OGSS：認識辞書作成について

Kazutaka Shimada, Kyushu Institute of Technology
2012/01/20

この文書について

■動作には音声認識エンジンJuliusが必要です.

認識用辞書の準備

はじめに

■OGSSでは3種類（大規模・中規模・小規模）の認識器を利用します.

■大規模認識器

Juliusに標準で添付されている統計的言語モデルをご利用ください

■中規模認識器

この認識器の利用は任意です．ただし，中規模認識器を用意する方が，照応解析の精度が向上することは実験的に確認されています.

■小規模認識器

タスクに応じて，適当なサイズで文法型の言語モデルをご用意ください.

■音声認識結果の解析のためにも，別途3種類（文型辞書・単語辞書・意味階層辞書）の辞書が必要です.

小規模認識器のための辞書について

■インストールされたJuliusのフォルダに，OGSSで使用する小規模認識器用の辞書を設置する必要があります.

■フォルダの名前は数字とし，1から始めます.

小規模認識器は複数設置できます．適当な語彙サイズ・文法で切り分けると精度が向上します.
添付のサンプルでは，介護施設で利用されるロボットを想定し，（1）患者からの発話を対象とした文法と語彙，（2）看護士からの発話を対象とした文法と語彙，（3）ロボットを制御するための文法と語彙，（4）質問文を対象とした文法と語彙，というようなレベルで作成されています.

■各フォルダ内に，Juliusが使用する dfa ファイルと dict ファイルを格納してください.

方法は2つあります

1．Julius形式の文法・語彙辞書を利用する

> Juliusの書式で文法・語彙辞書を作成した場合は，Julius付属の文法コンパイラを利用してください．http://julius.sourceforge.jp/index.php?q=doc/grammar.html

2．知能化プロジェクトの共通規格の辞書を利用する

> 共通規格の書式で文法・語彙辞書を作成した場合は，OpenHRIに付属しているプログラムを利用します．OpenHRIのソースファイルをダウンロードし，pyJuliusRTC以下にあるparsesrgs.pyを利用します．このparsesrgs.pyとOGSSに付属のmkDfaDic.pyを利用すると dfa フ

ァイルと dict ファイルが生成されます．具体的には，コマンドラインで以下のように実行してください

```
./parsesrgs.py GRAMMAR.XML | ./mkDfaDic.py idNum
```

ここで GRAMMAR.XML はコンパイルしたい共通規格の文法ファイルであり，idNumは，各辞書のフォルダ名となっている数字を指定してください．

中規模認識器用の辞書ファイルの設置

■中規模認識器は，照応解析（文中の「それ」や「あれ」，省略表現が何なのかを過去の履歴から推定）の精度向上のために必要です．

設置しなくてもOGSSおよび照応解析自体は動作しますが，照応解析の精度は低下します．

■中規模認識器には，（1からナンバリングされた）すべての認識器に含まれる単語を利用することをお勧めします．

■サンプル（Julius以下のmediumTypeフォルダ内）を基に作成してください．

サンプルは文法ベースですが，中規模認識器に統計的言語モデルを利用することも可能です．

サンプルを利用すれば，語彙に関する辞書のみを変更するだけで実装できます．

■以下は mediumType.grammar を変更せずに，voca ファイルのみを変更して利用する場合の手順です．

mediumType.vocaの内容を各自の目的に合わせて変更します．

vocaファイルの作成方法については，Juliusの仕様に沿ってください．

簡単に説明しますと単語の表記と音素を併記します．

ファイル内の % で始まる行は修正しないでください．

% はJuliusの辞書の書式で，単語のカテゴリを表します．

ある % から次の % までが一つのカテゴリに属する語です．

% agt

人名に関する単語を列挙します．

% obj

現在対象としている環境に存在する物体の名前を列挙します．

% loc

現在対象としている環境に存在する場所に関する単語を列挙します．

% connect1 と connect2

助詞などの機能語を必要であれば追加してください．

% other

動詞や述語に相当する単語を列挙してください．

vocaファイルの修正が終わったら，Julius付属の文法コンパイラで dfa と dict ファイルを生成します．

■必要に応じて mediumType.grammar を書きかえると照応解析の精度向上に繋がる可能性があります．

■十分な対話コーパスがあるのであれば，統計的言語モデルで中規模認識器を実行するとさらに精度向上が期待できます．

音声認識結果の解析辞書の準備

■照応解析処理には，文の解析のためにいくつかの辞書ファイルが必要です．

具体的には frame.xml，voca.xml，tree.xml です．これらがないと照応解析処理は実行できません．

■生活支援ロボット用のサンプルを添付していますので，それを参考に，利用環境・文法・語彙に合わせて修正してください．

frame.xml は文を述語ベースで解析するための辞書です．

```
<frame case="predicate" semantic="drink_V" semanticSub="飲みたい"
            surface="(飲みたい|飲ん)" required="obj" requiredSub="loc">
<case case="agt" semantic="person" marker="((*)さん|(*)くん)" />
<case case="agt" semantic="person" marker="(彼女|彼)" />
<case case="loc" semantic="loc" marker="(*)の(上|中|下|横|手前|奥)の" />
<case case="loc" semantic="loc" marker="(*)の(上|中|下|横|手前|奥)にある" />
<case case="loc" semantic="loc" marker="(ここ|そこ|あそこ)" />
<case case="obj" semantic="drink_N" marker="(*)(を|が)" />
<case case="obj" semantic="drink_N" marker="それ(を|が)" />
</frame>
```

case="predicate" がキーとなる述語です．
semantic はその述語の意味を表します．
semanticSub はその述語の一般型を記述してください．
surface はその述語の実際の出現のパターンを | で列挙します．
required および requiredSub は照応解析の際に探索をする要素と優先順位を表しています．
各caseにおける，caseには格の名前が入ります．中規模認識器を作成したときの基準を参考に格を決めてください．
semantic は解析時に利用する各要素の意味的属性を記述します．（後述のvoca.xmlと対応づける必要があります）
marker は，文から要素を抽出するために利用する表層表現のパターンです．* で指定された部分が抽出される単語になります．たとえば，田中さんが....という文があった場合，パターン (*)さん とマッチし，「田中」が agt として抽出されます．
例えば，このframe辞書では，「テーブルの上にあるジュースが飲みたいのだけど」のような文が解析できます．

voca.xml は 単語の意味属性をまとめたものです．
照応解析時の候補の絞り込みに利用します．
意味属性ごとに <semantic> で囲みます．

```
    <semantic semantic="drink_N">
```

```
    <item>コーヒー</item>
    <item>お茶</item>
    <item>ジュース</item>
  </semantic>
  <semantic semantic="sweets">
    <item>お菓子</item>
    <item>チョコレート</item>
    <item>チョコ</item>
  </semantic>
```

上記の例では，「ジュース」や「コーヒー」は「飲み物（drink_N）[*1]」で，「チョコレート」は「お菓子類（sweets)」という意味属性を持っていることを意味します．
例えば，「それを飲みたい」という文があり，候補として「チョコレート」と「ジュース」があった場合，意味素性の制約により「ジュース」が優先されます．

tree.xml は voca.xml で記述した意味属性の階層関係を記述するものです．
parタグが親を表し，chiはその子を表します．

```
<th par="eat_N" chi="sweets"/>
```

この場合，sweetsの親にあたる意味素性がeat_Nであることを表しています．
例えば，「それを食べたい」という入力があった場合，「それ」の先行詞は，eat_N以下の意味素性をもつ単語である，というような知識として利用されます．

補足：OGSSの設定

OGSSは適当なフォルダに設置してください．
OGSSに関する設定ファイルは ogss.conf です．
DSSRとは用意した複数の認識器です（1からナンバリングされたもの)．
midDSSRとは中規模認識器を指します．
-anaphora は，照応解析を有効にするかのオプションです．
各種閾値は，認識器の選択処理で利用します．必要であれば調整してください．
-semFeature は，照応解析精度を向上させるために利用する辞書に関するオプションです．
-sp 以降に Julius の設置フォルダのフルパスを記入してください．
-file には，Julius が読み込むconfファイルの場所を指定してください．
この conf ファイルについては，サンプルを参考に必要であれば修正を加えてください．
-micconf オプションには，RTCの環境に合わせて適宜適切なオプションを指定してください．
ファイルに関する設定は，ローカル環境で精度評価や確認に利用するためのものなので，特に変更する必要はありません．
出力は output.xml に共通規格に沿ったXML形式で書き込まれます．
この出力は発話ごとに上書きされます．

---

*1 drink_Nという表記自体には意味がありません．ここで定義する意味素性と frame.xml で定義した意味素性と対応付いていれば，意味素性名は何でも構いません．