Voice Search

Opportunities, Challenges, and Methods

8 October 2014
PROPOR 2014 - Sao Carlo, Brazil
Opportunities

Speech is core to the future of mobile
Evolution of Search

10 blue links

more than just keywords

mobile, voice
Voice search with spoken answers
Converse naturally with Google, on any platform
Accelerate how you get information & get things done
Challenges and Methods

It has to work everywhere, every time

How we’re getting there.
Progress in Speech Recognition

2006

2011

Now You're Talking!

Google has developed speech-recognition technology that actually works.

By Farhad Manjoo
Google Speech Recognition Levels

- Feature Extraction
- Acoustic Model
- Audio
- Frame
- State
- Phoneme
- Word
- Sentence
- Language Model
- Decision Trees
- Lexicon
- NLP
- Meaning
$$\text{argmax } p(W | O) = \text{argmax } p(O | W) p(W)$$
Weighted Finite State Transducers (FSTs)

Challenge: How to efficiently represent the very large phonetic, lexical, and language models needed for Voice Search and other LVR tasks.

WFSTs one way to express this as probabilistic transductions.

A mathematically sound way to express probabilistic graphs and algorithms over them. (e.g. Viterbi, forward-backward)

Powerful algorithms to combine and optimize these graphs.

Graphical representation:

Language Model

- Toy Example:
  - “call don”
  - “call dad”

- In reality:
  - 100M+ states
  - estimated using billions of training examples.
call: k ao l
dad: d ae d
mom: d aa n
Cascading Transductions: Composition

- Map output labels of Lexicon to input labels of Language Model.
- Join and optimize end-to-end graph.

Phoneme → Word → Sentence
Optimization via Determinization

- Redundancy causes excessive search
- Determinization creates an equivalent transducer that has no two transitions from a state with the same label

Other operations: Minimization, Epsilon removal, Weight pushing.

Uses our open-source OpenFst: www.openfst.org
Context-Dependent Phonemes

- Acoustic models of phonemes built in **context** (typically triphonic): \( d_{aa}n \): /aa/ preceded by /d/ and followed by /n/
- Application of context-dependency to the decoder graph very naturally implemented by composition with a **context-dependency** finite-state transducer:

```
d aa
```

```
aa
```

```
d_{aa}n / 1.0
```

```
aa n
```