

Finite-State Text Processing

Synthesis Lectures on Human Language Technologies

Editor

Graeme Hirst, *University of Toronto*

Synthesis Lectures on Human Language Technologies is edited by Graeme Hirst of the University of Toronto. The series consists of 50- to 150-page monographs on topics relating to natural language processing, computational linguistics, information retrieval, and spoken language understanding. Emphasis is on important new techniques, on new applications, and on topics that combine two or more HLT subfields.

Finite-State Text Processing

Kyle Gorman and Richard Sproat
2021

Embeddings in Natural Language Processing: Theory and Advances in Vector Representations of Meaning

Mohammad Taher Pilehvar and Jose Camacho-Collados
2020

Conversational AI: Dialogue Systems, Conversational Agents, and Chatbots

Michael McTear
2020

Natural Language Processing for Social Media, Third Edition

Anna Atefeh Farzindar and Diana Inkpen
2020

Statistical Significance Testing for Natural Language Processing

Rotem Dror, Lotem Peled, Segev Shlomov, and Roi Reichart
2020

Deep Learning Approaches to Text Production

Shashi Narayan and Claire Gardent
2020

Linguistic Fundamentals for Natural Language Processing II: 100 Essentials from
Semantics and Pragmatics

Emily M. Bender and Alex Lascarides
2019

Cross-Lingual Word Embeddings

Anders Søgaard, Ivan Vulić, Sebastian Ruder, Manaal Faruqi
2019

Bayesian Analysis in Natural Language Processing, Second Edition

Shay Cohen
2019

Argumentation Mining

Manfred Stede and Jodi Schneider
2018

Quality Estimation for Machine Translation

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold
2018

Natural Language Processing for Social Media, Second Edition

Atefeh Farzindar and Diana Inkpen
2017

Automatic Text Simplification

Horacio Saggion
2017

Neural Network Methods for Natural Language Processing

Yoav Goldberg
2017

Syntax-based Statistical Machine Translation

Philip Williams, Rico Sennrich, Matt Post, and Philipp Koehn
2016

Domain-Sensitive Temporal Tagging

Jannik Strötgen and Michael Gertz
2016

Linked Lexical Knowledge Bases: Foundations and Applications

Iryna Gurevych, Judith ECKLE-Kohler, and Michael Matuschek
2016

Bayesian Analysis in Natural Language Processing

Shay Cohen
2016

Metaphor: A Computational Perspective

Tony Veale, Ekaterina Shutova, and Beata Beigman Klebanov
2016

Grammatical Inference for Computational Linguistics

Jeffrey Heinz, Colin de la Higuera, and Menno van Zaanen
2015

Automatic Detection of Verbal Deception

Eileen Fitzpatrick, Joan Bachenko, and Tommaso Fornaciari
2015

Natural Language Processing for Social Media

Atefeh Farzindar and Diana Inkpen
2015

Semantic Similarity from Natural Language and Ontology Analysis

Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain
2015

Learning to Rank for Information Retrieval and Natural Language Processing, Second Edition

Hang Li
2014

Ontology-Based Interpretation of Natural Language

Philipp Cimiano, Christina Unger, and John McCrae
2014

Automated Grammatical Error Detection for Language Learners, Second Edition

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault
2014

Web Corpus Construction

Roland Schäfer and Felix Bildhauer
2013

Recognizing Textual Entailment: Models and Applications

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto
2013

Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax

Emily M. Bender
2013

Semi-Supervised Learning and Domain Adaptation in Natural Language Processing

Anders Søgaard

2013

Semantic Relations Between Nominals

Vivi Nastase, Preslav Nakov, Diarmuid Ó Séaghdha, and Stan Szpakowicz

2013

Computational Modeling of Narrative

Inderjeet Mani

2012

Natural Language Processing for Historical Texts

Michael Piotrowski

2012

Sentiment Analysis and Opinion Mining

Bing Liu

2012

Discourse Processing

Manfred Stede

2011

Bitext Alignment

Jörg Tiedemann

2011

Linguistic Structure Prediction

Noah A. Smith

2011

Learning to Rank for Information Retrieval and Natural Language Processing

Hang Li

2011

Computational Modeling of Human Language Acquisition

Afra Alishahi

2010

Introduction to Arabic Natural Language Processing

Nizar Y. Habash

2010

Cross-Language Information Retrieval

Jian-Yun Nie

2010

[Automated Grammatical Error Detection for Language Learners](#)
Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault
2010

[Data-Intensive Text Processing with MapReduce](#)
Jimmy Lin and Chris Dyer
2010

[Semantic Role Labeling](#)
Martha Palmer, Daniel Gildea, and Nianwen Xue
2010

[Spoken Dialogue Systems](#)
Kristiina Jokinen and Michael McTear
2009

[Introduction to Chinese Natural Language Processing](#)
Kam-Fai Wong, Wenjie Li, Ruifeng Xu, and Zheng-sheng Zhang
2009

[Introduction to Linguistic Annotation and Text Analytics](#)
Graham Wilcock
2009

[Dependency Parsing](#)
Sandra Kübler, Ryan McDonald, and Joakim Nivre
2009

[Statistical Language Models for Information Retrieval](#)
ChengXiang Zhai
2008

© Springer Nature Switzerland AG 2022

Reprint of original edition © Morgan & Claypool 2021

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means—electronic, mechanical, photocopy, recording, or any other except for brief quotations in printed reviews, without the prior permission of the publisher.

Finite-State Text Processing

Kyle Gorman and Richard Sproat

ISBN: 978-3-031-01051-4 paperback

ISBN: 978-3-031-02179-4 ebook

ISBN: 978-3-031-00190-1 hardcover

DOI 10.1007/978-3-031-02179-4

A Publication in the Springer series

SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES

Lecture #50

Series Editor: Graeme Hirst, *University of Toronto*

Series ISSN

Print 1947-4040 Electronic 1947-4059

Finite-State Text Processing

Kyle Gorman

Graduate Center, City University of New York

Richard Sproat

Google LLC

SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES #50

ABSTRACT

Weighted finite-state transducers (WFSTs) are commonly used by engineers and computational linguists for processing and generating speech and text. This book first provides a detailed introduction to this formalism. It then introduces Pynini, a Python library for compiling finite-state grammars and for combining, optimizing, applying, and searching finite-state transducers. This book illustrates this library's conventions and use with a series of case studies. These include the compilation and application of context-dependent rewrite rules, the construction of morphological analyzers and generators, and text generation and processing applications.

KEYWORDS

automata, finite automata, finite-state automata, finite-state transducers, grammar development, language processing, speech processing, state machines, text generation, text processing, Python, Pynini

Contents

	Preface	xv
	Acknowledgments	xvii
1	Finite-State Machines	1
1.1	State Machines	2
1.2	Formal Preliminaries	4
1.2.1	Sets	4
1.2.2	Relations and Functions	5
1.2.3	Strings and Languages	5
1.3	Acceptors and Regular Languages	6
1.3.1	Finite-State Acceptors	6
1.3.2	Regular Languages	7
1.3.3	Regular Expressions	8
1.4	Transducers and Rational Relations	8
1.4.1	Finite-State Transducers	8
1.4.2	Rational Relations	9
1.5	Weighted Acceptors and Languages	10
1.5.1	Monoids and Semirings	10
1.5.2	Weighted Finite Acceptors	12
1.5.3	Weighted Regular Languages	13
1.6	Weighted Transducers and Relations	13
1.6.1	Weighted Finite Transducers	14
1.6.2	Weighted Rational Relations	14
2	The Pynini Library	17
2.1	Design	17
2.2	Conventions	18
2.2.1	Copying	19
2.2.2	Labels	19
2.2.3	States	19
2.2.4	Iteration	19

	2.2.5	Weights	20
	2.2.6	Properties	20
2.3		String Conversion	21
	2.3.1	Text Encoding	21
	2.3.2	String Compilation	24
	2.3.3	String Printing	26
2.4		File Input and Output	28
2.5		Alternative Software	29
3		Basic Algorithms	31
	3.1	Concatenation	32
	3.2	Closure	33
	3.3	Range Concatenation	34
	3.4	Union	34
	3.5	Composition	35
	3.6	Difference	38
	3.7	Cross-Product	38
	3.8	Projection	39
	3.9	Inversion	40
	3.10	Reversal	41
4		Advanced Algorithms	43
	4.1	Optimization	43
	4.2	Shortest Distance	44
	4.3	Shortest Path	46
5		Rewrite Rules	49
	5.1	The Formalism	49
		5.1.1 Directionality	52
		5.1.2 Boundary Symbols	53
		5.1.3 Generalization	54
		5.1.4 Abbreviatory Devices	54
		5.1.5 Constraint-Based Formalisms	56
	5.2	Rule Compilation	56
		5.2.1 The Algorithm	57
		5.2.2 Efficiency Considerations	59

5.2.3	Rule Compilation in Pynini	59
5.3	Rule Application	60
5.3.1	Lattice Construction	60
5.3.2	String Extraction	60
5.3.3	Rewriting Libraries	62
5.4	Rule Interaction	62
5.4.1	Two-Level Rules	62
5.4.2	Cascading	63
5.4.3	Exclusion	64
5.5	Examples	67
5.5.1	Spanish Grapheme-to-Phoneme Conversion	67
5.5.2	Finnish Case Suffixes	71
5.5.3	Currency Expression Tagging	73
6	Morphological Analysis and Generation	77
6.1	Applications	78
6.2	Word Formation	80
6.3	Features	81
6.4	Paradigms	82
6.5	Examples	82
6.5.1	Russian Nouns	82
6.5.2	Tagalog Infixation	86
6.5.3	Yowlumne Aspect	88
6.5.4	Latin Verbs	91
7	Text Generation and Processing	93
7.1	Fuzzy String Matching	93
7.2	Date Tagging	96
7.3	Number Naming	97
7.4	Chatspeak Normalization	98
7.5	T9 Disambiguation	101
7.6	Weather Report Generation	102
8	The Future	105
8.1	Hybridization	106
8.2	Hardware Customization	107
8.3	Subregular Grammar Induction	108

A	Pynini Installation	109
	A.1 Anaconda Installation	109
	A.2 Source Installation	109
	A.3 Optional Dependencies	111
B	Pynini Bracket Parsing	113
C	Pynini Extended Library	115
D	Pynini Examples Library	117
E	Pynini Export Library	119
	Bibliography	121
	Authors' Biographies	137
	Index	139

Preface

This book is our attempt to provide a “one-stop” reference for engineers and linguists interested in using finite-state technologies for text generation and processing. As such, it begins with formal language and automata theory, topics covered in much greater detail by textbooks such as [Hopcroft et al. 2008](#) and handbook chapters such as [Mohri 2009](#). In our experience, full command of finite-state technologies requires familiarity with a number of matters that have not received much attention in prior literature. Among these topics is the theory of semirings, and algorithms specific to weighted automata such as the shortest-distance and shortest-path algorithms. These formalisms and algorithms are key for finite-state speech recognition. Furthermore, there exist many text processing applications that resemble weighted finite-state-based speech recognition insofar as hypotheses—that is, possible output strings—are represented as paths through a lattice constructed via composition of weighted automata, and inference/decoding involves computing the shortest path.

Users interested in text applications also stand to benefit from lesser-known “tricks of the trade” for finite-state development. These tricks include fuzzy string matching ([Figure 7.1](#)), efficient algorithms for optimizing arbitrary weighted finite-state transducers ([section 4.1](#)), compiling rewrite rules ([section 5.2](#)) and morphological analyzers and generators ([chapter 6](#)), and applying these transducers to sets of strings ([section 5.3](#)).

At the same time, we wish to go beyond algebraic formalisms and pseudocode. Thus, we illustrate our examples with Pynini, an open-source Python library for weighted finite-state transducers developed at Google. Still, we are skeptical that anything made out of dead trees is an appropriate medium for documenting a rapidly changing software library. So whereas earlier texts like *Finite State Morphology* ([Beesley and Karttunen 2003](#)) are in some sense *about* the Xerox finite-state toolkit as it existed at the time, we hope that this is not merely a book about Pynini. It is our hope that this melange of formalisms and algorithms, code and applications, meets the needs of our readers.

Finally, in the current age we would be remiss if we did not stress the importance of ethical use of this—or indeed any—technology. Ten years ago, [Sproat \(2010a:255\)](#) pointed out the potential dangers for society of language technology and its misuse, especially on social media platforms, noting that “language can be abused, and so can the technology that supports it”. The recent rise in disinformation on social media has unfortunately made those concerns seem all too prophetic. The ongoing pandemic, aggravated in large part by disinformation, has brought these dangers into even starker relief. It is therefore our profound hope that the technology described in this book only be used for the betterment of humankind. One example of this sort suggests itself: [Markov et al. \(2021\)](#) describe how regular expression matching is used

xvi PREFACE

to determine whether a post on social media mentions COVID-19 so it can be screened for disinformation.

Kyle Gorman and Richard Sproat
April 2021

Acknowledgments

We first owe an enormous debt to the many Google engineers who have contributed over the years to the OpenFst and OpenGrm libraries, particularly Cyril Allauzen, Brian Roark, Michael Riley, and Jeffrey Sorensen. Substantial improvements to the Pynini library have been made by Lawrence Wolf-Sonkin, and this book has greatly benefited from the user community of Google linguists, especially Sandy Ritchie. Thanks to Anssi Yli-Jyrä and an anonymous reviewer for their detailed reviews; to Jeffrey Heinz for detailed feedback on our pre-final draft; to Alëna Aksënova, Hossep Dolatian, Jordan Kodner, Constantine Lignos, Fred Mailhot, and Arya McCarthy, who provided useful comments on early drafts of the book; and to Chandan Narayan for notes on Pāṇini.

Kyle Gorman and Richard Sproat
April 2021