

Pierre M. Nugues

An Introduction to Language Processing with Perl and Prolog

An Outline of Theories, Implementation, and Application
with Special Consideration of English, French, and German

With 153 Figures and 192 Tables

Contents

1	An Overview of Language Processing	1
1.1	Linguistics and Language Processing	1
1.2	Applications of Language Processing	2
1.3	The Different Domains of Language Processing	3
1.4	Phonetics	4
1.5	Lexicon and Morphology	6
1.6	Syntax	8
1.6.1	Syntax as Defined by Noam Chomsky	8
1.6.2	Syntax as Relations and Dependencies	10
1.7	Semantics	11
1.8	Discourse and Dialogue	14
1.9	Why Speech and Language Processing Are Difficult	14
1.9.1	Ambiguity	15
1.9.2	Models and Their Implementation	16
1.10	An Example of Language Technology in Action: the Persona Project	17
1.10.1	Overview of Persona	17
1.10.2	The Persona's Modules	18
1.11	Further Reading	19
2	Corpus Processing Tools	23
2.1	Corpora	23
2.1.1	Types of Corpora	23
2.1.2	Corpora and Lexicon Building	24
2.1.3	Corpora as Knowledge Sources for the Linguist	26
2.2	Finite-State Automata	27
2.2.1	A Description	27
2.2.2	Mathematical Definition of Finite-State Automata	28
2.2.3	Finite-State Automata in Prolog	29
2.2.4	Deterministic and Nondeterministic Automata	30
2.2.5	Building a Deterministic Automata from a Nondeterministic One	31

2.2.6	Searching a String with a Finite-State Automaton	31
2.2.7	Operations on Finite-State Automata	33
2.3	Regular Expressions	35
2.3.1	Repetition Metacharacters	36
2.3.2	The Longest Match	37
2.3.3	Character Classes	38
2.3.4	Nonprintable Symbols or Positions	39
2.3.5	Union and Boolean Operators	41
2.3.6	Operator Combination and Precedence	41
2.4	Programming with Regular Expressions	42
2.4.1	Perl	42
2.4.2	Matching	42
2.4.3	Substitutions	43
2.4.4	Translating Characters	44
2.4.5	String Operators	44
2.4.6	Back References	45
2.5	Finding Concordances	46
2.5.1	Concordances in Prolog	46
2.5.2	Concordances in Perl	48
2.6	Approximate String Matching	50
2.6.1	Edit Operations	50
2.6.2	Minimum Edit Distance	51
2.6.3	Searching Edits in Prolog	54
2.7	Further Reading	55
3	Encoding, Entropy, and Annotation Schemes	59
3.1	Encoding Texts	59
3.2	Character Sets	60
3.2.1	Representing Characters	60
3.2.2	Unicode	61
3.2.3	The Unicode Encoding Schemes	63
3.3	Locales and Word Order	66
3.3.1	Presenting Time, Numerical Information, and Ordered Words	66
3.3.2	The Unicode Collation Algorithm	67
3.4	Markup Languages	69
3.4.1	A Brief Background	69
3.4.2	An Outline of XML	69
3.4.3	Writing a DTD	71
3.4.4	Writing an XML Document	74
3.4.5	Namespaces	75
3.5	Codes and Information Theory	76
3.5.1	Entropy	76
3.5.2	Huffman Encoding	77
3.5.3	Cross Entropy	80

3.5.4	Perplexity and Cross Perplexity	81
3.6	Entropy and Decision Trees	82
3.6.1	Decision Trees	82
3.6.2	Inducing Decision Trees Automatically	82
3.7	Further Reading	84
4	Counting Words	87
4.1	Counting Words and Word Sequences	87
4.2	Words and Tokens	87
4.2.1	What Is a Word?	87
4.2.2	Breaking a Text into Words: Tokenization	88
4.3	Tokenizing Texts	89
4.3.1	Tokenizing Texts in Prolog	89
4.3.2	Tokenizing Texts in Perl	91
4.4	<i>N</i> -grams	92
4.4.1	Some Definitions	92
4.4.2	Counting Unigrams in Prolog	93
4.4.3	Counting Unigrams with Perl	93
4.4.4	Counting Bigrams with Perl	95
4.5	Probabilistic Models of a Word Sequence	95
4.5.1	The Maximum Likelihood Estimation	95
4.5.2	Using ML Estimates with <i>Nineteen Eighty-Four</i>	97
4.6	Smoothing <i>N</i> -gram Probabilities	99
4.6.1	Sparse Data	99
4.6.2	Laplace's Rule	100
4.6.3	Good-Turing Estimation	101
4.7	Using <i>N</i> -grams of Variable Length	102
4.7.1	Linear Interpolation	103
4.7.2	Back-off	104
4.8	Quality of a Language Model	104
4.8.1	Intuitive Presentation	104
4.8.2	Entropy Rate	105
4.8.3	Cross Entropy	105
4.8.4	Perplexity	106
4.9	Collocations	106
4.9.1	Word Preference Measurements	107
4.9.2	Extracting Collocations with Perl	108
4.10	Application: Retrieval and Ranking of Documents on the Web	109
4.11	Further Reading	111
5	Words, Parts of Speech, and Morphology	113
5.1	Words	113
5.1.1	Parts of Speech	113
5.1.2	Features	114
5.1.3	Two Significant Parts of Speech: The Noun and the Verb ..	115

5.2	Lexicons	117
5.2.1	Encoding a Dictionary	119
5.2.2	Building a Trie in Prolog	121
5.2.3	Finding a Word in a Trie	123
5.3	Morphology	123
5.3.1	Morphemes	123
5.3.2	Morphs	124
5.3.3	Inflection and Derivation	125
5.3.4	Language Differences	129
5.4	Morphological Parsing	130
5.4.1	Two-Level Model of Morphology	130
5.4.2	Interpreting the Morphs	131
5.4.3	Finite-State Transducers	131
5.4.4	Conjugating a French Verb	133
5.4.5	Prolog Implementation	134
5.4.6	Ambiguity	136
5.4.7	Operations on Finite-State Transducers	137
5.5	Morphological Rules	138
5.5.1	Two-Level Rules	138
5.5.2	Rules and Finite-State Transducers	139
5.5.3	Rule Composition: An Example with French Irregular Verbs	141
5.6	Application Examples	142
5.7	Further Reading	142
6	Part-of-Speech Tagging Using Rules	147
6.1	Resolving Part-of-Speech Ambiguity	147
6.1.1	A Manual Method	147
6.1.2	Which Method to Use to Automatically Assign Parts of Speech	147
6.2	Tagging with Rules	149
6.2.1	Brill's Tagger	149
6.2.2	Implementation in Prolog	151
6.2.3	Deriving Rules Automatically	153
6.2.4	Confusion Matrices	154
6.3	Unknown Words	154
6.4	Standardized Part-of-Speech Tagsets	156
6.4.1	Multilingual Part-of-Speech Tags	156
6.4.2	Parts of Speech for English	158
6.4.3	An Annotation Scheme for Swedish	160
6.5	Further Reading	162

7	Part-of-Speech Tagging Using Stochastic Techniques	163
7.1	The Noisy Channel Model	163
7.1.1	Presentation	163
7.1.2	The N -gram Approximation	164
7.1.3	Tagging a Sentence	165
7.1.4	The Viterbi Algorithm: An Intuitive Presentation	166
7.2	Markov Models	167
7.2.1	Markov Chains	167
7.2.2	Hidden Markov Models	169
7.2.3	Three Fundamental Algorithms to Solve Problems with HMMs	170
7.2.4	The Forward Procedure	171
7.2.5	Viterbi Algorithm	173
7.2.6	The Backward Procedure	174
7.2.7	The Forward–Backward Algorithm	175
7.3	Tagging with Decision Trees	177
7.4	Unknown Words	179
7.5	An Application of the Noisy Channel Model: Spell Checking	179
7.6	A Second Application: Language Models for Machine Translation	180
7.6.1	Parallel Corpora	180
7.6.2	Alignment	181
7.6.3	Translation	183
7.7	Further Reading	184
8	Phrase-Structure Grammars in Prolog	185
8.1	Using Prolog to Write Phrase-Structure Grammars	185
8.2	Representing Chomsky’s Syntactic Formalism in Prolog	185
8.2.1	Constituents	185
8.2.2	Tree Structures	186
8.2.3	Phrase-Structure Rules	187
8.2.4	The Definite Clause Grammar (DCG) Notation	188
8.3	Parsing with DCGs	190
8.3.1	Translating DCGs into Prolog Clauses	190
8.3.2	Parsing and Generation	192
8.3.3	Left-Recursive Rules	193
8.4	Parsing Ambiguity	194
8.5	Using Variables	196
8.5.1	Gender and Number Agreement	196
8.5.2	Obtaining the Syntactic Structure	198
8.6	Application: Tokenizing Texts Using DCG Rules	200
8.6.1	Word Breaking	200
8.6.2	Recognition of Sentence Boundaries	201
8.7	Semantic Representation	202
8.7.1	λ -Calculus	202
8.7.2	Embedding λ -Expressions into DCG Rules	203

8.7.3	Semantic Composition of Verbs	205
8.8	An Application of Phrase-Structure Grammars and a Worked Example	206
8.9	Further Reading	210
9	Partial Parsing	213
9.1	Is Syntax Necessary?	213
9.2	Word Spotting and Template Matching	213
9.2.1	ELIZA	213
9.2.2	Word Spotting in Prolog	214
9.3	Multiword Detection	217
9.3.1	Multiwords	217
9.3.2	A Standard Multiword Annotation	217
9.3.3	Detecting Multiwords with Rules	219
9.3.4	The Longest Match	219
9.3.5	Running the Program	220
9.4	Noun Groups and Verb Groups	222
9.4.1	Groups Versus Recursive Phrases	223
9.4.2	DCG Rules to Detect Noun Groups	223
9.4.3	DCG Rules to Detect Verb Groups	225
9.4.4	Running the Rules	226
9.5	Group Detection as a Tagging Problem	227
9.5.1	Tagging Gaps	227
9.5.2	Tagging Words	228
9.5.3	Using Symbolic Rules	229
9.5.4	Using Statistical Tagging	229
9.6	Cascading Partial Parsers	230
9.7	Elementary Analysis of Grammatical Functions	231
9.7.1	Main Functions	231
9.7.2	Extracting Other Groups	232
9.8	An Annotation Scheme for Groups in French	235
9.9	Application: The FASTUS System	237
9.9.1	The Message Understanding Conferences	237
9.9.2	The Syntactic Layers of the FASTUS System	238
9.9.3	Evaluation of Information Extraction Systems	239
9.10	Further Reading	240
10	Syntactic Formalisms	243
10.1	Introduction	243
10.2	Chomsky's Grammar in Syntactic Structures	244
10.2.1	Constituency: A Formal Definition	244
10.2.2	Transformations	246
10.2.3	Transformations and Movements	248
10.2.4	Gap Threading	248
10.2.5	Gap Threading to Parse Relative Clauses	250

10.3	Standardized Phrase Categories for English	252
10.4	Unification-Based Grammars	254
10.4.1	Features	254
10.4.2	Representing Features in Prolog	255
10.4.3	A Formalism for Features and Rules	257
10.4.4	Features Organization	258
10.4.5	Features and Unification	260
10.4.6	A Unification Algorithm for Feature Structures	261
10.5	Dependency Grammars	263
10.5.1	Presentation	263
10.5.2	Properties of a Dependency Graph	266
10.5.3	Valence	268
10.5.4	Dependencies and Functions	270
10.6	Further Reading	273
11	Parsing Techniques	277
11.1	Introduction	277
11.2	Bottom-up Parsing	278
11.2.1	The Shift-Reduce Algorithm	278
11.2.2	Implementing Shift-Reduce Parsing in Prolog	279
11.2.3	Differences Between Bottom-up and Top-down Parsing ..	281
11.3	Chart Parsing	282
11.3.1	Backtracking and Efficiency	282
11.3.2	Structure of a Chart	282
11.3.3	The Active Chart	283
11.3.4	Modules of an Earley Parser	285
11.3.5	The Earley Algorithm in Prolog	288
11.3.6	The Earley Parser to Handle Left-Recursive Rules and Empty Symbols	293
11.4	Probabilistic Parsing of Context-Free Grammars	294
11.5	A Description of PCFGs	294
11.5.1	The Bottom-up Chart	297
11.5.2	The Cocke-Younger-Kasami Algorithm in Prolog	298
11.5.3	Adding Probabilities to the CYK Parser	300
11.6	Parser Evaluation	301
11.6.1	Constituency-Based Evaluation	301
11.6.2	Dependency-Based Evaluation	302
11.6.3	Performance of PCFG Parsing	302
11.7	Parsing Dependencies	303
11.7.1	Dependency Rules	304
11.7.2	Extending the Shift-Reduce Algorithm to Parse Dependencies	305
11.7.3	Nivre's Parser in Prolog	306
11.7.4	Finding Dependencies Using Constraints	309
11.7.5	Parsing Dependencies Using Statistical Techniques	310

11.8	Further Reading	313
12	Semantics and Predicate Logic	317
12.1	Introduction	317
12.2	Language Meaning and Logic: An Illustrative Example	317
12.3	Formal Semantics	319
12.4	First-Order Predicate Calculus to Represent the State of Affairs ...	319
12.4.1	Variables and Constants	320
12.4.2	Predicates	320
12.5	Querying the Universe of Discourse	322
12.6	Mapping Phrases onto Logical Formulas	322
12.6.1	Representing Nouns and Adjectives	323
12.6.2	Representing Noun Groups	324
12.6.3	Representing Verbs and Prepositions	324
12.7	The Case of Determiners	325
12.7.1	Determiners and Logic Quantifiers	325
12.7.2	Translating Sentences Using Quantifiers	326
12.7.3	A General Representation of Sentences	327
12.8	Compositionality to Translate Phrases to Logical Forms	329
12.8.1	Translating the Noun Phrase	329
12.8.2	Translating the Verb Phrase	330
12.9	Augmenting the Database and Answering Questions	331
12.9.1	Declarations	332
12.9.2	Questions with Existential and Universal Quantifiers	332
12.9.3	Prolog and Unknown Predicates	334
12.9.4	Other Determiners and Questions	335
12.10	Application: The Spoken Language Translator	335
12.10.1	Translating Spoken Sentences	335
12.10.2	Compositionality Semantics	336
12.10.3	Semantic Representation Transfer	338
12.11	Further Reading	340
13	Lexical Semantics	343
13.1	Beyond Formal Semantics	343
13.1.1	<i>La langue et la parole</i>	343
13.1.2	Language and the Structure of the World	343
13.2	Lexical Structures	344
13.2.1	Some Basic Terms and Concepts	344
13.2.2	Ontological Organization	344
13.2.3	Lexical Classes and Relations	345
13.2.4	Semantic Networks	347
13.3	Building a Lexicon	347
13.3.1	The Lexicon and Word Senses	349
13.3.2	Verb Models	350
13.3.3	Definitions	351

13.4	An Example of Exhaustive Lexical Organization: WordNet	352
13.4.1	Nouns	353
13.4.2	Adjectives	354
13.4.3	Verbs	355
13.5	Automatic Word Sense Disambiguation	356
13.5.1	Senses as Tags	356
13.5.2	Associating a Word with a Context	357
13.5.3	Guessing the Topic	357
13.5.4	Naïve Bayes	358
13.5.5	Using Constraints on Verbs	359
13.5.6	Using Dictionary Definitions	359
13.5.7	An Unsupervised Algorithm to Tag Senses	360
13.5.8	Senses and Languages	362
13.6	Case Grammars	363
13.6.1	Cases in Latin	363
13.6.2	Cases and Thematic Roles	364
13.6.3	Parsing with Cases	365
13.6.4	Semantic Grammars	366
13.7	Extending Case Grammars	367
13.7.1	FrameNet	367
13.7.2	A Statistical Method to Identify Semantic Roles	368
13.8	An Example of Case Grammar Application: EVAR	371
13.8.1	EVAR's Ontology and Syntactic Classes	371
13.8.2	Cases in EVAR	373
13.9	Further Reading	373
14	Discourse	377
14.1	Introduction	377
14.2	Discourse: A Minimalist Definition	378
14.2.1	A Description of Discourse	378
14.2.2	Discourse Entities	378
14.3	References: An Application-Oriented View	379
14.3.1	References and Noun Phrases	379
14.3.2	Finding Names – Proper Nouns	380
14.4	Coreference	381
14.4.1	Anaphora	381
14.4.2	Solving Coreferences in an Example	382
14.4.3	A Standard Coreference Annotation	383
14.5	References: A More Formal View	384
14.5.1	Generating Discourse Entities: The Existential Quantifier	384
14.5.2	Retrieving Discourse Entities: Definite Descriptions	385
14.5.3	Generating Discourse Entities: The Universal Quantifier	386
14.6	Centering: A Theory on Discourse Structure	387
14.7	Solving Coreferences	388

14.7.1	A Simplistic Method: Using Syntactic and Semantic Compatibility	389
14.7.2	Solving Coreferences with Shallow Grammatical Information	390
14.7.3	Salience in a Multimodal Context	391
14.7.4	Using a Machine-Learning Technique to Resolve Coreferences	391
14.7.5	More Complex Phenomena: Ellipses	396
14.8	Discourse and Rhetoric	396
14.8.1	Ancient Rhetoric: An Outline	397
14.8.2	Rhetorical Structure Theory	397
14.8.3	Types of Relations	399
14.8.4	Implementing Rhetorical Structure Theory	400
14.9	Events and Time	401
14.9.1	Events	403
14.9.2	Event Types	404
14.9.3	Temporal Representation of Events	404
14.9.4	Events and Tenses	406
14.10	TimeML, an Annotation Scheme for Time and Events	407
14.11	Further Reading	409
15	Dialogue	411
15.1	Introduction	411
15.2	Why a Dialogue?	411
15.3	Simple Dialogue Systems	412
15.3.1	Dialogue Systems Based on Automata	412
15.3.2	Dialogue Modeling	413
15.4	Speech Acts: A Theory of Language Interaction	414
15.5	Speech Acts and Human-Machine Dialogue	417
15.5.1	Speech Acts as a Tagging Model	417
15.5.2	Speech Acts Tags Used in the SUNDIAL Project	418
15.5.3	Dialogue Parsing	419
15.5.4	Interpreting Speech Acts	421
15.5.5	EVAR: A Dialogue Application Using Speech Acts	422
15.6	Taking Beliefs and Intentions into Account	423
15.6.1	Representing Mental States	425
15.6.2	The STRIPS Planning Algorithm	427
15.6.3	Causality	429
15.7	Further Reading	430
A	An Introduction to Prolog	433
A.1	A Short Background	433
A.2	Basic Features of Prolog	434
A.2.1	Facts	434
A.2.2	Terms	435

A.2.3	Queries	437
A.2.4	Logical Variables	437
A.2.5	Shared Variables	438
A.2.6	Data Types in Prolog	439
A.2.7	Rules	440
A.3	Running a Program	442
A.4	Unification	443
A.4.1	Substitution and Instances	443
A.4.2	Terms and Unification	444
A.4.3	The Herbrand Unification Algorithm	445
A.4.4	Example	445
A.4.5	The Occurs-Check	446
A.5	Resolution	447
A.5.1	Modus Ponens	447
A.5.2	A Resolution Algorithm	447
A.5.3	Derivation Trees and Backtracking	448
A.6	Tracing and Debugging	450
A.7	Cuts, Negation, and Related Predicates	452
A.7.1	Cuts	452
A.7.2	Negation	453
A.7.3	The once/1 Predicate	454
A.8	Lists	455
A.9	Some List-Handling Predicates	456
A.9.1	The member/2 Predicate	456
A.9.2	The append/3 Predicate	457
A.9.3	The delete/3 Predicate	458
A.9.4	The intersection/3 Predicate	458
A.9.5	The reverse/2 Predicate	459
A.9.6	The Mode of an Argument	459
A.10	Operators and Arithmetic	460
A.10.1	Operators	460
A.10.2	Arithmetic Operations	460
A.10.3	Comparison Operators	462
A.10.4	Lists and Arithmetic: The length/2 Predicate	463
A.10.5	Lists and Comparison: The quicksort/2 Predicate	463
A.11	Some Other Built-in Predicates	464
A.11.1	Type Predicates	464
A.11.2	Term Manipulation Predicates	465
A.12	Handling Run-Time Errors and Exceptions	466
A.13	Dynamically Accessing and Updating the Database	467
A.13.1	Accessing a Clause: The clause/2 Predicate	467
A.13.2	Dynamic and Static Predicates	468
A.13.3	Adding a Clause: The asserta/1 and assertz/1 Predicates	468

A.13.4 Removing Clauses: The <code>retract/1</code> and <code>abolish/2</code> Predicates	469
A.13.5 Handling Unknown Predicates	470
A.14 All-Solutions Predicates	470
A.15 Fundamental Search Algorithms	471
A.15.1 Representing the Graph	472
A.15.2 Depth-First Search	473
A.15.3 Breadth-First Search	474
A.15.4 A* Search	475
A.16 Input/Output	476
A.16.1 Reading and Writing Characters with Edinburgh Prolog ...	476
A.16.2 Reading and Writing Terms with Edinburgh Prolog	476
A.16.3 Opening and Closing Files with Edinburgh Prolog	477
A.16.4 Reading and Writing Characters with Standard Prolog ...	478
A.16.5 Reading and Writing Terms with Standard Prolog	479
A.16.6 Opening and Closing Files with Standard Prolog	479
A.16.7 Writing Loops	480
A.17 Developing Prolog Programs	481
A.17.1 Presentation Style	481
A.17.2 Improving Programs	482
Index	487
References	497