音声型電子レンジ

安藤　優希
法政大学情報科学部ディジタルメディア学科
15K1103

平成 28 年 7 月 20 日

概要

現代の情報社会で、私たちはパソコンやスマートフォンを媒体として SNS やウェブサイトなどの多くのインターネットサービスを利用した生活を送るようになり、インターネットという巨大な情報網をさらに有効的に使う方向に世界全体がシフトしている。このような状況の中で最も重要となることはすべての人がインターネットを簡単に利用することができるということだと私は考える。世界中にはたくさんの人がいるが体にコンプレックスを持つ人、機器の複雑な機能に慣れない人が一般的に情報機器を簡単に使うことができない人だろう。これらの人が簡単に利用することを考えると、音声を使うすることが大変有用的ではないだろうか。本プロジェクトでは、複雑な機能の機器に音声を取り入れることで簡単に使うことができることをコンセプトとし、情報機器ではないが、私が日ごろから使いにくいと感じていた電子レンジの音声化に取り組んだ。

1　電子レンジのシナリオ

シナリオは私の家の電子レンジをもとにして考えた。まず、大まかに説明すると、自動機能と手動機能の二つがある。自動機能は電子レンジがもとから備えている機能で例えば、「温め」、「解凍」や「牛乳」を温める機能、「シフォンケーキ」を焼く機能などがある。一方、手動機能は使い手がどの機能でどのくらいの時間で何度で利用するかを指定して電子レンジを使う機能である。温め方には、「グリル」、「オーブン」、「レンジ」、「発酵」の四つの機能がある。　シナリオの具体的な流れはプログラムが実行されるとまず、「こんにちは」と表示される。「こんにちは」と表示されたら、マイクに向かって注文を言う。注文の内容は自動機能を使う場合、機能や食品ををいえば実行される。例えば、「おかずを温めてください」と言えば言葉の「温め」という部分だけを読み取られ、「温め」機能が実行され、同じように「シフォンケーキを作ってください」と言えば、「シフォンケーキ」を焼く機能が実行される。手動機能を使う場合

人「グリルで100度で焼いてください。」
↓
電子レンジ「何度で焼きますか。」
↓
人「30分です。」
↓
電子レンジ「かしこまりました。」

図 1: 注文の流れ

は、焼き方、温度、時間の 3 つのキーワードを入力しないと実行されない。例えば、「グリルで 50 分間焼いてください。」と注文した場合、温度の値が入力されていないので温度を入力するように促される。他の場合も同様にキーワードが足りない場合は、入力するように促される。キーワードが 3 つ入力されたら実行される。私はこのシナリオを音声ソフトの julius と python を接続し、連携することで実装した。

2　julius

2.1　julius とは

Julius はフリーの高性能音声認識ソフトウェアで、数万語の語彙を対象とした文章発声の認識を行う能力を持つ。高速な音声認識を一般的なスペックの PC 上で実現し、 認識率は、20000 語彙の読み上げ音声で 90 %以上である。また、記述文法に基づく認識を行うこともでき、文のパターンを人手で記述した認識用文法（有限状態文法）を用いることで、小語彙の音声対話システムや音声コマンド入力など比較的小規模な音声認識システムを容易に構築することが可能である。

2.2　julius の仕組み

文章を認識する場合は，文パターンに関する制約を与える必要がある。julius で用いる文法の形式は独自の

```
S : NS_B KEY PLEASE NS_E
KEY : MODE
KEY : MODE DE NUM TIME
KEY : MODE DE NUM TIME DE NUM DO
KEY : MODE DE NUM DO DE NUM TIME
KEY : AUTO
KEY : NUM DO DE NUM TIME
KEY : NUM TIME
KEY : NUM DO
PLEASE : DE ONEGAI
PLEASE : NISHITE KUDASAI
PLEASE : DESU
PLEASE : DOUSI KUDASAI
PLEASE : DOUSI ONEGAI
```

図 2: grammar ファイル

```
% MODE
グリル          g u r i r u
オーブン        o b u N
発酵            h a k o u
レンジ          r e N j i:
% NUM
1       i c h i
1       i q
2       n i:
3       s a N
4       y o N
5       g o
5       g o:
6       r o k u
7       n a n a
7       s h i c h i
7       h i c h i
8       h a c h i
8       h a q
9       k y u:
% DO
度              d o
% TIME
分              f u n
秒              b y o u
% DE
で              d e
%KUDASAI
ください        k u d a s a i
```

図 3: voca ファイル

ものであり、文法規則を grammar ファイル (図 2) に，単語辞書を voca ファイル (図 3) にそれぞれ記述しなければならない。voca ファイルには語彙がグループに分けられ、例えば、私のプログラムでは mode というグループには、「グリル」、「オーブン」、「発酵」、「レンジ」という語彙が入っている。そして、grammar ファイルでは、注文の形を想定し、voca ファイルで定義した語彙のグループを並べる。

3　シナリオのプログラミング

私のプログラムは二つのクラスを使っている。一つは全体を管理するクラスで、もう一方は、機能を管理するクラスである。機能を管理するクラスでは、主に julius からデータを受け取り、そのデータから注文を読み取る作業をする。その読み取った注文からどの機能を実行すればよいかを判定し、実行するということをしている。具体的にはまず、socket のモジュールを使うこと

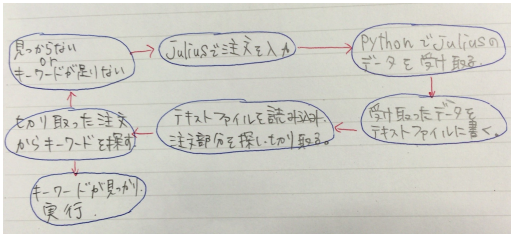

図 4: プログラムの流れ

で、julius の状態が常時データで python に送ることが可能となる。その送られてきたデータをいったんテキストファイルに書き出す。そして、テキストファイルを開き、読み込みをして、注文部分を正規表現を利用して探し、切り取る。　この切り取られた注文から、再び正規表現を利用して、焼き方、温度、時間を探し、if 文を利用して実行の判定をする。このとき、注文から、キーワードが一つも見つからなかったとき、自動モードのキーワード（「温め」など）を探す。自動機能のキーワードが 3 つそろっている場合または、手動機能のキーワードと一致する場合は実行される。キーワードが足りない場合は足りないキーワードを入力し、聞き取れない場合は入力をやり直す仕組みとなっている。

4　実行結果

何回か実行してみた結果、

- 実行が一回ずれてしまう。
- 雑音に反応して、誤認識が多い。

という問題点が見つかった。本プロジェクトでは、実行結果をコンソールに出すところで終わってしまったため、上記した問題点はあまり目立たないが、コンソールには入力をやり直すせと連続的に出力される。当初予定していた、実行結果を音声で出力するとなると、使いにくいことが明らかにわかる。julius から送られるデータから認識時間、正確度を読み取ることで問題点が緩和されるだろう。

参考文献

[1] ”車載情報機器の音声操作における文脈の保持とタイムアウト制御” 木下　悠, 川端　豪, http://ci.nii.ac.jp/naid/110009850962

[2] “julius を用いた音声認識インターフェースの作成”李　晃伸, 河原達也, http://sap.ist.i.kyoto-u.ac.jp/members/kawahara/paper/RI-HIS09.pdf

[3] "The Future Of Voice-Activated AI Sounds Awesome" https://techcrunch.com/2015/03/06/the-future-of-voice-activated-ai-sounds-awesome/