OpenFst: An Open-Source, Weighted Finite-State Transducer Library and its Applications to Speech and Language

Introduction

Cyril Allauzen - allauzen@google.com
Martin Jansche - mjanchse@google.com
Michael Riley - riley@google.com

May 31, 2009

Thanks to Boulos Harb, Johan Schalkwyk, Masha Shugrina, Mehryar Mohri, Richard Sproat, and Wojtek Skut.
OpenFst Library

• C++ template library for constructing, combining, optimizing, and searching *weighted finite-states transducers (FSTs)*.

• **Goals:** Comprehensive, flexible, efficient and scale well to large problems.

• **Origins:** AT&T, merged efforts from Google and the NYU Courant Institute.

• **Documentation and Download:** http://www.openfst.org

• Released under the Apache license.
Weighted Acceptors

- Finite automata with labels and weights.
- Example: *Pronunciation model acceptor:*

```
0 --> 1: d/1
1 --> 2: ey/0.5, ae/0.5
2 --> 3: t/0.3, dx/0.7
3 --> 4: ax/1
```
Weighted Transducers

- Finite automata with input labels, output labels, and weights.

- Example: *Pronunciation lexicon transducer:*

```
0  d:data/1
   ↓ d:dew/1

1  d: data/1
   ↓ ae:ε/0.5

2  ey:ε/0.5
   ↓ t:ε/0.3

3  dx:ε/0.7
   ↓ ax:ε/1

4  4/0

5  uw:ε/1

6  6/0
```
Motivation

- **Finite-State Acceptors**: Compact representations of regular (rational) sets that are efficient to search. Examples: pattern matching (grep, PCRE), tokenization, compression.

- **Finite-State Transducers**: Compact representations of rational binary relations that are efficient to search and combine/cascade. Examples: dictionaries, context-dependent rules

- **Weighted Automata**: Weights typically encode uncertainty as e.g. probabilities. Examples: n-gram language models, language translation models.
References

• General Background:

• Theory of Transductions and Rational Power Series:

• Transducers Applied to Speech and NLP:
Current OpenFst Applications

• **Speech recognition (speech-to-text):** lexicons, language models, phonetic context-dependency, recognizer hypothesis sets.

• **Speech synthesis (text-to-speech):** tokenization, text normalization, pronunciation models

• **Optical character recognition:** lexicons, language models

• **Machine Translation:** translation models, language model, translation hypothesis sets.

• **Information extraction:** pattern matching, text processing
Overview

1. Part I: Algorithms - Cyril Allauzen
2. Part II: Library Use and Design - Michael Riley
3. Part III: Applications - Martin Jansche