

3 音声認識用文法の作成

Julius では、自作の文法で認識させることができる。作成しなければならないのは、拡張子 `.grammar` のファイルに書く文法規則と、拡張子 `.voca` のファイルに書く辞書項目である。

3.1 辞書の作成

「果物購入タスク」の辞書 `fruit.voca` の冒頭部分は以下のようにになっている。

```
% FRUIT
蜜柑          m i k a N
リンゴ        r i N g o
ぶどう        b u d o:
% NUM
0             z e r o
1             i c h i
2             n i:
```

辞書では、1行で1項目を表わす。フォーマットは、冒頭の「蜜柑」が、認識結果として表示される語の表記である。発音部分では、音素記号を用いて、このエントリの発音を定義する。ここで使える音素記号は、音素モデルが用意されているものだけである。音素記号は、半角スペースで区切る。どのような音素が使えるのか、また、音節(おおよそ、かな一文字に対応する)がどのような音素で表現されるかを簡単に知るには、`SampleGrammars` フォルダにある `type` フォルダにある `type.voca` を見るのがよい。

0のように複数の発音がある語は、発音ごとにエントリを定義する。

```
% NUM
0             z e r o
0             r e:
```

3.2 文法のコンパイル方法

辞書を修正したら、文法をコンパイルし直さなければならない。それらのツールを `cygwin` のシェル上で利用する方法について述べる。

1. まず、`cygwin Bash Shell` (もしくは `cygwin terminal`) を右クリックし、管理者で実行する。
2. `Julius` を展開したトップフォルダ(ディレクトリ)に移動する。

```
$ cd (インストールした場所)/grammar-kit-v4.2-win
$ ls
00readme.txt  eval          HOWTO.txt    SampleGrammars  testmic.jconf
bin           hmm_mono.jconf  model       src              testmod.jconf
doc           hmm_ptm.jconf  sample.wav  testfile.jconf  TROUBLE.txt
```

3. ツールがおかれている `bin` ディレクトリに移動

```
$ cd bin
```

4. (初めての場合) 作業用のディレクトリ `tmp` を作成

```
$ mkdir tmp
```

5. (初めての場合) bin ディレクトリのコマンドを実行可能にする。

```
$ chmod +rx *
```

6. 環境変数 PATH を設定

```
$ PATH=.:$PATH
```

途中に空白を含めてはいけない

7. (初めての場合) mkdfa.pl の 11 行目を編集

```
$usrtmpdir = "";
```

これを次のように変更する。

```
$usrtmpdir = "./tmp";
```

8. 文法のコンパイル

```
$ perl mkdfa.pl ../SampleGrammar/fruit/fruit
```

これで、追加したエントリも認識できるようになったはずである。

練習 4 「果物購入タスク」に果物名を 10 個追加して、認識できるか確認せよ。

練習 5 「果物購入タスク」の辞書にエントリを追加して、「りんごを 3 つ (みつつ) 下さい」というように「つ」を使って個数を数える発話も認識できるようにし、認識できるか確認せよ。

3.3 文法の作成

「果物購入タスク」では、「りんご 1 個です」のように、同時には、一種類の果物しか注文できない。そこで、「りんご 3 個と蜜柑です」というように二種類の果物が同時に注文できるようにしてみる。

```
S      : NS_B FRUIT_N PLEASE NS_E
FRUIT_N : FRUIT
FRUIT_N : FRUIT NUM KO
PLEASE  : WO KUDASAI
PLEASE  : NISHITE KUDASAI
PLEASE  : DESU
```

「りんご 3 個」の部分は、FRUIT_N なので、その部分を 2 回繰り返す文法規則を用意する。

```
S      : NS_B FRUIT_NN PLEASE NS_E
FRUIT_NN : FRUIT_N
FRUIT_NN : FRUIT_N TO FRUIT_N
FRUIT_N  : FRUIT
```

FRUIT_N : FRUIT NUM KO
PLEASE : WO KUDASAI
PLEASE : NISHITE KUDASAI
PLEASE : DESU

TO に関しては、新たに辞書エントリを追加する。
追加したら前節の手順で文法を再コンパイルする。
文法を変更した場合は、上手く変更できているか確認した方がよい。
ツールのディレクトリに、文法を使ってランダムに文を生成するツール `generate` がある。

```
$ generate -n 10 ../SampleGrammars/fruit/fruit
```

とすることで、10 文生成する。

練習 6 「果物購入タスク」の文法を改造して、一度に何種類でも注文できるようにせよ。（「リンゴ 3 個と蜜柑 2 個とぶどう下さい」などが認識できるようにする）

練習 7 自分で適当な文法を設計し、文法と辞書を作成せよ。作成できたら、ランダムに文を生成し、意図通りに作成できたか確認せよ。さらに、実際に発話してみて認識できるか確認せよ。