回線上の混線などの問題もあるため、一般に直接マイク等で録音した音声に比べて質が悪く、音声認識が難しい。電話音声対話システムの性能の向上のためには、音響モデルは非常に重要であるといえる。

音響モデルの性能を向上させるには、適切な音声データを使って学習する必要がある。通常環境における音声データを帯域制限等の前処理を行なって擬似的な電話音声を作成し、学習する方法もあるが、より精度の向上を望むならば、実際の電話の音声を収集するしかない。

そこで、実際の電話音声 (固定電話、携帯電話) を収集し、音響モデルを作成した。

学習データは総時間約 20 時間、517 名の発話である。固定電話と携帯電話のデータがあるがそれらを混合して学習した。

得られたモデルは、通常環境におけるデータから作成したものに比べて高性能であることが確認されている。

4 音声認識エンジン Julius for SAPI

4.1 Julius/Julian について

Julius [2] と Julian [3] はともに京都大学音声メディア研究室で開発された大語彙連続音声認識エンジンである。両者の違いは使用する言語モデル (制約文法) であり、Julius は統計的言語モデル (SLM) を、Julian は BNF による記述文法を用いるが、どちらも同じコンセプトの 2 パス探索アルゴリズムを採用している。

Julius は IPA 「日本語ディクテーション基本ソフトウェア」における認識デコーダとして、ディクテーション用の単語辞書、統計的言語モデル、音響モデルなどとともに配布されている [4][5]。

本研究開発では、SAPI の要求機能であるディクテーション機能とコマンド認識機能を、それぞれ Julius 、Julian の機能を利用して実現した。

4.2 Speech API(SAPI) について

Microsoft 社は Windows Speech API(SAPI) と呼ばれる規格を提唱しており、2000 年に Speech API 5.0 とその SDK をリリースした。SAPI はアプリケーションが音声認識と音声合成を統合的に扱えるように規格化されている。音声認識では、認識エンジンの種々の機能を API として規格化し、アプリケーション開発者、エンジン開発者に提供するものである。また SAPI は、ウィンドウシステムにおけるマルチタスクアプリケーションのインターフェースとしての音声の利用を考慮した設計となっている。

最近では、Office xp は SAPI に対応しているし、また Windows xp にはコントロールパネルに音声の項目がデフォルトで登録されており、音声機能が利用可能な状態になっている。このようなことからも、SAPI が今後の Windows の音声機能のデファクトスタンダードになることは十分に考えられる。

なお、本研究開発では、2001 年 8 月にリリースされた Speech API 5.1 [1] を対象として開発を行った。

4.3 Julius for SAPI の動作仕様

Julius for SAPI は以前から実装を開始しているが、本研究開発ではさらなる機能の向上を目指す。今回実装した Julius for SAPI は、以下の機能を有する。

- ディクテーション機能
  Julius と同程度の認識性能を提供する。
- SAPI 独自の XML 形式文法 (以下、SAPI GRAMMAR) による認識
  SAPI で定められている文法形式ファイル (図 1) を認識する。オリジナルの Julian では、あらかじめ DFA 形式に変換したファイルを用いる必要があったが、Julius for SAPI では実行時に変換する。
- GUI による認識オプションの設定
  従来のコマンドラインの引数からのオプション指定を、SAPI のコントロールパネルから行えるようにした。

4.4 コマンド 文法について

コマンドとは、「前に戻れ」などの音声による指示のことであり、アプリケーション開発者が、ユーザが発話するであろうコマンドを前もって文法で記述する。音声認識エンジンはこの文法をユーザの発話の前提条件として利用することにより、効率良く認識処理を行うことができる。

SAPIGRAMMAR の例を図 1 に示す。

```
<?xml version="1.0" encoding="UTF-16"?>
<GRAMMAR>
  <RULE NAME="YESNO_MAIN" TOPLEVEL="ACTIVE">
    <L PROPNAME="ANSWER">
      <P VALSTR="YES">はい</P>
      <P VALSTR="NO">いいえ</P>
    </L>
  </RULE>
</GRAMMAR>
```

図 1: SAPI コマンド文法の例

SAPI はこの文法を RTN (Recursive Transition Network) 形式に変換してエンジン (Julius for SAPI) に提供する。Julius for SAPI は提供された RTN を、Julian で利用できる DFA に変換する。なお、SAPI-

GRAMMAR は文脈自由言語のクラスまで記述が可能であるが、Julian は正規言語のクラスまでしか扱うことができないが、一般の音声対話タスクにおいて文脈自由文法まで必要となるケースは極めて少ない[*1] ため、Julius for SAPI では正規言語のクラスに落とせる文法のみをサポートする。

また、Julian ではカテゴリという概念があるが SAPIGRAMMAR にはない。Julius for SAPI では生成した DFA の遷移のうち、遷移元状態と遷移先状態のペアの集合が完全に一致するような遷移記号の集合を 1 つのカテゴリとみなしている。

Julian では音声の切れ目となりうる任意の場所に sp 遷移を挿入することになっているが、SAPIGRAMMAR では明示的に指定しない。Julius for SAPI ではすべての遷移のあとに sp 遷移を強制的に挿入している。

4.5 オブジェクトモジュール化

SAPI の仕様では一つの入力音声に対して複数の文法で処理し、正しい認識結果を返す必要がある。また、文法や語彙がアプリケーション実行中に動的に変わる可能性も考慮しなくてはならない。

Julius/Julian のオリジナルのソースは C で書かれており、上記のような処理に関してはほとんど考慮されずに設計されている。そのため本研究開発ではクラスやインターフェースを設計し、全面的に C++ による書き換えを行った。

4.6 SAPI Compliance Test

SAPI の SDK に付属する Compliance Test を用いて、Julius for SAPI の性能を評価した。結果を表 2 に示す。○は PASSED、×は FAILED を表す[*2]。Compliance Test を完全にクリアするには至らなかったが、通常の利用には十分耐えうるものである。

4.7 アプリケーションからの利用

Julius for SAPI は SAPI のインターフェースを搭載しているため、実際にアプリケーションから Julius for SAPI を利用しようとする場合は、SAPI のマニュアルの通りに行えばよい。

以下にアプリケーションの例を示す。

- Dictation Pad
  Dictation Pad とは、SAPI の SDK に付属するサンプルの 1 つであり、ディクテーションとコマンドを利用し、音声による文法入力と簡単な編集作業ができるアプリケーションである。音声版のメモ帳であるといえる。
  SDK に付属の文法ファイルでは英単語が使用されていたが、Julius for SAPI では英単語を日本語として処理する機能はないため、日本語に書き換える必要がある。この作業はテキストエディタを用いて簡単に行うことができ、それによって Julius for SAPI で動作させることができた。
- 音声操作プロジェクタ
  音声によって PowerPoint などのプレゼンテーションソフトを操作する枠組みが開発されているが、それを Julius for SAPI と SAPI を使って実装した。実装は、SAPI と Julius for SAPI を使って認識した音声によって PowerPoint に命令を送るモジュールを作成することで行った。Office xp 付属の PowerPoint 2002 では音声がサポートされているが、今回の実装では以前のバージョンの PowerPoint (PowerPoint 2000) でも動作する。

SoundStart	○
SoundEnd	○
FalseRecognition	×
PhraseStart	○
Recognition	○
SoundStart → SoundEnd order	○
PhraseStart → Recognition order	○
Events Offset	○
Sync. before loading C&C	○
Sync. C&C after loading engine	○
App. Lexicon for C&C	○
Uses User lex. before app. lex. for C&C	○
Case sensitive lexicon	○
L tag	○
Expected Rule	○
P[hrase] tag	○
O[ptional] tag	○
RULE and RULEREF tags	○
/Disp/lex/pron	○
Case sensitive grammar	○
SpPhraseElements	×
Auto. pause engine on recog.	×
Invalidate top level rule	×
Invalidate non-top level rule	×
Multi instances	×
Multi app. contexts	×

図 2: SAPI Compliance test (required) の状況

5 VoiceXML インタプリタ

- VoiceXML について

[*1] 特に検索タスクでは一発話は長くないので、有限の状態でも十分実用的である。

[*2] Invalidate top level rule, Multi app. context などが FAILED になっているが、これらの機能はすでに実現され、他のアプリケーションで動作を確認しているため、テストに通らないのはもっと些細な問題が原因であると考えられる。

VoiceXML とは、インターネットの情報に電話などの音声を使ってアクセスする際の、システムとユーザとの対話部分を記述する言語である [7]。これは form を利用した HTML のインターフェースに似ており、対話によって得られた検索条件などの結果を従来の CGI を用いて処理させることができる。VoiceXML は AT&T、IBM などがスポンサーとして参加しており、業界標準となることが期待されている。最新の情報は VoiceXMLForum [7] などから得ることができる。

- VoiceXML インタプリタの仕様と実装内容
  現在 VoiceXMLForum では VoiceXML2.0 のワーキング・ドラフトが発表されているが、本研究開発では 2000 年 5 月に発表されている VoiceXML1.0 を対象とした。
  VoiceXML では、対話制御のためのスクリプトを ECMAScript [*3] によって記述するように定められている。ECMAScript のインタプリタとして Microsoft の Active Scripting エンジンを用いた。また、XML のパーサには Microsoft XML Core Services 4.0 [*4] を用いた。文法は、Julius for SAPI で扱う文法、すなわち SAPI の XML 形式のものが利用可能である。
  現時点でいくつかの VoiceXML ドキュメントに対して動作しているが、実装はまだ完全なものとは言えず、DTMF(電話のトーン音) の認識や一部のイベントには未対応である。

6 電話音声検索システムの構成

これまでの章で説明した電話音声モデル、Julius for SAPI、VoiceXML インタプリタを用いると、電話音声検索システムが作成可能である。システムの構成例を図 3 に示す。

7 今後の課題

Julius for SAPI については、今回 SAPI の Compliance Test で通らなかった部分を実装し、より完成度を高めていく予定である。またいずれ発表されるであろう SALT [*5] のスピーチタグが Julius for SAPI で動作するかを確認したい。

VoiceXML インタプリタについても、改善の余地が残されている。また将来、現在ドラフト段階である VoiceXML2.0 についての対応も行う予定である。

[*3] http://www.ecma.ch/ecma1/stand/ecma-262.htm
[*4] http://www.microsoft.com/japan/developer/xml/
[*5] http://www.saltforum.org/

図 3: システムの構成

また、これらの成果物による京都市のレストランなどの情報案内システムを実際に構築する。

8 まとめ

本研究開発では、電話音声音響モデル、Julius for SAPI、VoiceXML インタプリタといった、電話音声による情報検索サービス構築のための基盤となるソフトウェア群を作成した。これにより将来的には音声情報検索サービスの普及と一般化がより進むものと期待できる。

9 参加企業

- 財団法人 京都高度技術研究所
- 株式会社 電話放送局

参考文献

[1] Microsoft: *Microsoft Speech SDK version 5.1* (2001).
http://www.microsoft.com/speech/

[2] 李晃伸, 河原達也, 堂下修司: 文法カテゴリ対制約を用いた A*探索に基づく大語彙連続音声認識パーザ, 情報処理学会論文誌, Vol. 40, No. 4, pp. 1374–1382 (1999).

[3] 李晃伸, 河原達也, 堂下修司: 単語トレリスインデックスを用いた段階的探索による大語彙連続音声認識, 電子情報通信学会論文誌, Vol. J82-DII, No. 1, pp. 1–9 (1999).
http://winnie.kuis.kyoto-u.ac.jp/pub/julius/

[4] 河原達也, 李晃伸, 小林哲則, 武田一哉, 峯松信明, 嵯

峨山茂樹, 伊藤克亘, 伊藤彰則, 山本幹雄, 山田篤, 宇津呂武仁, 鹿野清宏: 日本語ディクテーション基本ソフトウェア (99 年度版) の性能評価, 情報処理学会研究報告, 2000-SLP-31-2 (2000).
http://winnie.kuis.kyoto-u.ac.jp/dictation/

[5] 鹿野清宏, 伊藤克亘, 河原達也, 武田一哉, 山本幹雄 編著: 音声認識システム, オーム社 (2001).

[6] 大語彙連続音声認識デコーダ Julius,
http://winnie.kuis.kyoto-u.ac.jp/pub/julius/

[7] VoiceXML Forum, http://www.voicexml.org/

[8] Voice eXtensible Markup Language VoiceXML Version 1.00,
http://www.voicexml.org/specs/VoiceXML-100.pdf