

University of Groningen

Finite-state pre-processing for natural language analysis

Prins, Robbert

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2005

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Prins, R. P. (2005). Finite-state pre-processing for natural language analysis s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

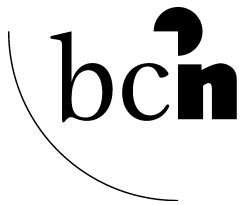
Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Finite-State Pre-Processing for Natural Language Analysis

Robbert Prins



The work in this thesis has been carried out under the auspices of the Behavioral and Cognitive Neurosciences (BCN) research school, Groningen, and has been part of the PIONIER project *Algorithms for Linguistic Processing*, supported by grant number 220-70-001 from the Netherlands Organization for Scientific Research (NWO).



Groningen Dissertations in Linguistics 53

ISSN 0928-0030

Document prepared with L^AT_EX

Printed by PrintPartners Ipskamp, Enschede

RIJKSUNIVERSITEIT GRONINGEN

Finite-State Pre-Processing for
Natural Language Analysis

Proefschrift

ter verkrijging van het doctoraat in de
Letteren
aan de Rijksuniversiteit Groningen
op gezag van de
Rector Magnificus, dr. F. Zwarts,
in het openbaar te verdedigen op
donderdag 12 mei 2005
om 14.45 uur

door

Robbert Paul Prins

geboren op 18 december 1977
te Ten Boer

Promotor: Prof. dr. ir. J. Nerbonne

Copromotor: Dr. G.J.M. van Noord

Beoordelingscommissie: Prof. dr. E.W. Hinrichs
Prof. dr. M. Johnson
Prof. dr. G. Satta

Acknowledgements

At the end of my research and the beginning of this thesis I would like to thank a number of people for helping me.

To start with, I would like to thank my supervisor Gertjan van Noord and my second supervisor and promotor John Nerbonne. Discussing topics with Gertjan always helped when I was stuck or in need of some inspiration. John also helped me out by reading and commenting on the chapters as I was writing them. Thanks go out to the members of my reading committee for taking upon them this task of reading, judging, and commenting on the whole of the thesis.

Then I would like to thank my other colleagues at Alfa-Informatica and especially my roommates Gerlof, Francisco, and Tamas. Francisco deserves a special mentioning in any Alfa-Informatica thesis for being the printer guy. Then there were the cooking club evenings with Leonoor, Tanja, and Menno, which I really enjoyed. Menno and I also took part in several running events, and to much critical acclaim! I want to thank Francisco en Tamas a second time for being so kind as to accept the role of “paranimfs” at my defence. Thanks go out to Stasinos for the stylefile used to format this dissertation.

I want to thank Jan Daciuk for writing and making available his finite-state automata package of which my own programs make heavy use, Mark-Jan Nederhof for taking the time to read and suggest improvements on part of my thesis, and Hans van Halteren for supplying me with data used in his own work, allowing me to perform additional comparative experiments.

Finally I thank my parents for their support during those four years.

Contents

1	Introduction	1
1.1	Ambiguity in language	1
1.2	Natural language parsing by computer	2
1.3	Overview of the dissertation	3
2	Finite-state syntactic analysis	5
2.1	Finite-state automata	5
2.1.1	Informal definition	6
2.1.2	Formal definition	10
2.2	Motivation for the finite-state paradigm	11
2.2.1	General motivation for finite-state techniques	11
2.2.2	Finite-state techniques in syntactic analysis	12
2.3	Methods of finite-state approximation	16
2.3.1	Approximation through RTNs	18
2.3.2	Approximation through grammar transformation	21
2.3.3	Approximation through restricted stack	22
2.3.4	Approximation using n-gram models	28
2.4	Discussion of methods of approximation	29
2.4.1	Computational complexity	29
2.4.2	Systems beyond context-free power	29
2.5	Discussion	30
3	Inferring a stochastic finite-state model	33
3.1	Grammatical inference	33
3.2	Inferring n-gram models	34
3.2.1	Estimating probabilities through counting	34
3.2.2	Markov assumptions	35
3.2.3	Examples	37
3.2.4	Visible versus hidden states	38
3.3	Hidden Markov models of language	38
3.3.1	Informal definition	39

3.3.2	Examples	40
3.3.3	Formal definition	41
3.4	POS-tagging model	42
3.5	Tasks related to HMMs	44
3.5.1	Learning an HMM from annotated observations	45
3.5.2	Computing forward and backward probabilities	46
3.5.3	Computing the probability of an observation	49
3.5.4	Finding the sequence of hidden states	49
3.6	Other approaches to inference	51
3.6.1	Merged prefix tree	51
3.6.2	Merged hidden Markov model	52
3.6.3	Neural network based DFA	53
3.7	Conclusion	53
4	Reducing lexical ambiguity using a tagger	55
4.1	Approximation through inference	55
4.2	The Alpino wide-coverage parser	57
4.2.1	Grammar	57
4.2.2	Robust parsing	58
4.3	Problems in wide-coverage parsing	60
4.4	Using a POS tagger as a filter	61
4.4.1	Mapping lexical categories to POS tags	62
4.4.2	The HMM tagger	63
4.4.3	Smoothing	66
4.4.4	Stand-alone results	68
4.4.5	Larger amounts of training data	71
4.5	Using the filter in the parser	72
4.5.1	Experimental results	72
4.5.2	Discussion	73
4.6	Conclusion	77
5	Modeling global context	79
5.1	A restriction of HMM tagging	79
5.1.1	A systematic error in tagging Dutch	80
5.2	Extending the model	81
5.2.1	Standard model	82
5.2.2	Extended model	83
5.3	Tagging experiment	85
5.3.1	Tagger	86
5.3.2	Method	88
5.3.3	Data	88

5.3.4	Results	89
5.3.5	Discussion of results	90
5.3.6	Using the model in the parser	92
5.4	Other applications	93
5.5	Other work on extending n-gram models	94
5.6	Conclusion	95
6	Reducing structural ambiguity using a chunker	97
6.1	Chunking	99
6.2	Chunking as tagging	99
6.3	Different methods	100
6.3.1	Naive two-step	101
6.3.2	Extended two-step	101
6.3.3	Naive combined	102
6.3.4	Combined	103
6.4	Combined tagging and chunking	104
6.4.1	POS tagging	104
6.4.2	BaseNP chunking	105
6.5	Reducing structural ambiguity	109
6.5.1	Chunk data useful to the parser	109
6.5.2	Assigning brackets based on IOB2 tags	110
6.5.3	Stand-alone innermost NP bracketing	112
6.5.4	Supplying brackets to the parser	115
6.6	Conclusion	117
7	Conclusion	119
7.1	Approximation through inference	119
7.2	Reduction of lexical ambiguity	120
7.3	Reduction of structural ambiguity	121
	Bibliography	122
	Summary	132
	Samenvatting	139
	Groningen Dissertations in Linguistics	146

