

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

Michael Riley, Cyril Allauzen, and Martin Jansche, Google Inc.

Finite-state methods are well established in language and speech processing. OpenFst (available from www.openfst.org) is a free and open-source software library for building and using finite automata, in particular, weighted finite-state transducers (FSTs). This tutorial is an introduction to weighted finite-state transducers and their uses in speech and language processing. While there are other weighted finite-state transducer libraries, OpenFst (a) offers, we believe, the most comprehensive, general and efficient set of operations; (b) makes available full source code; (c) exposes high- and low-level C++ APIs that make it easy to embed and extend; and (d) is a platform for active research and use among many colleagues.

4.1 Tutorial Outline

1. Introduction to OpenFst

The first part of the tutorial introduces operations on weighted automata such as determinization and intersection/composition as well as the corresponding OpenFst binaries and library-level APIs that implement those operations. We describe how to read FSTs from simple textual descriptions and combine them into larger and more complex machines, and optimize them using simple command-line and library calls.

- Introduction
 - Motivating examples
 - Finite-state methods in NLP and Speech
- A quick tour of OpenFst
 - Finite-state machines and operations
 - OpenFst binaries
 - High-level C++ API
 - Human-readable file formats
- Comparison of OpenFst and competing libraries
 - Comparison with the AT&T FSM Library™
 - Brief comparison with SFST and related libraries
- Advanced usage of OpenFst
 - Low-level C++ API
 - Constructing and modifying Fst objects programmatically
 - Implementing new concrete Fst classes
 - Adding new weight semirings
 - Customizing matchers and filters for composition

2. Applications

The second part of the tutorial focuses on several application areas of interest to the NAACL HLT audience, including speech recognition, speech synthesis, and general text processing. In each application area we discuss a straightforward example in detail, then delve into an advanced example that highlights important features of OpenFst, common pitfalls, efficiency considerations, new research directions, etc. These examples are drawn from our extensive experience in applying OpenFst to problems in speech and language processing.

- Automatic speech recognition
 - Context dependency transducers
 - Language models and grammars
- Natural Language Processing
 - Unicode processing
 - Text analysis, text normalization
 - Pronunciation models
- Other areas
 - Computational biology
 - Pattern matching

4.2 Target Audience

This tutorial is intended for students, researchers, and practitioners interested in applying finite-state methods. It is suitable for participants of a variety of backgrounds, including those without any background in finite-state techniques as well as advanced users of existing software packages. For those unfamiliar with finite-state transducers, the first portion of the tutorial provides an introduction to the theory and application areas. Users of other finite-state libraries will particularly benefit from the contrastive description of the unique features of OpenFst, its C++ API, and the examples drawn from real-world applications.

4.3 Presenters

Michael Riley began his career at Bell Labs and AT&T Labs where he, together with Mehryar Mohri and Fernando Pereira, introduced and developed the theory and use of weighted finite-state transducers (WFSTs) in speech and language. This work was recognized in best paper awards from the journals *Speech Communication* and *Computer Speech and Language*. He has been a research scientist at Google, Inc. since 2003. He is a principal author of the OpenFst library and the AT&T FSM LibraryTM. He has given several tutorials on WFSTs before: at ACL 1994, Coling 1997 and Interspeech 2002.

Cyril Allauzen is another key author of the OpenFst library. His main research interests are in finite-state methods and their applications to text, speech and natural language processing and machine learning. Before joining Google, he worked as a researcher at AT&T Labs – Research and at NYU’s Courant Institute of Mathematical Sciences.

Martin Jansche has applied the OpenFst library to several speech and language problems at Google. His FST-related interests are in text processing for speech tasks and in learning and applying pronunciation and transliteration models.