Assignment 2

Building an ASR System using PocketSphinx

CS4706

March 26, 2012

PocketSphinx

- Small-footprint continuous ASR system based on CMUSphinx
- Suitable for mobile devices
- Open-source, cross-platform
- Trigram and finite-state grammar language models
- Python language bindings

Part 1: Building a Grammar

- Write a grammar that handles your input domain.
- Record and submit 5 sentences that are in your grammar.
- Augment the pronunciation dictionary.
- Pick the best acoustic model.

JSGF Grammar Format

- Variables go in angle brackets, e.g. <city>
- Terminals must appear in your pronunciation dictionary (case sensitive)
- X Y is concatenation (e.g. I WANT)
- (X | Y) means X or Y - e.g., (WANT|NEED)
- Square brackets mean optional, (e.g., [ON] FRIDAY)
- Kleene star means that the expansion may be spoken zero or more times, e.g. <digit>*
- Plus operator means that the expansion may be spoken one or more times, e.g. <digit>+

JSGF Grammar Example

```
#JSGF V1.0;
grammar travel;

<city> = BOSTON | NEWYORK | WASHINGTON | BALTIMORE;
<time> = MORNING | EVENING;
<day> = FRIDAY | MONDAY;

public <query> = (((WHAT TRAINS LEAVE) | (WHAT TIME CAN I TRAVEL) |
                  (IS THERE A TRAIN)) (FROM|TO) <city>
                  [(FROM|TO) <city>] ON <day> [<time>]);
```

Pronunciation Dictionary

Sphinx uses the ARPAbet phoneset.

```
ELEVEN          AX L EH V AX N
ELEVEN(2)       IY L EH V AX N
EXIT            EH G Z AX T
EXIT(2)         EH K S AX T
EXPLORE         IX K S P L AO R
FIFTEEN         F IH F T IY N
```

Copy the default dictionary into your project directory, and add any missing words to it.

Acoustic Models

We provide you with 7 possible acoustic models to try:

- Default acoustic model trained on the Wall Street Journal corpus.
- HUB4 Broadcast News, 4000 senones
- HUB4 Broadcast News, 6000 senones
- WSJ, 8000 senones, 1 gaussian
- WSJ, 8000 senones, 4 gaussians
- WSJ, 8000 senones, 16 gaussians
- WSJ, 8000 senones, 256 gaussians

Try each of them with your five test utterances, and pick the one that gives the best **concept accuracy**.

Running the Recognition Script

Run: /proj/speech/users/cs4706/pasr/recognize_wav.py <your_wav_file> -g <your_grammar_file> -d <your_dictionary> -a <1-7>

- Your sample .wav file
- -g: your grammar file - required
- -d: your dictionary file - required only if your grammar contains words not in the default dictionary
- -a: which acoustic model (1-7) - optional; default is 1

The script will show you some output from Sphinx, with the recognized sentence at the end.

Part 2: Building a Concept Table

Write a script that takes in a .wav file, gets ASR output, and turns the ASR output into a concept table.
Example: `./recognize_concepts.py test/test2.wav`
Output:

```
Departure city:     Boston
Destination:        New York
Day:                Friday
Time:               UNSPECIFIED
```

Writing Your Concept Recognition Script

- /proj/speech/tools/pocketsphinx/example/
- example.py
- example.c