

Theory and Applications of Natural Language Processing

Series Editors:

Graeme Hirst (Textbooks)

Eduard Hovy (Edited volumes)

Mark Johnson (Monographs)

Aims and Scope

The field of Natural Language Processing (NLP) has expanded explosively over the past decade: growing bodies of available data, novel fields of applications, emerging areas and new connections to neighboring fields have all led to increasing output and to diversification of research.

“Theory and Applications of Natural Language Processing” is a series of volumes dedicated to selected topics in NLP and Language Technology. It focuses on the most recent advances in all areas of the computational modeling and processing of speech and text across languages and domains. Due to the rapid pace of development, the diversity of approaches and application scenarios are scattered in an ever-growing mass of conference proceedings, making entry into the field difficult for both students and potential users. Volumes in the series facilitate this first step and can be used as a teaching aid, advanced-level information resource or a point of reference.

The series encourages the submission of research monographs, contributed volumes and surveys, lecture notes and textbooks covering research frontiers on all relevant topics, offering a platform for the rapid publication of cutting-edge research as well as for comprehensive monographs that cover the full range of research on specific problem areas.

The topics include applications of NLP techniques to gain insights into the use and functioning of language, as well as the use of language technology in applications that enable communication, knowledge management and discovery such as natural language generation, information retrieval, question-answering, machine translation, localization and related fields.

The books are available in printed and electronic (e-book) form:

- * Downloadable on your PC, e-reader or iPad
- * Enhanced by Electronic Supplementary Material, such as algorithms, demonstrations, software, images and videos
- * Available online within an extensive network of academic and corporate R&D libraries worldwide
- * Never out of print thanks to innovative print-on-demand services
- * Competitively priced print editions for eBook customers thanks to MyCopy service <http://www.springer.com/librarians/e-content/mycopy>

For other titles published in this series, go to
www.springer.com/series/8899

Slav Petrov

Coarse-to-Fine Natural Language Processing

Foreword by Eugene Charniak

Slav Petrov
Google
slav@petrovi.de

ISSN 2192-032X e-ISSN 2192-0338
ISBN 978-3-642-22742-4 e-ISBN 978-3-642-22743-1
DOI 10.1007/978-3-642-22743-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011939484

© Springer-Verlag Berlin Heidelberg 2012

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: deblik, Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To my family

Foreword

Grammars for natural languages show how sentences (and their meaning) are built up out of smaller pieces. Syntactic parsing is the task of applying a grammar to a string of words (a sentence) in order to reconstruct this structure. For example, “The dog thought there was day-old food in his dish” has a sub-structure “there was day-old food in his dish” which in turn contains structures like “day-old food.” Before we can build the meaning of the whole we must at least identify the parts from which it is built. This is what parsing gives us.

As with most all areas of natural-language processing (NLP) parsing research has greatly benefited from the statistical revolution — the process of absorbing statistical learning techniques into NLP that began about twenty five years ago. Prior to that time we had no parser that could, say, assign a plausible structure for every sentence in your local newspaper. Now you can download several good ones on the web.

From the outside the result has looked sort of like a Moore’s law scenario. Every few years parsers got more accurate, or much more efficient, or both. From inside, however, things looked quite different. At more than one occasion we in the community had no idea where the next improvement would come from and some thought that we had, perhaps, reached the end of the road. The last time the improvement came from Slav Petrov and the ideas in this monograph. The embodiment of these ideas is the “Berkeley Parser.”

The best parsers models are all “supervised,” e.g., we have a corpus of sentences, in the case here the so-called “Penn tree-bank” where sentences have been analyzed by people so for each sentence has been broken down into a tree structure of components. A computer learns to parse new sentences by collecting statics from the training data that (we hope) reflect generalizations about a particular language, in this case English. We then recast the parsing problem as one of applied statistics and probability — find the most probable parse for the sentences according the the probabilities already obtained from the corpus.

To over simplify, until Slav’s work, the best parsers could be thought of as word-based — rules should be based upon the words found in their examples. A paradigmatic case wold be, say, the use of the prepositional phrase “out of ...”

when talking about removing something by “washing”, but not by, say, “sanding.” Unfortunately the number of words in English is very large (really unbounded), so this data would be missing many crucial word-grammar combinations. In this cases the parser would “back off” and look from grammar rules ignoring the particular words in question.

The Berkeley parser, however bases rules not on words, but on sets of words. The “coarse to fine” of the title refers to the granularity of these sets. So the counter claim would be that “washing” is not unique here, but is rather one of a group of words that also include “scrubbing” and in some cases “flooding” (I flooded the cinder out of my eye). Unfortunately such groups can be quite idiosyncratic, so it might be that we are still better off at the word level. Indeed, the two methods can be thought of as two ends of a continuum, and perhaps future work can now combine the approaches. But until the Berkeley parser we did have a good concrete example of this second approach.

Furthermore, for anyone with a good machine learning background, once you see how this parser works, it makes immediate sense. Thus for people like me, at least, Slav’s work is very easy to read. Perhaps I am not a “typical” person, but take it from me, there are a lot of papers in my research area that I do not find so easy.

Thus I strongly recommend Slav’s work to you. It is major advance in the area of syntactic parsing, and a great advertisement for the superiority of the machine-learning approach to the field.

Brown University

Eugene Charniak

Preface

This book is based on my homonymous PhD thesis filed at the University of California, Berkeley in 2009. It has been updated to reference new work that has happened since then. It has also been reformatted to fit this paper size.

Acknowledgements

This book would not have been possible without the support of many wonderful people.

First and foremost, I would like to thank my PhD advisor Dan Klein for his guidance throughout graduate school and for being a never ending source of support and energy. Dan's sense of aesthetics has shaped the way I see research and will hopefully stay ingrained in me throughout my career. Dan is unique in too many ways to list here, and I will always be indebted to him. Dan was the best advisor I could have ever asked for.

Graduate school would not have been the same without the Berkeley Natural Language Processing (NLP) group. Initially there were four members: Aria Haghighi, John DeNero, Percy Liang and Alexandre Bouchard-Cote. Adam Pauls, David Burkett, John Blitzer and Mohit Bansal joined the group while I was still there and many new faces have joined since I left, but the amazing spirit seems to have remained. Thank you all for a great time, be it at conferences or during our not so productive NLP lunches. I always enjoyed coming to the office and chatting with all of you, though I usually stayed home when I actually wanted to get work done. My plan was to work on a project and write a publication with each one of you, and we almost succeeded. I hope that we will stay in touch and continue our collaborations no matter how scattered around the world we are once we graduate.

I spent two great summers as an intern, working first with Mark Johnson, Chris Quirk and Bob Moore at Microsoft and then with Ryan McDonald and Gideon Mann at Google. I enjoyed my summer in New York so much that I joined Google after graduating.

I would also like to thank the NLP community at large and Eugene Charniak, David Chiang, Hal Daume, Jason Eisner, Tom Griffiths, Mary Harper, Michael Jordan, Dan Jurafsky, Kevin Knight, Chris Manning, Daniel Marcu, David McAllester, Fernando Pereira and Ben Taskar in particular. I enjoyed our numerous conversations so far, and look forward to many more in the future.

Finally, I would like to thank Carlo Tomasi for giving me the opportunity to work with him while I was an exchange student at Duke University and introducing me to research for the first time. Not only did I learn a tremendous amount from him

during that project, but it is also in part because of our work that I decided to pursue a PhD degree in the US.

And of course, thank you, dear reader. I feel honored and I hope you will find something useful in it. Besides my academic friends and colleagues, I would also like to thank my friends and family for helping me stay sane (at least to some extent) and providing balance in my life.

A big thank you is due to the two “fellas,” Juan Sebastian Lleras and Pascal Michailat. Living with them was a blast, especially after we survived the “cold war.” Graduate school would not have been the same without the two of them. Thank you JuanSe for being my best friend in Berkeley. I am grateful for the numerous trips that we did together (especially Colombia, Hawaii and Brazil), the uncountable soccer games that we played or watched together, and especially the many great conversations we had during those years. Thank you Pascal for literally being there with me from day one, when we met during the orientation for international students. I am grateful for the numerous ski trips, cooking sessions, and lots more. Whenever I make gallettes, I will be thinking about you.

Sports were a big part of my graduate school life and I would like to thank all the members of the Convex Optimizers and Invisible Hands. There were too many to list all, but Brad Howells and Ali Memarsdaeghi deserve special mention. I will not forget our titles and the many games that we played together.

Daniel Thalhammer, Victor Victorson, Arnaud Grunwald and Konstantinos Daskalakis were always there to explore restaurants, bars and clubs in the city and we had a lot of fun together. Thank you for dragging me out of Berkeley when I was feeling lazy, and for exploring the best places to eat good food, drink good (red) wine and listen to good electronic music.

Thanks also to my friends in Berlin, who always made me feel at home when I was there during the summer and over Christmas. We have known each other since high school and I hope we will always stay in touch.

Many thanks also to Natalie, from whom I learnt a lot about (and for) life. I grew a lot as a person during our relationship and I am grateful for having had you in my life.

My brother Anton deserves many thanks for being my best friend. It would be impossible to list all the things that I am grateful for, and I won't even attempt it. I know we will stay always close and that we have many good times ahead of us.

Last but not least, I would like to thank my parents Abi and Orlin for their infinite support and encouragement. I will always be grateful for the opportunities you gave me and Anton by moving from Bulgaria to Berlin. Thank you for raising us with a never ending quest for perfection, and teaching us to believe in ourselves and that we can achieve everything we want. Thank you for your love and thank you for making me who I am.

Contents

- 1 Introduction 1**
 - 1.1 Coarse-to-Fine Models 2
 - 1.2 Coarse-to-Fine Inference 4
- 2 Latent Variable Grammars for Natural Language Parsing 7**
 - 2.1 Introduction 7
 - 2.1.1 Experimental Setup 9
 - 2.2 Manual Grammar Refinement 10
 - 2.2.1 Vertical and Horizontal Markovization 11
 - 2.2.2 Additional Linguistic Refinements 12
 - 2.3 Generative Latent Variable Grammars 13
 - 2.3.1 Hierarchical Estimation 14
 - 2.3.2 Adaptive Refinement 16
 - 2.3.3 Smoothing 18
 - 2.3.4 An Infinite Alternative 19
 - 2.4 Inference 20
 - 2.4.1 Hierarchical Coarse-to-Fine Pruning 21
 - 2.4.2 Objective Functions for Parsing 26
 - 2.5 Additional Experiments 29
 - 2.5.1 Experimental Setup 29
 - 2.5.2 Baseline Grammar Variation 30
 - 2.5.3 Final Results WSJ 31
 - 2.5.4 Multilingual Parsing 31
 - 2.5.5 Corpus Variation 33
 - 2.5.6 Training Size Variation 34
 - 2.6 Analysis 35
 - 2.6.1 Lexical Subcategories 35
 - 2.6.2 Phrasal Subcategories 38
 - 2.6.3 Multilingual Analysis 40
 - 2.7 Summary and Future Work 42

3	Discriminative Latent Variable Grammars	47
3.1	Introduction	47
3.2	Log-Linear Latent Variable Grammars	47
3.3	Single-Scale Discriminative Grammars	50
3.3.1	Efficient Discriminative Estimation	50
3.3.2	Experiments	52
3.4	Multi-scale Discriminative Grammars	55
3.4.1	Hierarchical Refinement	56
3.4.2	Learning Sparse Multi-scale Grammars	58
3.4.3	Additional Features	61
3.4.4	Experiments	62
3.4.5	Analysis	65
3.5	Summary and Future Work	66
4	Structured Acoustic Models for Speech Recognition	69
4.1	Introduction	69
4.2	Learning	71
4.2.1	The Hand-Aligned Case	72
4.2.2	Splitting	73
4.2.3	Merging	73
4.2.4	Smoothing	74
4.2.5	The Automatically-Aligned Case	75
4.3	Inference	75
4.4	Experiments	76
4.4.1	Phone Recognition	77
4.4.2	Phone Classification	79
4.5	Analysis	79
4.6	Summary and Future Work	82
5	Coarse-to-Fine Machine Translation Decoding	83
5.1	Introduction	83
5.2	Coarse-to-Fine Decoding	85
5.2.1	Related Work	85
5.2.2	Language Model Projections	86
5.2.3	Multipass Decoding	87
5.3	Inversion Transduction Grammars	88
5.4	Learning Coarse Languages	90
5.4.1	Random Projections	90
5.4.2	Frequency Clustering	90
5.4.3	HMM Clustering	91
5.4.4	JCluster	91
5.4.5	Clustering Results	91
5.5	Experiments	92
5.5.1	Clustering	93
5.5.2	Spacing	94
5.5.3	Encoding Versus Order	94

Contentsxv

5.5.4

Final Results

95

5.5.5

Search Error Analysis

95

5.6

Summary and Future Work

97

6

Conclusions and Future Work.....

99

References.....

101

List of Figures

Fig. 1.1	(a) Syntactic parse trees model grammatical relationships. (b) Distribution of the internal structure of noun phrase (NP) constructions. Subject NPs use pronouns (PRPs) more frequently, suggesting that the independence assumptions in a naive context-free grammar are too strong	2
Fig. 1.2	Incrementally learned pronoun (PRP) subcategories for grammatical cases and placement. Categories are represented by the three most likely words	4
Fig. 1.3	Charts are used to depict the dynamic programming states in parsing. In coarse-to-fine parsing, the sentence is repeatedly re-parsed with increasingly refined grammars, pruning away low probability constituents. Finer grammars need to only consider only a fraction of the enlarged search space (the non-white chart items).	5
Fig. 1.4	There can be many syntactic parse trees for the same sentence. Here we are showing two that are both plausible because they correspond to different semantic meanings. In (a) statistics are used to solve a problem, while in (b) there is a problem with statistics that is being solved in an unspecified way. Usually there will be exactly one correct syntactic parse tree	6
Fig. 2.1	The original parse tree (a) gets binarized (b), and then either manually annotated (c) or refined with latent variables (d)	10
Fig. 2.2	Evolution of the DT tag during hierarchical splitting and merging. Shown are the top three words for each subcategory and their respective probability	15

Fig. 2.3	Hierarchical training leads to better parameter estimates. Merging reduces the grammar size significantly, while preserving the accuracy and enabling us to do more SM cycles. Parameter smoothing leads to even better accuracy for grammars with high complexity. The grammars range from extremely compact (an F_1 of 78% with only 147 nonterminal categories) to extremely accurate (an F_1 of 90.2% for our largest grammar with only 1,140 nonterminals)	18
Fig. 2.4	Hierarchical refinement proceeds top-down while projection recovers coarser grammars. The top word for the first refinements of the determiner tag (DT) is shown where space permits	22
Fig. 2.5	Bracket posterior probabilities (black = high) for the first sentence of our development set during coarse-to-fine pruning. Note that we compute the bracket posteriors at a much finer level but are showing the unlabeled posteriors for illustration purposes. No pruning is done at the finest level $G_6 = G$ but the minimum risk tree is returned instead	26
Fig. 2.6	Starting with a simple baseline grammar is advantageous because imposing too much initial structure causes overfragmentation in the long run	31
Fig. 2.7	Parsing accuracy starts dropping after five training iterations on the Brown corpus, while it is improving on the WSJ, indicating overfitting	33
Fig. 2.8	Parsing accuracy on the WSJ increases when more training data is used for learning the grammar. However, the last 30% of training data add only 0.3 in F_1 score ...	34
Fig. 2.9	Number of latent lexical subcategories determined by our split-merge procedure after 6 SM cycles	38
Fig. 2.10	Number of latent phrasal subcategories determined by our split-merge procedure after 6 SM cycles	39
Fig. 3.1	Average number of constructed constituents per sentence. Without pruning, the number of constituents grows exponentially and quickly leaves the plotted area.	53
Fig. 3.2	Multi-scale refinement of the $DT \rightarrow the$ production. The multi-scale grammar can be encoded much more compactly than the equally expressive single scale grammar by using only the shaded features along the fringe	56
Fig. 3.3	In multi-scale grammars, the categories exist at varying degrees of refinement. The grammar in this example enforces the correct usage of <i>she</i> and <i>her</i> , while allowing the use of <i>it</i> in both subject and object position	56

Fig. 3.4	A multi-scale chart can be used to efficiently compute inside/outside scores using productions of varying specificity	60
Fig. 3.5	Discriminative multi-scale grammars give similar parsing accuracies as generative split-merge grammars, while using an order of magnitude fewer rules.	63
Fig. 4.1	Comparison of the standard model to our model (here shown with $k = 4$ subphones per phone) for the word <i>dad</i> . The dependence of subphones across phones in our model is not shown, while the context clustering in the standard model is shown only schematically with 1,2,4,8 substates	70
Fig. 4.2	Iterative refinement of the /ih/ phone with 1,2,4,8 substates, as shown in (a), (b), (c), (d) above.....	71
Fig. 4.3	Phone recognition error for models of increasing size	78
Fig. 4.4	Phone confusion matrix. 76% of the substitutions fall within the shown classes	80
Fig. 4.5	Phone contexts and subphone structure. The // phone after three split-merge iterations is shown	81
Fig. 5.1	An example of hierarchical clustering of target language vocabulary (see Sect. 5.4). Even with a small number of clusters our divisive HMM clustering (Sect. 5.4.3) captures sensible syntactico-semantic classes	84
Fig. 5.2	Possible state projections π for the target noun phrase “the report for these states” using the clusters from Fig. 5.1. The number of bits used to encode the target language vocabulary is varied along the x-axis. The language model order is varied along the y-axis	86
Fig. 5.3	Example of state pruning in coarse-to-fine decoding using the language encoding projection (see Sect. 5.2.2). During the coarse one-bit word cluster pass, two of the four possible states are pruned. Every extension of the pruned one-bit states (indicated by the <i>grey shading</i>) are not explored during the two-bit word cluster pass	88
Fig. 5.4	Monotonic combination of two hypotheses during the inside pass involves scoring the fluency of the concatenation with the language model	90
Fig. 5.5	Results of coarse language model perplexity experiment (see Sect. 5.4.5). HMM and JClustering have lower perplexity than frequency and random clustering for all number of bits in the language encoding	92

Fig. 5.6	Coarse-to-fine decoding with HMM or JClustering coarse language models reduce decoding times while increasing accuracy	93
Fig. 5.7	Many passes with extremely simple language models produce the highest speed-ups	94
Fig. 5.8	A combination of order-based and encoding-based coarse-to-fine decoding yields the best results.....	95
Fig. 5.9	Coarse-to-fine decoding is faster than single pass decoding with a trigram language model and leads to better BLEU scores on all language pairs and for all parameter settings	96

List of Tables

Table 2.1	Horizontal and vertical Markovization: F_1 parsing accuracies and grammar sizes (number of nonterminals)	11
Table 2.2	Grammar sizes, parsing times and accuracies for latent variable grammars PCFGs with and without hierarchical coarse-to-fine parsing on our development set (1,578 sentences with 40 or less words from section 22 of the Penn Treebank)	26
Table 2.3	Different objectives for parsing with posteriors, yielding comparable results	28
Table 2.4	A 10-best list from our best G can be reordered as to maximize a given objective either using samples or, under some restricting assumptions, in closed form	29
Table 2.5	Treebanks and standard setups used in our experiments	30
Table 2.6	Generative latent variable grammars achieve state-of-the-art parsing performance on a variety of languages ...	32
Table 2.7	The most frequent three words in the subcategories of several part-of-speech tags	37
Table 2.8	The most frequent three productions of some latent phrasal subcategories	39
Table 2.9	The most frequent words, their translations and grammatical classification for several Bulgarian POS tags ({MASCULINE, FEMININE, NEUTER}-{SINGULAR, PLURAL})	41
Table 2.10	The most frequent words, their translations and grammatical classification for several Chinese POS tags ({MASCULINE, FEMININE, NEUTER}-{SINGULAR, PLURAL})	42
Table 2.11	The most frequent words, their translations and grammatical classification for several French POS tags ({MASCULINE, FEMININE}-{SINGULAR, PLURAL})	43

Table 2.12	The most frequent words, their translations and grammatical classification for several German POS tags ({MASCULINE, FEMININE, NEUTER}-{SINGULAR, PLURAL}-{NOMINATIVE, GENITIVE, DATIVE, ACCUSATIVE})	44
Table 2.13	The most frequent words, their translations and grammatical classification for several Italian POS tags ({MASCULINE, FEMININE, NEUTER}-{SINGULAR, PLURAL})	45
Table 3.1	Parsing times for different pruning regimes and grammar sizes	53
Table 3.2	L_1 regularization produces sparser solutions and requires fewer training iterations than L_2 regularization	54
Table 3.3	Discriminative training is superior to generative training for exact match and for F_1 -score when the same model and feature sets are used	55
Table 3.4	Our final test set parsing accuracies compared to the best previous work on English, French and German	64
Table 3.5	Complexity of highly split phrasal categories in generative and discriminative grammars. Note that subcategories are compared to production parameters, indicating that the number of parameters grows cubically in the number of subcategories for generative grammars, while growing linearly for multi-scale grammars	66
Table 3.6	Automatically learned suffixes with the highest weights for different languages and part-of-speech tags	66
Table 4.1	Phone recognition error rates on the TIMIT core test from Glass (2003)	78
Table 4.2	Phone classification error rates on the TIMIT core test	79
Table 4.3	Number of substates allocated per phone	82
Table 5.1	Test score analysis	97