HTKによる音響モデルの構築

<http://shower.human.waseda.ac.jp/~m-kouki/pukiwiki_public/92.html>

言語モデル

<http://julius.osdn.jp/juliusbook/ja/desc_lm.html>

juliusLibを使おう

<http://d.hatena.ne.jp/kazekyo/20100922/1285161893>

iremoconでミクミクにしてやんよ

<http://futurismo.biz/archives/154>

凹みさんのページ

<http://tips.hecomi.com/entry/20120111/1326296629>

単語辞書の設定方法

**1フィールド：言語エントリ（必須）**

第1フィールドでは，その単語の言語制約の対応エントリを書く．単語N-gram では N-gram 上の語彙，文法では属するカテゴリ番号となる．

#### **第2フィールド：エントリ内確率（オプション）**

単語N-gramの場合，`@' に続けて確率を常用対数(log10)で記述することで， 言語エントリ内の生起確率を追加指定することができる．以下に例を示す．

＜地名＞ @-2.33251 京都+52 [京都] ky o: t o  
＜地名＞ @-1.68893 奈良+52 [奈良] n a r a  
＜地名＞ @-2.63574 和歌山+52 [和歌山] w a k a y a m a

この例では，各単語の出現確率は「＜地名＞」という単語のN-gram 上の出現確率（対数尤度）にエントリ内生起確率を足した値となる．

Julius では，このように，N-gramとしてクラス単位のN-gram を与え，認識用辞書で，単語ごとにその属するクラスとクラス内確率を上記の要領で記述することで，クラスN-gramが実現できる．

#### **第3フィールド：出力文字列（オプション）**

その単語が認識されたときに認識結果に出力する文字列を指定する．値は "[" および "]" で囲まれていること．省略した場合，第1フィールドの値がそのまま出力される．

また，単語N-gramにおいて，このフィールドを "{" "}" 囲みで記述した場合，その単語は「透過語」として扱われる．透過語については以下の節を参照のこと．

#### **以降：音素列**

以降は，その単語の読みを音素列として記述する．音素列は，空白で区切って記述する．

Julius の辞書形式は，単語の複数読みに対応していない．ある単語が複数の読みを持つ場合は，それぞれを異なる単語として別個に登録する．

以下に辞書エントリの例を示す．

課税+1 [カゼイ] k a z e i   
課題+1 [カダイ] k a d a i  
課長+1 [カチョウ] k a ch o:   
課長+1 [カチョウ] k a ch o u   
過ぎ+過ぎる+102 [スギ] s u g i  
過ぎ+過ぎる+114 [スギ] s u g i

### **透過単語の指定について**

試験的機能の一つとして，N-gram 用の辞書において，辞書上の任意の単語を「透過語」に指定することができる．透過語は，N-gram 確率計算時に，コンテキスト上でスキップされる．例えば，文「今日 は ， 良い」の前向き3-gram計算時，透過語指定がない場合は左のように通常の3-gramが計算される．これに対して， 単語「，」を透過語に指定した場合，右側の下2行のように， 「，」が後続の単語のコンテキストから除外されて確率が計算される．このように，コンテキストとして情報が少ない単語を透過語にすることで，実質的なN-gramの距離を稼ぐことができる場合がある．

P(<s> 今日 は　，寒い </s>) P(<s> 今日 は　，寒い </s>)  
 = P(<s> <s> | 今日) = P(<s> <s> | 今日)  
 \* P(<s> 今日 | は) \* P(<s> 今日 | は)  
 \* P(今日 は | ，) \* P(今日 は | ，)  
 \* P(は ， | 寒い) \* P(今日 は | 寒い)  
 \* P(， 寒い | </s>) \* P(は 寒い | </s>)

透過語は，主に句読点やフィラーなどコンテキストと無関係に挿入されやすい単語に対して指定すると効果がある場合がある．透過単語とするには，辞書において第3フィールドを "{" "}" 囲みで記述する．以下は単語「、+、+75」を透過語と指定する例である．

</s> [] silE  
<s> [] silB  
、+、+75 {、} sp  
。+。+74 [。] sp   
？+？+73 [？] sp

### **無音用単語の追加について**

単語間のポーズに対応する単語を辞書に追加できる．単語間のポーズを言語モデル上でモデル化しておらず，辞書にもそれに対応するエントリが存在しない場合，このオプションを指定することで認識率が改善されることがある． 追加する場合は-iwspwordを指定する．また， 追加される単語の内容はオプション-iwspentryで変更できる．

第2フィールドでエントリ内確率を設定することで出てきやすい単語、出てきにくい単語の出現確率をあげることができるかもしれない。

調整する方法としては、似たような発音の単語の登録を避けることなのだが・・・

1.クラス間のN-gram接続確率・・・・N-gramファイル

2.クラス内の単語生起確率・・・・単語辞書内に記述

今回は2番目の単語生起確率を設定したいため、この方法について調べる。

クラス内単語生起確率は、各クラスに属する単語については、そのクラス内での生起確率と読みなどの定義を単語辞書上で定義する。まず通常の単語N-gramの場合は、辞書中の単語エントリは以下のようになっている。

　単語表記　[出力文字列]　音素列・・・

（例）

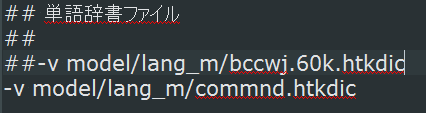
クラスN-gramを用いる場合は、これらの前に「属するクラス名」、「クラス内単語生起確率」を以下のように記述する。

クラス間N-gram上のクラス名　＠クラス内単語生起確率の対数値　単語表記・・・

第1カラム目の「クラス名」は、クラス間のN-gramでのクラス名を表記する。

第2カラム目の「クラス内単語生起確率の対数値」は、その単語のクラス内での生起確率を常用対数（log10）で指定する。

認識実行時には、このようにして作成したクラス付き単語辞書を、通常と同様に-ｖで指定します。



形式はjulius側で自動判別される。もちろん、同時に使用するクラスN-gramと第1カラム目の名前の対応がとれている必要がある。

### 部分クラスN-gramについて

Juliusの内部では，単語N-gramとクラスN-gramを明示的に区別しておらず，単語N-gramはすべて「1クラス1単語からなるクラスN-gram」として処理しています．このため，単語エントリとクラスエントリを混在させることが可能です．たとえば単語N-gramにおいて一部の品詞カテゴリのみをクラス化する，といったような部分クラスN-gramを使用することができます．

このような部分クラスN-gramを使用する場合，単語N-gramのエントリは通常と同じように単語辞書で記述し，クラス化されたエントリについてのみ，上記のようにクラスの情報を記述します．以下は，地名が「＜地名＞」でクラス化されている部分クラスN-gramを想定した場合の辞書の記述例です．

今日+42 [今日] ky o:  
は+16 [は] w a  
＜地名＞ @-2.33251 京都+52 [京都] ky o: t o  
＜地名＞ @-1.68893 奈良+52 [奈良] n a r a  
＜地名＞ @-2.63574 和歌山+52 [和歌山] w a k a y a m a  
出張+6 [出張] sh u q ch o:  
です+67 [です] d e s u