Android 携帯の音声認識

小池　裕介　　塚岡　悠人

要約

近年，携帯電話が普及してきたが，最近では通話ツールとしてではなく，インターネット端末として使われ始めている。

さまざまな機能が搭載されてきている現在，その機能を活用できるソフトウェアの開発が多くなされてきてる。

その中で，iPhone には Siri（iOS 向け秘書機能アプリケーションソフトウェア）と呼ばれる自然言語処理を用いて質問に答える機能がある。しかし，携帯電話会社のキャリアごとには音声サービスが行われているが，AndroidOS としての音声認識機能はあまりない状態にある。

音声認識の仕組みを考え，かつ解となる文章を音声化する手段を提示できるよう，今回の研究に至った。

In recent years, mobile phone usage has spread. However, they are not used only to make telephone calls, but as Internet terminals.

We took notice of the iPhone program called Siri. After Siri understands our words and analyzes it, it replies to our question. This function is not in Android phones, so we made our own version.

キーワード　　スマートフォン，Android OS，音声認識

Smart phone, Android OS,Speech recognition

１　序論

近年，携帯電話が普及してきたが，今まではフューチャーフォンと呼ばれる通常の携帯電話が主流であった。それが，スマートフォンと呼ばれる高機能型携帯電話が多くを占めるようになってきた。

それと同時に，携帯電話で使われるアプリケーションも会社主導のものから，個人で作成し，それを流通させるモデルができてきている。

高機能型であるがため，様々な機能が搭載されており，かつ，それを利用するアプリケーションも流通してきている。

その中で，音声コントロールについては，簡単な音声認識程度のものが作られていたが，iPhone4S の登場の際，使い物になる音声コントロールシステムの”Siri”が搭載され話題となった。

そのシステムを構築していくことを目的として，研究を行ってきた。

２　研究方法

２．１スマートフォンの現状

スマートフォンは，ある意味通信機能がついた PC とみることができる。そのため，スマートフォンの機能を管理するために OS が使用されている。

現在スマートフォンを管理する OS には，Apple 社の iOS，Google 社の AndroidOS，Microsoft 社の WindowsPhone，BlackBerry 社の BlackBerryOS と，いろいろな種類がある。その中でも AndroidOS は，無料で搭載することができるため，世界の様々なメーカーのスマートフォンに搭載されており，シェアの半数以上を占めるようになってきている。

２．２　開発環境

スマートフォンは，今までのフューチャーフォンに比べ，様々な種類のアプリケーションが多く作られている。これは，ソフト会社のみならず，個人，グループが作成したアプリケーションが多くあり，マーケットに登録され，ユーザーが簡単に利用できることに起因している。

また，個人かアプリケーションを開発する環境が手軽に手に入ることも要因の一つであると考えられる。

スマートフォンのOSの中で，Windowsでの開発環境が整っているものとしては，AndroidOSとWindowsPhoneがある。どちらも無料で入手することができる。そのうちAndroidOSは，OS自体がオープン化されており，アプリケーションが完成後，それを流通させるためのマーケットへの登録が容易である。それ故，今回の研究対象としてAndroidOSをターゲットとして考えた。

AndroidOSの開発言語としてはJava言語とC++言語とがあるが，一般的にJava言語が主流であるため，Java言語を使用した。また，Java言語に対してのSDK（Software Development Kit）が充実しているのも特徴である。

Java言語は1995年，Sun Microsystemsが開発した言語である。Java言語の利点はPlatformに依存しないアプリケーションの開発と配備が行うことができることである。

通常，テキストエディタでコードを書き，それコンパイラを使いコンパイルし，実行用プログラムを作成する手順が使われる。しかし，この方法では，コンパイル中のエラーの発生や実行用プログラムを起動したときのバグなどが発生した場合，再度テキストエディタでソースのデバッグを行い，再度コンパイラを起動しコンパイルするという作業になる。そのためそれぞれの手順を切り替える手間が多くなる。

開発効率を考え，すべてがそろっている開発環境を考えIBM社が開発した「Eclipse」を利用した。

「Eclipse」は無料の統合開発環境(IDE)である。IDEとはエディタ，コンパイラ，デバッガなど，プログラミングに必要なツールが一つのインターフェースで統合して扱えるような環境である。その中で「Eclipse」は，いろいろな言語（C，C++，C#，Javaなど）に対応しており，また，純正のエミュレーターを「Eclipse」上で動作させることができるため，AndroidOSのアプリケーション作成の標準ツールとなっている。

２．３　研究内容

今回作成を試みたアプリケーションは，音声を入力し，その音声を認識した後，音声を解析し，解を表示するモデルを考えた。

アプリケーションの流れは，①マイクで音声を入力②音声をひらがな化③ひらがなの文章化④文章の内容解析⑤内容からの解の検索，決定⑥解の文章化⑦文章の音声化と考えた。

最初に②と⑦の部分のプログラミングを行った。②では参考文献を利用し，プログラミングを行い（図１），エミュレーター上で動作を確認した。（図２）

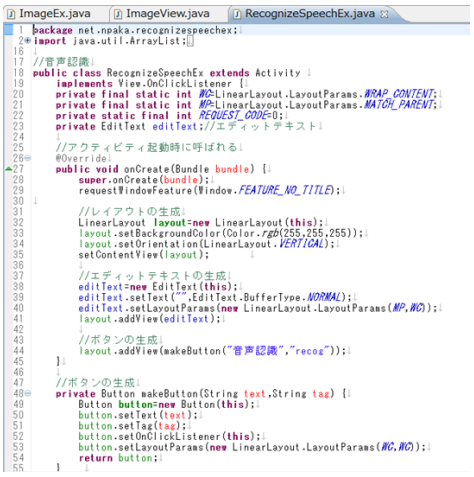

図１　音声のひらがな化（プログラム）

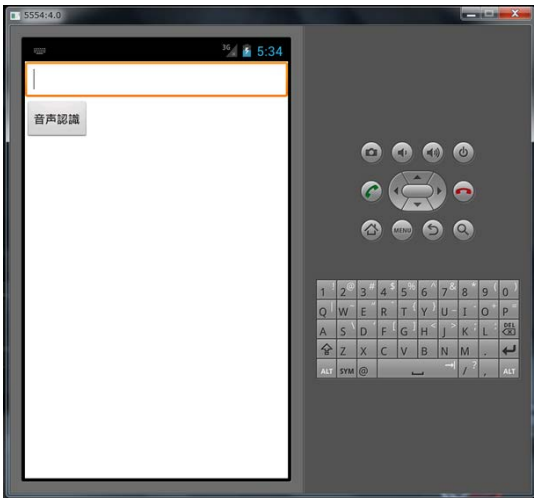

図２　音声のひらがな化（エミュレーター）

⑦においても同様に作業をし（図３），動作の確認を行った。（図４）

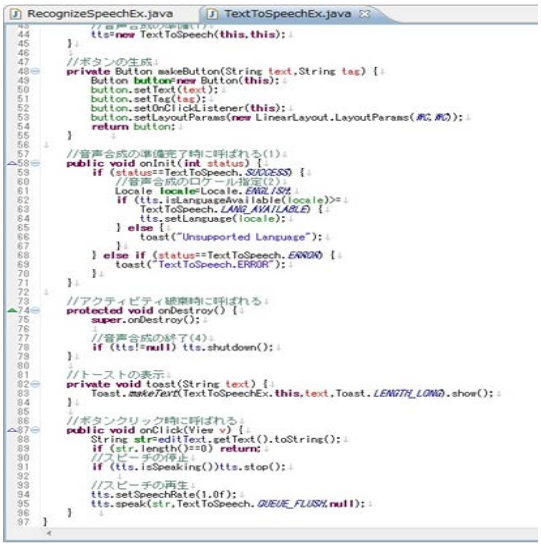

図３　文章の音声化（プログラム）

図４　文章の音声化（エミュレーター）

その後，AndroidOS を搭載したスマートフォン実機にプログラムを転送し，実機での動作の確認を行った。

３　研究結果

上記のアプリケーションの流れの②と⑦については，エミュレーター上での動作はできた。その際プログラミングに利用した PC がマイク非搭載であったため，②についてはエミュレーター上では確認がとれていなかった。実機に搭載したところ，動作することが確認できた。⑦についてはエミュレーター上で確認後，実機でも動作することが確認された。

実際の動作では，①は音声を入力したところ，テキストで表示されるときにひらがなで表示され，かつ，実機に搭載されている IME を利用し，漢字に変換された文章が何種類も表示された。これは，プログラムの制御ではなく実機の制御で行われた部分もあり（図５），詳しくは解析できていない。

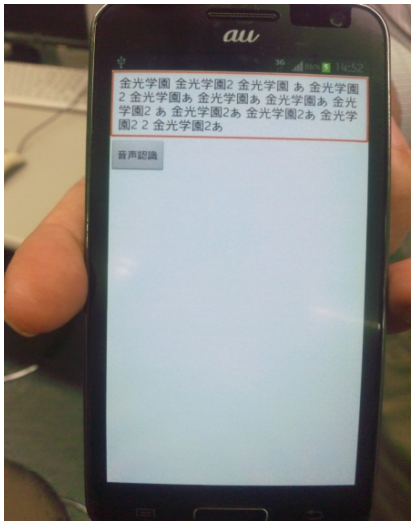

図５　文章のひらがな化（実機）

⑦については，エミュレーター上，実機上ともに日本語を入力して音声化しようとしたが（図６），日本語は受け付けず，英語のみの対応であった。

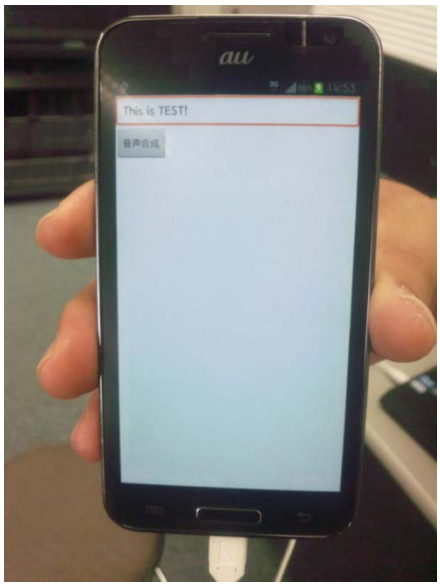

図６　文章の音声化（実機）

②と⑦のプログラミングおよび検証に時間がかかったため，③～⑥についてはプログラミングおよび検証をすることができなかった。

4　考察

今回の研究には様々な問題が生じた。最初に開発環境として，当時コーディングなしでビジュアル的にパーツを組み合わせてソフトをつくる Google 社の「App Inventor」があり，そのソフトを利用しようとしたが，Google 社が「App Inventor」の公開中止があり，当初の予定が変更になったこと。(のちに MIT により再度公開されたが, 結果利用するに至らなかった) 開発を「Eclipse」に変更したが，まず Java 言語の理解に時間がかかり，かつプログラミングで利用した PC の性能が低く，なかなか「Eclipse」が正常起動せず時間がかかり，全体のプログラムまで到達しなかった。

また別の問題点として，音声入力の日本語に非対応ということである。現在の研究では音声入力プログラムは英語のみ対応であり，日本語には対応していない。日本語の音声の文章化の手順は例として「わたしはいしです」という文を用いて説明しようと思う。 まず，「わたしはいしです」という音声より，「音響モデル」という「あ＝a，い＝i，カ行＝k」などという情報より，入力音声がどの音素の並びであるかを計算する。音素とは語と語の意味を区別する機能を持つ音声の最小単位である。たとえば，「たき（滝）」と「かき（柿）」の語頭の「ｔ」と「ｋ」などである。(出典：三省堂　大辞林) 次に，解析した音素を基に音素列に解析する。音素列とは，ここの例である「いし」で言うと「i/s/h/i」というものである。次に，解析した音素列と内蔵されている辞書とを照らし合わせ，音素列と単語を対応つける。そして，辞書と対応した結果，「いし」という音声に「意思・医師・石・・・」などを候補にあげ，「言語モデル（ルールグラマ)」を使用し，どの単語が適切かを判断する。言語モデル（ルールグラマ）とは，文の品詞や統語構造，単語と単語，文書と文書などの関係性について定型化したもののことである。

ここの例では，「私は」と「です」という情報より，「医師」が最も適切という風に計算することである。これらの手順を通し，「私は医師です」という認識結果が導きだされるのである。しかし，日本語の音声認識には多くの問題点が存在する。まず「同音異義語が多い」ということである。「医師・石・意思・・・」を識別させるプログラムを開発するのは非常に困難である。次に「付属語が多い」である。例として「私の・私が・私に・・・」である。英語には存在しない付属語を音声解析するプログラムの開発も非常に困難である。最後に「方言がある」ということだ。英語にもなまりがあると言っても，日本語ほどではない。日本語の場合，極端にイントネーションが違うため，音声認識をするときに正常に認識されない場合があるのである。方言も込みで日本語の音声認識プログラムを開発するにも困難である。以上の理由により日本語の音声認識プログラムの開発は難しいという事である。

5　課題

今後の研究課題としてはまず音声認識プログラムを作成し，今までに作成したプログラム一つ一つをアプリケーションとして一つにまとめるという事をしていきたい。他にも，日本語の音声認識を成功させる為の方法などを考察していきたい。

6　参考文献

[1]　Android プログラミングバイブル
　　布留川　英一
　　ソシム社（2011-12）
[2]　大辞林
　　三省堂

7　謝辞

今回の研究でお世話になった谷野先生に最高の感謝とお詫びをここに申しあげます。